

Interactive STEM Curriculum: Technological Tools and Programming Interface

Manjukumar Patil, Master's Degree Report, The Ohio State University

Abstract—The most important drawback of teaching mathematical equations to the middle school children is the lack of practical examples and interactive tools which can be provided to make concepts easier to grasp. Additionally, on a parallel note, computer programming has become increasingly important in the current era. The amalgamation of programming languages into the STEM curriculum in the early stages of the students' education would expose them to learn and be acquainted with these concepts at a much earlier age. Teaching STEM concepts using interactive learning tools would benefit students to visualizing the concepts in a more intuitive way. Traditional ways of teaching for linear algebra concepts such as linear equations, quadratic equations, and their associated graphs are not sufficient to reach students deeply with these concepts. However, with the use of technology and right tools (Stepper motor and Drone), we can make the curriculum fun, interactive and link the real-world applications of these concepts, and make students engage deeply into the curriculum.

I. INTRODUCTION

The motivation behind the project is to enhance the learning capabilities of the students for the STEM curriculum. The regular methods of teaching STEM concepts would not concentrate on real-world examples [7], [13]. This hinders the learning curve of the students who can't apply these STEM concepts in their day-to-day activities. Hence, we built the curriculum which would educate the simple mathematical equations in linear algebra using the interactive tools and graphical interface [10].

There are two interactive curricula which have been built to demonstrate the linear algebra equations. The first one is the linear equation, $y = mx+c$. This curriculum is the previous work which was built by conducting engineering experiments correlating the physics Ohms law. The Ohms law, $V = IR$, is the direct application of linear algebra with I-current, being the dependent variable and R-resistance, being the slope. The other application of the linear equation is the speed equation $v = at + u$. We used the stepper motor, Arduino UNO, to demonstrate the speed variations in the motor depending upon the changing variables in the equation. This demonstration has been the curriculum for learning linear equation in a few schools around the Columbus area. There were enhancements made to this curriculum addressing the effectiveness of the system.

The second curriculum extends the idea from the first in explaining the quadratic equation, $y = ax^2 + b$ (equation of a parabola) using the interactive Drone experiment with face detection technology. We are utilizing the fact that the face size in an image would increase as we move towards the face. To extract this feature and explain quadratic equations, we make use of the drone [9], [8], which can take pictures of the stationary face mannequin at different intervals.

One of the real-world applications of these technologies is in the surveillance system. Imagine a passive surveillance camera running face recognition algorithm to find a particular face. The placement of the camera and the percentage of the face in the image plays a vital role in recognition. Our drone curriculum simulates this prediction analysis by moving towards the face mannequin until the face size in the image is large enough to be adequately recognized. For every step forward, the face area in these images would increase in a quadratic fashion. These images can be learning parameters which help in understanding the quadratic equation.

Background and Context: This work builds on two seemingly orthogonal lines of work. First, smart classrooms are increasingly seen as most effective when pedagogical concepts are linked instantaneously [7], [6]. This requires extremely fast response time for web pages [18], [4]. Prior work has achieved fast and consistent Internet services through back-end provisioning [11], [12], [14], [15], [16], [17], [20], [1], [2]. In addition, linking back-end and curriculum design is critical [7]. Second, the era of IoT and edge-computing analytics has arrived [?], [?], [?], [?]. These analytics on IoT devices allow practical, hands-on lessons in classrooms [6]. The intersection of these trends still poses problems, i.e., using IoT in classrooms with instantaneous feedback. This project explored systems support for curriculum based on IoT remote-control cars and unmanned aerial vehicles.

