Nyamapfene, Abel, Lynch, Stephen, Burova, Iva and Andrade, Matheus Oliveira De (2022) MATLAB and Python Open Book Assessments: lessons from two UK institutions. In: 2022 IEEE Global Engineering Education Conference (EDUCON), 28 March 2022 - 31 March 2022, Tunis, Tunisia.

**DOI:** https://doi.org/10.1109/educon52537.2022.9766577

**Publisher:** IEEE

**Version:** Accepted Version

IEEE EDUCON 2022, Tunisia: Preprint

# MATLAB and Python Open Book Assessments: Lessons from Two UK Institutions

Abel Nyamapfene
Centre for Engineering Education
University College London
London, UK
ORCID: https://orcid.org/0000-0001-8976-6202

Iva Burova
Faculty of Engineering Science
University College London
London, UK
Email: iva.burova@ucl.ac.uk

Stephen Lynch
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
Email: s.lynch@mmu.ac.uk

ORCID: https://orcid.org/0000-0002-4183-5122

Matheus Oliveira De Andrade
Faculty of Engineering Science
University College London
London, UK
Email: m.deandrade@ucl.ac.uk

*Abstract*—**Although programming has increasingly become an integral part of Mathematics education in universities, the closed book exam, which excludes programming, remains the preferred main assessment component. However, the switch towards open book online assessments necessitated by the Covid pandemic has seen programming content being included in end of year assessments. In this paper we look at how programming has been integrated into the end of year open book online assessments in Mathematics at two UK universities. Preliminary conclusions suggest that open book online assessments incorporating programming content can be a viable replacement for the traditional closed book exam.**

***Keywords***— ***programming, MATLAB, Python, open book online assessment, closed book exam, mathematics assessment***

## I. INTRODUCTION

Powerful mathematics packages such as Maple™, Mathematica® and MATLAB® have been used in mathematics and engineering departments around the world for over 30 years. More recently, the open-source programming language Python has entered the arena. All these programming languages are used extensively by researchers and in recent years, they are being used more and more in teaching. In February 2018, the UK Government published the Blackett Review[1], highlighting the importance of computational modelling, programming, and simulation in both the public and private sectors. A few months later, Philip Bond published an independent review of knowledge exchange in the mathematical sciences[2], and amongst his many findings, one of his recommendations was that – "All mathematics students should acquire a knowledge of at least one programming language."

Computational modelling and simulation has proved useful in a variety of scientific disciplines, and a number of computational packages have been developed to support this growing need. For instance, Maplesoft have developed MapleSim™, MathWorks have developed the toolboxes Simulink® and Simscape®, whilst Wolfram have created WolframSystemModeller®. To add to these packages, a simulation package for Python, called PySim, has recently been released. This package is still in the developmental stages and is not as well used as the other simulation packages listed above.

With respect to programming languages, Python is currently the most popular, and it is being used more and more in both secondary and tertiary education[3, 4]. Pedagogies that integrate programming into mathematics education have been developing rapidly. In 2016, Nyamapfene and Lynch [5] presented two case studies, one from University College London (UCL), and the other one from Manchester Metropolitan University (MMU), showcasing the integration of MATLAB® into undergraduate mathematics and engineering mathematics teaching. A follow-up paper in 2020 by Lynch[6] shows how programming, computational modelling and simulation have become the core of the mathematics curriculum at Manchester Metropolitan University (MMU).

The objective of this paper is to identify and share good practice in the effective implementation of open-book examinations using MATLAB® and Python. We present a review of how two universities in the UK, namely MMU and UCL have gone about this task. We will discuss the nature of the exam questions and how they differ from previous years, if at all, the benefits and pitfalls of take-home examinations, the staff and student experiences, and future plans.

## II. OPEN BOOK ASSESSMENTS AT MMU

### *A. Objective*

Coursework and examinations have formed the methods of assessment at MMU for many years at all levels of the mathematics courses. The coursework components have not changed much because of the pandemic; however, the examinations posed a real problem. At MMU, a teaching block system, whereby students study one unit (module) in a six-week block, has been in place since 2019. Four such blocks span an academic year.

