



*Mathematical Tripos Part IB*

Computational Projects  
**2018/2019**

*CATAM*

# Mathematical Tripos Part IB

## Computational Projects

July 2018

*Edited by the Computational Projects Assessors Committee*

*Chair (2017-18): Dr S.J. Cowley*

*Chair and Course Director (2018-19): Dr J.R. Taylor*

*Assistant Course Director: Dr M. Spivack*

**Computer-Aided Teaching of All Mathematics**

**Faculty of Mathematics**

**University of Cambridge**

# Contents

## *Introduction*

### *Optional Introductory Non-Examinable Project*

#### 0.1 Root Finding in One Dimension

### *Core Projects*

#### 1.1 Matrices over Finite Fields

#### 1.2 Ordinary Differential Equations

### *Additional Projects*

#### 2.1 The Diffusion Equation

#### 2.2 Schrödinger's Equation

#### 2.3 Polynomial Images of Circles

#### 2.4 Simulation of Random Samples from Parametric Distributions

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 2018-19. At the time of writing (July 2018) the booklet for the academic year 2017-18 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 a MATLAB session and the introductory lectures. 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 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 time of submission 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 offer a short “feedback” supervision on one of the Core Projects to those who have submitted.

*Easter Term, Part IB:* you have one week to make *last-minute* changes to the additional projects. Then submit your work (see §6.2 below).

*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 early in 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 sites** 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 <sup>1</sup>

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

---

<sup>1</sup> These links work at the time of writing. Unfortunately *The MathWorks* have an annoying habit of breaking their links.

- *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>

A printable version is available from

[http://uk.mathworks.com/help/pdf\\_doc/matlab/getstart.pdf](http://uk.mathworks.com/help/pdf_doc/matlab/getstart.pdf)

- 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 will not be spoilt for choice: Google returns about 61,000,000 hits for the search ‘MATLAB introduction’ (up from 44,000,000 hits last year), and about 8,000,000 hits for the search ‘MATLAB introduction tutorial’ (up from 1,000,000 hits).

- 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,<sup>2</sup> especially if you have not programmed before, but *you are free to write your programs in any computing language whatsoever*.<sup>3</sup> Python, R,<sup>4</sup> C, C++, Mathematica,<sup>5</sup> Maple<sup>6</sup> and Haskell have been used by several students in the past, and Excel

<sup>2</sup> Except where an alternative is explicitly stated, e.g. see footnotes 4 and 6.

<sup>3</sup> In an average year, something less than 10% of projects submitted do not use the supported programming language.

<sup>4</sup> 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/>.

<sup>5</sup> Mathematica is a software package that supports symbolic computations and arbitrary precision numerical calculations, as well as visualisation. It is available on both the Mathematics and the Central/College MCS. 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://things.maths.cam.ac.uk/computing/software/mathematica/install\\_mathematica.html](https://things.maths.cam.ac.uk/computing/software/mathematica/install_mathematica.html)

<sup>6</sup> 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. At the time of writing it is expected to be available on the CATAM MCS, and Maple may also be available on the Central/College Windows MCS.

has been used for plotting graphs of computed results. The choice is your own, provided your system can produce results and program listings on paper.<sup>7</sup>

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 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.

---

<sup>7</sup> There is no need to consult the *CATAM Helpline* as to your choice of language.



**No marks** are given for the submission of program code without a report, or vice versa. **Both** program code **and** report must be submitted in **both** hard copy **and** electronic copy.

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.  $\alpha$ s or  $\beta$ s) are awarded. The maximum contribution to the final merit mark is thus 160 and roughly the same (averaging over the  $\alpha$  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*.<sup>8</sup> 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 print only one graph a 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.

---

<sup>8</sup> See also the paragraph on *Citations* in §5

- 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.

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  $O(f(n))$  or  $\Omega(f(n))$  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. However, each year some students just do not get it. 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, printouts 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. Remember that you are writing a report to be read by a *human being*, who will not want to wade through pages and pages of irrelevant or unimportant data. Twenty sides of graphs would be excessive for most projects, even if the graphs were printed one to a page.<sup>9</sup> 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. 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 *both* as hard-copy *and* electronically (both submissions must be identical). In particular, you will be asked to submit your write-ups electronically in Portable Document Format (PDF) form. Note that many word processors (e.g.  $\text{\LaTeX}$ , *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  $\text{\LaTeX}$  (or, if they are of an older generation,  $\text{\TeX}$ ), e.g. this document is written in  $\text{\LaTeX}$ . However, please note that although  $\text{\LaTeX}$  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/devices-networks-printing/compsoft/  
getting-microsoft-software/office365](https://help.uis.cam.ac.uk/service/devices-networks-printing/compsoft/getting-microsoft-software/office365).

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

---

<sup>9</sup> Recall that graphs should not as a rule be printed one to a page.

<http://www.libreoffice.org/download/download/>.

Both *Microsoft Word* and *LibreOffice* are available on the MCS.

*L<sup>A</sup>T<sub>E</sub>X*. If you decide to use a *L<sup>A</sup>T<sub>E</sub>X*, or you wish to try it out, *L<sup>A</sup>T<sub>E</sub>X* is available on the MCS. If you are going to use it extensively, then you will probably want to install *L<sup>A</sup>T<sub>E</sub>X* on your own personal computer. This can be done for free. For recommendations of *T<sub>E</sub>X* distributions and associated packages see

- <http://www.tug.org/begin.html#install> and
- <http://www.tug.org/interest.html#free>.

*Front end*. In addition to a *T<sub>E</sub>X* distribution you will also need a front-end (i.e. a ‘clever editor’). A [comparison of \*T<sub>E</sub>X\* editors](#) is available on WIKIPEDIA; below we list a few of the more popular *T<sub>E</sub>X* editors.

*T<sub>E</sub>Xstudio*. For Windows, Mac and Linux users, there is *T<sub>E</sub>Xstudio*. The *proT<sub>E</sub>Xt* distribution, based on *MiK<sub>T</sub><sub>E</sub>X*, includes the *T<sub>E</sub>Xstudio* front end.

*T<sub>E</sub>Xworks*. Again for Windows, Mac and Linux users, there is *T<sub>E</sub>Xworks*. The *MiK<sub>T</sub><sub>E</sub>X* distribution includes *T<sub>E</sub>Xworks*.

*T<sub>E</sub>XShop*. Many Mac aficionados use *T<sub>E</sub>XShop*. To obtain *T<sub>E</sub>XShop* and the *T<sub>E</sub>XLive* distribution see <http://pages.uoregon.edu/koch/texshop/obtaining.html>.

*T<sub>E</sub>XnicCenter*. *T<sub>E</sub>XnicCenter* 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 *L<sup>A</sup>T<sub>E</sub>X* as necessary.

*Learning L<sup>A</sup>T<sub>E</sub>X*. A *Brief L<sup>A</sup>T<sub>E</sub>X Guide for CATAM* is available for download from

<http://www.maths.cam.ac.uk/undergrad/catam/LaTeX/Brief-Guide.pdf> .

- The *L<sup>A</sup>T<sub>E</sub>X* 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/LaTeX/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 L<sup>A</sup>T<sub>E</sub>X* page; see

[http://www.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX\\_intro.html](http://www.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html).

*Layout of the first page*. So that your candidate number can be added to each project, on the first page of each project write-up you should **write the project number** clearly in the top left hand corner and should leave a gap 11 cm wide by 5 cm deep in the top right hand corner (for a sticky label).

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

*Further technicalities*. Please do not print text in red or green ink (although red and/or green lines on plots are acceptable). Please print on only one side of the paper, 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 **printed listings** 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. Many of you will use your own computer. However, you can also use the CATAM Managed Cluster Service (MCS), which is located in room GL.04 (commonly referred to as the *CATAM room*) in the basement of Pavilion G, CMS.

The CMS buildings are generally open from 8.30am–5.30pm, Monday–Friday. They are also open 8.30am–1pm on Saturdays during the Michaelmas and Lent Terms (but not necessarily the Easter Term). They are closed on Sundays. You should not remain in the *CATAM room* when the CMS buildings are locked.

You should report problems with the CATAM MCS, e.g. broken hardware, printers that are not working, to [help@maths.cam.ac.uk](mailto:help@maths.cam.ac.uk).

You can also use other computing facilities around the University; for further information (including which Colleges are linked to the MCS network) see<sup>10</sup>

<https://tinyurl.com/mcs-sites>

At most MCS locations you can access the MATLAB software just as in the *CATAM room*, and any files you store on the MCS from one location should be accessible from any other MCS location.

