Mathematical Tripos Part IB

Computational Projects

2019/2020

CATAM
Mathematical Tripos Part IB
Computational Projects

July 2019

Edited by the Computational Projects Assessors Committee

Course Director: 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 Random Binary Expansions
1.2 Ordinary Differential Equations

Additional Projects

2.1 The Restricted Three-Body Problem
2.2 Collapse of a Spherical Cavitation Bubble
2.3 Curves in the Complex Plane
2.4 Sensitivity of Optimisation Algorithms to Initialisation

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 2019-20. At the time of writing (July 2019) the booklet for the academic year 2018-19 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 manuals

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


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: see¹

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

¹ 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


A printable version is available from


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


There is a plethora of books on MATLAB. For instance:


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


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, especially if you have not programmed before, but you are free to write your programs in any computing language whatsoever. Python, R, C, C++, Mathematica, Maple and Haskell have been used by several students in the past, and Excel

---

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

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

4 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/.

5 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

6 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.\footnote{There is no need to consult the CATAM Helpline as to your choice of language.}

However, you should bear in mind the following points.

- The Faculty does \textit{not} promise to help you with programming problems if you use a language other than \textsc{Matlab}.

- Not all languages have the breadth of mathematical routines that come with the \textsc{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 \textsc{Matlab} (this can involve a fair amount of effort). To this end you may find reference books, such as \textit{Numerical Recipes} by W. H. Press \textit{et al.} (CUP), useful. You may use equivalent routines to those in \textsc{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 \textsc{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 \textit{CATAM Helpline} (see §4 below) can give clarification in specific cases.

- Subject to the aforementioned limited exceptions, you must \textit{write your own computer programs}. Downloading computer code, e.g. from the internet, that you are asked to write yourself counts as plagiarism (see §5).

2 Project Reports

2.1 Project write-ups: examination credit

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

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

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

Unless the project specifies a way in which an algorithm should be implemented, marks are, in general, not awarded for programming style, good or bad. Conversely, if your output is poorly presented — for example, if your graphs are too small to be readable or are not annotated — then you may lose marks.
No marks are given for the submission of program code without a report, or vice versa. 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. αs or βs) are awarded. The maximum contribution to the final merit mark is thus 160 and roughly the same (averaging over the α weightings) as for a 16-lecture course.

2.2 Project write-ups: advice

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

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

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

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

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

---

8 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. 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. Presenting mathematical results in a clear and concise way is an important skill and one that you will be evaluated upon in CATAM. Twenty sides of graphs would be excessive for most projects, even if the graphs were printed one to a page.\(^9\) 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. \LaTeX, Microsoft Word, LibreOffice) will generate output in PDF form. In addition, there are utility programs to convert output from one form to another, in particular to PDF form (e.g. there are programs that will convert plain text to PDF). Before you make your choice of word processor, you should confirm that you will be able to generate submittable output in PDF form. Please note that a PDF file including pages generated by scanning a hand-written report or other text document is **not** acceptable.

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

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

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

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

- **LibreOffice** can be installed for free for, *inter alia*, the Windows, MacOS and Linux operating systems from
  http://www.libreoffice.org/download/download/.

---

\(^9\) Recall that graphs should not as a rule be printed one to a page.
Both Microsoft Word and LibreOffice are available on the MCS.

\( \LaTeX \). If you decide to use a \( \LaTeX \), or you wish to try it out, \( \LaTeX \) is available on the MCS. If you are going to use it extensively, then you will probably want to install \( \LaTeX \) on your own personal computer. This can be done for free. For recommendations of \TeX distributions and associated packages see

- [http://www.tug.org/begin.html#install](http://www.tug.org/begin.html#install)
- [http://www.tug.org/interest.html#free](http://www.tug.org/interest.html#free)

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

- **\TeXstudio.** For Windows, Mac and Linux users, there is \TeXstudio. The pro\TeXt distribution, based on MiK\TeX, includes the \TeXstudio front end.
- **\TeXworks.** Again for Windows, Mac and Linux users, there is \TeXworks. The MiK\TeX distribution includes \TeXworks.
- **\TeXShop.** Many Mac aficionados use \TeXShop. To obtain \TeXShop and the \TeXLive distribution see [http://pages.uoregon.edu/koch/texshop/obtaining.html](http://pages.uoregon.edu/koch/texshop/obtaining.html).
- **\TeXnicCenter.** \TeXnicCenter is another [older] front end for Windows users.
- **LyX.** LyX is not strictly a front end, but has been recommended by some previous students. LyX is available from [http://www.lyx.org/](http://www.lyx.org/).

However, note that LyX uses its own internal file format, which it converts to \( \LaTeX \) as necessary.


- The \( \LaTeX \) source file (which may be helpful as a template), and supporting files, are available for download as a zip file from [http://www.maths.cam.ac.uk/undergrad/catam/files/Guide.zip](http://www.maths.cam.ac.uk/undergrad/catam/files/Guide.zip).

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

**Other sources of help.** A welter of useful links have been collated by the Engineering Department on their Text Processing using \( \LaTeX \) page; see [http://www.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html](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 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.

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

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.11 Malfunctions of your own equipment or the MCS are not an excuse for late submissions: leave yourself enough time before the deadline.

10 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).

11 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 is to use an online ‘cloud’ service; many of these services offer some free space. WIKIPEDIA has a fairly comprehensive list at [http://en.wikipedia.org/wiki/Comparison_of_online_backup_services](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](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/](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 2019 you can check that you are on the appropriate Faculty year list by referring to the [https://lists.cam.ac.uk/mailman/raven](https://lists.cam.ac.uk/mailman/raven) webpage (to view this page you will need to authenticate using Raven if you have not already done so). You should check that the *Maths-IB* mailing list is one of your current lists.

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

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

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

- please use an email address ending in cam.ac.uk (rather than a Gmail, etc. address) both so that we may identify you and also so that your email is not identified as spam;
• please specify, in the subject line of your email, ‘Part IB’ as well as the project number and title or other topic, such as ‘MATLAB query’, to which your email relates;

• please also restrict each email to one question or comment (use multiple emails if you have more than one question or comment).

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

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.

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 (also known as collusion).* If a general discussion *either* is happening regularly *or* gets to the point where physical or virtual notes are being exchanged (even on the back of an envelope, napkin or stamp), then it has reached the stage of unacceptable collaboration. 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. If you are uncertain about what constitutes an *unacceptable collaboration* you should seek advice from the CATAM Helpline.

*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 *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 you are in breach of these guidelines and may be charged with assisting another candidate to make use of unfair means.

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

*University-wide Statement on Plagiarism.* You should familiarise yourself with the University’s *Statement on Plagiarism.* There is a link to this statement from the University’s *Good academic practice and plagiarism* website

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

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

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

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

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

Declarations. To certify that you have read and understood these guidelines, you will be asked to sign an electronic declaration 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 A. You must list on the form anybody (students, supervisors and Directors of Studies alike) with whom you have exchanged information (e.g. by talking to them, or by electronic means) about the projects at any more than a trivial level: any discussions that affected your approach to the projects to any extent must be listed. Failure to include on your submission form any discussion you may have had is a breach of these guidelines.