On the mathematics degree programmes, coursework is released after the first week of teaching and the deadline date is set in the sixth week of that block. The seventh week, after each six-week block, is used for the final unit assessment, namely, the take-home examinations. Typically, students are given 27-30 hours to complete the examination which must be submitted as a single pdf file.

A compromise was reached on the setting of the examination papers. Coursework usually involves a high level of programming and mathematical/statistical/OR modelling.

The students were given four questions and five weeks to complete the assignment. Before the pandemic, the in-laboratory examinations consisted of eight shorter questions where the students were required to answer five out of eight questions using MATLAB/Python as a graphing calculator. Some of the examination questions were based on theory and required proof, which had been covered in the course material. Any programming required would only consist of a few lines of code. The main objective was to design an examination paper that retained the feel of a normal examination paper but considered the fact that it was now an open-book format.

## *B. Context of Implementation*

The redesign of the examination papers to take-home versions is an ongoing process which had to be implemented at the eleventh hour. The first casualties in the new implementation were the theoretical questions involving bookwork. The open book format restricted the type of questions one could ask – we did not want students simply looking up proofs from the notes and copying them into their answer sheets.

A decision was made to set four questions, the first half of each question would be like past paper questions and the second half would require a bit of MATLAB/Python code with an interpretation of the results. Section C below lists a typical coursework question, a typical pre-Covid examination question and a post-Covid take-home examination question.

## *C. Question Design*

The first example is taken from a piece of coursework for a final year unit on *Dynamical Systems and Chaos,* in fact, all questions will be taken from this unit to make legitimate comparison. There are typically four questions on the coursework and each question is worth 25 marks. During the pandemic, students were given 5 weeks to complete the coursework. Feedback for the coursework is given after the open book examination. Pre-Covid, the students would have had 4 weeks to complete the coursework and they would have received feedback before the summer examination.

### Typical Coursework Question

Fig. 1 depicts a simple model of a three-neuron module and is described by the difference equations:

$$\begin{aligned} x_{n+1} &= b_1 + w_{12}\sigma(y_n) \\ y_{n+1} &= b_2 + w_{21}\sigma(x_n) + w_{23}\sigma(y_n) \\ z_{n+1} &= b_3 + w_{32}\sigma(y_n), \end{aligned}$$

where $x_n, y_n, z_n$ are the activation levels of neurons x, y and z, the activation function is given by:

$$\sigma(x) = \frac{1}{1+e^{-x}},$$

$b_1, b_2, b_3$ are biases and $w_{ij}$ represent weights of synaptic connections.

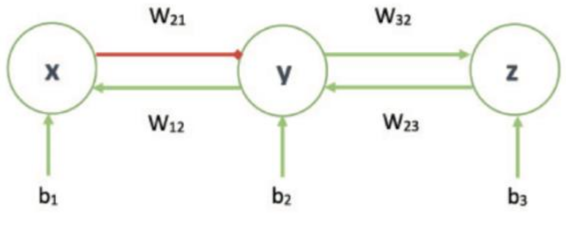

Fig. 1. A three-neuron module, where $w_{21}$ is an inhibitory synaptic connection and $w_{12}, w_{23}$ and $w_{32}$ are all excitatory.

(a) Given that $b_1 = -3, b_2 = 4, b_3 = -7$, $w_{12} = w_{32} = 8$ and $w_{23} = 6$, use MATLAB or Python to determine the number, location, and stability of the fixed points of period one when:

(i) $w_{21} = -3$; (ii) $w_{21} = -5$. [12 Marks]

(b) Using the same parameter values listed in part (a): write a MATLAB or Python program to plot a bifurcation diagram (with feedback) when the parameter $w_{21}$ is increased from $w_{21} = -12$, up to $w_{21} = -3$, and then ramped back down again to $w_{21} = -12$. [10 Marks]

(c) Give a physical interpretation of the results. [3 Marks]

The coursework question focussed on mathematical modelling and analysis using either MATLAB or Python. The questions are specifically designed to avoid bookwork, i.e. questions focussing primarily on recalling factual knowledge and the use of routine procedures as covered in textbooks and/or lectures [7] . To do well in these types of questions, students have to demonstrate an understanding of the underlying mathematics concepts, an ability to formulate solutions to the problem, programming ability to realise the solution using an appropriate programming language, and, finally, an ability to interpret the model outputs and apply them to the physical world.

