ISSN 2942-1373

4

October 4 – 6, 2022

3rd International JSXGraph Conference

Book of Abstracts

University of Bayreuth
Center for Mobile Learning with Digital Technology

CMLDT publications

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 4 **Year of publication** 2024

Title 3rd International JSXGraph Conference – Book of Abstracts

UNIVERSITÄT BAYREUTH

CMLDT

3rd International JSXGraph Conference 2022

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

Conference

The 3rd International JSXGraph Conference took place from 4th until 6th of October 2022. 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 3rd International JSXGraph Conference are also available on the JSXGraph website:

https://jsxgraph.org/conf2022

Videos

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

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

Playlist "3rd International JSXGraph Conference"

https://www.youtube.com/playlist?list=PLr10cPSXxWJdPubhJ63coFXMKJnEQaaoN

Talks at the 3rd International JSXGraph Conference

Interactive self-learning modules for engineering mathematics using STACK and JSXGraph

Stephan Bach, Johannes Knaut
Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik
Amberg
Germany
j.knaut@oth-aw.de

At the Ostbayerische Technische Hochschule Amberg-Weiden digital self-learning modules for engineering mathematics are developed within the project IdeaL – Innovationsnetzwerk für digitale adaptive Lehre (Innovation network for digital adaptive teaching). These are intended to offer students the opportunity to work on selected topics independently or to deepen their knowledge; for example, when no lecture is offered or for exam preparation. In addition, lecturers should be able to integrate the modules flexibly into their teaching, for example to create additional exercise opportunities.

The learning modules are implemented as courses in the learning management system Moodle and are composed of so-called interactive chapters, among other elements. These are intended to break up the division of the learning process into a phase of knowledge acquisition and a phase of practice and instead encourage students to process the learning content themselves at an early stage and on a regular basis. To achieve this goal, the interactive chapters are implemented using the Moodle activity Quiz. In the chapters, different, closely linked media elements are used. In addition to text-based content, interactive learning videos and digital STACK questions, this also includes numerous applets based on JSXGraph.

In the talk, various uses of JSXGraph will be demonstrated using the example of a pilot module on the topic of “complex numbers”. The resulting design decisions will be explained using concrete examples. Already in the preview

of the chapters, interactive applets motivate the following learning contents in a playful way. Within the chapters, JSXGraph is an essential tool for linking symbolic and graphical content. It also contributes substantially to the variety of learning and exercise possibilities. In the learning modules, a mandatory use of the JSXGraph applets is aimed at by linking them to videos or STACK questions or integrating them into the latter. In designing the applets, emphasis is placed on usability, and established theories of multimedia learning are taken into account, such as Mayer's Cognitive Theory of Multimedia Learning (2001).

In the past summer semester, the pilot module on complex numbers was tested with a small sample and essentially positive results. In the winter semester 2022/2023, this topic is now to be offered in a suitable course in purely digital form with the help of the learning module. The results are to be evaluated in a longitudinal study in order to obtain a direct comparison with classroom teaching.

Creating small simulations with JSXGraph and Stack to enhance learning activities

Luise Stromeyer
Hochschule für Technik und Wirtschaft
Berlin
Germany
luise.stromeyer@HTW-Berlin.de

Points, lines, circles, graphs ... JSXGraph has many elements that immediately make us think of mathematical constructions and relationships. Many teaching and learning activities are and were created with it and many things can be experienced by learners, which are otherwise difficult to convey.
At the Hochschule für Technik und Wirtschaft (HTW) Berlin these possibilities are also used in inner-mathematical contexts in the mathematics lectures for engineers.
But do WE have to limit ourselves to these inner-mathematical issues? Isn't there more possible? Students in engineering courses often want more practical relevance in the mathematics lecture. They want to know how to apply the mathematics they are learning. The connection and benefits of mathematics for engineering sciences are often not recognized and motivation for the subject decreases.
To counteract this, we at HTW Berlin are developing application-related tasks for mathematics lectures as part of the sub-project "Innovative digital MINT Curricula" (project "Curriculum Innovation Hub" funded by the "Innovation in der Hochschullehre" foundation). One challenge is, to develop meaningful applications for relevant and complex problems, so that 1st semester students can work with them. In my talk, I will use an example to show how JSXGraph and Stack can be used to embed applications in such a way that students can be well taken into the application context without long texts, but through small simulations.

In addition to the basic possibility of using JSXGraph beyond its mathematical contexts, I show how I use working with classes and randomized lists to effectively create as many different examples as possible, even in the context of simulations.

Online Learning Module Complex Numbers with Interactive Exercises and Visualizations

Karin Landenfeld
Faculty of Engineering and Computer Science
Hamburg University of Applied Sciences
Germany
karin.landenfeld@haw-hamburg.de

In our contribution we would like to present an online learning module about complex numbers developed at the Hamburg University of Applied Sciences, in which we use JSXGraph in various learning elements. These learning elements are used to illustrate the results, demonstrate the calculations or interactively apply what has been learned. See extended abstract (pdf) for detailed information.

The development of the learning module was funded by the Hamburg Open Online University (HOOU). The learning module will be published at https://www.hoou.de under the Creative Commons license CC-BY-SA. Free use is possible via the online learning environment https://www.viamint.de at the Hamburg University of Applied Sciences. A beta version is available for use at the moment.

Automatic multilingual subtitling and dubbing of video lectures

Jorge Civera
Universitat Politècnica de València
Spain
jorcisai@vrain.upv.es

Open educational resources (OER) are rapidly growing, but are not usually offered in multiple languages due to the lack of cost-effective solutions to translate the different objects comprising them and particularly videos. However, current state-of-the-art natural language processing (NLP) technologies including automatic speech recognition (ASR), machine translation (MT) and speech synthesis (SS) have reached a level of maturity which opens the possibility of producing automatic multilingual videos of publishable quality at low cost. More precisely, multilingual subtitles and speaker-adapted dubbed tracks can be automatically provided with minimum intervention from the lecturer. In this talk, we present our experience providing multilingual video lectures describing the systems, tools and integration components employed for such purpose. Finally, multilingual videos generated with our own NLP technology will be shown.

Smooth Animations in JSXGraph

Murray Bourne
Gold Coast
Australia
murraybourne@gmail.com

There are various methods for animating objects in JSXGraph. When it's a simple animation of only one or two items it's usually acceptable, but as the complexity of the animation grows, performance can really suffer especially when using low-end devices.

This talk will discuss various ways you can go about animations in JSXGraph, and how to monitor performance bottlenecks. We will also suggest a solution for super-smooth animations of a large number of objects.

To get the most out of this talk, it is assumed you are comfortable with javascript coding and have a rudimentary grasp of how SVGs are used to display images in HTML.