Section II provides a detailed explanation of the linear equation curriculum and the challenges faced in improving the performance and reliability of the curriculum. It primarily discusses the inconvenience of codebender, the ino files compiler (programming files of Arduino microcontroller), and the need to migrate to Arduino Create. It also focuses on challenges faced using the

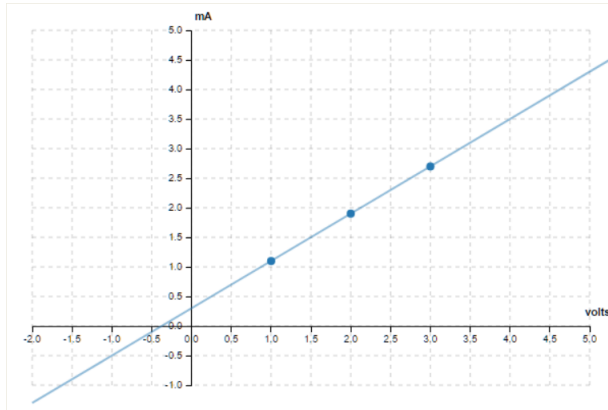


Fig. 1. Graph portal for the linear equation

Arduino UNO microcontroller and the steps taken to overcome it. Section III introduces to the new quadratic equation curriculum using face detection and Drones, its design and implementation. Section IV and Section V covers conclusion and future work of these STEM curriculum.

II. LINEAR EQUATION CURRICULUM ENHANCEMENTS

The linear equation curriculum, as explained in the previous section, is been implemented in the middle school curriculum and has a positive impact on the student's learning abilities [5]. The system is a web application supported by a cloud server that can be accessed from a web browser on any device. Students interactively change the variable values through the programming interface which alters the speed of the stepper motor. These values are noted and plotted in the graph. The graph plotted on the web portal for the stepper motor experiment is shown in Figure 1.

Even though the pilot of linear equation STEM curriculum was successful, there were some challenges in the space of performance and reliability.

One of the challenges is the server which compiles the ino files (compiler which runs the micro-controller code), the Codebender, was not reliable in providing the services. There were several instances where the server was down, and in a few occasions was not behaving the way it supposed to. There was also some ongoing news that the codebender might be deprecated. There was an urgent need in changing the service for compiling the files. We surveyed multiples services and one of the most prominent alternatives was Arduino Create. One of the main reasons why we were using the Codebender is because of its cost. It was relatively cheaper than most of the other compiles. But the Arduino Create is the most suited to compile these ino files, predominantly because the microcontrollers we use, Arduino UNO

and Arduino 101, are made by Arduino group which are all compatible. But, on the other hand, codebender does not support the Arduino 101, and it has a limited set of microcontrollers which it supports. Due to these reasons, migration from Codebender to Arduino Create was essential.

Basic steps involved in this migration are not too difficult since the compatibility was there between these two platforms. Firstly, we need to download all your Codebender Sketches and libraries. Next, put together all the sketches into a zip folder. Then, import ADAFRUIT MOTOR SHIELD LIBRARY and ADAFRUIT MOTOR SHIELD V2 LIBRARY, if they are not present in the inbuild libraries of Arduino Create. These libraries contain the necessary code in compiling the ino files. Finally, you can import all the sketches which are zipped into the Arduino Create platform. Figure 2 shows the working of Arduino Create for the linear equation ino file on Arduino 101.

The other challenging aspect is to maintain the performance of the stepper motor, with it's associated micro-controller, throughout its life span. The microcontroller we used was Arduino UNO which has limited performance standard and battery power. There were two main difficulties in Arduino UNO. One of them is the performance of Arduino UNO which is hindered by the uneven start of the two stepper motor wheels. Over time, due to the physical modifications, the two wheels of the stepper motor tend to start at different times. This leads to the stepper motor to lean towards the left direction or the right direction instead of moving straight. This was the major issue since it would hinder the learning experience of the student. One of the immediate solutions which we found to fix the issue is to provide initial velocity to the slower wheel through a programming interface. In $y = mx + c$, we can provide different values for 'c' to the left and right wheel to manually adjust it to go straight. This programming change was not visible to the student since it was done in other ino file and imported to the working ino file. This ensured that the students won't be exposed to this confusing differentiation in the value of 'c'.

