Enriching Undergraduate Mathematics Curriculum with Computer Science Courses

https://doi.org/10.3991/ijep.v11i5.21701

Chia Hung Kao
National Taitung University, Taitung, Taiwan
chkao@nttu.edu.tw

Abstract—Traditional mathematics curriculum faces several issues nowadays. The gap between course materials and students' real-life mathematical experiences, the scattering of knowledge in different courses, and the lack of mathematics applications to other subjects all hinder the learning of students. The emerging trends in data science, machine learning, and artificial intelligence also impel higher education to enrich and refine mathematics education. In order to better incubate students for future, the experience of enriching undergraduate mathematics curriculum with computer science courses is introduced in this study. The curriculum is designed and implemented for students who major in applied mathematics to better stimulate the learning, participation, exercise, and innovation. It provides students with comprehensive theoretical and practical knowledge for the challenges and industrial requirements nowadays. Evaluations, major findings, and lessons learned from three refined courses are discussed for more insight into the following deployment and refinement of the curriculum.

Keywords—curriculum design, mathematics, computer science

1 Introduction

Traditional mathematics curriculum faces several issues nowadays. As stated by Coffland and Xie [1], the gap between course materials and students' real-life mathematical experiences, the scattering of knowledge in different courses, and the lack of mathematics applications to other subjects all hinder the learning of students [2][3][4]. The emerging trends in data science, machine learning, and artificial intelligence also impel teachers to make corresponding changes in mathematics education [5][6]. Several experiments and improvements are thus carried out, including integrated curriculum, multidisciplinary programs [7], collaborative learning [8], blended learning [9], and capstone courses [10]. In order to better incubate students for future challenges, curriculum refinement and enrichment in mathematics should be performed [11].

Based on the aforementioned issues for mathematics education, literature reviews and course reviews were performed in the department of applied mathematics in one university to get more insight into the possible curriculum improvements [12]. For instance, programming languages taught in the computer programming course are C or Java for students in the department of applied mathematics. The course materials mainly focus on the syntax of programming languages. The concepts and language constructs are usually instructed separately. In that circumstance, students just memorize the programming languages and cannot accumulate computational thinking [13] [14] and problem-solving skills [15]. Students usually feel bored in the class and forget what they learn after the semester. Moreover, the connection between mathematics and computer programming is weak. Without curriculum refinement and enrichment, students cannot realize how to use programming languages and apply both computational thinking and problem-solving skills to their learning or research in mathematics. Besides, in the statistics course, students usually learn the theoretical knowledge and take practices in textbooks by topics. However, the scenarios described in textbooks are often limited in a certain scope or designed in a simpler form. A critical gap exists between theoretical understanding and the actual problem in the real world. Furthermore, students have no experience in the leverage of knowledge and skills from mathematics, statistics, and computer science, which are essential for research and industry nowadays.

According to the issues identified from previous studies and the course reviews, undergraduate mathematics curriculum is refined and enriched with computer science courses to achieve the following goals, including (1) The development of the capability to realize specific computation or problem solution by programming through computational thinking, (2) The development of the capability to solve real or complex problems thoroughly by mathematics, statistics, and computer science knowledge and skills [16][17], and (3) The accumulation of cross-domain capabilities, skills, and experiences for emerging challenges nowadays [18].

The curriculum is designed and implemented for students who major in applied mathematics to better stimulate their learning, participation, exercise, and innovation based on the knowledge of mathematics, statistics, and computer science. It aims to provide students with comprehensive theoretical and practical knowledge for the challenges and requirements nowadays [19]. Major courses selected and discussed in this study include “Computer Programming” in the freshman year, “Data Processing and Analysis” in the junior year, and “Statistical Software for Data Analysis” in the junior year. Evaluations, major findings, and lessons learned from the refined courses are also discussed in this study for more insight into the future deployment and refinement of the curriculum.

The remainder of this study is organized as follows. Section 2 introduces the design of the refined curriculum in the department of applied mathematics. Section 3 describes the evaluations of introduced courses from the perspectives of students. Section 4 depicts major findings and lessons learned from the courses. Finally, Section 5 presents the conclusion and future works of this study.

2 Curriculum design and implementation

The overview of computer science courses enriched and refined in the department of applied mathematics is depicted in Table 1. The courses in freshman year, including "Introduction to Computer Science" and "Computer Programming," are compulsory courses for students to acquire fundamental knowledge and skills in computer science. "Introduction to Computer Science" is the first course of computer science concentration in the department of applied mathematics. The goal of "Introduction to Computer Science" is to give the global view and understanding of fundamental computer science topics, including hardware, software, operating system, programing languages, network, web development, and data analytics. "Computer Programming" is the first programming course for students to learn and gain fundamental capabilities in computational thinking, problem-solving, and coding.