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

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

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

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

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

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

Your electronically submitted write-ups will be kept in anonymised form to check against write-ups submitted in subsequent years.
Sanctions. If plagiarism, collusion or any other method of unfair means is suspected in the Computational Projects, normally the Chair of Examiners will convene an Investigative Meeting (see §5.2). If the Chair of Examiners deems that unfair means were used, the case may be brought to the University courts. According to the Statues and Ordinances of the University 12 suspected cases of the use of unfair means (of which plagiarism is one form) will be investigated and may be brought to one of the University courts or disciplinary panels. The University courts and disciplinary panels have wide powers to discipline those found to have used unfair means in an examination, including depriving such persons of membership of the University, and deprivation of a degree.

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

5.2 Oral examinations

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

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

Investigative Meetings. When plagiarism, collusion or other unfair means are suspected the Chair of Examiners may summon a candidate to an Investigative Meeting 13. 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).

---

12 From https://www.admin.cam.ac.uk/univ/so/.
13 For more information see https://www.plagiarism.admin.cam.ac.uk/files/investigative_2016.pdf.
One possible outcome is that the case is brought to the University courts where serious penalties can be imposed (see Sanctions above).

Timing. Viva Voce Examinations, Examination Interviews and Investigative Meetings are a formal part of the Tripos examination, and if you are summoned then you must attend. In the case of the core projects these will usually take place during Lent Full Term (although they may take place exceptionally during Easter Full Term), and in the case of the additional projects these will usually take place during the last week of Easter Full Term. Easter Term Viva Voce Examinations are likely to take place on the Monday of the last week (i.e. Monday 8th June 2020), 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 A). Further details, including the definitive submission form, will be made available when the arrangements for electronic submission of reports and programs (see below) are announced.

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.

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 21st January 2020, 10am–4pm,**

while for the additional projects the submission date is

**Tuesday 28th April 2020, 10am–4pm.**

No submissions will be accepted before these times, and **4pm on 21st January 2020 and 28th April 2020 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.¹⁴

- 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.

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

If you cannot remember your UIS password you will need to follow that instructions provided by the University Information Service.¹⁵ 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.

---

¹⁴ Alternatively, the University’s procedure can be invoked via the Examination Access and Mitigation Committee; see the Guidance Notes for Dissertation and Coursework extensions.

¹⁵ See [https://password.csx.cam.ac.uk/forgotten-passwd](https://password.csx.cam.ac.uk/forgotten-passwd).
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.
A Appendix: Example Submission Form

PART IB MATHEMATICAL TRIPOS 2019-20
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 and, unless such possession is specifically authorized, the possession of any book, paper or other material relevant to the examination. No member of the University shall assist a candidate to make use of such unfair means.

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

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

---

16 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:

<table>
<thead>
<tr>
<th>Project Number</th>
<th>Brief Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[
x_N = f(x_{N-1}),
\]

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

\[
x_N = x_{N-1} - \frac{F(x_{N-1})}{F'(x_{N-1})}.
\]

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

Order of Convergence

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

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

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

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

---

\(^1\) A more inclusive definition of order of convergence, referred to as the Q-order of convergence, might be

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

\(^2\) This quantity should be evaluated on the assumption that numbers are represented to infinite precision, without rounding error.
(i) *Binary search* is first-order convergent.

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

**Examples**

The cases to be studied as examples are

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

and

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

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

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

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

**Binary Search**

**Programming Task:** write a program to solve equation (4) by binary search. Provide for termination of the iteration as soon as the truncation error is guaranteed to be less than $0.5 \times 10^{-5}$, 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.

$$f(x) = x - h(F(x)),$$  \hspace{1cm} \tag{6}

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

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

$^3$ Or *functional*. 
**Question 3** Use the program to solve (4) by fixed-point iteration by taking
\[ h(F) = \frac{F}{2 + k} \]  
(7a)
in (6), so that
\[ f(x) = \frac{3\sin x + kx - 5}{2 + k}, \]  
(7b)
for some constant \( k \).

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

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

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

(iv) Also run the case \( k = 16 \). This should converge only slowly, so set \( N_{\text{max}} = 50 \). 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
\[ h(F) = \frac{1}{20} F, \]  
(8a)
in (6), so that
\[ f(x) = \frac{1}{20}(-x^3 + 8.5x^2 + 8). \]  
(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 \( N_{\text{max}} = 1000 \);

(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 \( N \to \infty \).

**Newton-Raphson Iteration**

A refinement of (6) is to let \( h \) depend on the derivatives of \( F \), i.e.
\[ f(x) = x - h(F, F', F'', \ldots). \]  
(9a)
In Newton-Raphson iteration
\[ h = \frac{F}{F'}. \]  
(9b)

\(^4\) The references at the end may prove helpful.
**Programming Task:** modify your program to recalculate the root of equation (4), and the double root of equation (5a), using Newton-Raphson iteration.

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

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

*Hint:* you may want to use a smaller value for $\epsilon$.

**References**


1.1 Random Binary Expansions

This project requires an understanding of the Part IA Probability and Part IA Analysis courses.

Let $U = (U_1, U_2, \ldots)$ represent an infinite sequence of coin tosses, with $U_i = 1$ if the $i$th toss is heads and $U_i = 0$ if it is tails. Suppose the coin tosses are independent, and that the probability of heads is $p \in (0, 1)$ and the probability of tails is $q = 1 - p$.

Given such a sequence we can define a real-valued random variable $X = f(U)$, taking values in the interval $[0, 1]$, by

$$f(U) = \sum_{i=1}^{\infty} \frac{U_i}{2^i}.$$  

We may think of $U$ as a binary expansion of $X$ (though in fact some $x \in [0, 1]$ do not have a unique binary expansion, to wit, 0.1 = 0.011111\ldots).

Define the cumulative distribution function $F(x) = \mathbb{P}(X \leq x)$.

For most values of $p$ the function $F$ is pathological, but it does have some interesting properties.

Approximating $F$

One way to approximate $F$ is by Monte Carlo simulation, as follows. Fix $n \in \mathbb{N}$. Generate a finite sequence $U^n = (U_1, \ldots, U_n)$, and compute $X^n = \sum_{i \leq n} U_i / 2^i$. Repeat this $N$ times to generate a random sample $X^n_1, \ldots, X^n_N$. Now we can plot the empirical cumulative distribution function

$$\hat{F}(x) = \frac{1}{N} \sum_{j=1}^{N} 1[X^n_j \leq x]$$

where $1[A]$ is the indicator function for the event $A$. This should approximate the actual cumulative distribution function $F(x)$.

**Question 1** Write a program to generate such a random sample. Plot the empirical distribution function for $p = 2/3$, using $n = 30$ and $N$ suitably large.

Calculating $F$

It turns out that for some values of $x$, we can calculate $F(x)$ explicitly.

**Question 2** Suppose that

$$x = \sum_{i=1}^{n} \frac{x_i}{2^i}$$

for some $n \in \mathbb{N}$ and some sequence $x_1, \ldots, x_n$. (When this is so, we say $x$ has a finite binary expansion.) Find a formula for $F(x)$.

**Question 3** Use your formula to plot a graph of $F$, for $p = 3/4$ and $n = 11$, sampling $F(x)$ at $x = 0, 1/2^n, 2/2^n, 3/2^n, 4/2^n, \ldots, 1$. Comment briefly on how this graph compares to the graph you obtained in Question 1. Include also a short comparison of the complexity (number of time steps needed) of both algorithms for general $n$ and $N$.  

July 2019/Part IB/1.1 Page 1 of 3 © University of Cambridge
Properties of $F$

The plots you have produced should make you wonder: is $F$ continuous? Is it differentiable?

**Question 4** Let $c$ have a finite binary expansion. Prove that $F(x)$ is continuous at $x = c$. Do your plots suggest that $F$ is continuous elsewhere? Prove or disprove.