This was the temporary solution since this has to be done periodically with different units for different stepper motors. There were several other stepper motors with a different microcontroller such as Arduino 101, Arduino Zero and EMoRo which has a better specification with respect to performance. The specifications for both Arduino 101 and Arduino zero as compared with Arduino UNO is listed in the Table I. Experimentation with these two microcontrollers several different times resulted in no direction change. Bearing cost in mind, we selected the Arduino 101. The adoption of this microcontroller is already discussed in the above section.

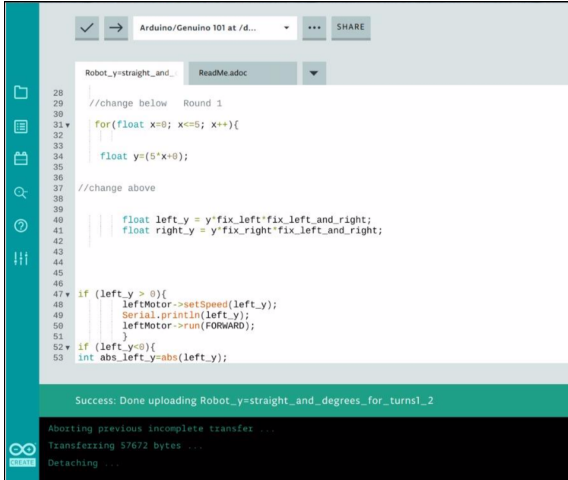


Fig. 2. Arduino Create running on Arduino 101

	UNO	ZERO	101
bits	8	32	32
Processor	AVR	ARM	Intel
I/O pins	14	20	14
Flash Memory	32KB	256KB	196KB
SRAM	2KB	32KB	24KB
Width	53mm	30mm	30mm
Weight	25g	12g	12g
Clock Speed	12MHz	48MHz	48MHz
Built-in LED	No	Yes	Yes
Bluetooth	No	No	Yes
Price	22	39	30

TABLE I
ARDUINO MICROCONTROLLER SPECIFICATIONS

The other difficulty is Arduino UNOs battery power limitation. It has a pack of 2 AA 1.5V batteries. These batteries wear out easily and teachers have to constantly change these batteries. But changing the batteries is not easy since they were packed in between the base of the stepper motor and the Arduino UNO microcontroller. In contrast, Arduino 101 has 4 AA 1.5V batteries. They last longer when compared to the Arduino UNO. But, still, the difficulty of removing the batteries still remains. This becomes increasing troublesome when the need for changing batteries arises between the class which happened in numerous cases. Multiple attempts are conducted to find the best spot for batteries so that the ease of replacement is high. Careful analysis of the stepper motor dimensions and components revealed that there is enough space for the battery right next to the Arduino board which can be attached lightly with the support of tape or even magnet.

III. QUADRATIC EQUATION CURRICULUM DESIGN

The tool developed is the web-based application which could be run by any web browsers. The adoption is for the Chromebooks which are being used in the schools for the first curriculum. This tool uses the same interface

which could be used by accessing the 'Drone' tab on the website.

The demonstration is initiated by the drone capturing the mannequin image. This image is sent to the local machine by the drone using the server. The OpenCV face detection or the dlib face detection algorithm is computed on top of the sent image to find the mannequin face. The rectangular bounding box is formed over the found face. Figure 4 shows the rectangular bounding box on a mannequin. These computations are used to plot the graph at several intervals to realize the quadratic equation. The JavaScript-based graphical interface provides this interaction.

A. Drone setup and actions

In this section, we will first introduce the initial setup of the drone and the drone actions to perform the demonstration. The class which has incorporated the drone curriculum will be equipped with the drone, the mannequin, and the tripod stand. Coordinator of the class is responsible to mount the mannequin on top of a tripod stand and fix the set up towards one corner of the class. The drone should be placed towards mannequin, at a certain distance, with gimbal (drone camera) facing mannequin head.