Table 1. Overview of computer science courses in the department of applied mathematics

Year	Course	Credit
Freshman Year	Introduction to Computer Science	3
	Computer Programming	3
Sophomore Year	Internet Practice	3
	Object-Oriented Programming	3
Junior/Senior Year	Data Structure	3
	Introduction to Mathematical Software	3
	Data Processing and Analysis	3
	Statistical Software for Data Analysis	3
	Capstone Course (I/II)	2

The courses in the sophomore year, including "Internet Practice" and "Object-Oriented Programming," are elective courses for students to learn front-end development and advanced programming knowledge and skills. "Internet Practice" provides students with modern web design and implementation skills (e.g., HTML, CSS, JavaScript, jQuery, and so on). The topic of data visualization for high-quality presentation of quantitative information will be emphasized during the course [20]. Students can integrate what they learn in statistics and "Internet Practice" to present and visualize data on websites. Based on the knowledge and skills learned from "Computer Programming," students can enroll for "Object-Oriented Programming" to learn advanced programming skills and gain valuable experience in the design and implementation of complex applications. The accumulated knowledge and programming skills are useful for the learning of advanced courses such as "Data Processing and Analysis" and "Statistical Software for Data Analysis." Finally, the courses in junior year and senior year, including "Data Structure," "Introduction to Mathematical Software," "Data Processing and Analysis," "Statistical Software for Data Analysis," and "Capstone Course," are elective courses for students to gain the knowledge, skills, and experience from mathematics, statistics, and computer science for real-world problems and complex applications. "Data Structure" provides students with advanced knowledge about the design of data structure, associated manipulations, and algorithms based on the knowledge learned from “Computer Programming” and “Object-Oriented Programming.” Topics such as tree structure and graph structure are discussed from the perspectives of “Data Structure” in computer science and “Graph Theory” in mathematics. Thus, students can get more insight into mathematical theories and corresponding real-world applications. “Introduction to Mathematical Software” introduces useful software tools and programming languages for students in their advanced learning and research tasks, including LaTeX, Mathematica, Matlab, Octave, Anaconda, SageMath, and so on. Both “Data Processing and Analysis” and “Statistical Software for Data Analysis” provide students with the knowledge, skills, and hands-on experience in each step of data processing and analysis process from the perspectives of mathematics, statistics, and computer science. Python, R programming language, and associated data science packages will be introduced for students. Finally, in the “Capstone Course,” students are encouraged to integrate what they learn in the past and conduct researches based on their interests [10]. The research topics include but not limit to financial mathematics, graph theory, statistics, data mining, mathematics education, and so on.

In the following subsections, education goals, syllabuses, and the corresponding course designs of “Computer Programming,” “Data Processing and Analysis,” and “Statistical Software for Data Analysis” are described.

2.1 The Design of the Computer Programming Course

The education goal of “Computer Programming” course helps students to develop the capabilities of computational thinking and problem solving with specific programming languages. Table 2 describes the syllabus of “Computer Programming” course designed and implemented in the department. The design of the “Computer Programming” course is divided into two major sections, including the introduction of Alice and the introduction of Python programming language.

Table 2. Syllabus of Computer Programming

Topic	Description
Alice Introduction	Introduction to Alice
Object, Methods, and Properties	Using variables and objects
Flow Control	Using if statements and loops
Event Control	Managing keyboard and mouse events
Game Design	Game design and implementation
Python Introduction	Introduction to Python
Variables and Data Types	Using variables and fundamental data types
Flow Control	Using if statements and loops
Data Structures	Using lists and dictionaries
Input and Output	Manipulating file and network I/O
Function	Functions design and implementation
Exception Handling	Exception handling and testing

Alice, developed by Carnegie Mellon University, is an educational software for teaching computer programming in a 3D environment [21]. Through the 3D environment and the interactive interface provided by Alice, students can learn and practice the concept of programming while creating their animations, stories, or games. Any implementation or changes to Alice programs can be executed and examined through the 3D environment instantly. Thus, students can practice and realize major programming concepts, including objects, methods, properties, flow control, and event control, visually and efficiently. During the learning of Alice, students are expected to do their assignments and group projects with particular programming concepts (e.g., operators and flow control) and export their works as videos and share it over the Internet.

In the second section of the "Computer Programming" course, Python will be introduced for students. In addition to general application development, Python is highly used in numerical computation, signal processing, statistics, symbolic mathematics, and algebra. Python also has lots of useful packages in data science, including machine learning, text mining, image processing, and so on, which are tightly related to the knowledge and application of mathematics. Therefore, Python is more suitable for students in the department of applied mathematics rather than programming languages like C and Java [22]. The materials of Python include variables, data types, flow control, data structures, input and output, functions, and exception handling. Students can acquire fundamental programming knowledge and skills for further programming courses and data science courses, such as "Object-Oriented Programming," "Data Processing and Analysis" and "Statistical Software for Data Analysis." It is expected that the capability of computational thinking, the skills of problem-solving, and the understanding and practice of programming concepts can be developed effectively and efficiently in the refined "Computer Programming" course.

2.2 The design of data processing and analysis course

The "Data Processing and Analysis" course helps students to learn and experience every important step of data processing and analysis tasks by the leverage of knowledge and skills in mathematics, statistics, and computer science. Table 3 describes the syllabus of "Data Processing and Analysis" course designed and implemented in the department of applied mathematics.

