UNIVERSITY OF CAMBRIDGE

Mathematical Tripos Part IB

Computational Projects

2024/2025

CATAM

Mathematical Tripos Part IB
Computational Projects

July 2024

Edited by the Computational Projects Assessors Committee

Course Director: Prof J.R. Taylor

Assistant Course Director: Dr D. Frank

Computer-Aided Teaching of All Mathematics

Faculty of Mathematics

University of Cambridge

Contents

Introduction

Optional Introductory Non-Examinable Project

Core Projects

Additional Projects

For maximum credit, you should attempt both projects from section 1 (*Core Projects* above), and then two additional projects chosen from section 2 (*Additional Projects*). You may not attempt more than two additional projects. All projects carry equal credit.

Introduction

1 General

Please read the whole of this introductory chapter before beginning work on the projects. **It contains important information that you should know as you plan your approach to the course.**

1.1 Introduction

The first Computational Projects course is an element of Part IB of the Mathematical Tripos (the second Computational Projects course is an element of Part II). Although a Part IB course, lectures and introductory sessions were given as part of the Part IA year. After the lectures and sessions, and once the manual has been published, you may work at your own speed on the examinable projects.

The course is an introduction to the techniques of solving problems in mathematics using computational methods. **It is examined entirely through the submission of project reports**; there are no questions on the course in the written examination papers.

The definitive source for up-to-date information on the examination credit for the course is the Faculty of Mathematics Schedules booklet for the academic year 2024-25. At the time of writing (July 2024) the booklet for the academic year 2021-22 states that

> *No questions on the Computational Projects are set on the written examination papers, credit for examination purposes being gained by the submission of reports. The maximum credit obtainable is 160 marks and there are no alpha or beta quality marks. Credit obtained is added directly to the credit gained on the written papers. The maximum contribution to the final merit mark is thus 160, which is roughly the same (averaging over the alpha weightings) as for a 16-lecture course. Projects are considered to be a single piece of work within the Mathematical Tripos.*

1.2 The nature of CATAM projects

CATAM projects are intended to be exercises in independent investigation somewhat like those a mathematician might be asked to undertake in the 'real world'. They are well regarded by external examiners, employers and researchers (and you might view them as a useful item of your *curriculum vitae*).

The questions posed in the projects are more open-ended than standard Tripos questions: **there is not always a single 'correct' response**, and often the method of investigation is not fully specified. **This is deliberate.** Such an approach allows you both to demonstrate your ability to use your own judgement in such matters, and also to produce mathematically intelligent, relevant responses to imprecise questions. Particularly with respect to the Additional Projects (2.1 to 2.4), you will also gain credit for posing, and responding to, further questions of your own that are suggested by your initial observations. You are allowed and encouraged to use published literature (but it must be referenced, see also §5) to substantiate your arguments, or support your methodology.

1.3 Timetable

The timetable below is given as a guide to the expected workload.

End of Lent Term and Easter Term, Part IA: work through the MATLAB booklet, and attend the introductory lectures, and a MATLAB session if available. If you have no previous computing experience then you may need to spend extra time learning the basics; the summer vacation is a good opportunity to do this.

Over the summer between Part IA and Part IB: it is strongly recommended that you do the *optional*, non-examinable, *Introductory Project*. Unlike the other projects you may collaborate as much as you like on this project, and your College can arrange a supervision on it. A model answer will be provided towards the start of the Michaelmas Term.

Note that, if you wish, you may start the core and/or additional projects over the summer (once they are published). However,

- you are advised to attempt the *Introductory Project* first;
- please make sure that you have read and understood §5, *Unfair Means, Plagiarism and Guidelines for Collaboration*, before starting the assessed projects.

Michaelmas Term and Christmas vacation, Part IB: complete the programming and write-ups for the **two core projects**. A good *aim* is to finish and submit these projects by the end of the Christmas vacation.

Lent Term and Easter vacation, Part IB: you have one week at the start of Lent Term to make *last-minute* changes to the core projects, which should then be submitted (see §6.2 below). Then undertake **two additional projects** (out of a choice of four) and write them up.

- Between the submission deadline and the end of Lent Full Term, you may be called either for a routine *Viva Voce Examination* or, if unfair means are suspected (see §5.2 below), for an *Examination Interview* or for an *Investigative Meeting.*
- To help you assess where marks have been gained/lost, the completed marking scheme for each *Core Project* will be returned to you before the end of Lent Full Term. At that stage the Faculty will aim to offer a short "feedback" session on one of the Core Projects to those who have submitted.

Easter Term, Part IB: you have one week to make any *last-minute* changes to the additional projects.

After the examinations: **you must be available** in the last week of Easter term in case you are called either for a routine *Viva Voce Examination* or, if unfair means are suspected (see §5.2 below), for an *Examination Interview* or an *Investigative Meeting.*

1.3.1 Planning your work

- You are strongly advised to complete all your computing work by the end of the Christmas and Easter vacations if at all possible, since the submission deadlines are at the start of Lent and Easter Terms.
- **Do not leave writing up your projects until the last minute.** When you are writing up it is highly likely that you will either discover mistakes in your programming and/or want to refine your code. This will take time. If you wish to maximise your marks, the

process of programming and writing-up is likely to be **iterative**, ideally with at least a week or so between iterations.

- It is a good idea to write up each project as you go along, rather than to write all the programs first and only then to write up the reports; each year several students make this mistake and lose credit in consequence (in particular note that a program listing without a write-up, or vice versa, gains no credit). You can, indeed should, review your write-ups in the final week before the relevant submission date.

1.4 Programming language[s]

This year the Faculty is supporting MATLAB as the programming language. During your time in Cambridge the University will provide you with a free copy of MATLAB for your computer. Alternatively you can use the version of MATLAB that is available on the Managed Cluster Service (MCS) that is available at a number of UIS and institutional sites around the Collegiate University.

1.4.1 Your copy of MATLAB

All undergraduate students at the University are entitled to download and install MATLAB on their own computer that is running Windows, MacOS or Linux; your copy should be used for non-commercial University use only. The files for download, and installation instructions, are available at

`http://www.maths.cam.ac.uk/undergrad/catam/software/matlabinstall/matlab-personal.htm`.

This link is Raven protected. Several versions of MATLAB may be available; if you are downloading MATLAB for the first time it is recommended that you choose the latest version.

1.4.2 Programming guides and manual[s]

The Faculty of Mathematics has produced a booklet *Learning to use MATLAB for CATAM project work*, that provides a step-by-step introduction to MATLAB suitable for beginners. This is available on-line at

`http://www.maths.cam.ac.uk/undergrad/catam/MATLAB/manual/booklet.pdf`

However, this short guide can only cover a small subset of the MATLAB language. There are many other guides available on the net and in book form that cover MATLAB in far more depth. In addition:

- MATLAB has its own extensive built-in help and documentation.
- The suppliers of MATLAB, *The MathWorks*, provide MATLAB Onramp, an interactive tutorial on the basics which does not require MATLAB installation: see

  `http://uk.mathworks.com/support/learn-with-matlab-tutorials.html`

- *The MathWorks* also provide the introductory guide *Getting Started with MATLAB*. You can access this by 'left-clicking' on the `Getting Started` link at the top of a MATLAB *'Command Window'*. Alternatively there is an on-line version available at

  `http://uk.mathworks.com/help/matlab/getting-started-with-matlab.html`

- Further, *The MathWorks* provide links to a whole a raft of other tutorials; see

  https://uk.mathworks.com/support/learn-with-matlab-tutorials.html

  In addition their *Matlab* documentation page gives more details on maths, graphics, object-oriented programming etc.; see

  http://uk.mathworks.com/help/matlab/index.html

- There is a plethora of books on Matlab. For instance:

  (a) *Numerical Computing with Matlab* by Cleve Moler (SIAM, Second Edition, 2008, ISBN 978-0-898716-60-3). This book can be downloaded for free from

  http://uk.mathworks.com/moler/chapters.html

  (b) *Matlab Guide* by D.J. Higham & N.J. Higham (SIAM, Second Edition, 2005, ISBN 0-89871-578-4).

  You may be spoilt for choice: Google returns about 100,000,000 hits for the search 'Matlab introduction', and about 11,000,000 hits for the search 'Matlab introduction tutorial'.

- The Engineering Department has a webpage that lists a number of helpful articles; see

  http://www.eng.cam.ac.uk/help/tpl/programs/matlab.html

1.4.3 To Matlab, or not to Matlab

Use of Matlab is recommended,[1] especially if you have not programmed before, but *you are free to write your programs in any computing language whatsoever.* Python, Julia,[2] R,[3] C, C++, Mathematica,[4] Maple[5] and Haskell have been used by several students in the past, and Excel has been used for plotting graphs of computed results. The choice is your own, provided your system can produce results and program listings for inclusion in your report.[6]

However, you should bear in mind the following points.

- The Faculty does *not* promise to help you with programming problems if you use a language other than Matlab.

- Not all languages have the breadth of mathematical routines that come with the Matlab package. You may discover either that you have to find reliable replacements, or that you have to write your own versions of mathematical library routines that are pre-supplied in

[1] Except where an alternative is explicitly stated, e.g. see footnotes 3 and 5.

[2] Julia is a high-level open source language well suited to numerical computation. An Introduction to Julia for CATAM under ongoing development is available from https://sje30.github.io/catam-julia/.

[3] R is a programming language and software environment for statistical and numerical computing, as well as visualisation. It is the recommended language for some Part II projects. R is available for free download for the Linux, MacOS and Windows operating systems from http://www.r-project.org/.

[4] Mathematica is a software package that supports symbolic computations and arbitrary precision numerical calculations, as well as visualisation. At the time of writing Mathematica is also available for free to mathematics students, but the agreement is subject to renewal. You can download versions of Mathematica for the Linux, MacOS and Windows operating systems from

https://www.maths.cam.ac.uk/computing/software/mathematica/

[5] Maple is a mathematics software package that supports symbolic computations and arbitrary precision numerical calculations, as well as visualisation. It is the recommended language for some Part II projects.

[6] There is no need to consult the *CATAM Helpline* as to your choice of language.

MATLAB (this can involve a fair amount of effort). To this end you may find reference books, such as *Numerical Recipes* by W. H. Press *et al.* (CUP), useful. You may use equivalent routines to those in MATLAB from such works so long as you acknowledge them, and reference them, in your write-ups.

- If you choose a high-level programming language that can perform advanced mathematical operations automatically, then you should check whether use of such commands is permitted in a particular project. As a rule of thumb, do not use a built-in function if there is no equivalent MATLAB routine that has been approved for use in the project description, or if use of the built-in function would make the programming considerably easier than intended. For example, use of a command to test whether an integer is prime would not be allowed in a project which required you to write a program to find prime numbers. The *CATAM Helpline* (see §4 below) can give clarification in specific cases.
- Subject to the aforementioned limited exceptions, *you must write your own computer programs*. Downloading computer code, e.g. from the internet, that you are asked to write yourself counts as plagiarism (see §5).

2 Project Reports

2.1 Project write-ups: examination credit

Each individual project carries the same credit. For each project, 40% of the marks available are awarded for writing programs that work and for producing correct graphs, tables of results and so on. A further 50% of the marks are awarded for answering mathematical questions in the project and for making appropriate mathematical observations about your results.

The final 10% of marks are awarded for the overall 'excellence' of the write-up. Half of these 'excellence' marks may be awarded for presentation, that is for producing good clear output (graphs, tables, etc.) which is easy to understand and interpret, and for the mathematical clarity of your report.

The assessors may penalise a write-up that contains an excessive quantity of irrelevant material (see below). In such cases, the 'excellence' mark may be reduced and could even become negative, as low as -10%.

Unless the project specifies a way in which an algorithm should be implemented, marks are, in general, not awarded for programming style, good or bad. Conversely, if your output is poorly presented — for example, if your graphs are too small to be readable or are not annotated — then you may lose marks.

No marks are given for the submission of program code without a report, or vice versa.

The marks for each project are scaled so that a possible maximum of 160 marks are available for the Part IB Computational Projects course. No quality marks (i.e. αs or βs) are awarded. The maximum contribution to the final merit mark is thus 160 and roughly the same (averaging over the α weightings) as for a 16-lecture course.

2.2 Project write-ups: advice

Your record of the work done on each project should contain all the results asked for and your comments on these results, together with any graphs or tables asked for, clearly labelled and referred to in the report. However, it is important to remember that the project is set as a piece of mathematics, rather than an exercise in computer programming; thus the most important aspect of the write-up is the **mathematical content**. For instance:

- Your comments on the results of the programs should go *beyond* a rehearsal of the program output and show an understanding of the mathematical and, if relevant, physical points involved. The write-up should demonstrate that you have noticed the most important features of your results, and understood the relevant mathematical background.

- When discussing the computational method you have used, you should distinguish between points of interest in the algorithm itself, and details of your own particular implementation. Discussion of the latter is usually unnecessary, but if there is some reason for including it, please set it aside in your report under a special heading: it is rare for the assessors to be interested in the details of how your programs work.

- Your comments should be pertinent and concise. Brief notes are perfectly satisfactory — provided that you cover the salient points, and make your meaning precise and unambiguous — indeed, students who keep their comments concise can get better marks. An over-long report may well lead an assessor to the conclusion that the candidate is unsure of the essentials of a project and is using quantity in an attempt to hide the lack of quality. Do not copy out chunks of the text of the projects themselves: you may assume that the assessor is familiar with the background to each project and all the relevant equations.

- Similarly you should not reproduce large chunks of your lecture notes; you will not gain credit for doing so (and indeed may lose credit as detailed in §2.1). However, you will be expected to reference results from theory, and show that you understand how they relate to your results. If you quote a theoretical result from a textbook, or from your notes, or from the WWW, you should give both a brief justification of the result and a *full reference*.[7] If you are actually asked to *prove* a result, you should do so concisely.

- Graphs will sometimes be required, for instance to reveal some qualitative features of your results. Such graphs, *including labels, annotations, etc.*, need to be computer-generated (see also § 2.3). Further, while it may be easier to include only one graph per page, it is often desirable (e.g. to aid comparison) to include *two or more* graphs on a page. Also, do not forget to clearly label the axes of graphs or other plots, and provide any other annotation necessary to interpret what is displayed. Similarly, the rows and columns of any tables produced should be clearly labelled.

