

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 2021-22. At the time of writing (July 2021) the booklet for the academic year 2020-21 states that

No questions on the Computational Projects are set on the written examination papers, credit for examination purposes being gained by the submission of reports. The maximum credit obtainable is 150 marks and there are no alpha or beta quality marks. Credit obtained is added directly to the credit gained on the written papers. The maximum contribution to the final merit mark is thus 150, which is the same as the maximum for a 16-lecture course. The Computational Projects are considered to be a single piece of work within the Mathematical Tripos.

1.2 The nature of CATAM projects

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

The questions posed in the projects are more open-ended than standard Tripos questions: **there is not always a single ‘correct’ response**, and often the method of investigation is not fully specified. **This is deliberate.** Such an approach allows you both to demonstrate your ability to use your own judgement in such matters, and also to produce mathematically intelligent, relevant responses to imprecise questions. 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 by the second week of 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 §5.2 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 **Managed Cluster Service** (MCS) that is available at a number of **UIS and institutional sites** around the Collegiate University.

1.4.1 Your copy of MATLAB

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

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

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

1.4.2 Programming guides and manual[s]

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

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

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

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

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

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

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

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

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

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

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

- There is a plethora of books on MATLAB. For instance:
 - (a) *Numerical Computing with MATLAB* by Cleve Moler (SIAM, Second Edition, 2008, ISBN 978-0-898716-60-3). This book can be downloaded for free from

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

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

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

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

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

1.4.3 To MATLAB, or not to MATLAB

Use of MATLAB is recommended,¹ but *you are free to write your programs in any computing language whatsoever*. Python, Julia,² R,³ C, C++, Mathematica,⁴ Maple⁵ and Haskell have been used by several students in the past, and Excel has been used for plotting graphs of computed results. The choice is your own, provided your system can produce results and program listings for inclusion in your report.⁶

However, you should bear in mind the following points.

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

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.

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

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

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

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

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

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

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

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.

The marks for each project are scaled so that a possible maximum of 150 marks are available for the Part II Computational Projects course. No quality marks (i.e. α s or β s) 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 N units, where $N > 30$ (i.e. if you submit more than the maximum number of units), then the following algorithm applies:

If your weakest project is M units with $M > N - 30$ then the mark on that project will be rescaled by $[M - (N - 30)]/M$. If $M \leq N - 30$ then that project will be discarded entirely, a revised N will be calculated, and the algorithm will be applied recursively.

This algorithm ensures that you can score no more than the overall maximum available, i.e. 150 Tripos marks, by reducing the mark only on your weakest project. There is no expectation that you submit more than 30 units; this algorithm is simply a way to calculate your mark if you do.⁷

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.
- 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*.⁸ If you are actually asked to *prove* a result, you should do so concisely.
- Graphs will sometimes be required, for instance to reveal some qualitative features of your results. Such graphs, *including labels, annotations, etc.*, need to be computer-generated

⁷ Needless to say, if you submit fewer than 30 units no upwards scaling applies.

⁸ See also the paragraph on *Citations* in §5

(see also § 2.3). Further, while it may be easier to include only one graph per page, it is often desirable (e.g. to aid comparison) to include *two or more* graphs on a page. Also, do not forget to clearly label the axes of graphs or other plots, and provide any other annotation necessary to interpret what is displayed. Similarly, the rows and columns of any tables produced should be clearly labelled.

- You should take care to ensure that the assessor sees evidence that **your programs do indeed perform the tasks** you claim they do. In most cases, this can be achieved by including a sample output from the program. If a question asks you to write a program to perform a task but doesn't specify explicitly that you should use it on any particular data, you should provide some 'test' data to run it on and include sample output in your write-up. Similarly, if a project asks you to 'print' or 'display' a numerical result, you should demonstrate that your program does indeed do this by including the output.
- **Above all, make sure you comment where the manual specifically asks you to.** It also helps the assessors if you answer the questions in the order that they appear in the manual and, if applicable, **number your answers** using the same numbering scheme as that used by the project. Make clear which outputs, tables and graphs correspond to which questions and programs.

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

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

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

2.2.1 Project write-ups: advice on length

The word *brief* peppers the last few paragraphs. 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.⁹ Indeed, the best reports are sometimes shorter than this.

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

2.3 Project write-ups: technicalities

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

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

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

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

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

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

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

⁹ Reports of projects with fewer/more units might be slightly shorter/longer.

¹⁰ Recall that graphs should not as a rule be printed one to a page.

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

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

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

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

T_EXstudio. For Windows, Mac and Linux users, there is T_EXstudio. The proT_EXt distribution, based on MiK_TE_X, includes the T_EXstudio front end.

T_EXworks. Again for Windows, Mac and Linux users, there is T_EXworks. The MiK_TE_X distribution includes T_EXworks.

T_EXShop. Many Mac aficionados use T_EXShop. To obtain T_EXShop and the T_EXLive distribution see <http://pages.uoregon.edu/koch/texshop/obtaining.html>.

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

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

<http://www.lyx.org/>.

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

Learning L^AT_EX. A *Brief L^AT_EX Guide for CATAM* is available for download from

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

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

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

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

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

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

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

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

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

Program listings. At the end of each report you should include complete **listings** of every major program used to generate your results. You do *not* need to include a listing of a program which is essentially a minor revision of another which you have already included. Make

sure that your program listings are the *very last* thing in your reports. Please do not mix program output and program listings together; if you do, the program output may not be marked as part of the report.

