

Designing maths-based STEAM activities for civil engineering curricula

Rolando Chacón
School of Civil Engineering
Universitat Politècnica de Catalunya
Barcelona, Spain
ORCID 0000-0002-7259-5635

M. Rosa Estela
School of Civil Engineering
Universitat Politècnica de Catalunya
Barcelona, Spain
ORCID 0000-0002-4717-9111

Abstract—This work-in-progress innovative practice describes a set of pedagogical activities designed for students of 1st, 2nd and 3rd year of civil engineering programs. The pedagogical project has a twofold aim: i) infusing fabrication, coding and instrumentation in civil engineering courses and ii) fostering motivation and creativity among students by merging STEAM and Maths. For this purpose, a set of hands-on activities described herein encompasses concepts of calculus (functions, curves and derivatives), concepts of surveying (construction and measurement using sensors and microcontrollers by means of a fabrication of a low-cost Lidar) and concepts of algorithmic programming in which measurement and programming are blended together. Three workshops coupled with multimedia documents are prepared. Firstly, a workshop related to programming mathematical curves and derivatives in the coding platform Processing whose aim is to reinforce coding skills to our students. Secondly, a workshop related to the development of the Lidar using electronic prototyping boards such as Arduino whose aim is to introduce instrumentation skills to them. Thirdly, a workshop related to the use of algorithmic programming design platforms such as Grasshopper whose aim is to couple measurement and coding together. These activities are conceived for formal infusion in the next generation of civil engineering courses to be deployed at our school from 2020 onwards.

Keywords—STEAM, Civil Engineering, Workshop

I. INTRODUCTION

Engineering educators are continuously challenged for cultivating technical competencies in students that are faced to a continuously changing professional sector. The field of civil engineering faces grand challenges when it comes to digitalization of the sector. As a result, the present generation of civil engineering students encounter massive advances in sensing, information and computing technologies that continuously provide to the Architecture, Engineering and Construction sector (AEC) ways for dealing with the large variety of technical information at all project stages. The whole project life cycle, from conception, design, construction, to operation and maintenance is continuously infused with a rising amount of ubiquitously generated and distributed data. New sensors, new forms of visualization and new interoperable design tools are increasingly intertwined. The manifold forms of visualization, information modelling and simulation (VIMS) are called to be part of centralized information hubs in which all stakeholders establish data flow seamlessly [1].

Instrumentation, data-gathering, sensing, automation and construction robotics in civil engineering education has been pinpointed as a need for several years [2][3]. Nevertheless, civil engineering schools worldwide has not yet massively integrated instrumentation, sensing technologies, automation and data integration within formal curricula. Even though technologies have allowed proper implementation of laboratory-based environments, their cost have been prohibitive for a massive application at undergraduate levels. As a result, academic experiences in which attempts to infusing such technologies in AEC education are scarce.

Education-wise, a laboratory-based pedagogy in science and engineering has been one of the greatest and most effective tools [4]. Physical experiences foster cognition and provide effective skill acquisition. Teaching labs are used in civil engineering to let the students discover and measure phenomena. Classical subjects are often provided with laboratory sessions in which students manipulate samples and develop reports on experiments in which magnitudes with practical applications are characterized. However, equipment is often expensive and limited. Consequently, students seldom develop their own measurement devices. In last years though, with the advent of low-cost electronics, the development of open-source labs has become plausible in several fields [5][6]. In recent years with the disruption of IoT, several AEC academic examples of implementation of low-cost systems in cities [7], buildings [8] water supply chain [9] or monitoring of historical constructions [10] have been published. When it comes to education, the development of IoT devices in civil engineering classrooms shows great potential due to the increasing affordability, both technical and economical [11]. AEC colleges and universities are increasingly infusing electronics, fabrication, sensors and coding. Examples in the fields of structural analysis [12-13], structural dynamics [14-15] and responsive architecture [16] show the great potential in many of the disciplines of the AEC sector.