- You should take care to ensure that the assessor sees evidence that **your programs do indeed perform the tasks** you claim they do. In most cases, this can be achieved by including a sample output from the program. If a question asks you to write a program to perform a task but doesn't specify explicitly that you should use it on any particular data, you should provide some 'test' data to run it on and include sample output in your write-up. Similarly, if a project asks you to 'print' or 'display' a numerical result, you should demonstrate that your program does indeed do this by including the output.

- **Above all, make sure you comment where the manual specifically asks you to.** It also helps the assessors if you answer the questions in the order that they appear in the manual and, if applicable, **number your answers** using the same numbering scheme as that used by the project. Make clear which outputs, tables and graphs correspond to which questions and programs.

[7] See also the paragraph on *Citations* in §5

The following are indicative of some points that might be addressed in the report; they are not exhaustive and, of course not all will be appropriate for every project. In particular, some are more relevant to pure mathematical projects, and others to applied ones.

- Does the algorithm or method always work? Have you tested it?
- What is the theoretical running time, or complexity, of the algorithm? Note that this should be measured by the number of simple operations required, expressed in the usual $O\big(f(n)\big)$ or $\Omega\big(f(n)\big)$ notation, where n is some reasonable measure of the size of the input (say the number of vertices of a graph) and f is a reasonably simple function. Examples of simple operations are the addition or multiplication of two numbers, or the checking of the (p, q) entry of a matrix to see if it is non-zero; with this definition finding the scalar product of two vectors of length n takes order n operations. Note that this measure of complexity can differ from the number of MATLAB commands/'operations', e.g. there is a single MATLAB command to find a scalar product of two vectors of length n.
- What is the accuracy of the numerical method? Is it particularly appropriate for the problem in question and, if so, why? How did you choose the step-size (if relevant), and how did you confirm that your numerical results are reliably accurate for all calculations performed?
- How do the numerical answers you obtain relate to the mathematical or physical system being modelled? What conjectures or conclusions, if any, can you make from your results about the physical system or abstract mathematical object under consideration?

In summary, it is the candidate's responsibility to determine which points require discussion in the report, to address these points fully but concisely, and to structure the whole so as to present a clear and complete response to the project. It should be possible to read your write-up without reference to the listing of your programs.

As an aid, for the **two core projects** only, some brief additional comments are provided giving further guidance as to the form and approximate length of answer expected for each question. These also contain a mark-scheme, on which your marks for each question will be written and returned to you during the Lent Term. For the additional projects you are expected to use your judgement on the marks allocation.

2.2.1 Project write-ups: advice on length

The word *brief* peppers the last few paragraphs. To emphasise this point, in general **six sides of A4 of text**, *excluding in-line graphs, tables, etc.*, should be plenty for a clear concise report. Indeed, the best reports are sometimes shorter than this.

To this total you will of course need to add tables, graphs etc. However, *do **not** include every single piece of output you generate*: include a selection of the output that is a *representative* sample of graphs and tables. It is up to you to choose a selection which demonstrates all the important features but is reasonably concise. Presenting mathematical results in a clear and concise way is an important skill and one that you will be evaluated upon in CATAM. Twenty pages of graphs would be excessive for most projects, even if the graphs were one to a page.[8] Remember that the assessors will be allowed to **deduct** up to 10% of marks for any project containing an excessive quantity of irrelevant material. Typically, such a project might be long-winded, be very poorly structured, or contain long sections of prose that are not pertinent.

[8] Recall that graphs should not as a rule be printed one to a page.

Moreover, if your answer to the question posed is buried within a lot of irrelevant material then it may not receive credit, even if it is correct.

2.3 Project write-ups: technicalities

As emphasised above, elaborate write-ups are not required. You are required to submit your project reports electronically. In particular, you will be asked to submit your write-ups electronically in Portable Document Format (PDF) form, and you should ensure that the submitted file can be printed (in portrait mode on standard A4 paper). Note that many word processors (e.g. LaTeX, *Microsoft Word*, *LibreOffice*) will generate output in PDF form. In addition, there are utility programs to convert output from one form to another, in particular to PDF form (e.g. there are programs that will convert plain text to PDF). Before you make your choice of word processor, you should confirm that you will be able to generate submittable output in PDF form. Please note that a PDF file including pages generated by scanning a hand-written report or other text document is ***not*** acceptable.

In a very few projects, where a *sketch* (or similar) is asked for, a scanned hand-drawing is acceptable. Such exceptions will be noted *explicitly* in the project description.

If it will prove difficult for you to produce electronic write-ups, e.g. because of a disability, then please contact the *CATAM Helpline* as early as possible in the academic year, so that reasonable adjustments can be made for you.

Choice of Word Processor. As to the choice of word processor, there is no definitive answer. Many mathematicians use LaTeX (or, if they are of an older generation, TeX), e.g. this document is written in LaTeX. However, please note that although LaTeX is well suited for mathematical typesetting, it is absolutely acceptable to write reports using other word-processing software, e.g. *Microsoft Word* or *LibreOffice*.

- *Microsoft Word* is commercial, but is available free while you are a student at Cambridge: see

  `https://help.uis.cam.ac.uk/service/collaboration/office365`.

- *LibreOffice* can be installed for free for, *inter alia*, the Windows, MacOS and Linux operating systems from

  `http://www.libreoffice.org/download/download/`.

LaTeX. If you decide to use LaTeX, you will probably want to install it on your own personal computer. This can be done for free. For recommendations of TeX distributions and associated packages see

- `http://www.tug.org/begin.html` and
- `http://www.tug.org/interest.html`.

Front end. In addition to a TeX distribution you will also need a front-end (i.e. a 'clever editor'). A comparison of TeX editors is available on WIKIPEDIA; below we list a few of the more popular TeX editors.

TeXstudio. For Windows, Mac and Linux users, there is TeXstudio. The proTeXt distribution, based on MiKTeX, includes the TeXstudio front end.

TeXworks. Again for Windows, Mac and Linux users, there is TeXworks. The MiKTeX distribution includes TeXworks.

TEXShop. Many Mac aficionados use TEXShop. To obtain TEXShop and the TEXLive distribution see `http://pages.uoregon.edu/koch/texshop/obtaining.html`.

TEXnicCenter. TEXnicCenter is another [older] front end for Windows users.

LyX. LyX is not strictly a front end, but has been recommended by some previous students. LyX is available from

`http://www.lyx.org/`.

However, note that LyX uses its own internal file format, which it converts to LATEX as necessary.

Learning LATEX. A *Brief LATEX Guide for CATAM* is available for download from

`http://www.maths.cam.ac.uk/undergrad/catam/files/Brief-Guide.pdf` .

- The LATEX source file (which may be helpful as a template), and supporting files, are available for download as a zip file from

  `http://www.maths.cam.ac.uk/undergrad/catam/files/Guide.zip` .

  Mac, Unix and most Windows users should already have an unzip utility. Windows users can download 7-Zip if they have not.

Other sources of help. A welter of useful links have been collated by the Engineering Department on their *Text Processing using LATEX* page; see

`http://www.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html`.

Layout of the first page. The first page of your report should include the **project name** and **project number**.

Your script is marked anonymously. Hence, your **name or user identifier should not appear anywhere** in the write-up (including any output).

Further technicalities. Please do not use red or green for text (although red and/or green lines on plots are acceptable). Please leave a margin at least 2 cm wide at the left, and number each page, table and graph.

Program listings. At the end of each report you should include complete **listings** (i.e. printout of source code) of every major program used to generate your results. You do *not* need to include a listing of a program which is essentially a minor revision of another which you have already included. Make sure that your program listings are the *very last* thing in your reports. Please do not mix program output and program listings together; if you do, the program output may not be marked as part of the report.

3 Computing Facilities

You may write and run your programs on any computer you wish, whether it belongs to you personally, to your College, or to the University.

When permitted by COVID protocols, you can also use other computing facilities around the University; for further information (including which Colleges are linked to the MCS network) see[9]

[9] Note that the Phoenix Teaching Rooms and the Titan Room are used during term-times for practical classes by other Departments, but a list of these classes is posted at each site at the start of each term so that you can check the availability in advance (see Opening Hours).

https://help.uis.cam.ac.uk/service/desktop-services/mcs/mcs-sites

At most MCS locations you can access the MATLAB software and any files you store on the MCS from one location should be accessible from any other MCS location.

If you believe that do not have access to an adequate computer to complete the CATAM projects, you should contact your Director of Studies and/or the CATAM helpline *well in advance* of any project deadlines.

3.1 Backups

Whatever computing facilities you use, **make sure you make regular (electronic and paper) backups of your work** in case of disaster! Remember that occasionally systems go down or disks crash or computers are stolen. **Malfunctions of your own equipment or the MCS are not an excuse for late submissions**: leave yourself enough time before the deadline.

Possibly one of the easiest ways to ensure that your work is backed up is to use an online 'cloud' service; many of these services offer some free space. WIKIPEDIA has a fairly comprehensive list at http://en.wikipedia.org/wiki/Comparison_of_online_backup_services. In particular note that eligible students have 5TB of OneDrive personal storage space via their University Microsoft account under a University agreement and unlimited storage via Google Drive (see https://help.uis.cam.ac.uk/individual-storage).

4 Information Sources

There are many ways of getting help on matters relating to CATAM.

The CATAM Web Page. The CATAM web page,

http://www.maths.cam.ac.uk/undergrad/catam/

contains much useful information relating to CATAM. There are on-line, and up-to-date, copies of the projects, and any data files required by the projects can be downloaded. There is also the booklet *Learning to use MATLAB for CATAM project work*.

CATAM News and Email. Any important information about CATAM (e.g. corrections to projects or to other information in the *Manual*) is publicised via *CATAM News*, which can be reached from the CATAM web page. You must read *CATAM News* from time to time (e.g. just before starting a project) to check for these and other important announcements, such as submission dates and procedures.

As well as adding announcements to *CATAM News*, occasionally we will email students using the year lists maintained by the Faculty of Mathematics. You have a responsibility to read email from the Faculty, and if we send an email to one of those lists we will assume that you have read it.

After 1 October 2024 you can check that you are on the appropriate Faculty year list by referring to the https://lists.cam.ac.uk/mailman/raven webpage (to view this page you will need to authenticate using Raven if you have not already done so). You should check that the *Maths-IB* mailing list is one of your current lists.

If you are not subscribed to the correct mailing list, then this can be corrected by contacting the Faculty Undergraduate Office (email: undergrad-office@maths.cam.ac.uk) with a request to be subscribed to the correct list (and, if necessary, unsubscribed from the wrong list).

The CATAM Helpline. If you need help (e.g. if you need clarification about the wording of a project, or if you have queries about programming and/or MATLAB), you can email a query to the *CATAM Helpline*: catam@maths.cam.ac.uk. Almost all queries may be sent to the *Helpline*, and it is particularly useful to report potential errors in projects. However the *Helpline* cannot answer detailed mathematical questions about particular projects. Indeed if your query directly addresses a question in a project you may receive a standard reply indicating that the *Helpline* cannot add anything more.

In order to help us manage the emails that we receive,

- please use an email address ending in cam.ac.uk (rather than a Gmail, etc. address) both so that we may identify you and also so that your email is not identified as spam;
- please specify, in the subject line of your email, 'Part IB' as well as the project number and title or other topic, such as 'MATLAB query', to which your email relates;
- please also **restrict each email to one question or comment** (use multiple emails if you have more than one question or comment).

The *Helpline* is available during Full Term and one week either side. Queries sent outside these dates will be answered subject to personnel availability. We will endeavour (but do not guarantee) to provide a response from the *Helpline* within three working days. However, if the query has to be referred to an assessor, then it may take longer to receive a reply. Please do not send emails to any other address.

The CATAM FAQ Web Pages. Before asking the *Helpline* about a particular project, please check the *CATAM FAQ web pages* (accessible from the main CATAM web page). These list questions which students regularly ask, and you may find that your query has already been addressed.

Advice from Supervisors and Directors of Studies. The general rule is that advice **must be general in nature**. You should not have supervisions on any work that is yet to be submitted for examination; however, you may have a supervision on the *Introductory Project*, and/or another non-examinable project, and/or any work set by your Director of Studies. A supervisor can also provide feedback on the *Core Projects* **after** they have been submitted (e.g. **after** your marks have been returned). To that end the Faculty will aim to offer you a short "feedback" session on one of the *Core Projects* towards the end of the Lent Term.

5 Unfair Means, Plagiarism and Guidelines for Collaboration

The objective of CATAM is for you to learn computational methods, mathematics and written presentation skills. To achieve these objectives, you must work independently on the projects, both on the programming and on the write-ups.

The work that you turn in **must** be your own. This applies equally to the source code and the write-ups, i.e. you must write and test all programs yourself, and all reports must be written *independently.*

Any attempt to gain an unfair advantage, for example by copying computer code, mathematics, or written text, is not acceptable and will be subject to serious sanctions.

If you have any questions about what constitutes unfair means, you should seek advice from the CATAM helpline.

Citations. It is, of course, perfectly permissible to use reference books, journals, reference articles on the WWW or other similar material: indeed, you are encouraged to do this. You may quote directly from reference works so long as you acknowledge the source (WWW pages should be acknowledged by a *full* URL). There is no need to quote lengthy proofs in full, but you should at least include your own brief summary of the material, together with a *full* reference (including, if appropriate, the page number) of the proof.

Programs. You must write your own computer programs. Downloading computer code, e.g. from the internet, that you are asked to write yourself counts as plagiarism even if cited.

Acceptable collaboration. It is recognised that some candidates may occasionally wish to discuss their work with others doing similar projects. This can be educationally beneficial and is accepted provided that it remains within reasonable bounds. Acceptable collaboration may include an *occasional general discussion* of the approach to a project and of the numerical algorithms needed to solve it. Small hints on debugging code (note the *small*), as might be provided by an adviser, are also acceptable.

