ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36607

Bringing differential equations to life by two- and three-dimensional visualizations of numerically simulated dynamic systems

Guenter Bischof (Associate Professor)

Günter Bischof holds a doctorate in physics and is currently an Associate Professor of Applied Mathematics at the University of Applied Sciences FH Joanneum in Graz.

Thomas Kainz

Thomas Kainz is an undergraduate automotive engineering student at the University of Applied Sciences FH Joanneum in Graz, Austria. Previously he finished the International Baccalaureate Diploma Programme at the Graz-International-Bilingual-School (GIBS) in Graz, Austria.

Eric Christian Menard

Eric Menard is an undergraduate automotive engineering student at the University of Applied Sciences Joanneum Graz. He is partaking in the formula student program to develop electric racing vehicles. Previously he was a student at the higher technical institute Lastenstraße in Carinthia (Austria) where he graduated in the field of electrical engineering.

Robert Poetsch

Robert Poetsch is currently studying Automotive Engineering at the University of Applied Sciences Joanneum in Graz. Before this he graduated in electronics and IT from the Higher Technical College in Mödling.

Christian J. Steinmann

Christian Steinmann is manager of HM&S IT-Consulting and provides services for Automotive SPiCE. Currently, his main occupation is process improvement for embedded software development for an automobile manufacturer. On Fridays, he is teaching computer science and programming courses at Joanneum University of Applied Sciences in Graz, Austria.

Maximilian Sterkl

Maximilian Sterkl graduated from a Higher Institute of Technical Education and worked as a internal quality engineer at a company producing elevator safety parts, before starting to study at the University of Applied Sciences FH Joanneum.

Christoph Tröster

Christoph Tröster is currently studying Automotive Engineering at the University of Applied Sciences Joanneum in Graz. After completion of his studies, he would like to participate in the development of electric cars.

Bringing differential equations to life by two- and three-dimensional visualizations of numerically simulated dynamic systems

Günter Bischof [1], Christian J. Steinmann [1, 2], Thomas Kainz [1], Eric C. Menard [1], Robert Poetsch [1], Maximilian Sterkl [1], and Christoph Tröster [1]

[1] Joanneum University of Applied Sciences, Institute of Automotive Engineering, Graz, Austria
[2] HM&S IT Consulting, Graz, Austria

Abstract – The use of technology in the teaching and learning of mathematics is becoming increasingly more prevalent in mathematics education. It affects not only how to teach mathematics, but also what mathematics becomes possible to be taught. Especially commercially available computer algebra systems have become ubiquitous tools, although there is some concern that they may detract students from understanding core mathematical concepts. Such an undesirable effect can be avoided by making use of high-level programming languages in mathematics education. Computer programming does not only strengthen problem solving skills and logical and sequential reasoning, it also provides a high degree of flexibility and an ample scope for applications in teaching and learning.

Here we present an example of the synergistic effect of the interaction of computer programming and mathematics in undergraduate engineering education within the framework of team-oriented project-based learning. The students of an Engineering Mathematics course were organized into teams of three and given the task to develop computer programs that make use of mathematical algorithms taught in this course. Two teams chose the simulation of the behavior of a non-linear dynamic system, namely a mathematical pendulum with an elastic suspension. The teams were required to solve the equations of motion numerically and to visualize the motion of the system by making use of a high-level programming language. The students were given free rein on creativity, which lead to equivalent computational results but significantly different visualization designs. One group used C# as programming language and chose a rather scientific approach. Their program offers several options like phase space visualizations and the comparison of different numerical methods with varying orders of the truncation error. The other group used the Unity game engine for the visualization of an elastic pendulum with three degrees of freedom. The motion of the system can be observed on the computer monitor or through a virtual reality viewer as a three-dimensional object in an immersive, virtual environment. By using a VR headset, the users have the possibility to move around the virtual environment – a huge pendulum hall – and immerse themselves into the spatial perception of the moving spring pendulum.

In this paper we present our teaching approach, the theoretical background and the outcome of the student projects. The different solution and visualization strategies of the student teams are contrasted with each other and discussed. The programs can be obtained from the authors free of charge.

Introduction

Computers and software have not only shaped the research landscape in more than a half-century; they have also increasingly found their way into mathematics education. Their emergence and application have changed what mathematics is, what mathematical methods are used, and what is important to teach [1]. In a 2007 study conducted on 47 AMS Group I university mathematics departments, professional mathematicians were asked to describe and explain the nature and extent of their mathematical software usage, and their opinion about the use of software in mathematics education [2]. In this work, it was shown that the computer algebra systems (CAS) MATLAB, Mathematica and Maple represent the primary tools of mathematicians, while LaTeX plays a central role in mathematical communication. And above all, computer programming is considered an essential skill that is important for the solution of both basic and advanced problems and should therefore be incorporated in the students' curriculum. In addition, the study showed that mathematicians are critical of the use of CAS in mathematics education, fearing that these systems might distract students' attention from in-depth thinking. However, in [3] it was shown that CAS can contributed to a significant increase in students' performance.

An unreflective use of ready-made algorithms in CAS can be avoided by having the mathematical algorithms programmed by the students themselves. For this purpose, high-level programming languages are particularly suitable because they make it easy to operationalize complex tasks with a computer program. In addition, programming skills are indispensable to advanced applications of commercial CAS software.

Since computer programming and mathematics are usually taught in separate courses, it is of particular importance to bring these two disciplines together in such a way that students can benefit from their synergies. At the Institute of Automotive Engineering at Joanneum University of Applied Sciences we have established a coherent procedure to familiarize our undergraduate students with the use of high-level programming languages for the solution of mathematical or engineering problems. The students complete software projects within a team-oriented project-based learning environment [4]. These projects are part of the requirements of both the Computer Programming course and at least one additional course within the curriculum in their second semester of study. Frequently, the Engineering Mathematics course in the second semester is that accompanying subject since it focuses on ordinary differential equations and on numerical methods for solving them. For both subjects, this is a fruitful collaboration, as their mutual usefulness becomes clearly visible.

It seems that a critical issue in teaching mathematics to engineering students is to find the right balance between the practical application of mathematical methods and in-depth understanding [5]. It is of great help when the teaching content can be linked to objects, materials or ideas that can be made visible. The computer programming projects provide our students with the opportunity to give shape to abstract mathematical concepts, and at the same time it allows them to improve their performance in the classical mathematics courses. In a way, mathematics gets solid ground under its feet and is linked to something real.