#### 3.1 Out-of-term work

The *CATAM room* is available most of the time that the CMS buildings are open, although the room is sometimes booked for other purposes during the vacations. Effort is made to ensure that it is available in the week after the end of the Michaelmas and Lent Full Terms, and in the week before the start of the Lent and Easter Full Terms. The availability of the room can be checked online at <http://www.maths.cam.ac.uk/internal/catam/catambook.html>.

#### 3.2 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.<sup>11</sup> **Malfunctions of your own equipment or the MCS are not an excuse for late submissions:** leave yourself enough time before the deadline.

---

<sup>10</sup> 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**).

<sup>11</sup> In the past a student has lost his CATAM work courtesy of a stolen computer, and another after water was accidentally poured over his computer.

Possibly one of the easiest ways to ensure that your work is backed up 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](http://en.wikipedia.org/wiki/Comparison_of_online_backup_services). In particular note that you have 1TB of OneDrive personal storage space via your University Microsoft account which is available under a University agreement (see <https://tinyurl.com/cambridge-microsoft>).

## 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*, availability of advisers, temporary closures of the *CATAM room*) 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 2018 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](mailto: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, or if you need an adviser to help you debug your programs), you can email a query to the *CATAM Helpline*: [catam@maths.cam.ac.uk](mailto: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](mailto: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.<sup>12</sup>

In addition to the *Helpline*, at certain times of the year, e.g. in the period immediately before submission, advisers may be available in the *CATAM room*. As well as answering queries about general course administration, programming and/or MATLAB, you are allowed to ask advisers to help you debug your programs. The times when they will be available will be advertised in *CATAM News*.

*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 offer you a short “feedback” supervision on one of the *Core Projects* towards the end of the Lent Term.

## 5 Unfair Means, Plagiarism and Guidelines for Collaboration

You **must** work *independently* on the projects, both on the programming and on the write-ups, i.e. you must write and test all programs yourself, and all reports must be written independently. 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. However, 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.

*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.

---

<sup>12</sup> For example emails sent directly to the CATAM Course Director may be subject to a far longer delay in answering (and could end up either being missed altogether or consigned to `/dev/null`).



*Acceptable collaboration.* 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.* 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. Indeed, assuming that you are interpreting the phrase *occasional general discussion* in the spirit that it is written, then if you have got to the stage of wondering whether a discussion has reached the limit of acceptable collaboration, or you have started a legalistic deconstruction of the term *acceptable collaboration*, you are almost certainly at, or past, the limit.

*Example.* As an instance 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 *Examination Interview* or 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).

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.

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, whatever the circumstances, you will be penalised yourself.

## 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*. This is reproduced as Appendix A. 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* that are reproduced as Appendix B. 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 at the start of the Michaelmas Term. You will receive an email with instructions as to how to do this at the start of the Michaelmas Term.

In order to certify that you have *observed* these guidelines, you will be required to sign a submission form when you submit your write-ups, and you are advised to read it carefully; it is reproduced (subject to revision) as Appendix C. 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 you have breached any of the other guidelines, you will be summoned to an *Examination Interview* or an *Investigative Meeting* (see §5.2). Either case may lead to a change in the marks you receive for the Computational Projects.

***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 *Examination Interview* (see §5.2). The Computational Projects are considered to be a single piece of work within the Mathematical Tripos; therefore, **if it is concluded that you have used unfair means for the whole or any part of the Computational Projects the likely outcome is that you will receive a mark of 0 for the Computational Projects in their entirety.**

If the Chair of Examiners deems the unfair means to be sufficiently significant, an *Investigative Meeting* will be held (see §5.2). One outcome of such a meeting could be that the case is referred to the University Advocate, who may send the case to the Court of Discipline. According to the University guidance given to Cambridge students regarding discipline<sup>13</sup>

The Court has power to impose sentences of deprivation or suspension of membership of the University, deprivation or suspension of degree, rustication, and any other sentence which it considers lighter, including requesting the Vice-Chancellor to issue a revised class-list awarding a different class of degree than that initially awarded by the Examiners, and may order payment of compensation.

**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; if you find them 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 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.* For most cases of suspected plagiarism, collusion or other unfair means 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 any work is deemed to have been plagiarised or otherwise produced using unfair means, the Examiners have the power to award a mark of 0 for the Computational Projects *in their entirety*. At the discretion of the Examiners and if mitigating circumstances warrant, the Examiners may award a mark greater than 0.

---

<sup>13</sup> From <http://www.cambridgestudents.cam.ac.uk/new-students/rules-and-legal-compliance/discipline>.

*Investigative Meetings.* For the most serious cases of suspected plagiarism, collusion or other unfair means, the Chair of Examiners may summon a candidate to an *Investigative Meeting*. 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).<sup>14</sup>

*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 10th June 2019), 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 a hard-copy of your project reports you will be required to sign a submission form 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 C). 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.

### 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 sign your submission form;
- submit, with your submission form, your written reports and program printouts for every project for which you wish to gain credit;
- sign a submission list.

---

<sup>14</sup> For more information see

[https://www.plagiarism.admin.cam.ac.uk/files/investigative\\_2016.pdf](https://www.plagiarism.admin.cam.ac.uk/files/investigative_2016.pdf).

Please note as part of the submission process your work will be placed into plastic wallets, with the individual wallets being sent to different examiners; hence each project should have its own wallet. You can provide your own wallets (which will speed up submission) or use the wallets provided on the day. If a project will not fit into a single wallet, then re-read section §2.2.1 on *Project write-ups: advice on length*.

The location for handing in your work will be announced via *CATAM News* and email closer to the time.

For the core projects the submission date is

**Tuesday 22nd January 2019, 10am–4pm,**

while for the additional projects the submission date is

**Tuesday 30th April 2019, 10am–4pm.**

No submissions will be accepted before these times, and **4pm on 22nd January 2019 and 30th April 2019 are the final deadlines**.

After these times, projects may be submitted only under exceptional circumstances. If an extension is likely to be needed, a letter of application and explanation is required from your Director of Studies. The application should be sent to the *CATAM Director* by the submission date as detailed above.<sup>15</sup>

- Applications must demonstrate that there has been an unexpected development in the student's circumstances.
- Extensions are not normally granted past the Friday of submission week.

A student who is dissatisfied with the CATAM Director's decision, can request within 7 days of the decision, or by the submission date (extended or otherwise), whichever is earlier, that the Chair of the Faculty review the decision.

The Computational Projects Assessors Committee reserves the right to **reduce** the marks awarded for any projects which are submitted late (including electronic submission).

### 6.3 Electronic submission

You are also required to submit electronically copies of both your reports and your program source files. The electronic submission must be identical to the hard-copy submission. 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.

Full details of the procedure will be announced about one week before the submission deadlines via *CATAM News* and email, so please do *not* make enquiries about it until then.

---

<sup>15</sup> Alternatively, the University's procedure can be invoked via the *Standing Committee on Applications*; see the *Guidance Notes for Dissertation and Coursework extensions* (the link is to the 2017/18 version that was current in July 2018; you should refer to the updated version for 2018/19).

**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 the instructions provided by the University Information Service.<sup>16</sup> 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.

## **6.4 (Non)-return of written work**

We regret that students' submitted work cannot be returned to them after the examination; it must be retained in case of a query or an appeal at a later stage. You are recommended to keep at least an electronic version of your work (or even make a photocopy before submitting the hard-copy).

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 make a hard copy of the projects you have attempted.

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.

---

<sup>16</sup> See <https://password.csx.cam.ac.uk/forgotten-passwd>.

## A Appendix: University Statement on Plagiarism

The General Board, with the agreement of the Board of Examinations and the Board of Graduate Studies, has issued this guidance for the information of candidates, Examiners, and Supervisors.<sup>17</sup> It may be supplemented by course-specific guidance from Faculties and Departments.

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. It is both poor scholarship and a breach of academic integrity.

Examples of plagiarism include **copying** (using another person's language and/or ideas as if they are a candidate's own), by:

- **quoting verbatim** another person's work without due acknowledgement of the source;
- **paraphrasing** another person's work by changing some of the words, or the order of the words, without due acknowledgement of the source;
- **using ideas** taken from someone else without reference to the originator;
- **cutting and pasting** from the Internet to make a pastiche of online sources;
- **submitting someone else's work** as part of a candidate's own without identifying clearly who did the work. For example, buying or commissioning work via professional agencies such as 'essay banks' or 'paper mills', or not attributing research contributed by others to a joint project.