Unacceptable collaboration (also known as collusion). If a general discussion *either* is happening regularly *or* gets to the point where physical or virtual notes are being exchanged (even on the back of an envelope, napkin or stamp), then it has reached the stage of unacceptable collaboration. As an example to clarify the limits of 'acceptable collaboration', if an assessor reading two anonymous write-ups were to see significant similarities in results, answers, mathematical approach or programming which would clearly not be expected from students working independently, then there would appear to be a case that the students have breached the limits. An *Investigative Meeting* would then be arranged (unless such similarities were deemed to be justified in light of the declared lists of discussions, see below). If you are uncertain about what constitutes an *unacceptable collaboration* you should seek advice from the CATAM Helpline.

Generative AI. Using generative AI (e.g. ChatGPT, Bing, Bard and similar) to produce some or part of the submitted write-up or source code would not be original work and hence is considered a form of academic misconduct. This interpretation is consistent with *University guidelines*. We use software that is capable of detecting AI-generated content, and where a case of unfair means is suspected, the Examiners may, at their discretion, examine a candidate by means of an Oral Examination.

The following actions are examples of *unfair means*

- copying any other person's program, either automatically or by typing it in from a listing;
- using someone else's program or any part of it as a model, or working from a jointly produced detailed program outline;
- copying or paraphrasing of someone else's report in whole or in part;
- turning in output from a generative AI either in the report or in the source code.

These comments apply just as much to copying from the work of previous Part IB students, or another third party (including any code, etc. you find on the internet), as they do to copying from the work of students in your own year. Asking anyone for help that goes past the limits of *acceptable collaboration* as outlined above, and this includes posting questions on the internet (e.g. StackExchange), constitutes *unfair means.*

Further, you should not allow any present or future Part IB student access to the work you have undertaken for your own CATAM projects, even after you have submitted your write-ups. If you knowingly give another student access to your CATAM work you are in breach of these guidelines and may be charged with assisting another candidate to make use of unfair means.

5.1 Further information about policies regarding plagiarism and other forms of unfair means

University-wide Statement on Plagiarism. You should familiarise yourself with the University's *Statement on Plagiarism*.

There is a link to this statement from the University's *Good academic practice and plagiarism* website

http://www.plagiarism.admin.cam.ac.uk/,

which also features links to other useful resources, information and guidance.

Faculty Guidelines on Plagiarism. You should also be familiar with the Faculty of Mathematics Guidelines on Plagiarism. These guidelines, which include advice on quoting, paraphrasing, referencing, general indebtedness, and the use of web sources, are posted on the Faculty's website at

http://www.maths.cam.ac.uk/facultyboard/plagiarism/.

In order to preserve the academic integrity of the Computational Projects component of the Mathematical Tripos, the following procedures have been adopted.

Declarations. To certify that you have *read and understood* these guidelines, you will be asked to sign an electronic declaration. Further instructions will be given during Michaelmas Term.

In order to certify that you have *observed* these guidelines, you will be required to sign an electronic submission form provided when you submit your write-ups, and you are advised to read it carefully; it will be similar to that reproduced (subject to revision) as Appendix A. You must list on the form *anybody* (students, supervisors and Directors of Studies alike) with whom you have exchanged information (e.g. by talking to them, or by electronic means) about the projects at any more than a *trivial* level: *any* discussions that affected your approach to the projects to *any* extent must be listed. Failure to include on your submission form any discussion you may have had *is* a breach of these guidelines.

However, declared exchanges are perfectly allowable so long as they fall within the limits of 'acceptable collaboration' as defined above, and you should feel no qualms about listing them. For instance, as long as you have refrained from discussing in any detail your programs or write-ups with others after starting work on them, then the limits have probably not been breached.

The assessors will not have knowledge of your declaration until after all your projects have been marked. However, your declaration may affect your CATAM marks if the assessors believe that discussions have gone beyond the limits of what is acceptable. If so, or if there is a suspicion that your have breached any of the other guidelines, you will be summoned to an *Investigative Meeting* (see §5.2). Ultimately, your case could be brought to the University courts and serious penalties could result (see *Sanctions* below).

Plagiarism detection. **The programs and reports submitted will be checked carefully both to ensure that they are your own work, and to ensure the results that you hand in have been produced by your own programs.**

Checks on submitted program code. The Faculty of Mathematics uses (and has used for many years) specialised software, including that of external service providers, which automatically checks whether your programs either have been copied or have unacceptable overlaps (e.g. the software can spot changes of notation). All programs submitted are screened.

The code that you submit, and the code that your predecessors submitted, is kept in *anonymised* form to check against code submitted in subsequent years.

Checks on electronically submitted reports. In addition, the Faculty of Mathematics will screen your electronically submitted reports using the *Turnitin UK* text-matching software. Further information will be sent to you before the submission date. The electronic declaration which you will be asked to complete at the start of the Michaelmas term will, *inter alia*, cover the use of *Turnitin UK*.

Your electronically submitted write-ups will be kept in *anonymised* form to check against write-ups submitted in subsequent years.

Sanctions. If plagiarism, collusion or any other method of unfair means is suspected in the Computational Projects, normally the Chair of Examiners will convene an *Investigative Meeting* (see §5.2). If the Chair of Examiners deems that unfair means were used, the case may be brought to the University courts. According to the Statues and Ordinances of the University [10]

> suspected cases of the use of unfair means (of which plagiarism is one form) will be investigated and may be brought to one of the University courts or disciplinary panels. The University courts and disciplinary panels have wide powers to discipline those found to have used unfair means in an examination, including depriving such persons of membership of the University, and deprivation of a degree.

The Faculty of Mathematics wishes to make it clear that any breach of these guidelines will be treated very seriously.

However, we also wish to emphasise that the great majority of candidates have, in the past, had no difficulty in keeping to these guidelines. Unfortunately there have been a small number of cases in recent years where *some individuals have been penalised by the loss of significant numbers of marks, indeed sufficient to drop a class.* If you find the guidelines unclear in any way you should seek advice from the *CATAM Helpline*. These policies and practices have been put in to place so that you can be sure that the hard work you put into CATAM will be fairly rewarded.

5.2 Oral examinations

Viva Voce Examinations. A number of candidates may be selected, either randomly or formulaically, for a *Viva Voce Examination* after submission of either the core or the additional

[10] From `https://www.admin.cam.ac.uk/univ/so/`.

projects. This is a matter of routine, and therefore a summons to a *Viva Voce Examination* should not be taken to indicate that there is anything amiss. You will be asked some straightforward questions on your project work, and may be asked to elaborate on the extent of discussions you may have had with other students. So long as you can demonstrate that your write-ups are indeed your own, your answers will not alter your project marks.

Examination Interviews. Additionally, the Chair of Examiners may summon a particular candidate or particular candidates for interview on any aspect of the written work of the candidate or candidates not produced in an examination room which in the opinion of the Examiners requires elucidation. If plagiarism or other unfair means is suspected, an Investigative Meeting will be convened (see below).

Investigative Meetings. When plagiarism, collusion or other unfair means are suspected the Chair of Examiners may summon a candidate to an *Investigative Meeting*[11]. If this happens, you have the right to be accompanied by your Tutor (or another representative at your request). The reasons for the meeting, together with copies of supporting evidence and other relevant documentation, will be given to your Tutor (or other representative). One possible outcome is that the case is brought to the University courts where *serious penalties can be imposed* (see *Sanctions* above).

Timing. Viva Voce Examinations, *Examination Interviews* and *Investigative Meetings* are a formal part of the Tripos examination, and if you are summoned then you must attend. In the case of the core projects these will usually take place during Lent Full Term (although they may take place exceptionally during Easter Full Term), and in the case of the additional projects these will usually take place during the last week of Easter Full Term. Easter Term *Viva Voce Examinations* are likely to take place on the Monday of the last week (i.e. Monday 16th June 2025), while *Examination Interviews* and *Investigative Meetings* may take place any time that week. If you need to attend during the last week of Easter Full Term you will be informed in writing just after the end of the written examinations. **You must be available** in the last week of Easter Full Term in case you are summoned.

6 Submission and Assessment

In order to gain examination credit for the work that you do on this course, you must write reports on each of the projects that you have done. As emphasised earlier it is the quality (not quantity) of your written report which is the most important factor in determining the marks that you will be awarded.

6.1 Submission form

When you submit your project reports you will be required to complete and upload the submission form provided, detailing which projects you have attempted and listing all discussions you have had concerning CATAM (see §5, *Unfair Means, Plagiarism and Guidelines for Collaboration*, and Appendix A). Further details, including the definitive submission form, will be made available when the arrangements for electronic submission of reports and programs (see below) are announced.

[11] For more information see
`https://www.plagiarism.admin.cam.ac.uk/files/investigative_2016.pdf`.

6.2 Submission of written work

In order to gain examination credit, you must:

- submit electronic copies of your reports and programs (see §6.3);
- complete and submit your submission form listing each project for which you wish to gain credit.

Further details about submission arrangements will be announced via *CATAM News* and email closer to the time.

For the core projects the submission deadline is

Wednesday 22st January 2025, 4pm,

while for the additional projects the submission deadline is

Wednesday 30th April 2025, 4pm.

Self-certified extensions may be obtained for a period of up to 7 days. A form to apply for an extension will be made available on Moodle at the start of the submission period. Students must inform their college Tutor that they are applying for an extension *before* filling out the extension request form. Whenever possible, students should apply for an extension before the original submission deadline. It will not be possible to apply for a self-certified extension later than 7 days after the original submission deadline. We strongly encourage all students to complete and turn-in their work well in advance of the original submission deadline to allow time to deal with any issues arising during submission and avoid impacts on other coursework and revision.

The Computational Projects Assessors Committee reserves the right to **reduce** the marks awarded for any projects (including reports and source code) which are submitted late (either the standard or extended deadline, as appropriate).

6.3 Electronic submission

You will be required to submit electronically copies of both your reports and your program source files. Electronic submission enables the Faculty to run automatic checks on the independence of your work, and also allows your programs to be inspected in depth (and if necessary run) by the assessors.

As regards your programs, electronic submission applies whether you have done your work on your own computer, on the MCS, or elsewhere, and is regardless of which programming language you have chosen.

Details of the procedure will be given in advance of the submission deadlines via *CATAM News* and email.

However please note that you will need to know your UIS password in order to submit copies of your report and program source files.

If you cannot remember your UIS password you will need to follow that instructions provided by the University Information Service.[12] Note that if you need a Password Reset Token then this may take some time to obtain, so check that you know your UIS password well before submission day.

[12] See `https://password.csx.cam.ac.uk/forgotten-passwd`.

6.4 Saving and sharing electronic files

After the submission deadline the electronic files will be taken offline and you will not be able to download your submitted work from the submission site. We recommend that you keep electronic copies of your work.

A copy of your submission is likely to be particularly useful for the core projects for which you will be given a breakdown of the marks you have obtained. Since the manuals will be taken off-line after the close of submission, you might also like to save a copy of the projects you have attempted.

It is critical that you do not make your reports or source code available to any present or future students. This includes posting to publically accessible repositories such as github.

Please note that all material that you submit electronically is kept in *anonymised* form to check against write-ups and program code submitted in subsequent years.

6.5 Returning from intermission

If a student is returning from intermission that began in an academic year during which they submitted some or all of the CATAM projects, then in certain circumstances it is possible to carry forward some or all of their CATAM marks from that year. Action is required by the Director of Studies. Hence, before attempting any further CATAM work, the student should discuss the options available with their Director of Studies and decide on their intended strategy.

The following general policies have been approved by the Faculty Board. If there are exceptional circumstances in which these seem inappropriate, the Director of Studies should discuss these with the CATAM Director: catam-director@maths.cam.ac.uk.

If a Part IB student intermits, and then repeats the entire year starting in Michaelmas Term, they should normally be expected to start CATAM afresh as a logical part of repeating the year.

If the student returns in the Lent Term, then they may EITHER carry forward any marks earned on the core projects in a previous year OR instead submit work for the current year's core projects. The student (following discussion with the DoS) must choose which option to take before the Lent Term submission date, and submission of new work will be taken to mean they have (irrevocably) taken the latter option. If they opt, instead, to carry their marks forward, then their Director of Studies must notify the Undergraduate Office and the CATAM Director that this should be done. The student would normally be expected to complete the current year's additional projects as a logical part of repeating the Lent Term.

If the student returns in Easter Term, then any marks earned on the core projects in a previous year should be carried forward. The student may EITHER carry forward any marks earned on the additional projects in a previous year OR instead submit work for the current year's additional projects. They must choose which option to take before the Easter Term submission date, and submission of new work will be taken to mean they have (irrevocably) taken the latter option. The Director of Studies must notify the Undergraduate Office and the CATAM Director about any marks that are to be carried forward.

A Appendix: Example Submission Form

PART IB MATHEMATICAL TRIPOS 2024-25

Computational Projects 2024

CORE COMPUTATIONAL PROJECTS

STATEMENT OF PROJECTS SUBMITTED FOR EXAMINATION CREDIT

Please observe these points when submitting your CATAM projects:

1. Your name, College or CRSid User Identifier **must not** appear anywhere in the submitted work.
2. Complete this declaration form and submit it electronically with your reports.
3. The Moodle submission site will close at 4pm on submission day and it is likely to be slow immediately prior to the deadline. Please turn in your work earlier if possible and be prepared for delays in the website on submission day.

IMPORTANT

Candidates are reminded that Discipline Regulation 7 reads:

> No candidate shall make use of unfair means in any University examination. Unfair means shall include plagiarism[13] and, unless such possession is specifically authorized, the possession of any book, paper or other material relevant to the examination. No member of the University shall assist a candidate to make use of such unfair means.

To confirm that you are aware of this, you **must** check and sign the declaration below and include it with your work when it is submitted for credit.

> *The Faculty of Mathematics wishes to make it clear that failure to comply with this requirement is a serious matter that could render you liable to sanctions imposed by the University courts.*

[13] Plagiarism is defined as submitting as one's own work, irrespective of intent to deceive, that which derives in part or in its entirety from the work of others without due acknowledgement.