We say that $F$ is left-differentiable at $c$ if the limit

$$\lim_{\delta \downarrow 0} \frac{F(c + \delta) - F(c)}{\delta}$$

exists and is finite, and that it is right-differentiable if the limit

$$\lim_{\delta \uparrow 0} \frac{F(c + \delta) - F(c)}{\delta}$$

exists and is finite. If $F$ is both left-differentiable and right-differentiable at $c$ and the two limits are equal, we say $F$ is differentiable at $c$.

**Question 5** Let $p = 3/4$ and $c = 9/16$. Plot $(F(c + \delta) - F(c))/\delta$ against $\delta$ for a suitable range of values of $\delta$ for which $c + \delta$ has a finite binary expansion. Does your plot suggest that $F$ is left-differentiable or right-differentiable at $c$?

**Question 6** Make a conjecture about whether $F$ is left-differentiable and/or right-differentiable at an arbitrary point $c$ with a finite binary expansion, for arbitrary $p \in (0, 1)$. Generate two or three plots which support your conjecture. Prove your conjecture.
**Project 1.1: Random Binary Expansions**

**Marking Scheme and additional comments for the Project Report**

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

<table>
<thead>
<tr>
<th>Question no.</th>
<th>marks available</th>
<th>marks awarded</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming task</strong> <em>Program:</em> for instructions regarding printouts and what needs to be in the write-up, refer to the introduction to the manual.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Question 1</strong> <em>Comments:</em> Explain why you are satisfied that your choice of $N$ is reasonable [approx. 5 lines]</td>
<td>3</td>
<td>C2.5</td>
</tr>
<tr>
<td><strong>Question 2</strong> <em>Comments:</em> Do not include trivial steps in your answer [approx. 10 lines]</td>
<td></td>
<td>M1</td>
</tr>
<tr>
<td><strong>Question 3</strong> <em>Comments:</em> Plot the graph obtained from the exact values of $F$, and use a separate graph (or two) for the comparison to the empirical distribution. In your comments try to argue quantitatively if possible. [approx. 5 lines]. Argue briefly for the complexity [approx. 3 lines]</td>
<td></td>
<td>C1.5+ M1</td>
</tr>
<tr>
<td><strong>Question 4</strong> <em>Comments:</em> Do not include trivial steps in your answer [approx. 1 page]</td>
<td></td>
<td>M3</td>
</tr>
<tr>
<td><strong>Question 5</strong> <em>Comments:</em> Scale the axes appropriately, and include a brief justification of your choice. [approx. 2 lines]</td>
<td></td>
<td>C2</td>
</tr>
<tr>
<td><strong>Question 6</strong> <em>Comments:</em> Do not include trivial steps in your proof. [approx. 1 page]</td>
<td></td>
<td>C2+M5</td>
</tr>
<tr>
<td><strong>Excellence marks.</strong> These are awarded for, among other things, mathematical clarity and good, clear output (graphs and tables) — see the introduction to the Project Manual.</td>
<td></td>
<td>E2</td>
</tr>
</tbody>
</table>

| Total Raw Marks | 20 |
| Total Tripos Marks | 40 |

---

1 C#, M# and E#: *Computational*, *Mathematical* and *Excellence* marks respectively.
2 For use by the assessor
3 This figure is only meant to be indicative of the length of your answer, rather than the exact number of lines you are expected to write.
1.2 Ordinary Differential Equations

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

1 Background Theory

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

\[ \frac{dy}{dx} = f(x, y), \]  

where \( y \) and \( f \) are vectors of length \( m \), subject to an initial condition

\[ y = y_0 \text{ at } x = x_0 \]  

for some constants \( x_0 \) and \( y_0 \). The exact solution of (1a)–(1b) will be denoted by \( y_e(x) \).

In the first part of the project (§2) the performance of three different numerical methods will be examined. A first-order equation (\( m = 1 \)) has been chosen for which \( y_e(x) \) has a known analytic form for comparison. In the second part (§3) one of the methods is extended to solve a second-order problem.

The numerical methods to be investigated are as follows.

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

\[ Y_{n+1} = Y_n + hf(x_n, Y_n), \]  

where \( Y_n \) denotes the numerical solution at \( x_n = x_0 + nh \), that is, at the \( n \)th step with step length \( h \). The Euler method is called a single-step method since \( Y_{n+1} \) is determined solely by the value \( Y_n \) at the previous step.

**Definition.** The global error after the \( n \)th step is defined as

\[ E_n = Y_n - y_e(x_n). \]  

**Definition.** The local error of the first step is defined as \( e_1 = Y_1 - y_e(x_1) \). For subsequent steps the local error is defined as

\[ e_n = Y_n - \tilde{y}(x_n), \]  

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

A numerical method is said to be \( p^{\text{th}} \)-order accurate if \( e_n \) is \( O(h^{p+1}) \) as \( h \to 0 \). It can be shown that the Euler method is first-order accurate.

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

\[ Y_{n+1} = Y_n + h \left[ \frac{3}{2} f(x_n, Y_n) - \frac{1}{2} f(x_{n-1}, Y_{n-1}) \right]. \]  

This is a two-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, such as the Euler method.
(c) The fourth-order-accurate Runge–Kutta (RK4) method employs the scheme:

\[ Y_{n+1} = Y_n + \frac{1}{6} h [k_1 + 2k_2 + 2k_3 + k_4] , \]  

(5a)

where

\[ k_1 = f(x_n, Y_n) , \]  

(5b)

\[ k_2 = f(x_n + \frac{1}{2} h, Y_n + \frac{1}{2} h k_1) , \]  

(5c)

\[ k_3 = f(x_n + \frac{1}{2} h, Y_n + \frac{1}{2} h k_2) , \]  

(5d)

\[ k_4 = f(x_n + h, Y_n + h k_3) . \]  

(5e)

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

2 Stability and accuracy of the numerical methods

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

\[ f(x, y) = -8y + 6e^{-2x} \]  

(6a)

and initial condition

\[ y = 0 \quad \text{at} \quad x = 0 . \]  

(6b)

This has the exact solution

\[ y = y_e(x) \equiv e^{-2x} - e^{-8x} . \]  

(7)

**Programming Task:** Write programs to apply each of the methods (a), (b) and (c) to this problem.

2.1 Stability

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

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

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

**Question 2**

(i) Find the analytic solution of the AB2 difference equation

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

(8)
with \[ Y_0 = 0 \ , \ Y_1 = 6h \] (from the Euler method). \hfill (9)

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

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

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

2.2 Accuracy

This subsection considers the accuracy of the three methods.

