MATH 3340 - Scientific Computing Assignment 8

Libao Jin

May 4, 2021

The deadline will be strictly enforced. If you do not submit in time there will be a 20% penalty for each day you're late. If you do not submit in time there will be a 20% penalty upfront plus another 20% for each day you're late. Remember that you are allowed to work in teams of two on this assignment. You are encouraged to prepare your work in LaTeX; a template will be provided to help you put it all together. If you choose to submit a hard copy, you may submit only one copy for a team, indicating the names of both contributors. Online submission is encouraged, however, in that case both members of a team should submit the PDF file containing their work and showing both their names.

All plots generated in this homework should have a title, legend, and labeled x and y-axes.

Instruction

1. Go to `https://www.overleaf.com` and sign in (required).
2. Click *Menu* (up left corner), then *Copy Project*.
3. Go to `LaTeX/meta.tex` (the file `meta.tex` under the folder `LaTeX`) to change the section and your name, e.g.,
   - change author to `\author{Albert Einstein \& Carl F. Gauss}`
4. For Problem 1 and 3, you are encouraged to type solutions in LaTeX. But if you want to write it on the printout, make sure your scanned work is *clear* enough, and compile all solutions *in order*, i.e., 1, 2, 3, in a single PDF (failure to do so will lead to points deduction).
5. For Problem 2 and 3, you need to write function/script files, store results to output files, and save graphs to figure files. Here are suggested names for function files, script files, output files, and figure files:

Problem	Function File	Script File	Output File	Figure File
2	`monteCarlo.m`	`hw8_p2.m`	`hw8_p2.txt`	
3	`goldenSection.m`	`hw8_p3.m`	`hw8_p3.txt`	`hw8_p3.pdf`

   Once finished, you need to upload these files to the folder `src` on Overleaf. If you have different filenames, please update the filenames in `\lstinputlisting{../src/your_script_name.m}` accordingly. You can code in the provided files in hw8.zip, and use the MATLAB script `save_results.m` to generate the output files and store the graphs to `.pdf` files automatically (the script filenames should be exactly same as listed above).
6. Recompile, download and upload the generated PDF to WyoCourses.
7. You may find LaTeX.Mathematical.Symbols.pdf and the second part of Lab 01 Slides and Lab 02 Slides helpful.

1 Problem 1

This computation should be done by hand. Use Romberg integration to compute the $R_{3,3}$ approximation for

$$I = \int_1^3 (x^3 - 1)e^{-x^2}\, dx. \tag{1.1}$$

Perform all calculations by rounding off to four decimal places.

Solution.

- Output file `hw8_p1.txt`:

```
I =
   0.232772618965638
R =
   0.003208654906254                   0                   0
   0.129813799674266   0.172015514596937                   0
   0.204184924875512   0.228975299942594   0.232772618965638
```

- Function file `romberg.m`:

```matlab
function [I, R] = romberg(f, a, b, n)

%ROMBERG
% Syntax: I = romberg(f, a, b, n)
% Inputs:
%   f = the integrand which is a function handle
%   a = the lower bound of the definite integral
%   b = the upper bound of the definite integral
%   n = number of subintervals
% Outputs:
%   I = the romberg integration of f(x) over the interval [a, b] using n subinterval
%   R = a lower triangular matrix
% Author: first_name last_name
% Date: 04/29/2020

h = b - a;
R = zeros(n);
R(1, 1) = (f(a) + f(b)) * h / 2;
for k = 2:n
    j = 1:2^(k - 2);
    R(k, 1) = 1 / 2 * (R(k - 1, 1) + h * sum(f(a + (2 * j - 1) * h / 2)));
    for j = 2:k
        R(k, j) = R(k, j - 1) + (R(k, j - 1) - R(k - 1, j - 1)) / (4^(j - 1) - 1);
    end
    h = h / 2;
end
I = R(end, end);

end
```

- Script file `hw8_p1.m`:

```matlab
% MATH 3340, Fall 2020
% Homework 8, Problem 1
```

```
% Author: first_name last_name
% Date: 11/23/2020

clear; close all; clc; format long;
% PUT YOUR CODE HERE
f = @(x) (x.^3 - 1) .* exp(-x.^2);
a = 1;
b = 3;
n = 3;
[I, R] = romberg(f, a, b, n)
format compact; format short;
```

□

2 Problem 2

Write a code that integrates the function $f(x, y, z) = 0.7(x^2 + y^2 + z^2)$ over the unit sphere $S = \{(x, y, z)|x^2 + y^2 + z^2 \leq 1\}$ using the Monte-Carlo method in three dimensions. Run the code with a sample of $M = 10^6$ points; do this ten times and compute the average of the ten results. Then compare this average result with the exact value of the integral, which can be easily calculated analytically. Print the error: the absolute value of the difference between the exact value and the average you got from your ten runs.

Solution.