This particular question demonstrates the modelling of a simple neuronal system with three neurons. Readers may be interested in the bifurcation diagram obtained for this system which has been included in Fig. 2.

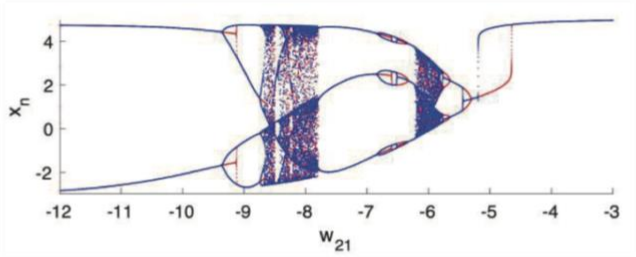

Fig. 2. Bifurcation diagram of a three-neuron module for the neuron x. There are period doublings and undoublings through chaos and a hysteresis cycle for $-5.2 < w_{21} < -4.6$, approximately.

### Pre-Covid Examination-Type Question

The second example is taken from a pre-Covid examination paper. There are typically eight questions, each worth 20 marks, on the paper and students would be required to answer five questions. Marks are awarded for the best five answers. Part (a) is a three-dimensional system of ordinary differential equations population model and is an unseen question. Part (b) is bookwork.

(a) An interacting species model of Balsan fir tree, moose and wolf at the Isle Royale National Park, USA, is given by:

$$\begin{aligned} \frac{db}{dt} &= b(1-b) - bm \\ \frac{dm}{dt} &= m(1-m) + bm - mw \\ \frac{dw}{dt} &= w(1-w) + mw, \end{aligned}$$

where $b(t)$ represents the population of Balsan fir trees, $m(t)$ is the population of moose, and $w(t)$ gives the population of wolves at time *t*. Determine the number and location of all critical points and show that there is a stable critical point for $b(t), m(t), w(t) > 0$. [10 Marks]

(b) The propagation of light through a simple fibre ring resonator can be modelled using the complex Ikeda map:

$$E_{n+1} = A + BE_n e^{i|E_n|^2},$$

where $E_n$ is the electric field strength in the ring on the n'th circulation, $A$ is related to input power and B represents a fibre coupling ratio. Prove that the circle of radius:

$$\frac{|AB|}{1-B}$$

is invariant for this mapping. [10 Marks]

The exam question is designed both to assess mathematical modelling and analysis using either MATLAB or Python, and to assess bookwork. Whilst students still need to demonstrate mastery of mathematical modelling and analysis concepts, the questions are not as demanding, in terms of time and complexity, as the coursework question which is assessed over a much longer period. The objective of the exam was to assess both bookwork and mathematical modelling and analysis over the entire breadth of the course using a range of questions. This contrasts with the coursework which is designed to ensure that student engages with a particular topic to a significant depth over a longer period.

### Post-Covid Examination-Type Question

The final example is taken from a post-Covid take home examination paper. The students are given four compulsory questions, each worth 25 marks. The example below would form one question. Part (a) is the same type of question used during pre-Covid, but part (b) requires a bit of extra work with MATLAB and/or Python.

(a) A planar system satisfies the following differential equations:

$$\frac{dr}{dt} = (r-1)^2$$
$$\frac{d\theta}{dt} = -1.$$

Given that, $r(0) = 2, \theta(0) = 0$, determine the Poincaré return map, $r_{n+1} = P(r_n)$. [10 Marks]

(b) A three-dimensional Lotka-Volterra population model is given by:

$$\frac{dx}{dt} = x(1-2x+y-5z)$$
$$\frac{dy}{dt} = y(1-5x-2y-z)$$
$$\frac{dz}{dt} = z(1+x-3y-2z),$$

where $x, y, z$ are species populations.