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

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

3 Numerical solutions of second-order ODEs

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

\[
\mu(\xi) \frac{\partial^2 \eta}{\partial \tau^2} = T_0 \frac{\partial^2 \eta}{\partial \xi^2}, \hfill (10a)
\]

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

\[
\eta(0, \tau) = \eta(L, \tau) = 0 \quad \text{for all } \tau. \hfill (10b)
\]

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

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

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

The problem admits ‘normal-mode’ solutions of the form

\[
\eta(\xi, \tau) = Ly(x) \cos(\omega \tau + \theta), \hfill (12)
\]
where the angular frequency $\omega$ and phase $\theta$ are constants, and dimensionless variables are defined by

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

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

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

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

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

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

subject to

$$y(0) = y(1) = 0. \quad \hfill (15b)$$

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

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

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

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

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

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

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

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

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

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

**Question 6** Taking $\alpha = 2$, run your program with $p = 4$ and $h = 0.1/2^k$ for $k = 0, 1, 2, \ldots, 12$ in turn, tabulating the numerical solution $Y_n$ at $x_n = 1$ and the global error $Y_n - y_e(1)$ against $h \equiv 1/n$. Repeat with $p = 5$. Do the errors behave as expected?
Programming Task: Write a program to search for eigenvalues using the ‘false position’ method – a variant of the bisection method where if a root of \( \phi(p) = 0 \) has been located to the interval \((p_1, p_2)\) with \( \phi(p_1) \) and \( \phi(p_2) \) having opposite signs, an estimate \( p_s \) for the root is calculated using the linear interpolant

\[
\Phi(p) \equiv \phi(p_1) \left( \frac{p_2 - p}{p_2 - p_1} \right) + \phi(p_2) \left( \frac{p - p_1}{p_2 - p_1} \right).
\]

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

\[
p = p_s \equiv \frac{\phi(p_2) p_1 - \phi(p_1) p_2}{\phi(p_2) - \phi(p_1)}
\]

which is accepted if \( |\phi(p_s)| < \epsilon \) where \( \epsilon \) is a specified small value; if not, the process is iterated with \( p_1 \) or \( p_2 \) replaced by \( p_s \) such that \( \phi(p_1) \) and \( \phi(p_2) \) still have opposite signs.

For the current task, \( \phi(p) \) is the numerical (RK4) solution \( Y_n \) of (15a)–(16) at \( x_n = 1 \) obtained with a suitably small value of \( h \equiv 1/n \).

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

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

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

\[
\int_0^1 (1 + x)^{-10} \left[ p^{(k)} y^{(k)}(x) \right]^2 dx = 1.
\]

(If you wish, this integral may be approximated using a black-box integration routine such as the MATLAB function \text{trapez}.) Explain carefully why you are satisfied that the eigenvalues found are indeed the smallest, and that they have the required accuracy.

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

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

Additional Reference


*See for example Bender, C.M. & Orszag, S.A., *Advanced mathematical methods for scientists and engineers*, Chapter 10 (in particular §10.1, Example 5).
## Project 1.2: Ordinary Differential Equations

### Marking Scheme and additional comments for the Project Report

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

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

| Total Raw Marks | 20 |
| Total Tripos Marks | 40 |

---

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

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

<sup>3</sup> Your aim is to answer succinctly the questions including the graphs and tables, and to make all important points. The length specified here should be sufficient space for you to do this but is not a target.
2.1 The Restricted Three-Body Problem

This project is related to the IA Dynamics and Relativity lecture course, but is self-contained.

1 Introduction

Determining the motion of a number of gravitating bodies is a classical problem. It can be solved analytically for two bodies, but not for three or more. Various simplifications have historically been considered, one of which is the ‘restricted three-body problem’ in which the third body is taken to be much smaller in mass than the other two and therefore has negligible influence on their motion. The problem is then to solve for the motion of the third body in the known gravitational field of the first two bodies. This project investigates the ‘planar circular restricted three-body problem’, a particular case where the first two bodies move in (stable) circular orbits around their joint centre-of-mass (taken as origin) and the motion of the third body is confined to the plane of the circles (taken as the \( x-y \)-plane).

It is convenient to transform to a rotating frame of reference in which the first two bodies appear stationary. Scalings may be chosen so that the angular velocity of this frame is 1 and the distance between the two bodies is 1. The only parameter then appearing is the quantity \( \mu \in (0, 0.5] \) defined such that the two masses are in the ratio \( \mu : 1 - \mu \) and are situated respectively at the points \((\mu - 1, 0)\) and \((\mu, 0)\), which will be referred to as \( P_l \) and \( P_h \). The equation of motion for the third body, whose position at time \( t \) is \((x(t), y(t))\) may then be written as:

\[
\begin{align*}
\ddot{x} - 2\dot{y} &= -\frac{\partial \Omega}{\partial x}, \\
\dot{y} + 2\dot{x} &= -\frac{\partial \Omega}{\partial y},
\end{align*}
\]

where

\[
\Omega = -\frac{1}{2} (x^2 + y^2) - \frac{\mu}{\sqrt{(x + 1 - \mu)^2 + y^2}} - \frac{1 - \mu}{\sqrt{(x - \mu)^2 + y^2}},
\]

i.e. \( \Omega \) is the potential of the centrifugal and gravitational forces.

Despite the substantial restriction to the full three-body problem which this represents, it is not possible to solve the system (1a), (1b) and (2) analytically.

Question 1 Show from (1a) and (1b) that the quantity

\[
J = \frac{1}{2} (\dot{x}^2 + \dot{y}^2) + \Omega(x, y)
\]

is constant following the motion. Deduce that trajectories must be confined to the region

\[
\Omega(x, y) \leq \Omega(x_0, y_0) + \frac{1}{2} (u_0^2 + v_0^2),
\]

where \( x_0, y_0, u_0 \) and \( v_0 \) are the initial values of \( x, y, \dot{x} \) and \( \dot{y} \), respectively.

Programming Task Write a program to solve the system (1a), (1b) and (2) numerically, given suitable initial conditions on \( x, y, \dot{x} \) and \( \dot{y} \). You may use a black-box ODE solver such as the MATLAB function \texttt{ode45} which automatically adapts the time-step according to specified absolute and relative error tolerances (these may need to be adjusted). A fixed-step solver (as in the Ordinary Differential Equations core project) may require very small time-steps.
Whenever you write a computer program to find a numerical solution, it is necessary to check that the program is generating accurate results. Standard checks include (i) testing the program against known analytic solutions (if there are any), and (ii) varying the time-step or error tolerances. For this problem, the fact that $J$ is constant provides not only a useful constraint on the behaviour of solutions, but also another possible check on numerical accuracy.

## 2 Space travel

Assume that the third body is a spacecraft, with the first two bodies being co-orbiting planets of equal mass, i.e. $\mu = 0.5$ (the so-called ‘Copenhagen problem’).

### Question 2

Consider motion sufficiently close to $P_h$ that its gravitational attraction dominates both that of $P_l$ and the centrifugal force, and (2) may be approximated by

$$\Omega = -\frac{0.5}{\sqrt{(x-\mu)^2 + y^2}}.$$  \hfill (5)

Show that in polar coordinates with

$$x(t) - \mu = r(t) \cos \theta(t), \quad y(t) = r(t) \sin \theta(t) \quad (6)$$

the approximate system (1a)–(1b) with (5) is equivalent to

$$\dot{\theta} = -1 + kr^{-2}, \quad \ddot{r} = -V'(r) \quad (7)$$

where $k$ is an arbitrary constant and $V'(r)$ is to be found, and (for particular initial conditions) has analytic solutions with the spacecraft in a circular orbit of radius $a$ about $P_h$ where $a$ can take any value [though of course (5) is a good approximation to (2) only if $a$ is small]. How is the constant $k$ related to $a$?

Modify your program to solve (1a)–(1b) with $\Omega$ specified by (5) instead of (2). Demonstrate, for one value of $a$, that the modified program can accurately reproduce the analytic circular-orbit solution.

### Question 3

Return to the original system (1a), (1b) and (2) with $\mu = 0.5$, and take initial conditions $x = 0.32$, $y = 0$, $\dot{x} = 0$, $\dot{y} = v_0$ with $v_0 = -1.0, -1.5, -1.73, -1.78, -1.853, -1.858, -2.3$ and -2.31 in turn. For each case, use your program to integrate from $t = 0$ to $t = 30$ and

(i) display the trajectory in the $(x,y)$ plane, and the forbidden region $\Omega(x,y) > J$ (shaded, say, using the MATLAB function `contourf`), on the same plot,

(ii) state the values of $x$ and $y$ at $t = 30$, giving a reasoned assessment of their accuracy.