DIVE into Math with JSXGraph in JupyterLite

Chung Chan
Department of Computer Science
City University of Hong Kong
Hong Kong
chung.chan@cityu.edu.hk

We developed a new Python package, divewidgets, that integrates JSXGraph into Jupyter Notebook to achieve literate programming for math, where explanations can be mixed with modifiable and executable codes to illustrate abstract mathematical concepts. The work is part of an effort in Project DIVE at CityU during the pandemic that aims to provide a Diversified, Interactive, Versatile, and Engaging virtual learning environment. Using the tool, our students have created a series of interactive Jupyter notebooks to illustrate secondary mathematics in Hong Kong DSE curriculum. The notebooks were deployed on a highly-accessible JupyterLite website and presented in a CityU classroom for educators. Implemented as ipywidgets, JSXGraphs can be modified dynamically by Python code. In addition to having separate JavaScript/HTML coding panels for customizing and running JSXGraphs, one can also enter JSXGraph code directly using cell magics.

Refactor playgameoflife.live – Utilize JSXGraph the React Way

Leslie Wong
Guangdong
China
79917148leslie@gmail.com

Motivation

Spending some time refactoring the JSXGraph-based game of life implementation Playgameoflife.live several months ago, I deem it would be worth presenting a talk, and share some experience using JSXGraph within the react ecosystem. Upon reflection, there are three points worth mentioning.

1. The way to run JSXGraph, from the sequential way with a script tag to the class way with the useEffect hook in a react functional component.
2. Architecture migration, from Heroku to AWS SAM.
3. The encoder and decoder implementation of RLE file, the popular format to share game of life patterns.

Proposition Definition

The universe of the Game of Life is an infinite, two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive and dead, (or populated and unpopulated, respectively). Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

- For a box with a live cell:
  - There existing fewer than two live neighbors, the cell dies, due to loneliness.

  - There existing more than three live neighbors, the cell dies, due to crowding.
  - There existing two or three live neighbors, the cell lives on to the next generation.
- For an empty box or a box with a dead cell:
  - There existing three neighbors, the box generates a new live cell, as if by reproduction.

Further Details

For further refactor details, please refer to
https://github.com/Leslie-Wong-H/game_of_life

First attempt to add interactivity with JSX Graphs on HELM workbooks in STACK

Konstantina Zerva
School of Mathematics, the University of Edinburgh
United Kingdom
K.Zerva@ed.ac.uk

Since the onset of the COVID-19 pandemic, the University of Edinburgh and Loughborough University collaborated to develop interactive online workbooks for the mathematics education of STEM students. These materials have been in use and evolved over the last two years, in which we have seen a significant amount of online teaching and students have engaged significantly better with the interactive online workbooks than in the other elements of the courses over this time.

Our approach to the online workbooks was to use the STACK online assessment system and combine our experience gained with running the fully online course “Fundamentals of Algebra and Calculus” (FAC) with materials from the HELM (Helping Engineers Learn Mathematics) project workbooks. Each interactive workbook is a Moodle quiz with textbook-style written content, worked examples, simple tasks and practice questions. Almost all questions are randomised so each student will have a collection of questions which use different values.

When we started translating HELM into STACK we didn’t have any interactive element because it required extra effort. This talk is an update on the HELM into STACK project. It focuses on updates that happened over the last year and how we started adding JSX graphs in some of our questions.

EXPERT Project: Examples of using JSXGraph foreignobject elements

Bernat Martinez
BonNouEdu
La Vila Joiosa
Spain
cabernat@gmail.com

Erasmus+ EXPERT project coordinated by Maribor University is trying to investigate how the use of videos could improve the teaching of STEM subjects.
In this presentation we are introducing some examples of using JSXGraph foreignobject elements. In our examples these elements include videos describing a physical situation where some measures can be performed.Then the JSXGraph construction is included in a Formulas question type in order to evaluate the correctness of student performance.
Examples can be shown at https://youtu.be/bzm2JZhCbK0

Programming workshop

Alfred Wassermann
Universität Bayreuth

https://jsxgraph.org/conf2022/pdfs/workshop_I.pdf
https://jsxgraph.org/conf2022/pdfs/workshop_II.pdf

Examples of JSXGraph 3D for Engineering Math

Wigand Rathmann
Department of Data Science
Friedrich-Alexander-Universität Erlangen-Nürnberg
Germany
wigand.rathmann@fau.de

In Higher Engineering Mathematics (HEM) several concepts are hard to sketch at the blackboard. Starting from stationary points at graphs of function with two unknowns or points with certain properties of the Hessian up to vector field or how coordinate transformations work are difficult the sketch. JSXGraph now supports the visualization of 3D interactive diagrams as a brand new feature. The talk will show some applications demonstrating the world of 3D JSXGraph and its use within the STACK question type.

The focus will be set on

- coordinate transformation:
  - polar coordinates,
  - spherical coordinates,
  - general coordinate transformation,
- calculus of function with two variable,
- visualization of vector fields.

MecLib – STACK and JSXGraph for e-Learning in Engineering Mechanics

Martin Kraska
Technische Hochschule Brandenburg
Brandenburg an der Havel
Germany
kraska@th-brandenburg.de

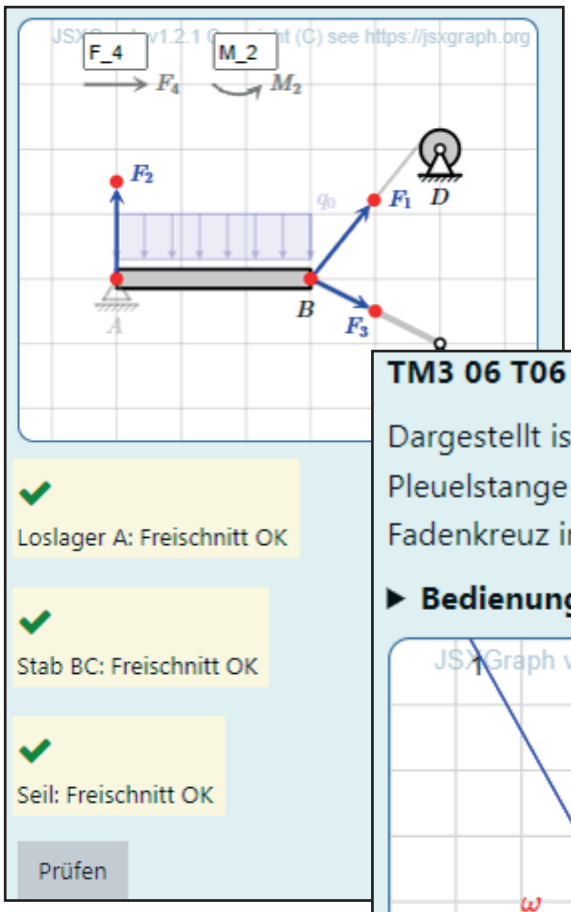
STACK is a Moodle question type which comes with built-in support of JSXGraph, a JavaScript library for interactive and dynamic graphics. It is possible to evaluate properties of interactively modified graphics from within STACK and to provide automatic feedback. Authors of STACK questions usually handle HTML, LaTeX and Maxima (the CAS engine used in STACK).
MecLib has been developed as a Maxima interface to a pre-defined set of high-level JSXGraph objects. This reduces the complexity of content development, because no knowledge of JSXGraph or Javascript is required.
The presentation gives an update on the current state of development and application.
On the development side, the code has been adapted to the recently released STACK version 4.4. Also, adjustments to newer versions of JSXGraph had to be made. The Meclib github wiki site has become a handy reference for content authoring.

