UNIVERSITY OF
CAMBRIDGE

Mathematical Tripos Part II

Astrophysics Tripos Part II

Computational Projects

2015-16

CATAM

Mathematical Tripos Part II
Astrophysics Tripos Part II

Computational Projects

July 2015

Edited by the Computational Projects Assessors Committee

Course Director: Dr M.B. Wingate

Assistant Course Director: Dr M. Spivack

Computer-Aided Teaching of All Mathematics

Faculty of Mathematics
University of Cambridge

Contents

You may choose freely from the projects above, independently of whether you are studying the examinable courses with which your chosen projects are connected. For up-to-date information on the maximum credit for the Computational Projects in Part II of the Mathematical Tripos, and the total number of units required to achieve that maximum, please consult both the Undergraduate Schedules of the Mathematical Tripos and § 2.1 of the Introduction. Please also see § 2.1 of the Introduction for more information on how credit is awarded.

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 course is a continuation of the Part IB Computational Projects course. The aim is to continue your study of the techniques of solving problems in mathematics using computational methods.

As in Part IB, **the course 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 2015-16. At the time of writing (July 2015) the booklet for the academic year 2014-15 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 notebooks. The maximum credit obtainable is 150 marks and there are no alpha or beta quality marks. Credit obtained is added directly to the credit gained in the written examination. The maximum contribution to the final merit mark is thus 150, which is the same as the maximum for a 16-lecture course.*

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. 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

You should work at your own speed on the projects contained in this booklet, which cover a wide range of mathematical topics.

A lecture covering administrative aspects of the course is given towards the start of the Michaelmas Term. You may also wish to take advantage of programming tutorials available on-line if you did not attempt the Computational Projects course in Part IB.

Your write-ups must be submitted in the Easter Term (see §6.2 below). Please also note that **you must be available** in the last week of Easter term in case you are called either for a routine *Viva Voce Examination*, or for an *Examination Interview* or an *Investigative Meeting* if unfair means are suspected (see §5.1 below).

1.3.1 Planning your work

- You are strongly advised to complete all your computing work by the end of the Easter vacation if at all possible, since the submission deadline is early in the Easter Term.
- **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]

As was the case last year, the Faculty of Mathematics is supporting MATLAB for Part II. 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 University and College Managed Cluster Service (MCS).

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 manual[s]

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

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

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

- MATLAB has its own built-in help and documentation.
- The suppliers of MATLAB, *The MathWorks*, 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[1]

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

  A printable version is available from

  `http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf`
- *The MathWorks* provide links to a whole a raft of other tutorials; see

  `http://www.mathworks.co.uk/academia/student_center/tutorials/launchpad.html`

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

  `http://www.mathworks.co.uk/help/matlab/index.html`
- There is a plethora of books on MATLAB. For instance:

  (a) *Numerical Computing with MATLAB* by Cleve Moler[2] (SIAM, Second Edition, 2008, ISBN 978-0-898716-60-3). This book can be downloaded for free from

  `http://www.mathworks.co.uk/moler/chapters.html`

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

  Further, Google returns about 44,000,000 hits for the search 'MATLAB introduction' (up from 24,300,000 hits last year), and about 1,070,000 hits for the search 'MATLAB introduction tutorial' (down from 3,210,000 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, but *you are free to write your programs in any computing language whatsoever*. Python, C, C++, C#, Java, Visual Basic, Mathematica and Maple have been used by several students in the past, and Excel has often been used for plotting graphs of computed results. A more complete list of possible alternative languages is provided in Appendix A.The choice is your own, provided your system can produce results and program listings on paper.[3]

However, you should bear in mind the following points.

[1] These links work at the time of writing. Unfortunately *The MathWorks* have an annoying habit of breaking their links.

[2] Cleve Moler is chief mathematician, chairman, and co-founder of MathWorks.

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

- The Faculty does *not* promise to help you with programming problems if you use a language other than MATLAB.
- Not all languages have the breadth of mathematical routines that come with the MATLAB package. You may discover either that you have to find reliable replacements, or that you have to write your own versions of mathematical library routines that are pre-supplied in MATLAB (this can involve a fair amount of effort). To this end you may find reference books, such as *Numerical Recipes* by W. H. Press *et al.* (CUP), useful. You may use equivalent routines to those in MATLAB from such works so long as you acknowledge them, and reference them, in your write-ups.
- If you choose a high-level programming language that can perform advanced mathematical operations automatically, then you should check whether use of such commands is permitted in a particular project. As a rule of thumb, do not use a built-in function if there is no equivalent MATLAB routine that has been approved for use, or if use of the built-in function would make the programming considerably easier than intended. For example, use of a command to test whether an integer is prime would not be allowed in a project which required you to write a program to find prime numbers. The *CATAM Helpline* (see §4 below) can give clarification in specific cases.
- Subject to the aforementioned limited exceptions, *you must write your own computer programs*. Downloading computer code, e.g. from the internet, that you are asked to write yourself counts as plagiarism (see §5).

1.4.4 Computer Algebra Systems

Some projects require the use of a Computer Algebra System (CAS). At present none is specifically recommended but possible choices include the Symbolic Math Toolbox in MATLAB, Mathematica and Maple.

Mathematica is installed on the Mathematics and the Central/College MCS, and at the time of writing it is expected that Maple will be available on the Mathematics MCS, and possibly on the University/College MCS (see §3). Mathematica and Maple are also sensible choices for several projects other than the ones for which a CAS is actually required, and you should feel free to use them for *any* of the projects, but you should be aware of a few points:

- For intensive numerical calculations Maple should be told to use the hardware floating-point unit (see help on `evalhf`).
- If you choose to use Maple, Mathematica, or any other CAS to do a project for which a CAS is not specifically required, you should bear in mind that you may not be allowed to use some of the built-in functions (see §1.4.3).

2 Project Reports

2.1 Project write-ups: examination 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 150 marks are available for the Computational Projects course. No quality marks (i.e. αs or βs) are awarded. The maximum contribution to the final merit mark is thus 150 and the same as the maximum for a 16-lecture course.

2.1.1 Examination credit: algorithm applied to the mark awarded for each project

Each project has a unit allocation. The mark awarded for each project is weighted according to the unit allocation, with each project unit equating to a maximum of 5 Tripos marks. The weighted marks for each project are summed to obtain a candidate's total Tripos mark.

To obtain maximum credit, you should submit projects with unit allocations that sum to 30 units. If you submit m units, where $m > 30$ (i.e. if you submit more then the maximum number of units), then each of your marks is multiplied by a factor $30/m$ to ensure that you can score no more than the overall maximum available, i.e. 150 Tripos marks.[4]

A fractional total Tripos mark resulting from the weighting/scaling process is rounded up or down to the nearest integer, with an exact half being rounded up.

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.

[4] Needless to say, if you submit fewer than 30 units no upwards scaling applies.

- 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*.[5] If you are actually asked to *prove* a result, you should do so briefly.

- 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.

- 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

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

$O\big(f(n)\big)$ or $\Omega\big(f(n)\big)$ notation, where n is some reasonable measure of the size of the input (say the number of vertices of a graph) and f is a reasonably simple function. Examples of simple operations are the addition or multiplication of two numbers, or the checking of the (p, q) entry of a matrix to see if it is non-zero; with this definition finding the scalar product of two vectors of length n takes order n operations. Note that this measure of complexity can differ from the number of MATLAB commands/'operations', e.g. there is a single MATLAB command to find a scalar product of two vectors of length n.

- What is the accuracy of the numerical method? Is it particularly appropriate for the problem in question and, if so, why? How did you choose the step-size (if relevant), and how did you confirm that your numerical results are reliably accurate?
- 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.

2.2.1 Project write-ups: advice on length

The word *brief* peppers the last few paragraphs. However, each year some students just do not get it. To emphasise this point, in general **eight sides of A4 of text**, *excluding in-line graphs, tables, etc.*, should be plenty for a clear concise report of a seven or eight unit project.[6] Indeed, the best reports are sometimes shorter than this.

To this total you will of course need to add tables, graphs, printouts etc. However, *do* ***not*** *include every single piece of output you generate*: include a selection of the output that is a *representative* sample of graphs and tables. It is up to you to choose a selection which demonstrates all the important features but is reasonably concise. Remember that you are writing a report to be read by a *human being*, who will not want to wade through pages and pages of irrelevant or unimportant data. Twenty sides of graphs would be excessive for most projects, even if the graphs were printed one to a page.[7] 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. However, *in a change from previous years*, 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

[6] Reports of projects with fewer/more units might be slightly shorter/longer.
[7] Recall that graphs should not as a rule be printed one to a page.

processor, you should confirm that you will be able to generate submittable output in PDF form. Please note that a PDF file generated by scanning a document is ***not*** acceptable; in particular, and for the sake of clarity, a PDF file generated by scanning a hand-written report 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* and *LibreOffice*. The former is commercial, while the latter can be installed for free for, *inter alia*, the Windows, MacOS and Linux operating systems from

`http://www.libreoffice.org/download/libreoffice-fresh/` and
`http://www.libreoffice.org/download/libreoffice-stable/`.

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` and
- `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 proTeXt distribution, based on MiKTeX, includes the TeXstudio front end.

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

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

TeXnicCenter. TeXnicCenter is another potential front end for Windows users (see also `http://www.maths.cam.ac.uk/computing/win7/home_texniccenter.html`).

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

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

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

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

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

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

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

  Mac and Unix users should already have an unzip utility, while Windows users can download 7-Zip if they have not.
- Please note that this introduction is still under development; you will be notified of updates via *CATAM News*.

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-h.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. Your **name or user identifier should not appear anywhere** in the write-up (including any printouts), as the scripts are marked anonymously.

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 MCS, e.g. broken hardware, printers that are not working, to catam-help@maths.cam.ac.uk. Note that this is a *different* address to that of the *CATAM Helpline*.

You can also use other computing facilities around the University; for further information (including which Colleges are linked to the MCS network) see[8]

[8] 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 also http://www.ucs.cam.ac.uk/desktop-services/mcs/local-access/hours).

http://www.ucs.cam.ac.uk/desktop-services/mcs/

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

Possibly one of the easiest ways to ensure that your work is backed up 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.

4 Information Sources

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

The CATAM Web Page. The CATAM web page,

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

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

CATAM News and Email. Any important information about CATAM (e.g. corrections to projects or to other information in the *Handbook*, 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 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 2015 you can check that you are on the appropriate Faculty year list by referring to the https://lists.cam.ac.uk/mailman/raven webpage (to view this page you

[9] In the past a student has lost his CATAM work courtesy of a stolen computer.

will need to authenticate using Raven if you have not already done so). You should check that the *Maths-II* 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 II' 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.[10]

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.

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

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. If a general discussion *either* is happening regularly *or* gets to the point where physical or virtual notes are being exchanged (even on the back of an envelope, napkin or stamp), then it has reached the stage of unacceptable collaboration. Indeed, assuming that you are interpreting the phrase *occasional general discussion* in the spirit that it is written, then if you have got to the stage of wondering whether a discussion has reached the limit of acceptable collaboration, or you have started a legalistic deconstruction of the term *acceptable collaboration*, you are almost certainly at, or past, the limit.

Example. As an instance to clarify the limits of 'acceptable collaboration', if an assessor reading two anonymous write-ups were to see significant similarities in results, answers, mathematical approach or programming which would clearly not be expected from students working independently, then there would appear to be a case that the students have breached the limits. An *Examination Interview* or an *Investigative Meeting* would then be arranged (unless such similarities were deemed to be justified in light of the declared lists of discussions, see below).

The following actions are examples of *unfair means*

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

These comments apply just as much to copying from the work of previous Part II 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 II student access to the work you have undertaken for your own CATAM projects, even after you have submitted your write-ups. If you knowingly give another student access to your CATAM work, whatever the circumstances, you will be penalised yourself.

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

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

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

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

Faculty Guidelines on Plagiarism. You should also be familiar with the Faculty of Mathematics *Guidelines on Plagiarism* that are reproduced as Appendix C. 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 D. 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 any 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 *Examination Interview* or an *Investigate Meeting* (see §5.1). Either case may lead to a change in the marks you receive for the Computational Projects.

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

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

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

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

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

Sanctions. If plagiarism, collusion or any other method of unfair means is suspected in the Computational Projects, normally the Chair of Examiners will convene an *Examination Interview* (see §5.1). The Computational Projects are considered to be a single piece of work within the Mathematical Tripos; therefore, **if it is concluded that you have used unfair means for the whole or any part of the Computational Projects the likely outcome is that you will receive a mark of 0 for the Computational Projects *in their entirety*.**

If the Chair of Examiners deems the unfair means to be sufficiently significant, an *Investigative Meeting* will held (see §5.1). One outcome of such a meeting could be that the case is referred to the University Advocate, who may send the case to the Court of Discipline. According to the University guidance given to Cambridge students regarding discipline[11]

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

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

We wish to emphasise that the great majority of candidates have had no difficulty in keeping to these guidelines in the past; if you find them unclear in any way you should seek advice from the *CATAM Helpline*. These policies and practices have been put in to place so that you can be sure that the hard work you put into CATAM will be fairly rewarded.

5.1 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

[11] From http://www.cambridgestudents.cam.ac.uk/new-students/rules-and-legal-compliance/discipline.

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

Examination Interviews. For most cases of suspected plagiarism, collusion or other unfair means the Chair of Examiners may summon a particular candidate or particular candidates for interview on any aspect of the written work of the candidate or candidates not produced in an examination room which in the opinion of the Examiners requires elucidation. If any work is deemed to have been plagiarised or otherwise produced using unfair means, the Examiners have the power to award a mark of 0 for the Computational Projects *in their entirety.* At the discretion of the Examiners and if mitigating circumstances warrant, the Examiners may award a mark greater than 0.

Investigative Meetings. For the most serious cases of suspected plagiarism, collusion or other unfair means, the Chair of Examiners may summon a candidate to an *Investigative Meeting.* If this happens, you have the right to be accompanied by your Tutor (or another representative at your request). The reasons for the meeting, together with copies of supporting evidence and other relevant documentation, will be given to your Tutor (or other representative).[12]

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. These will usually take place during the last week of Easter Full Term. *Viva Voce Examinations* are likely to take place on the Monday of the last week (i.e. Monday 6th June 2016), while *Examination Interviews* and *Investigative Meetings* may take place any time that week. If you are required to attend a *Viva Voce Examination*, an *Examination Interview* and or an *Investigative Meeting* you will be informed in writing just after the end of the written examinations. **You must be available** in the last week of Easter 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 D). 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.

[12] For more information see
http://www.admin.cam.ac.uk/univ/plagiarism/examiners/investigative.pdf.

6.2 Submission of written work

In order to gain examination credit, you must:

- submit electronic copies of your reports and programs (see §6.3);
- complete and 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.

The submission date is

Wednesday 27th April 2016, 10am–4pm.

No submissions can be accepted before this time, and **4pm on 27th April 2016 is the final deadline**. After this time, projects may be submitted only under exceptional circumstances via your College Tutor, with a letter of explanation. In any case, the CATAM Director will be permitted 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 ask the Computing Service to reset it.[13] This may take some time, so check that you know your UIS password well before submission day.

[13] E.g. see http://www.ucs.cam.ac.uk/docs/faq/accounts/n3.

Naming convention. To make submission and marking easier, please put all your source files related to different projects in separate directories/folders. Further, please name each directory/folder using a convention whereby the first few characters of the directory/folder name give the project number, with the dot replaced by either a minus sign or underscore (`_`). For example, all the programs written for project 2.3 should be placed in a directory/folder with a name beginning with `2-3` or `2_3`.

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).

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: Other Computer Languages and Packages

There are many computer packages and languages suitable for mathematics, and none is "best". Prior to MATLAB, the supported languages for CATAM were, respectively, FOCAL, BASIC, Pascal and C. Should you need, someday, to tackle a serious piece of computation then you will need to consider which language or package is most suitable. Factors in this decision include ease of programming, speed of execution and, in some cases, cost of purchase.

Some languages are *compiled* languages where source code written in the language has to be translated to machine code before it can be executed on a computer; other languages are *interpretative* languages such that no translation is necessary before execution (although in practice this distinction can get blurred, for example because most interpreting systems also perform some translation work, just like compilers). Interpretative languages allow you to type simple, or quite complicated, commands directly in to a window, after which the commands are interpreted and the results are printed out directly. Languages like this tend to be easier to learn, and simple programs can be quickly tested. The downside of interpretative languages is that the code typically executes slower than a compiled language. Many people therefore aim for the best of both worlds, by using an interpreted language initially to try simple programming ideas, and then transferring the developed ideas to a compiled language for more intensive work. The various packages and languages can differ a lot in detail, but the fundamental principles are fairly similar, and the experience of doing CATAM should make it much easier for you to pick up another language. It should be added that there are many tools designed for specific mathematical purposes (not mentioned here), so it makes sense to ask other people what they use.

Below are some general packages and languages that you might come across. The list is not complete, nor is it a list of recommendations. Further, while you are welcome to do the CATAM projects in any programming language,[14] **the Faculty only provides support for MATLAB; if you choose a language other than MATLAB you cannot expect support from the Faculty.**

A.1 Mathematical languages and packages

There are a number of programming languages and packages that have been specifically designed for mathematics. Some are specialised (e.g., the software package Magma has been designed to solve computationally hard problems in algebra, number theory, geometry and combinatorics), while others have more general applicability. All packages have their pros and cons, and devoted adherents and detractors. Some are *proprietary*, in which case they will often cost you or others money (but will usually come with professional support), while others are *free* (but will not come with support).[15] An advantage of *open source* software (normally free) is that you can see what is going on under the hood. Hence an attractive feature of, say, R (see below) is that once you have prototyped in R and are ready to make a production version in, say C, the C code under the hood is readily available for linking or cutting and pasting into your C code.

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

[15] It should be noted that while the user interface of these packages can be significantly different, this is often not the case under the hood. For instance, in almost all cases the vector/matrix operations which one uses to program in these languages are essentially implemented by the same (free) C and FORTRAN libraries written 40+ years ago; similarly for the implementations of other common tasks such as optimisation, the numerical solution of ODEs, PDEs, etc.

Below we list a number of packages that have been used for CATAM in the past, or might be suitable for CATAM.[16] **However, we emphasise that the Faculty only provides support for MATLAB.**

MATLAB is a *proprietary* numerical computing environment and programming language that allows easy implementation of numerical algorithms, as well as visualisation. It has a graphical debugger, and a Symbolic Math Toolbox allowing access to computer algebra capabilities. There is a comprehensive help facility, and extensive documentation. MATLAB is available on both the Mathematics and the Central/College MCS.

Octave is a *free* numerical computing environment which is mostly compatible with MATLAB (the Octave FAQ notes that there are still a number of differences between Octave and MATLAB, but in general differences between the two are considered as bugs). Octave does not have MATLAB's graphical interface. Programs might run at different speeds under MATLAB and under Octave, even on the same machine, due to the way the commands are executed; Octave is in general slower. Octave is available for free download for the Linux, MacOS and Windows operating systems from http://www.gnu.org/software/octave/.

Scilab is a *free* numerical computing environment. Like all the above packages it is a high level programming language. It is similar in functionality to MATLAB, and the syntax is similar, but not identical to MATLAB (Scilab includes a package for MATLAB-to-Scilab conversions). There is a graphical user interface. Scilab is available for free download for the Linux and Windows operating systems, and Intel versions of MacOS, from http://www.scilab.org/.

Maple is a general purpose *proprietary* mathematics software package that supports both symbolic computations and arbitrary precision numerical calculations, as well as visualisation (i.e., plotting of functions and data). It has a graphical debugger. There is a comprehensive help facility, and extensive documentation. 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.

Mathematica is another general purpose *proprietary* mathematics software package that supports both symbolic computations and arbitrary precision numerical calculations, as well as visualisation. It has a graphical debugger. It is available on both the Mathematics and the Central/College MCS. Under an agreement with the University[17] mathematics students can download versions of Mathematica for the Linux, MacOS and Windows operating systems from

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

R is a *free* programming language and software environment for statistical and numerical computing, and graphics. R uses a command line interface though several graphical user interfaces are available. For numerical calculations it has similar functionality (but *not* the same syntax) as MATLAB and Octave. R is available for free download for the Linux, MacOS and Windows operating systems from http://www.r-project.org/.

[16] Further comparisons are available from WIKIPEDIA, e.g.,

- http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems
- http://en.wikipedia.org/wiki/Comparison_of_numerical_analysis_software
- http://en.wikipedia.org/wiki/Comparison_of_statistical_packages

[17] At the time of writing this agreement is due to expire during summer 2016, but the agreement will most probably be renewed.

A.2 More General Purpose Languages

There are many popular languages in this class (such as C, C++, FORTRAN, Java, Python and so on). To use many of these languages you will need a compiler to convert your program (stored in a text file) into an executable binary file which can then be run.

C is an extremely widely used general purpose programming language. It allows complete control of data at the level of bits and bytes and is very efficient; much of the software you use every day was written in C. More complicated data structures (e.g., polynomials) can be handled by writing suitable functions. However while C offers great flexibility in handling data structures, the syntax is not particularly intuitive and bugs may be hard to detect.[18] There are a number of C compilers available. The best known *freely* available compiler is `gcc` available from http://gcc.gnu.org/.

C++ is another general purpose programming language that is a development of C aimed at higher-level structures (e.g., it introduces some object-oriented features to C). It is in widespread use, for example in the banking sector. As with C there are a number of C++ compilers available; the best known *freely* available compiler is `g++` available from http://gcc.gnu.org/.

C# is a simple, modern, general-purpose, object-oriented programming language. Some view this a Microsoft's answer to Java (see below); others do not.

FORTRAN is one of the oldest languages around. It is a general-purpose programming language that is especially suited to numerical computation and scientific computing. It was designed for computation with real numbers, and its evolved form remains popular in universities and in industry because it is still excellent for this purpose. The best known *freely* available compiler is `gfortran` available from http://gcc.gnu.org/.

Java is a programming language that derives much of its syntax from C and C++. Unlike C it is an interpreted language where the interpreter is the, so called, Java Runtime Environment (JRE). Java code will run on any architecture on which a JRE is installed without needing to be recompiled (as a result Java is popular for web applications). Java is available for *free* from http://www.oracle.com/technetwork/java/.[19] The computer Laboratory teaches Java to its students (see http://www.cl.cam.ac.uk/teaching/1314/ProgJava/).

Python is a *free* general-purpose high-level programming language. Its design philosophy emphasises programmer productivity and code readability. Python has a large standard library providing tools suited to many disparate tasks. Because of the wide variety of tools provided by the standard library combined with the ability to use a lower-level language such as C and C++, Python is sometimes viewed as a powerful *glue* between languages and tools. Python is available for *free* download for the Linux, MacOS and Windows operating systems from http://www.python.org/.

Julia is a relatively new kid on the block, and is a high-level dynamic programming language designed to address the requirements of high-performance numerical and scientific computing while also being effective for general purpose programming (see http://julialang.org/).

[18] As such it may not be the ideal language for a beginner to learn.
[19] There is also the *freely* available `gcj`, the GNU Compiler for Java (see http://gcc.gnu.org/java/).

B Appendix: University Statement on Plagiarism

The General Board, with the agreement of the Board of Examinations and the Board of Graduate Studies, has issued this guidance for the information of candidates, Examiners, and Supervisors.[20] It may be supplemented by course-specific guidance from Faculties and Departments.

Plagiarism is defined as submitting as one's own work, irrespective of intent to deceive, that which derives in part or in its entirety from the work of others without due acknowledgement. It is both poor scholarship and a breach of academic integrity.

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

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

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

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

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

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

Failure to conform to the expected standards of scholarship (e.g. by not referencing sources) in examinations may affect the mark given to the candidate's work. In addition, suspected cases

[20] For the latest version of this statement see

http://www.admin.cam.ac.uk/univ/plagiarism/students/statement.html

of the use of unfair means (of which plagiarism is one form) will be investigated and may be brought to one of the University's Courts. The Courts have wide powers to discipline those found guilty of using unfair means in an examination, including depriving such persons of membership of the University, and deprivation of a degree.

Discipline Regulation 6

No candidate shall make use of unfair means in any University examination. Unfair means shall include plagiarism[21] and, unless such possession is specifically authorised, the possession of any book, paper or other material relevant to the examination. No member of the University shall assist a candidate to make use of such unfair means.

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

C Appendix: Faculty of Mathematics Guidelines on Plagiarism

For the latest version of these guidelines please see

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

University Resources

The University publishes information on *Good academic practice and plagiarism*, including

- a *University-wide statement on plagiarism*;
- *Information for students*, covering
  - *Your responsibilities*
  - *Why does plagiarism matter?*
  - *Using commercial organisations and essay banks*
  - *How the University detects and disciplines plagiarism*;
- information about *Referencing and study skills*;
- information on *Resources and sources of support*;
- *FAQs*.

There are references to the University statement

- in the Part IB and Part II Computational Project Manuals,
- in the Part III Essay booklet, and
- in the M.Phil. Computational Biology Course Guide.

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

The Faculty Guidelines

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

What is plagiarism?

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

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

Checking for plagiarism

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

The scope of plagiarism

Plagiarism may be due to

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

How to avoid plagiarism

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

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

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

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

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

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

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

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

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

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

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

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

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

Links to University Information

- Information on *Good academic practice and plagiarism*, including
  - *Information for students.*
  - information on *Policy, procedure and guidance for staff and examiners.*

D Appendix: Example Submission Form

PART II MATHEMATICAL TRIPOS 2015-16

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. In particular, that means working out the credit-unit total in advance.
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 everyone else will also 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 6 reads:

> No candidate shall make use of unfair means in any University examination. Unfair means shall include plagiarism[22] and, unless such possession is specifically authorized, the possession of any book, paper or other material relevant to the examination. No member of the University shall assist a candidate to make use of such unfair means.

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

> *The Faculty of Mathematics wishes to make it clear that failure to comply with this requirement is a serious matter. It could result in all your marks for the Computational Projects being removed and also render you liable to further sanctions from the Examiners or the University Courts.*

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

DECLARATION BY CANDIDATE

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

Project Number	**Brief Title**	**Credit Units**
	Total Credit Units	

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 .....

1 Numerical Methods

1.1 Fourier Transforms of Bessel Functions (6 units)

This project assumes only material contained in Part IA and IB core courses. The Part II courses on Numerical Analysis, Further Complex Methods and Asymptotic Methods may provide relevant but non-essential background.

1 Introduction

Bessel's Equation of order n is the linear second-order equation

$$x^2y'' + xy' + (x^2 - n^2)y = 0. \tag{1}$$

Bessel Functions of the first kind are solutions of (1) which are finite at $x = 0$. They are usually written $J_n(x)$.

Question 1 Investigate (1) for $n = 0$, 1, 4 using a Runge–Kutta (or similar) method commencing the integration for a positive value of x and arbitrary values of y and y'. Integrate forwards *and* backwards in x for a few such initial conditions, plotting y. Describe what you observe, and illustrate interesting behaviour by five or so plots in your write-up.

Now try starting at $x = 0$. What happens, and why?

Question 2 The series solution for $J_n(x)$ is

$$J_n(x) = \sum_{r=0}^{\infty} \frac{(-1)^r(\frac{1}{2}x)^{2r+n}}{r!(n+r)!}. \tag{2}$$

Write a program to sum this series, and plot $J_n(x)$ for $n = 0$, 1, 4 for a range of x, e.g., for $0 \leqslant x \leqslant 100$. Identify a range of x for which this summation method is not accurate and explain why.

2 The Discrete Fourier Transform

The Fourier Transform $\hat{F}(k)$ of a function $F(x)$ may be defined as

$$\hat{F}(k) = \int_{-\infty}^{+\infty} F(x)\exp(-2\pi ikx)\,dx. \tag{3}$$

If $F(x)$ is a function which is only appreciably non-zero over a limited range of x, say $0 < x < X$, then it is possible to approximate $\hat{F}(k)$ by means of finite sums. Suppose

$$F_r = F(r\Delta x) \qquad \text{for } r = 0, \ldots, N-1,$$

where $\Delta x = X/N$. An approximation to (3), known as the Discrete Fourier Transform (DFT), is

$$\hat{\mathcal{F}}_s = \frac{X}{N}\sum_{r=0}^{N-1} F_r\,\omega_N^{-rs}, \tag{4}$$

where $\omega_N = e^{2\pi i/N}$. The *exact* inverse of (4) is

$$F_r = \frac{1}{X} \sum_{s=0}^{N-1} \hat{\mathcal{F}}_s \, \omega_N^{rs}. \tag{5}$$

In order to deduce the relationship between the $\hat{\mathcal{F}}_s$ and $\hat{F}(k)$, we first note from (4) that $\hat{\mathcal{F}}_s$ represents values of the Fourier Transform spaced by the "wavenumber" interval Δk, where $\Delta k = 1/X$. Also $\hat{\mathcal{F}}_s$ is periodic in s with period N; this corresponds to a "wavenumber" periodicity

$$K = N\Delta k = N/X = 1/\Delta x.$$

Now it is to be expected that (4) will fail to approximate to (3) when the exponential function oscillates significantly between sample points, that is when

$$|k| \gtrsim \frac{1}{2\Delta x} = \tfrac{1}{2}K. \tag{6}$$

This, together with its periodicity, suggests that $\hat{\mathcal{F}}_s$ will be related to $\hat{F}(k)$ by

$$\hat{\mathcal{F}}_s \cong \begin{cases} \hat{F}(s\Delta k) & s = 0, \ldots, \frac{1}{2}N - 1, \\ \hat{F}(s\Delta k - K) & s = \frac{1}{2}N, \ldots, N - 1. \end{cases} \tag{7}$$

Thus (5) is an approximation to

$$F(x) \cong \int_{-K/2}^{+K/2} \hat{F}(k) \exp(2\pi i k x) \, dk. \tag{8}$$

Because of the periodicity, the $\hat{\mathcal{F}}_s$ are usually thought of as a series with $s = 0, \ldots, N-1$, the upper half being mentally re-positioned to correspond to negative "wavenumber". Note that if $F(x)$ is real, and * denotes a complex conjugate, then

$$\hat{F}(k) = \hat{F}^*(-k). \tag{9}$$

Question 3 Under what limiting processes for N and X, possibly after a suitable change in origin in x, does the DFT tend to the Fourier Transform?

The DFT is best evaluated by the Fast Fourier Transform (FFT) method. You may use the MATLAB routine `fft`, or an equivalent routine in any other package, or you may write your own routine (but do ***not*** simply compute (4) directly). The FFT method is described in the Appendix, but it is not necessary to understand any of the details; it is sufficient simply to invoke the routine.

3 Fourier Transforms of Bessel Functions

Question 4 Show analytically that if $F(x)$ is a real even function and

$$I_1 = \int_0^X F(x) \exp(-2\pi i k x) \, dx, \qquad I_2 = \int_{-X}^{+X} F(x) \exp(-2\pi i k x) \, dx,$$

then

$$\mathrm{Im}(I_2) = 0, \qquad \mathrm{Re}(I_2) = 2\mathrm{Re}(I_1). \tag{10}$$

With the definitions of §2, the FFT algorithm is ideally suited to approximating I_1 rather than I_2. Hence if an approximation to I_2 is desired, an approximation to I_1 could first be calculated, and then the relations (10) could be used. If this procedure for calculating I_2 is adopted, and $F_N \neq F_0$, explain why F_0 should be replaced by $\frac{1}{2}(F_0 + F_N)$ before calculating the DFT. What is the equivalent result to (10) if $F(x)$ is a real odd function?

Question 5 Using an FFT, and the results of Question 4, find numerically the Fourier Transform of $J_n(x)$:

$$\hat{J}_n(k) = \int_{-\infty}^{+\infty} J_n(x) \exp(-2\pi i k x)\, dx.$$

Compare it with the theoretical formula

$$\hat{\mathcal{J}}_n(k) = 2(-i)^n (1 - 4\pi^2 k^2)^{-1/2}\, T_n(2\pi k), \tag{11}$$

where $T_n(\mu)$ is the Chebyshev polynomial of order n defined by

$$T_n(\mu) = \begin{cases} \cos n\theta, & \mu = \cos\theta \\ 0, & |\mu| > 1 \end{cases}$$

To obtain $J_n(x)$, you may either devise a method of your own (e.g., a combination of Questions 1 and 2), or you may use the MATLAB procedure `besselj`.

You should obtain results for $n = 0$, 1, 2, 4, and 8. Choose sufficient points in the transform to resolve the functions to your satisfaction subject to reasonable time constraints on whatever computer you are using.

Plots of $J_n(x)$ for a couple of representative values of n should be included in your write-up. You should also include plots of $\hat{J}_n$ and $\hat{\mathcal{J}}_n$ on the same graph. Choose a range of k which allows you to see the detailed behaviour in the interval $-1 \leqslant \pi k \leqslant 1$.

Comment on your results. *Inter alia* you should remark on how the FFT deals with any values of k which might be expected from the theoretical result to give problems, and you should describe the effects of varying N and X; in particular you should ***systematically*** examine how the numerical errors change as N and/or X are varied, e.g. in the light of your answer to Question 3.

You should also find a way to demonstrate from your computational results how the execution time necessary to calculate the transform varies with N.

Appendix: The Fast Fourier Transform

The Fast Fourier Transform (FFT) technique is a quick method of evaluating sums of the form

$$\lambda_r = \sum_{s=0}^{N-1} \mu_s\, \omega_N^{\sigma r s}, \qquad r = 0, \ldots, N-1, \qquad \sigma = \pm 1, \tag{12}$$

where N is an integer, μ_s is a known sequence and $\omega_N = e^{2\pi i/N}$. The "fast" in FFT depends on N being a power of a small prime, or combination of small primes; for simplicity we will assume that $N = 2^M$. Write

$$\lambda_r \longleftrightarrow \mu_s, \qquad r, s = 0, \ldots, N-1$$

to denote that (12) is satisfied. Introduce the half-length transforms

$$\left.\begin{aligned} \lambda_r^E &\longleftrightarrow \mu_{2s} \\ \lambda_r^O &\longleftrightarrow \mu_{2s+1} \end{aligned}\right\} \qquad r, s = 0, \ldots, \tfrac{1}{2}N - 1;$$

then it may be shown that

$$\left.\begin{aligned} \lambda_r &= \lambda_r^E + \omega_N^{sr}\lambda_r^O \\ \lambda_{r+N/2} &= \lambda_r^E - \omega_N^{sr}\lambda_r^O \end{aligned}\right\} \qquad r = 0, \ldots, \tfrac{1}{2}N - 1.$$

Hence if the half-length transforms are known, it costs $\frac{1}{2}N$ products to evaluate the λ_r.

To execute an FFT, start from N vectors of unit length (i.e., the original μ_s). At the sth stage, $s = 1, 2, \ldots, M$, assemble 2^{M-s} vectors of length 2^s from vectors of length 2^{s-1} – this "costs" $2^{M-s} \times \frac{1}{2}(2^s) = 2^{M-1} = \frac{1}{2}N$ products for each stage. The complete discrete Fourier transform has been formed after M stages, i.e., after $O(\frac{1}{2}N \log_2 N)$ products. For $N = 1024 = 2^{10}$, say, the cost is $\approx 5 \times 10^3$ products, compared to $\approx 10^6$ products in naive matrix multiplication!

A description and short history of the FFT are given in the book *Numerical Recipes* by Press *et al.*, chapter 12.

1 Numerical Methods

1.6 Multigrid Methods (10 units)

Knowledge of Part II Numerical Analysis would be advantageous for this project.