Comment on the trajectories, and how these and the allowed region change as $-v_0$ increases. Is the allowed region a useful guide to the size of the trajectory? What value of $v_0$ would be most suitable to travel from the neighbourhood of $P_h$ to the neighbourhood of $P_l$?

[It may be instructive to try other values of $v_0$ and/or integrate further in time.]
3 Lagrange points and asteroids

In this part of the project do not restrict attention to the case \( \mu = 0.5 \).

**Question 4** By examining contour plots of \( \Omega \) show that the system (1a)-(2) generally has five equilibrium points: three ‘collinear Lagrange points’ on the \( x \)-axis and two ‘equilateral Lagrange points’ at the third vertex of an equilateral triangle whose other two vertices are at \( P_l \) and \( P_h \). Display contour plots for three values of \( \mu \in (0, 0.5] \) with the equilibrium points marked. (You may wish to use a black-box root-finder such as the MATLAB function \texttt{fzero} to locate the collinear points accurately.)

Investigate numerically the linear stability of the collinear Lagrange points, i.e. stability to very small, formally \textit{infinitesimal} perturbations, by starting trajectories a small distance away from the equilibrium point and integrating forward in time. Display plots of a few representative trajectories in the \((x, y)\) plane, together with corresponding plots of \( x \) and \( y \) against \( t \), to illustrate your results. Explain why you are satisfied that these computations, which start with small but necessarily finite perturbations, have captured the behaviour for infinitesimal perturbations.

What do you conclude about the linear stability of the collinear Lagrange points? Does it depend on \( \mu \)? Confirm your numerical findings analytically by performing a linearised stability analysis about such points. (You may be able to deduce the necessary information about the second derivatives of \( \Omega \) by considering the shapes of the contours, rather than by detailed calculation.)

**Question 5** Continue with a numerical investigation of the linear stability of the equilateral Lagrange points for parameter values \( \mu = 0.01, 0.025, 0.05, 0.1 \) and 0.5. Illustrate the results in your write-up with at least one trajectory picture, and corresponding plots of \( x \) and \( y \) against \( t \), for each.

How do the stability properties change with \( \mu \)? By further numerical experimentation find, to within \( \pm 1\% \), the critical value \( \mu_c \) dividing values of \( \mu \) for which the point is linearly stable from those for which it is unstable, and present numerical results in support of your conclusion. Confirm it by performing a linearised stability analysis (which this time certainly does require calculation of the second derivatives of \( \Omega \)). For the stable cases, what does this analysis indicate about the form of the motion?

**Question 6** The (Jupiter) Trojans are asteroids observed near the Sun-Jupiter equilateral Lagrange points, for which \( \mu = 9.54 \times 10^{-4} \). Is the persistence of the Trojans near these points consistent with your findings above? The Earth-Moon system has \( \mu = 0.012141 \) but no analogue of the Trojans is observed: can you suggest why?

**Reference**

2.2 Collapse of a Spherical Cavitation Bubble

This project is self-contained, although knowledge of the Part IB Fluid Dynamics course is an advantage in interpreting the physical significance of the results.

1 Introduction

The aim of this project is to examine the influence of surface tension on the time of collapse of a spherical cavitation bubble, and to determine the corresponding changes in the large (and often damaging) fluid pressures nearby.

For computational purposes, it is convenient to use dimensionless variables: units are chosen so that the initial radius of the bubble, the fluid density and the pressure at large distance are all unity. (This can be achieved by scaling the usual \( r, t \) and \( \rho \) variables.)

The bubble is represented by an empty spherical cavity of radius \( R(t) \), where \( t \) is time, in an unbounded fluid which is initially at rest. It may be shown that the fluid pressure \( p(r, t) \) at time \( t \) is given by

\[
p(r, t) = 1 + \frac{1}{r} \frac{d}{dt} \left( R^2 \frac{dR}{dt} \right) - \frac{R^4}{2} \left( \frac{dR}{dt} \right)^2.
\]

(1)

An evolution equation for \( R(t) \) is obtained using the boundary condition

\[
p(R, t) = -2\lambda/R,
\]

(2)

where \( \lambda \) in these units is the (constant) surface tension at the surface \( (r = R) \) of the bubble. (In general, \( \lambda \) is surface tension scaled by pressure at infinity and initial bubble radius, and it measures the relative importance of surface tension and inertia in the system – an inverse Weber number.)

2 Analytic solutions for \( \frac{dR}{dt} \) and \( p(r, t) \)

Question 1 Show that the equation for \( R(t) \) admits a first integral

\[
\left( \frac{dR}{dt} \right)^2 = \frac{2}{3} \left( \frac{1}{R^3} - 1 \right) + 2\lambda \left( \frac{1}{R^3} - \frac{1}{R} \right).
\]

(3)

Plot on the same graph \( dR/dt \) against \( R \) for \( \lambda = 0.0, 0.1, 1.0, 10.0, \) and \( 100.0 \). (Note that you may find it helpful to use logarithmic scales for both \( dR/dt \) and \( R \).) Comment on your graph.

Question 2 Show that \( p(r, t) \) can be expressed in terms of \( r \) and \( R(t) \), that is it can be written \( p(r, t) \equiv p(r, R) \). Show that \( p(r, R) \) has a particularly simple form when expressed in terms of \( \alpha \) and \( \beta \), where

\[
\alpha = \frac{1}{4}(1 - 4R^3) + \frac{3}{4}\lambda(1 - 3R^2) \quad \text{and} \quad \beta = 1 - R^3 + 3\lambda(1 - R^2).
\]

(4)

Show that the largest pressure in the fluid at time \( t \) is

\[
p_{\text{max}} = \begin{cases} 1, & \alpha < 0 \\ 1 + R^{-3}(\alpha^4/\beta)^{1/3}, & \alpha > 0 \end{cases}
\]

(5)
Plot on the same graph \(p(r, R)\) against \(r\) for a range of values [about four will do] of \(R\) for the case \(\lambda = 0\). For each value of \(R\), what is \(p_{\text{max}}\) and what is the value of \(r\) which corresponds to \(p_{\text{max}}\)?

Plot similar graphs for the cases \(\lambda = 0.2\) and \(\lambda = 9.0\). You may also find it helpful to plot graphs of \(p(r, R)\) against \(r\) normalised so that \(p_{\text{max}} - p_{\text{min}} = 1\).

Comment on your graphs.

\section*{3 No surface tension case}

\textbf{Question 3} \hspace{1em} Take the case without surface tension \(\lambda = 0\) and write a program to solve equation (3) for \(R(t)\). You will find it worthwhile to observe the following points:

- Equation (3) is singular as \(R \to 0\); to remove this singularity, define a new dependent variable, \(x\), say, such that 
  \[x = R^{5/2}.\]

  Show that (3) then becomes
  \[
  \dot{x} = -\frac{5}{2} \left[\frac{2}{5}(1 - x^{6/5}) + 2\lambda(1 - x^{4/5})\right]^{1/2},
  \]
  with the condition \(x = 1\) at \(t = 0\). Justify mathematically and/or physically why the negative root has been chosen. Would it have made sense to choose the positive root?