Two applications will be presented:

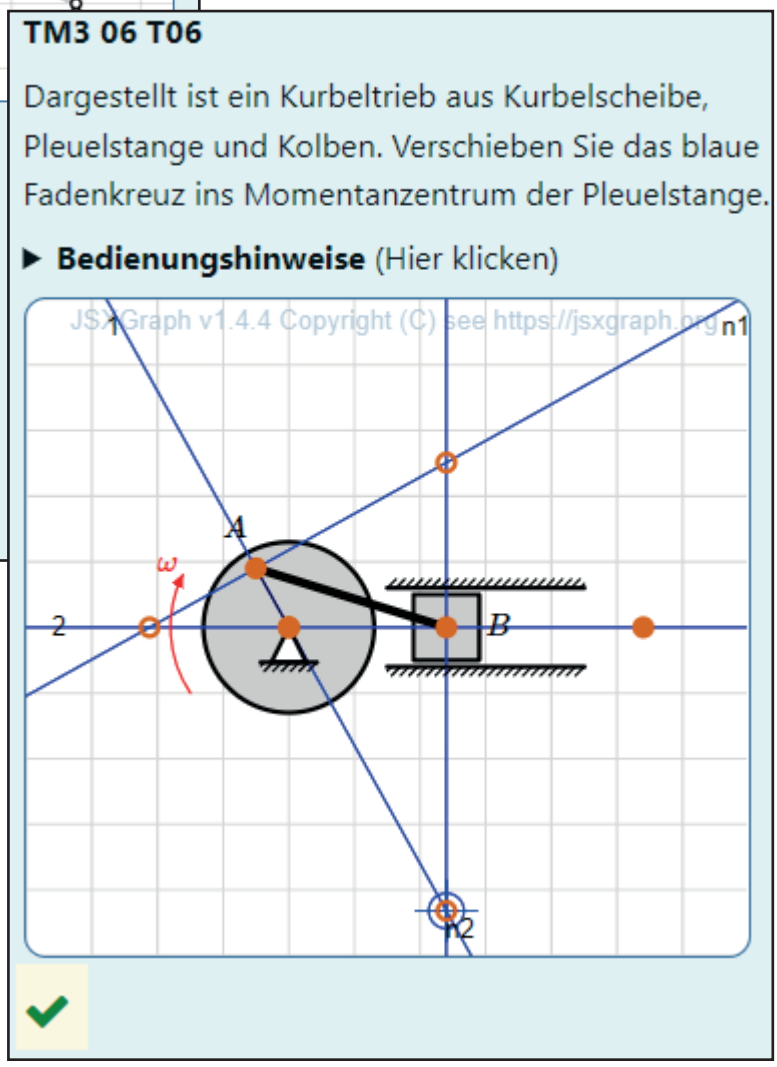
- an editor for free body diagrams with detailed formative feedback. This demonstrates the combined power of interactive meclib widgets and adaptive feedback functions.
- an interactive widget for finding the center of rotation in rigid body kinematics. This demonstrates the inclusion of multiple Meclib images in a single

question and the functional enhancement of Meclib images by local Javascript code.

Editor for free body diagrams:



Center of rotation problem:



Authoring the Next Generation of LMS Activities with LTI 1.3, STEMCstudio, JSXGraph, and more

David Holmes
CEO STEMCstudio
Raleigh NC, 27605
USA
david.geo.holmes@gmail.com

Tired of copying and pasting your plugin activity code into a tiny, lifeless, plugin text window in Moodle? Still hoping that it will run after you have made countless manual tweaks to integrate it with the plugin syntax-du-jour? Wishing you could write a real library to get code reuse and productivity? Forever jousting with your Moodle admin to install server-side utilities that don't scale to the number of users? Envious of those developers with smart IDEs, modern tools, and thinking that there must be a better way?
Well there is!

In this session we will explain:

- How LTI 1.3 creates a new paradigm for LMS activity development.
- How STEMCstudio provides a modern developer experience.
- How to author an activity application in STEMCstudio, and deploy it to a Moodle course.
- What is going on behind the scenes with LTI and STEMCstudio.
- How to use any standard JavaScript library in STEMCstudio, with a JSXGraph example.
- How to interact with the Moodle Gradebook.
- How to get symbolic mathematics in your activity without scaling problems.
- How to author a standard and modern JavaScript library, with a JSXGraph example.

EconPractice: JSX Graphs in a Learning Platform for Economics

Thomas Groll and Nikolas Nyby
Columbia University, New York, NY, USA
tg2451@columbia.edu

EconPractice is an open-access, instructor-driven learning platform developed at Columbia University with a current focus on applications in economics. The learning platform combines various open-source tools such as JSXGraph and Django to offer instructors and students the advantages of online illustrations, assessments, and practices produced by commercial products while supplying the flexibility of content creation, adoption, and sharing.
EconPractice offers a variety of features to instructors and students. The platform allows instructors to create economic illustrations and practice assignments that can be shared with students on EconPractice, embedded in a course's website, or incorporated in assignments on platforms such as Blackboard, Canvas, Doodle, and others. Instructors can provide interactive feedback within the illustrations for review or auto-graded assignments and share content with other instructors. Students can use EconPractice to review provided economic illustrations, create their own, or embed them in their own work.

The current development includes:

- Various diagram prototypes based on JSXGraph.
- A user-course management system.
- Verbal and numerical feedback for scoring.
- Integration into other platforms and online sites.

Current development work focuses on building assignments with adaptive pathways and analytics.