- Output file `hw8_p2.txt`:

```
Monte Carlo integration for f(x, y, z) = 0.7 * (x^2 + y^2 + z^2) over the ball x^2 + y^2 + z^2 <= 1:
       I_monte_carlo              I_exact                 error
            1.758680             1.759292              0.000612
```

- Function file `monteCarlo.m`:

```
function I = monteCarlo(f, checker, xmin, xmax, ymin, ymax, zmin, zmax, M)
%MONTECARLO: Monte Carlo Integration of f(x, y)
% Input:
%       f: function handle, the integrand of the integral
% checker: function handle, set the function value to zero if the points (x, y, z) is outside the region
%    xmin: scalar value, lower bound of x coordinate of the bounding box for the region
%    xmax: scalar value, upper bound of x coordinate of the bounding box for the region
%    ymin: scalar value, lower bound of y coordinate of the bounding box for the region
%    ymax: scalar value, upper bound of y coordinate of the bounding box for the region
%    zmin: scalar value, lower bound of z coordinate of the bounding box for the region
%    zmax: scalar value, upper bound of z coordinate of the bounding box for the region
%       M: scalar value, number of samples
% Output:
%       I: the integral of f(x, y, z) over the specific region
% Author: Libao Jin
% Date: 11/23/2020

x = rand(M, 1) * (xmax - xmin) + xmin;
y = rand(M, 1) * (ymax - ymin) + ymin;
```

```
z = rand(M, 1) * (zmax - zmin) + zmin;
g = @(x, y, z) checker(x, y, z) .* f(x, y, z);
V = (zmax - zmin) * (ymax - ymin) * (xmax - xmin);
I = V / M * sum(g(x, y, z));

end
```

- Script file hw8_p2.m:

```
% MATH 3340, Fall 2020
% Homework 8, Problem 2
% Author: first_name last_name
% Date: 11/23/2020

clear; close all; clc;
rng(1204);

% PUT YOUR CODE HERE
test = 0;
if test == 1
    f = @(x, y, z) x.^3 + y.^2 + z;
    xmin = -2;
    xmax =  2;
    ymin = -2;
    ymax =  2;
    zmin = -2;
    zmax =  2;
    r = (xmax - xmin) / 2;
    % checker = @(x, y, z) x.^2 + y.^2 + z.^2 <= ((xmax - xmin) / 2)^2;
    checker = @(x, y, z) -sqrt(r^2 - x.^2) <= y & y <= sqrt(r^2 - x.^2) & -sqrt(r^2 - x.^2 -
         y.^2) <= z & z <= sqrt(r^2 - x.^2 - y.^2);
else
    f = @(x, y, z) 0.7 * (x.^2 + y.^2 + z.^2);
    xmin = -1;
    xmax =  1;
    ymin = -1;
    ymax =  1;
    zmin = -1;
    zmax =  1;
    r = (xmax - xmin) / 2;
    checker = @(x, y, z) -sqrt(r^2 - x.^2) <= y & y <= sqrt(r^2 - x.^2) & -sqrt(r^2 - x.^2 -
         y.^2) <= z & z <= sqrt(r^2 - x.^2 - y.^2);
end
M = 1e6;
n = 10;
I = 0;
for i = 1:n
    I = I + monteCarlo(f, checker, xmin, xmax, ymin, ymax, zmin, zmax, M);
end
I_monte_carlo = I / n;

r = (xmax - xmin) / 2;
xmin2 = xmin;
xmax2 = xmax;
ymin2 = @(x) -sqrt(r^2 - x.^2);
```

```
ymax2 = @(x) sqrt(r^2 - x.^2);
zmin2 = @(x, y) -sqrt(r^2 - x.^2 - y.^2);
zmax2 = @(x, y) sqrt(r^2 - x.^2 - y.^2);
I_exact = integral3(f, xmin2, xmax2, ymin2, ymax2, zmin2, zmax2);

err = abs(I_monte_carlo - I_exact);

if test == 1
    fprintf('Monte Carlo integration for f(x, y, z) = x^3 + y^2 + z over the ball x^2 + y^2 + z^2 <= %d: (REPLACE THIS WITH YOUR OWN OUTPUT)\n', r^2);
else
    fprintf('Monte Carlo integration for f(x, y, z) = 0.7 * (x^2 + y^2 + z^2) over the ball x^2 + y^2 + z^2 <= %d:\n', r^2);
end
fprintf('%20s %20s %20s\n', 'I_monte_carlo', 'I_exact', 'error');
fprintf('%20.6f %20.6f %20.6f\n', I_monte_carlo, I_exact, err);
```

□

3 Problem 3

Write a MATLAB function that implements the golden search method. Use this function to find the minimum of the function $f(x) = \cos(x) - \sin(x)$ on the interval $[1, 3]$ with a tolerance $T = 10^{-7}$. You should first plot the function and check that it is unimodal on this interval. Also, find the number of iterations needed to locate the value of the minimum with a tolerance of at least $T = 0.1$. Do the latter calculation by hand.

Solution.