The process of our team-oriented project-based learning procedure is as follows. The students are first introduced to all the projects in a kickoff meeting with the supervisors early in the semester. The supervisors present the project tasks they have created and deliver information on the scope of the projects, the timetable and deadlines, the deliverables, and the evaluation criteria. A one-page task description of each project is displayed and time allotted for the project groups to form and make their selections. In general, two or more groups of three or four members work simultaneously and competitively on the same task. By having the option to select their own project, the students have the chance to delve into subjects of particular interest to them. It is actually intended that the degree of difficulty of the tasks is slightly higher than the solution competence of the students at the beginning of the project. The necessary knowledge and skills for a successful project completion are taught during the project work in the accompanying lectures, which causes increased attention and interest among the students in these courses.

The numerical simulation of a spring pendulum, which is presented in this paper, represents a typical project assignment for our student teams. It is just one of a varied list of project proposals offered to our students to cover a wide range of interests. In this academic year that list comprised the solution of the traveling salesmen problem with self-organizing Kohonen maps, the driving cycle optimization of battery electric vehicles, an optimized distribution of electric vehicle charging stations, a route-finding optimization, and the calculation and visualization of Mandelbrot sets. In this way, it became possible to offer a class of almost 70 students enough topics that a maximum of three teams of three could work on each task.

A pendulum with an elastic instead of an inextensible suspension is the simplest realization of an autonomous, conservative, oscillatory system of several degrees of freedom with nonlinear coupling. The task description for this project was

In the idealized mathematical pendulum, a mass conceived as point-shaped, which is suspended at a point by means of a massless pendulum rod, can swing back and forth in a vertical plane, neglecting friction effects, in particular air resistance. The pendulum performs an almost harmonic oscillation, the oscillation duration of which in the case of small deflections is determined exclusively by the length of the pendulum and the prevailing gravitational acceleration. In the case of small deflections, there exists a closed solution of the equation of motion.

The situation changes drastically as soon as these simplistic assumptions are discarded, with a variety of possibilities to approximate the mathematical model to a real pendulum. In this project, the restrictions to a small vibration amplitude and to an infinitely stiff pendulum rod are to be abandoned, which in both cases means that no closed solutions exist and therefore numerical solution methods have to be applied.

The finite stiffness of the rod leads to oscillations along both the r and θ degrees of freedom, and thus to the occurrence of Coriolis accelerations. The resulting coupled differential equations are nonlinear and lead to chaotic behavior for certain parameter combinations.

Write a computer program with which the equations of motion of the elastic pendulum are solved numerically with at least two integration methods of varying orders of the truncation error. Allow an animated representation of the movement of the pendulum, as well as separately the representation of the r and θ deflections over time and a representation of the respective phase spaces. An interactive variation of the simulation parameters would be desirable, as well as the output of the simulation data to external files.

After an initial consultation with the supervisors and the assignment of duties within the team, the students start researching their topics, acquire relevant background information and skills, find the mathematical solutions, and finally design and program the software. The role of the supervisors is to guide the students through these stages and to give them advice and support for a successful completion of the projects [6].

The final results of all projects are stand-alone computer programs that can be installed and run on Windows operating systems, and a technical report, consisting of the task description, the functional specification, the architecture of the software and the solution approach and a summary of the project correspondence and meeting minutes. At the end of the semester, the teams present their programs to the other students of the class and to their supervisors.

The remainder of this paper is structured by first providing a brief description of the equations of motion of a planar and a three-dimensional spring pendulum. Then the implementation into student-written computer programs is described and discussed.

Equations of motion of the moving spring pendulum

The physical system to be investigated consists of a pendulum bob that it is attached at the top to a universal joint by means of an idealized massless spring, which can be stretched but not bent. In addition, the pendulum suspension can be moved up and down according to the function $\tilde{z}(t)$. In the case of zero angular momentum about the vertical line through the point of suspension the motion is limited to a plane, i.e., the *x*-*z* plane. The system has thus two degrees of freedom, the deflection angle $\vartheta(t)$ and the spring elongation $r(t)$ (see Figure 1).

Let l_0 be the unstretched length of the spring, k its elasticity or stiffness and m the mass of the bob. At equilibrium the weight is balanced by the elastic restoring force

$$k(l - l_0) = mg \qquad (1)$$

so that the equilibrium length of the pendulum at $\vartheta = 0$ is $l = l_0 + mg/k$. The spring will thus have the elongation $l + r(t)$ at the time t.

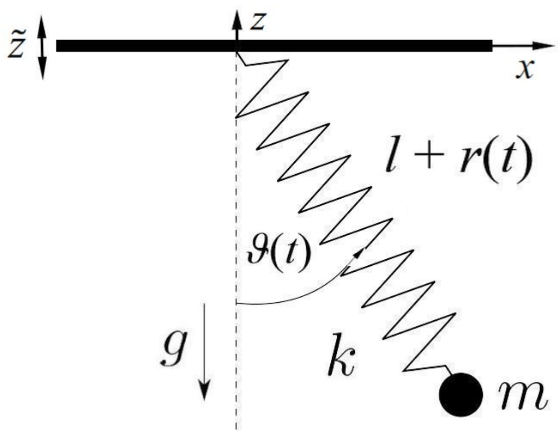

Figure 1: Two-dimensional elastic pendulum with external excitation $\tilde{z}$

In cartesian coordinates, centered at the point of suspension of the pendulum (see Figure 1), the Lagrangian is given by

$$L = T - U = \tfrac{1}{2} m\left(\dot{x}^2 + \dot{z}^2\right) - \tfrac{1}{2} k(r + mg/k)^2 - mgz\,, \tag{2}$$

with x and z being

$$x = (l+r)\sin(\vartheta) \;\text{ and }\; z = \tilde{z} - (l+r)\cos(\vartheta)\,, \tag{3}$$

expressed in cylindrical coordinates (see e.g. [7]).