(i) Explain the interactions of the three species.

[3 Marks]

(ii) Determine the location of all critical points.

[2 Marks]

(iii) Show that there is a solution plane

$x + y + z = \frac{1}{2}$. [4 Marks]

(iv) Use MATLAB and/or Python to plot solution curves in the $x + y + z = \frac{1}{2}$ plane.

[4 Marks]

(v) Interpret the results in terms of species behaviour. [2 Marks]

The COVID-era exam questions are similar to the pre-COVID exam questions. The significant difference is that care was taken to ensure that the questions focussing on could not be answered simply by looking up lecture notes or material in textbooks.

## *D. Lessons Learnt*

The last few years have seen unprecedented changes in education with the move to online teaching and new methods of assessment. At MMU, we wanted to retain the coursework component of the assessment where the students used MATLAB and/or Python to solve real-world problems, and the move to online teaching did not really affect this. What was greatly affected was the examinations.

When designing the new open book examination questions, we had to remove any bookwork questions and change the structure of the examination. We moved from optional questions, where students were able to choose five out of eight questions, each worth 20 marks, in a three-hour examination to an open book format of four compulsory questions each worth 25 marks.

The students were able to download the examination paper from Moodle on a given date and the solutions had to be submitted as a single pdf 27 hours later. Students with Personal Learning Plans or Exceptional Factors were given a total of 36 hours. This is the first time that we have used open book examinations in mathematics at MMU.

From our experiences, some of the benefits of the open book format are:

- Students prefer this method of examination.
- Pass and progression rates improved.
- Real-world questions can be posed.

The challenges of open format examinations include:

- Possible grade inflation.
- Collusion between students.

Before the coursework and examination papers are distributed to the students, they are told that they must not sit together to do the work nor share any files with each other. These warnings have been heeded by the students and we have had no cases of plagiarism or collusion during this time frame.

At MMU, we are currently continuing with the open book examinations for 2021/2022, however, we are planning to move to a mix of traditional exams and open book exams for the 2022/2023 session depending on the unit (module) content. Recently, the percentage grade weighting of module coursework to examination has shifted to 50:50 from the previous weighting ratio of 40:60.

## III. OPEN BOOK ASSESSMENTS AT UCL

## *A. Objective*

Engineering departments around the world have been using MATLAB for many decades, and at UCL MATLAB remains the programming language of choice in our teaching in engineering mathematics. Typically, at UCL students are given four pieces of take-home coursework, each consisting of four questions which takes at least 10 hours of student effort. The coursework involves assessment of mastery of mathematical theory, and mathematical modelling and analysis using MATLAB or any alternative programming language the students are familiar with.

Pre-Covid, units (modules) would have a 2-hour closed book examination consisting of four questions in an examination hall. The questions focussed exclusively on assessing mastery of mathematical theory and its applications to engineering, and programming was not required.

The total module grade weighting for the four take-home coursework was 40% for the first year and 30% for the second year. Each coursework is worth 10% and students are required to do the coursework within a four-week period. Students also do a series of online quizzes worth a total of 10% of the module weighting. The weighting of the closed book exam was 50% for the first year, and 60% for the second-year module.

### *B. Context of Implementation*

UCL reformed its undergraduate engineering curriculum in 2014, and as part of this curriculum reform, the current faculty-wide mathematics syllabus was developed [8]. Before the reforms, mathematics was taught from the Mathematics Department, and focussed primarily on mastery of mathematical theory. Following the reforms, mathematics teaching within the Faculty of Engineering Sciences was delegated to engineering staff with specific expertise in the application of mathematics to modelling and simulation of engineering problems. This change of emphasis can be seen in the integration of MATLAB as a core tool for applying mathematics to engineering design within the IEP [5].