Plagiarism might also arise from **colluding** with another person, including another candidate, other than as permitted for joint project work (i.e. where collaboration is concealed or has been forbidden). A candidate should include a general acknowledgement where he or she has received substantial help, for example with the language and style of a piece of written work.

Plagiarism can occur in respect to all types of sources and media:

- text, illustrations, musical quotations, mathematical derivations, computer code, etc;
- material downloaded from websites or drawn from manuscripts or other media;
- published and unpublished material, including lecture handouts and other students' work.

Acceptable means of acknowledging the work of others (by referencing, in footnotes, or otherwise) is an essential component of any work submitted for assessment, whether written examination, dissertation, essay, registration exercise, or group coursework. The most appropriate method for attribution of others' work will vary according to the subject matter and mode of assessment. Faculties or Departments should issue written guidance on the relevant scholarly conventions for submitted work, and also make it clear to candidates what level of acknowledgement might be expected in written examinations. Candidates are required to familiarize themselves with this guidance, to follow it in all work submitted for assessment, whether written paper or submitted essay, and may be required to sign a declaration to that effect. If a candidate has any outstanding queries, clarification should be sought from her or his Director of Studies, Course Director, or Supervisor as appropriate.

---

<sup>17</sup> For the latest version of this statement see

<https://tinyurl.com/UC-plagiarism-statement>

Failure to conform to the expected standards of scholarship (e.g. by not referencing sources) in examinations or assessed work may affect the mark given to the candidate's work. In addition, 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 University makes use of text-matching software for the purpose of plagiarism education and detection, and reserves the right to submit a candidate's work to such a service. For this purpose, candidates consent to the submission of their papers to the service and for the submitted papers to form part of the service's comparative source work database. To facilitate use of the service, students (and participating Examiners and Assessors) may be required to agree to the service provider's end-user agreement and provide a limited amount of personal data upon registration to the service, for instance, their name, email address, and course details.

### **Discipline Regulation 7**

No candidate shall make use of unfair means in any University examination. Unfair means shall include plagiarism<sup>18</sup> and, unless such possession is specifically authorised, 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. (*Statutes and Ordinances 2017, p.194-5.*)

Note that in all documentation produced prior to October 2015 this was known as Discipline Regulation 6.

---

<sup>18</sup> 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.



## B Appendix: Faculty of Mathematics Guidelines on Plagiarism

For the latest version of these guidelines<sup>19</sup> please see

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

### University Resources

The University publishes information on *Plagiarism and good academic practice*, including

- a *University-wide statement on plagiarism*,
- information for students, covering *Your responsibilities*, *Why does plagiarism matter?* and *Collusion*
- information about *Referencing* and *Study skills*,
- information on *Resources and support*,
- information on the *University's statement on proofreading*,
- *FAQs*.

There are references to the University statement

- in the **Part IB** and **Part II** Computational Project Manuals,
- in the Part III Essay booklet (linked from the **Part III Essays** page),
- in the Computational Biology Handbook (linked from the **Computational Biology Course** page).

*Please read the University statement carefully; it is your responsibility to read and abide by this statement.*

### The Faculty Guidelines

The guidelines below are provided by the Faculty to help students interpret what the University Statement means for Mathematics. However neither the University statement nor the Faculty guidelines supersede the University's Regulations as set out in the **Statutes and Ordinances**. If you are unsure as to the interpretation of the statement, or guidelines, or the **Statutes and Ordinances**, you should ask your Director of Studies or Course Director (as appropriate).

#### What is plagiarism?

Plagiarism can be defined as **the unacknowledged use of the work of others as if this were your own original work**. In the context of any University examination, this amounts to **passing off the work of others as your own to gain unfair advantage**.

Such use of unfair means will not be tolerated by the University or the Faculty. If detected, the penalty may be severe and may lead to failure to obtain your degree. This is in the interests of the vast majority of students who work hard for their degree through their own efforts, and it is essential in safeguarding the integrity of the degrees awarded by the University.

---

<sup>19</sup> These are as published as of July 2018 and may be subject to change in October 2018.



## Checking for plagiarism

Faculty Examiners will routinely look out for any indication of plagiarised work. They reserve the right to make use of specialised detection software if appropriate (the University subscribes to *Turnitin Plagiarism Detection Software*). See also the Board of Examinations' statement on [How the University detects and disciplines plagiarism](#).

## The scope of plagiarism

Plagiarism may be due to

- **copying** (this is using another person's language and/or ideas as if they are your own);
- **collusion** (this is collaboration either where it is forbidden, or where the extent of the collaboration exceeds that which has been expressly allowed).

## How to avoid plagiarism

Your course work, essays and projects (for Parts IB, II and III, the M.Phil. etc.), are marked on the assumption that it is your own work: i.e. on the assumption that the words, diagrams, computer programs, ideas and arguments are your own. Plagiarism can occur if, without suitable acknowledgement and referencing, you take any of the above (i.e. words, diagrams, computer programs, ideas and arguments) from books or journals, obtain them from unpublished sources such as lecture notes and handouts, or download them from the web.

Plagiarism also occurs if you submit work that has been undertaken in whole or part by someone else on your behalf (such as employing a 'ghost writing service'). Furthermore, you should not deliberately reproduce someone else's work in a written examination. These would all be regarded as plagiarism by the Faculty and by the University.

In addition you should not submit any work that is substantially the same as work you have submitted, or are concurrently submitting, for any degree, diploma or similar qualification at any university or similar institution.

However, it is often the case that parts of your essays, projects and course-work will be based on what you have read and learned from other sources, and it is important that in your essay or project or course-work you show exactly where, and how, your work is indebted to these other sources. The golden rule is that **the Examiners must be in no doubt as to which parts of your work are your own original work and which are the rightful property of someone else.**

A good guideline to avoid plagiarism is not to repeat or reproduce other people's words, diagrams or computer programs. If you need to describe other people's ideas or arguments try to paraphrase them in your own words (and remember to include a reference). Only when it is absolutely necessary should you include direct quotes, and then these should be kept to a minimum. You should also remember that in an essay or project or course-work, it is not sufficient merely to repeat or paraphrase someone else's view; you are expected at least to evaluate, critique and/or synthesise their position.

In slightly more detail, the following guidelines may be helpful in avoiding plagiarism.

## Quoting

A quotation directly from a book or journal article is acceptable in certain circumstances, provided that it is referenced properly:

- short quotations should be in inverted commas, and a reference given to the source;
- longer pieces of quoted text should be in inverted commas and indented, and a reference given to the source.

Whatever system is followed, you should additionally list all the sources in the bibliography or reference section at the end of the piece of work, giving the full details of the sources, in a format that would enable another person to look them up easily. There are many different styles for bibliographies. Use one that is widely used in the relevant area (look at papers and books to see what referencing style is used).

## Paraphrasing

Paraphrasing means putting someone else's work into your own words. Paraphrasing is acceptable, provided that it is acknowledged. A rule of thumb for acceptable paraphrasing is that an acknowledgement should be made at least once in every paragraph. There are many ways in which such acknowledgements can be made (e.g. "Smith (2001) goes on to argue that ..." or "Smith (2001) provides further proof that ..."). As with quotation, the full details of the source should be given in the bibliography or reference list.

## General indebtedness

When presenting the ideas, arguments and work of others, you must give an indication of the source of the material. You should err on the side of caution, especially if drawing ideas from one source. If the ordering of evidence and argument, or the organisation of material reflects a particular source, then this should be clearly stated (and the source referenced).

## Use of web sources

You should use web sources as if you were using a book or journal article. The above rules for quoting (including 'cutting and pasting'), paraphrasing and general indebtedness apply. Web sources must be referenced and included in the bibliography.

## Collaboration

Unless it is expressly allowed, collaboration is collusion and counts as plagiarism. Moreover, as well as not copying the work of others you should not allow another person to copy your work.

## Links to University Information

- Information on *Plagiarism and good academic practice*, including
  - *Students' responsibilities*,
  - *Information for staff*.

## C Appendix: Example Submission Form

### PART IB MATHEMATICAL TRIPOS 2018-19

#### CORE COMPUTATIONAL PROJECTS

#### STATEMENT OF PROJECTS SUBMITTED FOR EXAMINATION CREDIT

Name: .....

