THE 2017 TSINGHUA MATHCAMP COMPUTATIONAL MATHEMATICS PROJECTS (VERSION 1)

INSTRUCTOR: XIAOZHE HU

Project 1: (1) Consider solving the quadratic equation $ax^2 + bx + c = 0$ using the quadratic formula.

(a) Analyze possible loss of significance and propose alternative formula if necessary.

(b) Write a program to compute the roots. Your program should guard against unnecessary overflow, underflow, and loss of significance. Any root that is within the range of the floating point system (double precision) should be computed accurately, even if the other is out of range. (Possible test set of coefficients is listed as follows)

a	b	c
6×10^{154}	5×10^{154}	-4×10^{154}
0	1	1
1	-10^5	1
1	-4	3.999999
10^{-155}	-10^{155}	10^{155}

(2) Next consider the polynomials of degree greater than or equal to 3 ($n \geq 3$), i.e.

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{n-2}x^{n-2} + a_{n-1}x^{n-1} + a_n x^n.$$

Now numerical methods for finding the roots are preferred.

(a) When evaluating $p_n(x)$ at $x = z$ using the nested formulation, one can define $b_n = a_n$, $b_k = a_k + b_{k+1}z$, $k = n-1, n-2, \cdots, 1, 0$. We then use $b_1, \cdots, b_n$ to define another polynomial

$$q_{n-1}(x) = b_1 + b_2 x + \cdots + b_n x^{n-1}.$$

Show that $p_n(x) = b_0 + (x - z)q_{n-1}(x)$.

(b) Implement Newton's method to find one root of $p_n(x)$. Can you do that without explicitly compute $p_n'(x)$?

(c) Design an algorithm to compute all the roots of the polynomial $p_n(x)$ automatically (manually changing the initial guesses does not count), implement your algorithm on computer (possible test set of the polynomials: $p_3(x) = -2 - x + 2x^2 + x^3$ and $p_5(x) = 1 - 2x - x^4 + 2x^5$).

(3) Consider general functions $f(x)$ who may have several roots, design your algorithm to compute any many roots as possible.

Project 2: We consider a laboratory experiment. The experiment is begun by placing a mouse at one of the interior intersections of a maze with several exits. Once the mouse exits the maze, it cannot go back.

(1) Let us start with the first maze (see Figure 0.1). Assume that when the mouse is at an interior intersection, its choice of paths is assumed to be random. What is the probability that the mouse will find the "food" when it begins at the ith intersection?

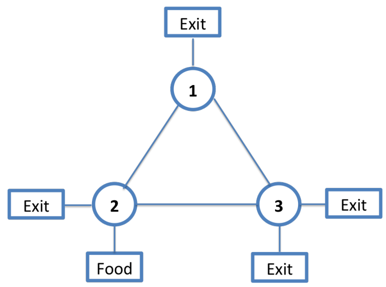

FIGURE 0.1. Maze 1

(2) How about we move the "food" to other exits? How does the probability change? (Try to write a program to help you compute)

(3) After several experiments, the mouse starts to remember that the "food" is located near one interior intersection. Then how do the probabilities change in this case?

(4) Now we switch to an 3×3 maze (see Figure 0.2). Again the mouse choose the paths at random. What is the probability that the mouse will find the "food" when it begins at the ith intersection?

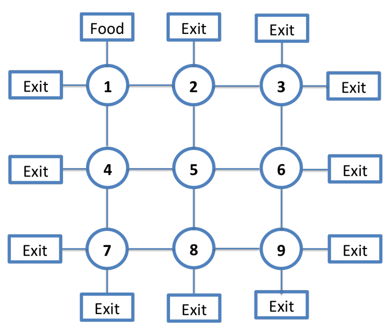

FIGURE 0.2. 3×3 maze

(5) Consider a $n \times n$ maze and the "food" is always placed at the top left corner as shown in Figure 0.2. The mouse choose the paths at random. What is the probability that the mouse will find the "food" when it begins at the ith intersection? (choose $n = 2, 4, 8, 16, 32, 64, 128$ and pay attention to the computational cost)

(6) Can you use the probabilities computed on the smaller maze to help you compute the probabilities on the bigger maze faster?

Project 3: Consider the following numerical integration

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \alpha f(-1) + \beta f(0) + \gamma f(1)$$

(1) Determine α, β, and γ such that the obtained numerical integration rule is exact for all quadratic polynomials, i.e., exact whenever $f(x) \in \mathcal{P}_2$.
(2) Show that this numerical integration rule is actually the unique quadrature formula of this form that is exact whenever $f(x) \in \mathcal{P}_4$.
(3) What will the corresponding quadrature formula be if the interval is $[a, b]$?
(4) Let us denote this numerical integration in the interval $[a, b]$ by I_a^b, can you estimate the error $\displaystyle\int_a^b f(x)\,\mathrm{d}x - I_a^b$?
(5) Let $c = \dfrac{a+b}{2}$, assume that $f^{(4)}(x)$ is approximately a constant in the interval $[a, b]$, estimate the relation between $|\displaystyle\int_a^b f(x)\mathrm{d}x - (I_a^c + I_c^b)|$ and $|(I_a^c + I_c^b) - I_a^b|$.
(6) Design adaptive quadrature algorithm based this quadrature formula and implement it on computer. Use it to approximate the integral

$$\int_1^{\pi} x^2 \sin(\omega x)\mathrm{d}x$$

where $\omega = 1, 3, 5$ with tolerance 10^{-6}. Report how many subintervales are needed for each case.
(7) Can you achieve similar accuracy using less subintervals?
(8) Can you design numerical integration for improper integrals? For example,

$$\int_1^{\infty} \frac{1}{1+x^2}\mathrm{d}\,x.$$

(9) Try to compute the following infinite series as accurate as you can (it was in Srinivasa Ramanujan's first letter to Gogfrey H. Hardy)

$$1 - 1! + 2! - 3! + \cdots = \int_0^{\infty} \frac{e^{-x}}{1+x}\mathrm{d}x = ?$$