A key objective of the current mathematics syllabus is to equip engineering students with the ability to apply mathematical concepts and theories to engineering problem solving. It is expected that on successfully completing the first- and second-year Mathematical Modelling and Analysis modules, students will be able to:

- Recognise the connections between mathematics and engineering, and how mathematical ideas are embedded in engineering contexts;
- Represent real-world systems from engineering in a mathematical framework;
- Identify and draw upon a range of advanced mathematical concepts to analyse specific problems and identify the appropriate mathematics to realise a solution;
- Employ appropriate computer programming and modelling techniques and statistical analysis to efficiently solve and evaluate the performance of engineering systems;
- Relate the behaviour of the output of mathematical models to the underlying physical or conceptual models of interest;
- Carry our engineering problem solving both collaboratively in a team and independently;
- Present and interpret mathematical results in effective and appropriate ways to varied audiences, including non-mathematical engineering audiences.

Since 2014, assessment design at UCL has focussed on assessing both the mastery of mathematics concepts and theories relevant to engineering, as well as assessing the ability to use mathematical modelling and analysis techniques in engineering problem solving.

The coursework is designed to enable students to master the use of MATLAB in mathematical modelling and analysis and was not affected by the advent of the COVID pandemic. The only piece of assessment that needed to be changed were the end of year closed book exams to ensure that they could be offered in an open book online format.

The university moved to online teaching and assessment at the onset of the pandemic, in March 2020. All first-year assessments were cancelled and replaced with a reflective pick of coursework in which the students had to demonstrate mastery of engineering concepts taught in the first year. Individual module assessment was retained for all other year levels. The only change made at this stage was the redesign of all the end of year exams, which had already been set by then, so that they would suit a 24-hour open book, online assessment format.

### *C. Question Design*

In this section we illustrate our approach towards question design in the COVID era. The first example is taken from part of a UCL coursework paper and subsequent questions are from a pre-Covid exam and post-Covid project. The first two questions are from the field of Partial Differential Equations (PDEs) and the post-Covid project presents real-world data which the students are asked to investigate using MATLAB.

#### Typical Coursework Question Part

The question below is a typical example of coursework questions used at UCL:

Consider the following Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = xe^y,$$
$$\partial w = \{(x, y): 0 < x < 1 \; and \; 0 < y < 1\},$$
$$u(0, y) = 0, u(2, y) = 2e^y \; for \; 0 \leq y \leq 1,$$
$$u(x, 0) = x, u(x, 1) = ex \; for \; 0 \leq x \leq 1.$$

(i) Using a Taylor series expansion, derive the central difference approximations for $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial y^2}$ at the point $(ih, jh)$ and quote the order of the error terms.

(ii) Give the difference equation that must be satisfied by $u(ih, jh)$ at each point $(ih, jh)$ after discretizing $\partial w$.

(iii) Obtain the 16 difference equations that must be solved to obtain the numerical solution to the given boundary value problem.

(iv) Write MATLAB code that solves this boundary value problem on a uniform mesh with $h = k = 0.2$. You may use an iterative scheme of your choice such as the Gauss-Seidel technique with the stopping criteria:

$$\left|u_{i,j}^{(n)} - u_{i,j}^{(n-1)}\right| < 10^{-5},$$

where the subscript denotes the iteration stage.

Alternatively, you may use Cramer's rule, or you may choose to invert the matrix obtained from the algebraic formulation of this problem i.e., when you write the PDE as a system:

$$Aw_i = b_i,$$

where $A$ is a 16x16 matrix.

(v) Within your MATLAB code, compare your numerical solutions (16 interior points) with the exact solution given by $u(x, y) = xe^y$.

The format of the UCL coursework questions is similar to the MMU coursework question format. Both question types avoid bookwork and emphasise mathematical modelling and analysis. An element of research is incorporated, as students are required to go beyond the material covered in lectures. As with the MMU coursework, students need to demonstrate an understanding of the underlying mathematics concepts, an ability to formulate solutions to the problem, MATLAB coding and implementation ability, and, finally, an ability to interpret the model outputs and apply the solutions to the physical world.