College: ..... CRSid User Identifier: .....

**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. The project number should be written clearly in the **top left hand corner** of the first sheet of the write-up. Leave a space 11 cm wide by 5 cm deep in the **top right hand corner** of the first sheet.
3. Complete the declaration overleaf before arriving at the submissions desk.
4. During the submission process your work will be placed into plastic wallets. The individual wallets will be sent to different examiners, so **each project should have its own wallet**.
5. Put your work into the plastic wallets so that the top is at the opening.
6. Without damaging your work or over-filling try not to use more than one wallet per project. (If the pages will not go into the wallet flat, you may need to use more than one wallet.) Ensure that the write-up is at the front and the program listing at the back; if you have used two wallets for a project, they should be securely attached to each other and the second one should contain the program listing.
7. Remember that many others are likely to hit the submissions desk 30 minutes before the deadline. You can avoid a stressful situation by submitting early.

#### 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<sup>20</sup> 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. It could result in all your marks for the Computational Projects being removed and also render you liable to further sanctions from the Examiners or the University Courts.*

---

<sup>20</sup> 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 written 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

$$x_N = f(x_{N-1}), \quad (1)$$

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

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

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  $N \rightarrow \infty$  is said, for the purposes of this project,<sup>1</sup> to have order of convergence  $p$  ( $\geq 1$ ) if

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

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  $N^{\text{th}}$  iterate is defined as  $\epsilon_N = x_N - x_*$ .<sup>2</sup> If the method is convergent, i.e.  $\epsilon_N \rightarrow 0$  as  $N \rightarrow \infty$ , it is said to be  $p^{\text{th}}$ -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.

---

<sup>1</sup> A more inclusive definition of order of convergence, referred to as the Q-order of convergence, might be

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

<sup>2</sup> 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

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

and

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

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

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

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

## 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  $0.5 \times 10^{-5}$ , 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  $\delta$  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.

$$f(x) = x - h(F(x)), \quad (6)$$

for some function<sup>3</sup>  $h(F)$  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.

---

<sup>3</sup> Or *functional*.

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

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

in (6), so that

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

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.<sup>4</sup>
- (ii) Determine 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

$$h(F) = \frac{1}{20} F, \quad (8a)$$

in (6), so that

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

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  $\epsilon_N$  is asymptotic to  $40/(7N)$  as  $N \rightarrow \infty$ .

## Newton-Raphson Iteration

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

$$f(x) = x - h(F, F', F'', \dots). \quad (9a)$$

In Newton-Raphson iteration

$$h = \frac{F}{F'}. \quad (9b)$$

---

<sup>4</sup> 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  $\epsilon$ .

## 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  $\text{GF}(p)$ , 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  $10 \times 10$ ). 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  $a$ ,  $1 \leq a \leq p-1$ , you will need to know its inverse  $a^{-1}$ ,  $1 \leq a^{-1} \leq p-1$ , 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  $1 \leq a \leq p-1$ , all values of  $b$  in the range  $1 \leq b \leq p-1$  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  $\Theta(f(p))$ , meaning of the order of magnitude of  $f(p)$ . To be exact, a function  $g(p)$  is  $\Theta(f(p))$  if there are positive constants  $c$  and  $C$  such that  $c \leq g(p)/f(p) \leq C$  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 *row echelon* form if

- for some  $r$ ,  $1 \leq r \leq m$ , the last  $m-r$  rows have only zero entries;
- for each  $i$ ,  $1 \leq i \leq r$ , there is a number  $1 \leq l(i) \leq n$  such that  $m_{ij} = 0$  for  $j < l(i)$  and  $m_{ij} = 1$  for  $j = l(i)$ ;
- $l(1) < l(2) < \dots < l(r)$ ;

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

Here is a  $4 \times 5$  matrix which is in row echelon form, if we take the entries mod 7; in this case  $r = 3$ ,  $l(1) = 1$ ,  $l(2) = 3$ ,  $l(3) = 4$ .

$$\begin{pmatrix} 1 & 4 & 0 & 7 & 5 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \pmod{7}$$

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 row echelon form.

**Question 3** Write a program to turn a matrix into 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.

$$A_1 = \begin{pmatrix} 0 & 1 & 7 & 2 & 10 \\ 8 & 0 & 2 & 5 & 1 \\ 2 & 1 & 2 & 5 & 5 \\ 7 & 4 & 5 & 3 & 0 \end{pmatrix} \begin{matrix} \text{(first mod 11} \\ \text{then mod 19)}, \end{matrix} A_2 = \begin{pmatrix} 6 & 16 & 11 & 14 & 1 & 4 \\ 7 & 9 & 1 & 1 & 21 & 0 \\ 8 & 2 & 9 & 12 & 17 & 7 \\ 2 & 19 & 2 & 19 & 7 & 12 \end{pmatrix} \pmod{23}$$

## 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 reducing  $A$  to row echelon form and then expressing  $x_{l(1)}, x_{l(2)}, \dots, x_{l(r)}$  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 following matrices.

$$B_1 = \begin{pmatrix} 4 & 6 & 5 & 2 & 3 \\ 5 & 0 & 3 & 0 & 1 \\ 1 & 5 & 7 & 1 & 0 \\ 5 & 5 & 0 & 3 & 1 \\ 2 & 1 & 2 & 4 & 0 \end{pmatrix} \begin{matrix} \text{(first mod 13} \\ \text{then mod 17)}, \end{matrix} B_2 = \begin{pmatrix} 3 & 7 & 19 & 3 & 9 & 6 \\ 10 & 2 & 20 & 15 & 3 & 0 \\ 14 & 1 & 3 & 14 & 11 & 3 \\ 26 & 1 & 21 & 6 & 3 & 5 \\ 0 & 1 & 3 & 19 & 0 & 3 \end{pmatrix} \pmod{23}$$

Let  $U$  be a subspace of the space of row vectors  $F^n$ . The annihilator  $U^\circ$  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  $U^\circ$  is the kernel of  $A$ .

**Question 5** State the relationship between the dimensions of  $U$  and  $U^\circ$ .

If  $S$  is a subspace of the space of column vectors, then we make an analogous definition of  $S^\circ$  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  $U^\circ$  where we work mod 19 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 11 with  $U$  the row space of  $A_1$  and  $W$  the row space of  $B_1$ .
- 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 23.

**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  $\text{GF}(p)$ ?

## 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 gained/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 <sup>1</sup>	marks awarded <sup>2</sup>
<b>Question 1</b> 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). <i>Modification: [approx. 3 lines]</i> <sup>3</sup>	C1+M1	
<b>Question 2</b> Do not include negligible terms, or terms specific to your particular implementation, in your final answer. [approx. 3 lines] <sup>3</sup>	C0+M1	
<b>Question 3</b> 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] <sup>3</sup>	C2+M1	
<b>Question 4</b> Produce evidence of output from your program and state your answers clearly. Explain briefly how your program works. [approx. 15 lines] <sup>3</sup>	C2+M1	
<b>Question 5</b> No proofs are required. [approx. 1 line] <sup>3</sup>	C0+M1	
<b>Question 6</b> 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] <sup>3</sup>	C1+M1	
<b>Question 7</b> 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] <sup>3</sup>	C2+M3	
<b>Question 8</b> No programming is required for this question. [approx. 3 lines] <sup>3</sup>	C0+M1	
<b>Excellence marks.</b> These are awarded for, <i>among other things</i> , mathematical clarity and good, clear output (graphs and tables) — see the introduction to the Project Manual.	E2	
Total Raw Marks	20	
<b>Total Tripos Marks</b>	<b>40</b>	

<sup>1</sup> C#, M# and E#: *Computational*, *Mathematical* and *Excellence* marks respectively.