This paper describes a set of pedagogical activities designed for students at the beginning of civil engineering bachelor degrees. The pedagogical project has a twofold aim: i) infusing fabrication, coding and instrumentation in civil engineering courses and ii) fostering motivation and creativity among students by merging (S)(T)(E)(A)(M) concepts. The activities are focused on the development of a set of workshops. During the 2019-2020 academic year, the first edition of these workshops has been under development. A description of the tasks performed so far is presented in the following sections. Each one of the components of the acronym are pinpointed throughout the text.

II. DRAWING DERIVATIVES

The first workshop deals with an activity in which functions, derivatives and an introductory programming challenge are intertwined. Applied introductory programming are always useful for civil engineering students due to their uneven prior acquaintance with coding skills.

The workshop is aimed at reinforcing concepts of differentiation throughout a hands-on coding activity. The formal Calculus syllabus (M) including derivatives is illustrated by students by designing a beautiful visual application (A). Students enrolled in the course are given an optional yet graded task. The challenge of the activity is: For a given implicit function, an interactive visual application must be developed. This application requires i) drawing the curve and ii) drawing a tangent line to the curve as the mouse pointer is placed on any of its points (x,y) . The first requirement is a straightforward static code in which points of coordinates (x,y) are plotted in a Canvas. The second requirement is more advanced dynamic interactive code that presumes an adequate use of time and space within the snippet. Students are entitled to use creativity and beauty in the development of such applications by plotting moving primitives (points, lines, rectangles, circles, triangles) in a 2D space with a creative formatting.

A. Preparation of the workshop

In order to set an adequate scope of the activity in a regular course (that generally has more than 100 students), the workshop was firstly tested with a limited amount of 2nd and 3rd year students during 6 hours. These students had no prior knowledge in coding but had passed successfully the Calculus course. During these hours, lessons for introductory programming of visual applications were taught using the platform Processing [17]. Processing is a programming language, development environment, and online community that promotes coding literacy within the visual arts as well as visual literacy within coding. The platform is considerably used by artists for the development of object-oriented, event-driven interactive applications in a diversity of realms such as mapping, music, animation, physics or installations. For the depicted activity, the full capability of the platform may seem overwhelming but its simplicity (and further potential in other workshops) gave to the facilitators strong reasons to use it. The language syntax is identical to Java but with a few modifications. Time and space are blended together intuitively.

Lessons were conceived for students with no prior knowledge in coding visual applications and were structured in six sections as shown in Table 1.

Table 1. Structure of the lessons for Workshop 1

Space	Time	Primitives
The Canvas The unit: Pixels The coordinates	Frames per second Milliseconds Delay	Points, Lines, Vectors Rectangles, Squares, Circles, Triangles
Data types If/then/Else, For, While Functions	Background, RGB, Color, Fill, Stroke	Translation Rotation
Basic instructions	Format	Movement

A basic structure of a visual program in Processing follows two functions: i) `setup()`, in which aspects such as size of the Canvas (in pixels), definition of path to files or to serial communication and other fixed parameters are established. This function executes once and must contain not invariable parameters. Subsequently, ii) the function `draw()` which runs indefinitely. It runs according to the order of written commands from top to bottom and repeats in a loop. The speed of execution between each loop can be regulated by using the delay command.

The first edition of the workshop has been set during the 2019-2020 academic year for 1st year students of Calculus at Bachelor levels. As documentation, a set of pdf files as well as an online video tutorial were given to the students. Other online resources available were recommended [17]. Unfortunately, face-to-face tutorships were not allowed during this edition due to Health Regulations in the country. As a result, only web meetings were held during the experience.

When it comes to numbers, a total amount of 158 students were enrolled. 50 curves of implicit functions were identified (cardioid, nephroid, lemniscata, cissoids, etc). The activity was developed by groups of up to 3 students. Each group was given a curve from the pool randomly.

It is worth pointing out that the Calculus course of first year has 6 European Credit Transfer System (ECTS) with an approximate amount of 3 hours of lectures per week during the whole academic year. The activity was set when the course had reached 75% of its regular path and represents an optional marking grade of 10% of the total.