The Lagrangian together with a Rayleigh dissipation function $R(\dot{q}_k) = (1/2)\beta\dot{q}_k^2$, which accounts for non-conservative damping forces is inserted in the Euler-Lagrange differential equations in order to derive the equations of motion of the system:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = -\frac{\partial R}{\partial \dot{q}_k}. \tag{4}$$

Equations (4), resolved after $q_1 = r$ and $q_2 = \vartheta$, result in the following equations of motion

$$\ddot{r} = \ddot{\tilde{z}}\cos(\vartheta) + (l+r)\dot{\vartheta}^2 - \frac{k}{m}r + g(\cos(\vartheta) - 1) - \frac{\beta}{m}\dot{r} + \frac{\beta}{m}\dot{\tilde{z}}\cos(\vartheta) \tag{5}$$

and

$$\ddot{\vartheta} = -\frac{g + \ddot{\tilde{z}}}{l+r} - \frac{2}{l+r}\dot{r}\dot{\vartheta} - \frac{\beta}{m}\dot{\vartheta} - \frac{\beta}{m(l+r)}\dot{\tilde{z}}\sin(\vartheta)\,. \tag{6}$$

In the case of a non-vanishing angular momentum about the z-axis, another degree of freedom perpendicular to the oscillation plane, i.e., the y-axis, comes into play (see e.g. [8]). Then the Lagrangian is generalized to

$$L = T - U = \tfrac{1}{2} m\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) - \tfrac{1}{2} k(r + mg/k)^2 - mgz \tag{7}$$

in cartesian coordinates.

A transformation into spherical coordinates introduces $\varphi(t)$ for the third degree of freedom, i.e.,

$$x = (l+r)\sin(\vartheta)\cos(\varphi)\,,\; y = (l+r)\sin(\vartheta)\sin(\varphi)\,, \text{ and }\; z = \tilde{z} - (l+r)\cos(\vartheta)\,. \tag{8}$$

When, for the sake of simplicity and clarity, the external excitation $\tilde{z}$ and the damping β are temporarily neglected, the Lagrangian L is given by

$$L = \tfrac{1}{2} m\left(\dot{r}^2 + ((l+r)\dot{\vartheta})^2 + ((l+r)\sin(\vartheta)\dot{\varphi})^2\right) - \tfrac{1}{2} k(r + gm/k)^2 - mg(l+r)\cos(\vartheta) \tag{9}$$

and the equations of motion are

$$\ddot{r} = (l+r)(\dot{\vartheta}^2 + \sin^2(\vartheta)\dot{\varphi}^2) - (k/m)r - g(1 + \cos(\vartheta))\,, \tag{10}$$

$$\ddot{\vartheta} = \sin(\vartheta)\cos(\vartheta)\dot{\varphi}^2 - \frac{2}{l+r}\dot{r}\dot{\vartheta} + \frac{g}{l+r}\sin(\vartheta)\,, \text{ and} \tag{11}$$

$$\ddot{\varphi} = -\frac{2}{l+r}\dot{r}\dot{\varphi} - 2\cot(\vartheta)\dot{\vartheta}\dot{\varphi}\,. \tag{12}$$

Both the two-dimensional (Equations (5) and (6)) and three-dimensional (Equations (10), (11), and (12)) equations of motion represent nonlinear coupled systems of second order differential equations, which cannot be solved analytically and numerical techniques must be employed.

Implementation into computer programs

In order to find numerical solutions to the equations of motion of the elastic pendulum, the above derived coupled systems of second-order differential equations are transformed into coupled systems of first-order differential equations of twice the original size. In the case of the planar pendulum, the substitution $z_1 = r,\ z_2 = \dot{r},\ z_3 = \vartheta, z_4 = \dot{\vartheta}, z_5 = t$ is carried out for Equations (5) and (6). In this way the non-autonomous differential equations with the second time derivatives of r and ϑ and explicitly time-dependent right-hand sides can be rewritten as a coupled system of nonlinear, autonomous first-order differential equations

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= -a\,\omega^2 \sin(\omega\, z_5) + (l + z_1) z_4^2 - \frac{k}{m} z_1 + g(\cos(z_3) - 1) - \frac{\beta}{m}(z_2 + a\,\omega \cos(\omega\, z_5)\cos(z_3)) \\
\dot{z}_3 &= z_4 \\
\dot{z}_4 &= \frac{a\,\omega^2 \sin(\omega\, z_5) - g}{l + z_1} - \frac{2}{l + z_1} z_2 z_4 - \frac{\beta}{m}\left(z_4 - \frac{a\,\omega}{l + z_1} \cos(\omega\, z_5)\sin(z_3) \right) \\
\dot{z}_5 &= 1
\end{aligned} \quad . \quad (13)$$

In Equations (13), the external excitation $\tilde{z}$ has been assumed to be sinusoidal and of the form $\tilde{z} = a \sin(\omega\, t)$.
Such a set of equations $\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z})$ can be easily solved numerically by means of algorithms taught in undergraduate mathematics courses (details of the mathematics courses that accompany the projects can be found in reference [9]). Both project groups have implemented the first-order explicit Euler method and the fourth-order Runge-Kutta (RK4) method for the numerical integration of the equations of motion. Thus, different orders of accuracy can be tested and compared for the same time step size, an additional benefit when the programs are used for instructional purposes.

Planar spring pendulum

A computer program, which focuses on the planar spring pendulum, was written by one of the project teams in C# within the software development environment Visual Studio 2017 [10]. C# is a type-safe object-oriented, general-purpose programming language. It takes up concepts of the programming languages Java, C, C++ and Delphi and supports the development of language-independent .NET components as well as COM components for use with Win32 application programs. Programming graphical user interfaces is particularly easy in C#, which facilitates the writing of user-friendly programs with little effort. Learning this programming language is mandatory for our students in the first year of study, as it is a valuable tool for the further course of their studies and also for their subsequent professional life. And, in addition, knowing a higher-level programming language is also extremely helpful when certain tasks or applications require the use of another programming language. Learning programming languages succeeds faster and easier in the case of a good command of any other language.

The planar spring pendulum program is organized in several subroutines (actually, due to object-oriented programming, *methods*), which comprise the numerical solution of the equations of motion (13), the readout and illustration of the results and the visualization of the motion of the pendulum, as depicted in Figures 2 to 4.