<sup>2</sup> For use by the assessor

<sup>3</sup> 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 theory covered in Part IA Differential Equations.*

### 1 Background Theory

The aim in the first part of this project (§2) is to study the performance of three different numerical methods for step-by-step integration of a first-order ordinary differential equation (ODE)

$$\frac{dy}{dx} = f(x, y) \quad (1)$$

with a given initial condition

$$y = Y_0 \text{ at } x = x_0 \quad (2)$$

for specified  $x_0$  and  $Y_0$ . A case has been chosen where the exact solution  $y(x)$  can be found in simple analytic form. In the second part of this project (§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 is the simplest method. It employs the scheme

$$Y_{n+1} = Y_n + hf(x_n, Y_n) \quad (3)$$

where  $Y_n$  denotes the numerical solution at  $x_n \equiv x_0 + nh$ , that is, at the  $n$ th step with step length  $h$ . The Euler method has first-order accuracy, which means that the local truncation error  $e_{n+1}$  is  $O(h^2)$  as  $h \rightarrow 0$ . The local truncation error is found by setting  $Y_n = y(x_n)$  (the exact solution at  $x_n$ ), computing  $Y_{n+1}$  using equation (3), then calculating  $e_{n+1} = Y_{n+1} - y(x_{n+1})$ . On the other hand, the global error in the numerical solution using  $n + 1$  steps starting from the initial condition (2) is denoted by  $E_{n+1}$ . The Euler method is called a *single-step* method, since  $Y_{n+1}$  is obtained from the previous step  $Y_n$ .

- (b) The **Leapfrog** (LF) method employs the scheme

$$Y_{n+1} = Y_{n-1} + 2hf(x_n, Y_n) \quad (4)$$

and has second-order accuracy, i.e.  $e_{n+1}$  is  $O(h^3)$  as  $h \rightarrow 0$ . It is a multi-step method, using both  $Y_{n-1}$  and  $Y_n$  to obtain  $Y_{n+1}$ , and the first step must be taken by a single-step method, e.g. the Euler method.

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

$$Y_{n+1} = Y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (5)$$

where

$$k_1 = hf(x_n, Y_n) \quad (6)$$

$$k_2 = hf(x_n + \frac{1}{2}h, Y_n + \frac{1}{2}k_1) \quad (7)$$

$$k_3 = hf(x_n + \frac{1}{2}h, Y_n + \frac{1}{2}k_2) \quad (8)$$

$$k_4 = hf(x_n + h, Y_n + k_3) \quad (9)$$

and has fourth-order accuracy, i.e.  $e_{n+1}$  is  $O(h^5)$  as  $h \rightarrow 0$ . When the RK4 method is used on coupled ODEs, each of  $f$ ,  $Y_n$ ,  $k_1$ ,  $k_2$ ,  $k_3$  and  $k_4$  become vectors of the same dimension as the number of coupled ODEs.

The theoretical background for the accuracy and stability 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 Comparison of the numerical methods for solving ODEs

The specific case to be studied in detail in this section is equation (1) with

$$f(x, y) = -4y + 3e^{-x} \quad (10)$$

and initial condition

$$y(0) = 0. \quad (11)$$

This has the exact solution

$$y(x) = e^{-x} - e^{-4x}. \quad (12)$$

**Programming Task:** Write program(s) to implement each of the methods (a), (b) and (c) above.

### 2.1 Stability

**Question 1** Starting with  $x_0 = 0$ ,  $Y_0 = 0$ , use the LF method (with the first step taken by the Euler method) to integrate the ODE (1), (10) with initial condition (11) numerically from  $x = 0$  to  $x = 10$  with  $h = 0.4$  [i.e. for  $n$  up to 25]. Tabulate the values of  $x_n$ , the numerical solution  $Y_n$ , the analytic solution  $y(x_n)$  from (12) and the global error  $E_n = Y_n - y(x_n)$ . You should find that the numerical result is unstable over this range of  $x$ : the error oscillates wildly with magnitude ultimately growing exponentially, proportional to  $e^{\gamma x}$  where the ‘growth rate’  $\gamma$  is a constant which you should estimate.

Repeat with  $h = 0.2$ ,  $0.1$  and  $0.05$  [i.e. for  $n$  up to 50, 100 and 200 respectively], not necessarily tabulating the output at every step. Comment on the effect of reducing  $h$  on the size of the instability, and on its growth rate.

**Question 2** (i) Find the analytic solution to the LF *difference* equation

$$Y_{n+1} = Y_{n-1} + 2h \left[ -4Y_n + 3 \left( e^{-h} \right)^n \right] \quad (13)$$

with

$$Y_0 = 0, \quad Y_1 = 3h \quad (\text{from the Euler method}). \quad (14)$$

(ii) Hence explain why instability occurs, and how its growth rate depends on  $h$ .

(iii) Show that in the limit  $h \rightarrow 0$ ,  $n \rightarrow \infty$  with  $x_n \equiv nh$  fixed, the solution of the LF-difference-equation problem (13)–(14) found in part (i) converges to the solution (12) of the differential-equation problem (1), (10), (11). Does this mean that the instability can be suppressed by using a sufficiently small value for  $h$ ?

### 2.2 Accuracy

**Question 3** Integrate the ODE (1), (10) with initial condition (11) numerically up to  $x = 4$ , using both the Euler and the RK4 method with  $h = 0.4$ . Tabulate both numerical solutions  $Y_n$  against  $x_n$ , and plot them with the exact solution  $y(x_n)$  superimposed.

**Question 4** For each of the Euler, LF and RK4 methods, tabulate the global error  $E_n$  at  $x_n = 0.4$  against  $h = 0.4/n$  for  $n = 2^k$ ,  $k = 0, 1, 2, \dots, 15$ , and plot a log–log graph of  $|E_n|$  against  $h$  over this range. Comment on the relationship of these results to the theoretical accuracy of the methods.

### 3 Numerical solutions of second-order ODEs

This section investigates the response of a simple harmonic oscillator with (possibly nonlinear) damping to a driving force.

The equation to be studied is

$$\frac{d^2y}{dt^2} + \frac{d}{dt} \left( \gamma y + \frac{1}{3} \delta^3 y^3 \right) + \Omega^2 y = a \sin(\omega t) \quad (15)$$

where  $\gamma$ ,  $\delta$ ,  $\Omega$ ,  $\omega$  and  $a$  are non-negative real constants and  $t$  and  $y$  are real variables. In the case of purely linear damping,  $\delta = 0$ , it can of course be solved analytically.

**Question 5** Find the analytic general solution to equation (15) for the linear, lightly damped case with  $\delta = 0$ ,  $0 < \gamma < 2\Omega$ . Show that

$$y \rightarrow A_s \sin(\omega t - \phi_s) \quad \text{as } t \rightarrow \infty \quad (16)$$

and write down expressions for the ‘steady-state’ amplitude  $A_s$  and the ‘steady-state’ phase shift  $\phi_s$  in terms of  $\gamma$ ,  $\Omega$ ,  $\omega$  and  $a$ .

Equation (15) can be rewritten as a pair of coupled first-order ODEs for

$$y^{(1)}(t) \equiv y(t) \quad \text{and} \quad y^{(2)}(t) \equiv \frac{dy(t)}{dt}, \quad (17)$$

namely

$$\frac{dy^{(1)}}{dt} = f^{(1)}(t, y^{(1)}, y^{(2)}) \equiv y^{(2)}, \quad (18)$$

$$\frac{dy^{(2)}}{dt} = f^{(2)}(t, y^{(1)}, y^{(2)}) \equiv -\gamma y^{(2)} - \delta^3 [y^{(1)}]^2 y^{(2)} - \Omega^2 y^{(1)} + a \sin(\omega t), \quad (19)$$

which can then be solved using either the Euler or the RK4 method. In this part of the project you are to use RK4, and take as initial conditions

$$y = \frac{dy}{dt} = 0 \quad \text{at } t = 0. \quad (20)$$

**Programming Task:** Write a program to solve equation (15) with initial conditions (20) using the RK4 method.

The next two questions are concerned with the particular case of equation (15) with  $\delta = 0$ ,  $\Omega = 1$  and  $a = 1$ , i.e.,

$$\frac{d^2y}{dt^2} + \gamma \frac{dy}{dt} + y = \sin(\omega t). \quad (21)$$

**Question 6** Write down the analytic solution of (21) for general  $\gamma$  ( $< 2$ ) and  $\omega$  subject to the initial conditions (20). Taking  $\gamma = 1$  and  $\omega = \sqrt{3}$ , use your program to compute  $Y_n$  for  $t$  up to 10 with  $h = 0.4$  [i.e. for  $n$  up to 25], and tabulate the numerical solution  $Y_n$ , the analytic solution  $y(t_n)$  and the global error  $E_n \equiv Y_n - y(t_n)$  against  $t_n$ . Repeat with both  $h = 0.2$  and  $h = 0.1$  [integrating up to  $t = 10$ , i.e. for  $n$  up to 50 and 100 respectively], not necessarily presenting all the output. Comment on the errors.