Now, with this setup, the coordinator can start a drone from its position by selecting the Launch button in the website portal. The drone has a certain predefined set of actions which it performs once it is launched. It takes off from the initial position, stops at the comfortable height where its camera has the clear visibility of the mannequin head. Now, the drone takes the picture of the mannequin and moves towards the mannequin with limited, predefined, equal interval stops. It takes pictures of the mannequin at all these positions which will be later used for plotting quadratic graphs. Finally, after taking the last picture, the drone automatically lands at its current position. Figure 3. Captures this demonstration.

B. Quadratic graph concept using drone images

With these set of images taken by a drone, we have a pattern where the size of the face increases as the drone advances towards mannequin. In fact, there is a quadratic increase in the size of the face as drone takes each step (Size of the face is determined by the bounding square box formed just around the corner of the face). To visualize, suppose the size of the bounding box for the initial image is 4 and for the second image, in which drone has moved one unit forward, is 16 and 64 for the third image and so on. Here with every step, the bounding box square is increasing quadratically. We are using this concept to teach students how to face size increases quadratically with a decrease in distance of drone from the mannequin.



Fig. 3. Drone Demonstration

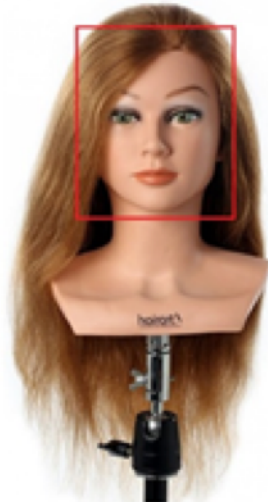


Fig. 4. Sample mannequin head with face box

C. Graph Portal

Figure 5 gives the section of the website portal view as seen by the students during the drone curriculum. Images taken by the drone are rendered to students in their portal. The bottom-left portion of Figure 5 shows all these images which are captured by the drone while moving towards the mannequin. All the images will be tagged with the image number at which position they are taken. Students have the liberty to choose a certain set of images from the pool of images rendered by drone. Suppose, we have predefined that the drone should capture 10 images during its routine, and each

student can choose certain images (numImages) out of 10.

To these selected images, the student must build the bounding face box which can be achieved through coding interface (explained next section). The bottom-right portion of Figure 5 shows the selected images and associated coding for face box creation. The box size is the parameter for the y-axis and the image number (drone distance from mannequin) is the parameter for the x-axis. Top-right portion of Figure 5 shows the graph which will be plotted automatically once the face box is created by the student for all the selected images and the red dots indicate the images which are already scanned. Middle-left section shows the quadratic equation involved in plotting the graph and its variables.

D. Coding choices for face detection

There are multiple coding choices which we can provide to students in identifying the face box. First, the simplest is by providing the drag and fit options on the image. Students can select their top-left tile and drag till bottom-right tile to form a face box. Its just like the image cropping technique for selecting the box.

The second option is the coordinate setting. It is much like the first choice except that students must give the coordinates of the top-left and bottom-right tile. Once the coordinates are given, the face box is formed in the image. Students can change the coordinates to construct their desired box.

The third option is to provide the programming for-loop interface in finding the coordinates for the face box. Initial top-left tile can be fixed by the students. for-loop will help in estimating the bottom-right tile. It is like the second option with the additional for-loop interface.

The final option is to run the face detection algorithms, such as dlib (its a face detection algorithm which generates bounding box), on all the images selected by the student. A student can run a for-loop on all the selected images in running the dlib algorithm. The function will return the coordinates of the face box which can be plotted into a graph.

E. Coordinators

Teacher of the class must know how the drone operates and the actions it performs for the demonstration. Graph portal of the teacher exclusively has the Launch button which starts the drone routine when the setup is ready. Before using the Launch button, the teacher must make sure that the drone can be operated in the classroom and the mannequin must be placed at one corner of the class with the predefined distance from the drone. The teacher must know the application of this curriculum in achieving a certain percentage of face size in the image.

