

Study Group ISCs (UK & Europe)
Assignment Brief

Partner HEI	Durham University
ISC/IC	Durham University ISC
Programme/Pathway	SCEN – PMP Science, Engineering and Mathematics
Module Title	Introduction to Scientific Computing (35 credit)

Assignment details

Assessment components	Marks available	Component weighting	Learning outcomes
A2 – Project 1 report (2000 words)	100 %	40 %	3, 6, 9–12

In order to pass the module, students must achieve a minimum of 50 % in all summative assessments. Resit and reattempt opportunities are available for all summative assessment. Progression will be based on the achievement of the agreed progression grade.

Learning outcomes being assessed
Knowledge and understanding

- 3 explain the key theoretical concepts and practical aspects of using numerical methods to solve scientific and mathematical problems

Subject-specific

- 6 design and develop fast and efficient algorithms using C and Python to solve real-world scientific and mathematical problems using numerical methods

Key skills and employability skill

- 9 use numbers in appropriate engineering and scientific formats when solving numerical problems
- 10 communicate effectively in writing using appropriate vocabulary and academic standards
- 11 employ appropriate computer skills to produce technical documents to appropriate scientific/engineering academic standards (e.g., IEEE standards, LaTeX Typesetting)
- 12 develop a range of effective interpersonal and problem-solving skills

Assignment task**Project 1****Numerical methods to solve ODE systems using C and Python****Description**

In this project, you will be challenged to identify a scientific and mathematical problem and apply your knowledge of numerical techniques along with your C and Python programming skills to develop a solution. You will also have the opportunity to reflect on the outcomes of your work, the problem-solving process and the limitations of the project.

You will be required to communicate your findings in the form of an academically rigorous technical report using LaTeX in Overleaf following the template provided and general IEEE standards.

It is expected that this project will be carried out over the last four weeks of term, and you will be expected to organise your workshop and personal study time to complete your work within that period.

Tasks and guidance

It is expected that you will carry out the following general activities to complete your project. Further guidance on writing the technical report is provided in the *Structure and content* section of this assessment brief.

1. Choose and introduce a system to investigate

Choose a suitable scientific and mathematical first or second-order initial-value problem to investigate. You can choose from the list of example ODE-based systems provided in Appendix A or choose one more closely related to your field of study. You must agree your choice with your tutor as part of Formative Assessment F3.

2. Explain the background theory of using numerical methods to solve ODEs

Explain and demonstrate the theoretical concepts and practical aspects of general numerical methods of solving first-order and second-order ODEs for initial-value problems. At a basic level, the methods should include the Euler method and at least one improvement on the Euler method such as Heun's method or the midpoint method. At a higher level, this could include a description of the Runge-Kutta methods, specifically the classical fourth-order RK method (RK4). You may use Jupyter Notebooks or individual C and Python code to demonstrate the general principles of each numerical method.

3. Implement a specific numerical solution to your problem using C and Python code

Develop fast and efficient algorithms and C and Python code to find solutions to the ODE system you have chosen to investigate. The code you produce must be able to run on the Linux server in the computer science lab D104. You must use Jupyter Lab Notebooks to demonstrate code and present output, but can also produce full, user-friendly standalone C and Python code files. All Jupyter files and code files must be submitted to the specified Turnitin portal, and will be downloaded and tested by the marker to verify their operation and output before grades can be awarded.

4. Present and discuss your results

Clearly present the results of your numerical solution and discuss your results in relation to your original problem. You must clearly identify the initial problem you wished to solve, and whether your methods and code have resulted in a valid solution. Furthermore, you should evaluate your algorithms and code in terms of accuracy, speed and efficiency. You should also reflect on the outcomes of your work, the problem-solving process and the limitations of your project.

5. Communicate your work effectively

Use the LaTeX template provided in Overleaf to write a technical report covering each of the tasks above. Follow the guidance in the next section when writing your report. Submit the PDF of your report to the specified Turnitin portal.

Structure and content of the project report

The project report should be typeset using the LaTeX template provided in Overleaf. The word count should be 2000 words. The following is a list of the required sections with suggested word counts and guidance on the content of each section. Look at grading criteria for more idea of the expected depth of content for each section.

Use the features in Overleaf to determine your word count for each of your sections. Try to stick to the word count for each section as closely as possible: the word limit is designed to stop you from adding too much irrelevant detail; your grade for each section will be affected if you are significantly under or over the specified word count.

Do not use enumerated lists e.g. 1 (a) in your report: just use clear sections, subsections and paragraphs. Make sure your paragraphs have a clear, appropriate topic sentence and are of an appropriate length - not too short and not too long. In Overleaf, just leave a blank line between paragraphs: the spacing will be automatic.

Do not write your report in a word processor such as Microsoft Word and then try to copy/paste into Overleaf: this will result in errors. Just type directly into Overleaf. Your tutor can provide extra support with using Overleaf/LaTeX.