In the beginning of the course, the steps of data processing and analysis, including problem definition, data collection, data cleaning, data management, data modeling, data analysis, data visualization, and findings presentation, are introduced for students to develop a global picture and an overall understanding. Students can get more insight into the roles of mathematics, statistics, and computer science in data processing and analysis. Open data [23] and big data will also be introduced for students to understand real-world trends, challenges, and problems they will encounter nowadays. After the preliminary introduction, the course introduces each step of data processing and analysis. More specifically, students can learn and experience how to perform data processing and analysis through Python and related data science packages based on the knowledge and skills accumulated in mathematics, statistics, and computer science courses. Important Python packages, including NumPy, Pandas, SciPy, SymPy, and Matplotlib, are also introduced as fundamental tools for students on data science and scientific computation [24]. In addition, open data will be used as major materials for learning and practice throughout the course. Popular formats of open data, including CSV, JSON and XML, and the corresponding manipulations will be introduced for students to perform data collection, data cleaning, data management, data modeling, data analysis, and data visualization tasks [25]. During the course, students are encouraged to do their assignments and group projects based on open data and to innovate based on their interests and creativities.

Table 3. Syllabus of data processing and analysis

Topic	Description
Course Introduction	Introduction to data processing and analysis
Process of Data Processing and Analysis	Major steps in data processing and analysis
Open Data	Introduction to open data and formats (CSV, JSON, and XML)
Python Basis	Python review
Data Collection and Management	Using NumPy and Pandas packages for data collection and management
Data Modeling, Analysis, and Visualization	Using statistics library and Matplotlib package for data analysis and visualization
Network and Graph	Using NetworkX package for network and graph modeling
Text Mining	Using Beautiful Soup, lxml, NLTK, and jieba packages for text mining
Image Processing	Using OpenCV package for image processing and recognition
Machine Learning	Using machine learning packages for classification, clustering, and regression
Advanced Topics or Case Studies	Selected topics in machine learning in different domains

After students are equipped with solid knowledge and skills of data processing and analysis, advanced applications such as network and graph modeling, text mining, and image processing will be introduced. Emerging topics and corresponding frameworks for machine learning can also be introduced. For instance, students can learn and use frameworks like Scikit-Learn, Tensorflow, PyTorch, and Keras for particular machine learning problems based on their interests. It is expected that a thorough understanding of data processing and analysis, the cross-domain capability based on mathematics, statistics, and computer science, and the programming skills for complex data processing and analysis tasks can be developed in the refined “Data Processing and Analysis” course.

2.3 The design of statistical software for data analysis course

The goal of “Statistical Software for Data Analysis” course helps students to learn and experience every important step of data processing and analysis using R programming language. R programming language provides efficient facilities for data manipulation, calculation, analysis, and visualization [26]. Both R programming language and Python are popular tools for data processing and analysis tasks. Therefore, it is worthwhile for students in the department of applied mathematics to learn and practice those skills for various data processing and analysis requirements. Based on previous knowledge of mathematics, statistics, and computer science, students can acquire additional capabilities in this course when performing data processing and analysis tasks in the future.

Table 4 depicts the syllabus of the "Statistical Software for Data Analysis" course designed and implemented in the department of applied mathematics. Similar to the "Data Processing and Analysis" course, the steps of data processing and analysis are introduced at the beginning for students to develop the overall understanding. Open data will also be introduced and used as course materials for students to learn and experience the data collection, cleaning, management, modeling, analysis, and visualization tasks by R programming language. Students are encouraged to do their assignments and group projects based on open data and to innovate based on their interests and creativities. Exploratory data analysis, descriptive statistics, predictive statistics, and selected topics in machine learning, including clustering, classification, and regression, are introduced for students. Students can also get more insight into the applications of data processing and analysis through case studies on financial mining, text mining, and social network mining [27]. It is expected that the thorough understanding of data processing and analysis, the cross-domain capability based on mathematics, statistics, and computer science, and the skills of R programming for data processing and analysis tasks can be developed in the refined "Statistical Software for Data Analysis" course.

Table 4. Syllabus of statistical software for data analysis

Topic	Description
R Introduction	Introduction to R programming language
Process of Data Processing and Analysis	Major steps in data processing and analysis
R Basis	Variables, vectors, and fundamental operations
Data Structures	Using matrix, array, list, and data frame
Open Data	Introduction to open data and formats (CSV, JSON, and XML)
Data Collection and Management	Data management and manipulation in R
Flow Control	Using if statements, switch, and loops
Data Visualization	Using R for data visualization
Statistics	Using R for descriptive and predictive statistics
Machine Learning	Using R for classification, clustering, and regression
Advanced Topics or Case Studies	Financial and social network mining by R

3 Course evaluation

At the end of every semester in the university, a course questionnaire is provided to students to carry out the evaluations from the perspectives of the course and the teacher. The evaluation statements are stated as the following.