B. Outcome and expected results

Results of the activity in 2019-2020 edition are available. During the development of the activity the students i) plotted the curve in the canvas, ii) found the differentiation of an implicit curve (using chain rule and other concepts) and iii) used the derivative function for drawing a tangent line whose angle of rotation must be in accordance to the coordinates generated by the mouse pointer $(x,y)=(mouseX, mouseY)$. Maths and interactive programming were successfully blended together as a learning activity.

Fig. 1 shows an example of a Lemniscata and its tangent developed by one of the participants. A simple plot of the function in a 2D space is presented. Axes are drawn using lines and triangles and the function is plotted using points. The tangent line appears only when the mouse pointer is set on the curve at any point. A text indicating the slope is added for verification purposes.

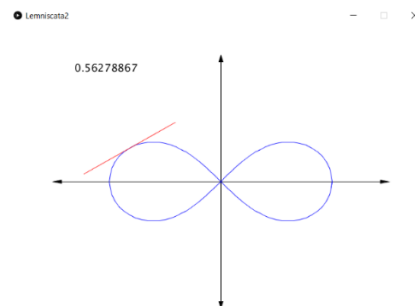


Fig. 1. Typical results obtained with a curve and its tangent line

III. TERRESTRIAL LASER SCANNER

Nowadays, Terrestrial Laser Scanners (TLS) are used in construction facilities for several surveying applications. TLS are also referred to terrestrial LIDAR since they represent a ground-based version of the airborne Laser Imaging Detection and Ranging. A TLS acquires (x,y,z) coordinates of numerous points as well as light intensity α by emitting laser pulses toward these points and measuring the distance from the device to the target. In its simplest form, measurements are based on the rotation about two axes (vertical and horizontal) and the distance measured by the laser which results in a spherical coordinate system. Initially, all points are referenced to it. Subsequently, by proper transformation, all points can be referenced to a rectangular system whose origin is the device itself. Fig. 2 displays a conceptual image of measurement the coordinates of an arbitrary point P. Professional TLS are sophisticated and costly devices that may gather millions of points coordinates in a very limited period. The number of measurements per second with high accuracy is quite high. Applications in geomorphology [18] and building technology [19] are pinpointed. Educationally, TLS are limited yet occasionally included within site and laboratory exercises of students in courses of topography, surveying and geomatics. However, not all students are allowed to manipulate TLS.

Geomatics begins at our School at 2nd year of Bachelor. The second activity consists of the development of a workshop for the preparation of a DIY TSL using electronic prototyping platforms and 3D printing. The workshop encompasses a set of activities in which fabrication and 3D printing (E), use of sensing technologies and electronics (T) and a critical scientific appraisal of results (S) are put together.

A. Preparation of the workshop

In order to set an adequate scope of the activity in a regular course (that generally has more than 50 students), the workshop was firstly presented to a limited amount of 2nd and 3rd year students during 8 hours. These students had no prior knowledge in electronics but had passed successfully the Geomatics course. During these hours, lessons for introductory electronics were taught using the platform Arduino [20]. Electronic prototyping platforms include a range of electronics such as programmable boards, sensors, mechanical parts, simple open-source software that can be afforded by the laboratory facilities (both technically and economically). A vast amount of online content and resources for students is nowadays available. The platform is considerably used by DIYers and STEAM-based educational programs. Object-oriented, event-driven interactive applications in a diversity of realms can be developed. The language syntax is identical to Processing, which was one of the reasons for choosing both resources together. Table 2 displays the structure of the required lessons.

Table 2. Structure of the lessons for Workshop 2

LEDs Knobs	Analog (LDR, Accel) Digital (Laser, Ultrasound)	Servomotor
Basic circuitry	Sensors	Actuators
Technical requirements Sketching platform	Measurement accuracy Rotation	Data types Functions
3D printing	Calibration	Coding

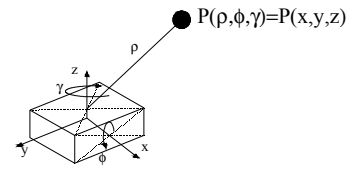


Fig. 2. Schematics of spatial coordinates on a TLS

Electronics-wise, the DIY TLS consists primarily of 3 parts: i) two servomotors that generate rotations γ and ϕ , ii) a laser sensor that measures the distance ρ , iii) an Arduino microcontroller on which a bespoke set of commands for measuring is uploaded. A basic structure of an Arduino code follows two functions: i) `setup()`, in which aspects such as initialization of functions, serial communication or other fixed parameters are established. This function executes once and must contain not invariable parameters. Subsequently, ii) `loop()` which runs indefinitely. It runs according to the order of written commands from top to bottom and repeats in a loop. The speed of execution between each loop can be regulated by using the `delay` command. We can see then than students acquainted with **workshop 1** follow a sort of continuation of tasks when both platforms are used.

