IB CATAM -- Intro Project Lecture

Rob Jack, rlj22@cam.ac.uk

things you already did / already know

You will need to hand in two CATAM core projects in Jan
... and two additional projects in May

You are competent with programming by now
(you can choose which language, MATLAB is an option)

You have attempted the Introductory Project
Ideally you have written a report on it, as practice

You will need to hand in your code and your report.
Within the report, printouts of the code are included **at the end**

[*the assessor will usually not look at the code, unless they suspect something strange is going on*]

things you already know about the report

40% of marks for programming, 50% "for mathematics"

if you attempted the intro project you will know :

even if the programming is easy, writing it up takes time

you have to think about "what the question is really asking"

these are intentional...

(note, perfect programs will only get you 8 out of 20)

Today : mostly comments about "what to write about"

resources

https://www.maths.cam.ac.uk/undergrad/catam/IB/

. Assessor's model answer

. Student's example write-up

both of these have comments about mark scheme etc

they are different in style (perhaps not surprising),
but they are both excellent as reports

we will discuss some aspects of "the maths" and "the writing",
to illustrate general principles

writing style and explanations

Writing about mathematics is not easy !

If you are asked to "explain" or "discuss" then there would be many possible things to say, you have to make choices

You have read enough maths to judge what is a "good explanation", eg think about the lecture notes / books / blogs that you like to read, and what makes them readable

A few guidelines :

ask yourself if your explanation would make sense to a typical IB student

don't be afraid to "state the obvious"

read back what you wrote and edit, then read and edit again, then put it aside for 3 days before reading again, etc...

format

The following slides contain a mixture of

quotes from model answer + student answer

parenthetical notes from me, mostly in [square brackets]

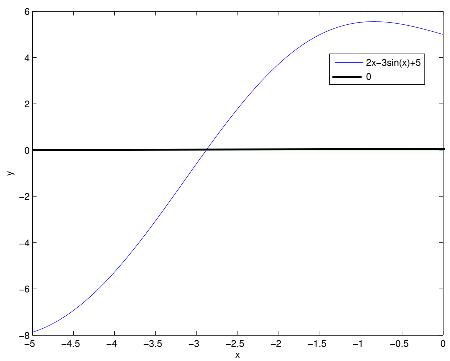
general comments on Computational Projects

Q1 Show graphically that $2x - 3\sin x + 5$ has exactly one root

[student answer]

To show graphically that equation $2x - 3\sin(x) + 5 = 0$ has exactly one root, plot $y = 2x - 3\sin(x) + 5$ and $y = 0$ and show that these two lines have only one point of intersection. I will plot the graph in the range $x \in [-5, 0]$. This is because, for $x < -5, 2x + 5 < 3\sin(x)$, so $y = 2x - 3\sin(x) + 5 < 0$ and for $x > 0, |3\sin(x)| < 2x + 5$, so $y = 2x - 3\sin(x) + 5 > 0$.

Thus $y < 0$ for $x < -5$, and $y > 0$ for $x > 0$, so no root can lie outside the range [-5,0].



0.5 mark for programming

1 mark for "maths" explanation eg, why do we only plot this range

[marking is out of 20, then it gets scaled]

Q2 Binary search

[model answer]

Table 1: Binary search results

Initial Interval	Number of Iterations	Final Iterate	Bound on Truncation Error
$[-3.0, -2.0]$	18	$x_{18} = -2.8832359\ldots$	$\pm 0.0000038\ldots$
$[-\pi, -5/2]$	17	$x_{17} = -2.8832414\ldots$	$\pm 0.0000049\ldots$
$[-5\pi/4, -3\pi/4]$	19	$x_{19} = -2.8832397\ldots$	$\pm 0.0000030\ldots$

A program using binary search to solve $2x + 5 - 3\sin x = 0$ is listed on page 11. Results for a representative number of runs are listed in table 1.

