1.1 Golden Section Search for the Mode of a Function

This project is not strongly related to any particular course.

1 Definition

A function f is *unimodal* on $[a, b]$ with a maximum at $c \in [a, b]$ if f is strictly increasing on $[a, c]$ and strictly decreasing on $[c, b]$. A function f is unimodal on $[a, b]$ with a minimum at c if and only if $-f$ is unimodal with a maximum at c.

2 The golden search method

Let f be unimodal with a maximum and let x, y be points in $[a, b]$ with $a < x < y < b$. We wish to locate the mode (i.e. the unique maximum) of f while minimising the number of evaluations of f that are required; this will be useful in cases where it is very expensive or time-consuming to evaluate f. If we evaluate f at x and at y we can make use of the following implications:

(i) $f(x) \geqslant f(y) \quad \Rightarrow \quad$ mode lies in $[a, y]$

(ii) $f(x) \leqslant f(y) \quad \Rightarrow \quad$ mode lies in $[x, b]$

to narrow the region of search for the mode to a shorter interval within which we already have one evaluation of f. A second evaluation within the subinterval enables us to repeat this process. We terminate the algorithm when the size of the interval is smaller than twice the required precision.

A convenient way to do this, with an efficiency not much less than that of the theoretical optimal method, is to divide $[a, b]$ at points x, y such that

$$(y - a)/(b - a) = (\sqrt{5} - 1)/2 \tag{1}$$

and

$$b - x = y - a\,; \tag{2}$$

that is, the interval is divided from each end in "golden section".

Question 1 Prove that the subinterval in which the mode is deduced to lie is found to be already divided in golden section from one end by the point in its interior at which we already have a function evaluation.

3 Programming

Programming Task: Write a program to implement the golden section search (to locate modes that are either maxima or minima). Your program should prompt the user to enter the interval's boundaries and the required precision. Ensure also that your program does not evaluate f more often than is necessary. In your write-up explain the following points.

(i) How does your program deal with the possibility that $f(x) = f(y)$ on one or more iterative steps?

(ii) Is it preferable to use equation (1) or equation (2) to locate the point for the second function evaluation in each new subinterval, and why?

(iii) How would your program function if the mode's position were at an end-point of the original interval?

Question 2 As a check that you understand the method, first program it to find the position of the mode in $[0, 1]$ of the function

$$f(x) = 1 + x + x^2 - 4x^4$$

to some appropriate accuracy. Your output should include the mode, the number of iterations performed and an indication of how accurate your result is.

3.1 Computational cost

Consider the alternative (and more intuitively obvious) algorithm in which, instead of using (1) and (2), the subdivisions are defined by $x - a = \frac{1}{3}(b - a)$, $y - a = \frac{2}{3}(b - a)$.

Question 3 What is likely to be the most time-consuming part of either algorithm in a real-life problem. How would the number of numerical operations required for this alternative algorithm compare with that required for the golden section search algorithm. Give quantitative estimates if possible.

[Note that no additional computational work should be done to answer this question.]

4 Theoretical considerations

Question 4 What properties of the function $f(x)$ determine the numerical accuracy that is attainable?

Question 5 If the mode was located to some accuracy, what would be the corresponding accuracy in the height of the mode? How does your answer depend on the properties of $f(x)$?

5 Application: Optimization of turntable and tone-arm design

With the resurgence in sales of *vinyl records*, attention has returned to optimizing the design of record players and, in particular, the positioning of the turntable and tone-arm.

The playing region of a vinyl gramophone record is an annulus of inner radius r and outer radius R. The record spins about its centre S. Sound is picked up from grooves in the record by a *stylus*, which is mounted at the end P of a *tone-arm* QP of length l which is fixed at Q (a point beyond the outer perimeter of the record). The grooves run (almost) in concentric circles around S. The sound pick-up is optimal if the stylus points in the same direction as the groove at P. See Figure 1, a plan view.