### Pre- COVID Closed Book Exam Question

The question below is a typical example of pre-COVID closed book exam questions used at UCL:

> Solve the following equation, which holds in the rectangle $0 < x < a$, $0 < y < b$, for $u(x,y)$:
>
> $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,$$
>
> $$\frac{\partial u}{\partial y}(x,0) = 0; \frac{\partial u}{\partial x}(0,y) = 0; u(a,y) = 2; u(x,b) = 2.$$
>
> The equation is to be solved in three stages, as described in parts (a), (b) and (c) below.
>
> (a) First, solve the following equation, which holds in the rectangle, $0 < x < a, 0 < y < b$, for $v(x,y)$:
>
> $$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = 0,$$
>
> $$\frac{\partial v}{\partial y}(x,0) = 0; \frac{\partial v}{\partial x}(0,y) = 0;$$
> $$v(a,y) = 0; v(x,b) = 2.$$
>
> Hint: This elliptic PDE is in standard form and can be solved by separation of variables.
>
> (b) Next, solve the following equation, which holds in the rectangle, $0 < x < a, 0 < y < b$, for $w(x,y)$:
>
> $$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = 0,$$
>
> $$\frac{\partial w}{\partial y}(x,0) = 0; \frac{\partial w}{\partial x}(0,y) = 0;$$
> $$w(a,y) = 2; w(x,b) = 0.$$
>
> Hint: do not solve the equation in detail. Instead, apply a suitable transformation to the result in part (a) to arrive at the desired result for part (b).
>
> (c) Finally, calculate $u(x,y)$ in the original PDE as a suitable combination of the results in parts (a) and (b).

As this question illustrates, pre-COVID exam questions comprised a bookwork element, and elements assessing mastery of theoretical concepts. Some questions also included elements assessing the application of mathematical techniques to engineering problems. The exam questions did not include any assessment of MATLAB programming and were specifically designed to complement the coverage of mathematical modelling and analysis in the coursework questions.

### Post-Covid Exam and Project Assessment

As at MMU, the redesign of the second-year mathematical modelling and analysis exam into an online open book format required the replacement of bookwork questions with questions that could not be answered simply by looking up lecture notes, textbook material, or carrying out Internet searches. Given the short period before the exam and the relative inexperience of the teaching staff in developing such questions, this was a significant challenge. Student performance in the open book online exam was far higher than the average performance of students in closed book exams set in previous years.

The high performance of the students in the second-year mathematical modelling and analysis open book exam was deemed an inaccurate representation of the students' subject mastery. This necessitated a rethink of our assessment approaches at UCL, leading to the introduction of a mathematical modelling and analysis project as the final assessment component. For the first year, students were required to do the project over a two-week period, whilst for the second year, students had to do the project within a 48-hour window. Both the first year and second year projects comprised of engineering case studies in which students were required to use mathematical modelling and analysis techniques to come up with viable engineering solutions.

Project questions were specifically designed to guide the students through the mathematical modelling and analysis steps, with questions focussing on analysis and interpretation of modelling outputs being given a higher weighting. Each project had an element of some independent research, which meant that students had to use the research skills taught in the accompanying first year and second year Design and Professional Skills modules to seek out information from the UCL online library resources, and from the open Internet. As with all other project work within the UCL Faculty of engineering Sciences, support structures were put in place to ensure that students would not flounder.

Below is a sample from the 2021 second year mathematics project:

> The file, *Section_3_data.txt* (given) contains pulse oximeter signal data (in arbitrary units) of variation in blood volume. The fluctuations in the signal are an indication of the heart rate of the patient and estimating the period of these fluctuations is a way to measure the period between heart beats. The data are recorded on a patient over a period of 100 seconds using a sample interval of 100 milliseconds (ms). The signal is contaminated by noise and is also sensitive to the patient's respiratory cycle (breathing rate), which can also be detected by the fluctuations in the signal. Use MATLAB to analyse the data as outlined below.
>
> (a) Obtain a plot of the data which shows the variation in signal over 100 seconds. [2 Marks]
> (b) Generate the Fourier transform of the data and plot the absolute values of the transform in the frequency domain. The code below will create a transform of the data stored in the vector x and plot over the interval from -5Hz to +5Hz. Note that the zero-frequency components will be shifted to the centre of the array to simplify the analysis. [2 Marks]
> (c) Use the plot generated in part (b) to extract both the patient's cardiac cycle and respiratory cycle frequencies (in units of Hz) and periods. [2 Marks]
> (d) The signal due to the cardiac cycle can be observed more easily by applying a "frequency filter" which suppresses the contributions at other frequencies.