1 Solution of Poisson's Equation by Relaxation Methods

We consider the problem of solving Poisson's equation in a square domain with homogeneous Dirichlet boundary conditions

$$\nabla^2 u = f \qquad \text{in} \qquad 0 < x < 1, \quad 0 < y < 1, \tag{1}$$

with $u = 0$ on $x = 0$, $x = 1$, $y = 0$ and $y = 1$.

A numerical solution is attempted by finding values for u at grid points in a square $N \times N$ mesh. The (i, j)th point is given by $(x_i, y_j) = (ih, jh)$ where $h = 1/(N-1)$. The value of $\nabla^2 u$ is approximated at each of the interior points by a finite-difference formula

$$(\nabla^2 u)_{i,j} \simeq \frac{1}{h^2}\left[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}\right]. \tag{2}$$

By requiring that $(\nabla^2 u)_{i,j}$ is equal to $f(x_i, y_j)$ at each of the interior points, we obtain $(N-2)^2$ linear equations for the $(N-2)^2$ unknowns $u_{i,j}$, $(1 \leqslant i \leqslant N-2,\ 1 \leqslant j \leqslant N-2)$, of the form

$$\frac{1}{h^2}\left[u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}\right] = f(x_i, y_j)\ . \tag{3}$$

The values of $u_{i,j}$ at the boundary points are set by the boundary conditions. Here $u_{i,j}$ is equal to zero at each boundary point.

We now have to solve these linear equations as quickly and as accurately as possible. Note that even if the solution of the linear equations were obtained with perfect accuracy, it would still be only an approximate solution to the original partial differential equation, since (2) is only an approximation to (1).

For even relatively modest values of N it is often impractical to solve the $(N-2)^2$ linear equations by direct methods (e.g. Gaussian elimination) because of storage limitations. An alternative approach is to use an iterative "relaxation" method. (3) may be reordered to suggest the iteration scheme

$$u_{i,j}^{n+1} = \frac{1}{4}\left[u_{i-1,j}^{n+1} + u_{i,j-1}^{n+1} + u_{i+1,j}^{n} + u_{i,j+1}^{n} - h^2 f(x_i, y_j)\right] \quad (i = 1, \ldots, N-2, j = 1, \ldots, N-2)\ , \tag{4}$$

where the superscripts denote the number of the iteration; this is conventionally called the Gauss-Seidel scheme. Note the appearance of $(n+1)$th iterates on the right-hand side. The calculation works through the grid with i and j increasing, and updated values are used as soon as they become available.

Question 1 Take $f(x, y) = x(1-x)y^2(1-y)$. Write a program to apply the Gauss-Seidel scheme (4) to solve (3) on an arbitrary sized $(N \times N)$ mesh. Use your program to investigate the convergence properties of the scheme as N varies. In particular, after a reasonably large number of iterations you should calculate:

(a) the variation over the grid of the residual error, $\epsilon^n_{i,j}$, where the residual error is the amount by which (3) is not satisfied, i.e.

$$\epsilon^n_{i,j} = \frac{1}{h^2}\left[u^n_{i+1,j} + u^n_{i-1,j} + u^n_{i,j+1} + u^n_{i,j-1} - 4u^n_{i,j}\right] - f(x_i, y_j)\,, \tag{5}$$

(b) an approximation for the asymptotic rate of convergence, i.e.

$$r_\infty = -\log\left(\lim_{n\to\infty}\left\{\frac{\max_{i,j}|\epsilon^{n+1}_{i,j}|}{\max_{i,j}|\epsilon^n_{i,j}|}\right\}\right)\,; \tag{6}$$

(why is this a good definition of the rate of convergence?).

What do you conclude about the number of iterations needed for convergence to a specified accuracy (e.g. for the magnitude of residual error to be less than a given tolerance at each point)? Estimate as a power of N the number of operations (i.e. additions, multiplications) needed for such convergence. Check your answer by measuring computational time in different cases. Suggested values for N that you might try are 9, 17, 33, 65, etc. Also estimate as a power of N the number of operations needed for convergence to an accuracy consistent with the truncation error of the discretisation of (1).

2 The Multigrid Method

Your calculations should show that the part of the error that decays slowest for each N (and therefore that which dominates after a large number of iterations) has a form very similar to the lowest Fourier mode that will fit into the domain. The convergence is thus limited by large scales, not by small scales.

This motivates the multigrid method described below. The basic idea is that the error left after a few iterations is on scales much larger than the grid scale. The correction needed to the approximate solution to remove this error may therefore be determined more efficiently by transferring the error to a coarser grid, iterating on the coarser grid where convergence is more rapid, then transferring the calculated correction back to the finer grid, updating the approximate solution, and iterating on the finer grid again. The whole procedure is then repeated until the required convergence is achieved. Furthermore the procedure need not be confined to two grids. It is natural to improve the convergence of the coarse grid problem by transferring the error in that to a coarser grid still, and so on.

The multigrid procedure may be defined more exactly as follows. Assume that we have a sequence of K grids, labelled by $J = 1, \ldots, K$ in increasing order of fineness, the Jth grid having size $N_J \times N_J$. It is convenient to take the mesh spacing of the $(J-1)$th grid to be twice that of the Jth grid, i.e. $N_J = 2N_{J-1} - 1$.

On the Jth grid we wish to solve the linear system

$$\mathcal{L}_J \mathbf{u}^{(J)} = \mathbf{r}^{(J)}. \tag{7}$$

where the operator $\mathcal{L}_J$ corresponds to that acting on the left-hand side of (3), if $N_J = N$. [It is important when writing down the form of $\mathcal{L}$ for arbitrary J to remember that h in (3) must be replaced by $1/(N_J - 1)$.]

Descending part of multigrid cycle

(A) Apply the Gauss-Seidel iteration scheme (hereafter G-S) ν_1 times to obtain an approximate solution $\tilde{\mathbf{u}}^{(J)}$. The error $\mathbf{v}^{(J)}$ in this solution therefore satisfies

$$\mathcal{L}_J \mathbf{v}^{(J)} = \mathbf{r}^{(J)} - \mathcal{L}_J \tilde{\mathbf{u}}^{(J)}. \tag{8}$$

(B) Transfer the problem of determining $\mathbf{v}^{(J)}$ to the coarser $(J-1)$th grid as

$$\mathcal{L}_{J-1}\mathbf{u}^{(J-1)} = \mathcal{R}(\mathbf{r}^{(J)} - \mathcal{L}_J\tilde{\mathbf{u}}^{(J)}) = \mathbf{r}^{(J-1)}. \tag{9}$$

The operator $\mathcal{R}$ is known as the *restriction* operator (see below).

The descending part of the cycle repeats (A) and (B), transferring the correction at each stage to coarser and coarser grids, starting with $J = K$ and ending with $J = 2$.

Coarsest grid

(C) On the coarsest grid apply G-S ν_2 times to obtain an approximate solution $\tilde{\mathbf{u}}^{(1)}$.

Ascending part of cycle

(D) Transfer the approximate solution on the $(J-1)$th grid to the Jth grid to give a new approximation to the solution to the problem on that grid

$$\tilde{\mathbf{u}}^{(J)}_{new} = \tilde{\mathbf{u}}^{(J)}_{old} + \mathcal{P}\tilde{\mathbf{u}}^{(J-1)} \tag{10}$$

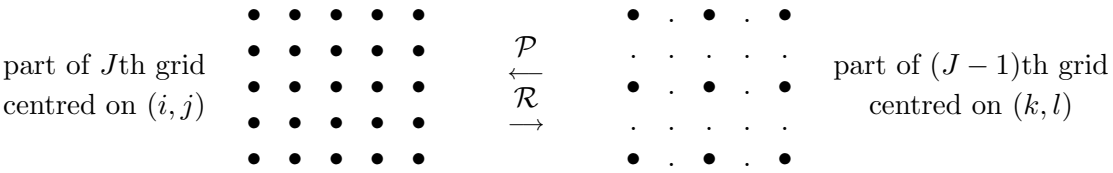
where $\mathcal{P}$ is the prolongation operator (see below).

(E) Apply G-S ν_3 times on the Jth grid to improve the approximation $\tilde{\mathbf{u}}^{(J)}$.

The ascending part of the cycle repeats (D) and (E), starting with $J = 2$ and ending with $J = K$ to leave an approximate solution to the full problem.

Note that within each multigrid cycle, the approximate solution $\tilde{\mathbf{u}}^{(J)}$ and the right-hand side $\mathbf{r}^{(J)}$ are generated from the problem on the $(J+1)$th grid during the descending part of the cycle and must be stored for use again at the Jth level during the ascending part of the cycle. Each $\mathbf{r}^{(J)}$ changes from cycle to cycle, except $\mathbf{r}^{(K)}$ which is always equal to $\mathbf{f}^{(K)}$ (i.e. the vector whose elements are f evaluated at each internal point of the Kth grid).

It remains to specify the restriction and prolongation operators $\mathcal{R}$ and $\mathcal{P}$ that you should use. It is natural to take both to be linear operators. Consider the following two sets of points.



That on the left is a set of points in the Jth grid with the centre point labelled (i, j). That on the right is the same region in the $(J-1)$th grid. In the latter only those points marked with a $\bullet$ are included in the grid, with the centre point now labelled (k, l) say.

The prolongation operator $\mathcal{P}$ maps a function defined on points in the $(J-1)$th grid onto the points in the Jth grid. Similarly the restriction operator $\mathcal{R}$ maps a function defined on points in the Jth grid onto the points in the $(J-1)$th grid. It is convenient to represent both by the "masks"

$$\mathcal{P} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \qquad \mathcal{R} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

That for $\mathcal{P}$ means that if $f = 0$ for all points in the $(J-1)$th grid except that labelled (k, l), then $\mathcal{P}f$ will be zero at all points on the Jth grid except for the square of nine points centred on (i, j) where it will take the values in the "mask". $\mathcal{P}f$ may be evaluated for general functions f by linearity. Each number in the mask for $\mathcal{R}$ represents the contribution from a point in the Jth grid, e.g. the square of nine points centred on (i, j) to the point (k, l) in the $(J-1)$th grid. (Points outside this square make no contribution.)

Question 2 Write a program to apply the multigrid method as specified above. You will probably find it useful to have separate procedures/subprograms, working on grids of arbitrary resolution, to carry out each of the operations of prolongation and restriction, to calculate the residual in the difference equations and to apply the Gauss-Seidel iteration (exploit your existing program from Q1 here).

Apply the multigrid method to the solution of the same equation as in Q1. Investigate the rate of convergence associated with a single multigrid cycle for a fixed resolution of the finest grid, particularly its dependence on

(i) the resolution of the coarsest grid

(ii) the number of times that the G-S iteration is applied at each stage, i.e. ν_1 (on each grid during the descending cycle), ν_2 (on the coarsest grid), and ν_3 (on each grid during the ascending cycle).

To start with, a suggested value for N_K is 65, for N_{K-1} is 33, etc. In each case estimate the number of operations and give a measure of the numerical efficiency. What are your conclusions about the best choices for the resolution of the coarsest grid, and for the numbers ν_1, ν_2 and ν_3? Justify carefully the measure of efficiency that you are using (e.g. remember to include the cost of the extra iterations), and give a brief explanation. Next, choose suitable values of N_1, ν_1, ν_2 and ν_3, and investigate the dependence of the rate of convergence on N_K. Finally, discuss the improvement in efficiency of multigrid over the simple Gauss-Seidel iteration in Q1.

References

[1] Briggs, William L., Henson, Van Emden, McCormick, Steve F. (2000) *A Multigrid Tutorial*, SIAM (ISBN 0-89871-462-1).

2 Waves

2.2 Dispersion (7 units)

This project assumes only the elementary properties of dispersive waves, covered in the Part II course Waves

1 Introduction

This project illustrates the way in which a disturbance in a dispersive system can change its shape as it travels. In order to fix ideas we will consider waves that are modelled by linear partial differential equations that (i) are second order in time, and (ii) have one spatial dimension.

We suppose that the disturbance is described by a function $F(x,t)$, where x is distance and t is time, e.g. F might represent the displacement to the free surface of a fluid. A formal solution for $F(x,t)$ can be obtained from the governing partial differential equation by the standard technique of taking a Fourier transform in the spatial direction. The general solution can then be written in the form

$$F(x,t) = \int_{-\infty}^{\infty} \left(A_1(k)\, \mathrm{e}^{-i\omega_1 t} + A_2(k)\, \mathrm{e}^{-i\omega_2 t} \right) \mathrm{e}^{2\pi i k x}\, dk \ , \tag{1}$$

where the amplitudes, A_1 and A_2, are fixed by the initial conditions, and the frequencies, $\omega_1(k)$ and $\omega_2(k)$, are solutions to the dispersion relation. We will assume that the model system is time-reversible so that $\omega_2(k) = -\omega_1(k)$.

For simplicity we will consider the case where, at $t = 0$,

$$F(x,0) = \frac{1}{\sqrt{\pi}} \exp\left(-\frac{x^2}{\sigma^2} \right) \quad \text{and} \quad \frac{\partial F}{\partial t}(x,0) = 0 \ . \tag{2}$$

The solution (1) then becomes

$$F(x,t) = \int_{-\infty}^{\infty} \hat{F}_0(k) \cos(\omega t)\, \mathrm{e}^{2\pi i k x}\, dk \ , \tag{3}$$

where we have written ω for ω_1, and

$$\hat{F}_0(k) = \int_{-\infty}^{\infty} F(x,0)\, \mathrm{e}^{-2\pi i k x} dx \ . \tag{4}$$

Question 1 If F is real, deduce a property of $\hat{F}_0(k)$.

In order to plot the above solutions some method is needed for evaluating the Fourier integrals in (3) and (4).

2 The Discrete Fourier Transform

Consider the Fourier Transform $\hat{G}(k)$ of a function $G(x)$:

$$\hat{G}(k) = \int_{-\infty}^{\infty} G(x)\, \mathrm{e}^{-2\pi i k x}\, dx \ . \tag{5}$$

Suppose that $G(x)$ is only appreciably non-zero over a limited range of x, say $-X/2 \leqslant x \leqslant X/2$, and that

$$G_r = G(r\Delta x) \quad \text{for} \quad r = -N/2, \ldots, (N/2-1) \ , \quad \text{where} \quad \Delta x = X/N \ . \tag{6}$$

Then an approximation to (5), known as the Discrete Fourier Transform (DFT), is

$$\hat{\mathcal{G}}_s = \frac{X}{N} \sum_{r=-N/2}^{N/2-1} G_r \, \Omega_N^{-rs} \quad \text{for} \quad s = -N/2, \ldots, (N/2-1) \ , \quad \text{where} \quad \Omega_N = \mathrm{e}^{2\pi i/N} \ . \tag{7}$$

The $\hat{\mathcal{G}}_s$ are approximations to the Fourier Transform at $k = s/X$.

The *exact* inverse of (7) is

$$G_r = \frac{1}{X} \sum_{s=-N/2}^{N/2-1} \hat{\mathcal{G}}_s \, \Omega_N^{rs} \ . \tag{8}$$

This is an approximation to

$$G(x) = \int_{-\infty}^{\infty} \hat{G}(k) \, \mathrm{e}^{2\pi i k x} dk \ , \tag{9}$$

based on the assumption that $\hat{G}(k)$ is only appreciably non-zero for $-N/2X \leqslant k \leqslant N/2X$. Hence the DFT and its inverse are asymptotic approximations under the dual limits $X \to \infty$ and $N/X \to \infty$.

3 The Fast Fourier Transform

The Fast Fourier Transform (FFT) technique is a quick method of evaluating sums of the form

$$\lambda_s = \sum_{r=0}^{N-1} \mu_r \, \Omega_N^{\sigma rs} \ , \quad s = 0, \ldots, N-1 \ , \quad \sigma = \pm 1 \ , \tag{10}$$

where the μ_r are a known sequence, and N is a power of a prime, or combination of primes; we will assume that N is a power of 2, say $N = 2^M$. A brief outline of the FFT is given in the appendix for reference, but it is not necessary to understand the details of the algorithm in order to complete the project. Indeed, for the computational part of this project you are advised to use a library FFT procedure, such as the Matlab routine `fft`. However, you are advised to note that sums are typically from 1 to N, while the theory in §2 has sums that run from $-N/2$ to $(N/2-1)$. This means that it may be necessary to re-position the part of the series with r, $s = N/2$, …, $(N-1)$ to $-N/2$, …, -1 (or vice versa), using the periodicity implicit in the definitions (7) and (8). Similar considerations also apply to available routines in other languages, and you should also in general take special care regarding sign conventions and scaling.

4 No Dispersion

Question 2 Write a program to draw graphs of F against x for various t.

Test it for times up to $t = 10$ for $\sigma = 2$ and the "dispersionless" dispersion relation $\omega^2 = 4\pi^2 k^2$. Choose appropriate values for the parameters X and N so that your plots are correct to "graphical accuracy"; present evidence of this accuracy in your write-up. Comment on your results.

5 Gravity Waves

Surface gravity waves in deep water have the dispersion relation

$$\omega^2 = 2\pi g|k|. \tag{11}$$

Question 3 For (11), and the values $\sigma = 10\,\mathrm{m}$ and $g = 9.81\,\mathrm{m\,s^{-2}}$ (in MKS units), draw graphs of F against x for various times.

- For $t = 2$ investigate the effects of changing the values of N and X (try starting with $N = 128$ and $X = 400$). Report the results of this investigation in your write-up, especially with respect to the errors in the solution, using both numerical values and plots. As regards understanding the behaviour of the solution for large $|x|$, you *may* find it helpful to evaluate an approximation to

$$\int_{-\infty}^{\infty} F(x,t)\,dx \tag{12}$$

  for different values of N and X.
- Plot the solution at times $t = 1, 2, \ldots, 6$. Arrange for solutions at more than one time to appear on the same plot. Include in your write-up a set of results that you have verified to be accurate, and that illustrate the behaviour of the disturbance best. Give justification for your choices of N and X. Comment on the features of the evolution, especially as related to the concept of group velocity. Can the concept of group velocity help you choose a suitable value of X for a given time?

6 Capillary Waves

Consider also the capillary-wave dispersion relation for surface waves where surface tension dominates over gravitational effects. This is given by

$$\omega^2 = \frac{8\pi^3\gamma}{\rho}|k|^3\ , \tag{13}$$

where γ is the surface tension and ρ is the density (for water $\gamma = 0.074\,\mathrm{N\,m^{-1}}$ and $\rho = 10^3\,\mathrm{kg\,m^{-3}}$ in MKS units).

Question 4 Perform similar calculations to those in Q3 for water using $\sigma = 10^{-2}$m. Contrast your results with those in Q3. *Inter alia* you will need to choose different values of the time-step and X.

Appendix: The Fast Fourier Transform

The "fast" in FFT depends on the number of modes, N, being a power of a prime, or combination of primes; for simplicity assume that $N = 2^M$. Write

$$\lambda_r \longleftrightarrow \mu_s \quad r, s = 0, \ldots (N-1) \tag{14}$$

to denote that (10) is satisfied. Introduce the half-length transforms

$$\left.\begin{array}{rcl} \lambda_r^E & \longleftrightarrow & \mu_{2s} \\ \lambda_r^O & \longleftrightarrow & \mu_{2s+1} \end{array}\right\} \quad r, s = 0, \ldots, (N/2-1)\ , \tag{15}$$

then it may be shown that

$$\left.\begin{array}{rcl} \lambda_r & = & \lambda_r^E + \Omega_N^{\sigma j}\lambda_r^O \\ \lambda_{r+N/2} & = & \lambda_r^E - \Omega_N^{\sigma j}\lambda_r^O \end{array}\right\} \quad r = 0, \ldots, (N/2-1)\ . \tag{16}$$

Hence if the half-length transforms are known, it costs $\frac{1}{2}N$ products to evaluate the λ_r.

To execute an FFT, start from N vectors of unit length (i.e. the original μ_s). At the s-th stage, $s = 1, 2, \ldots, M$, assemble 2^{M-s} vectors of length 2^s from vectors of length 2^{s-1} — this "costs" $2^{M-1} = \frac{1}{2}N$ products. The complete DFT has been formed after M stages, i.e. after $\mathcal{O}(\frac{1}{2}N\log_2 N)$ products. For $N = 1024 = 2^{10}$, say, the cost is $\approx 5 \times 10^3$ products, compared to $\approx 10^6$ products in naive matrix multiplication!

2 Waves

2.11 Fisher's Equation for Population Dispersal Problems (9 units)

This project is essentially self-contained, and does not directly rely on any Part II lecture course. However, attendance at a Part II Numerical Analysis course may be of some help, as may attendance at the Part II course, Mathematical Biology.

Problem Formulation

An equation commonly encountered in population genetics is the one-dimensional diffusion equation

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} = -\frac{\partial j}{\partial \hat{x}} + F(\hat{\rho}). \tag{1}$$

Here, $\hat{x}$ denotes the spatial position, $\hat{t}$ the time, $\hat{\rho}(\hat{x}, \hat{t})$ the population density, j the population flux, and $F(\hat{\rho})$ is a local source term that describes the net rate of growth in the population density.

A typical model for local population growth is given by the Pearl-Verhulst law

$$F(\hat{\rho}) = \begin{cases} \gamma\hat{\rho}(1 - \hat{\rho}/\hat{\rho}_s) & 0 < \hat{\rho} < \hat{\rho}_s; \\ 0 & \hat{\rho} \leqslant 0 \quad \hat{\rho} \geqslant \hat{\rho}_s. \end{cases} \tag{2}$$

This describes how a homogeneous population would grow, initially in an exponential manner, until the population saturated at some density $\hat{\rho}_s$.

The flux j is the source of the diffusive behaviour and is given by,

$$j = -D\frac{\partial \hat{\rho}}{\partial \hat{x}}. \tag{3}$$

If it is assumed that dispersal is due to random motion of individuals, then the diffusion coefficient D is constant and Fisher's equation is obtained. However, as a remedy to overcrowding, dispersal would be much more effective if the diffusion coefficient were population density-dependent. In fact this has been observed in populations of small animals. Here we consider the case $D = D_0\hat{\rho}$. With suitable non-dimensionalisation, we obtain the modified Fisher equation,

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x}\left(\rho\frac{\partial \rho}{\partial x}\right) + \rho(1 - \rho). \tag{4}$$

A similar equation also arises in combustion dynamics.

Travelling wave solutions to this equation are the subject of project 2.11(a). A situation of more practical interest is when the population density is known at some initial time, and the subsequent evolution of the population is required. In projects 2.11(b) and 2.11(c) the expansion of a population which is initially limited to a finite spatial range is considered. Thus solutions to (4) are required subject to the following boundary conditions:

$$\begin{aligned} \rho(x, 0) &= \begin{cases} \rho_0(x), & 0 \leqslant x \leqslant 1, \\ 0, & x > 1, \end{cases} \\ \frac{\partial \rho}{\partial x}(0, t) &= 0, \qquad t > 0, \\ \rho(x, t) &\to 0 \qquad \text{as} \quad x \to \infty. \end{aligned} \tag{5}$$

The form of the initial data leads us to consider a solution which is piecewise continuous with a single jump across $x = s(t)$, where conditions given by conservation laws must be satisfied. The initial boundary-value problem can now be reformulated as follows:

$$\begin{aligned} 0 \leqslant x \leqslant s(t): & \\ & \rho_t = (\rho\rho_x)_x + \rho(1-\rho), \\ & \rho(x,0) = \rho_0(x), \\ & \rho_x(0,t) = 0, \\ & \rho(s(t),t) = 0, \quad \rho_x(s(t),t) = -\dot{s}(t). \\ s(t) < x: & \\ & \rho(x,t) \equiv 0 \end{aligned} \tag{6}$$

We refer to $x = s(t)$ as the population front. From the initial population distribution (5) we see that $s(0) = 1$.

In project 2.11(b), solutions of (4) are obtained for a particular initial distribution, and the behaviour as $t \to \infty$ is examined. In project 2.11(c), the code developed in project 2.11(b) is used to examine the propagation of the population front.

Project 2.11(a): Travelling Wave Solutions

Here we consider solutions of (4) corresponding to steady expansion of a saturated population. By writing $\rho(x,t) = \phi(\xi)$, where $\xi = x - ct - x_0$, show that ϕ is governed by the nonlinear ODE,

$$\phi\phi'' + (\phi')^2 + c\phi' + \phi(1-\phi) = 0 \tag{7}$$

This is to be solved subject to the boundary conditions $\phi \to 1$ as $\xi \to -\infty$, $\phi \to 0$ as $\xi \to \infty$. General analytic solutions to (7) are not available and hence numerical solutions are required. Such solutions could be obtained by shooting, but here it is preferable to consider the asymptotic form of ϕ as $\xi \to -\infty$.

By linearising (7) about $\phi = 1$, show that as $\xi \to -\infty$,

$$\phi \sim 1 - Ae^{\lambda\xi} \tag{8}$$

where A is an arbitrary constant, due to the translational invariance of (7), and λ is to be determined as a function of c. This then provides suitable initial conditions for a forward integration in ξ, *ie.*,

$$\phi(\xi_0) = 1 - \delta, \qquad \phi'(\xi_0) = -\lambda\delta, \qquad \delta \ll 1, \tag{9}$$

for arbitrary ξ_0.

Obtain solutions for$c = 2, 1.5, 1$ and 0.8 using a suitable integration method. These travelling wave solutions should be plotted on axes with the origin chosen such that $\phi(\xi = 0) = \frac{1}{2}$ (to within graphical accuracy). Investigate the change in the wave profile as the wave speed c is decreased still further.

Show that

$$\phi(\xi) = \begin{cases} 1 - e^{(\xi-\xi_1)/\sqrt{2}}, & -\infty < \xi < \xi_1; \\ 0 & \xi_1 \leqslant \xi. \end{cases} \tag{10}$$

is an exact solution for a particular value of c to be determined. Comment on this solution, and plot the waveform, with ξ_1 chosen so that $\phi(\xi = 0) = \frac{1}{2}$, as before.

Project 2.11(b): Large-Time Limit for the Initial Value Problem

Travelling wave solutions often give clues to the general behaviour of solutions of a nonlinear wave equation. However, a more commonly encountered problem is when the population density is known at some initial time, and the subsequent evolution of the population is required. In this exercise we obtain solutions to (6). For numerical efficiency, renormalise the domain $[0, s(t)]$, to $[0, 1]$, by introducing a new spatial coordinate $y = x/s(t)$. Show that the evolution of $\rho(y, t)$ is given by

$$\frac{\partial \rho}{\partial t} = \frac{1}{2s^2}\frac{\partial^2(\rho^2)}{\partial y^2} + \frac{\dot{s}y}{s}\frac{\partial \rho}{\partial y} + \rho(1-\rho) \tag{11}$$

$$\rho_y(0,t) = 0, \qquad \rho(1,t) = 0, \qquad \rho_y(1,t) = -s\dot{s}. \tag{12}$$

The equation (11) is to be solved subject to the boundary conditions (12), and initial conditions

$$\rho(y,0) = \rho_0(y). \tag{13}$$

The final condition in (12) then determines the motion of the population front, with $s(0) = 1$.

Many methods exist for the numerical solution of parabolic equations. Here we consider a very simple finite-difference method, where spatial derivatives are expressed using centred differences and the solution is advanced in time using forward Euler. Writing $t_j = j(\Delta t)$, $y_n = n/N$, $(n = 0, 1, \ldots, N)$, and using the notation $\rho_{j,n} \equiv \rho(t_j, y_n)$, $s_j \equiv s(t_j)$ we discretise (11) in the form,

$$\begin{aligned}
\frac{\rho_{j+1,n} - \rho_{j,n}}{\Delta t} &= \frac{1}{2s_j^2}\frac{\rho_{j,n+1}^2 - 2\rho_{j,n}^2 + \rho_{j,n-1}^2}{(\Delta y)^2} + \frac{\dot{s}_j y_n}{s_j}\frac{\rho_{j,n+1} - \rho_{j,n-1}}{2(\Delta y)} + \rho_{j,n}(1-\rho_{j,n}), \\
&\qquad\qquad n = 1, 2, \ldots, N-1 \\
\rho_{j,0} &= \rho_{j,1}, \\
\rho_{j,N} &= 0, \\
\frac{s_{j+1} - s_j}{\Delta t} &= \dot{s}_j = -s_j^{-1}\,\frac{\rho_{j,N-2} - 4\rho_{j,N-1}}{2\Delta y},
\end{aligned}$$

where $\Delta y = 1/N$. The expression for $\dot{s}_j$ is obtained by using the final condition in (12) with a three-point backward difference expression for $\rho_y(y = 1)$.

There are several more sophisticated numerical methods of solving this system of equations, but the method described is very simple to implement and proves sufficient for the current purposes. The main drawback is that Δt must be chosen very small to ensure numerical stability.

Write your own program to solve (11) using the discretisation given above. Obtain solutions for initial distribution $\rho_0(x) = 0.3e^x(1-x)$. Start with $N = 100$ and $\Delta t = 0.0001$, but confirm that your code produces solutions that are independent of mesh-size. Plot the solution as a function of the original spatial variable x at $t = 0$, 2, 4, 6, 8 and 10. Also plot the velocity of the wave front, $\dot{s}(t)$, as a function of time. Compare the large-time wave profile with the travelling wave solutions obtained in project 2.11(a).

Project 2.11(c): Motion of the Wave Front

Using the same program written for project 2.11(b), we now investigate the early evolution of the wave front for different classes of initial population distribution.

Consider three different initial profiles:

(i) $\rho_0(x) = A_1 e^x (1-x)$;

(ii) $\rho_0(x) = A_2 e^{2x} (1-x)^2$;

(iii) $\rho_0(x) = A_3 e^{3x} (1-x)^3$;

where A_i are numerical constants characterising the total initial population.

Using the same mesh-size as above, obtain solutions for $0 < t \leqslant 0.5$, for initial distribution (i). Consider various values of the coefficient A_1, in the range $0.1 \leqslant A_1 \leqslant 0.9$. For the larger values of A_1 it may be necessary to reduce the time step-size. Do not include plots of $\rho(x,t)$ in your report, but concentrate on the motion of the wave front. Write down a relationship between the initial velocity of the wave front and the initial profile and show that this is in agreement with your numerical results.

Calculate solutions for initial distribution (ii) with $A_2 = 0.2$ for $0 < t \leqslant 1$. As before plot $\dot{s}(t)$ as a function of time. Repeat these calculations with the spatial mesh-size reduced to $\Delta y = 0.002$ and then $\Delta y = 0.001$, adjusting Δt as necessary. Describe the movement of the wave front. Repeat these calculations with $A_2 = 0.05$, for $0 < t \leqslant 0.75$.

Analysis suggests that for some classes of initial distributions, the population front is fixed until a certain waiting time t_w has elapsed, after which the population expands. For initial distributions which are locally quadratic in the vicinity of the wavefront, it can be shown that the waiting time is given by

$$t_w = \log(1 + \frac{1}{6g_2}) \tag{14}$$

where $\rho_0(x) \sim g_2 \, (1-x)^2$, as $x \to 1$. Are the numerical results you have obtained in broad agreement with this result? Discuss why such a phenomenon may occur in the evolution of a population.

Finally, calculate the motion of the population front for initial distribution (iii) with $A_3 = 0.2$. As with case (ii), reduce the mesh-size. Compare your results with the results of (ii).

References

A background to the biological models underlying these equations can be found in *Some exact solutions to a nonlinear diffusion problem in population genetics and combustion* by Newman (*J. Theoretical Biology* (1980) **85**, 325–334).

An in-depth analysis of equations of this form is presented in *The effects of variable diffusivity on the development of travelling waves in a class of reaction-diffusion equations* by King & Needham (*Phil. Trans. Roy. Soc. Lond.* A (1994) **348**, 229–260). This contains derivation of the results for waiting times, but reference to this paper is not necessary for the purposes of this project.

3 Fluid and Solid Mechanics

3.6 Particle Drift in a Periodic Flow Field (4 units)

This project builds on material in the Part IB Fluid Dynamics course.

A one-dimensional periodic flow in a fluid has velocity u in the x-direction only, given by

$$u = a\cos k(x - ct). \tag{1}$$

Without loss of generality, distance and time units may be chosen so that $k = 2\pi$ and $c = 1$, giving

$$u = a\cos 2\pi(x - t), \tag{2}$$

where u and a are now in the new units. Any material fluid element subject to this motion will have trajectory $X(t)$ satisfying

$$\frac{dX}{dt} = a\cos 2\pi(X(t) - t). \tag{3}$$

Question 1 Solve (3) numerically for a representative set of values of a, taking $X(0) = 0$. Plot the solutions against time. Justify the numerical accuracy of your results (for example, by considering results produced with different step-sizes or tolerances). Describe your results qualitatively.

Question 2 Verify from your numerical results that for $|a|$ sufficiently small, there is a time-averaged mean 'drift' velocity of $\frac{1}{2}a^2$. Include details of your method.

Question 3 Give a *physical* interpretation of the interaction between the flow and the material element. Do not confine your answer only to small $|a|$. [*Hint: You may wish to consider a graph of* $\dot{X}$ *against* X.]

Question 4 Analyse mathematically the above system, using any approach you see fit.

3 Fluid and Solid Mechanics

3.8 Wind-Forced Ocean Currents (10 units)

This project may well be attempted by someone who has attended the two Fluid Dynamics courses in Part IB and Part II.

1 Theory

A simple depth-independent model of the wind-forced ocean circulation is described by the governing equation for the streamfunction $\psi(x, y, t)$,

$$\zeta_t + J(\psi, \zeta) + v = -\epsilon\zeta + R\tau \tag{1}$$

in $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$ with $\psi = 0$ on the boundaries. x and y are Cartesian coordinates representing eastward and northward directions respectively. The vorticity ζ is related to ψ through the Poisson equation

$$\nabla^2\psi = \zeta, \tag{2}$$

and the x and y components of the velocity, respectively u and v, may be written in terms of ψ as

$$u = -\frac{\partial\psi}{\partial y}, \quad v = \frac{\partial\psi}{\partial x} \tag{3a, b}$$

$J(\psi, \zeta)$ is the Jacobian with respect to x and y and is an alternative way of writing the advective derivative term $\mathbf{u}.\nabla\zeta$. The $-\epsilon\zeta$ term on the right-hand side of (1) is a simple representation of the effect of bottom friction on the flow. The constant ϵ is a nondimensional frictional damping rate. The term $R\tau(x, y)$, representing the wind forcing, is equal to the curl of the wind stress. It is convenient to take τ to be a fixed function of x and y and, when investigating the behaviour of the model, to vary the strength of the forcing by varying the constant R.

The steady state form of (1) may be written in the form

$$\mathbf{u}.\nabla(\zeta + y) = R\tau - \epsilon\zeta, \tag{4}$$

implying that in the absence of wind-forcing and bottom friction the quantity $\zeta_a = \zeta + y$ would be conserved following the fluid motion. ζ_a is known as the 'absolute vorticity' and is the vertical (i.e. perpendicular to the Earth's surface) component of the vorticity measured with respect to an inertial frame (i.e. including the Earth's rotation as well as the motion of fluid relative to the Earth). The y contribution to the absolute vorticity is a simple representation of the variation of the vertical component of the rotation vector with latitude.

> **Question 1** Use incompressibility of $\mathbf{u}$ to rewrite the left hand side of (4). By integrating (4) over a region enclosed by a streamline and, using the divergence theorem on the left-hand side, deduce that if τ is one-signed then no steady state is possible if $\epsilon = 0$, i.e. friction is essential in the steady-state balance.

In this project you will be concerned with steady-state solutions to (1), and their variation as ϵ and R are changed. However, a convenient way to find the steady-state solution is to integrate (1) in time, say from initial conditions in which $\psi = 0$ for all x and y, until the steady state is achieved.