Video lectures taxonomy

Alenka Lipovec
University of Maribor, Faculty of Education
Faculty of Natural Sciences and Mathematics
Maribor
Slovenia
alenka.lipovec@um.si

This is joint work with Martin Putzlochner, Stiftland-Gymnasium Tirschenreuth, Germany, m.putzlocher@stiftland-gymnasium.de

Video has created a pedagogy for current learning and teaching practices – video pedagogy. Flipped learning with video learning materials was used extensively during the coronavirus pandemic. The highest positive correlation between flipped learning and student achievement was reported for STEM. The effectiveness of video lectures in education depends on many characteristics. The vast majority of empirical results relate to the university level. It appears that people learn better from an instructional video when: the lesson includes prompts to engage in summarising or explaining the material (generative activity), the instructor draws graphics on the board during the lecture (dynamic drawing), the instructor shifts eye gaze between the audience and the panel during the lecture (gaze guidance), and a demonstration is filmed from a first-person perspective (perspective principle). Standard formats for video lectures include lecture capture, picture-in-picture, and voiceover. Lecture capture involves videotaping a physical lecture. Picture-in-picture combines a full-screen presentation of the slide content with a small video recording of the lecturer (e.g., talking head in a lower corner). In contrast, voiceover combines a full-screen presentation with audio narration by the instructor. Another format combines images of the instructor with content that the instructor can monitor in real-time; this format is called a “live composite” and has a distinct

advantage over other video lecture formats. Further guidelines for video lecture design include multimedia (presentation of words and graphics), coherence (avoid redundant material in slides and script), signalling (highlight key material), redundancy (no subtitles that repeat the spoken word), spatial contiguity (place printed text next to the corresponding part of the graphic), temporal contiguity (present related visual and verbal material at the same time), segmentation (break a complex lecture into progressively presented parts), modality (present words as spoken text), personalisation (conversational language), voice (use appealing voice), and embodiment (display gesturing instructor).

The language of instruction plays a crucial role. It seems that subtitles cannot effectively overcome a language barrier. Instead of subtitles, we can use machine translation of audio into video. Additional features (e.g. no language-specific text) must be considered for multilingual videos created with automatic speech recognition and machine translation. The creation and use of educational video lectures for teaching purposes is not new. However, there is still no concrete guidance to help teachers choose the appropriate video for their students. We have therefore attempted to create a taxonomy of video lectures based on a hierarchical structure of levels. The goal is to use the characteristics/principles of video lectures to provide a foundation in the form of a taxonomy scheme that will help teachers determine the quality level of video resources they need for effective instruction. As an analogy, Bloom's hierarchical taxonomy was chosen, which is the most widely used in education. Reaching the lower levels is a prerequisite for achieving the higher taxonomic levels. The revised Bloom's taxonomy focuses on six levels: remember, understand, apply, analyse, evaluate and create.

We have summarised the various features of educational video lectures into five hierarchically organised levels: instruction (including generative activity and temporal contiguity); active learning (including generative activity and instructor visibility, e.g. recording format, gaze guidance, perspective, personalisation, voice and embodiment); interactivity; segmentation; dynamic visualisa-

tion (including dynamic drawing, multimedia, coherence, signalling and spatial contiguity) and multilingual principle (including redundancy and modality). For simplicity, the video lecture taxonomy is formatted as a checklist. he teacher scores the levels' realisation for the video lecture with points ranging from 0 to 4 (1 – minimal requirements, 2 – medium quality, 3 – high quality, 4 – excellent quality). For each level, descriptions and example videos help the teacher decide how many points could be awarded to the evaluated video lecture.

For instance, in the dynamic visualisation level, one point is awarded if the dynamic drawing principle is followed, two points are awarded if the video shows the lecturer using dynamic visualisation available on a third-party site, three points are awarded if the lecturer is using dynamic visualisation/geometry linked and four points are awarded if students themselves are using dynamic visualisation tool (e.g. as a part of JSXGraph interactive element in H5P). In our talk, we will further focus on the interactivity level and present some video lectures performed with different features and options of JSXGraph between H5P.

Tools and workflow for the development of interactive JSXGraph applets in a Moodle course

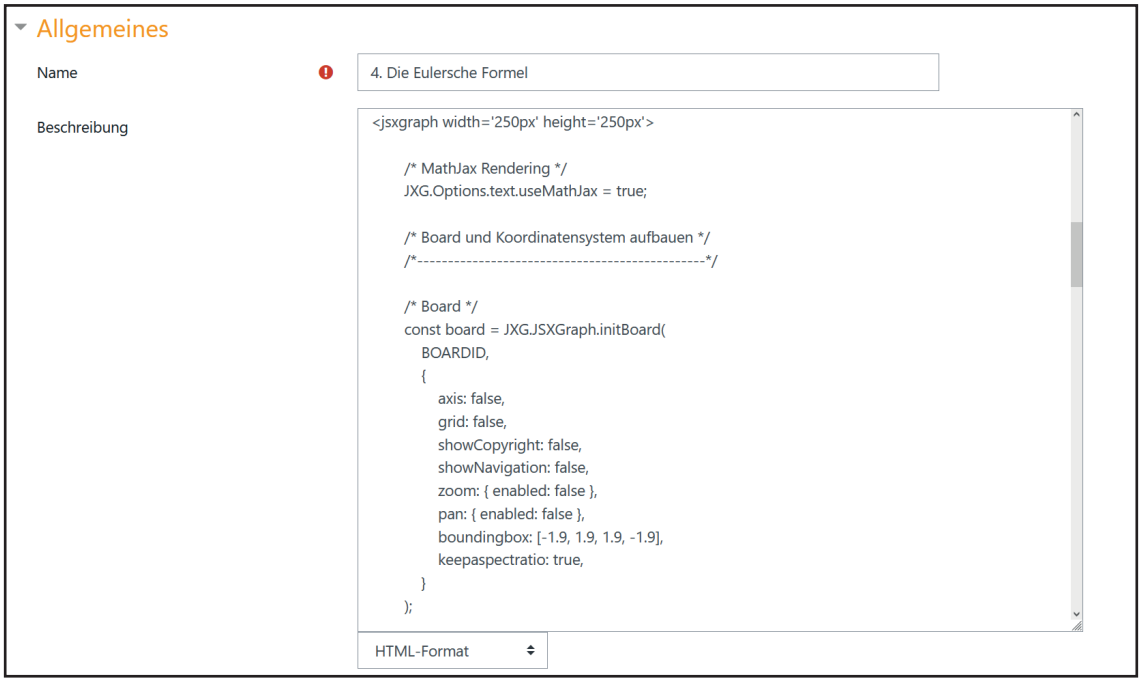
Bernhard Gailer
Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik
Amberg
Germany
b.gailer@oth-aw.de

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. Technically, to integrate JSXGraph into Moodle, the JSXGraph Moodle plugin is used on the one hand, and the JSXGraph version integrated into STACK on the other.

For programming the applets within Moodle, various text editors are available, such as the HTML editor Atto or a plain text editor. For STACK, the plain text editor is recommended to avoid problems with automatic formatting of WYSIWYG editors. These editors have limited functionality and are not fully designed for web development with JavaScript. For example, there is no syntax highlighting or linting in the plain text editor and only limited source code checking in the Atto editor. This makes the code difficult to debug and to read. Moreover, the ability to preview questions is only possible after saving the entire question. Development with direct output of the program code and automatic refresh, as is possible in web development with IDEs, cannot be achieved within Moodle editors. This makes applet development within Mood-

le less intuitive and a lot more time consuming compared to web development in an IDE.