DECLARATION BY CANDIDATE

I hereby submit my reports on the following projects and wish them to be assessed for examination credit:

Project Number	Brief Title

I certify that I have read and understood the section *Unfair Means, Plagiarism and Guidelines for Collaboration* in the Projects Manual (including the references therein), and that I have conformed with the guidelines given there as regards any work submitted for assessment at the University. I understand that the penalties may be severe if I am found to have not kept to the guidelines in the section *Unfair Means, Plagiarism and Guidelines for Collaboration.* I agree to the Faculty of Mathematics using specialised software, including *Turnitin UK*, to automatically check whether my submitted work has been copied or plagiarised and, in particular, I certify that

- the composing and writing of these project reports is my own unaided work and no part of it is a copy or paraphrase of work of anyone other than myself;

- the computer programs and listings and results were not copied from anyone or from anywhere (apart from the course material provided);

- I have not shown my programs or written work to any other candidate or allowed anyone else to have access to them;

- I have listed below anybody, other than the CATAM Helpline or CATAM advisers, with whom I have had discussions or exchanged information at any more than a trivial level about the CATAM projects, together with the nature of those discussions and/or exchanges.

Declaration of Discussions and Exchanges (continue on a separate sheet if necessary)

Signed .... Date ....

0.1 Root Finding in One Dimension

This is an optional, introductory, non-examinable project. Unlike the other projects ***there are no marks awarded for it****. Also, unlike the other projects, you may collaborate as much as you like, and (if your College is willing) have a supervision on the project. A model answer will be provided on the CATAM web site towards the start of the Michaelmas Term.*

The Methods

The aim of this project is to study iteration methods for the numerical solution of an algebraic or transcendental equation $F(x) = 0$. We consider two methods.

(i) *Binary search* (also known as bisection or interval halving).

(ii) *Fixed-point iteration*, which involves solving an equivalent system $x = f(x)$ by use of an iteration scheme

$$x_N = f\left(x_{N-1}\right) , \tag{1}$$

with a suitable initial guess x_0. We will consider two cases of fixed-point iteration:

(a) first, we will study an equivalent system derived by manipulating $F(x) = 0$ algebraically to the (non-unique) form $x = f(x)$;

(b) second, we will study *Newton-Raphson iteration*, which uses the scheme

$$x_N = x_{N-1} - \frac{F\left(x_{N-1}\right)}{F'\left(x_{N-1}\right)} . \tag{2}$$

The theoretical background to these methods is covered in most textbooks on *Numerical Analysis* (a few of which are listed at the end of this project).

Order of Convergence

A sequence $\{\delta_N\}$ which converges to zero as $N \to \infty$ is said, for the purposes of this project,[1] to have order of convergence p ($\geqslant 1$) if

$$|\delta_N| \sim C|\delta_{N-1}|^p \text{ as } N \to \infty \; , \quad \text{i.e.} \quad \lim_{N\to\infty} \frac{|\delta_N|}{|\delta_{N-1}|^p} = C \, , \tag{3}$$

where C is some strictly positive (finite) constant; first-order (or 'linear') convergence, $p = 1$, requires $C < 1$.

If an iteration method is attempting to approximate the exact root x_*, the *truncation error* in the N^{th} iterate is defined as $\epsilon_N = x_N - x_*$.[2] If the method is convergent, i.e. $\epsilon_N \to 0$ as $N \to \infty$, it is said to be p^{th}-order convergent if either the sequence $\{\epsilon_N\}$ has property (3), or there exists a sequence $\{y_n\}$ with property (3) such that $|\epsilon_n| < |y_n|$ for all n. For the two methods of interest the following is known.

[1] A more inclusive definition of order of convergence, referred to as the Q-order of convergence, might be

$$p = \sup\left\{ q : \limsup_{n\to\infty} \frac{|\delta_N|}{|\delta_{N-1}|^q} = 0 \right\} .$$

[2] This quantity should be evaluated on the assumption that numbers are represented to infinite precision, without rounding error.

(i) *Binary search* is first-order convergent.

(ii) *Fixed-point iteration*, when convergent, is *in general* first-order convergent for a simple root, i.e. one with $F'(x_*) \neq 0$. However, Newton-Raphson iteration, when convergent, is second-order convergent for a simple root, but only first-order convergent for a multiple root

Examples

The cases to be studied as examples are

$$F(x) \equiv 2x - 3\sin x + 5 = 0\,, \tag{4}$$

and

$$F(x) \equiv x^3 - 8.5x^2 + 20x - 8 = 0\,. \tag{5a}$$

Note that equation (5a) can be factorised and rewritten as

$$F(x) \equiv \left(x - \tfrac{1}{2}\right)(x-4)^2 = 0\,. \tag{5b}$$

Question 1 Show, with the help of a graph, that equation (4) has exactly one root (which is in fact $-2.88323687\ldots$).

Binary Search

Programming Task: write a program to solve equation (4) by binary search.Provide for termination of the iteration as soon as the truncation error is guaranteed to be less than 0.5×10^{-5}, and print out the number of iterations, N, as well as the estimate of the root. Run the program for a number of suitable starting values to check that it is working; include some of these results in your report.

Question 2 Suppose that the rounding error in evaluating $F(x)$ in equation (4) is at most δ for $|x| < \pi$. By considering a Taylor expansion of $F(x)$ near x_*, or otherwise, estimate the accuracy that may be expected for the calculated value of the root.

Hint: note that $|F'(x)| > 4$ for $-5\pi/4 < x < -3\pi/4$.

Fixed-Point Iteration

There are many possible choices of f, e.g.

$$f(x) = x - h(F(x))\,, \tag{6}$$

for some function[3] $h(F)$ such that $h(0) = 0$.

Programming Task: write a program to implement the iteration scheme in equation (1) for general f. Provide for termination of the process as soon as $|x_N - x_{N-1}| < \epsilon$ or when $N = N_{max}$, whichever occurs first. Print out the values of N and x_N for each N, so that you can watch the progress of the iteration.

[3] Or *functional.*

Question 3 Use the program to solve (4) by fixed-point iteration by taking

$$h(F) = \frac{F}{2+k} \tag{7a}$$

in (6), so that

$$f(x) = \frac{3\sin x + kx - 5}{2+k}\,, \tag{7b}$$

for some constant k.

(i) First run the program with $k = 0$, $\epsilon = 10^{-5}$, $x_0 = -2$, $N_{max} = 10$. Plot $y = f(x)$ and $y = x$ on the same graph, and use these plots to show why convergence should not occur. Explain the divergence by identifying a theoretical criterion that has been violated.[4]

(ii) Determine the values of k for which convergence is guaranteed if x_N remains in the range $(-\pi, -\pi/2)$.

(iii) Choose, giving reasons, a value of k for which *monotonic* convergence should occur near the root, and also a value for which *oscillatory* convergence should occur near the root. Verify that these two values of k give the expected behaviour, by running the program with $N_{max} = 20$.

(iv) Also run the case $k = 16$. This should converge only slowly, so set $N_{max} = 50$. Discuss whether the truncation error is expected to be less than 10^{-5} in this case?

(v) Discuss whether your results are consistent with first-order convergence.

Question 4 Now use your program to find the *double root* of equation (5a) by fixed-point iteration by taking

$$h(F) = \tfrac{1}{20}\,F\,, \tag{8a}$$

in (6), so that

$$f(x) = \tfrac{1}{20}(-x^3 + 8.5x^2 + 8)\,. \tag{8b}$$

By considering $f'(x_*)$ explain why convergence will be slow at a multiple root for any choice of differentiable function h in (6).

In your calculations some care may be needed over the choice of x_0. Also,

(a) since convergence will be slow, take $N_{max} = 1000$;

(b) suppress the printing of each iterate, but print out the *final* values of N and x_N.

Is this an example of first-order convergence? Does the termination criterion ensure a truncation error of less than 10^{-5}?

Note: it can be shown that the truncation error ϵ_N is asymptotic to $40/(7N)$ as $N \to \infty$.

Newton-Raphson Iteration

A refinement of (6) is to let h depend on the derivatives of F, i.e.

$$f(x) = x - h(F, F', F'', \ldots)\,. \tag{9a}$$

In Newton-Raphson iteration

$$h = \frac{F}{F'}\,. \tag{9b}$$

[4] The references at the end may prove helpful.

Programming Task: modify your program to recalculate the root of equation (4), and the double root of equation (5a), using Newton-Raphson iteration.

Question 5 For equation (4), experiment with various x_0 until you have demonstrated a case that converges, and also a case that has not converged in 10 iterations. In the unconverged case, show graphically what happened in the first few iterations.

For both equation (4) and equation (5a) do your (converged) results bear out the theoretical orders of convergence? Comment on the effects of rounding error.

Hint: you may want to use a smaller value for ϵ.

References

[1] Epperson, J.F., *An Introduction to Numerical Methods and Analysis*, John Wiley & Sons (2007). ISBN-13: 9780470049631.

[2] Kharab, A. and Guenther, R.B., *An Introduction to Numerical Methods: A MATLAB Approach*, Second Edition, CRC Press (2005). ISBN-13: 9781584885573

[3] Press, W.H., Teukolsky, S.A. and Vetterling, W.T. and Flannery, B.P., *Numerical Recipes: The Art of Scientific Computing*, Third Edition, Cambridge University Press (2007). ISBN-10: 0521880688.

[4] Süli, E. and Mayers, D., *An Introduction to Numerical Analysis*, Cambridge University Press (2003). ISBN-10: 0521810264 (hardback), ISBN-10: 0521007941 (paperback).

1.1 Matrices over Finite Fields

This project is about the elementary properties of vector spaces, which are introduced in the Part IA course Vectors and Matrices and are considered more generally in the Part IB course Linear Algebra.

1 Fields of Prime Order

We shall be considering algorithms for computing algebraic invariants attached to vector spaces and linear maps over a field F. In applications the field is often that of the real or complex numbers. In this project F will be $\mathrm{GF}(p)$, the finite field of p elements, represented by the integers modulo p for some prime p.

In the examples you will work with, p will be small (at most 30), as will the matrices (at most 10×10). However, when answering questions about complexity for large p you should be thinking about how the program would behave for *very* large p (i.e., let p tend to infinity).

2 Division

It will be necessary to be able to divide modulo p; that is, for each a, $1 \leqslant a \leqslant p-1$, you will need to know its inverse a^{-1}, $1 \leqslant a^{-1} \leqslant p-1$, such that $aa^{-1} \equiv 1 \pmod{p}$. Rather than compute a^{-1} afresh each time it is needed, the inverses should be computed once and stored.

> **Question 1** Write a program to store the inverses of the non-zero elements of F in an array of length $p-1$. Find the inverses by testing, for each a in the range $1 \leqslant a \leqslant p-1$, all values of b in the range $1 \leqslant b \leqslant p-1$ until you find one which works and then store it. (Note that the MATLAB command `mod(a,p)` gives the value of a modulo p.) Describe any *very* simple modification to speed up this procedure (say by a factor of 2).

> **Question 2** Estimate the complexity of the procedure of Question 1 in terms of p.
>
> [That is, give a simple function $f(p)$ of p, such as $\sqrt{p}$, p or 2^p, such that the number of steps is $\Theta\big(f(p)\big)$, meaning of the order of magnitude of $f(p)$. To be exact, a function $g(p)$ is $\Theta\big(f(p)\big)$ if there are positive constants c and C such that $c \leqslant g(p)/f(p) \leqslant C$ for all sufficiently large p.
>
> You may assume that in a single step your ideal computer can perform any elementary operation such as to store a number, or to add, subtract, multiply, divide or compare two numbers.]

3 Gaussian Elimination

A non zero matrix $M = (m_{ij})$ over F, with m rows and n columns, is in *reduced row echelon form* if

- for some r, $1 \leqslant r \leqslant m$, the last $m-r$ rows have only zero entries;
- for each i, $1 \leqslant i \leqslant r$, there is a number $1 \leqslant l(i) \leqslant n$ such that $m_{ij} = 0$ for $j < l(i)$ and $m_{ij} = 1$ for $j = l(i)$;

- $l(1) < l(2) < \cdots < l(r)$;
- for each k, $2 \leqslant k \leqslant r$, we have $m_{ij} = 0$ when $j = l(k)$ and $i < k$.

Here is a 4×5 matrix which is in reduced row echelon form if we take the entries mod 7 but not if we take the entries mod 2; in the former case $r = 3$, $l(1) = 1$, $l(2) = 3$, $l(3) = 4$.

$$\begin{pmatrix} 1 & 4 & 0 & 7 & 5 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 15 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \pmod 7$$

The *rank* of a matrix is the dimension of its *row space*; that is, the vector space (over F) spanned by its rows. The rank of the matrix M above is r. The following operations on a matrix leave its row space unaltered:

- $T(i, j)$, transpose rows i and j
- $D(i, a)$, divide row i by the element $a \in F \setminus \{0\}$
- $S(i, a, j)$, subtract a times row $j \neq i$ from row i.

Gaussian elimination uses the operations T, D, S to convert a matrix into reduced row echelon form.

> **Question 3** Write a program to turn a matrix into reduced row echelon form using Gaussian elimination. From your output, compute the ranks of each of the following matrices, and give bases for their row spaces.

$$A_1 = \begin{pmatrix} 11 & 1 & 7 & 2 & 0 \\ 8 & 0 & 2 & 5 & 11 \\ 2 & 1 & 2 & 6 & 5 \\ 7 & 4 & 5 & 3 & 1 \end{pmatrix} \begin{array}{l}\text{(both mod 5}\\ \text{and mod 11),}\end{array} \qquad A_2 = \begin{pmatrix} 0 & 1 & 1 & 3 & 5 & 2 \\ 1 & 2 & 3 & 8 & 9 & 0 \\ 0 & 1 & 1 & 2 & 3 & 2 \\ 2 & 1 & 3 & 7 & 9 & 1 \\ 2 & 1 & 3 & 8 & 10 & 0 \end{pmatrix} \pmod{23}.$$

4 Kernels and Annihilators

Let A be an $m \times n$ matrix and $\mathbf{x} = (x_j)$ an $n \times 1$ column vector over F. The *kernel* of A, denoted $\ker A$, is the space of solutions to $A\mathbf{x} = 0$. A basis can be found by putting A in reduced row echelon form and then expressing $x_{l(1)}$, $x_{l(2)}$, …, $x_{l(r)}$ in terms of the other x_j.