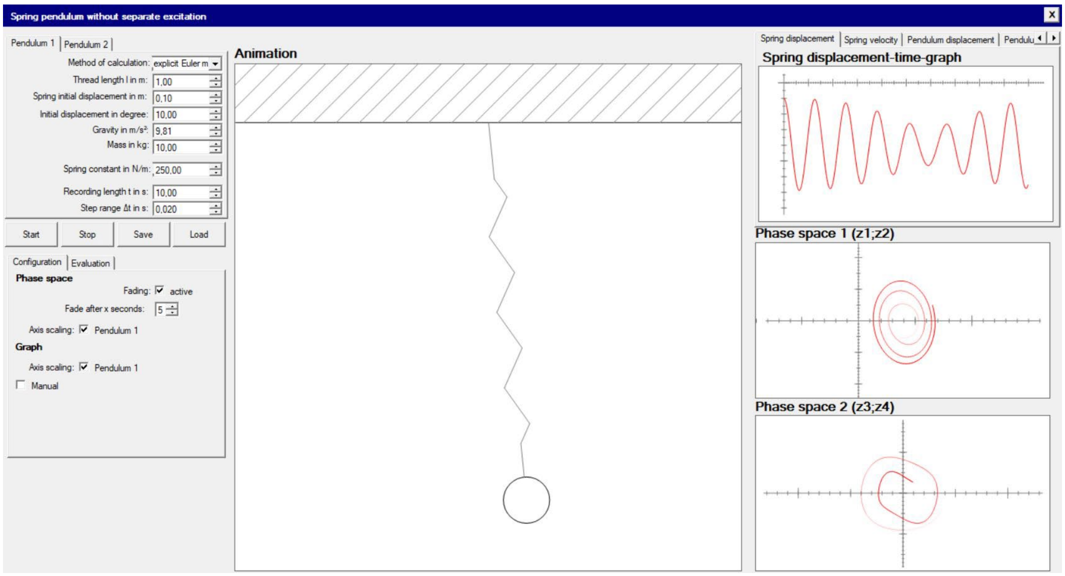

Figure 2: Graphical user interface of the planar spring pendulum program

The numerical integration of Equation (13) provides $z_1 = r$, $z_2 = \dot{r}$, $z_3 = \vartheta$, and $z_4 = \dot{\vartheta}$ instantaneously for every time step, so that both the spring elongation r and angle of deflection ϑ over time as well as the phase portraits $\dot{r}$ over r and $\dot{\vartheta}$ over ϑ can be plotted simultaneously.

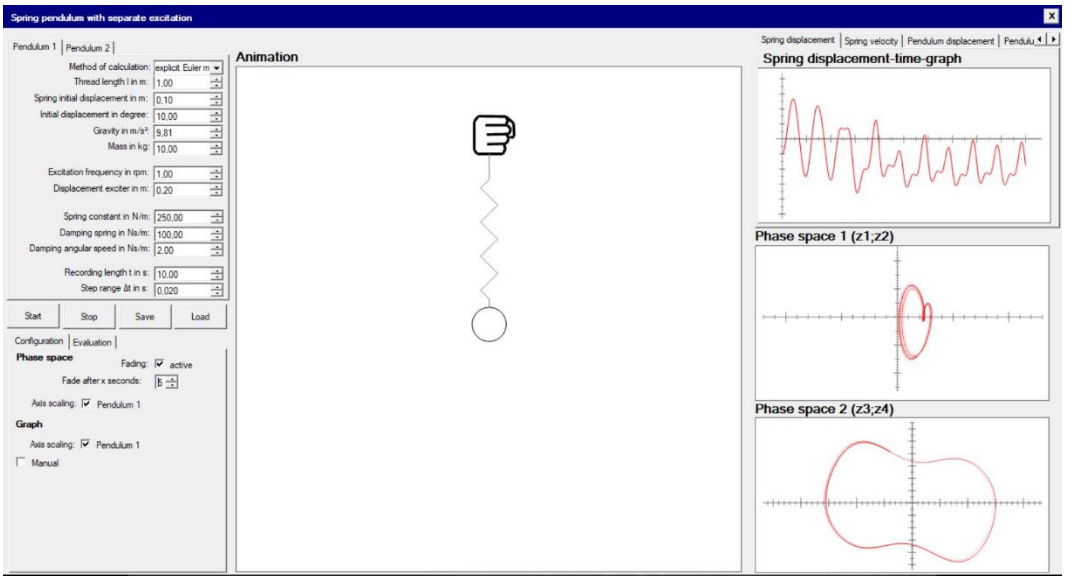

Figure 3: Graphical user interface of the planar spring pendulum program with external excitation

The graphical user interface shows an animation of the pendulum movement in the central window, while on the right-hand side optionally the pendulum or spring deflection or their time derivatives as well as the phase space portraits are displayed. On the left-hand side, the parameters of the pendulum can be set, and, in addition, a second pendulum can be added. The movement of that second pendulum can be displayed together with the first pendulum for comparison purposes (see Figures 2 to 4).

In this way, two different parameter sets can be examined with regard to their oscillation behavior, or the same pendulum can be computed with different integration methods and visualized in different colors.

The external excitation $\tilde{z}(t)$ was modelled and illustrated on the presentation of the lecturer in the mathematics course, in which he stimulated a spring-mass system to forced oscillations as shown in Figure 3. In Figure 4, the same externally excited pendulum has been computed with integrators of different order of accuracy, and the deviating state of motion is shown after a certain simulation time.