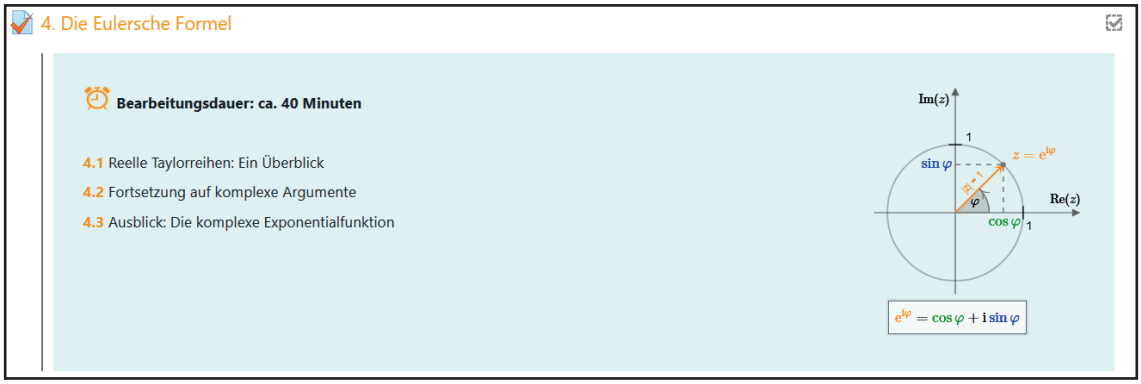
Another problematic aspect of question development in Moodle is tracking changes made to the developed questions, which is especially relevant in projects with multiple STACK developers. Adjustments made to a STACK question by other team members cannot be identified in the question. Version management can help here, but this is limited in Moodle.

From a developer's perspective, consistent programming based on style guides and layout templates is advantageous for better maintainability of the code. For instance, the consistent design and structuring of the questions content and the applets that appear in them can help developers to quickly understand the code and add new features. This is especially important in projects with multiple STACK developers and ensures an efficient programming workflow.

For the reasons mentioned above, the IdeaL project team tested various development tools outside of Moodle and developed a best-practice approach to the programming workflow of JSXGraph applets. Along with these tools, templates and style guides for programming are used to simplify the development process. In addition, a Gitlab project hosted on the university server was set up for version management and documentation of best-practices.

The workshop aims to give an overview over the tools used for JSXGraph programming within Moodle and to provide a hands-on experience in developing JSXGraph applets for a Moodle course. This workshop is intended for JSXGraph developers who construct JSXGraph applets in the Moodle editor (using the Moodle JSXGraph Plugin or JSXGraph in STACK).
In the workshop, the development tools used for programming JSXGraph applets in the project will be shown first. For each tool, the advantages for development will be explained. Based on this, the best practice approach for development in the project will be presented. In the second half of the workshop, a JSXGraph applet for visualizing complex numbers and Euler's formula will be developed in a live demo. In doing so, the previously presented workflow will be applied as part of the best-practice approach shown before. The coding basis for the live demo is a JSXGraph template, which will be made available to the participants. Finally, the developed code for the applet will be integrated into the learning module in the project's Moodle course.

The result will look like this:



The participants have thus gained an insight into the workflow of JSXGraph programming in the project IdeaL and have become familiar with tools for developing JSXGraph applets outside of Moodle.

JSXGraph integration within Eleda, a NoCode pedagogical authoring tool and platform

Christophe Bansart
Research and Development Director
KDetude EdTech company
Paris, France
christophe.bansart@kdetude.com

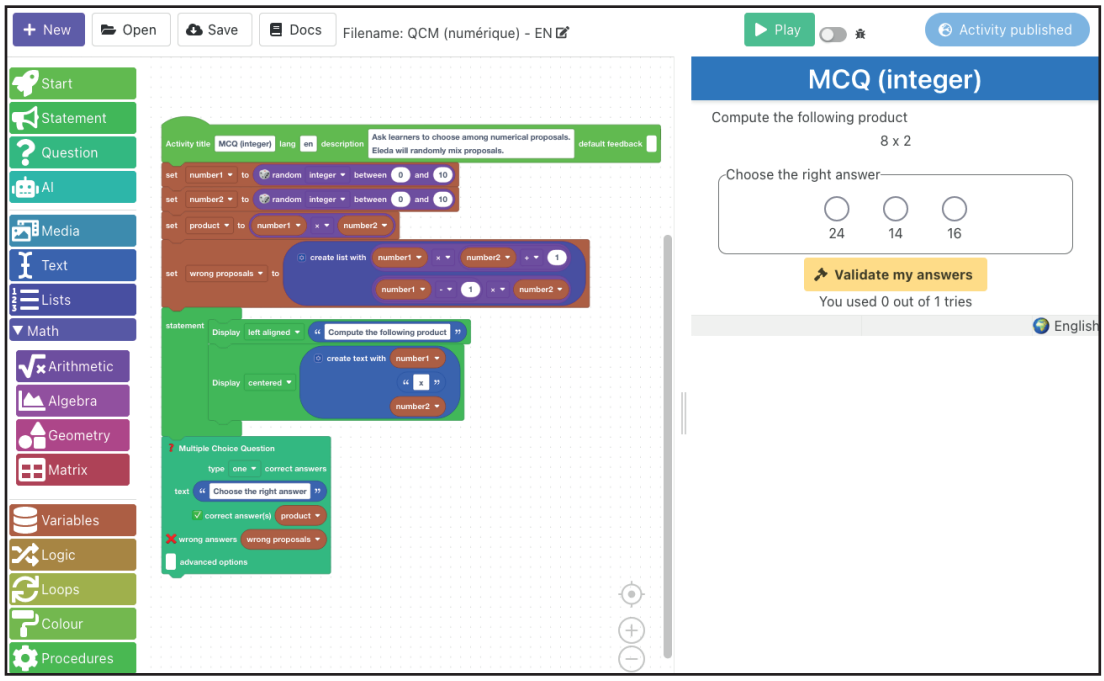
During this JSXGraph conference, we would like to talk about how we have implemented JSXGraph with our NoCode technology to facilitate mathematical activity design. Then we will explore on how teachers can reuse and adapt existing activities to their learners needs.
To illustrate that, we will demonstrate how to translate existing activity, how to add tailored feedback depending on learners regular mistakes.