> **Question 4** Write a program to compute a basis for the kernel of a matrix. Describe briefly how your algorithm works. Find bases for the kernels of the matrix A_1 modulo 5, modulo 7 and modulo 13.
>
> Now find bases for the kernels of the matrix A_2 modulo every prime below 30. Do you get the same result for every prime?

Let U be a subspace of the space of row vectors F^n. The annihilator U° consists of the set of column vectors $\mathbf{x}$ satisfying $\mathbf{u}\,\mathbf{x} = 0$ for every $\mathbf{u} \in U$. It is a subspace of the space of column vectors. Notice that if U is the row space of a matrix A, then U° is the kernel of A.

Question 5 State the relationship between the dimensions of U and U°.

If S is a subspace of the space of column vectors, then we make an analogous definition of S° as the space of row vectors $\mathbf{t}$ satisfying $\mathbf{t}\,\mathbf{s} = 0$ for every $\mathbf{s} \in S$. We have

$$(U^\circ)^\circ = U.$$

Question 6 Use your program from Question 4 to find U° where we work mod 23 and U is the row space of the matrix A_1. Similarly find $(U^\circ)^\circ$ and verify that it is equal to U.

For U and W subspaces of F^n it is known that

$$(U + W)^\circ = U^\circ \cap W^\circ$$

and

$$(U \cap W)^\circ = U^\circ + W^\circ.$$

Question 7 Write a program that, given matrices A and B with row spaces U and W, computes bases for U, W, $U + W$ and $U \cap W$. Explain briefly how your program works. Comment on the relationship between the dimensions of the four spaces computed. Run your program on the following examples:

- Modulo 19 with U the row space of

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 5 & 0 & 1 & 6 & 3 & 0 \\ 0 & 0 & 5 & 0 & 2 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 & 5 & 1 \\ 4 & 3 & 0 & 0 & 6 & 2 & 6 \end{pmatrix}$$

  and W the kernel of A_3. (Although the kernel of A_3 is naturally a space of column vectors, you should re-write it as a space of row vectors in the obvious way.)

- Again take U the row space of A_3 and W the kernel of A_3, but this time modulo 7.

Question 8 What feature of the very last part of Question 7 would be surprising to someone who carried out a similar project working over the real numbers instead of GF(p)?

Project 1.1: Matrices over Finite Fields

Marking Scheme and additional comments for the Project Report

The purpose of these additional comments is to provide guidance on the structure and length of your CATAM report. Use the same concepts to write the rest of the reports. To help you assess where marks have been lost, this marking scheme will be completed and returned to you during Lent Term. You are advised to keep a copy of your write-up in order to correlate your answers to the marks awarded.

Question no.	**marks available**[1]	**marks awarded**[2]
Question 1 Attach the printout of the program at the end of the report. Refer to the program at the start of your report. When you are asked to "write a program" without being given specific data to try, as in this case, you should include some very brief "test output" in your write-up to demonstrate that your program works (see the introduction to the projects manual). *Modification:* [*approx. 3 lines*][3]	C1+M1	
Question 2 Do not include negligible terms, or terms specific to your particular implementation, in your final answer. [*approx. 3 lines*][3]	C0+M1	
Question 3 Produce evidence of output from your program and state your answers clearly. There is no need to explain how Gaussian elimination works. [*approx. 10 lines*][3]	C2+M1	
Question 4 Produce evidence of output from your program and state your answers clearly. Explain briefly how your program works. [*approx. 15 lines*][3]	C2+M1	
Question 5 No proofs are required. [*approx. 1 line*][3]	C0+M1	
Question 6 Your answers should be supported by listing the input and output of each program used. Explain your method for checking that two subspaces are equal. [*approx. 10 lines*][3]	C1+M1	
Question 7 Give a brief discussion of the theory behind your solutions, explaining how you have used your earlier programs. There is no need to prove facts stated in the text. [*approx. 20 lines*][3]	C2+M3	
Question 8 No programming is required for this question. [*approx. 3 lines*][3]	C0+M1	
Excellence marks. These are awarded for, *among other things*, mathematical clarity and good, clear output (graphs and tables) — see the introduction to the Project Manual.	E2	
Total Raw Marks	20	
Total Tripos Marks	**40**	

[1] C#, M# and E#: *Computational*, *Mathematical* and *Excellence* marks respectively.

[2] For use by the assessor

[3] This figure is only meant to be indicative of the length of your answer, rather than the exact number of lines you are expected to write

1.2 Ordinary Differential Equations

This project builds on material covered in the Part IA lectures on Computational Projects, see http://www.maths.cam.ac.uk/undergrad/catam/part-ia-lectures. The Part IA Differential Equations and Part IB Methods courses are also relevant.

1 Background Theory

This project is concerned with the numerical step-by-step integration of ordinary differential equations (ODEs) of the form

$$\frac{d\mathbf{y}}{dx} = \mathbf{f}(x, \mathbf{y})\,, \tag{1a}$$

where $\mathbf{y}$ and $\mathbf{f}$ are vectors of length m, subject to an initial condition

$$\mathbf{y} = \mathbf{y}_0 \quad \text{at } x = x_0 \tag{1b}$$

for some constants x_0 and $\mathbf{y}_0$. The exact solution of (1a)–(1b) will be denoted by $\mathbf{y} = \mathbf{y}_e(x)$. In the first part of the project (§2) the performance of three different numerical methods will be examined. A first-order equation ($m = 1$) has been chosen for which $y_e(x)$ has a known analytic form for comparison. In the second part (§3) one of the methods is extended to solve a second-order problem.

The numerical methods to be investigated are as follows.

(a) The **Euler** method, or more precisely the **forward Euler** method, is the simple scheme

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + h\,\mathbf{f}(x_n, \mathbf{Y}_n)\,, \tag{2}$$

where $\mathbf{Y}_n$ denotes the numerical solution at $x_n \equiv x_0 + nh$, that is, at the nth step with step length h. The Euler method is called a *single-step* method since $\mathbf{Y}_{n+1}$ is determined solely by the value $\mathbf{Y}_n$ at the previous step.

Definition. The *global error* after the nth step is defined as

$$\mathbf{E}_n = \mathbf{Y}_n - \mathbf{y}_e(x_n)\,. \tag{3a}$$

Definition. The *local error* of the first step is defined as $\mathbf{e}_1 = \mathbf{Y}_1 - \mathbf{y}_e(x_1)$. For subsequent steps the *local error* is defined as

$$\mathbf{e}_n = \mathbf{Y}_n - \tilde{\mathbf{y}}(x_n)\,, \tag{3b}$$

where $\mathbf{y} = \tilde{\mathbf{y}}(x)$ is the exact solution to equation (1a) with $\mathbf{y} = \mathbf{Y}_{n-1}$ at $x = x_{n-1}$ (note that, in general, $\mathbf{Y}_{n-1} \neq \mathbf{y}_e(x_{n-1})$ for $n > 1$).

A numerical method is said to be p^{th}*-order accurate* if $\mathbf{e}_n$ is $O\left(h^{p+1}\right)$ as $h \to 0$. It can be shown that the Euler method is first-order accurate.

(b) The second-order-accurate **Adams-Bashforth** (AB2) method employs the scheme

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + h\left[\tfrac{3}{2}\mathbf{f}\left(x_n, \mathbf{Y}_n\right) - \tfrac{1}{2}\mathbf{f}\left(x_{n-1}, \mathbf{Y}_{n-1}\right)\right]\,. \tag{4}$$

This is a *two-step* method, using both $\mathbf{Y}_{n-1}$ and $\mathbf{Y}_n$ to obtain $\mathbf{Y}_{n+1}$, and the first step must be taken by a single-step method, such as the Euler method.

(c) The fourth-order-accurate **Runge–Kutta** (RK4) method employs the scheme:

$$\mathbf{Y}_{n+1} = \mathbf{Y}_n + \tfrac{1}{6}h\left[\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right] \,, \tag{5a}$$

where

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}(x_n,\ \mathbf{Y}_n)\,, && (5b)\\
\mathbf{k}_2 &= \mathbf{f}(x_n + \tfrac{1}{2}h,\ \mathbf{Y}_n + \tfrac{1}{2}h\mathbf{k}_1)\,, && (5c)\\
\mathbf{k}_3 &= \mathbf{f}(x_n + \tfrac{1}{2}h,\ \mathbf{Y}_n + \tfrac{1}{2}h\mathbf{k}_2)\,, && (5d)\\
\mathbf{k}_4 &= \mathbf{f}(x_n + h,\ \mathbf{Y}_n + h\mathbf{k}_3)\,. && (5e)
\end{aligned}$$

The theoretical background for the stability and accuracy of these methods is set out in, for example, *An Introduction to Numerical Methods and Analysis* by J.F.Epperson, *An Introduction to Numerical Methods* by A.Kharab and R.B.Guenther and *Numerical Recipes* by Press *et al.*

2 Stability and accuracy of the numerical methods

The example to be studied in detail in this section is the scalar version of equation (1a) with

$$f(x,y) = -8y + 6\mathrm{e}^{-2x} \tag{6a}$$

and initial condition

$$y = 0 \quad \text{at } x = 0.\ . \tag{6b}$$

This has the exact solution

$$y = y_e(x) \equiv \mathrm{e}^{-2x} - \mathrm{e}^{-8x}\ . \tag{7}$$

Programming Task: Write programs to apply each of the methods (a), (b) and (c) to this problem. For all the programming tasks in this project you should use 64-bit (8-byte) double-precision floating-point values.

2.1 Stability

This subsection considers the stability of the AB2 method (with first step by Euler).

Question 1 Starting with $Y_0 = 0$, compute Y_n for x up to 3 with $h = 0.5$ (i.e. for n up to $3/h = 6$). Tabulate the values of x_n, the numerical solution Y_n, the analytic solution $y_e(x_n)$ from (7) and the global error $E_n = Y_n - y_e(x_n)$. You should find that the numerical result is unstable: the error oscillates wildly and its magnitude ultimately grows proportional to $\mathrm{e}^{\gamma x}$, where the 'growth rate' γ is a positive constant which you should estimate.

Repeat with h = 0.375, 0.25, 0.125, 0.1 and 0.05 (you need only present a judicious selection of output to illustrate the behaviour). What effect does reducing h have on the instability and its growth rate?

Question 2
(i) Find the analytic solution of the AB2 *difference* equation

$$Y_{n+1} = Y_n + h\left[-12Y_n + 9\left(\mathrm{e}^{-2h}\right)^n + 4Y_{n-1} - 3\left(\mathrm{e}^{-2h}\right)^{n-1}\right] \tag{8}$$

with

$$Y_0 = 0 \ , \quad Y_1 = 6h \quad \text{(from the Euler method).} \tag{9}$$

(ii) Hence explain why and when instability occurs, and with what growth rate.

(iii) Show that in the limit $h \to 0$, $n \to \infty$ with $x_n \equiv nh$ fixed, the solution of the difference-equation problem (8)–(9) converges to the solution (7) of the differential-equation problem (1a), (6a), (6b).

How (if at all) would the conclusions in (ii) be altered if a more accurate method such as RK4 were used for the first step?

2.2 Accuracy

This subsection considers the accuracy of the three methods.

Question 3 For each of the three methods, integrate with $h = 0.08$ from $x = 0$ to $x = 2$. Tabulate and plot the numerical solution Y_n for each method against x_n, superposing a plot of the exact solution $y_e(x_n)$ given by equation (7).

Question 4 For each of the three methods, tabulate the global error E_n at $x_n = 0.16$ against $h = 0.16/n$ for $n = 2^k$, $k = 0, 1, 2, \ldots, 15$, and plot a log–log graph of $|E_n|$ against h over this range. Comment on the relationship of your results to the theoretical accuracy of the methods.

3 Numerical solutions of second-order ODEs

This section is concerned with small 'normal-mode' oscillations of a non-uniform string of length L under uniform positive tension T_0. The string's transverse displacement, $\eta(\xi, \tau)$, a function of longitudinal distance ξ and time τ, is assumed to satisfy the equation of motion (Newton's Second Law, linearised for small displacements and with gravity neglected)

$$\mu(\xi)\frac{\partial^2 \eta}{\partial \tau^2} = T_0 \frac{\partial^2 \eta}{\partial \xi^2} \, , \tag{10a}$$

where $\mu(\xi)$ is the mass per unit length of the string. If the ends $\xi = 0$ and $\xi = L$ are fixed, the appropriate boundary conditions are

$$\eta(0, \tau) = \eta(L, \tau) = 0 \quad \text{for all } \tau. \tag{10b}$$

Multiplying (10a) by $\partial\eta/\partial\tau$ leads to the energy conservation equation

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\int_{\xi_1}^{\xi_2}\left[\tfrac{1}{2}\mu(\xi)\left(\frac{\partial\eta}{\partial\tau}\right)^2 + \tfrac{1}{2}T_0\left(\frac{\partial\eta}{\partial\xi}\right)^2\right]\mathrm{d}\xi = \left[\left(T_0\frac{\partial\eta}{\partial\xi}\right)\frac{\partial\eta}{\partial\tau}\right]_{\xi_1}^{\xi_2} \quad \text{for any } \xi_1,\, \xi_2 \in [0, L], \tag{11}$$

i.e. the rate of change of total energy (kinetic plus potential) in the section $[\xi_1, \xi_2]$ is equal to the rate at which work is done at the ends (linearised transverse force times transverse velocity).