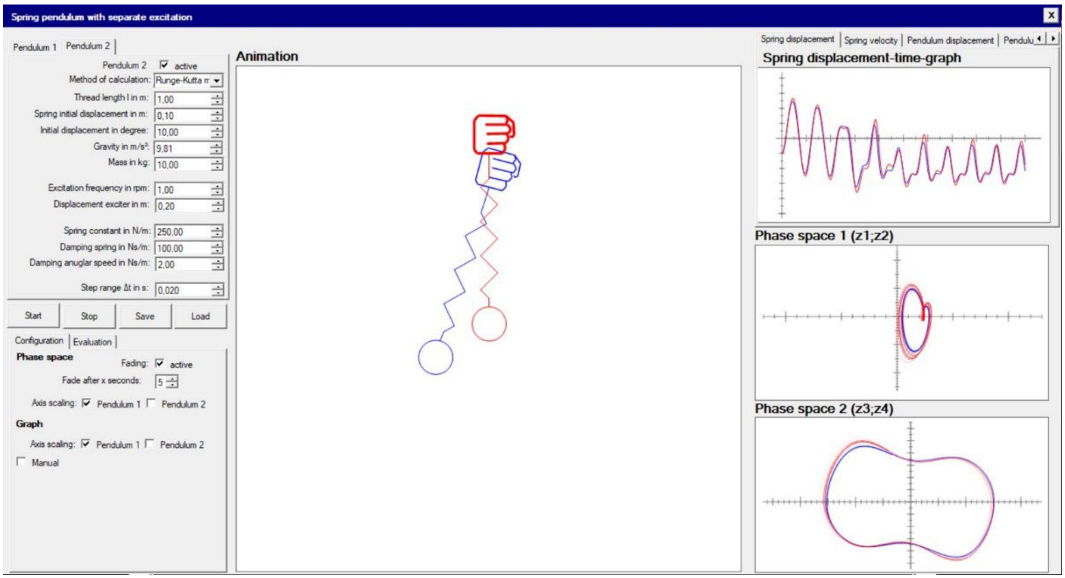

Figure 4: Excited oscillation of a spring pendulum simulated with different integration methods

The spring elongation $r(t)$ and pendulum deflection $\vartheta(t)$ time histories are depicted in Figures 5 and 6. The graphs derived by the first-order explicit Euler method are colored red and with the fourth-order Ruge-Kutta method blue, respectively.

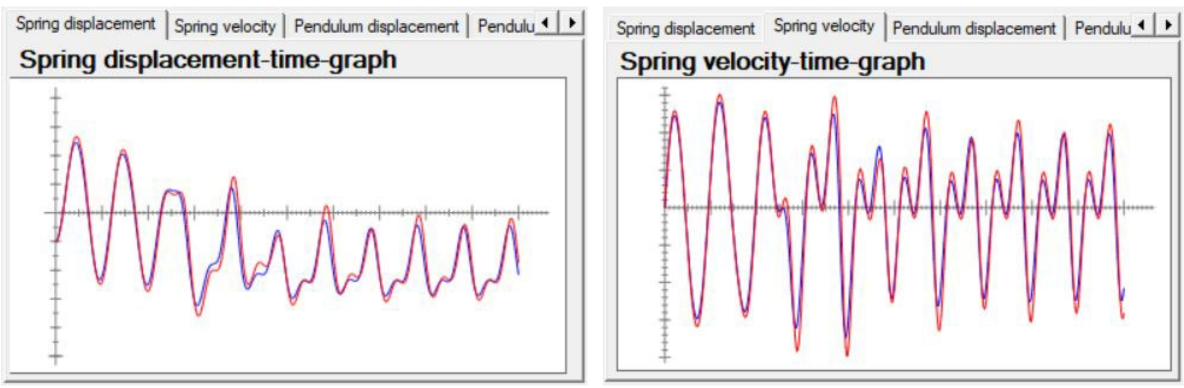

Figure 5: Spring elongation $r(t)$ time histories, computed with Euler's (red) and RK4 (blue) method

Figure 6 presents the associated pendulum deflection $\vartheta(t)$ and angular velocity $\dot{\vartheta}(t)$ time histories. Due to the periodic motion, the deviations caused by the different integration methods do not grow exponentially.

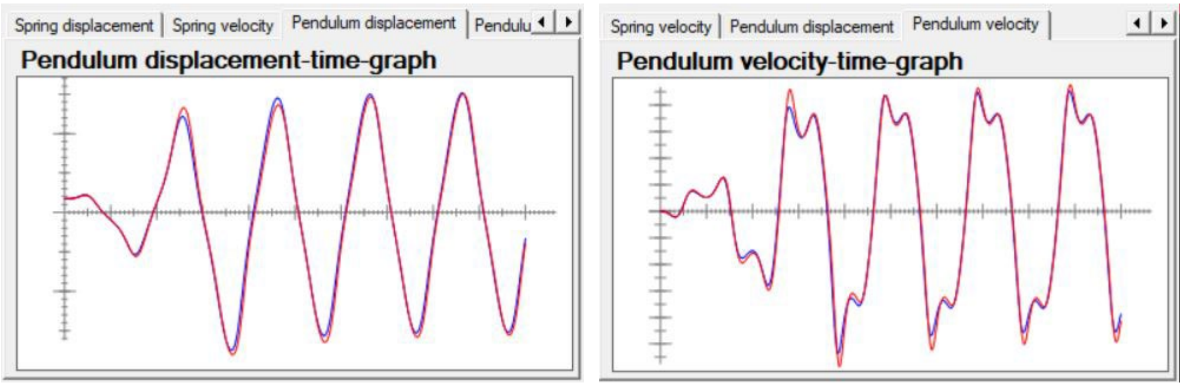

Figure 6: Pendulum deflection $\vartheta(t)$ time histories, computed with Euler's (red) and RK4 (blue) method

Figure 7 shows the phase space portraits $\dot{r}$ over r and $\dot{\vartheta}$ over ϑ for two spring pendula with different parameter settings. Those diagrams would become smeared out after some time in the case of chaotic behavior. In order to avoid this, the settings can be changed so that the trajectories fade to a pale blue or red for values further in the past.

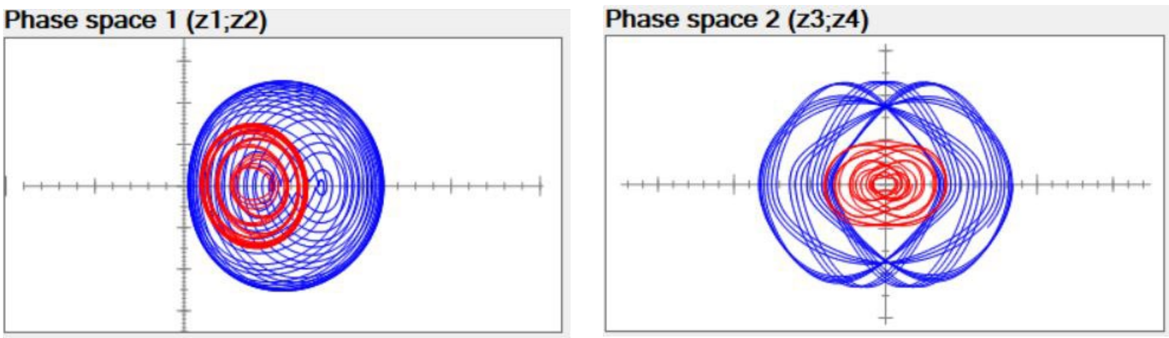

Figure 7: Phase portraits $\dot{r}$ over r (left) and $\dot{\vartheta}$ over ϑ (right) for different parameter configurations

<u>Three-dimensional elastic pendulum</u>

The second, competing, project group chose a different approach. Their goal was to touch the program users emotionally by perceiving the movement of the spring pendulum as persuasively as possible. That is why they have designed a three-dimensional (3D) pendulum that can move not only transversely to the observer, but also towards or away from him or her. To reinforce this impression, they have designed the program for a virtual reality in which the observer can immerse himself with the help of virtual reality (VR) glasses. However, it can also be used without such glasses as a projection on a computer screen.