1. The course was delivered based on the syllabus (C1)
2. The course was stimulating and interesting (C2)
3. The course objectives were achieved (C3)
4. Course materials were clear, appropriate and helpful (C4)
5. Assessment methods of the course were appropriate (C5)
6. The teacher was enthusiastic about teaching (T1)
7. The teacher presented in a clear and organized manner (T2)
8. The teacher presented appropriately based on the stated level of the class (T3)
9. The teacher was helpful when student had difficulties or questions (T4)
10. The teacher provided useful feedback and guidance (T5)

The level of agreement to the evaluation statements includes "Strongly Disagree," "Disagree," "Neutral," "Agree," and "Strongly Agree." Thus, score 1 to score 5 are given to the above levels correspondingly. Table 5, Table 6, and Table 7 present the evaluation results of "Computer Programming," "Data Processing and Analysis," and "Statistical Software for Data Analysis" based on refined curricula by the students in the department of applied mathematics, respectively.

Table 5. Course evaluation for Computer Programming

Statements	**Level of Agreement**				
	Strongly Disagree	***Disagree***	***Neutral***	***Agree***	***Strongly Agree***
C1	0	0	1	5	14
C2	0	0	1	10	9
C3	0	0	0	5	15
C4	0	0	1	5	14
C5	0	0	1	7	12
T1	0	0	0	5	15
T2	0	0	1	7	12
T3	0	0	1	6	13
T4	0	0	1	5	14
T5	0	0	0	6	14
Average Course Score					4.63
Average Score of Courses Offered by the Department					4.37
Average Score of Courses Offered by the College					4.34
Average Score of Courses Offered by the University					4.34

It was found that all the courses got better scores (4.63, 4.49, and 4.61) than average scores of courses offered by the department of applied mathematics (4.37, 4.37, and 4.38), the college of science and engineering (4.34, 4.34, and 4.34), and the university (4.34, 4.34, and 4.29). It shows that significant improvements are achieved through the curriculum refinement and enrichment. From the perspective of course materials (C4), the average scores of "Computer Programming," "Data Processing and Analysis," and "Statistical Software for Data Analysis" are 4.65, 4.46, and 4.74, respectively. It is confirmed that the corresponding course materials based on the refined curricula are clear, appropriate, and helpful for students in their learning tasks.

Table 6. Course evaluation for Data Processing and Analysis

Statements	**Level of Agreement**				
	Strongly Disagree	***Disagree***	***Neutral***	***Agree***	***Strongly Agree***
C1	0	0	1	14	9
C2	0	0	3	12	9
C3	0	0	0	12	12
C4	0	0	1	11	12
C5	0	0	0	12	12
T1	0	0	1	6	17
T2	0	0	0	9	15
T3	0	0	0	12	12
T4	0	0	0	9	15
T5	0	0	1	12	11
Average Course Score					4.49
Average Score of Courses Offered by the Department					4.37
Average Score of Courses Offered by the College					4.34
Average Score of Courses Offered by the University					4.34

Table 7. Course evaluation for Statistical Software for Data Analysis

Statements	**Level of Agreement**				
	Strongly Disagree	***Disagree***	***Neutral***	***Agree***	***Strongly Agree***
C1	0	0	2	4	13
C2	0	0	2	4	13
C3	0	0	0	10	9
C4	0	0	0	5	14
C5	0	0	1	4	14
T1	0	0	1	4	14
T2	0	0	0	7	12
T3	0	0	1	8	19
T4	0	0	1	4	14
T5	0	0	1	7	11
Average Course Score					4.61
Average Score of Courses Offered by the Department					4.38
Average Score of Courses Offered by the College					4.34
Average Score of Courses Offered by the University					4.29

From the perspective of course contents (C2), the average scores of "Computer Programming," "Data Processing and Analysis," and "Statistical Software for Data Analysis" are 4.4, 4.25, and 4.58, respectively. In addition, 95%, 88%, 89% of students in three courses agree that the courses were stimulating and interesting. Based on the statistics, it is believed that students can be inspired to learn, participate, exercise, and innovate by the refined and enriched curricula efficiently. Finally, from the perspective of course objectives (C3), all the students in three courses agree that the course objectives were achieved. Thus, according to course evaluations, the deployment of computer science courses based on the refined and enriched curricula brings lots of benefits and positive impacts on the students in the department of applied mathematics.

4 Major findings

In this section, major findings and lessons learned from the course implementations based on the refined and enriched curriculum are introduced. Several findings and experiences during the courses, including "The Use of Educational Software for Programming," "The Connection between Course Content and Real Life," and "Interdisciplinary Learning," are introduced and discussed. These findings, experiences, and lessons learned provide more insight into further deployment and refinement of the curriculum.

4.1 The use of education software for programming

One of the recommendations for current mathematics education is "targeting computational thinking and coding [28]." However, previous programming courses in the department of applied mathematics mainly focused on the syntax of programming languages. The concepts and language constructs are usually instructed separately. In that circumstance, students just memorized the programming languages and cannot accumulate computational thinking and problem-solving skills. Tough and boring course materials could make students lose interest and give up too early [29]. What is even worse is that many students hesitate to study the following computer science courses.

Therefore, when designing the "Computer Programming" course, Alice is used in the first section of the course. For students in the department of applied mathematics (or other non-computer science departments), Alice (or other educational software for programming such as Scratch and MIT App Inventor) could be a better choice to inspire the interests of students and teach programming concepts than programming languages like C and Java [30] [31]. Through the 3D environment and the interactive interface provided by Alice, students can learn and practice different programming concepts while creating their animations, stories, and games. Based on the visual presentation of program execution, students can get insight into the effects of programming concepts and language constructs (e.g., if statements and loops) effectively and efficiently. In addition, students show more interests and perform more proactively by using Alice for learning [32]. Innovation and creativity can also be incubated during the class, assignments, and group projects in the course.

