ISSN 2942-1373

5

CMLDT publications

October 10 – 12, 2023

4th International JSXGraph Conference

Book of Abstracts

University of Bayreuth
Center for Mobile Learning with Digital Technology

Publisher University of Bayreuth
Center for Mobile Learning with Digital Technology (CMLDT)
Universitätsstraße 30
95447 Bayreuth
Germany
+49 921 55 3266

Series CMLDT publications

ISSN 2942-1373

Editor Carsten Miller, carsten.miller@uni-bayreuth.de
Alfred Wasserman, alfred.wassermann@uni-bayreuth.de

Layout Carsten Miller, carsten.miller@uni-bayreuth.de

Internet https://mobile-learning.uni-bayreuth.de/CMLDTpublications

Issue 5 **Year of publication** 2024

Title 4th International JSXGraph Conference – Book of Abstracts

UNIVERSITÄT BAYREUTH

CMLDT

4th International JSXGraph Conference 2023

University of Bayreuth
Center for Mobile Learning with Digital Technology
95440 Bayreuth
Germany

Conference

The 4th International JSXGraph Conference took place from 10th until 12th of October 2023. The online format encouraged fruitful discussion and collaboration among users from all over the world. The schedule respected the location and the timezone of the speakers. The conference was organized by Carsten Miller and Alfred Wassermann from the University of Bayreuth, Germany.

Conference topics

- Usage of JSXGraph
  - for learning / teaching
  - e-Learning environments: moodle, ilias, STACK, ...
  - dynamic visualizations
- Best practices
- Tools
- Presentation of new JSXGraph developments

Website

The abstracts of the talks at the 4th International JSXGraph Conference are also available on the JSXGraph website:

https://jsxgraph.org/conf2023

Videos

Most of the recorded videos of the talks can be found on JSXGraph's YouTube Channel:

https://www.youtube.com/@jsxgraph4224

Playlist "4th International JSXGraph Conference"

https://www.youtube.com/playlist?list=PLr10cPSXxWJdm6QKjZkLg34ol0yeGe8z3

Talks at the 4th International JSXGraph Conference

Implementing drag and drop functionality for JSXGraph in STACK questions using custom bindings

Bernhard Gailer
Ostbayerische Technische Hochschule
Amberg-Weiden, Fakultät Elektrotechnik, Medien und Informatik
Kaiser-Wilhelm-Ring 23
92224 Amberg
Germany
b.gailer@oth-aw.de

At the OTH Amberg-Weiden digital self-learning modules for engineering mathematics are developed in the context of the project IdeaL – Innovationsnetzwerk für digitale adaptive Lehre (Innovation network for digital adaptive teaching). The learning modules are developed by means of STACK questions in the learning management system Moodle. JSXGraph is used as dynamic geometry software to link symbolic and graphical content and to assess mathematical skills within STACK questions.

In our learning module for linear algebra, we use many JSXGraph applets to visualize vectors in the Euclidean plane. We often use interactive applets inside STACK questions in which the students have to move vectors to the correct position oder can play little animations. To assess the movement of vectors in STACK questions, the JSXGraph binding functions for input binding are used (https://docs.stack-assessment.org/en/Authoring/JSXGraph/#general-considerations-when-building-interactive-graphs).

In one particular question, we wanted the students to drag and drop different terms to the corresponding objects in one of the JSXGraph applets. Instead of using the moodle question type “drag and drop onto image” with an Image of the applet, we decided to implement this functionality inside JSXGraph and STACK. This has some benefits over using the drag and drop moodle question

type, as we can use drag and drop inside STACK questions. Furthermore, we can use dynamic JSXGraph applets together with the drag and drop feature instead of using a static image. Regarding the technical side, there are two main challenges for using the drag and drop feature:

- Using JSXGraph objects for the droppable items and creating dropzones Binding of the interactivity on the board and the STACK question inputs
- To meet these two challenges, a custom input binding together with the JSXGraph concept of “magnetized points” was used.

In the talk, the general approach for binding inputs to JSXGraph applets will be shown first. This includes the binding functions provided by STACK, as well as the approach for a custom binding and when to use custom bindings. Next, a question prototype from the learning module where the JSXGraph drag and drop functionality is implemented will be shown. In this question, we go over the code and look at the custom binding that is used for the drag and drop functionality. Next to the custom binding, the “JSXGraph way” for drag and drop using magnetized points is shown and explained.