The project team has therefore decided to use the game engine *Unity* for the visualization of their computer animation [11]. Unity is one of the most widely used game development platforms in the world and offers a range of tools for the development of 3D games. In addition, it can also be used for learning and training purposes. Its development environment, i.e., the Unity editor, is modelled on the basis of common 3D animation programs. A main window represents the 3D scene; various menus and forms allow the manipulation of the *camera* – a device through which the player views the world – and of the scene. Parts of the scene can be selected, scaled, moved and rotated with the mouse. A Unity application is based around a scene graph, formed of *game objects*. Components like materials, sounds, physical properties, and scripts can be assigned to these game objects. Simple objects such as light sources or geometric primitives (planes, cubes, spheres) can be created directly in the editor. Complex components like 3D models, animations, textures and sounds (so-called *assets*), on the other hand, are created in supplementary programs – like *Blender* [12], a frequently chosen addition to the game development toolset – and can be imported via drag and drop. If they are changed in the course of the production of the game, the Unity editor updates them automatically. Graphical representation and behavior of the game are simulated in the "game view", and an export function enables the creation of executable applications.

The mechanisms built into Unity can be supplemented via self-written programs, so-called scripts. Scripts are necessary to describe gameplay and logic and comprise in the present case the numerical solution of the equations of motion (10) to (12). This had to take place mutatis mutandis, because Unity uses a left-handed coordinate system. Scripting in Unity is based on Mono and provides also C# as a possible scripting language, which has been reasonably chosen by the project team.

Figure 8 shows the architectural design of the virtual 3D spring pendulum world. The pendulum room is an open-roof hall, which enables the illumination of the scene by sunlight.

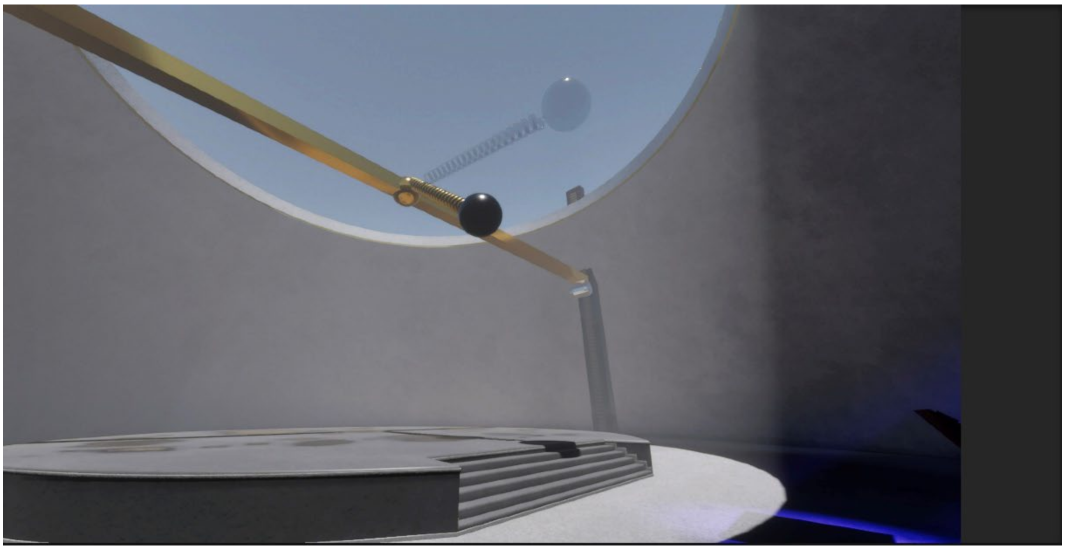

Figure 8: Three-dimensional spring pendulum animation via Unity

In the center of the room, a height-adjustable cross beam is mounted on two support columns and carries the suspension point of the pendulum. A base pedestal below the pendulum invites the user to climb up to the pendulum bob. The pedestal is surrounded in a semi-circle by flat screens on which the pendulum and spring deflection graphs and the phase space diagrams can be viewed (Figure 9).

Since Unity does not have a standard graph library, a custom-made graph script had to be created for the diagrams. The students opted for a point display because lines would have taken up too much computing time in the game environment. In front of the platform is a console with pressure switches, with which an external excitation of the spring pendulum in the vertical direction can be activated and deactivated.