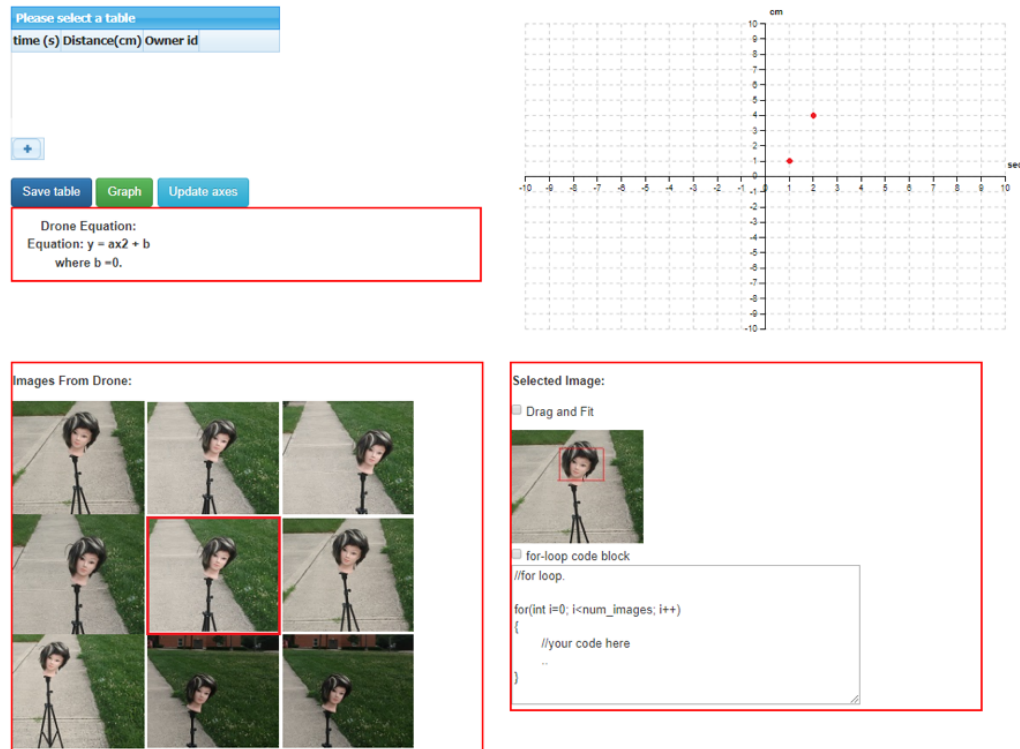


Fig. 5. Website Portal view for Graph and programming interface

IV. CONCLUSION

The advent of technological tools in the STEM curriculum has become increasingly important for the education system [3] [19]. The linear equation curriculum has gained its importance after being implemented in over 5 schools reaching around 700 students. The Drone curriculum for the quadratic equation is on the same idea and is ready to be launched as a stand-alone curriculum in these schools. The presence of technological tools such as stepper motor and drones triggers the interest in students to learn these concepts and subsequently help in grasping the concept easily. These curricula not only brings technology close to students but also introduces the programming languages at an early age. It would help students, if they are interested, choose their carrier in the technology and get an early head start.

V. FUTURE WORK

One of the challenges for these curricula is the online web server which compiles the ino files. Relying on the third party services such as Codebender and Arduino Create, for compiling the ino files is not good in the long run. By removing this reliance, we can completely remove the dependency from other sources. This also eliminates the cost which we provide for the subscriptions. Moving forward, it would be best to develop our own compiler which can compile the ino files.

Linear equation curriculum has been implemented in a few of the schools and has been known to improve the learning capabilities of the students. So, as a next step, we are planning to implement the quadratic equation curriculum, which has been designed on the similar lines of linear equation curriculum, into the classroom curriculum.

The goal is to create the technological curriculum for most of the mathematics and physics equations. Not just equations, rather most of the concepts in linear algebra, geometry, and physics whose proof are not easy to teach from the teachers perspective and not easy to grasp from the students perspective. With the advent in time, we can ideate similar scientific experiments which could demonstrate the complex mathematical equations into practical examples such as stepper motor and Drones.