2 Numerical solution of (1)

Define a rectangular grid covering the domain, with points

$$(x_i, y_j) = \left(\frac{i}{N_x}, \frac{j}{N_y}\right) \qquad 0 \leqslant i \leqslant N_x, 0 \leqslant j \leqslant N_y.$$

The grid spacings are $\delta_x = 1/N_x$ and $\delta_y = 1/N_y$ in the x and y directions respectively. The variables ζ and ψ are defined at each point on the grid and it is helpful to use the notation, $\psi^t_{i,j} = \psi(x_i, y_j, t)$, $\zeta^t_{i,j} = \zeta(x_i, y_j, t)$. (The superscripts denote the time at which a particular quantity is to be evaluated.)

In order to integrate (1) in time it is sensible to use the $\zeta^t_{i,j}$ as the working independent variable and derive all the other quantities by solving (2) and then using finite-difference approximations for spatial derivatives in (1). You are recommended to use the expressions

$$v^t_{i,j} \;=\; \frac{\psi^t_{i+1,j} - \psi^t_{i-1,j}}{2\delta_x} \tag{5a}$$

for v and

$$\begin{aligned} J^t_{i,j} = [&(\psi^t_{i+1,j+1} - \psi^t_{i-1,j+1})\zeta^t_{i,j+1} - (\psi^t_{i+1,j-1} - \psi^t_{i-1,j-1})\zeta^t_{i,j-1} \\ &- (\psi^t_{i+1,j+1} - \psi^t_{i+1,j-1})\zeta^t_{i+1,j} + (\psi^t_{i-1,j+1} - \psi^t_{i-1,j-1})\zeta^t_{i-1,j}]/4\delta_x\delta_y, \end{aligned} \tag{5b}$$

for the Jacobian, both evaluated at the point (x_i, y_j) and time t.

It is recommended that to integrate (1) in time you use the leapfrog scheme in the form

$$\zeta^{t+\Delta t}_{i,j} - \zeta^{t-\Delta t}_{i,j} + 2J^t_{i,j}\Delta t + 2v^t_{i,j}\Delta t = -\epsilon\left(\zeta^{t+\Delta t}_{i,j} + \zeta^{t-\Delta t}_{i,j}\right)\Delta t + 2R\tau_{i,j}\Delta t, \tag{6a}$$

for $1 \leqslant i \leqslant N_x - 1$, $1 \leqslant j \leqslant N_y - 1$. Note that the boundary condition on ψ means that evaluating $J_{i,j}$ via (5b) at points immediately adjacent to the boundary does not require knowledge of ζ on the boundary itself. There is no need to impose or determine ζ on the boundary at any stage. Note also the way in which the damping term $-\epsilon\zeta$ is represented. The leapfrog scheme exhibits a slowly growing numerical instability, but the effects of this may be removed by taking an occasional Euler step, of the form

$$\zeta^{t+\Delta t}_{i,j} - \zeta^t_{i,j} + J^t_{i,j}\Delta t + v^t_{i,j}\Delta t = -\tfrac{1}{2}\epsilon\left(\zeta^{t+\Delta t}_{i,j} + \zeta^t_{i,j}\right)\Delta t + R\tau_{i,j}\Delta t. \tag{6b}$$

Question 2 Write a program to integrate the above. Take $\tau = -\sin\pi x \sin\pi y$. You may use a library routine for the solution of Poisson's equation (see Appendix below). Try using a grid size $N_x = N_y = 16$. Note that numerical accuracy of the time integration is not particularly important here because it is only the final steady state that is of interest. So you should experiment a little to find the largest possible time step Δt for which the integration remains stable and approaches a steady state. Note that there does seem to be an advantage in using (6a), plus occasional (6b), rather than just using (6b) at all time steps, since the integration remains stable for larger Δt and it therefore takes less computational time to reach the steady state.

Concentrate first on the case where R is very small. Integrate to a steady state for $\epsilon = 0.2$ and $\epsilon = 0.05$ and plot contour maps of the stream function and vorticity fields. Describe your results in qualitative terms.

In this regime you may assume that ζ and ψ scale with R and thus you might find it helpful to redefine ζ and ψ as $\hat{\zeta} = \zeta/R$ and $\hat{\psi} = \psi/R$. Then in the limit $R \to 0$ the

nonlinear term involving the Jacobian may be neglected, and the steady state form of (1) can be approximated as

$$\hat{v} = -\epsilon\hat{\zeta} + \tau\,. \tag{7}$$

Can you explain the differences between the two cases? You should see that the solution for $\epsilon = 0.05$ is highly asymmetric in the x direction with a strong narrow flow close to the $x = 0$ boundary and a broad weaker flow in the interior. Which term in the equation (1) leads to this asymmetry? For the case $\epsilon = 0.05$, indicate which terms in the equation play a dominant role in the balance in different parts of the flow. (Remember the conclusion from the result derived in Q1.) Provide an arguemnt for why a strong current near the $x = 1$ boundary cannot exist. (Consider which terms of (7) would be dominant in this region, and approximate the situation by a simple differential equation.) In this linear (i.e. small R) case estimate the maximum value for ψ in the small-ϵ limit. (Consider where a boundary layer may lie, and which boundary conditions you can discard.) Repeat your integrations at higher resolution with $N_x = N_y = 32$. What are your conclusions?

Question 3 Now, for $\epsilon = 0.05$ investigate the steady-state behaviour as R increases through the range 5×10^{-4} to 10^{-1}. (You will find that the timestep Δt must be reduced as R increases, firstly to suppress numerical instability and secondly to allow a steady state to be achieved. For $R = 0.1$, $dt \approx 0.001$ is required. For such a small time step you will need to run your code for a while in order for a steady state to be achieved). Continue to use $N_x = N_y = 32$. Describe how the pattern of flow changes as R is increased. You may find it useful to look at contour plots of ψ and ζ and also of the quantity $y+\zeta$. Show sufficient plots in your report to illustrate your main points. By considering plots of ψ and ζ and $y + \zeta$, for large R determine what the dominant balance in the equations is. Plot a graph of the maximum value of the streamfunction in the domain, $\psi_{\max}$ against R, and try to explain its form. How would you expect $\psi_{\max}$ to depend on the frictional damping rate ϵ in the large-R and small-R limits?

3 Reference

This topic is discussed in some detail in Chapter 5 of the book by Pedlosky (Pedlosky, J., 1979: *Geophysical Fluid Dynamics.* New York, Springer-Verlag, 624pp.; reference copies in Betty and Gordon Moore Library) but it is certainly not necessary to understand all of this Chapter in detail in order to complete this project.

Appendix

A solver of the Poisson equation in (2) for ψ given $\zeta^t_{i,j}$, is provided on the CATAM website: II-3-8-2015-PoissonSolver.tar.gz, located among the data files.

[psi]=Poisson(m,endpt,N,f)

The solver assumes a square grid and that ψ is 0 on all 4 boundaries.

m:	Number of grid points in the interior in one of the dimensions, i.e. if $N_x = 32$ $m = 30$.
endpt:	Domain size. $0 \leqslant x, y \leqslant$endpt.
N:	Type of integration scheme. For 5 point N=5, 9 point N=9, modified 9 point N=10. For more information see reference below.
f:	Function to be integrated, in this case ζ. f should take the form,

$$f_{ij} = f(x_i, y_j), \qquad (x_i, y_j) = \text{endpt} \times \left(\frac{i}{N_x}, \frac{j}{N_y} \right), \qquad 1 \leqslant i \leqslant N_x - 1,\ 1 \leqslant j \leqslant N_y - 1.$$

psi: The solution to the Poisson equation. psi will be given as,

$$\psi_{ij} = \psi(x_i, y_j), \qquad (x_i, y_j) = \text{endpt} \times \left(\frac{i-1}{N_x}, \frac{j-1}{N_y} \right), \qquad 1 \leqslant i \leqslant N_x + 1,\ 1 \leqslant j \leqslant N_y + 1.$$

For more information on Poisson.m go to `https://cs.nyu.edu/~harper/poisson.htm`. Note that while the mathematics involved in the version described at the link are the same, the inputs of the Poisson solver provided have been modified slightly.

3 Fluid and Solid Mechanics

3.10 Smoke Rings (8 units)

This project discusses a simple model of the motion of smoke rings. Knowledge of Part IB Fluid Dynamics is required and knowledge of the Part II course Classical Dynamics will help with Question 2. The article by Acheson [1] and the book by Saffman [4] may be found helpful.

A smoke ring is a vortex tube wrapped around into a closed circle (a *vortex ring*), which propagates normal to the plane of the circle under its self-induced velocity field. The politically incorrect method of generating them involves the inhalation of noxious substances; a more socially acceptable method involves a volcano [5]. Neglecting various effects, we will throughout this project crudely model a three-dimensional axisymmetric vortex ring of diameter a and strength κ by a pair of point vortices in two-dimensional fluid of strengths κ and $-\kappa$, a distance a apart.

1 2D vortex dynamics: Theory

Question 1 Show that the equations of motion of a set of point vortices of strengths κ_i at positions $(x_i(t), y_i(t))$, in a two-dimensional inviscid fluid which is otherwise at rest, are

$$\begin{aligned} \frac{\mathrm{d}x_i}{\mathrm{d}t} &= -\frac{1}{2\pi}\sum_{j\neq i}\frac{\kappa_j(y_i-y_j)}{r_{ij}^2} \\ \frac{\mathrm{d}y_i}{\mathrm{d}t} &= \frac{1}{2\pi}\sum_{j\neq i}\frac{\kappa_j(x_i-x_j)}{r_{ij}^2}, \end{aligned} \tag{1}$$

where

$$r_{ij} = ((x_i-x_j)^2+(y_i-y_j)^2)^{1/2}. \tag{2}$$

Question 2 Show carefully that the equations of motion can be written in the form

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \kappa_i^{-1}\frac{\partial H}{\partial y_i}, \qquad \frac{\mathrm{d}y_i}{\mathrm{d}t} = -\kappa_i^{-1}\frac{\partial H}{\partial x_i} \qquad \text{(no summation)} \tag{3}$$

where

$$H = -\frac{1}{4\pi}\sum_{\substack{i,j\\ i\neq j}}\kappa_i\kappa_j\log r_{ij}. \tag{4}$$

H is invariant under space translations and rotations, which implies the existence of three scalar conserved quantities. What are they?

Hint: Suppose H has a continuous family of symmetries, i.e.

$$H(X(x,y,\delta),Y(x,y,\delta)) = H(x,y)\,,$$

such that $X(x,y,0)=x$ and $Y(x,y,0)=y$. Observe that

$$0 = \sum_i\left(\frac{\partial H}{\partial x_i}\frac{\partial X_i}{\partial \delta}+\frac{\partial H}{\partial y_i}\frac{\partial Y_i}{\partial \delta}\right)_{\delta=0}.$$

H is also invariant under time translations — what conserved quantity does this give?

Programming Task: Write a program to integrate the equations of motion (1). You should use an adaptive stepsize ODE integrator. You will find it useful to write your code to handle arbitrarily many vortices.

2 Simulations of smoke rings

Question 3 Use your code to investigate the motion of a single "axisymmetric vortex ring" under this model. This problem can also be solved analytically; use the analytic solution to test your code. Also demonstrate that your code preserves the constants of the motion.

Question 4 What happens when two smoke rings are fired towards each other on the same axis? Describe the resulting motion, giving clear physical explanations for the behaviour. You should start by considering two rings with equal strengths and widths, but should also explain what happens in the general case.

Question 5 What happens when two smoke rings are fired in the same direction on the same axis? Describe the resulting motion, giving clear physical explanations for the behaviour. You should start by considering two rings with equal strengths and widths, but should also explain what happens in the general case. You should not need to integrate to large times, but you should (where relevant) show several cycles of the motion.

Question 6 We have made a number of modelling assumptions in reducing the full three-dimensional problem to this simple two-dimensional version. How good are they? Would the behaviour that you have observed in question 5 occur in a real physical system? Are the physical explanations that you gave in questions 4 and 5 for two dimensions relevant in the real geometry? What other possible effects have we neglected?

3 Symmetries and instabilities

Question 7 Repeat a typical one of the simulations you did for question 5, but now integrate to large times and show your output. What happens? Is the resulting behaviour physically plausible for a pair of 3D vortex rings? What happens if you tighten the error tolerances of your ODE solver?

Programming Task: Produce a program which can only model coaxial smoke rings, but which explicitly enforces the symmetry about the axis. In other words, use the mirror symmetry of the model system to reduce the number of ODEs that you have to solve.

Note that this is not the right way to handle symmetries and conserved quantities in the numerical solution of ODEs. See Iserles et. al. *[2] for more information. For this project, however, this method will suffice.*

Question 8 Use this new program to repeat the simulation you did for question 7. Show your output and comment.

4 More smoke rings

Question 9 Consider three coaxial smoke rings, fired in the same direction. Use your new program to investigate the resulting motion. Give a survey of the possible kinds of behaviour, including a selection of your plots (4 should suffice).

Note that the parameter space you have to search is rather large. You should think of ways to reduce it.

References

[1] Acheson, D. J. 2000, Instability of vortex leapfrogging, *Eur. J. Phys.*, **21**, 269–273.

Follow links from `http://www.iop.org/Journals/EJP/` for downloadable version.*

[2] Iserles, A., Munthe-Kaas, H. Z., Nørsett, S. P. and Zanna, A. 2000, Lie-group methods, *Acta Numerica*, **9**, 215–365.

[3] Pullin, D. I. and Saffman, P. G. 1991, Long-time symplectic integration: the example of four-vortex motion, *Proc. Roy. Soc. Lond. A*, **432**, 481–494.

[4] Saffman, P. G. 1992, *Vortex Dynamics*, CUP.

[5] `http://www.swisseduc.ch/stromboli/etna/etna00/etna0002photovideo-en.html`

* At the time of writing http://iopscience.iop.org/0143-0807/21/3/310 should take you there directly.

4 Dynamics

4.5 Euler's Equations (8 units)

This project is self-contained, though material from the Part II(C) course Classical Dynamics is relevant.

1 Introduction

The angular velocity with respect to principal axes of inertia in a rigid body is taken to be

$$\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3). \tag{1}$$

If the principal moments of inertia are A, B, C with respect to these axes, then the angular momentum is

$$\mathbf{h} = (A\omega_1, B\omega_2, C\omega_3). \tag{2}$$

These axes are fixed in the body and have angular velocity $\boldsymbol{\omega}$ with respect to an inertial frame instantaneously coincident with the principal axes. The rate of change of angular momentum with respect to such an inertial frame is

$$\frac{d\mathbf{h}}{dt} + \boldsymbol{\omega} \wedge \mathbf{h}. \tag{3}$$

In the case when there is no net moment of external forces acting on the body, the law "rate of change of angular momentum = moment of external forces" gives:

$$\frac{d\mathbf{h}}{dt} + \boldsymbol{\omega} \wedge \mathbf{h} \ = \mathbf{0}. \tag{4}$$

Expanding this equation into components gives:

$$\left.\begin{aligned} A\tfrac{d\omega_1}{dt} + (C-B)\omega_2\omega_3 = 0 \\ B\tfrac{d\omega_2}{dt} + (A-C)\omega_3\omega_1 = 0 \\ C\tfrac{d\omega_3}{dt} + (B-A)\omega_1\omega_2 = 0 \end{aligned}\right\} \tag{5}$$

It can be shown analytically that these equations have two first integrals, which say that the energy and the magnitude of the angular momentum remain constant, as follows:

$$\tfrac{1}{2}A\omega_1^2 + \tfrac{1}{2}B\omega_2^2 + \tfrac{1}{2}C\omega_3^2 = E \tag{6}$$

$$A^2\omega_1^2 + B^2\omega_2^2 + C^2\omega_3^2 = H^2 \tag{7}$$

Since the moment of external forces is zero, we also know that the angular momentum vector $\mathbf{h}$ is constant when measured in an inertial frame.

2 Project work

2.1 Program requirements

Write a program to solve Euler's equations (5) numerically and plot the results. Output from your program should include:

1. Graphs of $\omega_1(t)$, $\omega_2(t)$ and $\omega_3(t)$ against t;
2. 3-D phase space plots of ω_1, ω_2 and ω_3. Choose the OX axis for ω_1, etc.

Equations (6) and (7) can be used to check the accuracy of your numerical results by calculating and displaying the values of these expressions at the beginning and end of runs.

The objective of the project is to investigate and classify all possible types of motion. The following questions provide some guidelines for the investigation.

Question 1 Since A, B, C, $\omega_1(0)$, $\omega_2(0)$ and $\omega_3(0)$ may all in principle take arbitrary values, the parameter space to be explored may seem very large. If A, B and C take distinct values, explain how the results from taking

$$A > B > C \tag{8}$$

can be generalised. Briefly explain what happens if any two (or all three) of A, B, C are equal. For given values of A/B and C/B, explain why we may take $B = 1$ without loss of generality.

Show further that choosing $E = 1$ is equivalent to re-scaling the time variable t, and give the scaling factor.

2.2 Results requirements

From this point on, take $B = 1$ and $E = 1$ and build these values into your program. Your program should allow you to set and change the values of A/B and C/B. Assume that $A/B > 1$ and $C/B < 1$, and work with $A = 1.4$, $B = 1$ and $C = 0.7$ unless other values are suggested. You may find it convenient to accept arbitrary input for $\omega_1(0)$, $\omega_2(0)$ and $\omega_3(0)$ and then scale the input values so that $E = 1$.

Question 2 Use your program to demonstrate that solutions are possible in which the vector $\boldsymbol{\omega}(t)$ rotates around the OX axis with small amplitude deviation from $(\sqrt{2/A}, 0, 0)$ (i.e., $(1, 0, 0)$ before scaling), and that similar stable solutions exist near the OZ axis. Include copies of your results.

Question 3 Approximate Euler's Equations (5) by linearising them for the cases where (i) $\omega_1 \approx \sqrt{2/A}$ and ω_2, ω_3 are small, and (ii) $\omega_3 \approx \sqrt{2/C}$ and ω_1, ω_2 are small. Describe the analytic solutions. Are your solutions consistent with the results obtained for question 2 above? Give analytic expressions for the period of the motion in each case and compare with your results. (You need not compute the periods; an estimate by eye from your graphs of ω vs t is sufficient.)

Question 4 Are your numerical results consistent with equations (6) and (7)? To what extent are further checks on numerical accuracy needed?

Question 5 What solutions do you obtain if starting conditions are chosen so that $\boldsymbol{\omega}(0)$ lies very close to the OY axis? Describe the motion physically.

Question 6 Make a plausible case based on your computed results that there exists a solution $\boldsymbol{\omega}(t)$ which begins away from the OY axis but which tends towards the steady (but unstable) solution parallel to the OY axis as $t \to \infty$. What happens if you attempt to simulate such a solution numerically? What value must H take for such a solution?

Question 7 Using the following scheme, classify all the possible qualitative types of motion of the system assuming A, B and C take distinct values. Take the solution discussed in question 6 as a type of its own, and use it to separate the remaining solutions into two types. Describe the range of behaviour observable for each type. Explain clearly how the solution discussed in question 6 divides the solution space into regions (can you find the equations of the boundaries of these regions?) and how solutions behave physically as the boundaries are approached.

Try different values of A/B and C/B. How does the choice of these parameters affect your results?

Question 8 The rigid body is now subjected to slow friction via a retarding couple $-k\boldsymbol{\omega}$, where k is a very small parameter. How does this affect equations (5), (6) and (7)? Alter your program to incorporate the couple and investigate a few types of solutions for the original values of A, B and C. You may find it useful to consider 3D phase plots of a *suitably normalised* $\boldsymbol{\omega}(t)$, as well as your normal plots.

Question 9 Is your classification in question 7 still of any use or has it become irrelevant? Is there still a division of the solution space into regions?

5 Quantum Mechanics

5.2 S-Wave Scattering (7 units)

This project relies on a knowledge of material covered in the course Applications of Quantum Mechanics.

1 Theory

When an incoming beam of particles is scattered by a spherically symmetric potential, the time-independent Schrodinger equation has a general solution of the form

$$u(r,\theta) = \sum_{l=0}^{\infty} \frac{\chi_l(r)}{r} P_l(\cos\theta) \tag{1}$$

where u is the wave-function, P_l is the Legendre Polynomial of order l, and χ_l satisfies

$$\frac{d^2\chi_l}{dr^2} + \left[k^2 - U(r) - \frac{l(l+1)}{r^2}\right]\chi_l = 0 \tag{2}$$

with boundary condition $\chi_l(0) = 0$. The energy E of the particles is proportional to k^2.

If $U(r)$ is zero for $r > a$, then asymptotically $\chi_l(r) \sim A_l \sin(kr - \frac{1}{2}l\pi + \delta_l)$, where δ_l is the phase shift of the l^{th} partial wave. The total scattering cross-section σ is given by

$$\sigma = \frac{4\pi}{k^2} \sum_{l=0}^{\infty} (2l+1) \sin^2 \delta_l. \tag{3}$$

The solution (1) is a sum of partial waves. The S-wave corresponds to $l = 0$, and the contribution of the S-wave to the cross-section is thus $(4\pi/k^2)\sin^2\delta_0$.

Reference: L.I. Schiff, *Quantum Mechanics*, section 19.

2 Computation

Write a program to investigate numerically the relation between the phase shift δ_0 and k, for the potential

$$U(r) = \begin{cases} -U_0(1 - r^{2.8}) & r \leqslant 1 \\ 0 & r > 1 \end{cases} \tag{4}$$

Question 1 This can be done numerically by solving (2) for $l = 0$, taking $\chi_0(0) = 0$ and $\chi_0'(0) =$ an arbitrary value. The arbitrary value will affect normalisation, not phase. Explain your choice of numerical method and give quantitative estimates of the accuracy of any solutions you obtain. Plot graphs of your solutions and comment on them; use values of k in the range 0 to 5 and U_0 in the range 1 to 10 (see below).

Question 2 Modify your program to calculate δ_0, using the formula

$$\delta_0 = \tan^{-1} \frac{\chi_0(r_2)\sin kr_1 - \chi_0(r_1)\sin kr_2}{\chi_0(r_1)\cos kr_2 - \chi_0(r_2)\cos kr_1} \tag{6}$$

where r_1 and r_2 are two r-values in $r > 1$. Explain your choice of r_1 and r_2 values. [Note: the multi-valued nature of (6) must be resolved by the requirement $\delta_0 \to 0$ as $k \to \infty$ and that δ_0 be a continuous function of k.]

Your program should plot graphs of phase shift and cross-section against k in the range 0 to 5. Take care to resolve the behaviour near $k = 0$.

Use 4 or 5 values of U_0 in the range 1 to 10, and include printouts of each graph. Give an interpretation of your results.

5 Quantum Mechanics

5.4 Monte Carlo Simulations in Particle Physics (8 units)

This project only requires knowledge of the addition and conservation of 4-momenta from the Part IA course Dynamics and Relativity. Although not necessary to do the project, the Part II course Applications of Quantum Mechanics introduces the cross section.

1 Introduction

Units will be used throughout in which energy is measured in GeV (1 GeV is the energy an electron gains when accelerated through a potential difference of 10^9 V), and $c = 1$.

In quantum field theory all calculations are performed using perturbation theory in the small coupling for the interaction, for example in processes involving the electromagnetic force the coupling is the fine structure constant $\alpha = 1/137$.* In these theories the forces are mediated by the exchange of massless vector bosons, e.g. the photon for the electromagnetic force and the gluon for the strong force. The increasing accuracy of modern particle physics experiments means that higher orders in this perturbation series must be calculated in order to compare the predictions with experimental results.

However, due to the complexity of the observables which are measured experimentally the calculations have to be performed numerically. This project aims to illustrate the techniques used in these calculations by considering a simple example.

2 Monte Carlo Integration

The Monte Carlo procedure is based on the following result. For a simple one-dimensional integral,

$$\int_a^b f(x)\,dx = \big\langle f(x)(b-a)\big\rangle. \tag{1}$$

The average, $\big\langle f(x)(b-a)\big\rangle$, can be approximated by calculating $f(x)(b-a)$ at N randomly chosen points in the interval (a, b), i.e.

$$\big\langle f(x)(b-a)\big\rangle \simeq \frac{1}{N}\sum_{i=1}^{N}(b-a)f(x_i) \equiv \overline{f}_N, \tag{2}$$

giving an estimate, $\overline{f}_N$, of the average. We will refer to $f(x)(b-a)$ as the *weight* and therefore the integral is given by the average of the weights. This method is particularly useful as we can also calculate an error on the estimate by computing the standard deviation and applying the central limit theorem:

$$\big\langle f(x)(b-a)\big\rangle = \overline{f}_N \pm \frac{\sigma_N}{\sqrt{N}}, \tag{3}$$

where $\sigma_N^2 = \overline{f_N^2} - \overline{f}_N^2$ and $\overline{f_N^2} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} f^2(x_i)(b-a)^2$.

*Here we ignore the fact that the fine structure actually depends on the energy scale of the process. This is discussed in Part III of the Mathematical Tripos.

The convergence of this method for numerically evaluating the integral goes as $1/\sqrt{N}$ with the number of function evaluations, N. This is slower than other commonly used techniques for numerical integration, e.g. the trapezium rule converges as $1/N^2$ and Simpson's rule as $1/N^4$. While the convergence of these other methods becomes far slower for higher dimensional integrals, e.g. the trapezium rule converges as $1/N^{2/d}$ and Simpson's rule as $1/N^{4/d}$ where d is the dimension of the integral, the Monte Carlo method will always converge as $1/\sqrt{N}$.

Hence for the performance of high-dimensional integrals the Monte Carlo technique is more efficient. This is particularly important in particle physics where we need to perform high-dimensional phase-space integrals. The Monte Carlo procedure is also well suited to integrating over complex regions, which are difficult with other methods and which often occur in particle physics, e.g. due to experimental cuts. Another advantage of this method is that we can evaluate a number of different quantities, e.g. differential distributions, at the same time whereas with other methods each distribution would have to be calculated separately.

The convergence of the Monte Carlo technique can be improved by reducing the standard deviation, σ_N. In principle if the integral can be performed analytically a Jacobian transform can be used to reduce the standard deviation to zero. In practice we can use a simple function which approximates the shape of the function we are integrating to improve the convergence of the integral, thus considerably reducing the time required for the numerical computation of the integral.

3 e^+e^- annihilation

In electron–positron scattering experiments, these particles annihilate and give a virtual photon which can then produce any charged particle pair. In this project we will consider the production of quarks, the particles which constitute the proton, the neutron, and other hadrons.

The quantity in which we shall be interested is the *cross section* σ, a dimensionful[†] measure of the likelihood of some process occuring. We will first concentrate on the *total* cross section σ_{tot} for a particular decay, but in later sections of the project we will consider cross sections which satisfy kinematic constraints called *cuts*.

We start by considering the production of a quark and an antiquark. This is the reaction

$$\mathrm{e^+e^-} \to \gamma \to \mathrm{q\bar{q}}\,.$$

(When the distinction between matter and antimatter is not important, it is usual to refer to quarks *and* antiquarks as quarks. We will sometimes follow this convention below.) The leading order process occurs via the annihilation of the electron and positron to briefly create a photon which in turn (here) creates a quark–antiquark pair. However, as the strong force which binds the quarks into the proton is much stronger than the electromagnetic force, corrections from higher order processes – in a perturbative expansion about small values of the strong coupling constant α_s – are important.

These corrections consist of two parts. One is a correction to the leading order process which has two particles in the final state, and the second involves the emission of a gluon, with three particles in the final state. Therefore, we consider cross section here (at this order in α_s) to be the sum

$$\sigma = \sigma^{\mathrm{q\bar{q}}} + \sigma^{\mathrm{q\bar{q}g}},$$

where $\sigma^{\mathrm{q\bar{q}}}$ is the contribution from two particles in the final state and $\sigma^{\mathrm{q\bar{q}g}}$ is the contribution from three particles in the final state.

[†]The cross section has units of area.

The cross section for the real gluon emission process is

$$\sigma^{\mathrm{q\bar{q}g}} = \frac{\alpha_s}{12\pi^3}\sigma_0 \int_0^{2\pi} d\alpha \int_{-1}^{1} d(\cos\beta) \int_0^{2\pi} d\gamma \int_0^1 dx_q \int_{1-x_q}^1 dx_{\bar{q}} \frac{x_q^2 + x_{\bar{q}}^2}{(1-x_q)(1-x_{\bar{q}})} F_3(p_q, p_{\bar{q}}, p_g), \quad (4)$$

with $x_q = 2E_q/E$ and $x_{\bar{q}} = 2E_{\bar{q}}/E$, where E_q, $E_{\bar{q}}$ are the energies of the quark and antiquark respectively, E_g is the energy of the gluon, E is the centre-of-mass energy of the collision and σ_0 is the cross section for the leading order process. The angles α, β and γ are the Euler angles specifying the orientation of the plane of the momenta. The function $F_3(p_q, p_{\bar{q}}, p_g)$ is equal to 1 for the total cross section. However, we include it here as a way to exclude contributions which fail to satisfy some kinematic constraint. We will make use of these cuts in Sections 5 and 6. Note that the total cross section for real emission $\sigma_{\mathrm{tot}}^{\mathrm{q\bar{q}g}}$ diverges as $x_q, x_{\bar{q}} \to 1$.

The incoming beams have equal and opposite momentum and can be considered to be massless. Hence their 4-momenta are

$$p_{\mathrm{e}^+} = (E/2, 0, 0, \quad E/2),$$
$$p_{\mathrm{e}^-} = (E/2, 0, 0, -E/2),$$

and the 4-momentum of the centre-of-mass system is given by

$$p_{\mathrm{cm}} = (E, 0, 0, 0).$$

Throughout the project you can also assume the masses of the quarks produced are negligible.

Question 1 Using the conservation of 4-momentum show that

$$E(E - 2E_{\bar{q}}) = (p_q + p_g)^2,$$

where p_q and p_g are the 4-momenta of the quark and gluon respectively, and hence find $(1 - x_{\bar{q}})$ in terms of x_q, $x_g = 2E_g/E$ and the angle θ_{qg} between the quark and the gluon. You should also obtain a similar expression for $(1 - x_q)$.

Using these results comment on the physical situations in which the cross section diverges.

These are called infrared singularities and occur because in certain situations we cannot tell the difference between a quark and a quark–gluon pair. These singularities are cancelled by opposite terms in the two particle piece of the cross section.

We can subtract the term

$$\sigma_{\mathrm{counter}} = \frac{\alpha_s}{12\pi^3}\sigma_0 \int_0^1 dx_q \int_{1-x_q}^1 dx_{\bar{q}} \int_0^{2\pi} d\alpha \int_{-1}^{1} d(\cos\beta) \int_0^{2\pi} d\gamma \left\{ \left[\frac{1}{1-x_{\bar{q}}} \left(\frac{2}{2-x_q-x_{\bar{q}}} - (1+x_q) \right) + \frac{1-x_q}{x_{\bar{q}}} \right] F_2(\tilde{p}_{qg}, \tilde{p}_{\bar{q}}) + \left[\frac{1}{1-x_q} \left(\frac{2}{2-x_q-x_{\bar{q}}} - (1+x_{\bar{q}}) \right) + \frac{1-x_{\bar{q}}}{x_q} \right] F_2(\tilde{p}_{\bar{q}g}, \tilde{p}_q) \right\}, \quad (5)$$

where

$$\tilde{p}_{\bar{q}} = \frac{1}{x_{\bar{q}}} p_{\bar{q}}, \qquad \tilde{p}_{qg} = p_{\mathrm{cm}} - \frac{1}{x_{\bar{q}}} p_{\bar{q}}, \qquad \tilde{p}_q = \frac{1}{x_q} p_q \qquad \text{and} \qquad \tilde{p}_{\bar{q}g} = p_{\mathrm{cm}} - \frac{1}{x_q} p_q.$$

Again, for the total cross-section, $F_2 = 1$, but will be used to implement the cuts described in Sections 5 and 6.

Question 2 Show that the integrand of the regularized total cross section

$$\sigma^{\mathrm{q\bar{q}g}} - \sigma_{\mathrm{counter}}$$

is finite in the limits $x_q \to 1$ and $x_{\bar{q}} \to 1$.

If we consider the term for two particles we get

$$\sigma^{\mathrm{q\bar{q}}} = \frac{\sigma_0}{4\pi} \int_0^{2\pi} d\phi \int_{-1}^{1} d(\cos\theta) \left(1 + \frac{4\alpha_s}{3\pi}\right) F_2(p_q, p_{\bar{q}}) - \sigma_{\mathrm{counter}}, \tag{6}$$

where here ϕ and θ give the direction of the quark momentum.

Question 3 We can now obtain the total cross section for $\mathrm{e^+e^-} \to$ hadrons because hadrons are always formed from quarks and gluons with unit probability. Hence

$$\sigma_{\mathrm{tot}}^{\mathrm{hadrons}} = \sigma_{\mathrm{tot}}^{\mathrm{q\bar{q}g}} + \sigma_{\mathrm{tot}}^{\mathrm{q\bar{q}}}\,.$$

Evaluate this cross section analytically, i.e. take $F_{2,3} = 1$.

Since the total cross section can be calculated analytically, it will useful in testing the computer code we write to calculation cross sections which cannot be solved analytically.

4 Numerical Work

We shall be evaluating integrals using the Monte Carlo technique.

Programming Task: We will perform the necessary integration in four pieces.

1. The first bit will be the piece which only has two particles in the final state. First you must randomly generate the $\cos\theta$ uniformly distributed on $[-1, 1]$ (θ can then be uniquely determined as it must lie between 0 and π), and ϕ uniformly distributed on $[0, 2\pi)$. You can then calculate the momenta of the particles, i.e.

$$\begin{aligned} p_q &= \tfrac{1}{2}E(1, \quad \sin\theta\cos\phi, \quad \sin\theta\sin\phi, \quad \cos\theta), \\ p_{\bar{q}} &= \tfrac{1}{2}E(1, -\sin\theta\cos\phi, -\sin\theta\sin\phi, -\cos\theta). \end{aligned}$$

   This configuration gives a weight

$$W_2 = \sigma_0 \left(1 + \frac{4\alpha_s}{3\pi}\right). \tag{7}$$

2. The next step is to randomly generate values of x_q between 0 and 1 and $x_{\bar{q}}$ between $1 - x_q$ and 1. In practice to avoid overflow problems it will be necessary to change the upper limits of both integrands to $1 - \epsilon$. A value of $\epsilon = 10^{-8}$ should be sufficient although you should provide evidence in your write-up that you have checked this does not affect the result. You can then construct the momenta of the particles in the x–z plane:

$$\begin{aligned} p_q &= \tfrac{1}{2}E\left(x_q, x_q, 0, 0\right), \\ p_{\bar{q}} &= \tfrac{1}{2}E\left(x_{\bar{q}}, -\frac{1}{2x_q}\left(x_q^2 + x_{\bar{q}}^2 - x_g^2\right), 0, \frac{1}{2x_q}\sqrt{4x_q^2x_{\bar{q}}^2 - (x_q^2 + x_{\bar{q}}^2 - x_g^2)^2}\right), \\ p_g &= \tfrac{1}{2}E\left(x_g, -\frac{1}{2x_q}\left(x_q^2 + x_g^2 - x_{\bar{q}}^2\right), 0, -\frac{1}{2x_q}\sqrt{4x_q^2x_{\bar{q}}^2 - (x_q^2 + x_{\bar{q}}^2 - x_g^2)^2}\right) \end{aligned}$$

where $x_q + x_{\bar{q}} + x_g = 2$. Check that these 4-momenta are massless and that energy and momentum are conserved. You now need to calculate the angles α, β and γ in the same way as before. The momenta should then be rotated by the Euler angles, i.e. the rotation matrix

$$\begin{pmatrix} \cos\gamma\cos\beta\cos\alpha - \sin\gamma\sin\alpha & \cos\gamma\cos\beta\sin\alpha + \sin\gamma\cos\alpha & -\cos\gamma\sin\beta \\ -\sin\gamma\cos\beta\cos\alpha - \cos\gamma\sin\alpha & -\sin\gamma\cos\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta \\ \sin\beta\cos\alpha & \sin\beta\sin\alpha & \cos\beta \end{pmatrix}$$

should be applied to the 3-momenta.