1. Introduction (300)

Introduce your chosen ODE problem to solve. Try to follow a standard three-stage structure: general context; specific topic; specific aims of the project. You must clearly and accurately show and explain the equations for your chosen problem.

2. Background theory (500)

Explain the general underlying theory of solving ODEs using numerical methods. This is where you can demonstrate your level of knowledge and understanding of the key theories and methods. Diagrams are often useful to illustrate numerical methods. Remember to show all equations clearly and accurately. Refer to the grading criteria for some idea of the depth required for this section.

3. Numerical solution (500)

This is the main programming task. The programming activities should be carried out in the workshop sessions (second lesson of the week) using the computer science server and workstations as well as in your own time using your Linux VM.

Remember to explain which numerical methods you are using and show the equations for your specific numerical solution in your report. Your solutions should be developed and presented as Jupyter Lab notebooks. Make sure your final notebooks work on the computer science lab servers. Remember to include some relevant markdown documentation cells and comments within your notebooks, not just code. Use this section of the report to explain your methods and any important choices you made when developing your code. Include PDF versions of your notebooks in an appendix.

4. **Results and discussion (500)**

Final graphs obtained in your Jupyter notebooks should be included as figures in your report. Use tables to show summaries of final numerical results in an appropriate format.

5. **Conclusion (200)**

Try to follow the standard three-stage structure used in your introduction but in reverse. At the end of your conclusion, you could make some recommendations for future studies, experiments or work. Research the RERUN method of writing a conclusion.

6. **References**

This section will be produced automatically by the LaTeX document in the appropriate IEEE style if you use the `\cite{}` command within your report and accompanying entries in your `references.bib` file. Try to cite relevant text books and articles to support your work. You are more likely to use internet sources for this project, as it refer to technology which is always changing at a fast pace. Try to evaluate the credibility of your sources before using them, as discussed in class.

7. **Appendices**

This section will be produced automatically by the LaTeX document if you upload your code file to your Overleaf project and use the correct `\lstinputlisting` command to include your code file. Make sure you have an appropriate title for each appendix section, and an appropriate caption for each code listing or console output listing. Follow the guidance provided in class.

Assessment criteria

A separate, more readable copy of the grading rubric will be provided on the VLE for this module.

P_ISCP35: Assessment Criteria for A2-Project 1 – Numerical solutions to ODEs

Aspect	Weighting	PROGRESSION+			PROGRESSION		PASS		FAIL	
		Advanced+ 100%-90% (outstanding)	Advanced- 89%-80% (excellent)	Effective+ 79%-73% (very good)	Effective- 72%-65% (good)	Pass+ 64%-58% (sound)	Pass- 57%-50% (acceptable)	Fai+ 49%-25% (narrow fail)	Fai- 24%-0% (unacceptable)	
Introduction: general context, specific focus, aims and ODEs	15%	Very appropriate context with clear and logical movement to specific topic. Very clear and relevant aims. Clear, appropriate and well-presented chosen ODEs for this project.		Context is mostly appropriate and may lack clarity or depth. The topic and aims are mostly appropriate and may lack clarity or be too general. The chosen ODEs are appropriate but may lack clarity in both explanation and presentation of equations.		Context only partially appropriate and/or aims may be unclear. The specific topic and aims are very limited, missing or very unclear. Only a very limited ability to show appropriate ODEs.		Fails to meet the required standards. >25% could pass with improvements to some sections <25% not enough work to warrant resit		
Background theory: relevant theory, understanding of topic	25%	Demonstrates excellent and clear K&U of underlying theory of numerical methods to solve ODEs, including advanced methods such as Huen's and/or RK4. Successfully cites some appropriate sources.		Demonstrates good K&U, but may lack depth and clarity in places. Relies on basic methods more than advanced methods. Only partially successful when citing appropriate sources.		Demonstrates some K&U, but lacking relevance or clarity and covers only very basic theory. Very limited or no success in citing appropriate sources.		Fails to meet the required standards. >25% could pass with improvements to some sections <25% not enough work to warrant resit		
Numerical solution: C and Python code advanced methods use of libraries e.g. GSL	25%	Demonstrates excellent and independent programming skills in both C and Python to develop a credible solution to the ODEs for this project using advanced features and libraries. Implementation is clearly and accurately documented.		Demonstrates very good programming skills in C and Python to develop a mostly credible solution to the ODEs for this project, but may be lacking in one language and may only use basic features and may have issues. Documentation may be lacking in depth and clarity.		Demonstrates some programming skills in either Python or C to develop a partial solution to the ODEs for this project using only basic methods. Documentation may be very limited or missing.		Fails to meet the required standards. >25% could pass with improvements to some sections <25% not enough work to warrant resit		
Results and discussion: well presented critical evaluation	25%	Excellent presentation of accurate and appropriate results with clear identification and critical evaluation of relevant issues in relation to theory and practice.		Creates and presents mostly appropriate results. Mostly relevant issues discussed. Evaluation is more superficial than critical in relation to theory or may be limited in depth.		Creates some relevant results but may be poorly presented or lacking in clarity. Only very limited success in identifying relevant issues. Very limited evaluation in relation to theory.		Fails to meet the required standards. >25% could pass with improvements to some sections <25% not enough work to warrant resit		
Professional communication: use of LaTeX templates, IEEE standards, logical structure and clarity, accuracy of language	10%	Excellent use of LaTeX template. Appropriate academic structure. Accurate and professional presentation of work to standards. Clear, logical development of ideas with appropriate paragraphing. Language is accurate and appropriate.		Good use of LaTeX template with minor issues. Mostly appropriate structure. Mostly accurate and professional presentation of work to standards with minor issues. Mostly clear, logical paragraphing. Language is mostly accurate and appropriate.		Limited success in using LaTeX template provided without issues. Most parts of the work have issues with presentation standards. Ideas and paragraphs lack clarity and organisation. Inaccuracies in language can cause problems for the reader.		Fails to meet the required standards. >25% could pass with improvements to some sections <25% not enough work to warrant resit		