REFERENCES

- [1] C. Hsu, Y. Zhang, M. A. Laurenzano, D. Meisner, T. F. Wenisch, J. Mars, L. Tang, and R. G. Dreslinski. Adrenaline: Pinpointing and reining in tail queries with quick voltage boosting. In *HPCA*, 2015.
- [2] J. Kelley, C. Stewart, N. Morris, D. Tiwari, Y. He, and S. Elnikety. Measuring and managing answer quality for online data-intensive services. In *2015 IEEE International Conference on Autonomic Computing*, pages 167–176. IEEE, 2015.
- [3] A. Misner. Impact of the 2016 global space balloon challenge on student attitudes towards science and their perceptions of how science is conducted. 2019.

- [4] R. Netravali, A. Goyal, J. Mickens, and H. Balakrishnan. Polaris: faster page loads using fine-grained dependency tracking. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, 2016.
- [5] A. Perez, K. Malone, S. M. Renganathan, and K. Groshong. Computer modeling and programming in algebra. In *CSEDU (2)*, pages 281–286, 2016.
- [6] R. Rao, C. Stewart, A. Perez, and S. Renganathan. Assessing learning behavior and cognitive bias from web logs. In *IEEE Frontiers in Education*, 2018.
- [7] S. Renganathan, C. Stewart, A. Perez, R. Rao, and B. Braaten. Preliminary results on an interactive learning tool for early algebra education. In *IEEE Frontiers in Education*, 2017.
- [8] J. L. Sanchez-Lopez, R. A. S. Fernández, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy. Aerostack: An architecture and open-source software framework for aerial robotics. In *International Conference on Unmanned Aircraft Systems*, 2016.
- [9] J. L. Sanchez-Lopez, M. Molina, H. Bavle, C. Sampedro, R. A. S. Fernández, and P. Campoy. A multi-layered component-based approach for the development of aerial robotic systems: the aerostack framework. *Journal of Intelligent & Robotic Systems*, 88, 2017.
- [10] C. V. Schwarz and Y. N. Gwekwerere. Using a guided inquiry and modeling instructional framework (eima) to support preservice k-8 science teaching. *Science education*, 91(1):158–186, 2007.
- [11] C. Stewart, A. Chakrabarti, and R. Griffith. Zoolander: Efficiently meeting very strict, low-latency slos. In *IEEE International Conference on Autonomic Computing*, 2013.
- [12] C. Stewart, T. Kelly, and A. Zhang. Exploiting nonstationarity for performance prediction. In *EuroSys Conf.*, Mar. 2007.
- [13] C. Stewart, M. Leventi, and K. Shen. Empirical examination of a collaborative web application. In *IEEE International Symposium on Workload Characterization*, 2008.
- [14] C. Stewart and K. Shen. Performance modeling and system management for multi-component online services. In *USENIX NSDI*, May 2005.
- [15] C. Stewart, K. Shen, S. Dwarkadas, M. L. Scott, and J. Yin. Profile-driven component placement for cluster-based online services. In *International Middleware Conference*, 2004.
- [16] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi. An analytical model for multi-tier internet services and its applications. In *ACM SIGMETRICS*, Banff, Canada, 2005.
- [17] A. Verma, L. Pedrosa, M. R. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes. Large-scale cluster management at Google with Borg. In *Proceedings of the European Conference on Computer Systems (EuroSys)*, Bordeaux, France, 2015.
- [18] X. S. Wang, A. Krishnamurthy, and D. Wetherall. Speeding up web page loads with shandian. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 109–122, 2016.
- [19] J. L. Weese. *Bringing computational thinking to K-12 and higher education*. PhD thesis, Kansas State University, 2017.
- [20] Z. Xu, N. Deng, C. Stewart, and X. Wang. Cadre: Carbon-aware data replication for geo-diverse services. In *IEEE ICAC*, 2015.