We are now in a position to calculate the remaining contributions to the cross section. First we will compute the contribution from the real emission. This is given by

$$W_3 = \frac{2\alpha_s}{3\pi}\sigma_0 x_q \frac{x_q^2 + x_{\bar{q}}^2}{(1-x_q)(1-x_{\bar{q}})} F_3(p_q, p_{\bar{q}}, p_g)\,. \tag{8}$$

We also need the contributions from the counter terms which should be treated in two parts.

3. The first

$$W^1_{\text{counter}} = \frac{2\alpha_s}{3\pi}\sigma_0 x_q \left[\frac{1}{1-x_{\bar{q}}}\left(\frac{2}{2-x_q-x_{\bar{q}}} - (1+x_q)\right) + \frac{1-x_q}{x_{\bar{q}}}\right] F_2(\tilde{p}_{qg}, \tilde{p}_{\bar{q}}) \tag{9}$$

which requires the 4-vectors $\tilde{p}_{qg}$ and $\tilde{p}_{\bar{q}}$.

4. The second

$$W^2_{\text{counter}} = \frac{2\alpha_s}{3\pi}\sigma_0 x_q \left[\frac{1}{1-x_q}\left(\frac{2}{2-x_q-x_{\bar{q}}} - (1+x_{\bar{q}})\right) + \frac{1-x_{\bar{q}}}{x_q}\right] F_2(\tilde{p}_{\bar{q}g}, \tilde{p}_q) \tag{10}$$

which requires the 4-vectors $\tilde{p}_{\bar{q}g}$ and $\tilde{p}_q$.

Question 4 You now have four contributions to the cross section. You should add the contributions from the real emission and counter term. This gives two contributions, W_2 and $W_3^s = W_3 - W^1_{\text{counter}} - W^2_{\text{counter}}$. You can now simply evaluate the weights many times recording the sum of the weights and their squares to calculate the integrals and the error. Find the total cross section $\sigma_{\text{tot}}^{\text{hadrons}}$ and the error, and compare this with your analytic result. You should adjust the number of weights you generate in order to achieve five decimal place accuracy on the cross section and justify your choice of how many weights to generate.

For all your calculations you should express the answers as a multiple of σ_0 and use an energy of 200 GeV and a strong coupling of $\alpha_s = 0.118$.

5 Thrust

Having calculated the total cross section and checked against the analytic result you can now use your code to evaluate more complicated quantities.

The thrust is given by

$$T = \max_{\mathbf{n}} \frac{\sum_i |\mathbf{p}_i \cdot \mathbf{n}|}{\sum_i |\mathbf{p}_i|}, \tag{11}$$

where the sum is over all the particles in the final state, $\mathbf{p}_i$ is the 3-momentum of the particle and $\mathbf{n}$ is a unit 3-vector. In this case calculating the thrust is relatively simple.

Question 5 Show for a pencil-like event, i.e. one where all the particles lie along one vector, that $T = 1$ and that for a spherical event, i.e. one where the momenta are uniformly distributed over a sphere, $T = \frac{1}{2}$. These are the limiting cases.

In our case the 2-body and counter terms will have thrust $T = 1$, and the 3-body terms $T = \max(x_q, x_{\bar{q}}, x_g)$.

Programming Task: Enhance your program to calculate the thrust for the different types of term.

We can now consider the effect of applying a cut on the thrust in our program. If we only want the cross section for $T > T_{\min}$ then we should calculate the thrust for the 2-body, 3-body and counter terms, and if the thrust is less than $T_{\min}$ set the weight for that piece for the calculation to zero (i.e. we are applying a Heaviside function inside the integrand).

Question 6 Evaluate the cross section for $T_{\min} = 0.65$, 0.8, 0.9 to an accuracy of three decimal places.

We can also find the differential cross section $d\sigma/dT$. All we need to do is to calculate the thrust and then plot a histogram of the weight vs. the thrust. To do this you should produce three arrays: one will store the sum of the weights in each bin; one will store the sum of the squares of the weights in each bin; and one will store the number of weights in each bin.

Programming Task: Modify your program to calculate the thrust and put the result into a bin, experimenting to find a reasonable bin size. If you have too few bins no detail can be seen but if you have too many the error in a given bin will be large.

Then find the error in each bin in the same way as before, using the number of weights in the bin. You should divide by the width of the bin and the total number of weights generated in order to normalize the results. Plot and comment on your results.

6 C-parameter

Another quantity is the C-parameter,

$$C = \frac{3}{2}\frac{\sum_{ij}\left[|\mathbf{p}_i||\mathbf{p}_j| - (\mathbf{p}_i \cdot \mathbf{p}_j)^2/|\mathbf{p}_i||\mathbf{p}_j|\right]}{(\sum_i |\mathbf{p}_i|)^2}, \tag{12}$$

where $\mathbf{p}_i$ is the three momentum of a particle and the sum is over all particles in the final state.

Question 7 Write some code to calculate the C-parameter for the four different terms in your program, i.e. the two-body piece, both counter terms and the three-body piece.

As with the thrust obtain the cross section for $C < C_{\max}$. Obtain the cross section for $C_{\max} = 0.45$, 0.3, 0.15 with 3 decimal place accuracy. You should also obtain the cross section $d\sigma/dC$ in the same way as for the thrust and plot the result.

Comment on these results and compare them with the thrust.

7 Mathematical Methods

7.3 Minimisation Methods (8 units)

There are no prerequisites for this project.

You should write your own minimisation method programs: it is not sufficient to use routines that are distributed as part of MATLAB, other software packages, or other programming languages. You can, however, exploit the matrix manipulation capabilities of MATLAB, other software packages, or other progamming languages. Note that this project is particularly concerned with clarity and conciseness of expression. Overly long write-ups in which the important points are buried within lots of irrelevant detail will be penalised. You should therefore aim to produce concise, relevant answers that address the issues, together with a carefully chosen selection of diagrams and graphs.

1 Introduction

There are many numerical methods for finding the least value of a function of N variables, $f(x_1, x_2, x_3, \ldots) = f(\mathbf{x})$, say, given that the first derivatives

$$g_i = \frac{\partial f}{\partial x_i}, \qquad i = 1, 2, \ldots N \ , \tag{1}$$

can be calculated. Most of the methods are iterative and each iteration reduces the value of $f(\mathbf{x})$ by searching along a descent direction in the space of the variables in the following way.

The iteration begins with a starting point $\mathbf{x}_0$, and at this point the gradient vector $\mathbf{g}$ is calculated. Then a search direction, $\mathbf{s}$, say, is chosen, that satisfies the condition $\mathbf{g} \cdot \mathbf{s} < 0$ (the dot denotes a scalar product). It follows that if we move from $\mathbf{x}_0$ in the direction of $\mathbf{s}$, then the value of $f(\mathbf{x})$ becomes smaller initially. In other words the function of one variable

$$\phi(\lambda) = f(\mathbf{x}_0 + \lambda \mathbf{s}) \tag{2}$$

satisfies the condition $\phi'(0) < 0$ which is equivalent to $\mathbf{g} \cdot \mathbf{s} < 0$. The next stage is to consider the function $\phi(\lambda)$, and choose a value of λ, λ^* say, that satisfies the inequality

$$f(\mathbf{x}_0 + \lambda^* \mathbf{s}) < f(\mathbf{x}_0) \ . \tag{3}$$

Usually λ^* will be chosen * to minimise $\phi(\lambda)$. The vector $\mathbf{x}_0$ is replaced by $\mathbf{x}_1 = \mathbf{x}_0 + \lambda^* \mathbf{s}$ and another iteration is begun.

The project is to investigate three well-known algorithms (*Steepest Descents*, the *Conjugate Gradient algorithm*, and the *DFP algorithm*) by applying them to two functions. Using x for x_1, y for x_2, etc., consider the "bedpan function"

$$x + y + \frac{x^2}{4} - y^2 + (y^2 - \frac{x}{2})^2 \ , \tag{4}$$

and the following function, which has similar properties to the Rosenbrock function,

$$(1 - x)^2 + 80(y - x^2)^2 \ . \tag{5}$$

*In your program for this project, you should allow for manual input of estimates for λ^*, based on plots of $\phi(\lambda)$. To speed up longer runs you may *if you wish* also use MATLAB routines or other automated search algorithms to minimise $\phi(\lambda)$, but this is not required. In either case, the values of λ^* used should appear in the hard copy of results.

In addition the following quadratic function of three variables will be used to demonstrate some properties of the DFP algorithm:

$$0.4x^2 + 0.2y^2 + z^2 + xz \ . \tag{6}$$

2 Steepest Descents

The Steepest Descents method simply uses the search direction $\mathbf{s} = -\mathbf{g}$. Write a program to implement the algorithm as described above. Use a simple x–y plot of $\phi(\lambda)$ to help you determine λ^* at each stage (you need never determine λ^* to more than 2 significant figures). At each stage after the first, arrange for your program to display the current value of $f(\mathbf{x})$ and the decrease achieved over the last step. Also arrange for a plot of the iteration points $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, etc., (a sequence of line segments will illustrate the methods well). The iteration point plot may be built up as the calculation proceeds, or you can store the data and produce it on command from your programme at a point of your choosing.

[N.B. A well-implemented fully automatic algorithm for general use will need to have checks for special cases and exceptions built into it. For example, if a point $\mathbf{x}_n$ is encountered for which $\mathbf{g} \approx \mathbf{0}$ then a stationary point has been found and the process should quit. Likewise, if the iteration points are not changing significantly a fully automatic algorithm ought to quit. You may find it helpful to include such features in your program. If you wish to proceed semi-automatically, with λ^* being decided by eye from the plot at each stage, there is no need to include the special checks in your code.]

Question 1 Obtain contour plots and/or surface plots of functions (4) and (5) (this should be fairly straightforward to do using MATLAB).

Work out analytically where they have minima and find their minimum values. Suitable axis intervals are $-1.5 \leqslant x, y \leqslant 1.5$.

Question 2 Using function (4) and starting from $(-1.0, -1.3)$, run the Steepest Descents method for 10 iterations. Produce a plot of the progress of the iteration. On the basis of your numerical results (i.e., imagine that you do not know the analytical answer), estimate the minimum value of the function at the point to which your iteration is converging, and estimate intervals in which the co-ordinates of the minimum lie. What general statement can you make about the precision with which the minimum value itself can be found, compared to the precision with which the minimum point is known? What property of the function being minimised gives rise to this effect?

Question 3 Using function (5) and starting from $(0.676, 0.443)$, run the Steepest Descents method for at least 9 iterations, and produce a plot of the progress of the iteration. To what point do you think the iteration will eventually converge? Comment on the rate of convergence. How sensitive is the iteration path to variations in the choice of λ^* at each stage? Comment on the circumstances that can make steepest descents inefficient.

3 Conjugate Gradients

The conjugate gradients algorithm uses steepest descents for its first step and then adjusts the search direction in an attempt to overcome the problems of steepest descents alone. Let $\mathbf{x}_0$,

$\mathbf{x}_1$ be two successive points where $\mathbf{x}_1$ has been obtained using steepest descents from $\mathbf{x}_0$, and let $\mathbf{g}_0$, $\mathbf{g}_1$ be the corresponding gradients (the initial search direction is $\mathbf{s}_0 = -\mathbf{g}_0$). Take the second search direction as

$$\mathbf{s}_1 = -\mathbf{g}_1 + \beta\ \mathbf{s}_0 = -\mathbf{g}_1 - \beta\ \mathbf{g}_0 \ \text{ where } \ \beta = \frac{\mathbf{g}_1 \cdot \mathbf{g}_1}{\mathbf{g}_0 \cdot \mathbf{g}_0}. \tag{7}$$

If $f(\mathbf{x})$ is a quadratic function of N variables then the choice of directions may be continued up to the N^{th} search direction to give the N conjugate directions

$$\mathbf{s}_k = -\mathbf{g}_k + \beta\ \mathbf{s}_{k-1} \ \text{ where } \ \beta = \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_{k-1} \cdot \mathbf{g}_{k-1}}\ .$$

In this case, if all the values of λ^* had been chosen to minimise the $\phi(\lambda)$ exactly at each stage, the algorithm would have converged. In practice of course $f(\mathbf{x})$ may not be quadratic and the values λ^* may not be chosen exactly, and in this case it is usual in practice to restart the method after N steps. When $N = 2$, as it is for functions (4) and (5), this implies that every other step is a steepest descent.

Write a program to implement the conjugate gradients algorithm, with the same features as used for the steepest descents method, but with the search direction determined as just described.

> **Question 4** For the function (4), repeat Q2 using the conjugate gradients algorithm, and compare results.

> **Question 5** For the function (5), repeat Q3 using the conjugate gradients algorithm, and compare results.

Does the conjugate gradients algorithm offer much of an improvement over steepest descents?

4 DFP Algorithm

The Taylor Series expansion of any smooth function $f(\mathbf{x})$ may be written

$$f(\mathbf{a} + \Delta\mathbf{x}) \cong f(\mathbf{a}) + \mathbf{g} \cdot (\Delta\mathbf{x}) + \tfrac{1}{2}(\Delta\mathbf{x})^T\mathbf{H}^{-1}(\Delta\mathbf{x}) + \cdots$$

where the gradient vector $\mathbf{g}$ is evaluated at $\mathbf{x} = \mathbf{a}$ and $\mathbf{H}^{-1} \equiv \mathbf{G}$ is the Hessian matrix, i.e., the matrix of second derivatives

$$G_{ij} = \frac{\partial^2 f}{\partial x_i\, \partial x_j}\ .$$

Finding a point where $\mathbf{g}$ vanishes is therefore similar to the Newton–Raphson method for a system of equations, and if $\mathbf{H}$ were known and $f(\mathbf{x})$ were a quadratic function, the point could be found in a single step. However the matrix $\mathbf{H}$ is not available initially unless second derivatives are calculated; this is not always easy and in any case can be time-consuming, especially for large N. Therefore we now study a very successful technique that extends the steepest descent method by forming a suitable $\mathbf{H}$-matrix as the calculation proceeds. It is known as the DFP algorithm and is one of the class of "variable metric methods". It can be shown that $\mathbf{H}^{-1}$ converges to the Hessian (you are not required to prove this).

The DFP algorithm works as follows. The search direction is taken as $\mathbf{s} = -\mathbf{H}\mathbf{g}$, where $\mathbf{H}$ is taken initially as the identity matrix. At each stage $\phi(\lambda)$ is minimised by choosing a value λ^* as before, but then $\mathbf{H}$ is modified by replacing it with

$$\mathbf{H}^* = \mathbf{H} - \frac{\mathbf{H}\mathbf{p}\mathbf{p}^T\mathbf{H}}{\mathbf{p}^T\mathbf{H}\mathbf{p}} + \frac{\mathbf{q}\mathbf{q}^T}{\mathbf{p}^T\mathbf{q}}, \tag{8}$$

where $\mathbf{p}$ and $\mathbf{q}$ are column vectors giving the changes in $\mathbf{g}$ and $\mathbf{x}$ respectively during the step, that is

$$\mathbf{p} = \mathbf{g}\left(\mathbf{x}_0 + \lambda^* \mathbf{s}\right) - \mathbf{g}\left(\mathbf{x}_0\right), \quad \mathbf{q} = \lambda^* \mathbf{s} \tag{9}$$

(Note: $\mathbf{H}^* \mathbf{p} = \mathbf{q}$ which is useful when checking your program.)

Write a program to implement the DFP algorithm, with the same features as used for the two preceding programs, but with the search direction determined as just described. Include provision to print out $\mathbf{H}$.

Question 6 A property of the DFP algorithm is that it calculates the least value of a quadratic function in at most N iterations for any initial choice of $\mathbf{x}_0$ if on each iteration the value of λ^* is calculated to minimise exactly the function $\phi(\lambda)$. Apply the DFP algorithm to (6) for three iterations from starting point $\mathbf{x}_0 = (1, 1, 1)$ using the sequence of values

$$\lambda^* = 0.3942,\ 2.5522,\ 4.2202.$$

There is no need to verify these values to this precision, but your program will already have facilities for checking that these values are appropriate. Investigate how sensitive the result obtained after three iterations is to small changes in these values. Verify that $\mathbf{H}$ does indeed tend to the inverse Hessian matrix. You may note that

$$\begin{pmatrix} 0.8 & 0 & 1 \\ 0 & 0.4 & 0 \\ 1 & 0 & 2 \end{pmatrix}^{-1} = \begin{pmatrix} 3.3333 & 0 & -1.6667 \\ 0 & 2.5 & 0 \\ -1.6667 & 0 & 1.3333 \end{pmatrix} \tag{10}$$

Question 7 For the function (4), repeat Q2 using the DFP algorithm. Examine $\mathbf{H}$ and compare with the true value.

Question 8 For the function (5), repeat Q3 using the DFP algorithm. Examine $\mathbf{H}$ and compare with the true value.

Question 9 Compare the performance of the three methods for these functions.

References

[1] Fletcher, R. and Powell, M.J.D. *Rapidly convergent descent method for minimisation*, Computer Journal, 7 (1963).

[2] McKeown, J.J., Meegan, D., and Sprevak, D. *An Introduction to Unconstrained Optimisation - A Computer Illustrated Text*, IOP Publishing (1990). Although references to the computer programs (designed for BBC micro) are best ignored, the text is still relevant.

7 Mathematical Methods

7.4 Airy Functions and Stokes' Phenomenon (9 units)

This project uses ideas from the Further Complex Methods course. It also covers some material which is lectured as part of the Asymptotic Methods course, but students not taking this course are at no disadvantage.

1 Introduction

The Airy functions $\mathrm{Ai}(z)$ and $\mathrm{Bi}(z)$, where z is a complex variable, are two linearly independent solutions of the differential equation

$$\frac{\mathrm{d}^2}{\mathrm{d}z^2}\, y(z) = zy(z) \tag{1}$$

satisfying

$$\mathrm{Ai}(0) = \alpha, \qquad \mathrm{Ai}'(0) = -\beta, \qquad \mathrm{Bi}(0) = \sqrt{3}\,\alpha, \qquad \mathrm{Bi}'(0) = \sqrt{3}\,\beta$$

where

$$\alpha = \frac{1}{3^{2/3}\,\Gamma(\frac{2}{3})} \approx 0.355028053887817, \qquad \beta = \frac{1}{3^{1/3}\,\Gamma(\frac{1}{3})} \approx 0.258819403792807.$$

Here Γ is the Gamma function, defined by

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} \mathrm{d}t, \tag{2}$$

but you do not need to know anything about its properties for this project. The Airy functions are useful in many problems involving transition regions of all kinds, for example in optical diffraction (the transition between relatively light and dark regions), wave theory, electron tunnelling, and asymptotic analysis. Ai and Bi have Maclaurin series given by

$$\mathrm{Ai}(z) = \alpha f(z) - \beta g(z), \qquad \mathrm{Bi}(z) = \sqrt{3}\big(\alpha f(z) + \beta g(z)\big)$$

where

$$f(z) = 1 + \frac{1}{3!}z^3 + \frac{1\cdot 4}{6!}z^6 + \frac{1\cdot 4\cdot 7}{9!}z^9 + \cdots$$

and

$$g(z) = z + \frac{2}{4!}z^4 + \frac{2\cdot 5}{7!}z^7 + \frac{2\cdot 5\cdot 8}{10!}z^{10} + \cdots .$$

For large $|z|$, any solution $y(z)$ of (1) is given asymptotically by the relation

$$y(z) \sim AF(z) + BG(z)$$

where A and B are complex constants, and where

$$F(z) = \frac{1}{\sqrt{\pi}}\, z^{-1/4} \exp(-\frac{2}{3}z^{3/2})(1 - \frac{5}{48}z^{-3/2} + \cdots)$$

and

$$G(z) = \frac{1}{\sqrt{\pi}}\, z^{-1/4} \exp(\frac{2}{3}z^{3/2})(1 + \frac{5}{48}z^{-3/2} + \cdots),$$

where the principal value is taken for any multi-valued function. The values of the constants A and B depend, of course, on precisely which solution y is being considered (Ai and Bi have different asymptotic behaviour, for instance). More surprisingly, perhaps, the values of A and B may also depend on which region of the complex plane is under consideration. This is known as Stokes' phenomenon, and the rays from the origin that divide the complex plane into different regions are known as Stokes lines. In the current case, there are three Stokes lines, two of which are given by the rays $\arg z = \pm\pi/3$.

In this project, we shall concentrate to start with on the region R given by $|\arg z| < \pi/3$. In that region, the appropriate values of A and B are $\frac{1}{2}$ and 0 respectively for $\mathrm{Ai}(z)$; for $\mathrm{Bi}(z)$, $B = 1$ but A is not important and can take any value (because $F(z)$ is negligible compared to $G(z)$ for large $|z|$ in R; that is, F is *subdominant*). Hence $\mathrm{Ai}(z) \to 0$ and $|\mathrm{Bi}(z)| \to \infty$ as $|z| \to \infty$ in R.

Programming note: You should write your own programs to compute the Airy functions: it is not sufficient simply to use the inbuilt MATLAB functions, or equivalent inbuilt functions for other software packages or programming languages, to calculate Airy functions although they are, of course, a convenient way to check your results. All calculations and evaluations are to be performed for *complex* numbers, not just real ones. Although MATLAB handles complex numbers quite well, most programming languages handle only real numbers, so you may have to write your own code to perform simple complex number operations such as multiplication. You should also ensure that all calculations performed by your program are in double precision.

Question 1 Show that

$$y(z) = \frac{1}{2\pi i}\int_C \exp(zt - \frac{1}{3}t^3)\,\mathrm{d}t$$

is a solution of 1. Here C is any contour that starts at $\infty e^{-2\pi i/3}$ and ends at $\infty e^{2\pi i/3}$. Show furthermore that this solution satisfies $y(0) = \alpha$, $y'(0) = -\beta$ and that it is therefore equal to $\mathrm{Ai}(z)$. [*Hint: deform C into two (straight) rays that meet at the origin. You may assume without proof the reflection formula for the Gamma function, viz.* $\Gamma(z)\,\Gamma(1-z) = \pi/\sin(\pi z)$.]

This integral representation of $\mathrm{Ai}(z)$ can be used to check the asymptotic expansion given above for large $|z|$, but you are not required to do this.

2 Numerical Integration of the Differential Equation

Question 2 Write a program to find $\mathrm{Bi}(z)$ for any $z \in R$, accurate to at least 4 significant figures, by performing a numerical integration of the defining differential equation (1) using any standard method. You should perform your integration along a ray joining the origin to z, using a real variable t to denote distance along the ray: this will require you to find a system of differential equations satisfied by $\mathrm{Re}\,y$ and $\mathrm{Im}\,y$ along the ray. Include the derivation of this system of equations in your write-up as well as the initial conditions (over which you are advised to take care). Also explain what checks you carried out to ensure the accuracy of your solutions. As a very simple first check, you may find it useful to know that $\mathrm{Bi}(1) \approx 1.20742$.

Use your program to evaluate $\mathrm{Bi}(z)$ at $z = 2, 4, 8, 16, e^{\pm i\pi/6}$ and one other non-real point of your choosing. Draw a graph of the behaviour of the (modulus of the) solution along one particular non-real ray of your choosing and give a plausible demonstration that the leading order asymptotic behaviour, $G(z)$, is indeed as stated in the Introduction.

Question 3 Modify your program to instead calculate $\mathrm{Ai}(z)$, and try to evaluate $\mathrm{Ai}(z)$ at the same points as in Question 2. You may find it useful to know that $\mathrm{Ai}(1) \approx 0.13529$. Draw a graph of $\mathrm{Ai}(z)$ for real positive z. Which of your evaluations are you confident are accurate? What goes wrong with the method? Why is this unavoidable? [*Hint: the term "stiff differential equation" (or "parasitic solution of a differential equation") may be relevant here: see, for instance, the book Numerical Recipes by Press et al.*]

One way to avoid this problem is, instead of integrating from $z = 0$ towards infinity, to start from a value of z with *large* modulus, and step towards the origin. The asymptotic expansion for $\mathrm{Ai}(z)$ (and the derivative of this expansion) can be used to approximate the initial conditions.

Question 4 Explain why this alternative approach should work. Write a program to implement it; start from $|z| = a$, for some large fixed constant a, and integrate towards the origin. Use only the zeroth order term of the asymptotic expansion (i.e., ignore $\frac{5}{48}z^{-3/2}$ and higher order terms in $F(z)$); a more advanced implementation might take more terms into account.

To start with you might like to use $a = 20$; but you should experiment with other values and explain what difference they might make. State the value you finally settle on and why.

Use your program to evaluate $\mathrm{Ai}(z)$ at the same points as in Question 2.

3 Matched expansions

Question 5 By finding series expansions about the origin, or otherwise, prove that the given expressions for the Maclaurin series of $\mathrm{Ai}(z)$ and $\mathrm{Bi}(z)$ are correct.

A much quicker, and more accurate, approach to evaluating the Airy functions is to avoid numerical integration altogether and instead use the analytic series expansions. In theory, the Maclaurin series for Ai and Bi are valid for all z, but in practice they are not very helpful for larger values of $|z|$ because of rounding errors caused by adding together large numbers of terms. Here we will try an approach based on using the Maclaurin series when $|z| < b$, for some fixed constant b, and using the asymptotic expansion when $|z| \geqslant b$; we hope to achieve accuracy at least as high as 4 significant figures, and preferably more.

Question 6 Investigate the feasibility and potential accuracy of this approach for evaluating $\mathrm{Ai}(z)$ on the positive real axis. You should use only the first two terms in the asymptotic expansion (i.e., do not attempt to find more terms in $F(z)$ than are given above), though you may use as many terms of the Maclaurin series as you wish. You should try various different values of b, and experiment with the number of terms to use from the Maclaurin series for best results. What level of accuracy is attainable?

Include a plot of your composite approximation and some sample values close to $|z| = b$.

How did you sum the Maclaurin series in order to minimize rounding errors?

How do you expect the time taken by this algorithm to compare with that for Question 4?

A professional implementation of this method (at least for real z) would use a selection of Chebyshev polynomial approximations in different overlapping regions and choose the best one automatically.

4 Stokes lines

Question 7 Use the programs you have developed in previous questions to describe how the behaviour of Ai and Bi with $|z|$ changes as the rays approach the Stokes line at $\arg z = \pi/3$ from within R.

Question 8 By experimenting with rays *outside* R, determine the location of the third Stokes line. How do Ai and Bi behave on this line?

Question 9 What can you say about the values of A and B in each of the three regions which lie between each pair of Stokes lines? Can you estimate these values from your numerical results?

What, if anything, can you say *on* the Stokes lines?

5 A particle in a constant force field

[Note that no knowledge of Quantum Mechanics is required for this section of the project: all required equations are given below.]

A one-dimensional quantum-mechanical particle is confined to the region $x > 0$ and is subjected to a force of constant magnitude k directed towards the origin. The governing equation for the wavefunction $\psi(x)$ is

$$-\frac{\hbar^2}{2m}\frac{\mathrm{d}^2\psi}{\mathrm{d}x^2} + kx\psi = \lambda\psi$$

with boundary conditions $\psi(0) = 0$, $\psi(x) \to 0$ as $x \to \infty$, where λ is the energy of the particle. This is a Sturm–Liouville problem with eigenvalue λ.

Question 10 Show, using your computed results from earlier questions, that there is a discrete set of energy eigenvalues λ_n. Find an approximate value for the first two of these eigenvalues in units where $\hbar^2k^2/2m = 1$.

9 Dynamic Programming

9.1 Policy Improvement for a Markov Decision Process (4 units)

This project is self-contained mathematically; background information is provided in the Part II course on Optimisation and Control.

1 A Car Replacement Problem

Car owners are haunted by the following problem. Every day, the operating cost for their car increases, as does the probability that the car breaks down. Even worse, when trading in the car for a different one dealers will pay less for older cars and charge more for newer ones. The problem, then, is to find an optimal policy for trading in the car.

We model the problem as a Markov decision process. Let $g_j(u)$ be the instantaneous cost incurred if one takes action u in state j and let $p_{jk}(u)$ be the probability of then moving to state k. Define sequences $\gamma^{(n)},\ f_j^{(n)}, u_j^{(n)}$ by the recursions

$$\gamma^{(n)} + f_j^{(n)} = g_j\big(u_j^{(n)}\big) + \sum_k p_{jk}\big(u_j^{(n)}\big) f_k^{(n)} \tag{1}$$

and

$$u_j^{(n+1)} \text{ is the } u\text{-value minimising } g_j(u) + \sum_k p_{jk}(u) f_k^{(n)}. \tag{2}$$

The following exercise may help to gain some intuition.

> **Question 1** Consider the following stationary policy: for fixed n, whenever state j occurs take action $u_j^{(n)}$. What is the long-term average cost of this policy?

Note that the values $f_j^{(n)}$ determined by (1) are arbitrary up to an additive constant and can be normalised, for example by letting $f_1^{(n)} = 0$. If the matrix of transition probabilities is irreducible in every stage, then (1) will always have a solution for f. The sequence $\gamma^{(n)}$ is non-increasing, and will converge to a minimum value γ in a finite number of steps if u can take only a finite number of values. The policy $u_j^{(n)}$ will then have converged to an average optimal policy.

> **Question 2** Instantiate the above framework for the car replacement problem. You may want to introduce states representing the age of the car in appropriately chosen units of time, and an additional state in which the car is written off and has a trade-in value of zero. Describe the set of actions, and define the instantaneous costs $g_j(u)$ and the transition probabilities $p_{jk}(u)$.

> **Question 3** Write a program to find the optimal replacement policy. You are not required to write your own linear algebra routines, but you should describe any mathematical manipulations involved in bringing the equations in the desired form. Give a

j	age in years	purchase price	trade-in price	operating cost	survival probability
1	0	5000	3500	860	0.963
2	2	3150	2170	1025	0.794
3	4	2285	1500	1225	0.568
4	6	1545	900	1430	0.255
5	8	1050	590	1815	0.001
6	10	600	330	2240	0.000

Table 1: Instance of the car replacement problem with time units of two years and $N = 6$

j:	1	2	3	4	5	6	7
sell/keep:	keep	keep	keep	keep	keep	sell	sell
buy car of age:	–	–	–	–	–	2	2

Table 2: Policy for the problem of Table 1

j	age in years	purchase price	trade-in price	operating cost	survival probability
1	0	5000	3500	200	0.999
2	0.5	4285	3000	210	0.995
3	1	3750	2650	220	0.990
4	1.5	3430	2375	230	0.979
5	2	3150	2170	240	0.968
6	2.5	2900	1950	250	0.956
7	3	2645	1850	260	0.936
8	3.5	2475	1625	275	0.917
9	4	2285	1500	290	0.898
10	4.5	2130	1350	300	0.879
11	5	1970	1225	315	0.860
12	5.5	1760	1060	320	0.836
13	6	1545	900	335	0.801
14	6.5	1400	780	350	0.761
15	7	1260	700	365	0.697
16	7.5	1140	625	380	0.600
17	8	1050	590	400	0.482
18	8.5	940	520	430	0.300
19	9	830	470	465	0.129
20	9.5	720	400	520	0.020
21	10	600	330	560	0.000

Table 3: Instance of the car replacement problem with time units of six months and $N = 21$

clear and concise description of your algorithm; don't forget to mention what starting conditions you use. Run your program on the data contained in the file *table1.cvs* available from the CATAM website and displayed in Table 1, and compare your results to the policy in Table 2.

Question 4 Give the optimal replacement policy for the data in the file *table2.cvs* available from the CATAM website and displayed in Table 3.

Question 5 Suppose that purchase price, trade-in price, operating cost, and survival probability are all monotonically increasing or decreasing in the obvious direction. Suppose further that the optimal policy tells you to sell a car when it reaches a certain age, but that you neglect to do so. Is it possible that the same policy stipulates hanging on to the car now that it is older? Either construct an example for which the optimal policy is of this kind, or prove that this is impossible.

References

[1] R.R.Weber, *Course notes on Optimisation and Control*, Section 7.4. `http://www.statslab.cam.ac.uk/~rrw1/oc/`.

9 Dynamic Programming

9.4 Option Pricing in Mathematical Finance (6 units)

This project is connected with material in the Stochastic Financial Models course. Students who are not taking that course but who wish to attempt the project will find the necessary definitions and background material in the references.

1 Black-Scholes model

A standard model used in option pricing is that the logarithm of the stock price follows a Brownian motion. Hence, if S_t is the stock price at time t, we assume that $\log(S_t/S_0)$ is normally distributed with mean μt and variance $\sigma^2 t$, where σ is the volatility, $\mu = \rho - \sigma^2/2$, and ρ is the continuously-compounded riskless interest rate.

The celebrated Black-Scholes formula gives the price of a call option (exercised only at expiry). The price of the option is

$$S_0\Phi\Big(\frac{\log(S_0/c) + (\rho + \sigma^2/2)t_0}{\sigma\sqrt{t_0}}\Big) - ce^{-\rho t_0}\Phi\Big(\frac{\log(S_0/c) + (\rho - \sigma^2/2)t_0}{\sigma\sqrt{t_0}}\Big) \tag{1}$$

where c is the strike price and t_0 is the expiry time. (See the Appendix for details of how to calculate Φ.)

Question 1 Write a routine to evaluate the Black-Scholes price (1). Compile a table of the price when $c = 40$, $S_0 = 52, 100$ or 107, $\sigma = 0.5$, $\rho = 0.035$, and $t_0 = 2$ or 3.

Question 2 How does the price vary with each of the parameters c, S_0, σ, ρ, t_0? Keep your explanations brief, but support them with solid mathematics where necessary.

2 Bernoulli approximation

The most widely-used method for approximating option prices which are based on the Black-Scholes model is to replace the Brownian motion by a discrete-time simple random walk. This approximation breaks up the interval $[0, t_0]$ into $[0, t_0/n, 2t_0/n, \ldots, (n-1)t_0/n, t_0]$ and assumes that between the times it_0/n and $(i+1)t_0/n$, $i = 0, \ldots, n-1$, the increment in the logarithm of the price is g or $-g$ with probability p or $1-p$ respectively, where g and p are chosen so that the increment has mean $\mu t_0/n$ and variance $\sigma^2 t_0/n$.

This approximation is primarily of interest for cases such as the American put where exact formulae are not available. For a European (or American) call option, the price obtained from the random walk will approximate the true price obtained from the Black-Scholes formula, and this can be a useful benchmark to judge the performance of the approximation.

One way to implement the approximation is to set

$$V_{i,j} = \big(pV_{i+1,j+1} + (1-p)V_{i+1,j}\big)e^{-\rho t_0/n} \quad \text{for } j = 0, \ldots, i \text{ and } i = n-1, \ldots, 0 \tag{2}$$

with boundary conditions

$$V_{n,j} = \Big(S_0 e^{(2j-n)g} - c\Big)^+ \quad \text{for } j = 0, \ldots, n. \tag{3}$$

Then $V_{0,0}$ is the approximate price.

Question 3 Calculate g and p as functions of the parameters.

Question 4 For the data in Question 1 and $n = 27$, compile a table of the approximate prices. How do they compare to the prices obtained from the Black-Scholes formula?

Question 5 What is the complexity of this algorithm, as a function of n?