The problem admits 'normal-mode' solutions of the form

$$\eta(\xi, \tau) = Ly(x)\cos(\omega\tau + \theta) \, , \tag{12}$$

where the angular frequency ω and phase θ are constants, and dimensionless variables are defined by

$$x = \frac{\xi}{L}\ , \quad m(x) = \frac{\mu(\xi)}{\mu(0)}\ . \tag{13}$$

From (10a) and (10b), it follows that the dimensionless $y(x)$ satisfies

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + p^2 m(x) y = 0 \quad \text{and} \quad y(0) = y(1) = 0\,, \tag{14}$$

where $p = \omega L\sqrt{\mu(0)/T_0}$ is a dimensionless constant and, without loss of generality, $p \geqslant 0$. The system (14) is an example of a *Sturm-Liouville eigenproblem*: only for a discrete set of values of p, i.e. the *eigenvalues* $0 \leqslant p^{(1)} < p^{(2)} < \ldots$, are there non-zero *eigenfunction* solutions for y.

The remainder of this project specialises to a mass distribution of the form $m(x) = (1+x)^{-\alpha}$ with α a constant, in which case (14) becomes

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + p^2 (1+x)^{-\alpha} y = 0\,, \tag{15a}$$

subject to

$$y(0) = y(1) = 0\ . \tag{15b}$$

The eigenvalues and eigenfunctions can be found in explicit analytic form only for special values of α, one such being $\alpha = 2$.

Question 5 Find the analytic solution $y = y_e(x)$ of equation (15a) with $\alpha = 2$ subject to the initial condition

$$y = 0\ , \quad \frac{dy}{dx} = 1 \quad \text{at } x = 0 \tag{16}$$

for a *general* value of p. [The substitution $1 + x = \mathrm{e}^z$ *may* be helpful.] Deduce *carefully* that the smallest (non-negative) eigenvalue of (15a)–(15b) with $\alpha = 2$ is

$$p = p^{(1)} \equiv \left[\tfrac{1}{4} + \left(\frac{\pi}{\ln 2}\right)^2\right]^{1/2} \tag{17}$$

and write down the general eigenvalue $p^{(k)}$ and corresponding eigenfunction $y^{(k)}(x)$.

Equation(15a) can be solved numerically by noting that it is equivalent to

$$\frac{dy}{dx} = f(x,y,z) \equiv z\ , \quad \frac{dz}{dx} = g(x,y,z) \equiv -p^2(1+x)^{-\alpha} y \tag{18}$$

which has the form (1a) with $\mathbf{y} = (y, z)$ and $\mathbf{f} = (f, g)$, and a numerical approximation $\mathbf{Y_n} = (Y_n, Z_n)$ can be obtained using any of the methods described in §1. Here you should use the RK4 method.

Programming Task: Write a program to compute an RK4 numerical approximation (Y_n, Z_n) to the solution of equations (18) with initial condition $Y_0 = 0$, $Z_0 = 1$ at $x_0 = 0$.

Question 6 Taking $\alpha = 2$, run your program with $p = 4$ and $h = 0.1/2^k$ for $k = 0, 1, 2, \ldots, 12$ in turn, tabulating the numerical solution Y_n at $x_n = 1$ and the global error $Y_n - y_e(1)$ against $h \equiv 1/n$. Repeat with $p = 5$. Do the errors behave as expected?

Programming Task: Write a program to search for eigenvalues using the 'false position' method – a variant of the bisection method where if a root of $\phi(p) = 0$ has been located to the interval (p_1, p_2) with $\phi(p_1)$ and $\phi(p_2)$ having opposite signs, an estimate p_s for the root is calculated using the linear interpolant

$$\Phi(p) \equiv \phi(p_1)\left(\frac{p_2 - p}{p_2 - p_1}\right) + \phi(p_2)\left(\frac{p - p_1}{p_2 - p_1}\right) . \tag{19a}$$

Solving for $\Phi(p) = 0$ gives the estimate

$$p = p_s \equiv \frac{\phi(p_2)\, p_1 - \phi(p_1)\, p_2}{\phi(p_2) - \phi(p_1)} \tag{19b}$$

which is accepted if $|\phi(p_s)| < \epsilon$ where ϵ is a specified small value; if not, the process is iterated with p_1 or p_2 replaced by p_s such that $\phi(p_1)$ and $\phi(p_2)$ still have opposite signs. For the current task, $\phi(p)$ is the numerical (RK4) solution Y_n of (15a) and (16) at $x_n = 1$ obtained with a suitably small value of $h \equiv 1/n$.

Question 7 Taking $\alpha = 2$, run the program to obtain an approximation to the smallest (positive) eigenvalue $p^{(1)}$ of (15a)–(15b) with error no more than $\pm 5 \times 10^{-6}$, using $(4, 5)$ as the initial interval and tabulating all the iterates in your write-up. What values are you using for ϵ and h and why? (If you can justify them without reference to the analytic solution, so much the better.)

The final question addresses the case $\alpha = 10$, which has no simple analytic solution.

Question 8 Find approximations to the five smallest (non-negative) eigenvalues $p^{(k)}$, $k = 1, 2, 3, 4, 5$ of (15a)–(15b) with $\alpha = 10$ correct to within $\pm 5 \times 10^{-6}$, and plot the corresponding eigenfunctions $y^{(k)}(x)$ using the 'energy' normalisation

$$\int_0^1 (1 + x)^{-10} \left[p^{(k)} y^{(k)}(x)\right]^2 \, \mathrm{d}x = 1 . \tag{20}$$

(If you wish, this integral may be approximated using a black-box integration routine such as the function `trapz` in MATLAB or NumPy/Python.) Explain carefully why you are satisfied that the eigenvalues found are indeed the smallest, and that they have the required accuracy.

Comment on the results, e.g. on the shape of the eigenfunctions, both from a mathematical point of view and in terms of the physical model.

Hint: it *may* be instructive to look at the form of the eigensolutions for larger p, and/or to note that when $p \equiv \delta^{-1}$ is 'large', equation (15a) has solutions of the form $\mathrm{e}^{S(x;\delta)}$ with complex exponent $S(x; \delta) = \delta^{-1} S_0(x) + S_1(x) + O(\delta)$ (the so-called Liouville-Green or WKB approximation.*)

Additional Reference

Boyce, W. E., and DiPrima, R. C., 2001, *Elementary Differential Equations and Boundary Value Problems*, 7th edition. Publ. John Wiley & Sons Inc.

*See for example Bender, C.M. & Orszag, S.A., *Advanced mathematical methods for scientists and engineers*, Chapter 10 (in particular §10.1, Example 5).

Project 1.2: Ordinary Differential Equations

Marking Scheme and additional comments for the Project Report

The purpose of these additional comments is to provide guidance on the structure and length of your CATAM report. Use the same concepts to write the rest of the reports. To help you assess where marks have been lost, this marking scheme will be completed and returned to you during Lent Term. You are advised to keep a copy of your write-up in order to correlate your answers to the marks awarded.

Question no.	**Marks available**[1]	**Marks awarded**[2]
Programming task *Program:* for instructions regarding printouts and what needs to be in the write-up, refer to the introduction to the manual.		
Question 1 *Tables:* for presentation and layout, refer to the introduction. [*up to six tables and a quarter page of writing*][3]	C2	
Question 2 *Analytic solution:* do not include trivial steps in your working. [*half page*][3]	M4.5	
Question 3 *Graphs:* you may use one graph, or two, or three.	C1	
Question 4 *Graphs:* ditto. *Comments:* what can be said about how the global error E_n for each method varies with h? How is this reflected in the plots? [*quarter page*][3]	C1 M1	
Question 5 *Analytic solutions:* do not include trivial steps in your working [*half page*][3].	M1.5	
Question 6 *Analytic solution and numerical solution compared:* The reason for computing an analytic solution is to check that the program is working correctly ('validation'). [*couple of lines*][3].	C1	
Question 7 *Numerical approximations to the smallest eigenvalue:* Tabulate all the iterates; explain how you have chosen ϵ and h to ensure that the final approximation has the required accuracy. [*table, graph(s) and quarter page of writing*][3]	C1 M1	
Question 8 *Numerical solutions*: explain how you have located the five smallest eigenvalues, and computed them to the required accuracy. *Comments:* first identify the salient features of the graphs. Then try to explain them using mathematical arguments; link to the theory of the physical system under investigation. [*one or two pages*][3]	C2 M2	
Excellence marks awarded for, *among other things*, mathematical clarity and good, clear output (graphs and tables) — see the introduction to the Project Manual.	E2	
Total Raw Marks	20	
Total Tripos Marks	**40**	

[1] C#, M# and E#: *Computational*, *Mathematical* and *Excellence* marks respectively.

[2] For use by the assessor.

[3] Your aim is to answer succinctly the questions including the graphs and tables, and to make all important points. The length specified here should be sufficient space for you to do this but is not a target.

2.1 The Diffusion Equation

IB Methods is relevant.

1 Background and Problem Formulation

The conduction of heat down a lagged bar of length L metres may be described by the one-dimensional diffusion equation

$$\frac{\partial\theta}{\partial t} = \kappa\frac{\partial^2\theta}{\partial x^2} \quad (0 < x < L) \tag{1}$$

where $\theta(x,t)$ is the temperature (in kelvin K) averaged over the cross-section A at distance x metres along the bar and time t seconds; κ is a positive constant, the so-called *thermal diffusivity* (measured in $\mathrm{m^2\,s^{-1}}$). This description is obtained on the basis that

(i) there is negligible heat flux through the sides,

(ii) the heat flux (in the positive x-direction) through the cross section at x is $-A\,k\,\partial\theta(x,t)/\partial x$ where A is the (constant) cross-sectional area and k the (constant) thermal conductivity, and

(iii) the total heat in $a < x < b$ is

$$A\int_a^b \sigma\rho\theta(x,t)\,\mathrm{d}x \tag{2}$$

where σ is the (constant) specific heat and ρ the (constant) density, its rate of change

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[A\int_a^b \sigma\rho\theta(x,t)\,\mathrm{d}x\right] = A\sigma\rho\int_a^b \frac{\partial\theta}{\partial t}(x,t)\,\mathrm{d}x \tag{3}$$

being equal to the heat flux in across $x = a$ less the heat flux out across $x = b$

$$-Ak\frac{\partial\theta}{\partial x}(a,t) + Ak\frac{\partial\theta}{\partial x}(b,t) = Ak\int_a^b \frac{\partial^2\theta}{\partial x^2}(x,t)\,\mathrm{d}x \tag{4}$$

for any a and b, implying (1) with $k = \kappa\sigma\rho$.

Suppose that for $t < 0$, the bar is at a uniform temperature θ_0, and that for $t \geqslant 0$, the temperature at one end ($x = 0$) experiences an increase proportional to time, while the other end ($x = L$) is either maintained at a constant temperature or insulated. Equation (1) is therefore to be solved for $t > 0$ subject to the initial condition

$$\theta(x,0) = \theta_0 \quad \text{for } 0 < x < L\ , \tag{5}$$

and to the boundary conditions

$$\theta(0,t) = \theta_0 + \alpha t \quad \text{for } t > 0, \tag{6}$$

where α is a positive constant (measured in $\mathrm{K\,s^{-1}}$), and either

$$\theta(L,t) = \theta_0 \quad \text{for } t > 0, \tag{7}$$

or

$$\frac{\partial\theta}{\partial x}(L,t) = 0 \quad \text{for } t > 0 \text{ (i.e. vanishing heat flux at the insulated end).} \tag{8}$$

The goal of this project is to study the performance of a simple finite-difference method on one of these problems, for which numerical solutions can be compared with an analytic one.

2 Analytic solutions

Question 1 First consider the case of a semi-infinite bar, for which the boundary condition (7) or (8) is replaced by

$$\theta \to \theta_0 \quad \text{or} \quad \frac{\partial \theta}{\partial x}(x,t) \to 0 \quad \text{as } x \to \infty. \tag{9}$$

Substitute

$$\theta(x,t) = \theta_0 + \alpha t F(x,t) \ , \tag{10}$$

and explain with the help of dimensional analysis why in both cases F is a function only of the similarity variable

$$\xi = \frac{x}{(\kappa t)^{1/2}} \ , \tag{11}$$

and independent of θ_0 and α. Find the equation and boundary conditions satisfied in each case by the function $F(\xi)$, and show that in both cases the *unique* solution is

$$F(\xi) = \left(1 + \tfrac{1}{2}\xi^2\right) \operatorname{erfc}\left(\tfrac{1}{2}\xi\right) - \pi^{-1/2}\xi e^{-\xi^2/4} \tag{12}$$

where

$$\operatorname{erfc}(s) = \frac{2}{\sqrt{\pi}} \int_s^\infty e^{-u^2} du \ . \tag{13}$$

[This might be done by differentiating the equation twice and proceeding to *derive* the solution, or by finding a second independent solution of the equation, say as a series.*]

Now return to the case of a finite bar and define non-dimensional variables X, T and U by

$$x = LX \ , \quad t = L^2\kappa^{-1}T \ , \quad \theta(x,t) = \theta_0 + \alpha L^2 \kappa^{-1} U(X,T) \ , \tag{14}$$

in terms of which the diffusion equation (1) becomes

$$U_T = U_{XX} \quad \text{for } T > 0 \ , \quad 0 < X < 1 \ , \tag{15}$$

with initial condition

$$U(X,0) = 0 \quad \text{for } 0 < X < 1 \tag{16}$$

and boundary conditions

$$U(0,T) = T \quad \text{for } T > 0 \tag{17}$$

and either

$$U(1,T) = 0 \quad \text{for } T > 0. \tag{18}$$

or

$$U_X(1,T) = 0 \quad \text{for } T > 0 \tag{19}$$

*N.B. An n^{th}-order ODE with (ostensibly) n boundary conditions may have a non-unique solution, or no solution at all: consider, for example, $d^2y/dx^2 + 3dy/dx + 2y = 0$ with $y = 1$ at $x = 0$, $y \to 0$ as $x \to \infty$, or $d^2y/dx^2 - 3dy/dx + 2y = 0$ with the same conditions.