- Equation (6) is non-analytic at \(x = 0\). Explain with reasons why as \(x \to 0\) higher-order numerical methods (such as Runge–Kutta) do not have their usual benefits over low order schemes such as the Euler method. For your numerical calculations use the Euler method.

- Equation (6) has the trivial solution \(x = 1\) for all \(t\). To avoid this, either use an alternative numerical scheme for the first step, or find a series solution for small time and start from a suitable non-zero value of time.

By making the substitution \(x = \sin^{5/3} \theta\) and performing a numerical integration, or otherwise, solve (6) for \(\lambda = 0\) to obtain a value for the time \(t_c\) for collapse.

Using your program, determine the time \(t_c\) for collapse. Plot \(R\) as a function of \(t\), and show also the behaviour of \(p_{\text{max}}(t)\) (choosing appropriate scales to show the variation in \(p_{\text{max}}(t)\)). Justify carefully the numerical accuracy of your results.

\section*{4 With surface tension case}

\textbf{Question 4} \hspace{1em} Repeat the computation for a representative set of [positive] values of \(\lambda\). Interpret your results both physically and mathematically. Comment on the limits \(\lambda \ll 1\) and \(\lambda \gg 1\).

\textbf{Question 5} \hspace{1em} What are likely to be the physical limitations of this model?

So far we have considered dimensionless variables. Now make a rough estimate of the actual initial radius of bubbles that might be generated by a boat propellor. Assuming that the bubbles are empty, use your graphs of \(dR/dt\) from Question 1 to estimate a radius below which the physical limitations of this model are likely to be important.
How might you change the model to allow for bubbles in a liquid that are not empty but contain small amounts of vapour or other gas?

_Historical Footnote._ Calculation of the collapse of a spherical cavitation bubble (in a slightly compressible fluid) was one of the earlier uses of computers in fluid dynamics (see C. Hunter’s Ph.D. thesis 1960 in the DAMTP library).
2.3 Curves in the Complex Plane

This project uses material found in both the Complex Methods and Complex Analysis courses.

1 Introduction

A curve $\gamma$ is a continuous map from the closed bounded interval $[0, 2\pi] \text{ or } [0, 1] \text{ or even } [a, b]$ to $\mathbb{C}$. We say that $\gamma$ is smooth if $\gamma(t) = (u(t), v(t))$ for $u, v$ functions that are infinitely differentiable and $\gamma$ is closed if $\gamma(0) = \gamma(2\pi)$.

Suppose that $w = f(z)$ is a complex polynomial in $z$ and we take $r > 0$. If $C_r$ is the circle of radius $r$ then it is the case that $f(C_r)$ is a smooth closed curve in the $w$-plane. Curves that are generated in this way have a number of interesting properties that will be investigated in this project.

2 Complex roots of $f(z)$

In this section you will find the complex roots of the polynomial

$$f_1(z) = z^3 + z^2 + (5 - 4i)z + 1 - 8i \quad (1)$$

and of its first derivative $f_1'(z)$.

**Programming Task:** Write a program that, given a polynomial $f(z)$, plots a graph of $f(C_r)$ and computes the coordinates and modulus of the closest point on $f(C_r)$ to $0 + 0i$. Your program should prompt you to enter a value for $r$.

**Question 1** Using your program find the three roots of $f_1(z)$ to three significant figures and record the roots in your write-up, together with the corresponding values of $r$. There is no need for your program to automate the search for an $r$ such that $\min ||f(C_r)|| = 0$ — trial and improvement is an adequate method. Nevertheless, you may find it helpful to include an option to carry out the search automatically. Taking your output for the roots, how can you justify that these answers are indeed correct to three significant figures?

**Question 2** Write down the first derivative $f_1'(z)$ of $f_1(z)$ and use your program to find the two roots of $f_1'(z)$ to three significant figures. Record the roots in your write-up, together with the corresponding values of $r$. Again, how is your answer justified?

3 Images $f(C_r)$ of $C_r$

In this section you will explore the geometry of the image $f(C_r)$ in the $w$-plane for some polynomials $f(z)$. Consider the polynomial $g(z) = z^3 + z$.

**Question 3** Change the program that you wrote for Question 1 so that it plots the image $g(C_r)$ for a given $r$. Examine what happens as $r$ increases from a very small value to a moderately large one. In your write-up explain what happens. Use plots of $g(C_r)$ for suitably chosen values of $r$ to illustrate your explanation.
Question 4  Repeat Question 3 for the polynomial \( h(z) = z^2 + 2z + 1 \). Again use plots of \( h(C_r) \), along with figures chosen to zoom in on relevant details of the image curves.

Question 5  In the light of what you have found in Questions 3 and 4, explain what happens to the image curve \( f_1(C_r) \) of the original polynomial \( f_1 \) in Question 1 as \( r \) increases from a suitably small value to a large one. Again, use plots of \( f_1(C_r) \) for suitably chosen values of \( r \) to illustrate your explanation, including those chosen to zoom in on particular details.

4 Curvature of images \( f(C_r) \) of \( C_r \)

A smooth curve \( x : [a, b] \to \mathbb{C} \) is said to be regular if \( x'(t) \neq 0 \) for all \( t \in [a, b] \). It is a fact that any smooth regular curve \( x \) admits a smooth reparametrisation \( x(s) \) (where we are using \( x \) for both the original and reparametrised curve) such that the parameter \( s \) is the distance travelled along the curve, which we regard as the “natural” parametrisation. Figure 1 shows coordinates and vectors associated with a curve such as \( f(C_r) \). Here \( x(s) \) is the position vector of a point on the curve. The distance from \( x(s) \) to \( x(s+ds) \) is \( dx = |x(s+ds) - x(s)| \) for an infinitesimal distance \( ds \) along the curve. Hence \( |\dot{x}(s)| = 1 \) where

\[
\dot{x}(s) = \lim_{ds \to 0} \frac{x(s+ds) - x(s)}{ds} \tag{2}
\]

(this is sometimes phrased as \( x(s) \) is a unit speed curve). Hence the tangent vector \( t(s) = \dot{x}(s) \) to the curve \( x(s) \) is always a unit vector. The vector \( k(s) = \ddot{x}(s) \) is the curvature vector on the curve at the point \( x(s) \). The curvature \( |\kappa| \) of the curve at the point \( x(s) \) is the magnitude of \( k(s) \),

\[
|\kappa| = |k(s)| \tag{3}
\]

and the radius of curvature \( \rho \) is

\[
\rho = \frac{1}{|\kappa|} = \frac{1}{|k(s)|} \tag{4}
\]

The function \( f(z) \) is not a natural representation of the curve \( f(C_r) \) because neither is \( z \) a scalar nor is \( z_2 - z_1 \) the distance in the complex plane along the curve between the two points \( f(z_1) \) and \( f(z_2) \). However, by using suitable coordinate transforms, an expression for the curvature vector can be found for an arbitrary parametric representation of \( f(z) \).