Question 6 Consider an at-the-money case ($c = S_0$). Plot a graph of your approximation as a function of n. Indicate on your graph the true value obtained from the Black-Scholes formula. What do you notice? Explain this behaviour.

Question 7 Estimate the rate at which the error decreases as n increases. Explain every step and justify your answers.

3 American Put

Consider the case of an American put; now early exercise of the option may be optimal and no closed-form formula exists for the price.

Question 8 Modify your programs to approximate the price of the option by considering how equations (2) and (3) should be changed in this situation. Compile a table of the approximate price of the option for the same values of the parameters used in Question 1. Comment briefly on how the approximate price varies with n in this case.

4 Extrapolation

Suppose that f_n is the approximation to the option price, and we wish to find the limiting value of f_n as $n \to \infty$. One method of extrapolation assumes that f_n may be approximated by a polynomial in $1/n$:

$$f_n \approx g_0 + g_1 n^{-1} + g_2 n^{-2} + \cdots + g_s n^{-s}.$$

The limiting value of f_n is then approximated by g_0. One way to achieve this is as follows. Let $n_m = r^m n_0$ and calculate f_n at $n = n_0, \ldots, n_s$. Set

$$a_{m,0} = f_{n_m} \quad \text{for } m = 0, \ldots, s$$

and recursively calculate

$$a_{m,i} = a_{m,i-1} + \frac{a_{m,i-1} - a_{m-1,i-1}}{r^i - 1} \quad \text{for } m = i, \ldots, s \text{ and } i = 1, \ldots, s.$$

Then $a_{s,s}$ is taken as the approximation for g_0.

Question 9 Experiment with this extrapolation procedure for small values of r and s, say 2 to 4, for the at-the-money European call case studied above. How does this extrapolation compare in accuracy with just calculating f_n for a single suitably large value of n? Try to estimate the error analytically. Does your answer depend on whether n_0 is odd or even? If so, carefully explain why.

5 Binomial approximation

The approximation in Section 2 uses a Bernoulli (two-valued) distribution between time steps. The method may be refined by replacing the Bernoulli distribution between times it_0/n and $(i+1)t_0/n$ by, say, a binomial distribution taking $k+1$ equally-spaced values (for some $k \geqslant 1$) with mean and variance chosen to match those of the Brownian motion.

> **Question 10** Implement this refinement, and explain how you calculate p and g in this case. How do prices produced by the refined algorithm ($k > 1$) differ from those produced by the Bernoulli scheme ($k = 1$)? Does your answer depend on whether you are pricing the European call or American put option, and if so why? How does the computation time for this algorithm vary with n and k?

Appendix

An easy method to approximate the standard normal distribution function is as follows. For $x \geqslant 0$ set

$$1 - \Phi(x) = \frac{t}{2} \exp\Big(-\frac{x^2}{2} + \sum_{i=0}^{9} a_i t^i\Big)$$

where $t = (1 + x/\sqrt{8})^{-1}$ and

$$\begin{aligned}(a_0, \ldots, a_9) = (&-1.26551223, 1.00002368, 0.37409196,\\ &0.09678418, -0.18628806, 0.27886807,\\ &-1.13520398, 1.48851587, -0.82215223, 0.17087277).\end{aligned}$$

For $x < 0$ set $\Phi(x) = 1 - \Phi(-x)$.

If you are using CCATSL, `PhiCL(x)` calculates $\Phi(x)$ for you.

References

[1] J. Hull, *Options, Futures and Other Derivative Securities*. Prentice-Hall, 1989.

10 Statistics

10.9 Markov Chain Monte Carlo (6 units)

Bayesian inference is covered in the IB Statistics course, and developed further in the II(D) Principles of Statistics course. Knowledge of the IB Markov Chains course, whilst useful, is not necessary, and there is no requirement to quote results from it.

Introduction

In Bayesian statistics, it is essential to be able to sample from the posterior distribution of unknown parameters given some data. In arbitrary, high-dimensional problems, this is not possible analytically, but in recent years Markov Chain Monte Carlo methods (MCMC) have become a popular alternative.

Markov Chain

The key idea is quite simple. We want to sample from a distribution $\pi(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^m$, but cannot do so directly. Instead we create a discrete-time Markov chain $\mathbf{X}(n)$ (taking values in $\mathbb{R}^m$) such that $\mathbf{X}(n)$ has equilibrium distribution π. Then

$$\mathbf{X}(n) \to \mathbf{X} \sim \pi \quad \text{in distribution, as } n \to \infty \tag{1}$$

and

$$\frac{1}{N}\sum_{n=1}^{N} f\big(\mathbf{X}(n)\big) \to \mathbb{E}_\pi\big(f(\mathbf{X})\big) \quad \text{as } N \to \infty$$

where f is any real-valued function on $\mathbb{R}^m$ for which the right-hand side above is well-defined. The second of these limits can be used to calculate means and variances of components of $\mathbf{X}$, as well as approximations to the distribution functions. For example, $f(\mathbf{x}) = x_i$ gives the mean of the ith component X_i, and $f(\mathbf{x}) = I(x_i \leqslant b)$ gives the distribution function of X_i at point b.

Gibbs sampler

Suppose we do not have a tractable closed-form expression for the equilibrium density $\pi(\mathbf{x}) = \pi(x_1, \ldots, x_m)$, but we do know the induced full conditional densities $\pi(x_i|\mathbf{x}_{-i})$, where $\mathbf{x}_{-i}$ is the vector $\mathbf{x}$ omitting the ith component, $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$.

A systematic form of the Gibbs sampler algorithm proceeds as follows. First, pick an arbitrary starting value $\mathbf{x}^0 = (x_1^0, \ldots, x_m^0)$. Then successively make random drawings from the full conditional distributions $\pi(x_i|\mathbf{x}_{-i})$, $i = 1, \ldots, m$, as follows:

$$\begin{aligned}
&x_1^1 \text{ from } \pi(x_1|\mathbf{x}_{-1}^0)\\
&x_2^1 \text{ from } \pi(x_2|x_1^1, x_3^0, \ldots, x_m^0)\\
&x_3^1 \text{ from } \pi(x_3|x_1^1, x_2^1, x_4^0, \ldots, x_m^0)\\
&\vdots\\
&x_m^1 \text{ from } \pi(x_m|\mathbf{x}_{-m}^1).
\end{aligned}$$

This cycle completes a transition from $\mathbf{x}^0 = (x_1^0, \ldots, x_m^0)$ to $\mathbf{x}^1 = (x_1^1, \ldots, x_m^1)$. Repeating the cycle produces a sequence $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots$ which is a realization of a Markov chain, which is

known as the Gibbs sampler. We call $\pi(\mathbf{x}, \mathbf{y})$ the transition probability density of this Markov chain.

Question 1 Assume that the Markov chain $\mathbf{X}(n)$ takes values in a finite subset $S \subset \mathbb{R}^m$. Verify that π is an equilibrium distribution for this chain. That is, check that for all $\mathbf{y} \in S$,

$$\sum_{\mathbf{x}} \pi(\mathbf{x})\pi(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y}).$$

It can be shown that this implies that π is the equilibrium distribution of the Gibbs sampler, in the sense of (1), but do not attempt to prove it. Thus our estimate of $\mathbb{E}_\pi\big(f(\mathbf{X})\big)$, taken over N iterations, is

$$\frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}^n).$$

Football data

Data from the performance of K football teams, over T years has been scored on a scale of 0 (no wins) to 114 (win in all 38 games), with a win scoring three and a draw scoring one point. Let us model Y_{kt}, the score of the kth team in year t, as

$$Y_{kt}|\text{parameters} \sim \mathrm{N}(\mu_k, \sigma_k^2), \quad \text{for } k = 1, \ldots, K \text{ and } t = 1, \ldots, T$$

with the hierarchical prior structure that the team mean μ_k and variance σ_k^2 are independently distributed, given θ, as

$$\begin{aligned} \mu_k|\theta &\sim \mathrm{N}(\theta, \sigma_0^2) \\ \sigma_k^{-2} &\sim \Gamma(\alpha_0, \beta_0), \end{aligned}$$

where σ_0^2, α_0 and β_0 are known parameters, and θ is a second-stage prior with distribution

$$\theta \sim \mathrm{N}(\mu_0, \tau_0^2),$$

where μ_0 and τ_0^2 are known parameters.

The Gibbs sampler is well suited to the analysis of hierarchical models, since the full one-dimensional conditional distributions often have extremely simple forms. For example, in the above model,

$$\begin{aligned} \mu_k|\boldsymbol{\mu}_{-k}, \theta, \boldsymbol{\sigma}^2, \mathbf{y} &\sim \mathrm{N}\left(\frac{\sigma_k^{-2}\sum_{t=1}^{T} y_{kt} + \theta\sigma_0^{-2}}{T\sigma_k^{-2} + \sigma_0^{-2}}, \frac{1}{T\sigma_k^{-2} + \sigma_0^{-2}}\right) \\ \theta|\boldsymbol{\sigma}^2, \boldsymbol{\mu}, \mathbf{y} &\sim \mathrm{N}\left(\frac{\sigma_0^{-2}\sum_{k=1}^{K} \mu_k + \mu_0\tau_0^{-2}}{K\sigma_0^{-2} + \tau_0^{-2}}, \frac{1}{K\sigma_0^{-2} + \tau_0^{-2}}\right) \\ \sigma_k^{-2}|\boldsymbol{\sigma}^2_{-k}, \boldsymbol{\mu}, \theta, \mathbf{y} &\sim \Gamma\left(\alpha_0 + \frac{T}{2}, \beta_0 + \frac{1}{2}\sum_{t=1}^{T}(y_{kt} - \mu_k)^2\right). \end{aligned}$$

Question 2 Verify the one-dimensional conditional distributions given above. What is the marginal prior distribution of μ_k?

Question 3 Implement the Gibbs sampler to sample from the posterior distribution of $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \theta)$ given $\mathbf{y}$. You can find the data for $\mathbf{y}$ in the file `II-10-9-2015football.csv` on the CATAM website. Take as known $\sigma_0 = 10$, $\alpha_0 = 10^{-5}$, $\beta_0 = 10^{-3}$, $\mu_0 = 60$, $\tau_0 = 20$. Briefly discuss what these prior parameter values have assumed about the football data. With reference to these priors, how did you choose the initial state of the Markov chain?

If you wish, you can use a package to simulate distributions, but you should implement the Gibbs sampler yourself without using library routines.

Question 4 Use your Gibbs sampler to estimate the posterior mean of each parameter θ, μ_k, σ_k^2. Plot a histogram of the posterior distribution of θ and comment on your histogram. Explain how you obtained it.

Question 5 Now choose a team k. Estimate the posterior probability that your chosen team is above average, $\mathbb{P}(\mu_k > \theta|\mathbf{y})$.

Question 6 Build up an idea of how accurate your estimates for μ_k and $\mathbb{P}(\mu_k > \theta|\mathbf{y})$ are, for your chosen team k, by performing independent runs of the Gibbs sampler, computing estimates for each of the parameters on each run and then computing the sample variances of these estimates. Comment on how fast your estimates converge by considering sample variances at different values of N.

Question 7 Now try letting the algorithm run for an initial period of M cycles before calculating estimates based on a further N iterations. This might allow the distribution to settle down to equilibrium before being measured. Calculate sample variances (as the previous question) for a few suitable values of M to see if this makes any noticeable difference. Explain why you do or do not see a difference.

Question 8 In any MCMC procedure we must ensure that we are exploring the full sample space. One way to check this is to run a number of chains that start from different points. Using a few widely dispersed starting points confirm, or otherwise, that your results are independent of the starting point. What is the effect of running an initial M cycles in this situation?

Hint. Recall that the Gamma distribution $\Gamma(\gamma, \lambda)$ has density

$$f(x) = \frac{\lambda^\gamma x^{\gamma-1} e^{-\lambda x}}{\Gamma(\gamma)},$$

with mean γ/λ and variance γ/λ^2. Also recall that a Gamma $\Gamma(n/2, \lambda)$ has the same distribution as the scaled chi-squared $(2\lambda)^{-1}\chi_n^2$.

You can assume that a $\Gamma(2.50001, \lambda)$ is approximately distributed as a χ_5^2 (suitably scaled), which in turn is exactly equal to the sum of two independent exponentials plus an independent normal squared.

10 Statistics

10.15 Variable Selection and the Bias-Variance Tradeoff (8 units)

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

1 Introduction

Consider the following linear model with a univariate response and p covariates:

$$Y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2). \tag{1}$$

We assume throughout that all variables are zero mean. Note also that introducing an intercept is not necessary, since it can be captured by augmenting the covariate and regression vector as follows:

$$b + X\beta = \begin{pmatrix} 1 & X \end{pmatrix} \begin{pmatrix} b \\ \beta \end{pmatrix}.$$

We will denote a training dataset of N response-covariate tuples by $\mathcal{T} = \{(y_t, x_t) \mid t = 1, \ldots, N\}$, where each x_t is a $1 \times p$ row vector representing an observation. Alternatively, we may employ matrix notation, letting $\mathbf{y} = (y_t)_{t=1:N}$ be a row vector and $\mathbf{x} = (x_{ti})$ an $N \times p$ matrix where each row corresponds to an observation. The *least squares* (LS) estimate of β then is the minimiser of the *residual sum of squares* (RSS) over the training set:

$$\mathrm{RSS}(\hat{\beta}; \mathcal{T}) = \frac{1}{N}\sum_{t=1}^{N}(y_t - x_t\hat{\beta})^2 = \frac{1}{N}(\mathbf{y} - \mathbf{x}\hat{\beta})^T(\mathbf{y} - \mathbf{x}\hat{\beta}), \text{ and } \hat{\beta}^{\mathrm{LS}}(\mathcal{T}) = \operatorname*{argmin}_{\hat{\beta}} \mathrm{RSS}(\hat{\beta}; \mathcal{T}). \tag{2}$$

Assuming $N > p + 1$, which we do, the LS estimator can be written in closed-form as

$$\hat{\beta}^{\mathrm{LS}}(\mathcal{T}) = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}. \tag{3}$$

The dependence on the training set will be omitted when understood. In this project we will often refer to a subset of covariates as a *model* $\mathcal{M} \subseteq \{1, \ldots, p\}$. The LS estimate for $\mathcal{M}$ is computed on a reduced dataset $\mathcal{T}^{\mathcal{M}}$, obtained by deleting all covariates not in the model:

$$\mathcal{T}^{\mathcal{M}} = \left(y_t, (x_{ti})_{i \in \mathcal{M}}\right)_{t=1:N}.$$

For computational ease, we will instead represent the LS estimate for $\mathcal{M}$ in the p-dimensional space of the original model. We denote this representation by $\hat{\beta}^{\mathcal{M}}$, where

$$\hat{\beta}^{\mathcal{M}}_j(\mathcal{T}) = \begin{cases} \hat{\beta}^{\mathrm{LS}}_{\pi(j)}(\mathcal{T}^{\mathcal{M}}) & \text{if } j \in \mathcal{M}, \\ 0 & \text{otherwise.} \end{cases}$$

Here $\pi : \mathcal{M} \to \{1, 2, \ldots, ||\mathcal{M}||\}$ maps indices of covariates in the model to their respective indices in the reduced dataset $\mathcal{T}^{\mathcal{M}}$, so that $\mathbf{x}\hat{\beta}^{\mathcal{M}}(\mathcal{T})$ is a (simpler) notation for $\mathbf{x}^{\mathcal{M}}\hat{\beta}^{\mathrm{LS}}(\mathcal{T}^{\mathcal{M}})$.

2 The Bias-Variance tradeoff

Question 1 Assume model (1). Let $\hat{\beta}$ be an estimator of β. Show that the expected squared prediction error at a fixed, arbitrary location u can be decomposed as follows:

$$\mathbb{E}_{\mathcal{T},y|u}(y - u\hat{\beta})^2 = \sigma^2 + \left(u\beta - \mathbb{E}_{\mathcal{T}}[u\hat{\beta}]\right)^2 + \mathrm{Var}_{\mathcal{T}}[u\hat{\beta}],$$

where $y \perp \mathcal{T}$. The summands on the right-hand side are often referred to as the *irreducible variance*, *squared estimation bias* and *estimation variance*, respectively. Describe what effect deleting the pth covariate can have on each of these quantities for the LS estimator (i.e., switch $\hat{\beta} = \hat{\beta}^{\mathrm{LS}}$ with $\hat{\beta} = \hat{\beta}^{\{1,\ldots,p-1\}}$). You may consider the simplest non-trivial case, where $\mathbf{x}$ is fixed such that $\mathbf{x}^T\mathbf{x} = \mathbf{I}_p$. It might further be useful to look at $\beta_p = 0$, and then at $\beta_p \neq 0$.

Question 2 Consider model (1) with $p = 10$, $\sigma^2 = 1$, and $X \sim N(0, I_p)$, and set

$$\beta = (-0.5, 0.45, -0.4, 0.35, -0.3, 0.25, -0.2, 0.15, -0.1, 0.05)^T.$$

Simulate a training dataset with $N_{\mathrm{tr}} = 30$ and a test dataset with $N_{\mathrm{te}} = 1000$. Now consider

$$\mathcal{M}_1 = \{1\},\ \mathcal{M}_2 = \{1, 2\},\ \ldots,\ \mathcal{M}_p = \{1, \ldots, p\}.$$

Write a procedure that computes the training and test error of $\hat{\beta}^{\mathcal{M}_j}$ for $j = 1, \ldots, p$. Repeat the experiment 100 times and report your results in a plot of training and test RSS averaged over experiments, against model size. What happens if $N_{\mathrm{tr}} = 200$, and why?

The above demonstrates an effect that holds in much greater generality, namely that suitably reducing the complexity of a model (in this instance, the number of variables involved) can improve prediction accuracy. There may also be gains in *model discovery*, *interpretability*, and, of course, *reduced observation costs*. Consequently, variable selection methods are of interest.

3 Variable selection methods

We consider two approaches to variable selection, *subset selection* and *shrinkage-based methods*. Subset selection methods look among all possible subsets of variables for the one that minimises some suitable estimate of prediction error. The search problem becomes infeasible for large p, and non-exhaustive greedy search methods have to be employed. Shrinkage-based variable selection methods will instead penalise the RSS by a penalty term that forces the LS regression coefficients to shrink in a manner that favours *exact* zeros in $\hat{\beta}$.

3.1 Subset selection

Question 3 **Best subsets selection.** Write a procedure *bestsubset* which takes as input a training dataset $\mathcal{T}$ and outputs a $p \times p$ matrix B, whose jth column contains $\hat{\beta}^{\mathcal{M}_j}$ for the best performing model (in the sense of RSS) of size j, $\mathcal{M}_j$:

$$\mathcal{M}_j(\mathcal{T}) = \underset{\mathcal{M}:\, ||\mathcal{M}||=j}{\mathrm{argmin}}\ RSS\left(\hat{\beta}^{\mathcal{M}}(\mathcal{T}); \mathcal{T}\right).$$

What is the size of the model space $\{\mathcal{M} \mid \mathcal{M} \subseteq \{1, \ldots, p\}\}$? Your procedure will handle with difficulty values of p for which the size of the search space $\{\mathcal{M} \mid \mathcal{M} \subseteq \{1, \ldots, p\}, ||\mathcal{M}|| = j\}$ exceeds 10^5 for any $j \in \{1, \ldots, p\}$. What is the smallest such p (show your work)?

Question 4 **Greedy subset selection**. Write a procedure *greedysubset*, using the same input-output format as before, that incrementally builds up the model sequence $\mathcal{M}_j$ by adding at each iteration the covariate that improves model fit the most:

$$\mathcal{M}_0 = \emptyset, \quad \mathcal{M}_{d+1}(\mathcal{T}) = \mathcal{M}_d(\mathcal{T}) \cup \left\{ l \mid l = \operatorname*{argmin}_j RSS\left(\hat{\beta}^{\mathcal{M}_d(\mathcal{T})\cup\{j\}}(\mathcal{T}); \mathcal{T}\right) \right\}.$$

Can the fact that the family of models $\mathcal{M}_0, \ldots, \mathcal{M}_p$ is nested be used to gain in computational efficiency? Explain how, without effecting the change. Assuming that $\mathcal{M}_j = \{1, \ldots, j\}$, you might want to consider the upper left $j \times j$ block of $\left((x^{\mathcal{M}_{j+1}})^T x^{\mathcal{M}_{j+1}}\right)^{-1}$.

Question 5 **Forward F-test.** Amend *greedysubset* to stop whenever the newly added variable does not significantly improve fit (at the $p = .05$ level), using the F-statistic

$$\frac{\mathrm{RSS}(\hat{\beta}^{\mathcal{M}_d}) - \mathrm{RSS}(\hat{\beta}^{\mathcal{M}_{d+1}})}{\mathrm{RSS}(\hat{\beta}^{\mathcal{M}_{d+1}})/(N-d-1)},$$

which you may assume follows an $F_{1,N-d-1}$ distribution (you may use the MATLAB function *cdf*). Would this method work for best subset selection?

Question 6 We can represent a sparse (linear regression) estimator more generally as an algorithm that takes as input a training set $\mathcal{T}$ and outputs a sequence of p candidate regression vectors for each model size (i.e., the jth candidate $\hat{\beta}^{(j)}(\mathcal{T})$ has precisely $p - j$ zeros). Best and greedy subset search are special cases of this definition for which each candidate is a least squares solution, a condition we will not insist on here. We would like to select among candidates on the basis of estimated prediction error $\hat{\mathrm{PE}}$:

$$\hat{\beta}^{\mathrm{CV}}(\mathcal{T}) = \hat{\beta}^{j^\star}(\mathcal{T}), \text{ where } j^\star = \operatorname*{argmin}_j \left\{ \hat{\mathrm{PE}}(j, \mathcal{T}) \right\}.$$

The prediction error can be estimated using 10-fold cross-validation as

$$\hat{\mathrm{PE}}(j, \mathcal{T}) = \frac{1}{10} \sum_{k=1}^{10} RSS\left(\hat{\beta}^{(j)}(\mathcal{T}^{-k}); \mathcal{T}^k\right),$$

where T^k is the kth fold of the training set and $\mathcal{T}^{-k}$ its complement:

$$\begin{aligned} \mathcal{T}^k &= \left\{ (y_{\pi(n)}, x_{\pi(n)}) \mid k - 1 < \frac{10n}{N} \leqslant k \right\}, \\ \mathcal{T}^{-k} &= \left\{ (y_{\pi(n)}, x_{\pi(n)}) \mid \frac{10n}{N} \leqslant k - 1 \text{ or } \frac{10n}{N} > k \right\}, \end{aligned} \tag{4}$$

where π is a random permutation of $\{1, \ldots, N\}$ (you may use the MATLAB function *randperm*). Implementing the above for an arbitrary sparse estimator would involve a function taking another function as an argument. In MATLAB, this can be achieved using *function handles*, as demonstrated by *handle_demo.m* and *testerror.m* available from the CATAM website. Write a procedure *crossval* that implements the above (the MATLAB functions *ismember* and *find* might be useful in this). This procedure should take as input $\mathcal{T}$ and a sparse estimator, and output $\hat{\beta}^{\mathrm{CV}}(\mathcal{T})$.

3.2 The Lasso estimator

The Lasso estimator penalises the RSS by the $L1$ norm of the regression coefficients:

$$\hat{\beta}^{(\mathrm{L},\lambda)}(\mathcal{T}) = \underset{\hat{\beta}}{\operatorname{argmin}} \left\{ \mathrm{RSS}\left(\hat{\beta}; \mathcal{T}\right) + \lambda \sum_{j=1}^{p} \left|\hat{\beta}_j\right| \right\} \tag{5}$$

Question 7 Express the Lasso as a quadratic program with linear constraints.

In the Lasso estimator, the degree of sparsity is controlled *indirectly* via the penalty weight λ, rather than directly as in earlier methods. For $\lambda = 0$ the full model is employed, whereas increasingly many covariates are deleted from the model as $\lambda \to \infty$. Given an algorithm for solving (5), we can then use cross-validation to select among any finite set of values $\lambda_1 < \lambda_2 < \cdots < \lambda_q$ for λ. For simplicity, we will continue here to perform cross-validation to select model size rather than λ. To do so, we will rely on the LARS algorithm, which, subject to certain minor assumptions and modifications that do not concern us here, allows us to compute in an efficient manner one Lasso solution for each model size. The file *monotonic_lars.m* available from the CATAM website contains an implementation of this modified LARS algorithm that can be used as input to *crossval*.

Question 8 The file *prostate.dat* available from the CATAM website contains a prostate cancer dataset.* The dataset has been preprocessed to standardise the covariates and make all variables zero mean, so that you can avoid using an intercept. Column 1 contains the response, *lpsa*, and columns 2 to 9 the covariates *lcavol*, *lweight*, *age*, *lbph*, *svi*, *lcp*, *gleason*, and *pgg45*. Augment the dataset by adding four zero-mean, unit-variance covariates sampled from a distribution of your liking *independently of variables in prostate.dat*. Separate the data into a training dataset of size 70 and a test dataset of size 27. Perform a variable selection analysis of the data using the tools developed above. Present and discuss your results.

*reproduced from `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`

11 Statistical Physics

11.4 Molecular dynamics simulation of a classical gas (10 units)

This project can be done with basic knowledge of classical statistical physics, as covered in the Statistical Physics course.

The aim of this project is to study the statistical physics of a gas of classical molecules by simulating their microscopic trajectories.

We wish to mimick the trajectories of the molecules as they interact with each other. At any given time, the acceleration of an individual molecule i is given by Newton's second law: the molecule's acceleration times its mass $m_i\mathbf{a}_i$ is given by the sum of the forces due to the other particles. Given initial positions and velocities, then we seek to solve

$$\begin{aligned} \frac{d\mathbf{r}_i}{dt} &= \mathbf{v}_i \\ \frac{d\mathbf{v}_i}{dt} &= \mathbf{a}_i = \frac{1}{m_i}\sum_{j\neq i}\mathbf{F}_{ij} \end{aligned} \tag{1}$$

where $\mathbf{F}_{ij}$ is the force on the ith particle due to the jth particle. In this project we will assume all particles have unit mass: $m_i = 1$ for all i.

Let us take the intermolecular force to be determined by the Lennard-Jones potential: $\mathbf{F}_{ij}(\mathbf{r}_{ij}) = -\nabla V(r_{ij})$, with $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ the displacement vector, $r_{ij} = |\mathbf{r}_{ij}|$, and

$$V(r) = V_0\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]. \tag{2}$$

This potential gives a fairly accurate description of a Van der Waals gas: weak attraction at long and moderate distances with a strong repulsion at short distances. In this project, we will work with units such that $V_0 = 4$ and $\sigma = 1$.

1 Particle trajectories

In the simulation, we need a discrete version of the equations of motion (1). Given particle positions $\mathbf{r}_i$ at times $t - \Delta t$ and t, we find $\mathbf{r}_i(t + \Delta t)$ and $\mathbf{v}_i(t)$ using

$$\begin{aligned} \mathbf{r}_i(t+\Delta t) &= 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\Delta t) + \mathbf{a}_i(t)(\Delta t)^2 \\ \mathbf{v}_i(t) &= \frac{1}{2\Delta t}\left[\mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t-\Delta t)\right]. \end{aligned} \tag{3}$$

Question 1 Show that in the small Δt limit the expressions (3) approach the exact equations of motion (1). Why is this discretization better than the simpler method

$$\begin{aligned} \mathbf{r}_i(t+\Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t \\ \mathbf{v}_i(t+\Delta t) &= \mathbf{v}_i(t) + \mathbf{a}_i(t)\Delta t \ ? \end{aligned}$$

For simplicity, here we will study a two-dimensional gas of molecules. Generalization to three dimensions is straightforward, but increases computational time.

The potential (2) is singular as the interparticle distance becomes small. In order avoid beginning with a configuration of molecules which has an arbitrarily large energy, we will insist that the initial positions of the molecules lie close to the points of a grid, or lattice. Failure to take this into account could lead to an initial condition where 2 particles (or more) are very close, in which case the corresponding $V(r)$ is too large and could lead to artificial numerical instabilities. We do not wish to investigate these instabilities here, so be sure to follow the directions below.

In order to simulate a homogeneous gas where intermolecule interactions are the dominant effect in changing particle trajectories, we will implement periodic boundary conditions. For $0 \leqslant x < L_x$ and $0 \leqslant y < L_y$ and integers n_x and n_y, any point which can be written as $(x + n_x L_x, y + n_y L_y)$ is identified with (x, y). When computing the distance between molecules, you should ensure that you use the shortest distance. For example, if the x-component of the displacement has length $\hat{\mathbf{x}} \cdot (\mathbf{r}_i - \mathbf{r}_j) > L_x/2$ then the distance should be computed using $\mathbf{r}_j \to \mathbf{r}_j + (L_x, 0)$.

Programming Task:

First we must initialize the molecule positions and velocities. Let us follow these steps

1. Allow for the following inputs: Number of molecules, N. Size of the system in each direction, L. Maximum distance which any particle can initially be separated from a grid point, δr. Initial speed parameter, v_0. Time step, Δt.
2. Define coordinates for grid points $\mathbf{g}_i$. The number of grid points should be at least as large as N, but not too much larger.
3. Let the ith particle have initial position $\mathbf{r}_i(0) = \mathbf{g}_i + \mathbf{s}\,\delta r$, where the two components of $\mathbf{s}$ are random numbers uniformly distributed between -1 and 1.
4. For the initial velocities, we will want to investigate the following choices:

   IC1: Random direction, uniform magnitude: $\mathbf{v}_i(0) = \boldsymbol{\sigma} v_0$, where $\boldsymbol{\sigma} = (\cos\theta, \sin\theta)$ and θ is a random angle uniformly distributed between 0 and 2π.

   IC2: Random direction and magnitude: $\mathbf{v}_i(0) = 2q\,\boldsymbol{\sigma} v_0$, where $\boldsymbol{\sigma} = (\cos\theta, \sin\theta)$, θ is a random angle between 0 and 2π, and q is a random number uniformly distributed in $[0, 1)$.

   IC3: As in IC1 or IC2 with all initial velocities having positive components in the y-direction.
5. Compute and store fictitious "previous" positions using $\mathbf{r}_i(-\Delta t) = \mathbf{r}_i(0) - \mathbf{v}_i(0)\,\Delta t$.

[Those of you interested in object-oriented programming may wish to define a "molecule" class to store the data particular to each molecule as well as manipulate that data. One could also have a "system" class which contains an array of "molecules". MATLAB has useful documentation starting with the "Object-Oriented Programming with MATLAB" page, which may be found by searching the MATLAB help pages. Other languages such as C++ and python are even better options for object-oriented programming. This is entirely optional and recommended only for those interested in using or teaching themselves object-oriented programming.]

Programming Task:

Now we implement the updating procedure. This requires iterating through the N molecules twice.

1. First iteration through molecules:

(a) For molecule i, loop through all other molecules, labelled by j, say. Compute displacement $\mathbf{r}$ between the two molecules, *taking into account the periodic boundary conditions* as described above.
(b) Compute $\mathbf{F}_{ij}$ and add to net force on molecule i.
(c) After finding the net force $\mathbf{F}_i = \sum_j \mathbf{F}_{ij}$ (and hence the acceleration $\mathbf{a}_i$), compute and store new position $\mathbf{r}_i(t + \Delta t)$ using Eq. (3) *without overwriting the current position.*
(d) Update $\mathbf{v}_i(t)$ using Eq. (3).

2. Second iteration through molecules: relabel current positions as previous, and new positions as current.

Question 2 Why is it important to have a separate iteration where all positions are updated separately from the first iteration? That is, why would it be incorrect to overwrite current positions with updated positions during the first loop through particles?

2 Statistical mechanics

Programming Task: You will want to plot histograms of the molecule speeds and velocity components, as well as make other numerical measurements. Since we cannot work with very large N (think about why this is so), it is not sufficient to look at only one time t, but to average speeds over time intervals. Write code which keeps track of particle speeds over a fixed interval of time, so that a histogram can be made. You should be able to perform the following steps

1. Initialize variables/arrays to store data.
2. For N_{meas} times
   (a) Evolve the system N_{skip} times, that is, over a time interval $N_{\text{skip}}\,\Delta t$.
   (b) Store data such as molecule velocities, kinetic energy, etc.
3. Plot quantities such as kinetic energy, for example, as functions of simulation time (these plots are called "time histories").
4. Average over the N_{meas} measurements, also finding their variances and hence their standard errors.

You should normalize the histograms, since these are estimates of probability distributions. It is sufficient to use simple trapezoidal numerical integration (e.g. the function `trapz` in MATLAB), although one should expect some error due to this approximation.

Question 3 For $N = 16$, $L = 10$, $v_0 = 1$, $\delta r = 0.1$, and $\Delta t = 0.02$, plot (and include in your write-up) the time histories of the kinetic energy per molecule for $(N_{\text{meas}}, N_{\text{skip}}) = (50, 1)$ and for $(N_{\text{meas}}, N_{\text{skip}}) = (50, 50)$. Do this for both cases IC1 and IC2 for initial molecule velocities. Comment on the short-time and long-time behaviour of this quantity and on the role of the different initial conditions. You may wish to try other values for $(N_{\text{meas}}, N_{\text{skip}})$ to check your understanding.

Question 4 Produce and include normalized histograms of molecule speed as well as for x- and y-velocity components as described in the above programming task, and for the parameters in the previous question. Discuss the short- and long-time behaviour of these histograms. You may want to look at distributions for longer and/or shorter simulations as well to inform your discussion.

Question 5 How do things change when you start with case IC3 for the initial molecule velocities?

Question 6 Determine the mean kinetic energy per molecule $\langle E \rangle / N$ for all 3 initial conditions IC1-IC3. You may alter Δt, N_{skip}, and N_{meas} in order to get the best estimate, provided your code runs in a few minutes. Why might you want to increase or decrease Δt? N_{skip}? N_{meas}?

Question 7 Having determined the mean kinetic energy, use the equipartition theorem to infer the temperature kT of the gas.

Question 8 How do your histograms of molecular speeds compare with what should be expected from basic statistical physics, i.e. from the Maxwell distribution?

3 Equation of state

The 2-dimensional "volume" of the gas is fixed to be L^2, and from work in the preceding section we can infer the temperature of the gas. The last themodynamic quantity of interest is the pressure P. Here we will make use of the *virial theorem* which, in this case, says

$$2PL^2 = 2\langle E \rangle + \left\langle \sum_{i<j} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \right\rangle \tag{4}$$

where E is the kinetic energy. (The last term is the defined to be "the virial" up to a numerical factor.)

Programming Task:

1. Modify your code to be able to compute $\langle \sum \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \rangle$, and hence the pressure.
2. Be prepared to run your code for several values of temperature, which you can vary by changing v_0.
3. For each simulation, be sure that your numerical measurements are made only after thermal equilibrium is reached. That is, for each new v_0 you should find a sufficiently large MD evolution time T_{eq} such that for simulation times greater than T_{eq}, time-histories of observables fluctuate about a mean.

Question 9 Using at least 5 values of v_0 (but keep $v_0 < 2$) make plots of $\langle \sum \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \rangle$ and pressure vs. temperature. Comment on your findings.

12 Nonlinear Dynamics/Dynamical Systems

12.3 The Lorenz Equations (10 units)

Some familiarity with the Part II course Dynamical Systems would be helpful for this project, which is concerned with bifurcations and chaos in ordinary differential equations.