**Question 7** Use your RK4 program (with suitable  $h$ ) to generate and plot numerical solutions of (20)–(21) up to  $t = 40$  for  $\omega = 1$  and  $\gamma = 0.25, 0.5, 1.0$  and  $1.9$ , checking that they agree with the analytic solutions. Do likewise for  $\omega = 2$  and the same values of  $\gamma$ . Explain the differences between the various cases in terms of the mathematics and the physics of the system under investigation.

The last question considers a case with nonlinear damping,

$$\frac{d^2y}{dt^2} + \frac{d}{dt} \left( \frac{1}{3} \delta^3 y^3 \right) + y = \sin t, \quad (22)$$

for which an analytic solution is not available. The initial conditions are as before,

$$y = \frac{dy}{dt} = 0 \quad \text{at } t = 0. \quad (23)$$

**Question 8** For  $\delta = 0.25, 0.5, 1.0$  and  $20$ , use your RK4 program to generate and plot numerical solutions to (22)–(23) for  $t$  up to  $60$ , using suitable value(s) of  $h$  (justify your choice). Comment on the solutions, comparing them with each other and with those of Question 7 for  $\omega = 1$ .

*Hint:* it *may* be helpful to observe that when  $\delta$  is ‘small’, equation (22) has a  $2\pi$ -periodic solution of the form

$$y = \sum_{n=-1}^{\infty} \delta^n y_n(t) \quad (24)$$

where each  $y_n(t)$  is periodic in  $t$  with period  $2\pi$  and

$$y_{-1}(t) = A \cos t, \quad y_0(t) = B \sin t + C \sin 3t \quad (25)$$

for suitable values of the constants  $A, B$  and  $C$  [recall that  $\cos^3 \theta = \frac{3}{4} \cos \theta + \frac{1}{4} \cos 3\theta$ , and note that to determine  $y_0$  completely it is necessary to consider terms of order  $\delta$ ]. What if  $\delta$  is ‘large’?



## 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 gained/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 <sup>1</sup>	marks awarded <sup>2</sup>
<b>Programming task</b> <i>Program</i> : for instructions regarding printouts and what needs to be in the write-up, refer to the introduction to the project manual.		
<b>Question 1</b> <i>Tables</i> : for presentation and layout, refer to the introduction. [approx. 2 lines] <sup>3</sup>	C2+M0	
<b>Question 2</b> <i>Analytic solution</i> : do not include trivial steps in your worked answer. [approx. 12 lines] <sup>3</sup>	C0+M3.5	
<b>Question 3</b> <i>Graphs</i> : you may use one graph or two.	C1+M0	
<b>Question 4</b> <i>Graphs</i> : you may use one graph, or two, or three. <i>Comments</i> : what can be said about how the global error of each method depends on $h$ ? How is this reflected in the plots? [approx. 4 lines] <sup>3</sup>	C1+M0 C0+M1	
<b>Question 5</b> <i>Analytic solution</i> : do not include trivial steps in your worked answer; be sure to specify $A_s$ and $\phi_s$ unambiguously. [approx. 4 lines] <sup>3</sup>	C0+M1	
<b>Question 6</b> <i>Analytic and numerical solutions compared</i> : the purpose of this step is to check that the program works and gives accurate answers ('validation'). Do the errors behave as expected when $h$ is decreased? [approx. 2 lines] <sup>3</sup>	C2+M1	
<b>Question 7</b> <i>Comments</i> : first identify the salient features of the plots. Examine the nature of the functions that you are plotting: what are their components and how do these contribute to the overall solutions? Then use mathematical arguments (cf. the Part IA course <i>Differential Equations</i> ) to explain the behaviour of the plots; link to the theory of the physical system under investigation. [approx. 30 lines] <sup>3</sup>	C1+M2	
<b>Question 8</b> <i>Numerical solutions</i> : explain why you are satisfied that your chosen value(s) of $h$ will deliver sufficiently accurate results. <i>Comments</i> : identify the key similarities and differences between the various solutions, and with the help of the hint, or otherwise, try to explain them mathematically and/or physically. [approx. 20 lines] <sup>3</sup>	C1+M0 C0+M1.5	
<b>Excellence marks</b> awarded for, <i>among other things</i> , mathematical clarity and good, clear output (graphs and tables) — see the introduction to the project manual.	E2	
<b>Total Raw Marks</b>	20	
<b>Total Tripos Marks</b>	40	

<sup>1</sup> C#+M# : computational and mathematical marks

<sup>2</sup> For use by the assessor

<sup>3</sup> 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

## 2.1 The Diffusion Equation

*The Part IB Methods course is relevant.*

### 1 Background

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

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

where  $\theta(x, t)$  is the temperature (in kelvin) averaged over the cross-section (at distance  $x$  metres along the bar and time  $t$  seconds), and  $K$  is a positive constant, the so-called *thermal diffusivity* (measured in metres-squared per second). 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  $-A k \partial \theta / \partial x(x, t)$ , where  $A$  is the (constant) cross-sectional area and  $k$  the (constant) thermal conductivity;
- (iii) the total heat in  $a < x < b$  is

$$A \int_a^b \sigma \rho \theta(x, t) \, dx, \quad (2)$$

where  $\sigma$  is the (constant) specific heat and  $\rho$  the (constant) density, with its rate of change

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

being equal to the net heat flux in

$$-A k \frac{\partial \theta}{\partial x}(a, t) + A k \frac{\partial \theta}{\partial x}(b, t) = A k \int_a^b \frac{\partial^2 \theta}{\partial x^2}(x, t) \, dx \quad (4)$$

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

### 2 Formulation

Suppose that for  $t < 0$ , the bar is at uniform temperature  $\theta_0$ , and that for  $t \geq 0$ , the temperature of one end ( $x = 0$ ) is suddenly altered to a different value  $\theta_1$  and thereafter maintained at this value, while the other end ( $x = L$ ) is either insulated or maintained at constant temperature. Equation (1) is therefore to be solved for  $t > 0$  subject to the initial condition

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

and to the boundary conditions

$$\theta(0, t) = \theta_1 \quad \text{for } t > 0, \quad (6)$$

and either

$$\frac{\partial \theta}{\partial x}(L, t) = 0 \quad \text{for } t > 0 \quad (7a)$$

(i.e. vanishing heat flux at the insulated end), or

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

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

### 3 Analytic Solutions

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

$$\frac{\partial \theta}{\partial x}(x, t) \rightarrow 0 \quad \text{or} \quad \theta(x, t) \rightarrow \theta_0 \quad \text{as } x \rightarrow \infty, \text{ respectively.} \quad (8)$$

If

$$\theta(x, t) = \theta_0 + (\theta_1 - \theta_0) F(x, t), \quad (9)$$

explain with the help of dimensional analysis why in both cases  $F$  must have the ‘similarity’ form

$$F(x, t) = f(\xi), \quad \xi = \frac{x}{(Kt)^{1/2}}. \quad (10)$$

Show that in both cases

$$f(\xi) = \operatorname{erfc}\left(\frac{1}{2}\xi\right) \equiv \frac{2}{\sqrt{\pi}} \int_{\xi/2}^{\infty} \exp(-u^2) du. \quad (11)$$

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

$$x = LX, \quad t = L^2 K^{-1} T, \quad \theta(x, t) = \theta_0 + (\theta_1 - \theta_0) U(X, T), \quad (12)$$

in terms of which the diffusion equation (1) becomes

$$U_T = U_{XX} \quad \text{for } T > 0, \quad 0 < X < 1, \quad (13)$$

with initial condition

$$U(X, 0) = 0 \quad \text{for } 0 < X < 1, \quad (14)$$

and boundary conditions

$$U(0, T) = 1 \quad \text{for } T > 0, \quad (15)$$

and either

$$U_X(1, T) = 0 \quad \text{for } T > 0, \quad (16a)$$

or

$$U(1, T) = 0 \quad \text{for } T > 0. \quad (16b)$$

**Question 2** First find an analytic solution of the fixed-endpoint-temperature problem (13)–(15) and (16b) in the form

$$U(X, T) = 1 - X + \sum_{n \geq 1} g_n(T) \sin(n\pi X), \quad (17)$$

where the  $g_n(T)$  are to be found. Adapt this method to obtain an (infinite-series) analytic solution of the insulated-end problem (13)–(16a) of the form

$$U(X, T) = U_s(X) + \sum_{n \geq 1} G_n(T) H_n(X), \quad (18)$$

for suitable functions  $U_s(X)$ ,  $G_n(T)$  and  $H_n(X)$ .