Figure 1 depicts the racing game and the puzzle game designed and implemented by Alice in the midterm project of the "Computer Programming" course. In the racing game, the player can use keyboard to control the vehicle to pass checkpoints and avoid to hit obstacles. A countdown mechanism and a scoring system are implemented through the corresponding programming concepts and language constructs. In the puzzle game, the player can use keyboard to control the rabbit to explore the whole area and use mouse to interact with objects in the game (e.g., open a refrigerator) to find all the hidden carrots.

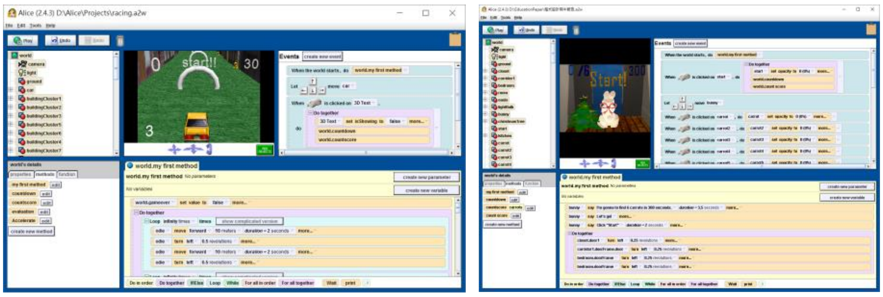

Fig. 1. Games designed and implemented by Alice

By reviewing assignments and group projects by students, it is confirmed that important programming concepts, including objects, methods, variables, loops, flow control, and event controls, are exercised and learned effectively and efficiently. In addition, students show great enthusiasm for developing their own animations, stories, and games. Based on accumulated programming concepts and development experience through Alice, students have better performance when learning other programming languages (e.g., Python and Java) and are not afraid of coding for particular tasks in the future [33]. Therefore, for students in the department of applied mathematics (or other non-CS major students), educational software for programming is a better choice to stimulate and inspire the learning, participation, and exercise in the beginning [34]. This also facilitates the development of the capability to realize specific computation or problem solution by programming through computational thinking.

4.2 The Connection between Course Content and Real Life

One of the issues about traditional mathematics curricula is the gap between course materials and students' "real-life" mathematical experiences [1][4]. For instance, based on course reviews, students enrolled in the statistics course usually learned the theoretical knowledge and did limited or simplified practices in textbooks by topics separately. A critical gap exists between theoretical understanding and actual problem in real life. Students have no experience in the leverage of knowledge and skills learned from mathematics, statistics, and computer science, which are essential for research and industry today.

Therefore, both in “Data Processing and Analysis” and “Statistical Software for Data Analysis” courses, open data is used as materials for learning and practice [35] in addition to limited or simplified examples in textbooks. Through the introduction of open data in the courses, students can find and use interesting open data for further analysis, application development, and even innovation. Based on statistical theories and programming skills, students can learn and practice how to explore, collect, clean, manage, model, analyze, and visualize actual data from open data portal. The analysis results usually reveal interesting findings which are beneficial to the general public and the society. Thus, students can not only realize the data processing and analysis methods but also get more insight into the applications of statistical theories and programming skills in the real world.

Figure 2 present the assignments implemented by Python and R based on government open data. The first one is to process, analyze, and visualize the dengue fever cases reported in major cities in Taiwan by Python, NumPy, Pandas, and Folium in “Data Processing and Analysis” course. In the implementation, users can select a period of time and observe the associated dengue fever cases through the map service. The information of descriptive and predictive analytics of dengue fever cases is also provided. The second one is to collect, process, analyze, and present travel information in Tainan city in Taiwan by R, ggplot2, ggmap, and googleVis in “Statistical Software for Data Analysis” course. Students collect and integrate several travel-related open data from government open data portal, including accommodation, attractions, and Wi-Fi hotspots, and develop a travel application for travelers. As shown in Figure 2, the location of Wi-Fi hotspots in Tainan city is presented for travelers. Travelers can use the application to plan, navigate, and acquire corresponding travel resources efficiently.