[1.5 programming marks]

Ordinarily, the assessor will not "read the program"

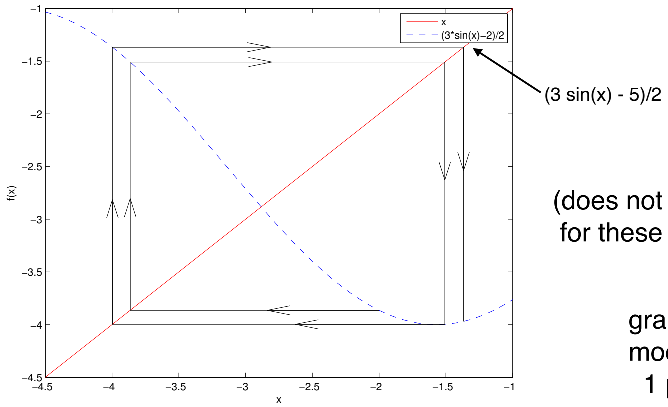
Hence, the *report must have a clear record (with evidence)* of what you did

The assignment specifically requests output for several initial values, so it is not enough to say (eg):
"I checked several initial intervals and obtained consistent results"

Q3 Fixed point iteration

$$x_{N+1} = f(x_N), \qquad f(x) = \frac{3\sin x + kx - 5}{2+k}$$

Does the resulting sequence converge to the root of $2x - 3\sin x + 5$?



(does not converge for these parameters)

graph from model answer. 1 programming mark

Figure 2: The first few iterations based on (1) with $k = 0$ and $x_0 = -2$.

Q3 Explain the [lack of convergence] by ...

[student answer]

$$\begin{aligned} x_{N+1} = f(x_N) &= f(x_*) + (x_N - x_*)f'(x_*) + ... \qquad [\text{ Taylor }] \\ &\approx x_* + (x_N - x_*)f'(x_*) \end{aligned}$$

Recall that the truncation error in the N^{th} iterate is e_N. [$e_N = x_N - x_*$]

$$\Rightarrow e_{N+1} \approx e_N f'(x_*)$$

Thus the iteration will diverge if $|f'(x_*)| = \frac{|3\cos(x_*)|}{2} > 1$,

[errors grow in magnitude when x_N is close to the root]

1 "maths" mark for this *explanation*

A "rigorous proof" might need more detail and precision, not required here (you could additionally cite a textbook for a more rigorous argument or a general theorem).

Q3 Also run the case $k = 16$. This should converge only slowly, so set $N_{max} = 50$. Discuss whether the truncation error is expected to be less than 10^{-5} in this case?

[we stop iterating when $|x_N - x_{N-1}| < \epsilon$ with $\epsilon = 10^{-5}$, the question is whether the resulting estimate of the root has an error less than ϵ.]

It says "discuss" so it is not sufficient to simply answer "no" !

[from student answer, here is part of a "maths" explanation (1 mark):]

$$\begin{aligned} x_N = f(x_{N-1}) &= f(x_*) + (x_{N-1} - x_*)f'(x_*) + ... \\ &\approx x_* + (x_{N-1} - x_*)f'(x_*) \end{aligned}$$

Thus

$$\begin{aligned} x_N - x_{N-1} &\approx (x_* - x_{N-1}) + (x_{N-1} - x_*)f'(x_*) \\ &= (x_{N-1} - x_*)(f'(x_*) - 1) \\ &\approx (f'(x_*) - 1)\{\frac{x_N - x_*}{f'(x_*)}\} \end{aligned}$$

[subtract x_{N-1} from both sides]

[Hence, at stopping point

$$\Rightarrow (x_N - x_*) \approx (x_N - x_{N-1})\{\frac{f'(x_*)}{f'(x_*) - 1}\}$$

$$|e_N| \lesssim \frac{\epsilon|f'(x_*)|}{|f'(x_*) - 1|}$$]

Q3 Discuss whether your results are consistent with first-order convergence.

[from student answer]

first-order convergence if $|e_N|/[e_{N-1}|$ tends to some constant, $C < 1$.