Question 2 Find an analytic solution (as an infinite series) of the first problem (15)–(18) as follows. Ignoring the initial condition (16) for the time being, subtract off the simplest function which satisfies the boundary conditions (17) and (18),

$$U(X,T) = T(1-X) + V(X,T) \tag{20}$$

$$\Rightarrow \quad 1 - X + V_T = V_{XX} \ , \quad V(0,T) = 0 \ , \quad V(1,T) = 0 \ , \tag{21}$$

and noting that there is a particular solution with V independent of T, subtract that off,

$$V(X,T) = -\tfrac{1}{3}X + \tfrac{1}{2}X^2 - \tfrac{1}{6}X^3 + W(X,T) \ , \tag{22}$$

to obtain the *homogeneous* problem

$$W_T = W_{XX} \ , \quad W(0,T) = 0 \ , \quad W(1,T) = 0 \tag{23}$$

which has separable solutions for W. Now construct a superposition of these separable solutions which satisfies the initial condition (16). Show that

$$W(X,T) \sim \frac{2}{\pi^3} \sin(\pi X) \mathrm{e}^{-\pi^2 T} \quad \text{as } T \to \infty. \tag{24}$$

Adapt this method to obtain an (infinite-series) analytic solution of the second problem (15)–(17) and (19).

Programming Task: Write a program to evaluate both analytic solutions by summing a finite number of terms of each series. Tabulate $U(X,T)$ for both problems at $T = 0.25$ and $X = 0.125n$, $n = 0, 1, \ldots, 8$ and also tabulate the semi-infinite solution (11)–(12) evaluated at these T- and X-values [note that there is a MATLAB function erfc]. Plot the non-dimensionalised temperature profiles, U against X, for all three at $T = 0.03125$, 0.125, 0.25, 0.375, 0.75 and 1.5; also plot the non-dimensionalised heat flux $-\partial U/\partial X$ at $X = 0$ for all three against T over this range.

Explain why you are satisfied that enough terms have been kept in the truncated series to provide 'sufficiently' accurate solutions (at least for $T \geqslant 0.03125$; take into account what accuracy will be needed for question 3 below). Compare how the three sets of temperature profiles evolve in time, and discuss.

3 Numerical Integration

The first problem (15)–(18) is now to be solved numerically as follows. Let the domain $0 \leqslant X \leqslant 1$ be divided into N intervals, each of length $\delta X = 1/N$, and let U_T be approximated by a forward difference in time:

$$\frac{\partial U(X,T)}{\partial T} = \frac{U(X,T+\delta T) - U(X,T)}{\delta T} + O\left(\delta T\right) \;, \tag{25}$$

and U_{XX} by a centred difference in space at the current time:

$$\frac{\partial^2 U(X,T)}{\partial X^2} = \frac{U(X+\delta X,T) - 2U(X,T) + U(X-\delta X,T)}{\left(\delta X\right)^2} + O\left(\left(\delta X\right)^2\right) \;, \tag{26}$$

giving the numerical scheme

$$U_n^{m+1} = U_n^m + C\left[U_{n+1}^m - 2U_n^m + U_{n-1}^m\right] , \tag{27}$$

where U_n^m is an approximation to $U(n\delta X, m\delta T)$ and $C = \delta T/\left(\delta X\right)^2$ (the so-called *Courant number*).

Question 3

Programming Task: Write a program to implement this numerical scheme, and run it with $N = 4, 8, 16, 32$ and $C = \frac{2}{3}$, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{6}$ and $\frac{1}{12}$. For the case $N = 4$, $C = \frac{1}{2}$, tabulate both the analytic and the numerical solutions, and the value of the error, at $T = 0.125$, 0.25 and 0.375.

Discuss both the stability and the accuracy of the numerical scheme for the different values of N and C. Are the results consistent with the theoretical order of accuracy of the scheme? Illustrate the discussion with appropriate short tables and/or graphs.

Reference

Ames, W.F. *Numerical Methods for Partial Differential Equations*, Academic Press.

2.2 Schrödinger's Equation

Part IB Quantum Mechanics is useful but not essential. (All required background material can be found in the project itself and/or the references.)

1 Introduction

Consider a single particle of mass m and energy ϵ, in a (real) one-dimensional potential $v(x)$. Then Schrödinger's time-independent wave equation is

$$\left[-\frac{\hbar^2}{2m}\frac{d^2}{dx^2} \;+\; v(x)\right]\psi(x) \;=\; \epsilon\,\psi(x) \tag{1}$$

where $2\pi\hbar$ is Planck's constant, The state of the system is represented by a (complex) time-dependent wavefunction $\psi(x)e^{-i\epsilon t/\hbar}$.

Let ϵ_1 be the unit of energy so that $E = \epsilon/\epsilon_1$ is the dimensionless total energy. Define also a dimensionless co-ordinate $X = x\sqrt{2m\epsilon_1}/\hbar$. Then the equation simplifies as

$$\left[-\frac{d^2}{dX^2} \;+\; V(X)\right]Y(X) \;=\; E\,Y(X) \tag{2}$$

where $V(X) = v(x)/\epsilon_1$ is the dimensionless potential energy, and $Y(X) = \psi(x)$ is the stationary wave function.

This project is about *bound states*, in which the particle is localised near $X = 0$. In bound states, $Y(X)$ tends to zero as $|X| \to \infty$, sufficiently fast that the wave function is normalisable, that is,

$$\int_{-\infty}^{\infty} |Y(X)|^2 dX < \infty.$$

Another feature of bound states is that the real and imaginary parts of Y decay monotonically for sufficiently large positive or negative X. Such solutions to Eq. (2) only exist for certain values of E, which are the 'eigenvalues'.

The aim of this project is to determine a few of these eigenvalues, and their corresponding eigenfunctions $Y(X)$, numerically. This will be achieved using 'forward shooting', which involves finding values of E, $Y(0)$ and $Y'(0)$ by trial-and-error, such that the solution of (2) behaves appropriately as $|X| \to \infty$.

2 Harmonic oscillator

Consider the harmonic oscillator potential

$$V(X) = X^2\,. \tag{3}$$

2.1 Theory

For this potential, information about the solutions to Eq. (2) can be found in textbooks about quantum mechanics, for example Refs. [1,2,3]. We summarise the relevant information for this project. The general solution to Eq. (2) is

$$Y(X) = \left[c_{\mathrm{e}} f_{\mathrm{e}}(X) + c_{\mathrm{o}} f_{\mathrm{o}}(X)\right] \exp\left(-X^2/2\right)$$

where f_e is an even function [that is, $f_\mathrm{e}(X) = f_\mathrm{e}(-X)$] and f_o is an odd function [$f_\mathrm{o}(X) = -f_\mathrm{o}(-X)$]. The constants $c_\mathrm{e}, c_\mathrm{o}$ depend on the values chosen for $Y(0)$ and $Y'(0)$. The most important properties of f_e and f_o are

- Normalisable solutions exist if and only if E is an odd positive integer. Otherwise, all solutions diverge as $|X| \to \infty$.
- If $(E-1)/2$ is an even integer then any normalisable solution must be even ($c_\mathrm{o} = 0$). Similarly, if $(E-1)/2$ is odd then any normalisable solutions must be odd ($c_\mathrm{e} = 0$).
- The functions f_e and f_o that appear in the normalisable solutions are called Hermite polynomials. For $E = 1, 3, 5$, the relevant Hermite polynomials are 1, X, and $1 - 2X^2$.
- In general, one has

$$f_\mathrm{e}(X) = \sum_{n=0}^{\infty} a_n X^{2n}, \qquad f_\mathrm{o}(X) = \sum_{n=0}^{\infty} b_n X^{2n+1},$$

  where $a_0 = 1$, $b_0 = 1$, and the remaining coefficients satisfy recurrence relations

$$a_{n+1} = \frac{4n+1-E}{(2n+2)(2n+1)}\, a_n \;, \qquad b_{n+1} = \frac{4n+3-E}{(2n+3)(2n+2)}\, b_n \;.$$

The first three points above can all be derived by using the last point. (If E is an odd positive integer then one sees from the recurrence relations that either f_e or f_o will be a polynomial.)

2.2 Computation

In this part of the project, we restrict our attention to odd solutions by taking

$$Y(0) = 0 \text{ and } Y'(0) = 1\,. \tag{4}$$

(Setting $Y'(0) = 1$ does not lose any generality.)

Programming Task: Write a program to solve Schrödinger's equation (2) with the potential (3) by numerical integration, starting from the initial conditions (4). The program should find and (optionally) plot the solution $Y(X)$ for a given value of E and a range of integration $X \in [0, X_\mathrm{max}]$, where you will decide on appropriate values for X_max. You may use one of the built-in MATLAB solvers, *e.g.* `ode45` for which you can control the relative and absolute tolerances with `odeset('RelTol', rtol, 'AbsTol', atol)`, inserting sensible values for `rtol` and `atol`. Alternatively, you might use the fixed-steplength Runge-Kutta outine you wrote for the Ordinary Differential Equations core project.

Question 1 Run the program with $E = 2.9$ to obtain the value of $Y(5)$ correct to 8 significant figures. Explain how you have tested that the input parameters of the ODE solver (tolerances or steplength) are appropriate for this purpose, and present evidence that the required accuracy has been achieved.

Question 2 Run the program with $X_\mathrm{max} = 5.0$ and $E = 2.9995$ and 3.0005 in turn, and plot *one* graph with both solutions $Y(X)$ superposed. Why are you satisfied that the numerical results are correct?

Question 3 Why can you be sure, without integrating beyond $X = X_{\max}$, that the solutions found in the previous question will tend monotonically to $\pm\infty$ over the range $[X_{\max}, \infty)$?

Question 4 How will a *numerical* solution of (2)–(4) with $E = 3$ behave as $X \to \infty$, and why? (It may be instructive to vary the parameters of the ODE solver.)

3 Nearly-square potential well

Programming Task: Modify your program so that $V(X)$ is given by

$$V(X) = -\frac{\Delta V}{1 + X^4} \tag{5}$$

where ΔV is a strictly positive constant. Also modify your program so that you can use initial conditions appropriate to either even or odd solutions. Be sure to specify these initial conditions in your write-up.

Question 5 For bound states of this $V(X)$, why is there no loss of generality in restricting to solutions which are either even or odd in X?

3.1 Shallow potential : $\Delta V = 1$

The potential (5) has at least one bound state for any $\Delta V > 0$, and more for larger ΔV. For $\Delta V = 1$, it turns out that there is exactly one bound state.

Question 6 Use your program to verify that the potential (5) with $\Delta V = 1$ has only one bound state, at least to the extent of trying $E = -1$, $E = 0$ and a few judiciously chosen values of E between -1 and 0. You should check for both even and odd solutions. Present plots of these solutions over a suitable range $[0, X_{\max}]$, explaining your choice(s) for $X_{\max}$ and the input parameters of the ODE solver.

Question 7 Why is there no need to consider values of E greater than 0 or less than $-\Delta V$ when seeking bound-state solutions for the potential (5)? Argue carefully why your numerical results indicate that for $\Delta V = 1$ there can be no more than one bound state.

Question 8 Determine the single [negative] energy eigenvalue, correct to 3 significant figures, by interval-halving (or otherwise). Be sure to use appropriate value(s) of $X_{\max}$ (with justification). Include in your write-up a graph with superimposed plots of $Y(X)$ for a final pair of integrations which bracket the eigenvalue sufficiently closely, remembering to identify them with the values of E used.

Question 9 Explain why there must be a bound state with energy between these values. [*Hint*: what is the asymptotic behaviour of $Y(X)$ as $X \to \infty$?]

3.2 Stronger potentials

For larger ΔV, there tend to be more bound states. To understand this, two theoretical results can be useful.

1. Consider a 'square' potential well of width $2L$:

$$V_{\text{square}}(X) = \begin{cases} -\Delta V & \text{if } |X| < L \\ 0 & \text{if } |X| > L \end{cases} \tag{6}$$

As discussed in textbooks such as Refs. [1,2,3], this potential has exactly N bound states if

$$\frac{2L}{\pi}\sqrt{\Delta V} \in (N-1, N] \,. \tag{7}$$

Bound states for the square potential and the potential of Eq. (5) turn out to be qualitatively similar, but one must choose L appropriately. For example, one might estimate the 'width' of the the potential (5) as

$$L = \left(\frac{\int_0^\infty X^2 V(X) \mathrm{d}X}{\int_0^\infty V(X) \mathrm{d}X} \right)^{1/2} = 1. \tag{8}$$

2. The WKB approximation (discussed in Refs. [1,2,3]) can be used to analyse situations where ΔV is very large. Specifically, as $\Delta V \to \infty$ then the bound state energies asymptote to $\tilde{E}_n = -\Delta V/(1+X_n^4)$ for $n = 0, 1, 2, \ldots, N-1$, where N is the number of bound states and X_n is determined by

$$\int_{-X_n}^{X_n} \sqrt{\tilde{E}_n - V(X)} \, \mathrm{d}X = \left(n + \tfrac{1}{2}\right) \pi \tag{9}$$

Also, N is determined (asymptotically) by the condition that

$$\frac{2\tilde{L}}{\pi}\sqrt{\Delta V} \in \left(N - \tfrac{1}{2}, N + \tfrac{1}{2}\right] \,. \tag{10}$$

where

$$\tilde{L} = \frac{1}{2} \int_{-\infty}^{\infty} \sqrt{\frac{-V(X)}{\Delta V}} \, \mathrm{d}X = \int_0^\infty \frac{\mathrm{d}X}{\sqrt{1+X^4}} \approx 1.85 \tag{11}$$

Question 10 For the potential (5) with $\Delta V = 36$, find all possible bound-state energy eigenvalues correct to at least 3 significant figures and display plots of the corresponding eigenfunctions, recording the number of times each takes the value zero.

Explain carefully why you are satisfied that there are no other bound states. Mention any precautions needed to ensure that all eigenvalues are obtained to the required accuracy.

Discuss the relationship between your results and the theories for the square well and the WKB approximation.

References

[1] LI Schiff, *Quantum Mechanics*, 3rd edition, McGraw-Hill 1968.

[2] DJ Griffiths, *Introduction to Quantum Mechanics*, Prentice Hall 1995.

[3] S Gasiorowicz, *Quantum Physics*, Wiley 2003.

2.3 Non-Euclidean Geometry

Some of the material from this project is covered in the IB Geometry course (and in the given reference).

1 Introduction

The aim of this project is to create some pictures that illustrate the action of certain groups of isometries on the sphere, and on the disc model of the hyperbolic plane.