**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  $X = 0.125n$ ,  $n = 0, 1, \dots, 8$ , and also tabulate the semi-infinite solution (10)–(11) evaluated at these values of  $T$  and  $X$ .<sup>†</sup> Plot the non-dimensionalised temperature profiles,  $U$ , against  $X$ , for all three at  $T = 0.0625, 0.125, 0.25, 0.5, 1.0$  and  $2.0$ . Also plot the non-dimensionalised heat flux  $-U_X$  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  $T \geq 0.0625$ ; 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.

## 4 Numerical Integration

The insulated-end problem (13)–(16a) is now to be solved numerically as follows. Let the domain  $0 \leq X \leq 1$  be divided into  $N$  intervals, each of length  $\delta X = 1/N$ , and let  $U_T$  be approximated by a first-order forward difference in time:

$$\frac{\partial U(X, T)}{\partial T} = \frac{U(X, T + \delta T) - U(X, T)}{\delta T} + O(\delta T), \quad (19)$$

and  $U_{XX}$  by a second-order central difference in space at the current time:

$$\frac{\partial^2 U(X, T)}{\partial X^2} = \frac{U(X + \delta X, T) - 2U(X, T) + U(X - \delta X, T)}{(\delta X)^2} + O((\delta X)^2), \quad (20)$$

giving the numerical scheme

$$U_n^{m+1} = U_n^m + C [U_{n+1}^m - 2U_n^m + U_{n-1}^m], \quad (21)$$

where  $U_n^m$  is an approximation to  $U(n\delta X, m\delta T)$  and  $C = \delta T / (\delta X)^2$  (the so-called *Courant number*). The derivative boundary condition (16a) can be incorporated by solving (21) for  $1 \leq n \leq N$  with  $U_{N+1}^m = U_{N-1}^m$  for all  $m \geq 0$ ; why? You should take  $U_0^0 = 0.5$ ; why?

### Question 3

**Programming Task.** Write a program to implement this numerical scheme, and run it with  $N = 8, 16, 32$  and  $C = \frac{1}{12}, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}$  and  $\frac{2}{3}$ . For the case  $N = 8, C = \frac{1}{2}$ ,

- (i) tabulate both the analytic and the numerical solutions, and the value of the error, at  $T = 0.125, 0.25, 0.5$  and  $1.0$ ;
- (ii) plot on the same graph both the analytic and the numerical solutions for  $T = 0.0625, 0.125, 0.25, 0.5, 1.0$  and  $2.0$ .

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

### Reference

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

---

<sup>†</sup> Note that there is a MATLAB function `erfc`.

## 2.2 Schrödinger's Equation

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

### 1 Introduction

Schrödinger's [time-independent] wave equation for a single particle of mass  $m$  and energy  $\epsilon$ , moving in one dimension in a given [real] potential  $v(x)$ , is

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

where  $\psi(x)e^{-i\epsilon t/\hbar}$  is the time-dependent wave function representing the state of the system, and  $2\pi\hbar$  is Planck's constant. If we measure energy in units of  $\epsilon_1$ , say, and define a dimensionless co-ordinate  $X = \sqrt{2m\epsilon_1} x/\hbar$ , then the equation can be simplified to

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

where  $E = \epsilon/\epsilon_1$  (a constant) and  $V(X) = v(x)/\epsilon_1$  are the dimensionless total and potential energies, and  $Y(X) = \psi(x)$  is the stationary wave function.

In order to represent a bound state,  $Y(X)$  must tend to zero as  $|X| \rightarrow \infty$ , its real and imaginary parts doing so monotonically for all sufficiently large positive or negative  $X$ , and sufficiently fast that the wave function is 'normalisable', i.e.  $\int_{-\infty}^{\infty} |Y(X)|^2 dX$  is finite. This is possible only for certain values of  $E$ , the 'eigenvalues'. The aim of this project is to determine a few of these eigenvalues, and their corresponding 'eigenfunctions'  $Y(X)$ , numerically using 'forward shooting' — finding by trial-and-error values of  $E$ ,  $Y(0)$  and  $Y'(0)$  which give a solution of (2) with appropriate behaviour as  $|X| \rightarrow \infty$ .

### 2 Harmonic oscillator

We first consider the harmonic oscillator potential

$$V(X) = X^2. \quad (3)$$

Two independent analytic solutions of (2) with this potential and any  $E$  can be found in the form  $\exp(-X^2/2)$  times a series in even or odd powers of  $X$ :

$$Y_e(X) \equiv \exp(-X^2/2) \sum_{n=0}^{\infty} c_n X^{2n}, \quad Y_o(X) \equiv \exp(-X^2/2) \sum_{n=0}^{\infty} d_n X^{2n+1}. \quad (4)$$

where

$$c_{n+1} = \frac{4n+1-E}{(2n+2)(2n+1)} c_n, \quad d_{n+1} = \frac{4n+3-E}{(2n+3)(2n+2)} d_n \quad (5)$$

(see for example [1], §13 or [2], §2.3 or [3], chapter 5 G) and without loss of generality  $c_0 = d_0 = 1$ . If  $E$  is not an odd positive integer, neither series terminates and it can be shown that both  $Y_e(X)$  and  $Y_o(X)$  become unbounded as  $X \rightarrow \infty$ , asymptoting to  $C_{e,o}(E)X^{-(1+E)/2} \exp(X^2/2)^*$

---

\*The analysis between (5-119) and (5-120) of [3] is possibly misleading on this point.

where  $C_e(E)$  and  $C_o(E)$  are continuous functions of  $E$ . On the other hand, if  $E = 2p + 1$  with  $p$  a non-negative integer, the even (odd) series terminates if  $p$  is even (odd) — implying that  $C_e(E) = 0$  if and only if  $(E - 1)/2$  is a non-negative even integer, and  $C_o(E) = 0$  if and only if  $(E - 1)/2$  is a non-negative odd integer. It follows that there are bound states with energy eigenvalues  $E = 1, 3, 5, \dots$  and corresponding eigenfunctions proportional to  $\exp(-X^2/2)$  times the ‘Hermite’ polynomials  $1, X, 1 - 2X^2, X - \frac{2}{3}X^3, \dots$

In this part of the project, restrict attention to odd solutions satisfying

$$Y(0) = 0 \text{ and (WLOG) } Y'(0) = 1. \quad (6)$$

**Programming Task:** Write a program to solve Schrödinger’s equation (2) with the potential (3) by numerical integration, starting from the initial conditions (6). The program should find and (optionally) plot the solution  $Y(X)$  for a given value of  $E$  and a range of integration  $X \in [0, X_{\max}]$ , 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 outline 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  $X_{\max} = 5.0$  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  $[X_{\max}, \infty)$ ?

---

The asymptote can be established by substituting

$$Y(X) = e^{S(X)} \Rightarrow (S')^2 + S'' = X^2 - E$$

and noting that this is satisfied [formally] by the infinite series

$$S' \sim a_0 X + \sum_{n=1}^{\infty} a_n X^{-2n+1} \Rightarrow S \sim \frac{1}{2}a_0 X^2 + a_1 \ln X [+a \text{ constant, } = 0 \text{ WLOG}] - \frac{1}{2}a_2 X^{-2} + \dots$$

with

$$a_0 = -1, \quad a_1 = \frac{1}{2}(E - 1), \quad a_2 = \frac{1}{8}(E - 1)(E - 3), \quad \dots$$

or

$$a_0 = 1, \quad a_1 = -\frac{1}{2}(E + 1), \quad a_2 = -\frac{1}{8}(E + 1)(E + 3), \quad \dots,$$

implying that equation (2)–(3) has a solution  $Y_-(X)$  with exponential decay as  $X \rightarrow \infty$ ,

$$\begin{aligned} Y_- &= \exp \left[ -\frac{1}{2}X^2 + \frac{1}{2}(E - 1) \ln X - \frac{1}{16}(E - 1)(E - 3)X^{-2} + \dots \right] \\ &= \exp \left( -\frac{1}{2}X^2 \right) X^{(E-1)/2} \left[ 1 - \frac{1}{16}(E - 1)(E - 3)X^{-2} + \dots \right], \end{aligned}$$

and an independent solution  $Y_+(X)$  with exponential growth,

$$Y_+ = \exp \left( \frac{1}{2}X^2 \right) X^{-(E+1)/2} \left[ 1 + \frac{1}{16}(E + 1)(E + 3)X^{-2} + \dots \right],$$

the general linear combination being dominated for large  $X$  by the second (unless that is completely absent).