1 The Lorenz equations

The Lorenz equations are named after the meteorologist who first studied them in 1963:

$$\begin{aligned} \dot{x}(t) &= f_1(x,y,z) &&= 10(y-x)\ , \\ \dot{y}(t) &= f_2(x,y,z) &&= rx - y - xz\ , \\ \dot{z}(t) &= f_3(x,y,z) &&= xy - 8z/3\ . \end{aligned}$$

Question 1 Integrate the equations for values of $r = 0$, 15, 21 and 29. Use $x = y = 1$, $z = r - 1$ as the initial conditions. You may use any standard integrating packages that are available and enable you to choose an appropriate step-length and then fix on it*; comment however on the effect of changing the step-length and why you chose your particular value. You should plot $x(t)$ against $z(t)$ to show your results. For some of the above values of r consider plotting the solution only for $t > T$ for some time $T > 0$; why can this be useful?

A stationary point is a point (x, y, z) where $\dot{x} = \dot{y} = \dot{z} = 0$. It is (locally) stable if all the eigenvalues of the Jacobian matrix $Df(x) = (\partial f_i/\partial x_j)_{1\leqslant i,j\leqslant 3}$ evaluated at the stationary point have negative real part.)

Question 2 Investigate analytically the existence and stability of stationary points of the flow. How do these results relate to the behaviour observed in question 1 above?

2 The strange attractor

The persistent erratic non-periodic oscillations seen when $r = 27$ are due to the existence of a "strange attractor" in the flow. (The existence of this attractor was discovered numerically by Lorenz but there is still no completely rigorous proof that it exists and has the properties we are about to study). This attractor is stable for (approximately) $r > 24.06$, but for $r < 24.06$ some solutions spend a long time wandering about near it before eventually tending towards a stable stationary point. (It exists, but is unstable, for approximately $13.9236 < r < 24.06$.) This phenomenon is known as *intermittency*.

Question 3 For various initial conditions as given in the following list, plot $x(t)$ against t at r-values of your choice in $23 < r < 25$: in each case include in your write-up one or two plots showing the different possible behaviours.

(i) Start very close to the origin $(0, 0, 0)$ but not on the z-axis (why not?).

(ii) Start very close to one of the other fixed points.

*For example you can download a suitable solver from http://www.mathworks.com/matlabcentral/answers/98293

(iii) Start near $x = y = 1$, $z = r - 1$.

Which type of initial condition is best suited for deciding the r-value at which the strange attractor becomes attracting? Which is useful to confirm your stability analysis for the stationary points obtained in question 2 above? Which best illustrates intermittency?

Question 4 For initial conditions which produce trajectories displaying intermittent behaviour in $r < 24.06$, plot the time spent wandering erratically before the trajectory spirals steadily into one of the stationary points against $(24.06 - r)$. You should decide on some criterion for deciding the time t_c at which the solution you are calculating starts heading towards a stable stationary point, and calculate the average of the values of t_c obtained for 5 different initial conditions (all of which should display several "erratic" oscillations before t_c is reached) at each r-value. Explain how you determine t_c.

You will find that nearby initial conditions sometimes give very different values of t_c, and as r increases towards 24.06 you may find that it becomes increasingly difficult to find t_c for all of your chosen initial conditions; you should start with $r = 20$ and be prepared to stop increasing r when the amount of machine time used becomes excessive.

Question 5 Suggest a formula for the way in which the average t_c value increases with r. You will need a fairly large sample of t_c values to make a reasonable estimate.

Question 6 For $r = 27$, write a program to record the successive z-values z_1, z_2, z_3, ... at which a trajectory achieves a local maximum in z. Plot these on a scatter diagram of z_{n+1} against z_n; include also for reference the diagonal line $z_{n+1} = z_n$. What property do portions of the trajectory which generate high points (large values of z_{n+1}) on this diagram have? Does the information that the origin $(0, 0, 0)$ is actually part of the strange attractor help you to find a numerical method to compute (approximately) the largest value of z_{n+1} that could appear on this diagram? If so, do it and add an appropriate point to your figure.

You should not plot the first few points obtained from any given trajectory in order to give any transient behaviour time to die out. You may generate points from many trajectories or from one long trajectory. You will observe that the points on this scatter diagram all lie very near to a certain curve C, which can therefore be used as a predictor for the successive z_i values.

Question 7 Describe in some detail the chief features of this curve and how they relate to your numerical solutions. In particular, consider the following points:

- Where are the intersections of C with the diagonal?
- Does C intersect the diagonal at $z = r - 1$?
- On your diagram, is it possible to draw a square whose top-right and bottom-left corners lie on the diagonal, whose top side touches the peak of C, whose bottom-right corner lies on C, but whose bottom edge does not otherwise intersect C?

Question 8 On a copy of your diagram, draw an approximation to the curve C and use this hand-drawn curve (which you should include in your write-up) to predict a succession of z_i values. For how many steps does your prediction agree well with an actual sequence produced by the numerically computed trajectory? Are there any features of the curve which would lead you to expect this result?

3 The effect of varying r

Question 9 How does the curve C vary as r decreases? Draw the curves for $r = 24.3$ and 22.9, extending them in a sensible way to $z = r - 1$. (For $r = 22.9$ you will need to use initial conditions which give intermittent trajectories in order to generate much of the curve). Describe how the features of the curve change, and explain how these changes relate to the other aspects of behaviour studied in this project.

References

[1] Colin Sparrow *The Lorenz equations: bifurcations, chaos, and strange attractors.* Springer, 1982.

12 Nonlinear Dynamics & Dynamical Systems

12.6 Chaos and Shadowing (10 units)

Familiarity with the Part II Dynamical Systems course would be very helpful for this project, which is concerned with the behaviour of nonlinear maps and uses concepts and tools from nonlinear dynamics.

1 Introduction: dynamical systems, chaos and shadowing

This project considers issues that arise in the numerical solution of dynamical systems which display complicated 'chaotic' behaviour. We first consider the discrete-time case, defining Lyapunov exponents which measure the rate at which nearby points separate under iteration. Then we discuss how 'noisy' trajectories of an iterated map, where the 'noise' arises through numerical errors, are actually close to true trajectories of the system - this property is known as 'shadowing'. Finally the project considers a continous-time (ODE) example of complicated motion motivated by celestial mechanics.

Let D be a closed bounded subset of $\mathbb{R}^m$, and let $F(\mathbf{x})$ be a continuously differentiable map from D to itself. A major task in dynamical systems is to characterise the behaviour of points under repeated iteration of the map F. We call the sequence of points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ constructed by setting $\mathbf{x}_{n+1} = F(\mathbf{x}_n)$ the *trajectory* from the initial condition $\mathbf{x}_0$. The standard notation for the repeated composition of F is to let F^n denote the n-fold composition of F with itself, i.e. $\mathbf{x}_n = F(\mathbf{x}_{n-1}) = F^2(\mathbf{x}_{n-2}) = \cdots = F^n(\mathbf{x}_0)$.

In many situations the rate at which nearby trajectories separate from each other is of interest. This can be characterised by the Lyapunov exponents $\lambda(\mathbf{x}_0, \mathbf{v})$, defined to be the asymptotic rate of divergence of trajectories with initial conditions $\mathbf{x}_0$ and $\mathbf{x}_0 + \mathbf{v}$, where $\mathbf{v}$ is a small perturbation from $\mathbf{x}_0$:

$$\lambda(\mathbf{x}_0, \mathbf{v}) = \lim_{n\to\infty} \lim_{\epsilon\to 0} \frac{1}{n} \log \frac{\|F^n(\mathbf{x}_0 + \epsilon\mathbf{v}) - F^n(\mathbf{x}_0)\|}{\|\epsilon\mathbf{v}\|} \tag{1}$$

Under the conditions given above it can be shown that the limit exists. For a given $\mathbf{x}_0$ there will in general be m (possibly non-distinct) values of $\lambda(\mathbf{x}_0, \mathbf{v})$ as we choose different vectors $\mathbf{v}$ – divergence occurs at different rates in directions corresponding to the different eigenvectors of the Jacobian matrix DF evaluated at $\mathbf{x}_0$. The formula above for $\lambda(\mathbf{x}_0, \mathbf{v})$ will give the largest positive Lyapunov exponent of the system for almost all choices of the vector $\mathbf{v}$. We denote the largest positive Lyapunov exponent by $\lambda(\mathbf{x}_0)$, or simply by λ. If $\mathbf{x}_0$ is a fixed point then the Lyapunov exponents are simply the (real parts of the) Floquet multipliers, so in some sense the idea of a Lyapunov exponent developed above is a generalisation of the idea of a Floquet multiplier to arbitrary trajectories.

For the purposes of this project we will define a map to be chaotic if it appears that $\lambda(\mathbf{x}_0) > 0$ for almost all $\mathbf{x}_0$, so that in general neighbouring points will separate exponentially.

1.1 A map on a square

Here we consider a 2-dimensional (area preserving) map on the unit square $(x, y) \in [0, 1]^2$. Given some initial condition (x_0, y_0), we define

$$x_{n+1} = x_n + \frac{K}{2\pi} \sin(2\pi y_n) \pmod 1 \tag{2}$$

$$y_{n+1} = y_n + x_{n+1} \pmod 1 \tag{3}$$

Note that here (mod 1) means that the map is restricted to the unit square.

Question 1 For $K = 3$, generate a set of double precision pairs (x_n, y_n), $1 \leqslant n \leqslant 1000$, for a range of choices of (x_0, y_0). Some suggested choices of initial conditions are : (0.5,0.5); (10^{-8}, 0); (0.1,0.5); (0.8,0.6); (0.3521,0.424).

Plot the distribution of your points (x_n, y_n). Describe the structure of the phase portrait. What are the fixed points of the map? Describe how the different features of the map change with K.

1.2 Local Chaos

In contrast to the asymptotic quantity $\lambda(\mathbf{x}_0)$ as defined above, a possibly more useful quantity is the local Lyapunov exponent, $\lambda_l(\mathbf{x}_0)$, defined as

$$\lambda_l(\mathbf{x}_0) = \lim_{\mathbf{\Delta} \to \mathbf{0}} \frac{1}{N} \sum_{n=0}^{N} \log \frac{\|\mathbf{x}_{n+1} - \mathbf{x}_{n+1}^{(\mathbf{\Delta})}\|}{\|\mathbf{x}_n - \mathbf{x}_n^{(\mathbf{\Delta})}\|} \tag{4}$$

where $\mathbf{x}_n$ is the n^{th} iterate of $\mathbf{x}_0$, $\mathbf{x}_n^{(\mathbf{\Delta})}$ is the n^{th} iterate of $\mathbf{x}_0 + \mathbf{\Delta}$, and N is a suitable finite number of iterations of the map. For infinitesimal perturbations $\mathbf{\Delta}$, $\lambda_l(\mathbf{x}_0) > 0$ indicates a local expansion of trajectories starting near $\mathbf{x}_0$.

Note that N should be chosen neither too small, nor too large. In practice, you might also want to discard the first few terms of this sum in your numerical calculations.

Question 2 For $K = 3$, find the maximum local Lyapunov exponent for different initial conditions (x_0, y_0) and small perturbation $\|\mathbf{\Delta}\| \ll 1$, using the Euclidean norm in equation (4). What value of N did you use? How did you decide? What is your estimate of the global maximum Lyapunov exponent?

Question 3 We can define a different "Lyapunov exponent", $\lambda_2 = \log_2 e^{\lambda_l}$. Why, when doing binary arithmetic, might λ_2 be more interesting than λ? What is your interpretation of the information λ_2 provides? Given that the calculations here are done to 16 significant figures (or to whatever precision achieved by the code you have used), what would you expect the number of iterations required to be before the results obtained become meaningless? How does your answer compare with what you found numerically?

1.3 Shadowing

Numerical calculations introduce round-off and truncation errors into iteration. For chaotic maps, such as the 2D map above this introduces an effective error at each iteration; this is in some sense equivalent to the explicit perturbation in the initial conditions we considered above.

Since a large class of interesting problems is reducible to iterating nonlinear, and chaotic, maps, it is of some interest to consider whether any numerical calculation can be said to follow the "true" trajectory of such systems.

Here we will consider the simple example used above, assuming the "true" trajectory is given by a double precision calculation of the trajectory, while a single precision calculation provides a "noisy" trajectory.

For some nonlinear systems it is possible to define a "shadow" trajectory to a noisy trajectory (obtained by adding a small perturbation), such that the shadow trajectory is a "true" trajectory of the system, and the "shadow distance" (initially the perturbation) of the shadow trajectory from the noisy trajectory is bounded ([1], [2], [5]). In 2D when there exists one *unstable* (expanding) direction and one *stable* (contracting) direction, it has been proved that, for sufficiently small perturbations, "shadow" trajectories can exist for arbitrarily long times. In many other systems it is still possible to define a "shadow" trajectory for a finite time.

> **Question 4** Let L_n be the Jacobian matrix of the map at iteration n. Construct and write down explicitly the four elements of the Jacobian matrix of the standard map above.

Define $e_{n+1} = p_{n+1} - f(p_n)$, where e_n is the error iterating the map, f, on the vector p by one step. We want to construct a correction term, Φ_n, such that

$$\tilde{p}_n = p_n + \Phi_n \tag{5}$$

defines a "shadow" orbit of p, i.e. $\{\tilde{p}_n\}$ is a true orbit of the dynamical system.

Solving for Φ, we find:

$$\Phi_{n+1} = f(\tilde{p}_n) - e_{n+1} - f(p_n). \tag{6}$$

For Φ_n small, we can expand $f(\tilde{p}_n)$ to linear order, and

$$\Phi_{n+1} = L_n \Phi_n - e_{n+1}. \tag{7}$$

At each iteration, small perturbations along the contracting direction will decay exponentially forward in time, while small perturbations in the expanding direction will grow exponentially forward in time. The reverse will happen when evolving backwards in time.

We therefore want to find basis vectors u_n, s_n aligned with the directions defining the maximum expansion and contraction of the local volume of phase space at step n. We can construct u_n, s_n by iterating the equations:

$$u_{n+1} = \frac{L_n u_n}{||L_n u_n||} \tag{8}$$

and

$$s_{n+1} = \frac{L_n s_n}{||L_n s_n||}. \tag{9}$$

That is, take some initial u_0, s_0 (eg. $(1/\sqrt{2}, 1/\sqrt{2})$, $(-1/\sqrt{2}, 1/\sqrt{2})$), and a vector $p_0 = (x_0, y_0)$. Iterate u_n forwards, i.e. start with your u_0 and iterate equation (8) forward until it has converged to the local direction of expansion. To construct s_n, do the same, but take the initial s_N for

some finite (not too big nor too small) number of iterations of the map, and iterate s_n backwards to find s_0. You will want $N \gg 1$ and $N \ll N_c$, where N_c was the number of iterations at which the sum in equation (4) needed to be truncated.

This procedure will naturally converge onto the direction of maximum expansion when going forward in time, because the term corresponding to the maximum eigenvalue will become dominant for sufficiently large n. Conversely, when going backwards in time the term corresponding to the smaller eigenvalue will become dominant, because it will be proportional to the inverse of a small quantity.

Clearly, since u_n, s_n span the phase space, we can write

$$\Phi_n = \alpha_n u_n + \beta_n s_n \tag{10}$$

and

$$e_n = \eta_n u_n + \xi_n s_n \tag{11}$$

for some α, β, η, ξ.

Using equation (7) we find

$$\alpha_{n+1} u_{n+1} + \beta_{n+1} s_{n+1} = L_n(\alpha_n u_n + \beta_n s_n) - (\eta_{n+1} u_{n+1} + \xi_{n+1} s_{n+1}). \tag{12}$$

Question 5 Substitute equations (8) and (9) into equation (12) to find a recursion relation for α_n, β_n. As before, solve for the α_n by forward iteration from $n = 0$ and for the β_n by backward iteration from $n = N$ for some suitable, fixed N.

You now have a constructed shadow map of the trajectory.

Question 6 Integrating the standard map in double precision, from some known initial condition, with a known error, show that the shadow map of the erroneous initial conditions follows the true trajectory within some shadow distance. If necessary, iterate the shadowing to get a more closely shadowed orbit.

Now integrate your initial condition with single precision (introducing some, in principle unknown) error per iteration, and construct the corresponding double precision shadow trajectory.

Plot your trajectories and comment. (Finding and plotting such trajectories can be tricky!)

Note that shadowing does not always work. A trivial counter-example is provided by the one dimensional logistic map $f(x) = 1 - 2x^2$, $x \in (-1, 1)$.

Near $x = 0$, no true orbit can shadow general noisy orbits, as noise in $f(x)$ may take the map out of the domain and iterating the subsequent trajectory will take x to $-\infty$.

1.4 Application: the Sitnikov Problem

It is known that the N-body problem, of $N > 2$ bodies moving under their own mutual gravitational attraction only, is chaotic.

Here we consider a well known special case of the restricted three-body problem, where one of the bodies has zero mass. In this particular problem, known as the Sitnikov problem ([4], [3]), the motion of the zero mass is restricted to the z axis, defined by the normal to the plane of motion of the two massive bodies, through the center of mass. The two massive bodies move on Keplerian ellipses with eccentricity $\epsilon \in [0, 1]$ around their centre of mass.

Without loss of generality, we consider the two massive bodies to have masses, $M_1 = M_2 = 1/2$. We are interested in bound motion, with semi-major axis $a = 1$.

The motion of the two massive bodies is uniquely described by their elliptic orbit (the phase is irrelevant to the dynamics we are interested in, by rotational symmetry). The separation of the massive bodies from the center of mass is $r(t) = (1 - \epsilon \cos t) + O(\epsilon^2)$, .

We want to consider the motion of the third, zero mass body on the z-axis. Define $v = dz/dt$, then

$$\frac{dv}{dt} = -\frac{z}{(z^2+1)^{3/2}} - \frac{3z\epsilon\cos(t+t_0)}{(z^2+1)^{5/2}}. \tag{13}$$

The equations of motion may be integrated numerically using a high order integrator, such as the Runge–Kutta scheme, given some initial conditions. Without loss of generality, we choose initial conditions $t_0 = 0, z(0) = 0$, $v(0) = v_0$.

> **Question 7** Write down the energy of the third mass, i.e. $\lim_{m\to 0}(E/m)$ and solve for the critical velocity, v_c, for which the energy is zero. Write down $z(t)$ for $\epsilon = 0$. Write down the Jacobian matrix of this map.
>
> It is useful to define the initial velocity as some multiple of v_c. We are interested in (initially) bound motion, so $v_0 \leqslant v_c$.
>
> For $\epsilon = 0.03, 0.04, 0.05$ and $v_0/v_c = 0.92, 0.94, 0.96$, plot $z(t)$ vs t. Comment.

In continuous time we can define a (maximum) Lyapunov exponent exactly analogous to the discrete-time case:

$$\lambda(\mathbf{z}_0) = \lim_{t\to\infty}\lim_{\epsilon\to 0}\frac{1}{t}\log\frac{\|\phi_t(\mathbf{z}_0+\epsilon\mathbf{w}) - \phi_t(\mathbf{z}_0)\|}{\|\epsilon\mathbf{w}\|} \tag{14}$$

for almost all choices of perturbation $\mathbf{w}$, where $\mathbf{z} = (z, \dot{z})$ and ϕ_t denotes the evolution operator defined by integrating the ODEs forwards in time.

> **Question 8** As before, construct a trajectory in $(z, \dot{z})$ space with an initial "error", δ, and integrate the true and erroneous trajectories for a chosen value of $v_0 \approx 0.95 v_c$.
>
> Estimate numerically the Lyapunov exponent of the mapping for the different cases. Is the motion chaotic?

> **Question 9** Using the method discussed in the previous section, construct a shadow trajectory for the zero mass body, and compare "true" trajectories integrated with double precision arithmetic, with the corresponding "shadow" trajectories integrated from the same initial condition with single precision arithmetic.
>
> Comment on the integrability of the N-body problem. Do you think numerical integrations of N-body systems are reliable – or can be made reliable – in some sense?

References

[1] Bowen, R, 1975 *J. Diff. Eqns.*, 18, 333.

[2] Grebogi, C., Hammel, S.M., Yorke, J.A., Sauer, T., 1990 *Phys. Rev. Lett.*, 65, 1527

[3] Liu, J., Sun, Y.-S., 1990 *Cel. Mech. and Dyn. Astro.*, 49, 285.

[4] Marchal, C., 1990 *The Three-Body Problem*, Elsevier (Oxford).

[5] Quinlan, G.D., Tremaine, S., 1992 *MNRAS*, 259, 505.

14 General Relativity

14.5 Cosmological distances (8 units)

Although this project is based on general relativistic cosmology, no detailed knowledge of General Relativity is required. All relevant equations are defined and explained in the project itself.

1 Introduction

In cosmology there are many ways to specify the distance between two points because, in the expanding Universe, the distances between objects are changing and Earth-bound observers look back in time as they look out in distance. All these distances measure the separation between events on radial null trajectories, trajectories of photons which terminate at the observer.

The metric for a homogeneous isotropic universe, in spherical polar coordinates for the spatial part, is

$$ds^2 = c^2dt^2 - R(t)^2\left[\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + \sin^2\theta d\phi^2)\right], \tag{1}$$

where, by a suitable choice of radial coordinate r, $k = -1$, 0 or 1 for open, Euclidean or closed geometries.

In this metric the redshift relative to an observer at the spatial origin is given by

$$1 + z = R(t_0)/R(t_1), \tag{2}$$

where t_0 is the coordinate time at which the photon is received and t_1 that at which it was emitted. Thus, for a given observer, the redshift depends only on the radial scale factor of the Universe at the time the photon was emitted divided by its value at the observer's time. The redshift is important because it can be measured easily from the observed wavelengths of atomic transition lines with known rest wavelengths.

When the matter density at time t is ρ and the pressure is zero one of the Einstein field equations with the cosmological term becomes

$$\frac{\dot{R}^2}{R^2} + \frac{kc^2}{R^2} - \frac{\Lambda c^2}{3} = \frac{8\pi G}{3}\rho, \tag{3}$$

where $\dot{R} = \frac{dR}{dt}$ and Λ is a constant. The other field equation can be combined with this to give the conservation of matter equation

$$\rho R^3 = \text{const.} \tag{4}$$

For small distances the redshift $cz = H_0 d$, where d is the distance to the source. Then H_0, the Hubble constant, gives the local expansion rate. It is often written in the form $H_0 = 100h\,\mathrm{km\,s^{-1}\,Mpc^{-1}} = 3.2409 \times 10^{-18} h\,\mathrm{s^{-1}}$, where h is dimensionless. The actual value of h is still uncertain, and hotly debated, but most would agree on measurements of 0.72 ± 0.08. The megaparsec is an astronomical length unit appropriate for separations between galaxies, $1\,\mathrm{Mpc} = 3.0856 \times 10^{22}\,\mathrm{m}$. The Hubble time $t_\mathrm{H} = 1/H_0 = 3.0856 \times 10^{17} h^{-1}\,\mathrm{s}$ and the Hubble distance $D_\mathrm{H} = c/H_0 = 9.26 \times 10^{25} h^{-1}\,\mathrm{m}$. Take the number of seconds in one year to be 3.1556926×10^7 s.

Our Universe can be described by two parameters, the matter density now ρ_0 and the cosmological constant Λ, and we can express these in a dimensionless form using H_0 as

$$\Omega_\mathrm{m} \equiv \frac{8\pi G\rho_0}{3H_0^2} \tag{5}$$

and

$$\Omega_\Lambda \equiv \frac{\Lambda c^2}{3H_0^2}. \tag{6}$$

By means of equation (3), at time t_0, the curvature value k can be parameterised by Ω_k so that

$$\Omega_\mathrm{m} + \Omega_\Lambda + \Omega_k = 1. \tag{7}$$

Then the function

$$E(z) = \sqrt{\Omega_\mathrm{m}(1+z)^3 + \Omega_k(1+z)^2 + \Omega_\Lambda}$$

is proportional to the time derivative of the logarithm of the scale factor, $\dot{R}/R$, at redshift z (see *e.g.* Peebles 1993, pp 310 – 321).

Where specific values are required in what follows you should take $H_0 = 72\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$.

2 Lookback Time

The lookback time t_L is the difference between the age t_0 of the Universe now and the age t_e when the photons were emitted

$$t_\mathrm{L} = t_\mathrm{H} \int_0^z \frac{dz'}{(1+z')E(z')}. \tag{8}$$

Question 1 If $\Omega_\mathrm{m} = 1$ and $\Omega_\Lambda = 0$, obtain an expression for the lookback time to an object with redshift z and show that the age of the Universe is $t_\mathrm{L}(z=\infty) = \frac{2}{3}t_\mathrm{H}$.

Question 2 Write a program to determine the lookback time in Gyr for general H_0, Ω_m and Ω_Λ. If $H_0 = 72\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$, tabulate the lookback time to $z = 0.1$, 1.0, 2.0, 4.0 and 6.7 (one of the highest individual object redshifts measured so far) for

(1) an Einstein-de-Sitter universe $\Omega_\mathrm{m} = 1$, $\Omega_\Lambda = 0$,

(2) a classical closed universe $\Omega_\mathrm{m} = 2$, $\Omega_\Lambda = 0$,

(3) a baryon dominated low density universe $\Omega_\mathrm{m} = 0.04$, $\Omega_\Lambda = 0$ and

(4) the currently popular Universe $\Omega_\mathrm{m} = 0.27$, $\Omega_\Lambda = 0.73$.

What is the age of the Universe for each of these models [to the nearest 100 million years]?

Produce a graph showing lookback time against redshift for the four models and comment on any overall trends.

3 Distance Measures

There are three useful ways to define distance.

(1) The line of sight comoving distance

$$D_\mathrm{C} = D_\mathrm{H} \int_0^z \frac{dz'}{E(z')}. \tag{9}$$

(2) The angular diameter distance is the ratio of an object's physical size to its angular size (in radians). For an object of size ℓ at redshift z the angular size is $\theta = \ell/D_\mathrm{A}$, where θ is a small angle (so $\sin\theta \approx \tan\theta \approx \theta$).

$$D_{\rm A} = \begin{cases} D_{\rm H}\frac{1}{\sqrt{\Omega_k}(1+z)}\sinh\left[\sqrt{\Omega_k}D_{\rm C}/D_{\rm H}\right], & \text{for } \Omega_k > 0, \\ D_{\rm C}/(1+z), & \text{for } \Omega_k = 0, \\ D_{\rm H}\frac{1}{\sqrt{|\Omega_k|}(1+z)}\sin\left[\sqrt{|\Omega_k|}D_{\rm C}/D_{\rm H}\right], & \text{for } \Omega_k < 0. \end{cases} \tag{10}$$

(3) The luminosity distance $D_{\rm L}$ is defined by the relationship between the observed photon energy flux f, integrated over all frequencies, and the intrinsic energy output from the source L by

$$f = \frac{L}{4\pi D_{\rm L}^2}.$$

It is related to the angular diameter distance by

$$D_{\rm L} = (1+z)^2 D_{\rm A}. \tag{11}$$

Question 3 Obtain an analytic expression for the angular diameter distance $D_{\rm A}$ as a function of redshift in the case where $\Omega_{\rm m} = 1$ and $\Omega_\Lambda = 0$ and show that it has a maximum value when $z = 1.25$.

Question 4 Write a program to determine the luminosity and angular diameter distances given the redshift z and plot the dimensionless values $D_{\rm A}/D_{\rm H}$ and $D_{\rm L}/D_{\rm H}$ for redshifts $0 < z < 7$ for $(\Omega_{\rm m}, \Omega_\Lambda) = (1, 0)$, $(0.04, 0)$ and $(0.27, 0.73)$. For these three cases tabulate the values at redshifts $z = 1$, 1.25, 2.0 and 4.0.

4 Comoving Volume

The comoving volume $V_{\rm C}$ is the volume measure in which the number density of non-evolving objects is constant with redshift. The comoving volume element in solid angle $\sin\theta d\theta d\phi$ and redshift interval dz is

$$dV_{\rm C} = D_{\rm H}\frac{(1+z)^2 D_{\rm A}^2}{E(z)}\ \sin\theta\, dz\, d\theta\, d\phi.$$

Integrating this from the present to redshift z gives the total comoving volume over the whole sky to redshift z,

$$V = \frac{4\pi}{3}\frac{D_{\rm L}^3}{(1+z)^3} = \frac{4\pi}{3}D_{\rm C}^3 \qquad \text{for} \quad \Omega_k = 0. \tag{12}$$

A method to test whether a sample of objects has a uniform comoving density and luminosity which does not change with cosmic time is to use the $< V/V_{\rm max} >$ test. It is assumed that all objects with observed flux $f > f_0$ are detected and the observed flux f and the redshift z are measured for each object. For a given luminosity L there is a maximum redshift $z_{\rm max}(L)$ at which the observed flux is f_0 so that the object is just included. Corresponding to this redshift is a maximum volume $V_{\rm max}(L)$. Then, if we have a distribution of luminosities so that $\Phi(L)dL$ is the number per unit comoving volume with luminosity between L and $L + dL$, the total number of objects in the sample is

$$\int_0^\infty \Phi(L) \int_0^{V_{\rm max}(L)} dV dL.$$

where V is the comoving volume.

Question 5 Show that for a uniform comoving distribution of objects the expectation value $\langle V/V_{max}\rangle = \frac{1}{2}$.

Question 6 Write a program to read pairs of numbers z and f/f_0, determine V and $V_{\rm max}$ for these for Universe models for which $\Omega_k = 0$ and determine the average value $< V/V_{\rm max} >$ for each model.

Verify that for small values of z the program gives the Euclidean limit, for individual cases $V/V_{\rm max} \propto (f/f_0)^{-\frac{3}{2}}$.

Apply the program to the sample, listed below, of 114 quasars from an area of sky. What is the value of $< V/V_{\rm max} >$ for this sample if $\Omega_m = 0.27$ and $\Omega_\Lambda = 0.73$? Is the value of $< V/V_{\rm max} >$ what you would expect from a constant comoving population? How might you interpret the result you obtain?

Question 7 The sample in the previous question was also subject to the constraints $z > 0.20$ and $z < 3.0$, because it is only in this range that an object be recognised as a quasar. How would you modify the $V/V_{\rm max}$ quantity so that for a uniform distribution in this redshift range the average value is still $\frac{1}{2}$? What is the result of using this on the sample of 114 quasars?

References

[1] Peebles, P. J. E., 1993, *Principles of Physical Cosmology*, Princeton University Press.

Quasar data. The following may also be found in the file `quasar.dat` in the `data` directory on the CATAM website:

z	f/f_0	z	f/f_0	z	f/f_0	z	f/f_0	z	f/f_0	z	f/f_0
0.202	1.570	0.217	3.250	0.225	2.884	0.237	3.630	0.246	1.213	0.259	1.330
0.274	1.614	0.298	1.330	0.315	2.032	0.322	1.066	0.332	1.976	0.351	1.018
0.362	1.096	0.373	1.191	0.385	2.937	0.402	2.355	0.433	1.853	0.449	4.168
0.460	5.105	0.479	1.706	0.492	1.629	0.507	1.940	0.530	1.472	0.549	1.419
0.571	2.511	0.582	2.089	0.590	1.599	0.609	1.406	0.624	1.018	0.641	1.018
0.659	3.564	0.672	2.511	0.679	2.013	0.692	1.294	0.714	1.294	0.723	1.584
0.737	1.342	0.754	1.076	0.774	1.753	0.781	2.128	0.791	2.779	0.803	2.421
0.832	1.106	0.847	1.527	0.874	1.158	0.892	1.202	0.913	2.167	0.934	1.629
0.955	1.887	0.973	2.208	0.993	2.558	1.012	1.355	1.025	1.247	1.040	1.318
1.056	1.213	1.072	1.803	1.092	1.330	1.115	1.342	1.140	1.086	1.152	1.180
1.182	1.180	1.205	1.393	1.220	1.247	1.234	1.342	1.247	2.535	1.263	1.047
1.288	1.541	1.313	1.028	1.332	1.037	1.343	1.235	1.376	2.779	1.388	1.202
1.400	1.086	1.440	1.127	1.455	1.009	1.469	1.056	1.487	1.330	1.511	1.330
1.543	1.028	1.559	1.202	1.583	1.819	1.593	1.294	1.619	1.614	1.641	3.047
1.664	1.393	1.684	1.158	1.700	1.513	1.727	1.629	1.756	1.137	1.776	1.355
1.810	2.831	1.844	1.018	1.878	2.208	1.913	2.606	1.941	1.342	1.961	1.555
1.976	2.910	2.005	1.106	2.035	1.306	2.075	1.887	2.092	1.958	2.106	1.367
2.134	1.406	2.187	2.089	2.244	1.202	2.297	1.106	2.329	1.009	2.388	1.737
2.442	1.644	2.523	1.000	2.595	1.393	2.649	1.282	2.786	1.527	2.936	1.445

14 General Relativity

14.6 Isolating Integrals for Geodesic Motion (8 units)

This project assumes material taught in the Part II course General Relativity. Some of the calculations may be done more simply using a Computer Algebra System (CAS) such as Mathematica or Maple. Throughout we use geometrical units with $c = G = 1$.

1 Geodesic Motion in Axisymmetric Spacetimes

A general axisymmetric metric can be written in the form

$$\mathrm{d}s^2 = g_{tt}\mathrm{d}t^2 + 2g_{t\phi}\mathrm{d}t\mathrm{d}\phi + g_{\phi\phi}\mathrm{d}\phi^2 + g_{rr}\mathrm{d}r^2 + g_{\theta\theta}\mathrm{d}\theta^2 \tag{1}$$

where the metric components are functions of the coordinates r and θ only.

Question 1 By considering the Euler-Lagrange equations for t and ϕ of the geodesic action

$$\mathcal{S} = \int \sqrt{g_{ij}\frac{\mathrm{d}x^i}{\mathrm{d}\tau}\frac{\mathrm{d}x^j}{\mathrm{d}\tau}}\,\mathrm{d}\tau \tag{2}$$

where the affine parameter τ is the proper time along the geodesic, or otherwise, show that

$$\begin{aligned} E &= g_{tt}\frac{\mathrm{d}t}{\mathrm{d}\tau} + g_{t\phi}\frac{\mathrm{d}\phi}{\mathrm{d}\tau} \\ L_z &= -\left(g_{t\phi}\frac{\mathrm{d}t}{\mathrm{d}\tau} + g_{\phi\phi}\frac{\mathrm{d}\phi}{\mathrm{d}\tau}\right) \end{aligned} \tag{3}$$

are constants of geodesic motion in any axisymmetric spacetime (1). Hence, derive the mass conservation integral

$$g_{rr}\left(\frac{\mathrm{d}r}{\mathrm{d}\tau}\right)^2 + g_{\theta\theta}\left(\frac{\mathrm{d}\theta}{\mathrm{d}\tau}\right)^2 = -V_{\text{eff}}(r, \theta, E, L_z) \tag{4}$$

where the *effective potential*, V_{eff}, should be found in terms of E, L_z and the metric components.

The effective potential defines the allowed regions of geodesic motion for a particular choice of the energy, E, and angular momentum, L_z. Motion is only possible where $V_{\text{eff}} \geqslant 0$.

The Kerr metric has components

$$\begin{aligned} &g_{tt} = 1 - \frac{2\,m\,r}{\Sigma}, \qquad g_{t\phi} = \frac{2\,a\,m\,r\,\sin^2\theta}{\Sigma}, \qquad g_{\phi\phi} = -\left(\Delta + \frac{2\,m\,r\,(r^2+a^2)}{\Sigma}\right)\sin^2\theta, \\ &g_{rr} = -\frac{\Sigma}{\Delta}, \qquad g_{\theta\theta} = -\Sigma \end{aligned} \tag{5}$$

where $\Sigma = r^2 + a^2\,\cos^2\theta$, $\Delta = r^2 - 2\,m\,r + a^2$, m is a constant (the mass of the black hole) and a is another constant (the spin parameter of the black hole).