To do so, write \( f(z) \) in terms of the angle \( \phi = \arg z \) to give a representation \( x = x(\phi) \) of the curve \( f(C_r) \) in terms of \( \phi \):

\[
x(\phi) = \text{Re} \left[ f(z(\phi)) \right] = \text{Re} \left[ f(\phi) \right] \tag{5}
\]

\[
y(\phi) = \text{Im} \left[ f(z(\phi)) \right] = \text{Im} \left[ f(\phi) \right] \tag{6}
\]

where \( x(\phi) = (x(\phi), y(\phi)) \).
Question 6  Show that

\[ |\kappa| = \frac{|\mathbf{x}' \times \mathbf{x}''|}{|\mathbf{x}'|^3} , \]  

(7)

where

\[
\mathbf{x}' = \left( \text{Re} \left[ \frac{df(z(\phi))}{d\phi} \right], \text{Im} \left[ \frac{df(z(\phi))}{d\phi} \right] \right) \]  

(8)

\[
\mathbf{x}'' = \left( \text{Re} \left[ \frac{d^2f(z(\phi))}{d\phi^2} \right], \text{Im} \left[ \frac{d^2f(z(\phi))}{d\phi^2} \right] \right) . \]  

(9)

Equation (7) gives us the magnitude but not the sign of \( \kappa \). We (arbitrarily) define the sign of \( \kappa \) to be positive if the curve \( \mathbf{x}(s) \) is turning anticlockwise about its local centre of rotation. By expressing \( \mathbf{x}' \) and \( \mathbf{x}'' \) in terms of \( \mathbf{x}(s) \), \( \mathbf{t}(s) \), \( \mathbf{k}(s) \) and derivatives of \( s \) with respect to \( \phi \), find a way to compute the sign of \( \kappa \).

Write (8) and (9) in terms of derivatives of \( f(z) \) with respect to \( z \).

Programming Task: Write a program to compute the integral

\[ \kappa_{\text{tot}} = \int_{f(C_r)} \kappa \, ds \]  

(10)

for a range of values of \( r \). Use your answer to Question 6 to ensure that your program calculates the sign of \( \kappa \) correctly.

Question 7  Using your program plot a graph of \( \kappa_{\text{tot}} \) against \( r \) for each of the polynomials \( f_1(z) \), \( g(z) \) and \( h(z) \). Use what you have found out in Questions 1–6 to help you to explain what you see in your graphs. For a general polynomial \( f(z) \), what does \( \kappa_{\text{tot}} \) tell you about the curve \( f(C_r) \)?

Reference

2.4 Sensitivity of Optimisation Algorithms to Initialisation

This project requires an understanding of the Part IA Probability and Part IA Analysis courses.

1 Introduction

In modern statistics and machine learning, it is common to derive estimators and make decisions by minimising some objective function \( f \). This task can generally not be solved in closed form. As such, a standard solution is to apply an iterative optimisation algorithm to attempt to find an approximate minimiser.

However, increasingly, statistical problems lead to complicated objective functions which admit multiple local minima. It is common practice to then run the optimisation algorithm numerous times from different initial conditions, in hope of finding the true optimum or a satisfactory one eventually. It is thus of interest to understand the sensitivity of our optimisation algorithms to their initialisation, and to understand which features of the objective function inform the outcomes of these algorithms.

2 Gradient Descent

The optimisation algorithm of study in this project is gradient descent. To run it, one must specify a differentiable objective function \( f : D \rightarrow \mathbb{R} \) with domain \( D \subseteq \mathbb{R}^d, d \geq 1 \), an initial point \( x_0 \in D \), and a step-size \( h > 0 \). The iterates of the algorithm are then defined recursively as

\[
x_t = x_{t-1} - h \nabla f(x_{t-1}) \quad \text{for } t \in \{1, 2, \ldots\},
\]

(1)

and the hope is that as \( t \rightarrow \infty \), the iterates \( x_t \) converge (numerically) to a minimiser of \( f \) if \( h \) is sufficiently small.

In what follows, we will focus our attention on minimisation of the ‘double-well’ toy function \( f_\theta \), defined for \( \theta \in (0, \pi) \) by

\[
f_\theta : [-1, 1] \rightarrow \mathbb{R}, \quad x \mapsto \left(x^2 - \frac{3}{4}\right)^2 - x \cos \theta.
\]

(2)

**Question 1:** Find the stationary points of this function in analytic form, and classify them as local minima, maxima, or saddlepoints. [Hint: the stationary points can all be expressed as trigonometric functions; in particular, in your calculations you may use an expression for the cosine of the triple angle.]

Given that we know the analytic form of the stationary points of \( f_\theta \), we can easily evaluate the performance of gradient descent (or indeed, any other optimisation algorithm).

**Question 2:** Take \( \theta = \frac{\pi}{6}, h = 0.01 \), and run gradient descent on \( f = f_\theta \) for 1000 steps, from initial points \( x_0 \in \{ k^{1/50} | k = -50 \} \). What do you observe about the outcomes?
3 The Monte Carlo Method

In computational settings, it is often the case that a quantity of interest \( \nu \) is naturally expressed as an expectation, i.e. there is some random variable \( X \) and some function \( g \) such that

\[
\nu = \mathbb{E}[g(X)].
\]  

(3)

The Monte Carlo method (MC) involves drawing \( N \) independent samples \( \{X^i\}_{i=1}^N \) which are distributed like \( X \), and forming the estimator

\[
\hat{\nu}_N \triangleq \frac{1}{N} \sum_{i=1}^N g(X^i).
\]  

(4)

**Question 3:** Show that \( \hat{\nu}_N \) is unbiased for \( \nu \), i.e. \( \mathbb{E}[\hat{\nu}_N] = \nu \). Assuming that \( \text{Var}(g(X)) < \infty \), obtain an expression for the variance of \( \hat{\nu}_N \).

Returning to our toy function above, fix \( \theta \in (0, \pi) \), \( f = f_\theta \), and let \( \{X^h_t\}_{t=0,1,...} \) be the sequence of random variables obtained by i) sampling an initial point \( X^h_0 \sim \text{Uniform}([-1,1]) \), and ii) iterating \( X^h_t = X^h_{t-1} - h \nabla f(X^h_{t-1}) \).

For some \( T, h > 0 \) such that \( Th^{-1} \in \mathbb{N} \), we are interested in studying the behaviour of

\[
\mu^h \triangleq \mathbb{E}[X^h_{Th^{-1}}] \quad \text{and} \quad \mu \triangleq \lim_{h \to 0^+} \mu^h,
\]  

i.e. the outcome of running gradient descent from a randomised initial point, using smaller and smaller step-sizes, and run for longer and longer. We take \( T = 10 \) fixed throughout, as in this example, this is approximately sufficient for convergence to take place.

A basic approach to estimating \( \mu \) is to take \( h \) as small as possible, and to estimate \( \mu^h \) as accurately as possible by the Monte Carlo estimate \( \hat{\mu}_N^h \) by taking \( N \) as large as possible.

**Question 4:** Test this method out: fix \( \theta = \frac{\pi}{4} \), and for \( k \in \{0, 1, \cdots, 10\} \), take \( h = 0.1 \cdot 2^{-k} \), and estimate \( \mu^h \) using \( N_k = 2^{20-k} \) samples, so that the same amount of computational time is used for each \( k \). What do your estimates suggest about the behaviour of \( \mu^h \) as \( h \) decreases? What do they suggest about the variance of \( X^h_{Th^{-1}} \) as \( h \) decreases?