1. Product context

Nowadays, creating and adapting pedagogical activities in STEM is a difficult task for teachers. The reason for this is that there are no easy to use authoring tools to create interactive and randomised activities. Therefore, teachers are coerced into reusing existing resources as they are, without being able to adapt them to their learners' needs.
Based on these observations, our company KDetude, strives to bring computational thinking to teachers' creativity to make learners succeed. To achieve this, KDetude is developing Eleda, a unique educative mathematical ecosystem. It offers schools' and teachers' communities the ability to make randomised mathematical activities with building blocks, as known as NoCode approach. As such, learners can train without limits, and develop their critical thinking and scientific mindset. Eleda wants to make teachers the "makers" of their online pedagogical activities in math.

Our solution Eleda offers :

- A new generation of authoring tool for teachers: just as the Scratch building blocks system does, we want to offer teachers the opportunity to easily design randomised activities through blocks. Computational thinking and Makers spirit fuel all domains and education is being following suit. In the end, one activity made by a teacher can generate an infinite number of problems to solve, allowing their learners to experiment endlessly with math.



The authoring User Interface (on the left side the building blocks, on the right side the result)

- A new generation of pedagogical activities: learners receive smart and tailor-made feedback to figure out why they have not succeeded. Thus, learners experiment until they understand underlying concepts and become comfortable with them.
- A “makers” multilingual community: an open community to share, translate, adapt, and reuse activities among “makers”.

2. Our JSXGraph implementation

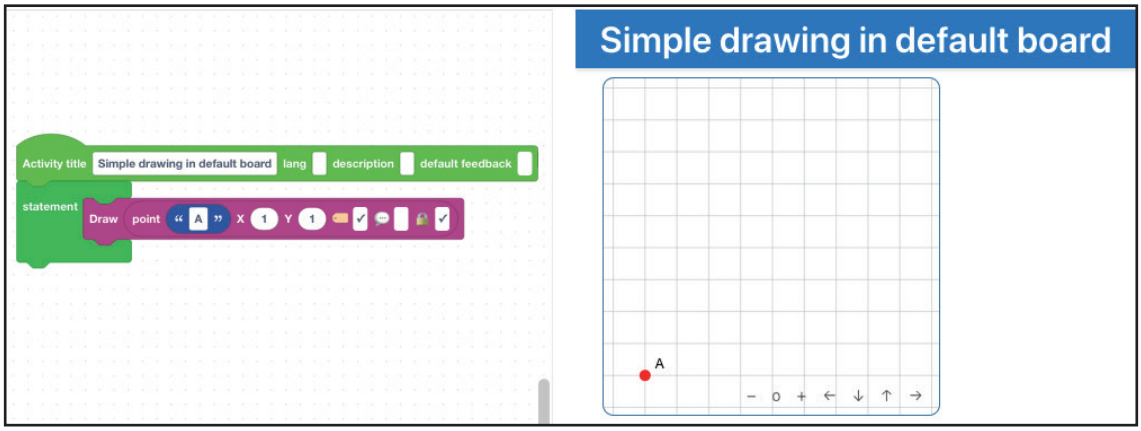
- We have implemented major geometric JSXGraph objects through blocks.

For instance a circle with the block circle:

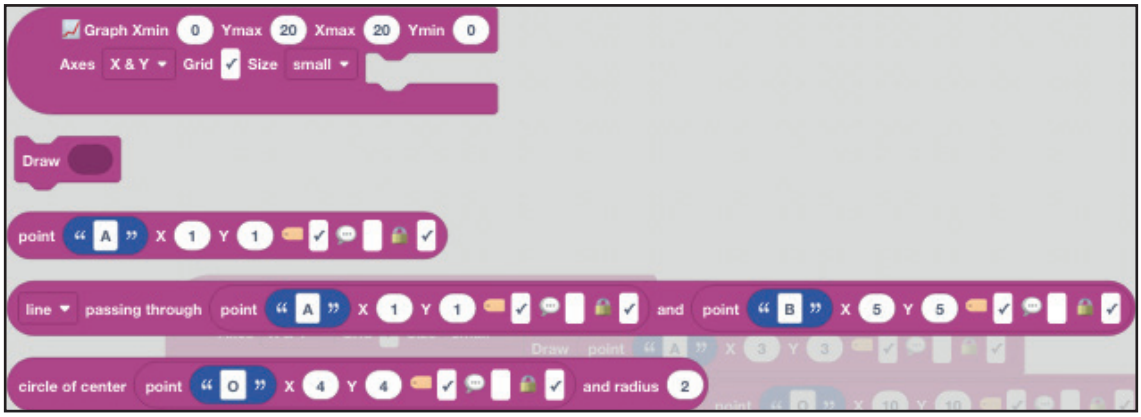
a line:

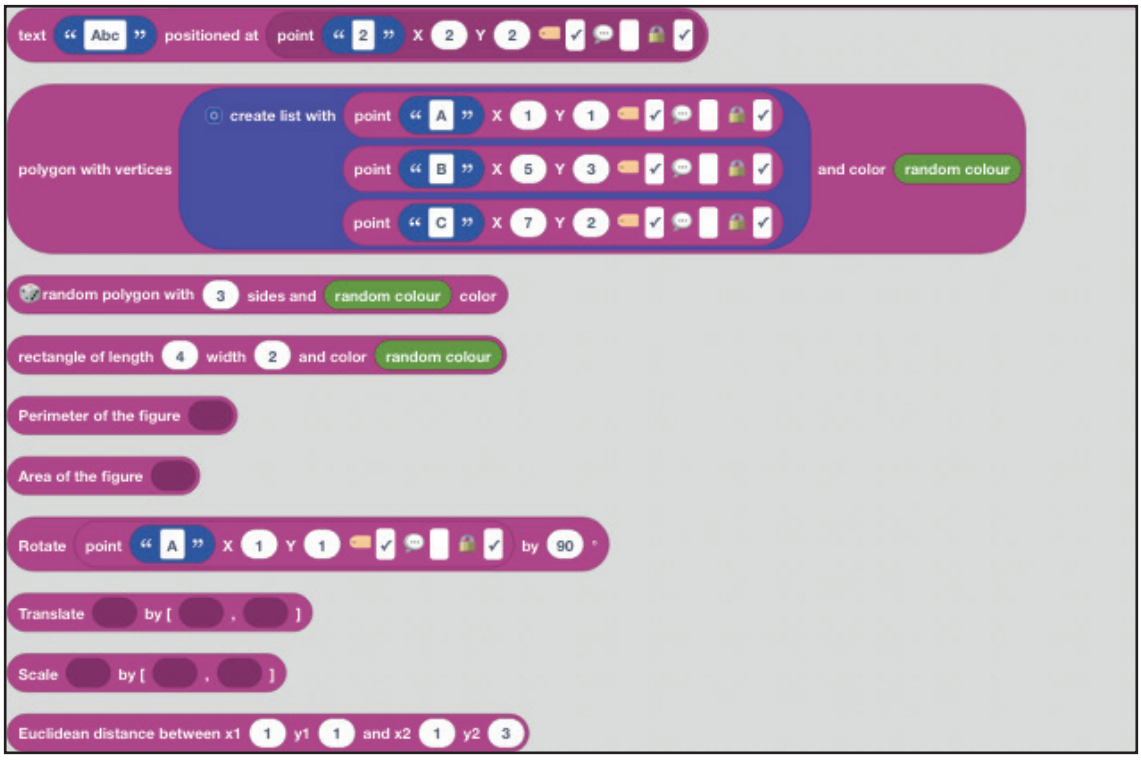
a point:

allows to produce a simple geometric figure as follows:

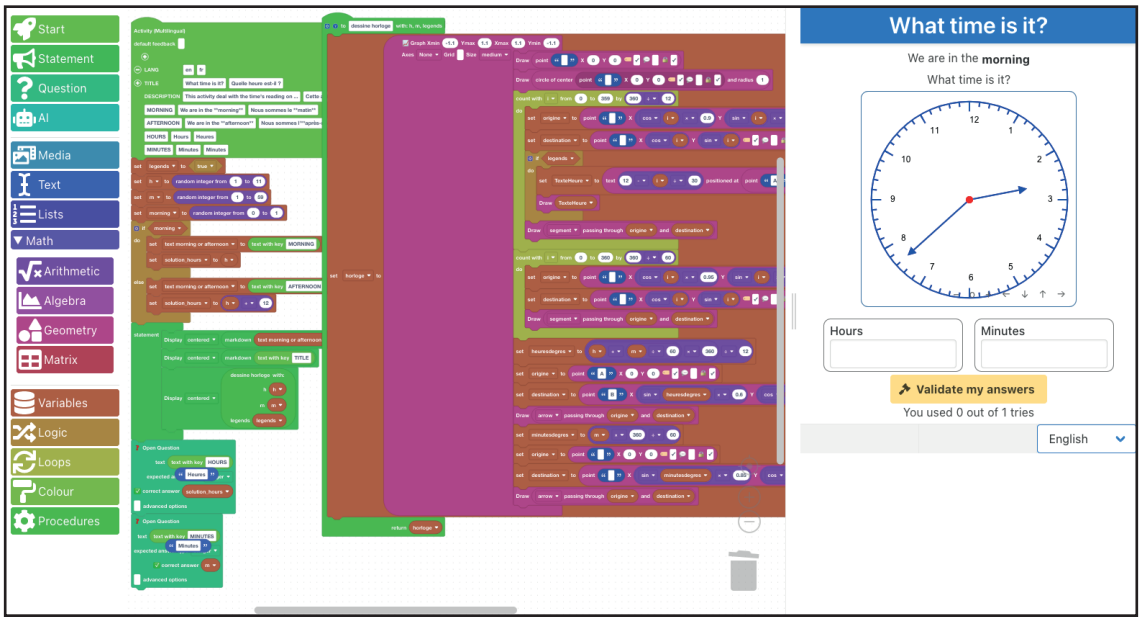


We have also added to the JSXGraph related blocks several general geometry blocks to compute for instance area and perimeter of a figure or compute Euclidean distance between two points.





Thanks to that, we could design more complex geometric objects.

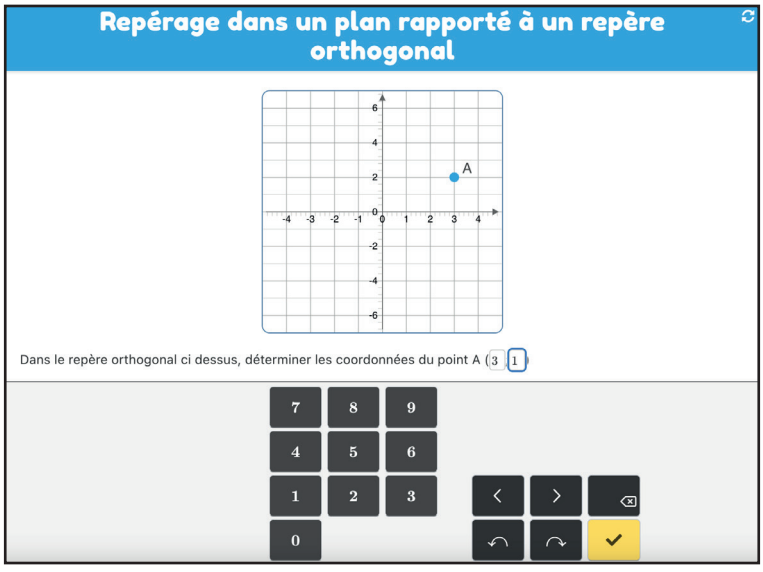


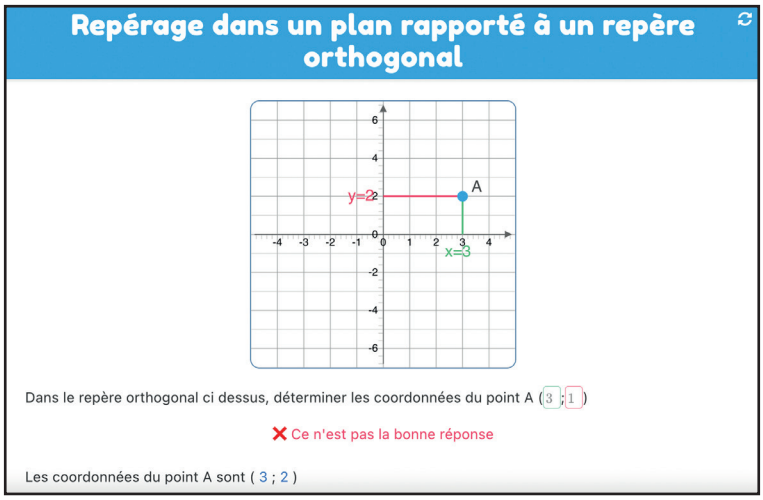
3. Bridging JSXGraph blocks with pedagogical blocks

We've implemented a wide math blocks library to allow teachers to design math activities. For instance

- Question blocks to design MCQ, Open Question to retrieve learners answers in different forms: fraction, numeric, algebric expressions
- Numeric blocks to manipulate and compute with numbers, fractions
- Computer Algebra systems blocks to manipulate algebraic items
- Feedback blocks to design activities with step-by-step questions and feedback that take into account learners' answers. The objective is to guide learners and explain why his answer his wrong or incomplete in the activity's given context.

Our NoCode implementation provides ability to mix all these latter blocks with jsxgraph blocks. Therefore, teachers can design complex pedagogical activities involving geometrical drawings. Geometrical drawings can evolve during the activity session. They can be updated depending on the learner's answers to show or hide geometrical elements during an activity. It could be used to provide specific feedback to explain more or less in detail the reasoning, or explain by showing why an answer doesn't work in a specific case.