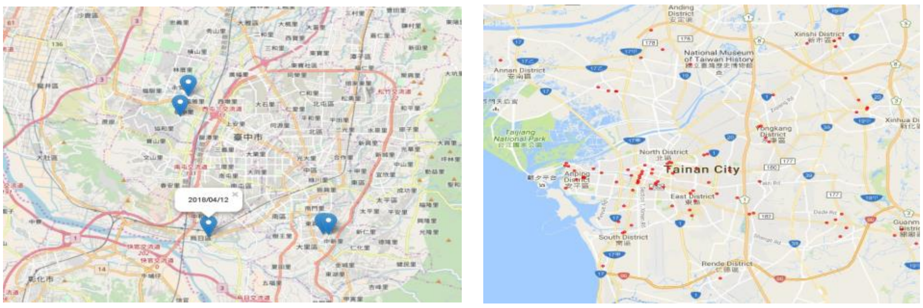

Fig. 2. Data visualization by Python and R

Through the connection between course content and real-life application based on open data, students can realize the global picture of data processing and analysis, and practice major steps in data processing and analysis comprehensively rather than limited or simplified examples in textbooks. In addition, complexities and problems during the processing and analysis tasks, including data cleaning, data munging, encoding problems, and so on, can help students to develop solid knowledge and skills which are useful for future research and career. It is discovered that students show great enthusiasm for learning and practice due to the use of open data. Students will search and choose open data they are interested in. Innovative applications of data processing and analysis can be found in the assignments and group projects during the courses. According to these positive results, the connection between course content and real-life is successfully established for students. This achieves the development of the capability to solve real or complex problems thoroughly by mathematics, statistics, and computer science knowledge and skills. In the future, more connections can be further identified, designed, and deployed to all the mathematics, statistics, and computer science courses for students.

4.3 Interdisciplinary Learning

Another issue about traditional mathematics curricula is the lack of "the connection between mathematics to other subjects as an interdisciplinary approach [3][36]." In traditional mathematics courses, students usually learn an abundance of mathematical theories and calculations. However, the applications of those theories in different domains are less mentioned. For instance, the graph theory can be used to build recommendation engines in e-commerce or social networks [37]. In addition, the knowledge of matrix and associated manipulations can be applied to image presentation and processing. In order to facilitate the learning, practice, and knowledge accumulation in an interdisciplinary approach, lots of assignments and group projects based on mathematics, statistics, and computer science are designed. Students are encouraged to integrate what they learn in different courses and to innovate based on their interests and creativities.

Figure 3 presents the projects built by students during the "Data Processing and Analysis" course. In the first project, students integrated what they learned in front-end development, computer programming, data processing and analysis, statistics, and data visualization methods to build a responsive web page for their travel in Japan. The web page presents the route, the photos, the expense statistics, and the activities in timeline of their travel. The project also got an award in the competition held by college of science and engineering in the university. Based on the feedback from one student, she said it was a wonderful experience to integrate what she learned in the past and develop a great website from scratch. Through the course and the project, she had more confidence and was not afraid of computer-related tasks in the future. In the second project, a student who also enrolled in "Graph Theory" course tried to model and implement a graph coloring algorithm to present the coloring result. Although the student encountered many difficulties, he solved those problems and completed the implementation by searching, learning, and trying numerous online resources and code examples. Based on the feedback from those students, they gained interdisciplinary knowledge, hands-on experience, and problem-solving skills through the projects or assignments. They also got great pleasure and confidence when they completed the tasks. The interdisciplinary learning achieved by the refined and enriched curriculum makes students learn, practice, and accumulate various knowledge and skills in different disciplines efficiently [38] [39]. Besides, confidence, innovation, and creativities of students can be incubated during the learning activities.

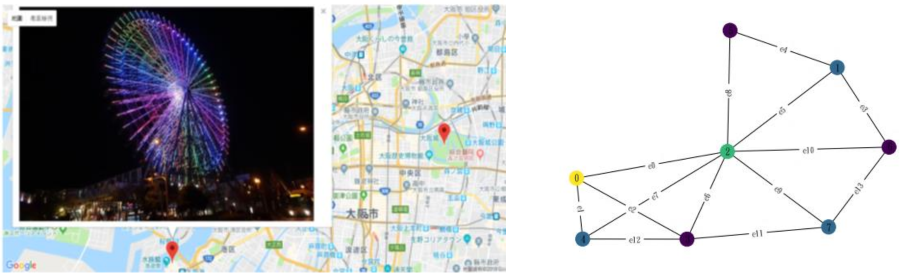

Fig. 3. Projects based on interdisciplinary knowledge and skills

5 Conclusion

Many universities begin to propose several innovations and refinements in education, including teaching improvement, integrated curriculum, multidisciplinary programs, and capstone courses to better incubate students for future. In this study, the experience of enriching undergraduate mathematics curriculum with computer science courses is introduced. Based on the original focus on mathematical and computation theories, the employment of computer science courses provides students with comprehensive theoretical and practical knowledge for the challenges and industrial requirements nowadays. It is anticipated that students can (1) realize specific computation or problem solution by programming through computational thinking, (2) solve real or complex problems thoroughly by mathematics, statistics, and computer science knowledge and skills, and (3) accumulate cross-domain capabilities, skills, and experiences for emerging challenges. The design and implementation of three computer science courses, including “Computer Programming,” “Data Processing and Analysis,” and “Statistical Software for Data Analysis,” are introduced. Based on the course evaluations from students’ perspectives, the deployment of computer science courses and the refined and enriched curriculum bring lots of benefits and positive impacts on the students in the department of applied mathematics. Major findings, course experiences, and lesson learned from the course implementation, including “the use of educational software,” “the connection between course content and real life,” and “interdisciplinary learning,” are found to be highly beneficial to the learning activities. Computational thinking, interdisciplinary knowledge, problem-solving skills, confidence, and creativities of students can be incubated during the courses effectively and efficiently. According to current implementation results, future works include further evaluation and curriculum refinement. More information about the design and implementation of computer science courses can be accumulated and analyzed as time goes on. In addition to the questionnaire designed by the university, several IT systems can be built and leveraged to collect and analyze the learning status of students. Thus, more qualitative and quantitative analysis can be performed to further evaluate and refine the curriculum in the future. Finally, the feasibility of deploying computer science courses to the undergraduate program in other disciplines can also be explored. Through the findings and lessons learnt, a more improved curriculum can be developed to incubate students for a better future.