In this approach, because \( h \) is not exactly 0, we incur a finite bias, i.e. even in the limit of infinitely many samples, our estimator would converge to \( \mu^h \neq \mu \). As such, the variance of our estimator would not fully reflect its accuracy. Instead, it is standard to use the following more general measure of accuracy. The *mean squared error* (MSE) of an estimator \( T \) of a quantity \( \tau \) is defined by

\[
\text{MSE}(T; \tau) \triangleq \mathbb{E}[(T - \tau)^2].
\]  

(6)

**Question 5:** Prove the ‘bias-variance decomposition’, i.e. show that

\[
\text{MSE}(T; \tau) = (\mathbb{E}[T] - \tau)^2 + \text{Var}(T).
\]  

(7)

We present the following facts without proof: for \( h \) sufficiently small, there are constants \( A_1, A_2, A_3 \in (0, \infty) \) such that

1. the bias of the approximation \( \mu^h \) is bounded as \( |\mu^h - \mu| \leq A_1 h \),
2. the variance of \( X^h_{Th^{-1}} \) is bounded as \( \text{Var}(X^h_{Th^{-1}}) \leq A_2 \), and
3. for \( t \in \{0,1,\ldots\} \), the cost of generating a sample of \( X^h_t \) satisfies \( \text{Cost}(X^h_t) = A_3 t \).
Question 6: Suppose we estimate $\mu$ by fixing $h > 0$, $N \in \mathbb{N}$, generating $N$ i.i.d. samples $\{Y^i\}_{i=1}^N$ distributed as $X_{T_{h^{-1}}}^h$, and forming the estimator

$$\hat{\mu}_N = \frac{1}{N} \sum_{i=1}^N Y^i. \quad (8)$$

Use the bias-variance decomposition to show that the MSE of $\hat{\mu}_N$ can be bounded above by $A_1^2 h^2 + \frac{\delta^2}{N}$. Suppose now that the computational budget is $C$, i.e. the cost of generating all of the random variables used in the MC estimator is bounded above by $C$. Assume that we use our full budget, i.e. we choose $(N, h)$ such that $N \cdot \frac{A_1^2}{h^2} = C$. Use this to express the upper bound on the MSE as a function of only $h$, and derive the $h$ which minimises this upper bound. How does the optimal MSE scale with $C$?

4 Multi-Level Monte Carlo

For a given computational budget $C$, it is possible construct estimators with less variability than $\hat{\mu}_N$, and hence improve our accuracy. We exploit the intuition that if the initial point $x_0$ is fixed, we expect that the paths of $X_t^h$ and $X_{2t}^{h/2}$ will stay close together, and thus that $\mu^h \approx \mu^{h/2}$.

In order to justify this later on, we introduce an extra fact without proof: for $h$ sufficiently small, there is a constant $A_4 \in (0, \infty)$ such that

4. if two sequences of gradient descent iterates have the same initial point $X_0 \sim \text{Uniform}([-1, 1])$, then $\text{Var} \left( X_{T_{h^{-1}}}^h - X_{2T_{h^{-1}}}^{h/2} \right) \leq A_4 h^2$.

We note quickly that the facts presented before Question 6 remain true in what follows.

For $X_0 \sim \text{Uniform}([-1, 1])$ and $l = 0, \ldots, L$, $L \in \mathbb{N}$, define $h_l = 0.1 \times 2^{-l}$, let $X_{T_{h_l^{-1}}}^h$ be the $(T_{h_l^{-1}})^{th}$ gradient descent iteration for $f_\theta$, with $X_0^h = X_0$, and with step-size $h_l$. Define the random variables

$$Y_0 = X_{T_{h_0^{-1}}}^{h_0} \quad \text{and} \quad Y_l = X_{T_{h_l^{-1}}}^{h_l} - X_{T_{h_{l-1}^{-1}}}^{h_{l-1}}, \quad l = 1, \ldots, L. \quad (9)$$

We can then formally write that

$$\mu = \sum_{l \geq 0} \mathbb{E} [Y_l]. \quad (10)$$

Question 7: Justify that the above sum converges absolutely, and find an upper bound for the truncation error incurred by approximating $\mu \approx \sum_{l=0}^L \mathbb{E} [Y_l]$.

With this in mind, we can aim to approximate $\mu$ by taking a truncation level $L$, a sequence of level sizes $\{N_l\}_{l=0}^L$, and forming the Multi-Level Monte Carlo estimator (MLMC)

$$\hat{\mu}_{N_l,L} \triangleq \sum_{l=0}^L \left[ \frac{N_l}{N_l} \sum_{i=1}^{N_l} Y_l^i \right]. \quad (11)$$

where for each $i$, $\{Y_l^i\}_{i=1}^{N_l}$ are independent, identically-distributed (iid) samples of $Y_l$, i.e., for each $(i, l)$ we independently draw an initial point $X_0^{(i,l)} \sim \text{Uniform}([-1, 1])$ and define $Y_l^i$ as in display (9) using $X_0^{(i,l)}$ rather than $X_0$. Hence, $\{Y_l^i\}_{i,l}$ are mutually independent.
**Question 8:** For $\theta \in \{\frac{k\pi}{27}\}_{k=1}^{26}$, compute $\hat{\mu}_{N_{1:L}}$, taking $L = 10$ and using level sizes i) $N_l \equiv 5$ and ii) $N_l = 2^{L-l}$. Which estimator exhibits greater variability?

**Question 9:** Derive an upper bound for the MSE of the MLMC estimator with truncation level $L$ and level sizes $\{N_l\}_{l=0}^L$.

We now try to choose $L$ and $\{N_l\}_{l=0}^L$ such that the MSE of the resulting MLMC estimator is minimised given a fixed computational budget.

**Question 10:** Suppose that the computational budget is $C$, i.e. the cost of generating all of the random variables used in the MLMC estimator is bounded above by $C$. By treating the $N_l$ as continuous variables, $L = \infty$, derive an allocation of level sizes $\{\tilde{N}_l\}_{l=0}^L$ which minimises the upper bound for the MSE of the resulting MLMC estimator derived in question 9. [Hint: To minimise a function $F(x)$ subject to the constraint that $G(x) = c$, it suffices to identify stationary points of $H(x, \lambda) = F(x) + \lambda(G(x) - c)$. This is known as the method of Lagrange multipliers.]

From now on, we move back to integer-valued levels by taking $N_l = \lfloor \tilde{N}_l \rfloor$.

**Question 11:** How do you find that $L$ scales with $C$? Derive an expression for how the optimal MSE scales as $C$ grows, and compare this to the MC estimator from Question 5.

5 Application to Double-Well Loss Function

We will now use the estimators derived above to study the behaviour of gradient descent on the double-well function $f_\theta$ defined earlier.

**Question 12:** Define $m_1(\theta)$ and $m_2(\theta)$ as the local minima of $f_\theta$ in $[-1, 1]$, defined such that $m_1(\theta) < m_2(\theta)$. Suppose that $h > 0$ and $T \in \mathbb{N}$ are sufficiently small and large, respectively, so that $\min\{|m_1(\theta) - X_{T}^{h}|, |m_2(\theta) - X_{T}^{h}|\} \approx 0$ for any initial point $x_0 \in [-1, 1]$.

Define

$$p_1(\theta) = \mathbb{P}\left(\lim_{T \to \infty} X_{T}^{h} = m_1(\theta)\right) \quad \text{and} \quad p_2(\theta) = \mathbb{P}\left(\lim_{T \to \infty} X_{T}^{h} = m_2(\theta)\right).$$

Derive an expression for $p_1(\theta)$ and $p_2(\theta)$ in terms of $\mu, m_1(\theta)$ and $m_2(\theta)$.

**Question 13:** Use a Multi-Level Monte Carlo scheme with the optimal level sizes, as derived in Question 9, to form estimates of $p_1(\theta)$ and $p_2(\theta)$ for $\theta \in \{\frac{k\pi}{27}\}_{k=1}^{26}$. Plot your estimates to show how they vary with $\theta$. For which values of $\theta$ does the outcome of running gradient descent on $f_\theta$ vary most?