Question 2 Using a CAS, or otherwise, derive expressions for the Christoffel symbols

$$\Gamma^i_{jk} = \frac{1}{2}\, g^{im}\left(g_{jm,k} + g_{km,j} - g_{jk,m}\right) \tag{6}$$

for the Kerr metric (5). You can present these results in your write-up in the form of a printout from a CAS worksheet.

Programming Task: Write a program to numerically integrate the second order timelike geodesic equations

$$\frac{\mathrm{d}^2 x^i}{\mathrm{d}\tau^2} = -\Gamma^i_{jk}\frac{\mathrm{d}x^j}{\mathrm{d}\tau}\frac{\mathrm{d}x^k}{\mathrm{d}\tau} \tag{7}$$

for the Kerr metric (5). Do not make use of the first integrals derived above (3)–(4), but write the four second order equations as a set of eight coupled first order equations. We will use the first integrals to verify the numerical accuracy of the integrations.

2 Geodesic motion in the Schwarzschild metric

The Schwarzschild metric may be obtained by setting $a = 0$ in the Kerr metric (5).

Question 3 What are the non-zero Christoffel symbols for the Schwarzschild metric? Take $m = 1$ and find the zeros of the effective potential (4) in the equatorial plane, $\theta = \pi/2$, for the case $E = 0.97$, $L_z = 4$. Hence determine the allowed range of radii, r, of bound orbits in the equatorial plane. Then, using your geodesic code, do the following

a Take initial conditions $r = 15$, $\theta = \pi/2$, $\mathrm{d}r/\mathrm{d}\tau = 0$ and the value of $\mathrm{d}\theta/\mathrm{d}\tau$ determined from the effective potential (4). Plot the coordinates, $(t,\, r,\, \theta,\, \phi)$, of the particle as a function of τ over several orbits. Check that the three conservation laws (3)–(4) are satisfied at a reasonable level of numerical accuracy.

b For the same choice of E and L_z, take a range of initial conditions that lead to bound motion (e.g., consider initial conditions in the equatorial plane with $\mathrm{d}r/\mathrm{d}\tau = 0$ and a range of values of $r(0)$). Output the values of r and $\mathrm{d}r/\mathrm{d}\tau$ every time the orbit crosses the equatorial plane, $\theta = \pi/2$, with $\mathrm{d}\theta/\mathrm{d}\tau > 0$. Plot these values on a graph, with r on the horizontal axis, and $\mathrm{d}r/\mathrm{d}\tau$ on the vertical axis. What do you notice?

c Experiment with a few different values of E, L_z and initial conditions.

You have plotted a Poincaré map for these orbits. If the Poincaré map of an orbit is a closed curve it indicates the possible existence of an extra isolating integral for the motion.

3 Geodesic motion in the Kerr metric

We now consider $a \neq 0$ in the Kerr metric (5).

Question 4 Take $a = 0.9$, $E = 0.95$ and $L_z = 3$, and use the effective potential to find the allowed range of r_0 for which the initial conditions $\theta = \pi/2$, $r = r_0$ and $\mathrm{d}r/\mathrm{d}\tau = 0$ lead to bound motion. Plot a Poincaré map as described above for a range of initial conditions of this type. Is the result similar to what you saw for the Schwarzschild metric?

Question 5 Show that the quantity

$$Q = (aE\sin\theta - L_z\mathrm{cosec}\theta)^2 + (r^2 + a^2\cos^2\theta)^2\left(\frac{\mathrm{d}\theta}{\mathrm{d}\tau}\right)^2 + \delta\, a^2\,\cos^2\theta \tag{8}$$

is conserved for geodesic motion in the Kerr metric, where δ is a numerical constant that should be determined. You may use a CAS to help demonstrate this, but should include evidence of the calculation. What does Q become in the limit $a = 0$, i.e., for the Schwarzschild metric? Provide a physical interpretation if possible.

Programming Note

Maple (as available on the PWF) includes a tensor-manipulation package, which may be accessed using **with(tensor)**. This includes a routine **Christoffel2** which computes the Christoffel symbols for a metric. You may find this useful.

References

[1] Chandrasekhar, S.; *The Mathematical Theory of Black Holes*; Clarendon Press: Oxford; 1992.

[2] D'Inverno, R.; *Introducing Einstein's Relativity*; Clarendon Press: Oxford; 1992.

[3] Goldstein, H., Poole, C. & Safko, J.; *Classical Mechanics*, third edition, Pearson Education International: New Jersey; 2002.

15 Number Theory

15.5 Reduction of Binary Quadratic Forms (10 units)

Background material for §§1–3 is contained in the Part II course Number Theory. A good understanding of this course, or the Part II course Number Fields, is also necessary for the later sections. Alternatively, the books [1]–[4] can be used.

1 Introduction

We start with a *binary quadratic form* $f(x, y) = ax^2 + bxy + cy^2$ with a, b, $c \in \mathbb{Z}$, which we shall abbreviate as (a, b, c). The *discriminant* of the form (a, b, c) is $d = b^2 - 4ac$. Note that d is always congruent to 0 or 1 modulo 4. We consider only *non-trivial* forms, for which d is not a square.

Two forms f, g are *equivalent* if one can be transformed into the other by a *unimodular substitution* M, that is, if $g(x, y) = fM(x, y) = f(sx + ty, ux + vy)$ where s, t, u, $v \in \mathbb{Z}$ and $sv - tu = 1$, i.e.

$$M = \begin{pmatrix} s & t \\ u & v \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}).$$

Equivalent forms have the same discriminant, but the converse is not true in general. We say that f *represents* n if there are integers x, y such that $f(x, y) = n$. A form is *positive definite* if it represents only positive integers (and zero), *negative definite* if it represents only negative integers (and zero) and otherwise *indefinite*. A form is definite if and only if it has negative discriminant.

2 Reduction of definite forms

We define a positive definite form (a, b, c) to be *reduced* if either $-a < b \leqslant a < c$ or $0 \leqslant b \leqslant a = c$. There are only finitely many reduced positive definite forms of given discriminant. It is known that distinct reduced positive definite forms are inequivalent, and that every form is equivalent to a reduced form. The number of equivalence classes of discriminant d is the *class number* $h(d)$ and this is equal to the number of reduced forms (in this positive definite case).

> **Question 1** Find bounds for the coefficients of a reduced positive definite form of given discriminant and use these to write a procedure to list all the reduced positive definite forms with given discriminant d. Find all the reduced forms of discriminant d for $-20 < d < 0$.
>
> Tabulate the class number $h(d)$ for d between 0 and -100. How does the amount of computing time needed grow with d? If you can see any way of significantly reducing the computing time (for large d) then you should go back and make the necessary changes.
>
> How many discriminants $d < 0$ can you find with $h(d) = 1$? Repeat for $h(d) = 2$ and $h(d) = 3$.

We can find the reduced form equivalent to a given positive definite form f by *reduction*. If f is not reduced then $c < a$, or $|b| > a$, or $a = -b$, or $a = c$ and $b < 0$. We define operations S, T and T^{-1} on forms by

$$S\colon (a, b, c) \mapsto (c, -b, a),$$

$$T\colon (a,b,c) \mapsto (a, b+2a, a+b+c),$$
$$T^{-1}\colon (a,b,c) \mapsto (a, b-2a, a-b+c).$$

These operations are represented by matrices $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ and $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ in $\mathrm{SL}_2(\mathbb{Z})$, so that each operation yields an equivalent form. Note that $S = -S^{-1}$. If a form is not reduced then one of these operations may be applied and the result will be "closer" to a reduced form (in the sense that $|a| + |b|$ is made smaller).

Question 2 Write a program to find the reduced form equivalent to a given positive definite form. Your program should read in the coefficients a, b, c and print the reduced form equivalent to (a, b, c) together with the sequence of reduction operations which are needed. Run your program on the forms $(1064, 2629, 1624)$ and $(1885, 2933, 1141)$, and on several randomly chosen positive definite forms.

3 Automorphs of definite forms

An *automorph* of a binary quadratic form f is a unimodular substitution which transforms f into itself. The identity matrix I is an obvious automorph of any form, as is $-I$. The only primitive definite forms ("primitive" means that the coefficients a, b, c have no common factor) with non-trivial automorphs are those of discriminant -3 and -4.

If $M \in \mathrm{SL}_2(\mathbb{Z})$ and f is a positive definite form then reduction of the forms f and fM must give the same answer. An appropriate composition of M and the reduction steps will then give an automorph of f.

Question 3 Write a program to represent any $M \in \mathrm{SL}_2(\mathbb{Z})$ as a product of matrices of the form S, $T^{\pm 1}$ and $-I$, by taking a positive definite form f with only trivial automorphs (for example $(1, 1, 2)$) and reducing the forms f and fM. Run your program for several values of M and for several (not necessarily reduced) choices of f for each M. Comment on the representations of M that you obtain. What happens if you apply the same method with the forms $(1, 0, 1)$ or $(1, 1, 1)$?

4 Reduction of indefinite forms

We now turn to indefinite forms. As mentioned in §1, we only consider forms whose discriminant is not a square. We define the *roots* of a form (a, b, c) to be the roots $z_\pm$ of the associated quadratic equation $az^2 + bz + c$. If a definite form is reduced, then one of its roots will lie in the domain

$$\Delta = \left\{ z : |\operatorname{Re} z| \leqslant \tfrac{1}{2} \text{ and } |z| \geqslant 1 \right\}.$$

The roots $z_\pm$ of an indefinite form are real. We define an indefinite form f to be *reduced* if the circle $C(f)$ with centre on the real line which passes through the two roots also intersects the domain Δ. (Note that some alternative definitions exist; you should use the present one.) It is not obvious that every indefinite form is equivalent to a reduced form, but we prove this by observing that $C(fM) = C(f)M^{-1}$, where M acts on $\mathbb{C}$ as a Möbius transformation, and so it is sufficient to show that one point in $C(f)$ can be brought into Δ by a suitable M. The point

$$w = \frac{-b + i\sqrt{b^2 - 4ac}}{2a}$$

is on $C(f)$ and is a root of the definite form $\big(2a^2, 2ab, b^2 - 2ac\big)$. Thus the indefinite form (a, b, c) is reduced if the definite form $(2a^2, 2ab, b^2 - 2ac)$ is reduced, although not conversely. (Note also in passing that a given indefinite form may have more than one equivalent reduced form, unlike the case of definite forms.)

Question 4 Write a program to reduce the indefinite form (a, b, c) by reducing the definite form $\big(2a^2, 2ab, b^2 - 2ac\big)$ and applying the same operations to (a, b, c). Run your program on some indefinite forms, and investigate whether equivalent forms reduce to the same form or not.

For an indefinite reduced form f, we define the *neighbours* of f to be those of the forms fS, fT or fT^{-1} which are still reduced. Most forms f have exactly two neighbours.

Question 5 Write a program which takes a reduced form and finds all its neighbours, their neighbours and so on. Apply it to the reduced forms you found in Question 4. Comment on the resulting sequence of forms.

Question 6 For each reduced indefinite form found in Question 4, give a non-trivial automorph in $\mathrm{SL}_2(\mathbb{Z})$. What can you say about the group of automorphs of a given reduced indefinite form?

There are only finitely many reduced indefinite forms of given discriminant d.

Question 7 Find bounds for the coefficients of a reduced indefinite form of given discriminant and use these to write a procedure to list all the reduced indefinite forms with given discriminant d.

Write a program which uses this procedure to find all the reduced forms of discriminant d for $0 < d < 50$ with $d \equiv 1 \bmod 4$, d not a square, and which finds the equivalence classes. You may assume that two reduced indefinite forms which are equivalent can be connected by a sequence of neighbours.

Tabulate the class number $h(d)$, the number of equivalence classes of discriminant d, for a larger range of d. How does the amount of computing time needed grow with d?

References

[1] Baker, A. *A concise introduction to the theory of numbers.*

[2] Cassels, J. W. S. *Rational quadratic forms.*

[3] Jones, B. W. *The arithmetic theory of quadratic forms.*

[4] Watson, G. L. *Integral quadratic forms.*

15 Number Theory

15.10 The Continued Fraction Method for Factorization (8 units)

This project is related to material in the Part II course Number Theory.

1 Factor bases

In this project N will be a (usually large) integer, that we would like to factor, and B will be a finite set of (usually small) primes. We call B the factor base. Sometimes it is convenient to allow -1 as an element of B.

> **Question 1** Write a program, using trial division, to test whether N is a product of primes in B (we say that N is B-smooth), and if so to give the prime factorization. Use your program to estimate, for a suitable range of d, the probability that a d-digit integer is B-smooth where B is the set of primes less than 50.

The integers that may be accurately represented in your chosen language may be limited to 10^{15} or similar. For example in MATLAB numbers are represented by default as doubles, meaning that they are stored to 16 significant (decimal) figures. For integers larger than 10^{15} functions such as `mod` and `int2str` may give incorrect answers. If your language is able to handle larger integers, then you are still expected to comment where appropriate on how you would manage if you were restricted to 10^{15}.

On a modern computer it is practical to factor integers of the size considered in this project by trial division. We study a factoring method that remains practical for much larger values of N. To enable comparisons, we therefore make the following artificial restriction: the factor base B is only allowed to contain primes that are less than 50.

2 Continued fractions

The continued fraction algorithm applied to a real number $x_0 = x$ forms a sequence of partial quotients a_n by the transformation

$$\begin{aligned} a_n &= \lfloor x_n \rfloor \\ x_{n+1} &= \frac{1}{x_n - a_n} \end{aligned}$$

where as usual $\lfloor x \rfloor$ denotes the greatest integer $\leqslant x$. The algorithm terminates if $x_n = a_n$; this happens if and only if the initial x is rational.

The convergents P_n/Q_n are defined for $n \geqslant 0$ by

$$\begin{aligned} P_n &= a_n P_{n-1} + P_{n-2} \\ Q_n &= a_n Q_{n-1} + Q_{n-2} \end{aligned}$$

with initial conditions $P_{-2} = 0$, $P_{-1} = 1$ and $Q_{-2} = 1$, $Q_{-1} = 0$.

Question 2 Show that if $x = \sqrt{N}$ for some positive integer N then each x_n may be written in the form $(r + \sqrt{N})/s$ with r, s integers and $s \mid (r^2 - N)$.

Write a program to develop the continued fraction expansion of $\sqrt{N}$. Your program should work with integers as far as possible, so that there is no risk of rounding errors. In some programming languages it is best to use an integer type, but there is no particular advantage in doing this if you are using MATLAB.

Tabulate the partial quotients of $\sqrt{N}$ for $1 \leqslant N \leqslant 50$ and comment on the results. Investigate how large r and s can become (in terms of N).

For a fixed value of the positive integer N, the equation $x^2 - Ny^2 = 1$, in integer unknowns x and y, is called Pell's equation. The negative Pell equation is $x^2 - Ny^2 = -1$.

Question 3 Tabulate the quantities $P_n^2 - NQ_n^2$ for some values of N and hence comment on the use of continued fractions to solve Pell's equation, and the negative Pell equation. Can you see any simple condition on N which ensures that the negative Pell equation is insoluble?

Write a procedure that given $x, y, N \leqslant 10^{15}$ tests whether $x^2 - Ny^2 = \pm 1$, being careful to avoid integer overflow. (Hint: try working mod p for several primes p.)

Use your observations to write a program to find non-trivial solutions to Pell's equation. Tabulate the solutions found for each N in the ranges $1 \leqslant N \leqslant 100$ and $500 \leqslant N \leqslant 550$, when such a solution exists. You may find a few values of N, such as 509, beyond the capacity of your program; you do not need to correct for this. You should however make sure that all answers you do give are correct.

3 Factorization

By "factorization" we mean the task of finding a non-trivial factor of a composite integer N. We will not be concerned with finding the complete factorization of N into primes. You should assume from now on that N is odd.

Question 4 Suppose we are given a supply of integers x, y with $x^2 \equiv y^2 \bmod N$. How might this help us factor N? Estimate the complexity of the steps involved. If N is composite, can we be sure that suitable x and y exist?

One source of integers x, y with $x^2 \equiv y^2 \bmod N$ arises from the convergents of the continued fraction expansion of $\sqrt{N}$.

Question 5 Modify your programs to compute $P_n \bmod N$ and $P_n^2 \bmod N$ for N up to 10^{10}. Explain how you avoid integer overflow. (Hint: a multiplication mod N can be done in two pieces.) Run your program for $N = 2012449237,\ 2575992413$ ans 3548710699.

Question 6 Let A be a matrix over the field $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$. Write a program using Gaussian elimination to determine whether there exists a non-zero column vector $\mathbf{v}$ with $A\mathbf{v} = 0$, and to find one if there is.

Question 7 Implement the continued fraction method for factorization, and discuss briefly how it works. Your program should be capable of giving details of the intermediate steps involved. Give some detailed worked examples, including the values of N in Question 5. Investigate the number of convergents typically required for factorization.

Discuss the choice of factor base B. What improvements could be made to your method to make it more efficient for very large values of N?

References

[1] Davenport, H., *The higher arithmetic.*

[2] Koblitz, N., *A course in number theory and cryptography.*

[3] Riesel, H., *Prime numbers and computer methods for factorization.*

16 Algebra

16.1 The Galois Group of a Polynomial (7 units)

This project is related to material in the Part II course Galois Theory.

1 Introduction

The *Galois group* $G(f)$ of a polynomial f defined over a field K is the group of K-automorphisms of the field generated over K by the roots of f (the Galois group of the splitting field for f over K).

We shall consider Galois groups over the rationals and polynomials f which are monic and have coefficients in $\mathbb{Z}$. Assume that f has no repeated factor. Let f have degree $\partial f = n$ with Galois group a subgroup of the symmetric group S_n acting on the roots of f. The *decomposition group of f modulo p* is the Galois group $G_p(f)$ of f regarded as a polynomial over the finite field $\mathrm{GF}(p)$, provided that f modulo p does not have a repeated factor. The key result we shall use is that the decomposition group (when defined) is isomorphic to a subgroup of the Galois group of f. Furthermore, it is isomorphic in a way which preserves cycle types, so the cycle type of the generator of the decomposition group will also occur in the Galois group.

We shall use the decomposition groups to derive information about the Galois group of a polynomial f. For example, if $G(f)$ contains a 2-cycle, a $(n-1)$-cycle and an n-cycle, then it must be S_n. Since the decomposition group is always cyclic, this information does not distinguish between groups with the same abelian subgroups, but computing a sufficient number of decomposition groups will usually determine the cyclic subgroups and hence often determine $G(f)$, although there is always the possibility that our answer will be too small if we do not compute enough.

2 The algorithms

To find the decomposition group of f modulo p, we need information about the factorisation of f over $\mathrm{GF}(p)$. There is a repeated factor in f iff f has a factor in common with its formal derivative f' and this can be determined by applying the Euclidean algorithm. Since $\mathrm{GF}(p^r)$ is the splitting field of any irreducible polynomial of degree r over $\mathrm{GF}(p)$ and the Galois group of $\mathrm{GF}(p^r)$ over $\mathrm{GF}(p)$ is the cyclic group C_r generated by $x \mapsto x^p$, it is only necessary to find the degrees of the irreducible factors in f in order to find its Galois group over $\mathrm{GF}(p)$. We let f_r be the product of all the irreducible factors of degree r in f: then there are $n_r = \partial f_r / r$ factors of degree r in f and the Galois group $G_p(f)$ of f over $\mathrm{GF}(p)$ is cyclic, where the generator has n_r r-cycles for each r.

We determine f_r by the observation that the elements of $\mathrm{GF}(p^r)$ all satisfy the equation $\phi_r(X) = X^{p^r} - X = 0$ and hence, if we proceed by successively removing the factors $f_1, \ldots, f_{r-1}$ then at the r^{th} stage we can obtain f_r by taking the highest common factor of the residue with ϕ_r.

> **Question 1** Write procedures to compute the quotient and remainder from dividing two polynomials over $\mathrm{GF}(p)$ and use them to write a procedure to find the highest common factor of two polynomials over $\mathrm{GF}(p)$. Include in your report some test output from all three procedures. Describe an efficient way of using your procedures to compute a large power of one polynomial modulo another polynomial.

Question 2 Write a procedure to compute the decomposition group of f modulo p. You should check first that the group is defined, that is, that f and f' have no common factor, and then decompose f into factors f_r. You should try to make this procedure reasonably efficient computationally. Use your procedures in a program that will read in a monic polynomial f with integer coefficients, and print out its decomposition groups for p up to, say, 97.

Question 3 Run your program for the polynomials

$$
\begin{aligned}
&X^2 + X + 41,\\
&X^3 + 2X + 1,\\
&X^3 + X^2 - 2X - 1,\\
&X^4 - 2X^2 + 4,\\
&X^4 - X^3 - 4X + 16,\\
&X^4 - 2X^3 + 5X + 5,\\
&X^4 + 7X^2 + 6X + 7,\\
&X^4 + 3X^3 - 6X^2 - 9X + 7,\\
&X^5 + 36,\\
&X^5 - 5X + 3,\\
&X^5 + X^3 - 3X^2 + 3,\\
&X^5 - 11X^3 + 22X - 11,\\
&X^6 + X + 1,\\
&X^7 - 2X^6 + 2X + 2,\\
&X^7 + X^4 - 2X^2 + 8X + 4,\\
&X^7 + X^5 - 4X^4 - X^3 + 5X + 1.
\end{aligned}
$$

Your program should tabulate its output in columns, so that the results for this question take only a few pages in total.

Question 4 Discuss the Galois groups of these polynomials in the light of your output, with special reference to the reducible polynomials. Assuming, in each case, that the group is the smallest possible, formulate a conjecture as to the relative frequencies of the various cycle shapes for a fixed polynomial f as p varies. Do any of the polynomials (especially those of smaller degree) appear to contradict this conjecture? If so, run your programs for these polynomials for higher values of p and see if this rectifies the matter.

Programming note

If you use MATLAB then you may wish to use the `DocPolynom` class that is included as an example in the help browser. To use this you should create a directory `@DocPolynom` and place `DocPolynom.m` into it. This will enable you to define and display (non-zero) polynomials and to carry out standard algebraic manipulations with them. There is no need to include the class file in your program listings (assuming you do not modify it).

If you use a computer algebra package (such as MAPLE), then you may find that some of the routines asked for in this project are included in the package. In such cases, no credit will be given for using the packaged routines — you are expected to write your own programs. You may wish to compare the answers given by your program and by the packaged routines.

References

[1] B. L. van der Waerden, *Modern Algebra vol. 1*

16 Algebra

16.5 Permutation Groups (7 units)

This project is self-contained, building on theory covered in the Part IA course Groups. Some knowledge of the groups part of the Part IB course Groups, Rings and Modules would be useful.

1 Introduction

We are given a set of permutations of $X = \{1, \ldots, n\}$. They generate a finite permutation group $G \leqslant S_n$. The aim of this project is to replace the set of generators of G with another generating set of a very special nature. This allows us to deal with various questions. You may assume for programming purposes that $n \leqslant 20$.

2 Permutations

A permutation π of X is a bijective function from X to X. If x is an element of X then the image of x under π is written πx. If π_1 and π_2 are permutations then their product $\pi_1 \cdot \pi_2$ maps x to $\pi_1(\pi_2 x)$. The set of all permutations of the set $X = \{1, \ldots, n\}$ is the symmetric group S_n. If π is a permutation and $y = \pi y$ then y is called a *fixed point of* π.

> **Question 1** Write procedures to compute the inverse π^{-1} of a permutation π and the product $\pi_1\pi_2$ of two permutations π_1 and π_2. What is the complexity of your method for computing inverses (as a function of n)?

3 Groups

Suppose the permutation group G is generated by permutations $\pi_1, \ldots, \pi_k$. First we reduce the number of generators with the Stripping Algorithm of Sims. Let A be an $n \times n$ array of permutations which is initially empty.

Suppose we have already put the first $l-1$ permutations into the array. If π_l does not fix 1 and the $\pi_l(1)$th entry in the first row is still empty then put π_l there. Suppose the $\pi_l(1)$th entry is the permutation g. Then modify π_l to be $g^{-1}\pi_l$ so the new π_l fixes 1. Go to the second row.

If π_l does not fix 2 and the $\pi_l(2)$th entry in the second row is still empty then put π_l there. If the entry is g then modify π_l to be $g^{-1}\pi_l$ which hence fixes 1 and 2. Go on to the third row ...

If we reach the last row then we must have produced the trivial permutation which can be omitted from the generating set.

Once a permutation is placed in the array, or deemed to be the trivial permutation, we go on to try to place the next permutation in the array.

> **Question 2** Show that the modified set of permutations generates the group G. Give an upper bound for the size of the modified set of generators and for the number of operations needed to complete the algorithm. (As a function of n and the size of the original generating set, noting that, e.g., storing a permutation is $O(n)$ operations.)

Question 3 Write a procedure which computes the array of a permutation group given by a set of generators. It should receive a set of permutations as input and give a set of permutations as output, which generate the same group and are in the above reduced form. (Here, as elsewhere in this project, you should give some examples to demonstrate that your program is working correctly.)

4 Orbit and Stabilizer

Let G be a permutation group of X. If $\alpha \in X$ then the set $A = \{\beta \in X \mid \exists g \in G, g\alpha = \beta\}$ is called the *orbit* of α (under G). If $\beta \in X$ is in the orbit of α then an element $g \in G$ is called a *witness* of this if $g\alpha = \beta$. It is easy to see that β is in the orbit of α if and only if the orbit of β is the same as the orbit of α. Hence different orbits are disjoint and the orbits form a partition of X.

The *stabilizer* of an element α in X is $G_\alpha = \{g \in G \mid g\alpha = \alpha\}$. It is a subgroup of G.

Question 4 Write down a bijection between the set of left cosets of G_α in G and the orbit of α. State the orbit-stabilizer theorem.

Question 5 Write a procedure which computes the orbit with witnesses of a given element under a permutation group G given with a set of generators. It should receive as input a set of permutations and an element α and should return as a output a list of elements forming the orbit of α, together with a witness in each case. Briefly explain how your procedure works.

5 Schreier's Theorem and the final algorithm

Suppose G is a permutation group of X, given with a set of generators Y, α is an element of X and T is a complete set of left coset representatives of G_α in G. Let the surjective map $\varphi\colon G \to T$ be defined via $g\alpha = \varphi(g)\alpha$.

Question 6 Let x be an element of G_α. Write $x = y_r \dots y_1$ with each y_i an element of Y. Let t_1 be the element of T belonging to G_α. Let $t_{i+1} = \varphi(y_i t_i)$ for $i = 1, 2, \dots, r$. Show that $t_{r+1} = t_1$. Deduce that G_α is generated by the set of elements:

$$\{\varphi(yt)^{-1} \cdot y \cdot t \mid y \in Y, t \in T\}.$$

This is a special case of Schreier's Theorem.

Question 7 Write a procedure which computes a generating set of a stabilizer of a permutation group given with a set of generators. It should receive a set of permutations and an element α as input and give a set of permutations as output which generate the stabilizer. Use Question 5 to obtain T, then use Schreier's Theorem and finally reduce the set of generators with the Stripping Algorithm. Comment on the complexity of your algorithm.

Question 8 Write a program which computes the order of a permutation group G given with a set of generating permutations. The program should receive a set of permutations as input and give a natural number as output which is the order of G. You should

first reduce the number of generators with the Stripping Algorithm, and recursively find a nontrivial orbit and use the previous question until you reach a subgroup of order 1. Use the orbit-stabilizer theorem in the recursive part to find the order of G.

Give an estimate for the number of operations required by this algorithm. Use this program to compute the order of the groups generated by the following sets:

- $\{(1,2)(3,4),\ (2,4)(5,6),\ (1,3,7)(2,5,4)\}$,
- $\{(1,2,3,4)(5,6,7,8),\ (1,3,8,4)(2,7,6,9),\ (1,3,4,10)(5,7,8,11)\}$,
- $\{(1,2,3)(4,5,6),\ (7,8,9,10),\ (4,9,6)(5,10,7)(11,12)\}$,
- $\{(1,2,3,4,5,6,7,8,9,10,11,12),\ (1,13)(5,14,9,15)\}$,
- $\{(1,2,3,4,5,6,7,8,9,10,11,12,13,14),\ (1,16)(8,15)\}$

You should make a note of the group order and number of generators (before and after stripping) for each of the subgroups computed. Briefly discuss how the algorithm might perform if we did not use the Stripping Algorithm at each stage.

Are any of the generators for the groups listed above redundant?

17 Combinatorics

17.1 Graph Colouring (7 units)

This project is based on the material found in the Part II Graph Theory course.

In this project you will need to be able to generate graphs from $\mathcal{G}(n,p)$ and $\mathcal{G}_k(n,p)$. The space $\mathcal{G}(n,p)$ is that of graphs with n labelled vertices, edges appearing independently and at random with probability p. The space $\mathcal{G}_k(n,p)$ differs from $\mathcal{G}(n,p)$ only in that ij is never an edge if $i - j \equiv 0 \pmod{k}$.

1 A simple colouring algorithm

A *colouring* of a graph G is an assignment of a colour to each vertex of G, so that no two adjacent vertices receive the same colour. The *chromatic number* of G, denoted by $\chi(G)$, is the smallest number of colours for which it is possible to produce a colouring. It is believed that finding the chromatic number of a graph G is, in general, very hard. At no stage in this project are you required to implement a procedure for the exact evaluation of $\chi(G)$. You will, however, develop procedures for finding upper and lower bounds for $\chi(G)$.

The *greedy algorithm* colours a graph whose vertex set is *ordered* by colouring vertices one at a time in the order given, using colours from $\{1, 2, 3, \ldots\}$. The colour chosen for a vertex is the least colour from among those not already assigned to any previously coloured neighbours.

Question 1 Write a procedure which applies the greedy algorithm to a graph with a given ordering of the vertices. Test your program on ten members of $\mathcal{G}(70, 0.5)$, and compare the number of colours used when the vertices are ordered in the following ways: (i) by increasing degree, (ii) by decreasing degree, (iii) where v_j has minimum degree in the graph $G - \{v_{j+1}, \ldots, v_n\}$, (iv) at random.

Do the same for $\mathcal{G}_3(70, 0.75)$.

Question 2 What ordering will guarantee that the greedy algorithm uses no more than 3 colours for $\mathcal{G}_3(70, 0.75)$? Why do you think the probability 0.75 was chosen here? For each n give an example of a graph G of order $3n$ such that $\chi(G) = 3$ but on which greedy might need $n + 2$ colours.

2 Cliques

A *clique* in a graph G is a complete subgraph of largest order in G. (This definition differs from some in the literature.) Notice that $\chi(G)$ is at least as large as the order of a clique.

A greedy-type algorithm for finding a complete subgraph in G would start with a subgraph of order one (a vertex) and repeatedly try to find a vertex joined to all vertices of the subgraph selected so far, until no further such vertex could be found.

Question 3 Give an argument to suggest that it is unlikely the greedy-type algorithm will find a complete subgraph of order 14 in a graph from $\mathcal{G}(2000, 0.5)$. How large do you think a clique is likely to be in a graph from $\mathcal{G}(2000, 0.5)$?

Question 4 Write a procedure to find a clique in a graph G. [Note: this procedure may be time-consuming but should not be excessively so on the examples here.] Compare, for several graphs, the resulting lower bound you get on $\chi(G)$ with the upper bounds obtained previously.

3 Colouring

An *independent set* in a graph is a subset of the vertex set which spans no edges. A colouring is thus just a partition of the vertices into independent sets.

Question 5 Convert your clique procedure to find an independent set of maximum order in a graph. Hence write a procedure to colour a graph by the following method. First find a largest independent set I_1. Then find a largest independent set I_2 in $G - I_1$, then I_3 in $G - I_1 - I_2$, and so on until nothing remains. Compare the upper bounds on $\chi(G)$ so obtained with previous bounds. Try your program on ten members of $G_7(70, 0.5)$ also. Is there a change in behaviour as p is increased, say from 0.4 to 0.6? If your program can handle larger graphs in a finite time, obtain further data of interest.

None of the above methods for bounding $\chi(G)$ is guaranteed to find $\chi(G)$ exactly.

Question 6 Estimate (crudely) the theoretical running times of all the algorithms used above as functions of n when the input is a typical member of $\mathcal{G}(n, 0.5)$. Describe in outline, but do not implement, a procedure for colouring a graph with exactly $\chi(G)$ colours, and estimate its running time.

References

[1] Bollobas, B., Modern Graph Theory, Springer 1998.

17 Combinatorics

17.3 Hamiltonian cycles (5 units)

This project is based on the material found in the Part II Graph Theory course.

In this project you will need to be able to generate graphs from $\mathcal{G}(n,p)$, the space of graphs with n labelled vertices, edges appearing independently and at random with probability p.

A *Hamiltonian cycle* in a graph is a cycle which contains every vertex.

Question 1 Describe a simple algorithm to check whether a graph has a Hamiltonian cycle, and implement it. Test your program on a few particular graphs; and then use it on a selection of graphs from $\mathcal{G}(n,p)$ with n up to, say, 21 and p firstly varying from 0.1 to 0.9 and then varying from $0.1 \ln n/n$ to $1.9 \ln n/n$. Tabulate your results, showing for each n and p the number of graphs from your selection which had a Hamiltonian cycle.

Question 2 Estimate the theoretical running time of your algorithm as best you can. Compare the answers for the worst case and an average case.

Question 3 Find a simple property possessed by many of your non-Hamiltonian examples that is sufficient (though maybe not necessary) to force a graph to be non-Hamiltonian. Why do you think the second range of values of p was chosen?

You will notice that your algorithm rapidly becomes prohibitively expensive as the order of the graph increases. In this case an "approximation algorithm" can be useful. An approximation algorithm for the Hamiltonian cycle problem would seek to make a very good attempt at finding a cycle in a short space of time. If it succeeds, well and good. If it fails, there may have been a cycle it missed, but it is hoped that the probability of this will be small.

Here is a simple algorithm to search for a Hamiltonian cycle. Construct a sequence of paths $P_1, P_2, \ldots$, where P_1 is just a single vertex v_0. Given a path P_j from v_0 to v_k, proceed as follows:

1. If P_j has length $n-1$ and $v_0v_k \in E(G)$, output a Hamiltonian cycle;
2. if P_j has length less than $n-1$ and v_k is joined to a vertex not in P_j, extend the path P_j to a path P_{j+1}. If there are several neighbours not in P_j, pick one of them at random;
3. otherwise construct a new path of the same length as P_j in this way: select a neighbour v_i of v_k in P_j at random. Then P_{j+1} is the path $v_0 \ldots v_{i-1}v_iv_kv_{k-1}\ldots v_{i+1}$.

Question 4 Implement this algorithm, and try it on your earlier examples. You should set a stopping time T for the procedure so that, if it has constructed P_T and still found no cycle, it quits. What functions work well in practice (i.e. fairly reliably find a cycle but aren't too expensive)?

Question 5 In general, the stopping time T should be a function on n and p. What do you think would be a reasonably good function to use for large n? How do you think the running time might vary with p? Estimate the complexity of the algorithm in the average case.

References

[1] Bollobas, B., Modern Graph Theory, Springer 1998.

19 Communication theory

19.1 Random codes (5 units)

Background material for this project is given in the Part II course Coding and Cryptography.