- The number of iterations needed to locate the value of the minimum with a tolerance of at least $T = 0.1$:

- Output file hw8_p3.txt:

```
f(x) = cos(x) - sin(x) reaches the minimum -1.41421356 at x = 2.35619446.
```

- Figure file hw8_p3.pdf:

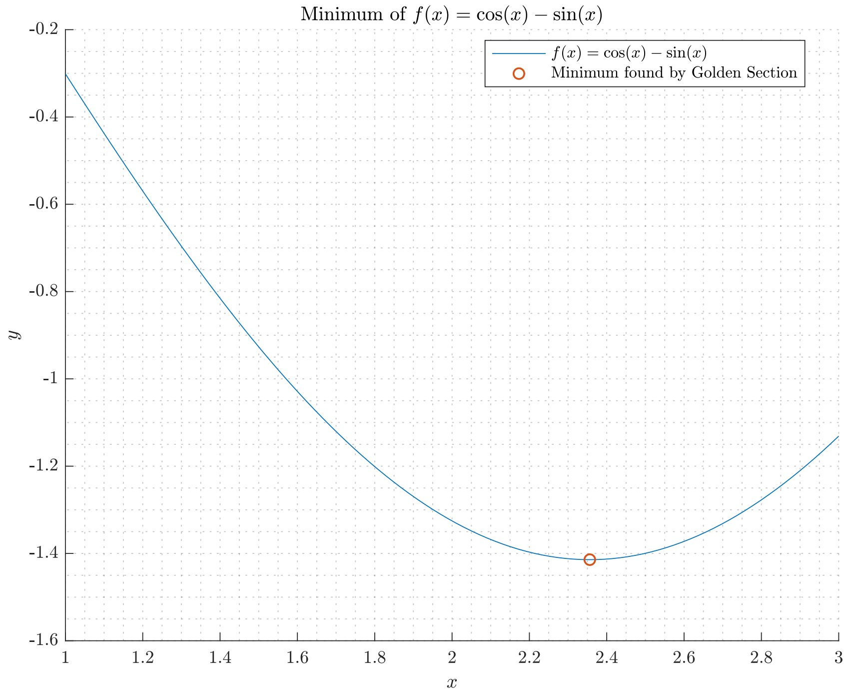

Figure 1: $f(x)$ is unimodal.

- Function file `goldenSection.m`:

```
function [m, fm] = goldenSection(f, a, b, tol)

%GOLDENSECTION
% Syntax: [m] = goldenSection(f, a, b, tol)
% Inputs:
%   f   = function of which the minimum is desired, function handle
%   a   = left endpoint of the interval
%   b   = right endpoint of the interval
%   tol = tolerance
% Outputs:
%   m   = the minimizer at which the minimum of f(x) can be obtained
%   fm  = the minimum of the function
% Author: first_name last_name
% Date: 04/29/2020

r = (3 - sqrt(5)) / 2;
s = 1 - r;
it = 0;
while abs(b - a) > tol
    m1 = a + r * (b - a);
    m2 = a + s * (b - a);
    if f(m1) < f(m2)
        b = m2;
    else
        a = m1;
```

```
    end
    it = it + 1;
end
m = a;
fm = f(m);
it;
end
```

- Script file hw8_p3.m:

```
% MATH 3340, Fall 2020
% Homework 8, Problem 3
% Author: first_name last_name
% Date: 11/23/2020

clear; close all; clc;
set(groot,'defaulttextinterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex');

% PUT YOUR CODE HERE
test = 0;
if test == 1
    f = @(x) x.^2 + 4 * x + 3;
    a = -3;
    b = 1;
else
    f = @(x) cos(x) - sin(x);
    a = 1;
    b = 3;
end
tol = 1e-7;

[m, fm] = goldenSection(f, a, b, tol);

if test == 1
    fprintf('f(x) = x^2 + 4*x + 3 reaches the minimum %.8f at x = %.8f. (REPLACE THIS WITH
        YOUR OWN OUTPUT)\n', fm, m);
else
    fprintf('f(x) = cos(x) - sin(x) reaches the minimum %.8f at x = %.8f.\n', fm, m);
end

x = linspace(a, b);
y = f(x);
figure(1); hold on;
plot(x, y);
plot(m, fm, 'o');


grid minor;
xlabel('$x$');
ylabel('$y$');

if test == 1
    legend({'$f(x) = x^{2} + 4x + 3$', 'Minimum found by Golden Section'}, 'Location', 'best
```

```
        ');
    title('Minimum of $f(x) = x^{2} + 4x + 3$ (REPLACE THIS WITH YOUR OWN PLOT)');
else
    legend({'$f(x) = \cos(x) - \sin(x)$', 'Minimum found by Golden Section'}, 'Location', 'best');
    title('Minimum of $f(x) = \cos(x) - \sin(x)$');
end

r = (3 - sqrt(5)) / 2;
s = 1 - r;
tol = 0.1;
n = log(tol * 2 / (b - a)) / log(s);
```

□