B. Outcome and expected results

The available results of this at the time this version of the paper is submitted are related to a workshop developed with a limited amount of students. Limitations concerning face-to-face activities and health regulations during the present academic year undermined its implementation. The next academic year 2020-2021, it is expected to implement a workshop with students from the course of Geomatics (2nd year). It is worth pointing out that results are quite promising. Students were able to develop the TLS in due time. The structure displayed in table 2 is the result of a set of iterations done by the authors for balancing and adapting content and complexity to civil engineering bachelor students.

In the first edition of the workshop the students should: i) connect basic circuitry of the sensors and servomotors, ii) fabricate and 3D print bespoke parts and iii) generate a .txt file with measurements of a simple geometry using the Arduino Integrated Development Environment. Fig. 3 shows some of the hitherto obtained results.

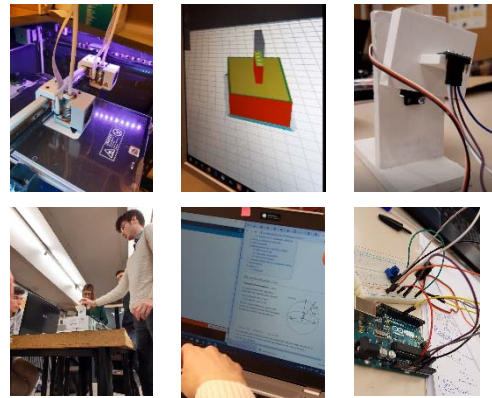


Fig. 3. Results obtained during trials for the workshop 2. TLS

The development of a TLS from scratch by students is expected to become a learning activity in Geomatics (2nd year).

IV. ALGORITHMIC PROGRAMMING

In the AEC sector, many designers are exploring the use of algorithmic programming platforms that encompass computer-aided parametric design, structural analysis and Building Information Modelling (BIM). Namely, currently available popular Software for parametric representation of geometry, such as Grasshopper in the Rhino environment [21-22], or Dynamo in the AutoCAD environment [23-24] provide “parametric representation” of a geometrically- and semantically-rich instance in which the user can define a complex geometric model by incorporation a sequence of algorithmic calculations. That is to say, changing parameters on such models can trigger a cascading recalculation of the output, similar to the recalculation of a spreadsheet but on a CAD environment.

Workshop 3 is aimed at providing a common ground for workshops 1 and 2 as well as for (S)(T)(E)(A)(M) components. The chosen platform is Grasshopper as a versatile algorithmic programming environment. The professional sector is increasingly demanding competences in the use of algorithmic modeling but their use in academia is still in its infancy (in civil engineering schools at least). This workshop is presently under study by the authors and during the production of this paper, the trial version is being taught only online.

A. Preparation of the workshop

In order to explore adequate scope and limitations of the activity in a regular course, the workshop is under development with 2nd and 3rd year students that have performed workshops 1 and 2 already. These students had no prior knowledge in algorithmic programming but are acquainted to Calculus, Geomatics and coding.

The main idea is to establish a link between measurements from real structures, connectivity with sensors and TLS and drawing real curves in a BIM environment. In other words, students should build a structure, measure its shape and draw its digital twin using algorithmic programming. Deploying measurements from TLS in Grasshopper can be done in many ways thus an exploration of different methods has been necessary. Data in (x,y,z) text form can be read and transformed to points instances of the “Rhino.Geometry” set of classes. Rhino.Geometry wraps a set of methods that are applicable to several geometrical instances such as 3D points, lines, planes, surfaces and volumes using Python scripting. The result is a geometric language in which the designer develops complex geometries mathematically. Table 3 displays the present structure of the lessons. The final structure of such workshop will be defined after scrutinizing the results obtained in the trial version of the workshop.

