MATH 3340 - Scientific Computing Assignment 7

Libao Jin

November 11, 2020

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 2, 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 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
3	`gauss_quad.m`	`hw7_p3.m`	`hw7_p3.txt`	

   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 hw7.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.

Problem 1. Compute, by hand, the value of

$$\int_0^{2\pi} x^2 \sin^2(x)\, dx$$

using both the trapezoidal rule and Simpson's rule. For a fair comparison, keep the same number of function evaluation, in this case at five equi-spaced $\{x_0, x_1, x_2, x_3, x_4\}$. Use the same round-off strategy as in the first problem, keeping a minimum of four decimal places in all your calculations.

Solution.

- Output file `hw7_p1.txt`:

```
    n           a           b  Trapezoidal      Simpson        Exact
    5    0.000000    6.283185    38.757846    51.677128    39.770906
```

- Function file `trapezoidal.m`:

```
function [I] = trapezoidal(f, a, b, n)
%TRAPEZOIDAL: Trapezoidal Rule
% Syntax: [I] = trapezoidal(f, x, h, M)
% Inputs:
%   f = a function handle, the function of which the derivative will be evaluated
%   a = a scalar, the lower limit
%   b = a scalar, the upper limit
%   n = a scalar, the number of points
% Outputs:
%   I = a scalar, the integral of f over a to b using n points
%
% Author: first_name last_name
% Date: 04/20/2020

x = linspace(a, b, n);
dx = x(2) - x(1);
N = length(x) - 1;
I = 0;
for i = 1:N
    I = I + dx / 2 * (f(x(i)) + f(x(i + 1)));
end

end
```

- Function file `simpson.m`:

```
function [I] = simpson(f, a, b, n)
%SIMPSON: Simpson's Rule
% Syntax: [I] = simpson(f, a, b, n)
% Inputs:
%   f = a function handle, the function of which the derivative will be evaluated
%   a = a scalar, the lower limit
%   b = a scalar, the upper limit
%   n = a scalar, the number of points
% Outputs:
%   I = a scalar, the integral of f over a to b using n points
%
```

```
% Author: first_name last_name
% Date: 04/20/2020

x = linspace(a, b, n);
dx = x(2) - x(1);
N = length(x) - 2;
I = 0;
for i = 1:2:N
    I = I + dx / 3 * (f(x(i)) + 4 * f(x(i + 1)) + f(x(i + 2)));
end

end
```

- Script file hw7_p1.m:

```
% MATH 3340, Fall 2020
% Homework 7, Problem 1
% Author: Libao Jin
% Date: 11/11/2020

clear; close all; clc;
% Change default text interpreter to LaTeX
set(groot,'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex')


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

n = 5;
I_t = trapezoidal(f, a, b, n);
I_s = simpson(f, a, b, n);
I_e = integral(f, a, b);

fprintf('%5s %12s %12s %12s %12s %12s\n', 'n', 'a', 'b', 'Trapezoidal', 'Simpson', 'Exact');
fprintf('%5d %12.6f %12.6f %12.6f %12.6f %12.6f\n', n, a, b, I_t, I_s, I_e);
```

□

Problem 2. This computation should again be done by hand. Use Gauss quadrature with $N = 2$, $N = 3$ and $N = 4$ to compute the approximate value for

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

Perform all calculations by rounding off to four decimal places.

Solution.

- Gauss quadrature with $N = 2$:
- Gauss quadrature with $N = 3$:
- Gauss quadrature with $N = 4$:

$\square$

Problem 3. Write a MATLAB code that implements the Gauss quadrature calculation in the problem above, but allows for a wider range of values of N, from $N = 1$ through $N = 5$.

Solution.

- Output file `hw7_p3.txt`:

```
Summary for f(x) = (x^3 - 1) * e^(-x^2):
    N    I_gauss    I_exact      error
    1   0.256419   0.227879   0.028540
    2   0.269340   0.227879   0.041461
    3   0.223064   0.227879   0.004815
    4   0.227980   0.227879   0.000101
    5   0.227895   0.227879   0.000015
```

- Function file `richardson.m`:

```
function [I] = gauss_quad(f, a, b, N)
%gauss_quad: Gauss Quadrature on a general interval [a, b]
% Syntax: [I] = gauss_quad(f, a, b, N)
% Inputs:
%   f = the integrand (function handle)
%   a = lower limit of the integral (scalar)
%   b = upper limit of the integral (scalar)
%   N = the number of Gauss nodes (scalar)
% Outputs:
%   I = the values of the integration (scalar)
%
% Author: Libao Jin
% Date: 11/11/2020

% PUT YOUR CODE HERE
g = @(x) f((b - a) / 2 * x + (b + a) / 2) * (b - a) / 2;
[x, w] = legendre_pair(N);
I = dot(w, g(x));

end
```

- Script file `hw7_p3.m`:

```
% MATH 3340, Fall 2020
% Homework 7, Problem 3
% Author: first_name last_name
% Date: 11/11/2020

clear; close all; clc;
% Change default text interpreter to LaTeX
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 + cos(x);
    a = -2;
```

```
    b = 2;
    N = 5;
else
    f = @(x) (x.^3 - 1) .* exp(-x.^2);
    a = 1;
    b = 3;
    N = 5;
end

n = 1:N;

for i = 1:length(n)
    I(i) = gauss_quad(f, a, b, n(i));
    I_exact(i) = integral(f, a, b);
    error(i) = abs(I(n(i)) - I_exact(n(i)));
end

if test == 1
    fprintf('Summary for f(x) = x^2 + cos(x): (REPLACE THIS FILE WITH YOUR OWN)\n');
    fprintf('%5s %10s\n', 'N', 'I_gauss');
    for i = 1:length(n)
        fprintf('%5d %10.6f\n', n(i), I(i));
    end
else
    fprintf('Summary for f(x) = (x^3 - 1) * e^(-x^2):\n');
    fprintf('%5s %10s %10s %10s\n', 'N', 'I_gauss', 'I_exact', 'error');
    for i = 1:length(n)
        fprintf('%5d %10.6f %10.6f %10.6f\n', n(i), I(i), I_exact(i), error(i));
    end
end
```

□