The (binary) Hamming space $\{0,1\}^n$ consists of all possible n-tuples of 0s and 1s. We define a *code* C of length n and size r to be a subset C of the Hamming space $\{0,1\}^n$ with r elements. The *Hamming distance* between two elements $\mathbf{x} = (x_i)$, $\mathbf{y} = (y_i)$ of $\{0,1\}^n$ is the number of places in which $\mathbf{x}$, $\mathbf{y}$ differ. The *minimum distance* of a code C is the minimum Hamming distance $d(C)$ between distinct elements of C. The *information rate* of C is $\frac{1}{n}\log_2 r$. We define the *error-control rate* to be $(d-1)/n$ (note: elsewhere it is often defined as d/n).

In this project we investigate how high the information rate of a randomly-generated code can be, subject to constraints on its error-control rate. In the first study, any randomly chosen code may be considered, whereas in the second study only randomly generated linear codes are allowed. In each study we try two approaches: specify the information rate of the random code and see what error-control rate can be achieved, or specify the error-control rate and see what information rate can be achieved.

Question 1 Write a procedure to find the minimum distance of a code. Use your procedure to write a program which generates random codes of length n and size r and then computes the minimum distance.

Run your program several times with various values of n and r, for each choice finding the best (i.e., largest) d that you can.

Question 2 Now generate codes of length n and minimum distance d by starting with an initial code vector, say $(0,\ldots,0)$ and randomly generating further vectors, adding a new vector to the code if it has distance at least d from all the vectors already in the code.

Run your program several times for each choice of parameters n and d, finding the best (i.e., largest) r that you can.

Question 3 Take the output from the two previous questions and plot the corresponding points on a graph with information rate and error-control rate as the two axes. Comment on your results.

We call a code *linear* if it forms a subspace of the Hamming space, regarded as a vector space over the field F of 2 elements. The *weight* $w(\mathbf{x})$ of a vector $\mathbf{x}$ is the number of non-zero components, that is, the Hamming distance $d(\mathbf{x},\mathbf{0})$. The minimum distance of a linear code is just the minimum non-zero weight. The *rank* k of a linear code is the dimension of the code as a subspace, and the size of a linear code is $r = 2^k$.

Question 4 Write a procedure to find the minimum non-zero weight of a code generated over F by a set $g_1, \ldots, g_k$ of k generators. Use your procedure to find linear codes of given length n and either given rank k or given minimum distance d by considering random sets of generators. As before, run your programs several times to plot the information and error-control rates and comment on the results.

Question 5 Comment on your results in relation to known constraints on the design of codes and the Shannon coding theorems.

Comment on these methods as a way of designing effective error-control codes.

References

[1] C.M. Goldie and R.G.E. Pinch, *Communication theory*, CUP, 1991.

20 Probability

20.5 Percolation and the Invasion Process (9 units)

This project requires general knowledge of probability theory, at the level of IA Probability. It also requires competency in programming.

1 Introduction

This project concerns certain probability models for bond percolation. The book by Grimmett [1] is a good source to learn more about percolation.

We work on a connected graph $G = (V, E)$, that is, a collection of nodes V connected by edges E. To each edge $e \in E$, we assign, independently, a uniform random variable $U_e \sim U[0, 1]$. We decide on a value $p \in [0, 1]$; we declare the edge e to be *p-open* if $U_e < p$, and we declare it to be *p-closed* otherwise.

When p is very small, very few edges are open; but as we increase p, there appear open clusters, i.e. sets of nodes connected by open edges.

Percolation theory is the study of the geometry of the open clusters. In particular, important questions are whether or not there exists an infinite cluster of open edges; and if one does exist, how many infinite clusters there are. Clearly if $p = 0$ there is none and if $p = 1$ there is one open cluster, namely the graph G itself.

2 The binary tree

Let V, the set of nodes of the graph, consist of finite strings, as follows: V contains the empty string '' (also known as *Eve*), and the three strings '1', '2' and '3' (also known as *Eve's daughters*), and also every string that is one of Eve's daughters followed by a finite sequence of '1's and '2's. Two nodes are connected by an edge if one can be obtained by appending one digit to the other. For example, '3221' is connected to '322' (its mother) and to '32211' and '32212' (its two daughters). As before, each edge e is assigned a random variable $U_e \sim U[0, 1]$.

(We can use this as a crude model to describe the propagation of a defective gene in a population.)

Question 1 Let ϕ_p be the probability that Eve's daughter '1' is in an infinite open cluster consisting of her own descendents. Show that

$$\phi_p = 2p(1-p)\phi_p + p^2\big(\phi_p^2 + 2\phi_p(1-\phi_p)\big).$$

It can be shown that ϕ_p is the maximal solution to this equation. Find ϕ_p. (One way to obtain a 'merit' mark in this project, though not the only way, is to show that ϕ_p *is* the maximal solution.)

Now let θ_p be the probability that Eve is in an infinite open cluster. Find θ_p and draw its graph as a function of p.

Question 2 Show that, for $p \leqslant \frac{1}{2}$, there are almost surely no infinite clusters. How many infinite clusters are there if $\frac{1}{2} < p < 1$? Justify your answer.

This model exhibits a property which is general to percolation models: there exists a critical probability p_c such that for $p < p_c$ there is no infinite cluster and for $p > p_c$ we find at least one infinite cluster.

The region $p > p_c$ is called the *supercritical* region. We can ask questions like: what is θ_p, the probability that a node chosen arbitrarily lies in an infinite open cluster? (In Question 1, in calculating θ_p, we could of course have designated any node to be Eve.)

The region $p < p_c$ is called the *subcritical* region. We ask questions like: how likely are we to observe an open cluster of size n? if there is an open cluster of size n, what shape is it?

Most of the interesting (unsolved) problems relate to the geometry of open clusters when p is near p_c. For example, there are many graphs (e.g. $\mathbb{Z}^3$) where it is not known (but strongly conjectured) that there is no infinite p_c-open cluster.

3 The square lattice

The square lattice in two dimensions $\mathbb{L}^2$ is a graph with $V = \mathbb{Z}^2 = \{(m, n) : m, n \in \mathbb{Z}\}$. If the distance function is

$$d\big((k, l), (m, n)\big) = |k - m| + |l - n|,$$

the edges of the graph are straight lines connecting nodes which are distance 1 apart.

Let us look at two techniques which will help us estimate the critical probability p_c above which there is an infinite cluster and below which there is none.

Lower bound

Let us start at the origin. Let σ_n be the number of self-avoiding paths (i.e., paths which traverse each edge at most once) of length n leading away from the origin.

Question 3 Let $\lambda = \limsup_{n\to\infty} \sigma_n^{1/n}$. Show that $\lambda \leqslant 3$. Show further that $p_c \geqslant \lambda^{-1}$.

The actual value of λ is an open problem.

Upper bound

The general behaviour of large clusters in the subcritical region $p < p_c$ is described in the following result. Some notation first: We say $x \leftrightarrow y$ if there exists an open connected path between x and y. Define the open sphere S_n to be

$$S_n = \big\{x \in \mathbb{Z}^2 : d(x, 0) \leqslant n\big\}.$$

The boundary ∂S_n consists of the nodes where $d(x, 0) = n$. Let $P_p(0 \leftrightarrow \partial S_n)$ be the probability that there exists a p-open path connecting the origin to some node in ∂S_n. It can be shown that, for $p < p_c$, there exists $\psi_p > 0$ such that

$$P_p(0 \leftrightarrow \partial S_n) < e^{-n\psi_p} \quad \text{for all } n. \tag{1}$$

The proof is beyond the scope of this project but can be found in [1, Sections 5.2 and 6.1].

Armed with the above result, we aim to show that $p_c \leqslant \frac{1}{2}$. We do this as follows. Consider the following subgraph $G_n = (V_n, E_n)$ of the square lattice.

$$V_n = \big\{(k,l) \,:\, 0 \leqslant k \leqslant n,\ 0 \leqslant l \leqslant n-1\big\}.$$

Let E_n be all the edges in E connecting these nodes. We call $\{(k,l) \in V_n : k = 0\}$ the left boundary, and $\{(k,l) \in V_n : k = n\}$ the right boundary. Let A be the event that some node in the left boundary is connected to the right boundary via a path consisting of open edges.

Question 4 Show that $P_{1/2}(A) = \frac{1}{2}$. *Hint. You may find it useful to consider the dual graph $\bar{G}_n$, which has nodes at*

$$V_n' = \big\{(k+\tfrac{1}{2}, l-\tfrac{1}{2}) \,:\, 0 \leqslant k \leqslant n-1,\ 0 \leqslant l \leqslant n\big\},$$

and edges joining those nodes which are distance 1 apart, and whose edges are open or closed depending on whether the edges of G are open or closed, in a manner which you should specify.

Question 5 By constructing n events $\{A_i\}$ such that $A = \cup A_i$, and using (1), prove that $p_c \leqslant \frac{1}{2}$.

4 The Invasion Process

We can try and use computing power to estimate p_c and θ_p. Suppose that at time $n = 0$ you are an invading force standing at the origin. We will call I_n the set of nodes you have invaded by time n. At time $n+1$ you invade another node by looking at the edge-boundary of your territory and walking along the edge with the least value of U_e attached to it. Formally, we define

$$\partial I_n = \big\{e \in E : I_n \overset{e}{\leftrightarrow} \mathbb{Z}^2 \setminus I_n\big\}.$$

We use the notation $x \overset{e}{\leftrightarrow} y$ to mean that the edge e connects x and y. You walk from I_n along the edge $e_n \in \partial I_n$ which satisfies

$$U_{e_n} = \min\{U_f : f \in \partial I_n\},$$

so that I_{n+1} is I_n with the node at the other side of e_n added. It can be shown that, almost surely,

$$\limsup_{n\to\infty} U_{e_n} = p_c. \tag{2}$$

(The proof is not hard, and is outlined at the end of this project.) The advantage of using the sequence U_{e_n} to estimate p_c is that the amount of memory required to store U_{e_n} and to calculate $U_{e_{n+1}}$ is $O(n)$. This is true whether we are working in $\mathbb{L}^2$ or $\mathbb{L}^{47}$.

Question 6 Implement the invasion process. Describe your algorithm.
Explain in particular why it only requires $O(n)$ storage space to calculate the first n values of U_{e_n}, and what it does to ensure it never revisits a vertex.
Comment also on the complexity of the algorithm (the number of time steps needed).

Question 7 Use your program to estimate p_c for $\mathbb{L}^2$, and explain your method. Estimate p_c for $\mathbb{L}^3$ (which is defined like $\mathbb{L}^2$, but using $\mathbb{Z}^3$ rather than $\mathbb{Z}^2$). Include in your report any appropriate plots.

It is desired to plot θ_p, the density of the infinite cluster.

Question 8 Explain how the invasion process can be used to estimate θ_p simultaneously for all p. Produce a plot of θ_p against p for $\mathbb{L}^2$ by running the simulation for large n several times (at least $n = 5000$, at least 500 times). For what values of p do you expect your plot to be inaccurate? Why?

Appendix

Here is an outline of the proof of (2).

Let $p > p_c$. Then there exists an infinite p-open cluster. Let T_p be the first time that the invasion process hits the cluster (i.e. the first time that the vertex $I_{T_p} \setminus I_{T_p-1}$ is in the infinite p-open cluster). It can be shown that $P(T_p < \infty) = 1$. For all $n \geqslant T_p$, there will always be an edge in ∂I_n with $U_e \leqslant p$. It follows that

$$\limsup_{n\to\infty} U_{e_n} \leqslant p.$$

Since $p > p_c$ was arbitrary, $\limsup_{n\to\infty} U_{e_n} \leqslant p_c$.

Now let $p < p_c$. Suppose that with some probability $\alpha > 0$,

$$\limsup_{n\to\infty} U_{e_n} \leqslant p < p_c.$$

Then we have (with a positive probability) an infinite cluster, with all but finitely many of its edges p-open. It follows that (with a positive probability) there exists an infinite p-open cluster. This contradicts the definition of p_c as the critical probability.

References

[1] G.R. Grimmett. *Percolation.* Springer-Verlag, Berlin 1989 and 1999

20 Probability

20.6 Loss Networks (9 units)

This project requires knowledge of discrete- and continuous-time Markov chains, covered in the Part IB Markov Chains and Part II Applied Probability courses respectively.

Introduction

Nearly a century ago the mathematician Erlang, working for the Copenhagen Telephone Company, devised the first mathematical theories for telecommunications networks. The technology we have now would seem like science fiction to Erlang, yet his insight into the essential structure of networks means that his theorems are just as useful for designing an optically-routed backbone for the Internet as they were for early Danish telephony.

In this project we will deal with certain models for telephone networks, arising from Erlang's work. We consider a network to be a collection of links (cables). Each telephone call occupies a certain amount of space on certain links; for example, a telephone call from a Cambridge college to a London house might occupy 8 kbit/sec of space on the link from the college to the university exchange, and on the link from the university exchange to BT's Cambridge exchange, and on the link from there to a London exchange, and so on. This space is occupied for the duration of the call. The links which comprise the national telephone network only have limited capacity, and when they are full we get a busy signal. Some interesting questions are: What is the probability of a busy signal? How does it depend on the volume of traffic? Can we reduce this probability by strategies such as offering multiple routes for a call?

For further reading see [1].

1 The Erlang link

Consider a single link with the capacity to carry C simultaneous calls. Suppose that new calls arrive as a Poisson process of rate ν, that each call lasts for a duration which is exponential with mean 1, and that all call durations are independent of each other and of the arrival process. If a new call arrives when the link is already carrying C calls, then the new call is *blocked*. This system is known as the "Erlang link".

> **Question 1** Set up a continuous-time Markov model for the Erlang link. Calculate the equilibrium probability that there are i calls in progress.

Define $E(\nu, C)$ to be the equilibrium probability that there are C calls in progress.

We say that an arriving call "sees" the system in state s if the system is in state s just before it arrives. The PASTA property (Poisson arrivals see time averages) says that the long-run proportion of arrivals which see the system in state s is equal to the equilibrium probability that the system is in state s.

> **Question 2** Show that the PASTA property holds for the Erlang link. *Hint. One approach is to consider the discrete-time Markov chain which records the state of the system after each event, where an event is an attempted arrival or a departure, and to*

express the equilibrium distribution of this chain in terms of the equilibrium distribution for the continuous-time chain. Deduce that the long-run proportion of calls which are blocked is $E(\nu, C)$.

Question 3 Write a program to simulate the Erlang link and to measure the blocking probability. Compare the empirical blocking probability to that given by $E(\nu, C)$ for a range of values of C up to 600 and an appropriate range of values of ν.

2 Alternative Routing

Consider now a network of links. For simplicity, suppose that the network is a complete graph on K nodes, i.e., there is a link between every pair of nodes $\{1, \ldots, K\}$, and that each link has capacity C. Suppose that for every pair of nodes (a, b), calls between a and b arise as a Poisson process of rate ν. It might seem reasonable, in order to reduce the blocking probability, to offer an alternative route if the direct link is full. Specifically, suppose that calls between a and b are routed as follows:

1. If there is spare capacity on the direct link $a \leftrightarrow b$, route the call over that link.
2. Otherwise, pick a new node c uniformly at random from the other $K - 2$ nodes. If there is spare capacity on $a \leftrightarrow c$ and on $c \leftrightarrow b$, route the call over these two links.
3. Otherwise, the call is blocked.

We will call this the "Alternative Routing" system. One way (but not the only way) to obtain a 'merit' mark in this project is to prove that the Alternative Routing system satisfies the PASTA property.

As you can see, it is possible in principle to set up a Markov process model for this system, and thereby to calculate the equilibrium distribution; but the number of states is so large that, even for moderate K, it is not computationally practical to do so. Instead, we can use a famous approximation called the *Erlang fixed point approximation*, which is that blocking occurs independently on different links. This leads to the formula

$$B = E(\nu + 2\nu B(1 - B), C) \tag{1}$$

where B is the probability that an incoming call cannot be routed on its chosen direct link.

Question 4 Give a careful intuitive explanation for (1). In what sense could $\nu + 2\nu B(1 - B)$ be called the "offered load" on a link?

Question 5 Demonstrate numerically (e.g., by plotting an appropriate graph) that for some values of C and ν this equation has a unique solution, and that for other values it has multiple solutions.

Now pick $C = 600$ and choose ν such that (1) has multiple solutions.

Question 6 Write a program to simulate the Alternative Routing system. Explain clearly and concisely the algorithm you have used. Run your simulation, for $K = 5$, and record the cumulative count of the number of calls blocked. Plot a graph which shows this count as a function of time. *Programming hint. Run your simulation for at least a million transitions of the Markov process; this should take less than 10 minutes on a modern computer. You may find it convenient to write a small subsample of your simulation output to a file, then use Excel or some other program to plot the result.*

You should find, from your simulation, that the system spends some of the time in a high-blocking regime and some of the time in a low-blocking regime, corresponding to solutions of (1).

Question 7 Give an intuitive explanation for why there are multiple solutions. It is a standard result from Markov chain theory that this Markov process has a unique equilibrium distribution; comment briefly on how this result relates to the existence of multiple solutions.

Question 8 You should observe that there is one solution to (1) which is *not* reflected in your simulations. By considering fixed points of the map $B \leftarrow E(\nu + 2\nu B(1-B), C)$, suggest why this is so.

Question 9 Use the Erlang fixed-point approximation to find the probabilities that an incoming call is blocked in the high-blocking regime and in the low-blocking regime. How do these compare to a network without alternative routing, i.e., in a network in which a call may only be routed on the direct link?

3 Trunk reservation

A telecoms operator would be alarmed at the situation you investigated in the last section, and would seek to control the network so that it stays in the low-blocking state. One way to do this is with a technique called "trunk reservation". We modify the call admission procedure, so that calls arising between a and b are routed as follows:

1. Consider the direct link $a \leftrightarrow b$. If this link has spare capacity, route the call over it.
2. Otherwise, pick a new node c uniformly at random from the other $K-2$ nodes, and consider the two links $a \leftrightarrow c$ and $c \leftrightarrow b$. If on each of these links the number of calls in progress is less than $C - s$, then route the call over these two links.
3. Otherwise, the call is blocked.

The parameter $s > 0$ is known as the *trunk reservation parameter*.

Question 10 Develop a fixed-point approximation for this system. *Hint. First set up a suitable Markov model for the number of calls in progress on a single link with two classes of traffic.*

Question 11 Compare the blocking probability to those you found in Question 9. Has trunk reservation improved matters? How large should the trunk reservation parameter be?

References

[1] F. P. Kelly, *Network Routing*, Philosophical Transactions of the Royal Society series A, 1991. `http://www.statslab.cam.ac.uk/~frank/loss/`

23 Astrophysics

23.5 Ionization of the Interstellar Gas near a Star (8 units)

No knowledge of Astrophysics is assumed or required: all relevant equations are defined and explained in the project itself.

1 Introduction

The interstellar medium surrounding a hot star is ionized by the radiation from the star. In this project we calculate the size of the ionized region, and gain some insight into how its structure depends on the nature of the radiation from the star.

Assuming that there is a uniform, static, constant temperature gas surrounding a spherically symmetric star provides a good approximation, which keeps the essentials of the situation without allowing unnecessary distractions. Each gas element is assumed to be in ionization equilibrium, so the ionization rate for each atom is balanced by recombinations. We assume that the sole source of ionization is by absorption of radiation from the star giving a bound electron enough energy to escape from the atom. Recombination occurs when a free electron is captured by an ion with the creation of a photon. Since hydrogen is the most common element in the Universe, its behaviour will dominate in most cases, so we consider only a pure hydrogen interstellar medium.

2 Radiation from a star

The radiation from the star is specified as L_ν, the total energy output from the star per unit frequency ν per unit time, so the luminosity per unit frequency interval L_ν is expressed in W Hz^{-1}. The total energy output radiated from the star is the integral of this quantity over all frequencies, or $L = \int_0^\infty L_\nu \mathrm{d}\nu$. In some cases a star spectrum is reasonably well approximated by a black-body, so the radiation flux in a frequency interval $\mathrm{d}\nu$ at frequency ν emerging per unit area from the surface of the star where the temperature is T_* is given by

$$I_\nu \mathrm{d}\nu = \frac{2\pi h}{c^2} \frac{\nu^3}{\exp(\frac{h\nu}{kT_*}) - 1} \mathrm{d}\nu,$$

where h is Planck's constant, c is the velocity of light, and k is Boltzmann's constant (given below). Thus for a star of radius R

$$L_\nu = 4\pi R^2 \frac{2\pi h}{c^2} \frac{\nu^3}{\exp(\frac{h\nu}{kT_*}) - 1}.$$

Question 1 The sun has radius $R = 6.96 \times 10^8$ metres, and the total luminosity $L = 3.90 \times 10^{26}$ W. Show, using the above equation, that its surface temperature is close to 5800 K.

A 7 solar mass star has a surface temperature $T_* = 20,000$K and a luminosity $L = 4.0 \times 10^{29}$W. What is its radius? What is the radius of a 12 solar mass star which has a surface temperature $T_* = 25,000$K and a luminosity $L = 4.0 \times 10^{30}$W.

3 Equations for ionization and recombination

An element of gas at a distance r from the (centre of the) star will receive $\frac{L_\nu}{4\pi r^2}$ W m^{-2} s^{-1} if the radiation is not attenuated by any material between it and the star. The number of photons received per second per unit frequency interval is L_ν divided by the energy, $h\nu$ per photon. If these photons have frequency $\nu \geqslant \nu_0 = 3.29 \times 10^{15}$Hz they have enough energy to separate a hydrogen atom into a proton and an electron. The rate at which this happens depends on the frequency of the radiation, and is given by the photon rate per second $\times$ an absorption coefficient a_ν per hydrogen atom. For an energy flux $I_\nu(r)$, so photon flux $I_\nu(r)/h\nu$, the rate of ionization per unit volume is

$$n_{\mathrm{H}^0} \int_{\nu_0}^{\infty} \frac{I_\nu(r)}{h\nu} a_\nu \mathrm{d}\nu,$$

where n_{H^0} is the number density of neutral hydrogen atoms (i.e. number of neutral hydrogen atoms per cubic metre). The coefficient a_ν depends only on the atomic species being considered, and for neutral hydrogen

$$\begin{aligned} a_\nu &= a_{\nu_0} \left(\frac{\nu_0}{\nu}\right)^3 \quad \text{for } \nu \geqslant \nu_0 \\ a_\nu &= 0 \ \text{ for } \nu < \nu_0, \end{aligned}$$

where $a_{\nu_0} = 6.3 \times 10^{-22}$ m^2. If absorption can occur at the gas element, it can also occur in all the elements between the star and the one under consideration. As we go along a path dr the radiation is attenuated, so at a given frequency ν,

$$\frac{\mathrm{d}I_\nu}{\mathrm{d}r} = -n_{H^0} a_\nu I_\nu,$$

and so

$$I_\nu(r) = I_\nu(R) e^{-\tau_\nu},$$

where τ_ν is defined by

$$\frac{\mathrm{d}\tau_\nu}{\mathrm{d}r} = n_{H^0}(r) a_\nu$$

and $\tau_\nu(R) = 0$. τ_ν is referred to as the *optical depth* at the frequency ν. There is no redistribution of the photons in frequency (they are effectively absorbed from the point of view of this calculation), so we can determine τ_ν from

$$\frac{\mathrm{d}\tau_{\nu_0}}{\mathrm{d}r} = n_{H^0}(r) a_{\nu_0}$$

and

$$\tau_\nu = \tau_{\nu_0} \left(\frac{\nu_0}{\nu}\right)^3.$$

This absorption of radiation and the ionization it causes must be balanced by recombination of protons and electrons at the same rate per unit volume. This rate depends on the number densities of the protons (n_p) and the electrons (n_e), their relative velocity and a velocity-dependent

cross-section for the interaction which has to be integrated over the velocity distribution at whatever temperature the gas is at. These velocity-dependent terms are combined into recombination coefficients α_B which are tabulated for various gas temperatures T:

T (K)	α_B
5 000	4.54×10^{-19} m^3 s^{-1}
10 000	2.59×10^{-19}
20 000	2.52×10^{-19}

Then we are in a position to write down the ionization balance equation

$$n_{\mathrm{H}^0} \int_{\nu_0}^{\infty} \frac{L_\nu}{4\pi r^2 h\nu} a_\nu e^{-\tau_\nu} \mathrm{d}\nu = n_p n_e \alpha_B(T), \tag{1}$$

and subsidiary equations

$$\frac{\mathrm{d}\tau_{\nu_0}}{\mathrm{d}r} = n_{H^0}(r) a_{\nu_0}, \tag{2}$$

$$\tau_\nu = \tau_{\nu_0} \left(\frac{\nu_0}{\nu}\right)^3, \tag{3}$$

$$n_p = n_e, \tag{4}$$

$$n_p + n_{H^0} = n_H, \tag{5}$$

which govern the ionization balance for the hydrogen-filled interstellar medium near a star.

[We have omitted a step here, and assumed that the energy released by recombination does not give rise to an ionizing photon. Often it does, but we assume that all this does is cause another ionization nearby until recombination occurs to an upper level in the hydrogen atom, and then the energy is lost from the system through radiation at frequencies less than ν_0. The net result is a change in the effective recombination coefficient, to the one quoted here. Those interested in more details will find them in Osterbrock's book (1989)]

4 Ionization near stars

A typical interstellar medium hydrogen number density is $n_H = 10^6$ m^{-3}, and a typical temperature is $T = 10^4$K.

> **Question 2** Write a program to solve the ionization equations to obtain the neutral hydrogen and proton densities as a function of distance from the centre of the star, assuming that the interstellar gas has a constant temperature and density. Note that the coefficients involved have large powers of 10, so for some compilers a rescaling of variables may be desirable. Describe any transformations used. Are there any advantages to using τ_{ν_0} instead of r as the radial coordinate?

Now apply this program to some realistic cases:

> **Question 3** Determine the ionization and neutral fractions of hydrogen as a function of distance from the star with surface temperature $T_* = 20,000$K and luminosity 4×10^{29}W for an interstellar gas density $n_H = 10^6$ m^{-3} and $T = 10^4$K, and plot the results. Show in particular that at some radial distance, r_1, there is quite a sharp transition from the gas being mostly ionized to mostly neutral. What is the value of r_1, the distance from the centre of the star where $n_p = n_{H^0}$? Give your answers to two significant figures.

Repeat the calculation for gas of the same temperature and density but with the 12 solar mass star at the centre and again with the Sun as the central star. Also, compute the ionization fractions for all three stars in the cases that the gas temperature is 5000K and 20, 000K. Comment on any similarities or differences between the three cases.

Provide plots of the nine cases (three stellar masses and three gas temperatures) and a table containing the values of r_1 for each case.

4.1 An approximation to r_1

Equation (1) can be integrated over volume from $r = 0$ to $r = \infty$ by using the definition of τ_ν to replace dr and assuming that the recombination term is well approximated by $n_p = n_e = n_H$ for $r \leqslant r_1$ and $n_p = n_e = 0$ for $r > r_1$. This r_1 is called the Strömgren radius.

Question 4 Show that, under these circumstances,

$$Q(H) \equiv \int_{\nu_0}^{\infty} \frac{L_\nu}{h\nu} \mathrm{d}\nu = \frac{4\pi}{3} r_1^3 n_H^2 \alpha_B,$$

where $Q(H)$ is the total number of ionizing photons emitted by the star per second.

Calculate the values of $Q(H)$ for the cases given above and compare the r_1 determined from this approximation with the values you have computed for $T = 10,000$K.

5 The effect of a quasar on the host galaxy

The energy output from a quasar is very different from that of a star, both in intensity and frequency dependence. A bright quasar has energy output of 4×10^{39}W, and resides in the centre of a galaxy of radius 3×10^{20}m (= 30,000 light years).

Question 5 If the frequency dependence of the quasar luminosity at frequencies $\nu \geqslant \nu_0$ is given by

$$L_\nu = 10^{24} \left(\frac{\nu}{\nu_0} \right)^{-1.4} \exp\left(-\frac{\nu}{10\nu_0} \right),$$

determine whether or not any of the interstellar gas in the galaxy has neutral fraction $n_{H^0}/n_H > 0.5$. (Assume that the gas temperature is 10^4K, and density 10^6 hydrogen atoms per cubic metre. Assume also that there is no gas within 10^{18}m of the quasar position, so start the computation there.)

Tabulate the results for the hydrogen neutral fraction as a function of distance from the quasar.

Useful constants:

velocity of light	$c = 2.998 \times 10^8 \mathrm{ms}^{-1}$
Planck's constant	$h = 6.626 \times 10^{-34}$Js
Boltzmann's constant	$\mathrm{k} = 1.381 \times 10^{-23}\mathrm{JK}^{-1}$

References

[1] Osterbrock, D.E., 1989. *Astrophysics of Gaseous Nebulae and Active Galactic Nuclei* University Science Books: Mill Valley, CA (Especially chapter 2).

23 Astrophysics

23.6 Accretion Discs (8 units)

No knowledge of astrophysics is assumed or required in this project. All relevant equations are defined and explained in the project.

1 Fluid Equations

Accretion discs are composed of fluid orbiting a central object. They evolve viscously so that matter falls inwards while angular momentum drifts outwards. Accretion discs are found in binary star systems, around forming stars and in active galactic nuclei. We use cylindrical polar coordinates (R, ϕ, z) and assume axisymmetry for this project. When the central object dominates the gravitational field the angular velocity of the matter in the disc is Keplerian so that

$$\Omega = \left(\frac{GM}{R^3}\right)^{1/2}, \tag{1}$$

where M is the mass of the central object and G is the gravitational constant. The disc is made up of annuli of matter lying between R and $R + \Delta R$ with mass $2\pi R \Delta R \Sigma$, where $\Sigma(R, t)$ is the surface density (with dimensions ML^{-2}) of the disc at time t.

The equation describing conservation of mass is

$$R\frac{\partial \Sigma}{\partial t} + \frac{\partial}{\partial R}(R\Sigma V_{\mathrm{R}}) = 0, \tag{2}$$

where V_{R} is the radial velocity in the disc. Conservation of angular momentum gives us

$$R\frac{\partial(\Sigma R^2\Omega)}{\partial t} + \frac{\partial}{\partial R}(R\Sigma V_{\mathrm{R}} R^2\Omega) = \frac{1}{2\pi}\frac{\partial \Gamma}{\partial R}, \tag{3}$$

where the viscous torque

$$\Gamma = 2\pi R\nu\Sigma R^2\Omega' \tag{4}$$

where $\Omega' = \frac{d\Omega}{dR}$ and $\nu(R, \Sigma)$ is the viscosity in the disc.

Question 1

Show that, for Ω independent of time,

$$R\frac{\partial \Sigma}{\partial t} = -\frac{\partial}{\partial R}\left[\frac{1}{2\pi(R^2\Omega)'}\frac{\partial \Gamma}{\partial R}\right]. \tag{5}$$

and hence, using equation 1, for a Keplerian disc that

$$\frac{\partial \Sigma}{\partial t} = \frac{1}{R}\frac{\partial}{\partial R}\left[R^{1/2}\frac{\partial}{\partial R}(3\nu\Sigma R^{1/2})\right] \tag{6}$$

This is the basic equation that governs the evolution of the surface density of a Keplerian accretion disc, which we shall assume for the rest of this project. The viscosity may be a function of Σ, R and t and so this equation may be non-linear.

Question 2

The mass accretion rate $\dot{m}$ through the disc is

$$\dot{m}(R) = -2\pi R\Sigma V_{\rm R} \tag{7}$$

Show that

$$V_{\rm R} = -\frac{3}{\Sigma R^{1/2}}\frac{\partial}{\partial R}(\nu\Sigma R^{1/2}). \tag{8}$$

In a steady state disc the accretion rate through the disc is constant. Find, analytically, the steady state solution for $\nu\Sigma$ from equation (6). Use the inner boundary condition $\Sigma = 0$ at $R = R_{\rm in}$ where ν is finite at $R = R_{\rm in}$. Plot $\nu\Sigma$ in units of $\dot{m}$ against $R/R_{\rm in}$ in the range $1 < R/R_{\rm in} < 100$.

We can make the problem dimensionless by setting $r = R/R_0$, $\tau = t/t_0$, $\sigma(r,\tau) = \Sigma/\Sigma_0$, and $\eta(r,\sigma) = \nu/\nu_0$. Find a condition for t_0 such that equation (6) remains the same for these dimensionless variables. Assume that the accretion rate is small enough that $M \approx$ const.

Question 3

Use the substitution $X = r^{1/2}$ in equation 6 to derive

$$\frac{\partial f}{\partial \tau} = \frac{\partial^2 g}{\partial X^2} \tag{9}$$

where f and g are to be determined. We consider X to be discrete with values X_i where $i = 1, 2, ...100$ and $X_1 = \Delta X$ where ΔX is a constant. We let $X_{i+1} = X_i + \Delta X$ and time be τ_n where $n = 1, 2, ...$ with $\tau_{n+1} = \tau_n + \Delta\tau$. We have $\tau_1 = 0$ and we let $f_i^n = f(X_i, \tau_n)$ and $g_i^n = g(X_i, \tau_n)$.

Show that equation (9) can be represented by the difference equation

$$f_i^{n+1} = f_i^n + \frac{\Delta\tau}{(\Delta X)^2}(g_{i+1}^n - 2g_i^n + g_{i-1}^n). \tag{10}$$

Question 4

Write a program to solve equation (9) taking $\eta = 1$ and with the boundary conditions $\sigma(r_{\rm in}, \tau) = 0$ and $\sigma(r_{\rm out}, \tau) = 0$. Set $r_{\rm in} = 0.0004$ and $r_{\rm out} = 4$ and use 100 grid points equally spaced in X between $X = 0.02$ and $X = 2$. For formal stability the timestep must satisfy

$$\Delta\tau \leqslant \frac{1}{2}(\Delta X)^2\frac{f}{g} \tag{11}$$

at all points in the disc. However you may find that you need to use something smaller. Evolve from an initial mass distribution

$$\sigma(r, 0) = \exp\left(-\frac{(r^{1/2} - 1)^2}{0.001}\right). \tag{12}$$

Plot the initial surface density, $\sigma(r, 0)$, against r. Plot σ against r at times $\tau = 0.002$, 0.008, 0.032, 0.128 and 0.512 on the same axes.

Question 5

Adapt your program so that you can find the height and position of the peak in the surface density. Make a table of the surface density at the peak and the position of the peak at the times used in question 4. Furthermore, adapt your program to plot the time evolution of the total angular momentum in the disc (normalised to its value at $t = 0$), and the position of the peak angular momentum surface density ($R^2\Omega\Sigma$) as a function of time.

Comment on the difference in behaviour between the surface density and angular momentum.

Question 6

The evolution of a particle in the disk is given by $dR/dt = V_{\rm R}(R, t)$ where $V_{\rm R}$ is given by equation 8. Rewrite equation 8 in a form similar to equation 10 so that you can adapt the code written for question 4 to plot radial velocity (in dimensionless units) as a function of radius for the timesteps used in question 4 on the same figure, taking care with choice of axis to show where this is positive and negative.

Question 7

Use the radial velocities found by your code to follow the evolution of particles' orbits, and plot that evolution up to $\tau = 1$ for particles initially at $r_0 = 0.9, 0.95, 1.0, 1.05, 1.1$. Plot the maximum radius attained by a particle as well as the time it took to reach that distance and, for those particles that do so, the time it takes to reach a boundary as a function of its initial radius for the range $r_0 = 0.9 - 1.1$.

Question 8

Use the results from question 7 to work out the range of initial radii r_0 that have reached the inner boundary by $\tau = 0.512$ and hence, using equation 12, the fraction of the initial mass that has reached the inner boundary by this time. Also do the corresponding calculation for the outer boundary. Compare these values with the fraction of the initial mass that remains at that time that you derive using the surface density profile from question 4.

MATLAB Specific Issues

Use of the MATLAB in-build function GRADIENT is prohibited, as its algorithm or accuracy is not documented.