6 Acknowledgment

This research was funded by the Ministry of Science and Technology of the Republic of China under grant MOST 108-2221-E-143-003-MY3.

7 References

[1] D. A. Coffland and Y. Xie, *The 21st Century Mathematics Curriculum: A Technology Enhanced Experience*. Cham: Springer International Publishing, 2015, pp. 311–329. https://doi.org/10.1007/978-3-319-02573-5_17

[2] G. Wiggins and J. McTighe, *Understanding By Design*. Association for Supervision and Curriculum Development, 2005.

[3] Z. Usiskin, *The Current State of the School Mathematics Curriculum*. National Council of Teachers of Mathematics, 2010, pp. 25–39.

[4] J. Boaler, *What's Math Got to Do with It?: How Parents and Teachers Can Help Children Learn to Love Their Least Favorite Subject*. Penguin Books, 2009.

[5] Y. Kim, H.-E. Chu, and G. Lim, *Science Curriculum Changes and STEM Education in East Asia*. Cham: Springer International Publishing, 2015, pp. 149–226. https://doi.org/10.1007/978-3-319-16390-1_7

[6] A. A. Zaher and I. W. Damaj, Extending STEM Education to Engineering Programs at the Undergraduate College Level, *International Journal of Engineering Pedagogy*, vol. 8, no.3, pp. 4-16, 2018. https://doi.org/10.3991/ijep.v8i3.8402

[7] J. Williams, W.-M. Roth, D. Swanson, B. Doig, S. Groves, M. Omuvwie, R. B. Ferri, and N. Mousoulides, *Interdisciplinary Mathematics Education: A State of the Art*. Springer International Publishing, 2016. https://doi.org/10.1007/978-3-319-42267-1

[8] B. L. Smith and J. T. MacGregor, "What is collaborative learning?" in *Collaborative Learning: A Sourcebook for Higher Education*. National Center on Postsecondary Teaching, Learning, and Assessment, Pennsylvania State University, USA, 1992.

[9] O. Imas, V. Kaminskaya and A. Sherstneva, "Teaching math through blended learning," in *2015 International Conference on Interactive Collaborative Learning (ICL)*, 2015, pp. 511-514. https://doi.org/10.1109/icl.2015.7318081

[10] J. Wilbarger and S. Howe, "Current practices in engineering capstone education: Further results from a 2005 nationwide survey," in *Proceedings. Frontiers in Education. 36th Annual Conference*, 2006, pp. 5–10. https://doi.org/10.1109/fie.2006.322616

[11] J. C. Adams and R. J. Pruim, "Computing for stem majors: Enhancing non cs majors' computing skills," in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 457–462. https://doi.org/10.1145/2157136.2157270

[12] J. vom Brocke, A. Simons, B. Niehaves, B. Niehaves, K. Riemer, R. Plattfaut, and A. Cleven, "Reconstructing the Giant: On the Importance of Rigour in Documenting the Lit-

erature Search Process," in *17th European Conference on Information Systems (ECIS)*, 2009, pp. 2206-2217. https://doi.org/10.17705/1cais.03709

[13] P. B. Henderson, W. Barker, S. Epp, and W. Marion, "Math educators, computer science educators: Working together," *SIGCSE Bull.*, vol. 35, no. 1, pp. 236–237, Jan. 2003. https://doi.org/10.1145/792548.611906

[14] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006.

[15] L. D. Miller, L. Soh, V. Chiriacescu, E. Ingraham, D. F. Shell, S. Ramsay, and M. P. Hazley, "Improving learning of computational thinking using creative thinking exercises in cs-1 computer science courses," in *2013 IEEE Frontiers in Education Conference (FIE), 2013*, pp. 1426–1432. https://doi.org/10.1109/fie.2013.6685067

[16] L. D. English and J. Gainsburg, *Problem Solving in a 21st-Century Mathematics Curriculum*. Taylor and Francis, 2016, pp. 313–335.

[17] G. Stillman, J. Brown, P. Galbraith, and K. E. D. Ng, *Research into Mathematical Applications and Modelling*. Singapore: Springer Singapore, 2016, pp. 281–304.

[18] W.-M. Roth, *Interdisciplinary Approaches in Mathematics Education*. Dordrecht: Springer Netherlands, 2014, pp. 317–320.

[19] J. Eckroth, "Teaching future big data analysts: Curriculum and experience report," in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2017, pp. 346–351. https://doi.org/10.1109/ipdpsw.2017.122

[20] G. Xiaohui and Z. Zhimin, "Construction of curriculum system on webpage design and website development in higher vocational colleges," in *2011 6th International Conference on Computer Science Education (ICCSE)*, 2011, pp. 657–659. https://doi.org/10.1109/iccse.2011.6028724

[21] S. Cooper, W. Dann, and R. Pausch, "Alice: A 3-d tool for introductory programming concepts," *J. Comput. Sci. Coll.*, vol. 15, no. 5, pp. 107–116, Apr. 2000.