The designer of the tone-arm has the following parameters under his control:

(i) the distance d of the tone-arm pivot Q from the centre S of the record;

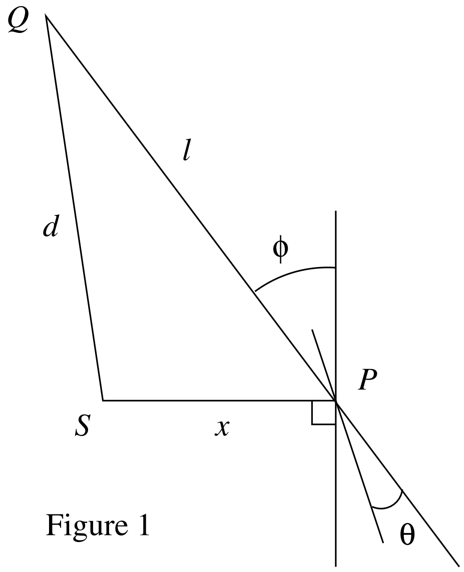

Figure 1

(ii) the toe-in angle θ between the tone-arm and the direction of the stylus.

The designer wishes the stylus to track so that its direction diverges as little as possible from alignment with the groove; that is, so that as the stylus moves from the outer edge of the record to the inner one, the maximum absolute value of $(\phi - \theta)$ is minimised, where ϕ is the angle between QP and the tangent to the groove at P.

> **Question 6** Taking $r = 6.5$, $R = 16$, $l = 24$ (all in cm), find the optimum values of d and θ using golden section search. Your program will need to carry out a double iteration. The inner iteration should find the variation of ϕ considered as a function of x (the distance SP); in other words, find $\Delta\phi = \max\phi(x) - \min\phi(x)$. The outer iteration then adjusts d to minimise $\Delta\phi$, and finally the optimum choice of θ can easily be made.
>
> Your write-up should address the following points:
>
> (i) What initial intervals did you use for the inner and outer iterations, and why?
>
> (ii) Is it clear that the functions you were minimising/maximising are unimodal on the relevant intervals?
>
> (iii) Approximately how many inner function evaluations are required to find d to one decimal place? How is the accuracy of your result affected by the accuracy with which you perform the inner and outer iterations?

Note that no extra marks are available for solving this part of the project analytically, though you may find that this is possible. No proofs are required at any stage: qualitative arguments supported by appropriate graphs will suffice.

Project 1.1: Golden Section Search for the Mode of a Function

Marking Scheme and additional comments for the Project Report

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

Question no.	**Marks available**[1]	**Marks awarded**[2]
Programming task *Program:* for instructions regarding printouts and what needs to be in the write-up, refer to the introduction to the manual.		
Question 1 *Comments:* An analytical solution is required to show that the subinterval is divided in golden section. Qualitative arguments are acceptable for (i), (ii) and (iii). For (iii), it is insufficient just to show that your program works. [*approx. 8 lines*][3]	C1.5+M2.5	
Question 2 *Comments:* Write the values of the mode, number of iterations and accuracy obtained from computation using your program.	C2+M0	
Question 3 *Comments:* Quantitative details are required for the comparison of the complexity of the golden section search algorithm and the alternative algorithm. No computations should be done. [*approx. 15 lines*][3]	C0+M1	
Question 4 *Comments:* Give quantitative arguments where possible. [*approx. 4 lines*][3]	C0.5+M1.5	
Question 5 *Comments:* No computations are required. [*approx. 5 lines*][3]	C0+M1	
Question 6 *Comments:* This must be solved computationally. Graphs can be used to show that the functions are unimodal. [*approx. 20 lines*][3]	C4+M4	
Excellence marks awarded for, *among other things*, mathematical clarity and good, clear output (graphs and tables) — see the introduction to the Project Manual.	E2	
Total Raw Marks	20	
Total Tripos Marks	**40**	

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