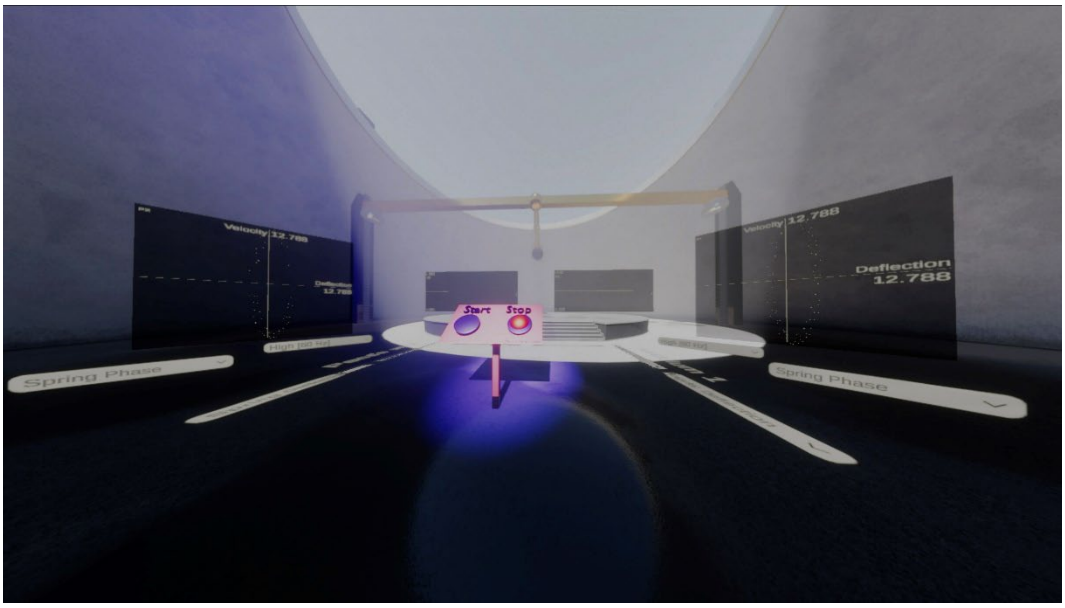

Figure 9: Pendulum Hall with flat screen monitors on the walls and tooltips on the floor

As with the planar spring pendulum, a second pendulum with different simulation parameters can be displayed simultaneously. The project team opted for a semi-transparent, almost ghostly representation, as shown in Figures 8 and 10.

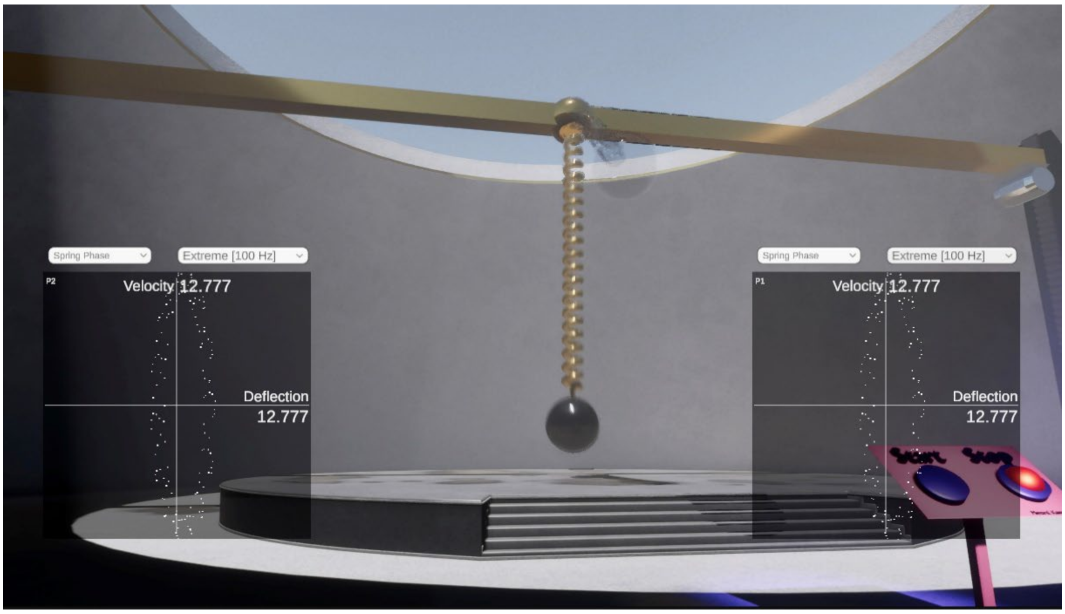

Figure 10: Pendulum with activation and stop buttons for the external excitation $\tilde{z}(t)$

It is also possible to change the input parameters during program execution, which immediately affects the pendulum behavior. In Figure 11 a student, immersed in the virtual reality of the pendulum room, is pointing at the pendulum using his hand controllers.

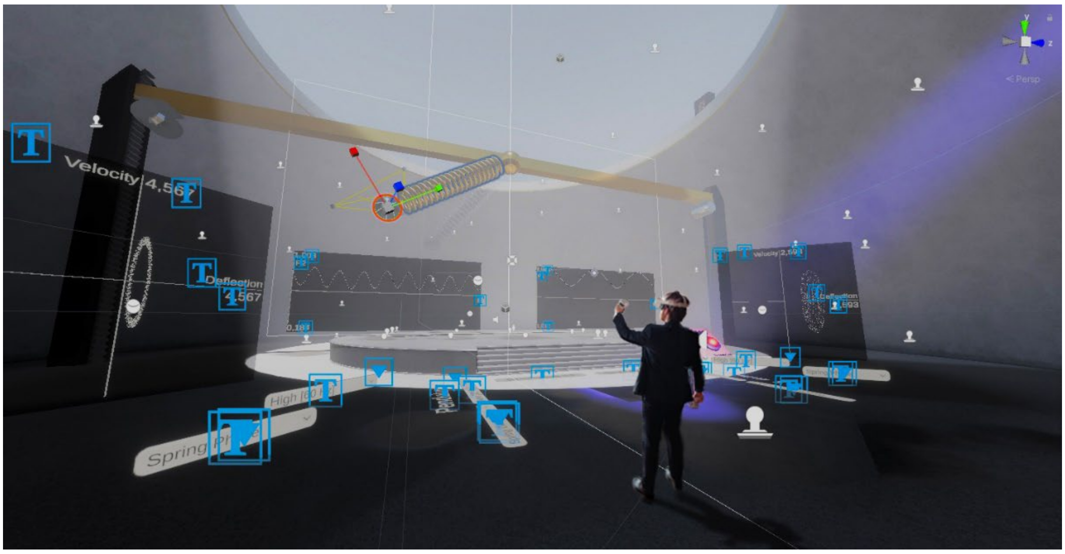

Figure 11: Virtual reality environment with a student avatar in the pendulum room

As an "icing on the cake", the pendulum simulation generates wind noise, in which the Doppler effect has also been integrated; the sound becomes higher as the pendulum moves towards and lower as it moves away from the observer. In addition, a ray tracing render pipeline is used for the physically correct illumination of the scenario.

Impact on students' learning success

The description of the student projects shall be completed by some thoughts on the different visualization strategies of the student teams and the impact of our team-oriented project-based learning concept on the learning success of our students.