The talk focuses on the technical aspects of implementing the drag and drop functionality using custom bindings in JSXGraph and is intended for JSXGraph developers who want to use custom input bindings in STACK questions together with JSXGraph to create new possibilities of interaction with their JSXGraph applets.

Rubik‘s cube with JSXGraph

Thomas Miller
Markgräfin-Wilhelmine-Gymnasium
Bayreuth
Germany
thomas.miller@uni-bayreuth.de

As part of a cooperation between the Markgräfin-Wilhelmine-Gymnasium Bayreuth and the University of Bayreuth, students attend various lectures and have to write a seminar paper.
I attended the lecture „WWW Grundlagen“, which covered HTML, CSS and other web technologies as well as JSXGraph. I combined these topics in my seminar paper. Therefore I developed a website where the user can take two photos of a scrambled Rubik’s Cube. A JSXGraph file with an adjustable net will be superimposed over the photos. The user drags the net above the photo. The colors of the cube will be analyzed and a solution algorithm will be executed. At the end the website will give a step by step solution (visualized with JSXGraph) to solve the Rubik‘s Cube.

JSXGraph Dynamic Editor

Cyrille Piatecki, Yvan Stroppa
Université d'Orléans, Faculté DEG, Rue de blois
45067 Orléans Cedex 2
France
cyrille.piatecki@univ-orleans.fr
yvan.stroppa1@univ-orleans.fr

Many scientific fields use diagrams to teach or to illustrate new ideas. Unfortunately, students are not able to efficiently follow the construction of this diagram and to draw them on their notes. This has a huge drawback: when, later, they must read those notes they often lack the true order of their construction which can lead to a misinterpretation.

In searching for a way to overcome this difficulty, we have found the JSXGRAPH Javascript library which not only help to draw those diagram but also to animate them. Unfortunately, the difficulty of the langage, due to the necessity to capture the different events which should be listen before to initiate a movement, forbade the most part of the scientist to construct the diagram of their dreams.

To try to overcome at least a little this difficulty, we have tried to construct JDE (Jsxgraph Dynamic Editor) a new editor which could facilitate not only the static conception of a diagram but which can cope also with some of its dynamical caracteristics.

Workshop I&II

Alfred Wassermann
University of Bayreuth
Germany

https://jsxgraph.org/conf2023/pdfs/workshop.pdf
https://jsxgraph.org/conf2023/pdfs/workshop_II.pdf

Using JSXGraph from F#: A functional visualization EDSL for end-users

Allister Beharry
University of London; 2 Russell Sq
London WC1B 5DN
United Kingdom
ab796@student.london.ac.uk

This talk is an expansion of the demo presented at ICFP FARM '23: A Functional EDSL for Mathematics Visualization That Compiles to JavaScript. We developed an F# interface to JSXGraph designed to be used both in F# library code and interactively in Jupyter notebooks for mathematics visualization. The Sylvester language shares some similarities to the approach used by the Penrose language to visualization in that one describes mathematical objects and operations like functions and vectors and matrices using the Sylvester notation which are then drawn and styled using user-specified properties and parameters, as opposed to requiring the user to manipulate vector graphics domain objects like points, lines, arrows, axes, function graphs etc. Visualization in Sylvester is targeted to educators and students and end-users and does not require vector graphics or web development domain knowledge. In this presentation we'll go into some more details of the Sylvester F#-JavaScript interface and JSXGraph: F# quotations and their use in code generation , the FunScript F#-JavaScript compiler, desiging a statically typed, functional, JSXGraph interface, and composing a higher-level object-oriented visualization API using JSXGraph for mathematical domain objects from vectors to economic functions.

sketchometry 2.0

Carsten Miller, Andreas Walter
University of Bayreuth
Germany
carsten.miller@uni-bayreuth.de
andreas.walter@uni-bayreuth.de

sketchometry is a gesture-based dynamic mathematics software that allows you to quickly and easily create and explore interactive constructions with your finger on tablets, smartphones or electronic whiteboards. The innovative gesture control of sketchometry turns mobile devices into an electronic sketching tool that can be used directly in the classroom. sketchometry is available free of charge as an app or directly as a live version in the browser:

https://sketchometry.org

Meclib: A library of objects for dynamic and interactive sketches of mechanical systems. Recent developments

Martin Kraska, Aaron Amran Bin Amiruddin
Technische Hochschule Brandenburg
University of Applied Sciences
Magdeburger Str. 50
14770 Brandenburg an der Havel
Germany
martin.kraska@th-brandenburg.de

Meclib (https://github.com/mkraska/meclib) is a JavaScript (JS) library for efficient configuration of embedded JSXGraph sketches in STACK, a question type of the Moodle learning management system. It can be used to efficiently produce visually consistent static and randomized dynamic system sketches for exercises in engineering mechanics. Question authors just specify the graphics objects using Maxima lists and thus don't need to master JS on top of the Maxima CAS system, which is the back-end of STACK. Dedicated objects with control points enable interactive modification or even generation and annotation of objects for input purposes. Important applications are interactive free body diagrams, moment and shear diagrams.