Table 3. Structure of the lessons for Workshop 3

Points, lines, planes, surfaces, volumes	Draw functions Draw derivatives Draw tangent planes using gradient vectors	Firefly firmata Arduino sync
Grasshopper	Functions in GH	Connection
Text files Real measurements	GH and Python. List, tuples, dictionaries	Output Interoperability
Import point cloud	Coding	BIM

B. Outcome and expected results

The expected outcome in this case is an outdoors activity. Students must build real surfaces using fabrication. These shell structures may have complex geometries even with simple fabrication. One can imagine a square fabric simply supported on its 4 sides on which a weight is put at a random position. The surface that is formed represents a complex shape. Once the structure is built, it will be necessary to acquire data. A DIY TLS represents an ideal device for generating a .txt file or alternatively, to provide real-time measurements. The result can be imported by Grasshopper. Fig. 4 shows a typical window of an algorithmic programming snippet. One can see boxes and links between those boxes, which are conceived for treating certain types of input and for generating certain types of output. The result of these codes is visualized in Rhino as shown in Fig. 5.

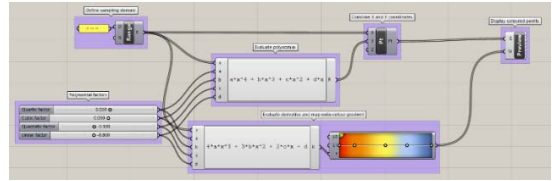


Fig. 4. Schematics of spatial coordinates on a TLS

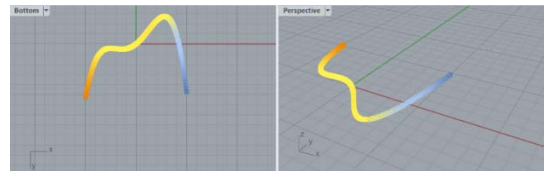


Fig. 5. Schematics of spatial coordinates on a TLS

It is then required to merge data from sensors and geometries within Grasshopper. Fig. 6 displays a schematic representation of this connection. Figure 7 displays the potential result in the outdoor activity. All ingredients for the third workshop are available. The activity is expected to become a learning workshop for new Bachelor degree of Civil Engineering, in which these activities will be infused in optional courses for the generation of Digital Twins in Construction.

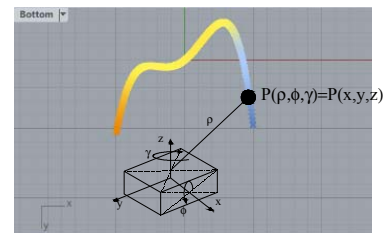


Fig. 6. Schematics of spatial coordinates on a TLS

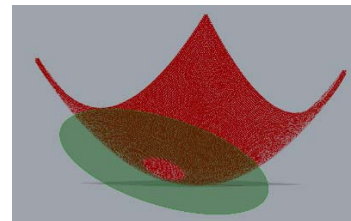


Fig. 7. Tangent plane on surfaces visualized in Rhino

V. CONCLUSIVE REMARKS

In this paper, a set of workshops encompassing fabrication, coding and instrumentation in civil engineering courses are presented. The workshops are conceived in a way added technical competences are added to motivation and creativity among students by merging (S)(T)(E)(A)(M) concepts. Blending together these concepts is becoming paramount in civil engineering schools since the present generation of civil engineering students encounter massive advances in sensing, information and computing technologies that continuously provide to the Architecture, Engineering and Construction sector (AEC) ways for dealing with the large variety of technical information at all project stages.