  - (i) Generate a filter which is a sum of two narrow Gaussian functions centred at the positive and negative cardiac frequencies, respectively. [3 Marks]
  - (ii) Apply the filter to the Fourier transform of the signal obtained in part (b) by multiplying the filter and the transformed signal and plot a comparison between the unfiltered and filtered FT signal. [3 Marks]
- (e) Obtain the inverse Fourier transform of the filtered data using the MATLAB code in the file, *Section3_5.m,* and plot the filtered data values stored in the vector x2. [2 Marks]
- (f) Explore the effect of varying the widths of the Gaussian functions in your filter and discuss your findings. [6 Marks]

### *D. Student Outcomes for the Project*

Feedback from students regarding the project has been very positive. Students appreciate the opportunity to link mathematics to engineering-oriented contexts, and students report that the project has given them the opportunity to explore and demonstrate their individual creativity to mathematical problem solving, something they were unable to do in the closed book exams. Importantly, students appreciate the fact that engineering mathematics is no longer just a theoretical module, but a module in which they can actively engage in engineering problem solving. In addition, students appreciate that even though the project is an individual effort, there is support available for them should they need it.

## IV. CONCLUDING REMARKS

In this paper we have presented case studies from two universities demonstrating the use of MATLAB and Python in open book online exams in mathematics. Preliminary feedback from both institutions suggest that students appreciate this mathematical modelling and analysis to mathematics assessments. Indications from both institutions also suggest that collusion and plagiarism levels in open book online assessments may not be as high as anticipated. However, this research is ongoing, and more investigations need to be carried out to fully ascertain this aspect.

## REFERENCES

[1] M. Peplow, "Computational Modelling: Technological Futures. Government Office for Science and Council for Science and Technology.," 2018. [Online]. Available: https://www.gov.uk/government/publications/computational-modelling-blackett-review

[2] P. Bond, "The Era of Mathematics – Review Findings on Knowledge Exchange in the Mathematical Sciences. ," EPSRC and the Knowledge Trasnfer Network, 2018. [Online]. Available: https://epsrc.ukri.org/research/ourportfolio/themes/mathematics/strategy/kereview/

[3] Z. Hussain and M. S. Khan, "Introducing Python Programming for Engineering Scholars," *International Journal of Computer Science and Network Security,* vol. 18, pp. 26-33, 2018.

[4] V. F. Ochkov, A. A. Sutchenkov, and A. I. Tikhonov, "Python Computational Web Apps for STEM Engineering Education " *International J. of Educationand Information Technologies,* vol. 15, pp. p130-136, 2021.

[5] A. Nyamapfene and S. Lynch, "Systematic integration of MATLAB into undergraduate mathematics teaching: Preliminary lessons from two UK institutions," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 10-13 April 2016 2016, pp. 1145-1148, doi: 10.1109/EDUCON.2016.7474699.

[6] S. Lynch, "Programming in the mathematics curriculum at Manchester Metropoltan University," *MSOR Connections,* vol. 18, pp. 5-12, 2020.

[7] G. Smith, L. Wood, M. Coupland, B. Stephenson, K. Crawford, and G. Ball, "Constructing mathematical examinations to assess a range of knowledge and skills," *International Journal of Mathematical Education in Science and Technology,* vol. 27, no. 1, pp. 65-77, 1996/01/01 1996, doi: 10.1080/0020739960270109.

[8] J. E. Mitchell, A. Nyamapfene, K. Roach, and E. Tilley, "Faculty wide curriculum reform: the integrated engineering programme," *European Journal of Engineering Education,* vol. 46, no. 1, pp. 48-66, 2021/01/02 2021, doi: 10.1080/03043797.2019.1593324.