[data for $k = 4$]

N	xN	|eN/eN-1|
1	-2.6213154	
2	-2.8294373	0.2054035
3	-2.8731801	0.1869296
4	-2.8813873	0.1839113
5	-2.8828977	0.1833783
6	-2.8831747	0.1832765
7	-2.8832255	0.1832313
8	-2.8832348	0.1830783

$$e_{N+1} \approx e_N f'(x_*)$$

Data (left) and theory (right) are *both* consistent with 1st order convergence

Even more : the theory predicts (correctly) the limit of the sequence in the right column (this was not checked in the student report but it is in the model answer, also other nice observations, worthy of "excellence marks")

Q4 apply the method to the double root of $F(x) \equiv (x - \frac{1}{2})(x-4)^2$

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

This time it does not say "discuss" but it is *still not sufficient* to simply answer "no" and "no" !

[from student answer]

[Taylor again, use $f'(x_*) = 1$ at double root]

$$
\begin{aligned}
x_N = f(x_{N-1}) &= f(x_*) + (x_{N-1} - x_*)f'(x_*) + \frac{1}{2}f''(x_*)(x_{N-1} - x_*)^2 + ... \\
&\approx x_{N-1} + \frac{1}{2}f''(x_*)(x_{N-1} - x_*)^2 \\
&= x_{N-1} + \frac{1}{2}f''(x_*)(e_{N-1})^2
\end{aligned}
$$

Thus convergence will be slow

[$e_N \approx e_{N-1} + f''(x_*)e_{N-1}^2/2$
so $(e_N/e_{N-1}) \to 1$,
not first order]

[Note : if you consider Taylor up to first order then you get $x_N = x_{N-1}$, this should be a hint that you need to go to higher order]

Q4 apply the method to the double root of $F(x) \equiv (x - \frac{1}{2})(x-4)^2$

repeating: $x_N \approx x_{N-1} + \frac{1}{2}f''(x_*)e_{N-1}^2$

[from student answer]

If the iterations are terminated when $|x_N - x_{N-1}| < \epsilon$,

[used $\epsilon = 10^{-5}$]

$$|e_N| \approx |\sqrt{\frac{2\epsilon}{f''(x_*)}}| = |\sqrt{\frac{40\epsilon}{7}}| \approx 0.007559$$

```
>> q4_fixedpoint2(5,1000)
N= 740, xN= 4.0075342
```

[estimating root numerically]

Theory and numerical result agree rather well, *evidence* that all is correct

Note ϵ/x_* must be at least machine precision ($\sim 10^{-15}$) so best possible relative error is of order $\sqrt{\epsilon} \sim 10^{-7}$.

For single roots it would be of order ϵ (much more accurate)

Q5 Newton Raphson method

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

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

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

This question basically wants you to repeat the analysis of Q3 and Q4 for a different method

The main trap is : the question is short, but doesn't mean that the answer will be short

(you need to re-consider several questions from previous sections)

[2.5 "programming" and 2.5 "maths" marks are available]

Summary (so far)

You need to provide *evidence* that you have written the relevant programs and obtained the output required

Graphs are often good for this
(or tables, or sequences of numbers)

[this gets you the "programming" marks]

You need to *explain* the underlying mathematics

[this gets you the "maths" marks]

Don't be afraid to write equations

Simpler explanations are usually better than complicated ones
(as long as they actually work)

If you can **verify consistency** of mathematical "theory" and computational "experiment", this is evidence that everything is working

Other things to remember

[see also "excellence" marks]

Mathematical writing is designed to be read by humans
... you can make this easy by nice layout, clearly labelled graphs, properly formatted equations/tables, etc

Things that are "obvious" to you at time of writing may not be obvious if you read it again one week later (and they may not be obvious to the reader)
... be *self-critical* of your explanations

The assessors will only give marks for things that you write in the report. Make sure that you give evidence for the work that you did.

The are no marks for "good code", but good code will usually get you quicker to the right answer

You have many demands on your time, don't spend too long worrying about (eg) the difference between 17/20 and 19/20.

... finally

Please read carefully the project instructions, the hand-in instructions, etc

You will decide how much time you are prepared to spend on CATAM
... we recommend : don't leave it until the last minute

the two IB core projects may differ in time required

Do feel free to ask us questions using the catam helpline (although we are not going to tell you how you should answer the questions!)

We hope the Computational Projects will give you insight into the mathematics that you learn in other courses ... and you might even enjoy some of the projects.

good luck !