Still, the objects representing basic elements of mechanical systems were essentially static parametric objects with just the option to interactively activate and deactivate them in free body diagrams. Recently, these objects are under re-designed to provide animations and kinematic coupling to other elements in order to create animated and interactive mechanisms within STACK questions. The extended capabilities and scripting examples are shown in a life demonstration.

Interactive in 3D: Vector fields and Geometry

Wigand Rathmann
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Department Mathematik
wigand.rathmann@fau.de

Spatial concepts play an important role in the education of engineers. For the understanding of integrals over surfaces different concepts are necessary at the same time.
First is the representation of a parameterized surface in space. From the field of geometry, the triple product is used to calculate the content of the infinitesimal surface piece. For oriented surface integrals, the representation of a vector field on the surface supports the understanding and the interpretation of the integral to be calculated. Since summer 2022 it is also possible to create three-dimensional interactive diagrams with JSXGraph, which allows the realization of diagrams as described at the beginning.
The visualization of vector fields is also helpful for the understanding of ordinary differential equations. For visualization in the planes, the visualization of vector fields has been greatly simplified with version 1.6.0. JSXGraph also includes Runke-Kutta methods so that trajectories through given points can be easily drawn. This is possible in the plane as well as in space.
In my amount I present several applications from these areas.
The work for these graphs was funded by the ERASMUS+ project “Interactive digital assessments in mathematics”.

https://jsxgraph.org/conf2023/pdfs/rathmann2023talk.pdf

Graphical Exercises with JSXGraph using MUMIE

Andreas Maurischat
Integral Learning, Germany
andreas.maurischat@integral-learning.de

This talk is about graphical exercises in which the student solves the task by interacting with the graphics. Such tasks strengthen the students' ability to understand mathematical terms and concepts by actively linking the definitions and analytical expressions with the corresponding geometric objects.
We will present a small variety of our examples of such graphical exercises. The exercises are written in the MUMIE-framework which uses JSXGraph for its interactive visualizations, and can be embedded into Moodle or ILIAS via plugin.

Playful learning and assessment with KMap (using JSXGraph in shadowDOM)

Holger Engels
Valckenburgschule
Ulm
Germany
h.engels@valckenburgschule.de

KMap is a math learning platform with emphasis on clarity about the relations between mathematical subjects and a playful learning experience. The whole content of KMap is Creative Commons, so it's a truly open education resource. Besides lots of knowledge cards with interactive JSXGraph elements, KMap provides different types of interactive assessment components like dnd assignment, dnd fill-in, equation solving and JSXGraph components. As KMap is built with WebComponents, it requires these assessment components to be WebComponents as well. The talk explains some special characteristics of WebComponents and gives insights into how the integration of JSXGraph with WebComponents within KMap is actually achieved.

Interactive graphics in STACK questions

Hans Jakob Rivertz
Norwegian University of Science and Technology
Norway
hans.j.rivertz@ntnu.no

1. Best practice for writing to STACK input fields through JSXGraph.
   - I will talk about my current understanding of integrating JSXGraph in STACK. In Trondheim, we discovered that many STACK-exercises stopped working. I went through the exercises and found that they used some ad-hoc solutions.
   - Some of these JSXGraph code had in common that they extracted the ans1 by calling the elements directly instead of using the JSXGraph API. In a newer implementation of the interaction between STACK and JSXGraph this does not work anymore. There are many reasons why this was a not good practice, besides that it does not work anymore. One of these is that there might be several exercises on one page with the same name on the input-fields. Another is that there is a reason why APIs are deployed.
   - In addition, many of the exercises used the same input variable for the state and the answer. By using a JSON-string for the state and writing directly to ans1 as a string in the format given in the Input: ans1-field, it is much easier to write the potential response trees.
2. Interactive clickable exercises for graphs, logic and set theory.
   - I will also present some old and some new interactive exercises.