ACKNOWLEDGMENT

The authors acknowledge the financial support provided by the Institut de Ciències de l'Educació de l'Universitat Politècnica de Catalunya in the frame of the annual grants for innovative practices for the project "de las MATES al STEAM". Likewise, the authors acknowledge the commitment of the students working in the project

REFERENCES

- [1] F. Leite, Y. Cho, A. Behzadan, S. Lee, S. Choe, Y. Fang, Y. R. Akhavian, and S. Hwang. "Visualization, Information Modeling, and Simulation: Grand Challenges in the Construction Industry". *Journal of Computing in Civil Engineering*, vol. 30(6), 2016.
- [2] W. Boles, J. Wang. "Construction Automation and Robotics in Civil Engineering Education Programs". *Journal of Professional Issues in Engineering Education and Practice*, vol. 122(1), pp. 12-16. 1996
- [3] S. Christodoulou. "Educating Civil Engineering Professionals of Tomorrow." *Journal of Professional Issues in Engineering Education and Practice*, vol. 130(2), pp. 90-94. 2004
- [4] L. Feisel and A. Rosa. "The Role of the Laboratory in Undergraduate Engineering Education." *Journal of Engineering Education*, vol. 94(1), pp. 121-130. 2005
- [5] J. Pearce. "Building research equipment with free, open-source hardware." *Science*, vol. 337(6100), pp. 1303-1304, 2012.
- [6] F. Bouquet, J. Bobroff, M. Fuchs-Gallezot, and L. Maurines. "Project-based physics labs using low-cost open-source hardware". *American Journal of Physics*, vol. 85, pp. 216-222, 2017.
- [7] A. Alavi, P. Jiao, W. Buttlar and N. Lajnef. "Internet of Things-enabled smart cities: State-of-the-art and future trends". *Measurement*, vol. 129, pp- 598-606. 2018
- [8] M. Karami, G. McMorrow and L. Wang. "Continous monitoring of indoor environmental quality using an Arduino-based data acquisition system". *Journal of Building Engineering*, vol. 16, pp. 412-419. 2018
- [9] J. Paul and W. Buytaert. "Chapter One: Citizen Science and Low-Cost Sensors dor Integrated Water Resources Management". *Advances in Chemical Pollution, Environmental Management and Protection*, vol. 3, pp. 1-33. 2018
- [10] C. Basto, L. Pelà and R. Chacón. "Open-source digital technologies for low-cost monitoring of historical constructions". *Journal of Cultural Heritage*, vol. 25, pp. 31-40. 2017.
- [11] R. Chacón, D. Codony, and A. Toledo, "From physical to digital in structural engineering classrooms using digital fabrication". *Computer Applications in Engineering Education*, vol. 25, pp. 927-937, 2017.
- [12] J. Huang, S.K. Ong and A. Nee. "An approach for augmented learning on finite element analysis". *Computer Applications in Engineering Education*, vol. 27 (4). 2019
- [13] R. Chacón and S. Oller. "Designing experiments using digital fabrication in structural dynamics". *Journal of Professional Issues in Engineering Education and Practice*, vol. 143, pp. 1-9, 2016.
- [14] R. Slocum, R. Adams, K. Buker, D. Hurwitz, H. Mason, C. Parrish and M. Scott. "Response spectrum devices for active learning in earthquake engineering education." *HardwareX*, 4, e00032. 2018
- [15] K. Kensek. "Integration of Environmental Sensors with BIM: case studies using Arduino, Dynamo, and the Revit API." *Informes de la Construcción*, vol. 66(536), e044, 2014.
- [16] R. Chacón, H. Posada, A. Toledo and M. Gouveia. "Development of IoT applications in civil engineering classrooms using mobile devices". *Computer Applications in Engineering Education*, vol. 25, pp. 927-937, 2017.
- [17] <http://processing.org>
- [18] D. Lague, N. Brodu, J. Leroux. "Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z)". *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 82, pp. 10-26. 2013
- [19] P. Kim, J. Chen and Y. Cho. "Automated Point Cloud Registration Using Visual and Planar Features for Construction Environments." *Journal of Computing in Civil Engineering*, vol. 32(2), 04017076. 2017
- [20] <http://arduino.cc>
- [21] Grasshopper. Algorithmic modelling for Rhino. <https://www.grasshopper3d.com/>
- [22] Rhinoceros. Robert Mcneel and associates. <https://www.rhino3d.com/>
- [23] Dynamo. Open source graphical programming for design. <https://dynamobim.org/>
- [24] Revit. Multidisciplinary BIM Software. Autodesk. <https://www.autodesk.com/>