**Question 4** How will a *numerical* solution of (2)–(6) with  $E = 3$  behave as  $X \rightarrow \infty$ , 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

$$V(X) = -\frac{\Delta V}{(1 + X^2)^2} \quad (7)$$

where  $\Delta V$  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** Why, when looking for bound states for this  $V(X)$ , is there no loss of generality in restricting attention to solutions which are either even or odd in  $X$ ?

The potential (7) has at least one bound state for any  $\Delta V > 0$ , and more for larger  $\Delta V$ . Naively it might be expected to behave something like the ‘square’ potential well  $V_{\text{square}}(X)$  with the same dimensionless depth and comparable half-width<sup>†</sup>, i.e.

$$V_{\text{square}}(X) = \begin{cases} -\Delta V & \text{if } |X| < L \\ 0 & \text{if } |X| > L \end{cases} \quad \text{with (say) } L = \left( \frac{\int_0^\infty X^2 V(X) dX}{\int_0^\infty V(X) dX} \right)^{1/2} = 1, \quad (8)$$

which has exactly  $N$  bound states if  $(N - 1)\pi < 2L\sqrt{\Delta V} \leq N\pi$  (see for example [1], §9 or [2], §2.6 or [3], chapter 5 B).

Alternatively we might appeal to the *WKB approximation*, which is based on the [non-trivial!] observation (see for example [1], §34 or [2], chapter 8) that in the limit  $\Delta V \rightarrow \infty$  the bound-state energy eigenvalues asymptote to  $\tilde{E}_n \equiv -\Delta V / (1 + X_n^2)^2$  where  $X_n$  is determined from

$$\int_{-X_n}^{X_n} \sqrt{\tilde{E}_n - V(X)} dX \equiv \sqrt{\Delta V} \int_{-X_n}^{X_n} \left[ \frac{1}{(1 + X^2)^2} - \frac{1}{(1 + X_n^2)^2} \right]^{1/2} dX = \left(n + \frac{1}{2}\right) \pi \quad (9)$$

for  $n = 0, 1, 2, \dots, N - 1$  [cf. the ‘Bohr-Sommerfeld quantisation rule’ of the ‘old quantum theory’], the number of bound states  $N$  being determined (asymptotically) by

$$\left(N - \frac{1}{2}\right) \pi < \int_{-\infty}^{\infty} \sqrt{-V(X)} dX \equiv 2\tilde{L}\sqrt{\Delta V} \leq \left(N + \frac{1}{2}\right) \pi \quad \text{with} \quad \tilde{L} = \int_0^\infty \frac{dX}{1 + X^2} = \frac{1}{2}\pi. \quad (10)$$

**Question 6** Both these arguments *suggest* – correctly as it happens – that the potential (7) with  $\Delta V = 1$  has only one bound state. Verify that this is indeed the case, at least to the extent of trying  $E = -1$ ,  $E = 0$  and a few judiciously chosen values of  $E$  between  $-1$  and  $0$ , for both even and odd solutions. Present plots of these solutions over a suitable range  $[0, X_{\text{max}}]$ , explaining your choice(s) for  $X_{\text{max}}$  and the input parameters of the ODE solver.

---

<sup>†</sup>This cannot be defined unambiguously, e.g. we might use  $L = \int_0^\infty X V(X) dX / \int_0^\infty V(X) dX = 2/\pi$  instead of (8).

**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(7)? 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  $X \rightarrow \infty$ ?]

**Question 10** For the potential (7) with  $\Delta V = 49$ , 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.

## References

- [1] LI Schiff, *Quantum Mechanics*, 3rd edition, McGraw-Hill 1968.  
[https : //archive.org/details/QuantumMechanics\\_500](https://archive.org/details/QuantumMechanics_500)
- [2] DJ Griffiths, *Introduction to Quantum Mechanics*, Prentice Hall 1995.  
[https : //archive.org/details/IntroductionToQuantumMechanics\\_718](https://archive.org/details/IntroductionToQuantumMechanics_718)
- [3] S Gasiorowicz, *Quantum Physics*, Wiley 2003.



## 2.3 Non-Euclidean Geometry

*This project uses material from the Part IA Groups and Part IB Geometry courses.*

### 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

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

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  $\alpha, \beta, \gamma$  (the side of length  $a$  being opposite the vertex with angle  $\alpha$ ) then

$$\cos \alpha + \cos \beta \cos \gamma = \sin \beta \sin \gamma \cos a. \quad (2)$$

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

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

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  $\alpha, \beta, \gamma$  (the side of length  $a$  being opposite the vertex with angle  $\alpha$ ) then

$$\cos \alpha + \cos \beta \cos \gamma = \sin \beta \sin \gamma \cosh a. \quad (4)$$

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  $\pi/n$ . [Hint: Use the formulae (2) and (4) to find the vertices.]

## 5 Symmetry groups and tessellations

Let  $p$  be an odd prime. Let  $\mathrm{SL}(2, p)$  be the finite group of all  $2 \times 2$  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

$$\sigma_1 = \pm \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \sigma_2 = \pm \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}, \quad \sigma_3 = \pm \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \quad (5)$$

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  $\Delta_0$  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  $\Delta_0$  opposite the vertices with internal angles  $\pi/2, \pi/3$  and  $\pi/p$  (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  $\Delta$  be a triangle. The pair  $(g, \Delta)$  is *admissible* if it is obtained from  $(\pm I, \Delta_0)$  by a sequence of operations replacing  $(g, \Delta)$  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  $\sigma_1, \sigma_2$  and  $\sigma_3$  until no new elements are found.

**Question 5** Let  $p = 3$ . For each  $g \in \mathrm{PSL}(2, p)$  find a triangle  $\Delta$  such that  $(g, \Delta)$  is admissible. Use your programs to plot these triangles  $\Delta$ . 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  $\mathbb{P}(X \leq x) = F(x)$ . 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

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

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  $X_1, \dots, X_n$ , (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  $F(x) = 1 - e^{-\theta x}$ ,  $x \geq 0$ , corresponding to an exponential density with rate  $\theta$ , mean  $\theta^{-1}$ , and probability density function

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

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

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

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

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

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

Derive analytically  $\hat{m}_n$ , 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  $f(x | \theta) = \theta^2 x e^{-\theta x}$ ,  $x \geq 0$ , and integrate it to find  $F(x)$ . Can you compute  $F^{-1}$  in closed form?

**Question 6** The *log-likelihood* function is now

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

Calculate the maximum likelihood estimator for  $\theta$ .

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

**Question 8** We investigate the distribution of  $\hat{\theta}_n$  as follows. Take  $\theta_0 = 2.2$  and  $N = 200$ . Take  $\mathbf{x}(1), \dots, \mathbf{x}(N)$  as  $N$  independent random samples each of size  $n = 10$  from  $f(x | \theta_0)$ . Let  $\hat{\theta}_n(1), \dots, \hat{\theta}_n(N)$  be the corresponding maximum likelihood estimators of  $\theta$ . Generate the histogram\* of  $\hat{\theta}_n(1), \dots, \hat{\theta}_n(N)$ . 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,  $X_1, \dots, X_n$ , a random sample from  $N(\mu, \sigma^2)$ , the normal distribution with mean  $\mu$  and variance  $\sigma^2$ . An *unsubtle* method would be to use  $X = F^{-1}(U)$ , where  $U \sim \text{Unif}[0, 1]$ , and  $F(x) = \int_{-\infty}^x e^{-\frac{1}{2}v^2} dv / \sqrt{2\pi}$ . 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  $(\Phi, V)$  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 `hist` in MATLAB to draw one.

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

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

**Question 9** Show that if  $f(\phi, v) = \frac{1}{4\pi} e^{-v/2}$ ,  $0 \leq \phi \leq 2\pi$ ,  $v \geq 0$ , 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  $\text{Unif}[0, 1]$  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  $(x_1, \dots, x_n)$  of size  $n$  from distribution  $N(\mu, 1)$ . Explain how to construct an 80% confidence interval for  $\mu$ .

**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  $\mu$ . 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  $\mu$ ?

**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  $\mu$ ?

## 4 The $\chi^2$ Distribution

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

- chi-square with 1 degree of freedom ( $\chi_1^2$ );
- chi-square with 5 degrees of freedom ( $\chi_5^2$ );
- chi-square with 40 degrees of freedom ( $\chi_{40}^2$ ).

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?