2 Spherical geometry

We work in the extended complex plane $\mathbb{C} \cup \{\infty\}$. In this project a *spherical line* will be either a straight line through the origin, or a circle of the form

$$\{z \in \mathbb{C} : |z - a|^2 = |a|^2 + 1\} \tag{1}$$

where $a \in \mathbb{C}$. In fact, under stereographic projection, a spherical line corresponds to a great circle on the unit sphere. Moreover stereographic projection preserves angles. A *spherical triangle* is obtained by joining points $z_1, z_2, z_3 \in \mathbb{C} \cup \{\infty\}$ by spherical lines.

> **Question 1** If the spherical line (1) passes through distinct points z_1 and z_2, find a formula for a in terms of z_1 and z_2. Write a program that given $z_1, z_2, z_3 \in \mathbb{C} \cup \{\infty\}$, draws and fills in the spherical triangle with vertices z_1, z_2, z_3.
>
> How do spherical lines meet the unit circle $|z| = 1$?

The *second cosine rule* states that if a spherical triangle has side lengths a, b, c and internal angles α, β, γ (the side of length a being opposite the vertex with angle α) then

$$\cos\alpha + \cos\beta\cos\gamma = \sin\beta\sin\gamma\cos a. \tag{2}$$

The side lengths here are "spherical distances" where for example the spherical distance between 0 and z is $2\tan^{-1}|z|$.

3 Hyperbolic geometry

We work in the unit disc $D = \{z \in \mathbb{C} : |z| < 1\}$. A *hyperbolic line* is (the intersection of D and) either a straight line through the origin, or a circle of the form

$$\{z \in \mathbb{C} : |z - a|^2 = |a|^2 - 1\} \tag{3}$$

where $a \in \mathbb{C}$ with $|a| > 1$. Note that (1) and (3) differ by a change of sign. A *hyperbolic triangle* is obtained by joining points $z_1, z_2, z_3 \in D$ by hyperbolic lines.

> **Question 2** Write a program that given $z_1, z_2, z_3 \in D$, draws and fills in the hyperbolic triangle with vertices z_1, z_2, z_3.
>
> How do hyperbolic lines meet the unit circle $|z| = 1$?

The *second cosine rule* states that if a hyperbolic triangle has side lengths a, b, c and internal angles α, β, γ (the side of length a being opposite the vertex with angle α) then

$$\cos\alpha + \cos\beta\cos\gamma = \sin\beta\sin\gamma\cosh a. \tag{4}$$

The side lengths here are "hyperbolic distances" where for example the hyperbolic distance between 0 and z is $2\tanh^{-1}|z|$.

4 Regular polygons

A spherical or hyperbolic n-gon is obtained by joining n vertices by spherical or hyperbolic lines.

> **Question 3** For which integers n does there exist $r > 0$ such that the spherical n-gon with vertices at the roots of $z^n = r^n$ has internal angle $2\pi/3$? What happens if we replace 'spherical' by 'hyperbolic'? [Hint: Work out a formula for the area of an n-gon, by subdividing into triangles and quoting a result from the Geometry course.]
>
> Use your programs from Questions 1 and 2 to illustrate how these n-gons may be subdivided into $2n$ triangles each with internal angles $\pi/2$, $\pi/3$ and π/n. [Hint: Use the formulae (2) and (4) to find the vertices.]

5 Symmetry groups and tesselations

Let p be an odd prime. Let $\mathrm{SL}(2,p)$ be the finite group of all 2×2 matrices whose entries are integers mod p, and have determinant 1. Let $\mathrm{PSL}(2,p)$ be the quotient of $\mathrm{SL}(2,p)$ by the subgroup $\{\pm I\}$ of order 2. It can be shown that $\mathrm{PSL}(2,p)$ is generated by

$$\sigma_1 = \pm\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \qquad \sigma_2 = \pm\begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}, \qquad \sigma_3 = \pm\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{5}$$

Inversion in a Euclidean circle C is the unique map of the form

$$z \mapsto \frac{a\bar{z} + b}{c\bar{z} + d}$$

that fixes each point of C. For example inversion in the unit circle is $z \mapsto 1/\bar{z}$.

Let L be a spherical or hyperbolic line. We define *reflection in L* to be either the Euclidean reflection (if L is a straight line) or inversion in L (if L is a Euclidean circle).

Let Δ_0 be one of the triangles in Question 3, where now $n = p$ is a prime. Let R_1, R_2, R_3 be the reflections in the sides of Δ_0 opposite the vertices with internal angles $\pi/2, \pi/3$ and π/p (in that order).

> **Question 4** Describe the transformations $S_1 = R_2R_3$, $S_2 = R_3R_1$ and $S_3 = R_1R_2$ geometrically. Show that they have the same orders as the group elements in (5).

Let $g \in \mathrm{PSL}(2,p)$ and let Δ be a triangle. The pair (g, Δ) is *admissible* if it is obtained from $(\pm I, \Delta_0)$ by a sequence of operations replacing (g, Δ) by $(\sigma_i g, S_i\Delta)$ for some $i = 1, 2$ or 3. Note that we can list the elements of $\mathrm{PSL}(2,p)$ by starting with $\pm I$ and then repeatedly multiplying by the generators σ_1, σ_2 and σ_3 until no new elements are found.

Question 5 Let $p = 3$. For each $g \in \mathrm{PSL}(2, p)$ find a triangle Δ such that (g, Δ) is admissible. Use your programs to plot these triangles Δ. How many triangles are there in total? Do the triangles ever overlap? Could we obtain any more triangles by applying the transformations S_1, S_2 and S_3? Repeat the question for $p = 5$ and $p = 7$.

Question 6 How do your pictures for $p = 3$ and $p = 5$ relate to the Platonic solids? To which other well known groups are $\mathrm{PSL}(2, 3)$ and $\mathrm{PSL}(2, 5)$ isomorphic, and how is this suggested by your pictures?

Reference

P.M.H. Wilson, *Curved spaces*, CUP 2008.

2.4 Simulation of Random Samples from Parametric Distributions

This project requires an understanding of the Part IB Statistics course.

1 Introduction

Let X be a random variable and let $F(x)$ be the distribution function of X, so that $\mathbb{P}(X \leqslant x) = F(x)$. We will assume that $F(x)$ is a continuous strictly increasing function, and define $U = F(X)$, which is therefore a random variable with values in $[0, 1]$. It is easy to find the distribution function of U, for

$$\begin{aligned} \mathbb{P}(U \leqslant u) &= \mathbb{P}\big(F(X) \leqslant u\big) \\ &= \mathbb{P}\big(X \leqslant F^{-1}(u)\big) \\ &= F\big(F^{-1}(u)\big) = u, \quad \text{for } 0 \leqslant u \leqslant 1. \end{aligned}$$

Hence U has the *uniform* or *rectangular* distribution on $[0, 1]$. We write this as

$$U \sim \text{Unif}[0, 1].$$

Clearly, given $U \sim \text{Unif}[0, 1]$, if we define X by $X = F^{-1}(U)$, then X will have distribution function $F(x)$. We can use this fact to generate $X_1, \ldots, X_n$, (pseudo-)random variables from a given distribution function $F(x)$.

Note: The MATLAB function `rand` *generates* $\text{Unif}[0, 1]$ *(pseudo-)random variables which may be assumed independent. To generate any other kind of random variable you will need to write your own routines. The use of existing routines, for example from the statistics toolbox or R will not earn you any credit. However, you may wish to compare your random number generators to pre-existing ones.*

2 The Exponential Distribution

Take $F(x) = 1 - e^{-\theta x}$, $x \geqslant 0$, corresponding to an exponential density with rate θ, mean θ^{-1}, and probability density function

$$f(x \mid \theta) = \theta e^{-\theta x}, \quad x \geqslant 0.$$

Question 1 Suppose that instead of indexing the probability distribution function by its rate θ, we decide to index it by its median m given by

$$\int_0^m f(x \mid \theta)\, dx = \frac{1}{2}.$$

Find θ as a function of m and hence find $g(x \mid m) = f\big(x \mid \theta(m)\big)$.

Question 2 Take $(u_1, \ldots, u_n)$, sampled from Unif$[0, 1]$, and hence compute the x_i defined by $u_i = 1 - e^{-\theta_0 x_i}$, giving $(x_1, \ldots, x_n)$, sampled from $f(x \mid \theta_0)$. Try this for $n = 6$, $\theta_0 = 1.2$. Plot the resulting *log likelihood* function $\ell_n(m)$ against m where

$$\ell_n(m) = \log \prod_{i=1}^{n} g(x_i \mid m).$$

Derive analytically $\widehat{m}_n$, the value of m which maximises $\ell_n(m)$, and compare this with m_0, the true value of the median.

Question 3 Repeat all of question 2 above for $n = 25, 50, 100$, and comment on the qualitative changes you observe (if any) in the shape of $\ell_n(m)$.

Question 4 Suppose that X, Y are independent random variables, each with a probability distribution function corresponding to an exponential with mean $1/\theta$. Calculate the moment generating function $M_X(\lambda) = E(e^{\lambda X})$ of X. Show that $X + Y \sim \Gamma(2, \theta)$.

Note that if X is distributed as a $\Gamma(n, \theta)$ random variable, then it has density function $f(x) = \theta^n x^{n-1} e^{-\theta x}/(n-1)!$ and moment generating function $E(e^{tX}) = (1 - t/\theta)^{-n}$, $(t < \theta)$.

Question 5 Take $f(x \mid \theta) = \theta^2 x e^{-\theta x}$, $x \geqslant 0$, and integrate it to find $F(x)$. Can you compute F^{-1} in closed form?

Question 6 The *log-likelihood* function is now

$$\ell_n(\theta) = \log \prod_{i=1}^{n} f(x_i \mid \theta)$$

Calculate the maximum likelihood estimator for θ.

Question 7 Take $\theta_0 = 2.2$, generate a random sample of $x_1, \ldots, x_n$ from $f(x \mid \theta_0)$ and plot $\ell_n(\theta)$ against θ, for $n = 10, 30, 50$. For each sample, calculate the maximum likelihood estimator for θ, and compare it with θ_0, describing any similarities or differences between this case and that in question 3.

Question 8 We investigate the distribution of $\widehat{\theta}_n$ as follows. Take $\theta_0 = 2.2$ and $N = 200$. Take $\mathbf{x}(1), \ldots, \mathbf{x}(N)$ as N independent random samples each of size $n = 10$ from $f(x \mid \theta_0)$. Let $\widehat{\theta}_n(1), \ldots, \widehat{\theta}_n(N)$ be the corresponding maximum likelihood estimators of θ. Generate the histogram[*] of $\widehat{\theta}_n(1), \ldots, \widehat{\theta}_n(N)$. How does this histogram change if we increase n from 10 to 50?

3 The Normal Distribution

Since the normal or Gaussian distribution plays such an important role in probability and statistics, it is clearly of interest to know how to generate, say, $X_1, \ldots, X_n$, a random sample from $N(\mu, \sigma^2)$, the normal distribution with mean μ and variance σ^2. An *unsubtle* method would be to use $X = F^{-1}(U)$, where $U \sim \text{Unif}[0, 1]$, and $F(x) = \int_{-\infty}^{x} e^{-\frac{1}{2}v^2} dv \big/ \sqrt{2\pi}$. This method is *not* recommended, because $F(x)$ here is not of closed form. So what do we do instead?

Recall the following from Part IA Probability: if (Φ, V) have joint density $f(\phi, v)$, and we define

$$\begin{aligned} X &= X(\Phi, V) \\ Y &= Y(\Phi, V) \end{aligned}$$

[*] For a definition of the histogram see, for example, *Statistical Theory* (1976) by B.W. Lindgren, pp. 206–7. You can use `histogram` in MATLAB to draw one.

so that (X, Y) is a 1–1 function of (Φ, V), then (X, Y) has joint density $g(x, y)$ where

$$g(x,y) = f\big(\phi(x,y), v(x,y)\big) \left| \frac{\partial(\phi, v)}{\partial(x,y)} \right| .$$

Question 9 Show that if $f(\phi, v) = \frac{1}{4\pi} e^{-v/2}$, $0 \leqslant \phi \leqslant 2\pi, v \geqslant 0$, and if we define

$$\begin{aligned} X &= \mu_1 + \sigma\sqrt{V}\cos\Phi, \\ Y &= \mu_2 + \sigma\sqrt{V}\sin\Phi, \end{aligned}$$

then X, Y are independent $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$ random variables, i.e.,

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\{(x-\mu_1)^2+(y-\mu_2)^2\}/2\sigma^2}, \quad -\infty < x, y < \infty.$$

We apply this by

- generating A, B, independent Unif$[0, 1]$ variables;
- defining $\Phi = 2\pi A$ and $V = -2\log(1 - B)$;
- defining $X = \mu + \sigma\sqrt{V}\cos\Phi$ and $Y = \mu + \sigma\sqrt{V}\sin\Phi$.

Question 10 Write a program to generate a random sample $(x_1, \ldots, x_n)$ of size n from distribution $N(\mu, 1)$. Explain how to construct an 80% confidence interval for μ.

Question 11 For $\mu = 0$, generate a sample of size $n = 100$ from distribution $N(\mu, 1)$ and check whether the confidence interval does indeed contain μ. Repeat this procedure 25 times and display the results in a table with four columns, containing the sample mean, the lower and upper bound of the confidence interval, and 1 or 0 to indicate whether or not the interval contained the true mean. How many times did the interval *not* contain μ?

Question 12 If questions 10 and 11 were to be repeated with $n = 50$ and $\mu = 4$, how many times would you expect the confidence interval *not* to contain μ?

4 The χ^2 Distribution

Question 13 Write a program to generate a random sample of size n from each of the following distributions:

(a) chi-square with 1 degree of freedom (χ^2_1);
(b) chi-square with 5 degrees of freedom (χ^2_5);
(c) chi-square with 40 degrees of freedom (χ^2_{40}).

Run your program for $n = 100, 300, 500$ and include a histogram in each case. How do these histograms change in shape as you change the degrees of freedom?