How to get an Eleda account

We have opened a early adopters program to allow teachers to build their own or reuse math activities freely. To get a Eleda free access contact us.

Using JSXGraph in an industrial environment

Brais Oubiña Vila
Focke Meler Gluing Solutions
boubina@meler.eu

My presentation will be a talk targeting a general audience: no previous knowledge will be required to understand it and its content, very different to what has been presented in the previous editions of the Conference, showcases how JSXGraph can be a very valuable tool even in contexts where it would not be expected to be found.

One way of applying hot-melt adhesive, commonly called *swirling*, consists of blowing air around an adhesive jet to make it spin, forming a prolate cycloid on the substrate:

Figure 1: High-speed photograph of a swirl application.

The adhesive density and number of self-intersections are key features for a good bonding, so I used JSXGraph (for the first time) to visualize, on a simple plot with a few sliders, how the parameters of application change the curve:

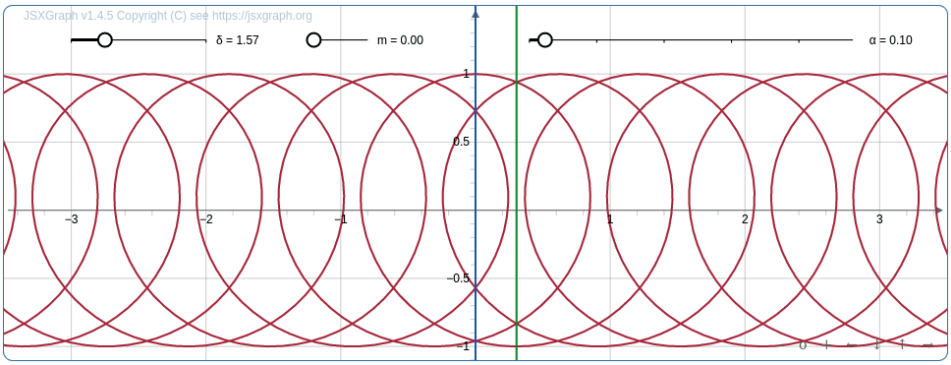


Figure 2: Mathematical curve representing a swirl application.

Surprised by how easy it was to do the programming, I started using it for more complex tasks, particularly to create two applications that have been distributed among our salespeople and technicians to help them understand a new product that we have launched:

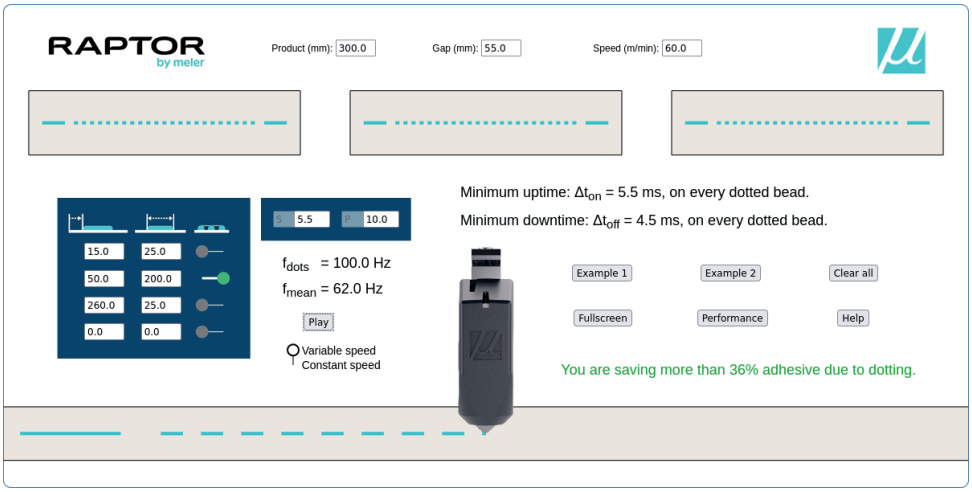


Figure 3: The Raptor Infinity Application Planner.

The Raptor Infinity Application Planner lets the user (usually salespeople) simulate an application given a set of parameters from the customer and evaluate whether it might be feasible with the specific applicator for which this tool has been designed.

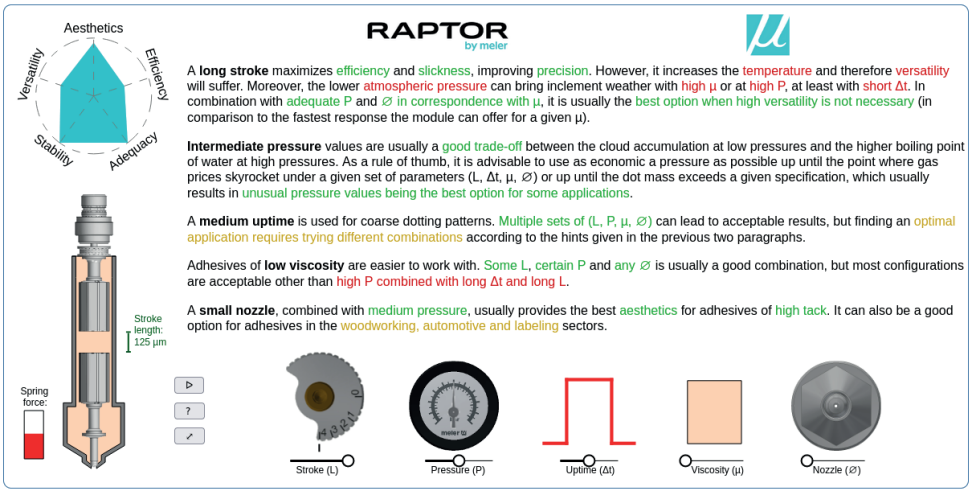


Figure 4: The Raptor Infinity Parameter Explorer.

The Raptor Infinity Parameter Explorer helps the user (commonly service technicians) understand how the applicator will respond to changes in the input parameters and why, which is necessary to be able to optimize the results obtained.

In this talk, I will explain what my previous background was, why I started using JSXGraph, which results I was able to obtain and the amount of effort I needed to make. My expectation is that the majority of the audience will find it easy to follow and particularly interesting as something that one would not necessarily expect to see, but is simple and useful, being practical to the point that *it just works*.

Sequences with JSXGraph, Visualization and Algebraic Description synchronized

Andreas Maurischat
IntegralLearning
FH Aachen
Germany
andreas.maurischat@integral-learning.de

The concept of sequences and their limit and accumulation points poses a main difficulty for university students in the first semester. Having a visual idea of these concepts and their relation to the algebraic expressions, therefore, is of great value. In this talk, we show how one can depict sequences and their limit and accumulation points with JSXGraph. We will also show how to easily create such visualizations and also graphical questions in the LMS Mumie.