[22] C. Sotomayor-Beltran, G. W. Z. Segura, and A. Roman-Gonzalez, "Why should python be a compulsory introductory programming course in lima (peru) universities?" in *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, 2018, pp. 1–4. https://doi.org/10.1109/ica-acca.2018.8609808

[23] O. Maksimenkova and V. Podbelskiy, "On practice of using open data in construction of training and assessment tasks for programming courses," in *2015 10th International Conference on Computer Science Education (ICCSE)*, 2015, pp. 233–236. https://doi.org/10.1109/iccse.2015.7250248

[24] J. Bernard, "Use python for scientific computing," *Linux Journal*, 2008.

[25] Y. Zhu, "Introducing google chart tools and google maps api in data visualization courses," *IEEE Computer Graphics and Applications*, vol. 32, no. 6, pp. 6–9, 2012. https://doi.org/10.1109/mcg.2012.114

[26] S. Datta and V. Nagabandi, "Integrating data science and r programming at an early stage," in *2017 IEEE 4th International Conference on Soft Computing Machine Intelligence (ISCMI)*, 2017, pp. 1–5. https://doi.org/10.1109/iscmi.2017.8279587

[27] M. Prakash, G. Padmapriy, and M. V. Kumar, "A review on machine learning big data using r," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 1873–1877. https://doi.org/10.1109/icicct.2018.8473342

[28] L. D. English, *Advancing Mathematics Education Research Within a STEM Environment*. Singapore: Springer Singapore, 2016, pp. 353–371.

[29] W. C. Chu, S. Hung, C. Chang, and S. I. Ahamed, "Applying spocs for programming course to improve study quality," in *2015 Second International Conference on Trustworthy Systems and Their Applications*, 2015, pp. 128–134. https://doi.org/10.1109/tsa.2015.28

[30] A. Rusu, "Introducing gaming tools for computing education in stem related curricula," in *2015 IEEE Frontiers in Education Conference (FIE)*, 2015, pp. 1–8. https://doi.org/10.1109/fie.2015.7344166

[31] O. Mironova, I. Amitan, J. Vendelin, J. Vilipõld, and M. Saar, "Object-Oriented Programming for non-IT Students: Starting from Scratch," *International Journal of Engineering Pedagogy*, vol. 5, no.4, pp. 22-28, 2015. https://doi.org/10.3991/ijep.v5i4.4734

[32] J. Preston and B. Morrison, *Entertaining Education – Using Games-Based and Service-Oriented Learning to Improve STEM Education*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 70–81. https://doi.org/10.1007/978-3-642-11245-4_7

[33] Z. Jiang, E. B. Fernandez, and L. Cheng, "P2n: A pedagogical pattern for teaching computer programming to non-cs majors," in *Proceedings of the 18th Conference on Pattern Languages of Programs*. New York, NY, USA: Association for Computing Machinery, 2011. https://doi.org/10.1145/2578903.2579163

[34] M. Hacker and J. Kiggens, *Gaming to Learn: A Promising Approach Using Educational Games to Stimulate STEM Learning*. Sense Publishers, 2011, pp. 257–279. https://doi.org/10.1007/978-94-6091-549-9_14

[35] D. Mouromtsev and M. d'Aquin, *Open Data for Education: Linked, Shared, and Reusable Data for Teaching and Learning*. Springer International Publishing, 2016.

[36] K. Weber, M. Inglis, and J. P. Mejia-Ramos, "How mathematicians obtain conviction: Implications for mathematics instruction and research on epistemic cognition," *Educational Psychologist*, vol. 49, no. 1, pp. 36–58, 2014. https://doi.org/10.1080/00461520.2013.865527

[37] A. Majeed and I. Rauf, "Graph theory: A comprehensive survey about graph theory applications in computer science and social networks," *Inventions*, vol. 5, no. 1, 2020. https://doi.org/10.3390/inventions5010010

[38] P. Charlton and K. Avramides, "Knowledge construction in computer science and engineering when learning through making," *IEEE Transactions on Learning Technologies*, vol. 9, no. 4, pp. 379–390, 2016. https://doi.org/10.1109/tlt.2016.2627567

[39] A. Zeidmane and S. Cernajeva, "Interdisciplinary Approach in Engineering Education," *International Journal of Engineering Pedagogy*, vol. 1, no.1, pp. 36-41, 2011. https://doi.org/10.3991/ijep.v1i1.1604

8 Author

Chia Hung Kao received his Ph.D. degree in the Department of Electrical Engineering, National Cheng Kung University, Taiwan, in 2009. After getting his degree, he joined Institute for Information Industry (III) as a senior engineer and participated in the development of several cloud computing and big data analytics products. Since 2015, he joined National Taitung University as an assistant professor in the department of applied mathematics. His research interests include software engineering, cloud computing, big data analytics, and education.

Article submitted 2021-02-03. Resubmitted 2021-07-20. Final acceptance 2021-07-24. Final version published as submitted by the author.