3 Computing Facilities

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

When permitted by COVID protocols, you can also use other computing facilities around the University; for further information (including which Colleges are linked to the MCS network) see¹¹

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

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

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

3.1 Backups

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

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

4 Information Sources

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

The CATAM Web Page. The CATAM web page,

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

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

¹¹ Note that the **Phoenix Teaching Rooms** and the **Titan Room** are used during term-times for practical classes by other Departments, but a list of these classes is posted at each site at the start of each term so that you can check the availability in advance (see **Opening Hours**).

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

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

After 1 October 2021 you can check that you are on the appropriate Faculty year list by referring to the <https://lists.cam.ac.uk/mailman/raven> webpage (to view this page you will need to authenticate using Raven if you have not already done so). You should check that the *Maths-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), 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.

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.

5 Unfair Means, Plagiarism and Guidelines for Collaboration

You **must** work *independently* on the projects, both on the programming and on the write-ups, i.e. you must write and test all programs yourself, and all reports must be written independently. It is recognised that some candidates may occasionally wish to discuss their work with others doing similar projects. This can be educationally beneficial and is accepted provided that it remains within reasonable bounds. However, any attempt to gain an unfair advantage, for example by copying computer code, mathematics, or written text, is not acceptable and will be subject to serious sanctions.

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

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

Acceptable collaboration. Acceptable collaboration may include an *occasional general discussion* of the approach to a project and of the numerical algorithms needed to solve it. Small hints on debugging code (note the *small*), as might be provided by an adviser, are also acceptable.

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

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

The following actions are examples of *unfair means*

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

These comments apply just as much to copying from the work of previous Part 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 you are in breach of these guidelines and may be charged with assisting another candidate to make use of unfair means.

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

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

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

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

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

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

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

Declarations. To certify that you have *read and understood* these guidelines, you will be asked to sign an electronic declaration at the start of the Michaelmas Term. You will receive an email with instructions as to how to do this at the start of the Michaelmas Term.

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

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

The assessors will not have knowledge of your declaration until after all your projects have been marked. However, your declaration may affect your CATAM marks if the assessors believe that discussions have gone beyond the limits of what is acceptable. If so, or if there is a suspicion that you have breached any of the other guidelines, you will be summoned

to an *Investigative Meeting* (see §5.2). Ultimately, your case could be brought to the University courts and serious penalties could result (see *Sanctions* below).

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

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

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

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

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

Sanctions. If plagiarism, collusion or any other method of unfair means is suspected in the Computational Projects, normally the Chair of Examiners will convene an *Investigative Meeting* (see §5.2). If the Chair of Examiners deems that unfair means were used, the case may be brought to the University courts. According to the Statutes and Ordinances of the University¹²

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

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

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

¹²From <https://www.admin.cam.ac.uk/univ/so/>.

5.2 Oral examinations

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

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

Investigative Meetings. When plagiarism, collusion or other unfair means are suspected the Chair of Examiners may summon a candidate to an *Investigative Meeting*¹³. If this happens, you have the right to be accompanied by your Tutor (or another representative at your request). The reasons for the meeting, together with copies of supporting evidence and other relevant documentation, will be given to your Tutor (or other representative). One possible outcome is that the case is brought to the University courts where *serious penalties can be imposed* (see *Sanctions* above).

Timing. *Viva Voce Examinations*, *Examination Interviews* and *Investigative Meetings* are a formal part of the Tripos examination, and if you are summoned then you must attend. 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 13th June 2022), 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 Full Term in case you are summoned.

6 Submission and Assessment

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

6.1 Submission form

When you submit your project reports you will be required to complete and upload 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*,

¹³ For more information see

https://www.plagiarism.admin.cam.ac.uk/files/investigative_2016.pdf.

and Appendix A). Further details, including the definitive submission form, will be made available when the arrangements for electronic submission of reports and programs (see below) are announced.

6.2 Submission of written work

In order to gain examination credit, you must:

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

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

The submission deadline is

Wednesday 4th May 2022, 4pm.

After this time, projects may be submitted only under exceptional circumstances. If an extension is likely to be needed, a letter of application and explanation is required from your Director of Studies. The application should be sent to the *CATAM Director* by the submission date as detailed above.¹⁴

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

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

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

6.3 Electronic submission

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

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

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

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

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

If you cannot remember your UIS password you will need to follow that instructions provided by the University Information Service.¹⁵ Note that if you need a Password Reset Token then this may take some time to obtain, so check that you know your UIS password well before submission day.

6.4 Saving and sharing electronic files

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

Since the manuals will be taken off-line after the close of submission, you might also like to save a copy of the projects you have attempted.

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

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

6.5 Returning from intermission

If a student is returning from intermission during an academic year where they submitted some or all of the CATAM projects, the Director of Studies should contact the CATAM helpline (catam@maths.cam.ac.uk) and the Undergraduate Office (undergrad-office@maths.cam.ac.uk).

¹⁵ See <https://password.csx.cam.ac.uk/forgotten-passwd>.

A Appendix: Example Submission Form

PART II MATHEMATICAL TRIPOS 2021-22

Computational Projects 2022

COMPUTATIONAL PROJECTS

STATEMENT OF PROJECTS SUBMITTED FOR EXAMINATION CREDIT

Please observe these points when submitting your CATAM projects:

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

IMPORTANT

Candidates are reminded that Discipline Regulation 7 reads:

No candidate shall make use of unfair means in any University examination. Unfair means shall include plagiarism¹⁶ and, unless such possession is specifically authorized, the possession of any book, paper or other material relevant to the examination. No member of the University shall assist a candidate to make use of such unfair means.

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

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

¹⁶ Plagiarism is defined as submitting as one's own work, irrespective of intent to deceive, that which derives in part or in its entirety from the work of others without due acknowledgement.

DECLARATION BY CANDIDATE

I hereby submit my 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