Submission of the assignment

VLE submission deadline

You must submit your PDF report and ZIP file of your Jupyter notebooks to the VLE no later than the end of week 16 by 13:00 UK time; the specific submission date will be provided on the VLE page for this module..

VLE upload submission instructions

Create a PDF of your Overleaf LaTeX report and a ZIP file of your original Jupyter notebooks and rename your files in the format

GROUP_P_ISC35_A2_abcd12_20123456

replacing GROUP with your group code, e.g. SPS, JPS or JP2S, 20123456 with your student ID number and abcd12 with your VLE user name. Please ensure your files are uploaded before the deadline to avoid late penalties and take a screen shot or print a copy of your digital receipt. The latest submission time of both files will be taken as the submission time, which means if either one of your files is late, your entire submission will be recorded as being late.

Plagiarism and collusion

It is very important that any work you present as yours must in fact be your own work, and not taken from another place or done by another person. Cheating, collusion (working together with another person on an assessment which is not intended to be collaborative) and copying from unacknowledged sources (plagiarism) are all serious offences and must be avoided.

Academic impropriety

Academic impropriety is a term that covers cheating, attempts to cheat, plagiarism, collusion and any other attempts to gain an unfair advantage in assessments. Assessments include all forms of written work, presentations and all forms of examination. Academic impropriety, in any form, is a serious offence and the penalties imposed would reflect this, although drawing a distinction between minor and major instances of misconduct.

Further details

For further details on how to complete the front cover of your assignment, please check the relevant module VLE page under Assessment.

For further details on academic impropriety and plagiarism, please check the relevant module VLE page under Assessment.

For full details of the Personal Circumstances, Appeals and Academic Impropriety processes, please refer to pages 59–74 of the Student Handbook (available on the VLE Virtual Reception).

Appendices

A Example ODE systems and problems

The following are some examples of systems which can be modelled using ODEs. More details about each system can be provided by your tutor when discussing your chosen system during formative F3.

A.1 Population: predicted growth

This simple first-order system can be used to model population growth. The problem could be to predict the population growth in a specific area given certain environmental factors.

A.2 Mass-spring-damper: train suspension

This second-order oscillatory system is commonly used to model the behaviour of vehicle suspension systems. For example, it could be used to determine the optimum damping required for a large mass transport vehicle such as a high-speed train.

A.3 Pendulum with drag: helicopter slung load

This second-order oscillatory system can be used to model a slung load such as a helicopter transporting a heavy load. An example problem would be to determine the optimum forward velocity of the helicopter such that if the forward velocity suddenly came to zero, the mass would not swing too wildly.

A.4 Free-falling body with parachute: air-drop load

This simple second-order system can be used to model a set of large linked loads being air-dropped from an aeroplane. The problem could be to determine the optimum height and parachute parameters to safely drop the linked loads in the shortest time.

A.5 Free-falling body with elastic rope: bungee jumpers

This second-order oscillatory system would model the behaviour of a jumper connected to a fixed platform using an elastic bungee rope. A typical problem to solve would be to determine the optimum parameters of the rope for a set of linked jumpers to provide a thrilling but safe experience.

A.6 Projectile: launch velocity

This second-order system could be used to model the launching of a space rocket or other projectile into orbit. The problem could be to determine the initial trajectory and velocity required to achieve a low-earth orbit.

A.7 The SIR model for the spread of disease: modelling local virus outbreaks

This set of simple first-order ODEs can be used to model the spread of a disease in a population as well as the infection and recovery rates. This could be used to model the outbreak of a virus in a small localised population such as in an educational establishment and to determine best policies to minimise the number of infections.

A.8 Valuing an annuity paying continuous cash flow

One common application of simple first-order ODEs in finance is in modelling continuous compounding and financial plans that pay a continuous flow of cash. The problem could be to determine the optimum time a teacher should retire.