The elastic pendulum was one of the two project topics chosen by only two student teams. This was, however, still sufficient to initiate a competition of ideas and thus boost the motivation of the teams. Due to the fact that the technical implementation, i.e., the numerical solution of the equations of motion, was predetermined, the students' creativity essentially flowed into the presentation of the results and the visualization. Here it turned out fortunate that one team had already some experience with Unity, and that VR glasses could be made available. In order to be able to fully exploit virtual reality, the project team accepted the additional effort of programming a three-dimensional pendulum configuration. The other group countered this by placing an easily comprehensible presentation and a better comparability of the results at the center of its efforts. This led to the significantly different visualization results of the two groups.

In addition, we would like to address a question that frequently comes up in connection with our project-based learning concept, namely that of its impact on the students' learning success and course outcomes.

The Institute's faculty tries to invent every year new project topics in order to give the students the certainty of working on a new problem that has not yet been solved by students in previous classes. This leads to a wide variety of project topics, which thus has different effects on the lessons learned and experience gained by our students. However, all projects have in common that the team orientation promotes the development of certain generic skills strongly required by industry, like the ability to work in teams, to keep records and to meet deadlines.

An assessment of the learning effect through our project-based teaching method in the context of a comparative study proves difficult, in particular because we do not have a reference group of students who do not participate in these projects. However, there is no doubt that the software projects offer the opportunity to show our students quite plainly the value of the just learned methods and algorithms, thus increasing their attentiveness in the accompanying lectures. Furthermore, they give them the chance to look beyond the standard curriculum of engineering education.

And, in addition, an indirect evaluation of the learning success can be provided by the above-average performance of our students in subsequent project work, which culminates in the development of racing cars within the framework of Formula SAE [13] (in Europe: Formula Student [14]). In this international competition, our student teams are consistently placed in the top ranks [15]. Occasionally it happened that subtasks from this project have been outsourced to our programming projects [16].

Summary and Conclusions

Starting from their freshman year, our students are involved in project work within the framework of team-oriented project-based learning. Software projects supplemental and complementary to mathematics, physics and engineering courses are part of the curriculum in the early phases of our undergraduate degree program. The software project introduced in this paper was the numerical simulation of the dynamic behavior of a spring pendulum, i.e., a mathematical pendulum with an elastic suspension.

The project was completed by two teams of engineering students in their second semester of study. Since we pay particular attention to a good and comprehensive education in mathematics and computer programming in the first year of undergraduate study, an above-average level is achieved by our students in these subject areas. The mathematics education in the second semester focuses on ordinary differential equations and on analytical and numerical methods for solving them. In combination with the mastery of a high-level programming language like C#, numerical simulations of complex systems are therefore already accessible to second semester students.

The two competing teams were required to solve the equations of motion numerically and to visualize the motion of the system for arbitrarily variable parameters. The students were given free rein on creativity, which lead to equivalent computational results but significantly different visualization designs. One group focused on a planar spring pendulum and chose a rather technical approach. Their C# program offers several visualization options of the system’s behavior and the simultaneous comparison of different parameter sets. The other group used the Unity game engine for the visualization of a three-dimensional elastic pendulum. The motion of the system can be observed on the computer monitor or through a virtual reality viewer as an almost tangible object in a realistic looking virtual environment. By using a VR headset, the users have the possibility to immerse themselves into in the virtual world of the spatial spring pendulum.

Both teams have met the demands placed on them in a creative way, despite the restrictions imposed on them by the Covid-19 pandemic. The programs can be obtained free of charge from the student teams on request.

Authorship declaration and acknowledgments

The planar spring pendulum computer program presented in this paper was developed by Maximilian Sterkl, Christoph Tröster, and Moritz Vogt, and the three-dimensional elastic pendulum by Thomas Kainz, Eric Christian Menard, and Robert Poetsch, both under the supervision of Günter Bischof and Christian Steinmann. The last-mentioned authors (G.B. and C.S.) would like to express their sincere gratitude to all participating students for their high motivation and excellent performance during their project work.

Bibliography

[1] R. Howe, *The AMS and Mathematics Education: The Revision of the "NCTM Standards"*, Notices of the AMS, Volume 45, Number 2 (1998)

[2] J. E. Quinlan, *Profiles of software utilization by university mathematics faculty*, Dissertation, Ohio State University, Columbus, OH (2007)

[3] C. L. Tokpah, *The effects of computer algebra systems on students' achievement in mathematics*, Dissertation, Kent State University, Kent, OH (2008)

[4] E. Bratschitsch, A. Casey, G. Bischof, and D. Rubeša, *3-phase multi subject project-based learning as a didactical method in automotive engineering studies*, Proceedings of the ASEE Annual Conference and Exposition, Honolulu, HI (2007)

[5] S. S. Sazhin, *Teaching Mathematics to Engineering Students*, Int. J. Engng Ed. Vol. 14, No. 2, 145 (1998)

[6] G. Bischof, E. Bratschitsch, A. Casey, and D. Rubeša, *Facilitating engineering mathematics education by multidisciplinary projects*, Proceedings of the ASEE Annual Conference and Exposition, Honolulu, HI (2007)

[7] V. Ciocanel, *Modeling and numerical simulation of the nonlinear dynamics of the forced planar string pendulum*, Dissertation, Duke University, Department of Mathematics, Durham, NC (2012)

[8] P. Lynch, *Resonant motions of the three-dimensional elastic pendulum*, International Journal of Non-Linear Mechanics 37, 345-367 (2002)

[9] G. Bischof, A. Zwölfer, and D. Rubeša, *Correlation Between Engineering Students' Performance in Mathematics and Academic Success*, Proceedings of the ASEE Annual Conference and Exposition, Seattle, WA (2015)

[10] C# by Microsoft, Accessed Jan 31, 2022. https://docs.microsoft.com/de-de/dotnet/csharp/

[11] Unity by Unity Technologies, Accessed Jan 31, 2022. https://unity.com/

[12] Blender by Blender Foundation, Accessed Jan 31, 2022. https://www.blender.org/

[13] Formula SAE, Accessed April 11, 2022. https://www.fsaeonline.com/

[14] Formula Student, Accessed April 11, 2022. Institution of Mechanical Engineers, http://www.formulastudent.com/

[15] Joanneum Racing Team, Accessed April 11, 2022. https://www.joanneum-racing.at/language/en/home/

[16] G. Bischof, et al., *The impact of the formula student competition on undergraduate research projects*, Proceedings of the 39th IEEE Frontiers in Education Conference, San Antonio, TX (2009)