

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	<code>gauss_quad.m</code>	<code>hw7_p3.m</code>	<code>hw7_p3.txt</code>	

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 [\$\text{\LaTeX}\$.Mathematical.Symbols.pdf](#) and the second part of [Lab 01 Slides](#) and [Lab 02 Slides](#) helpful.

1 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:

1	n	a	b	Trapezoidal	Simpson	Exact
2	5	0.000000	6.283185	38.757846	51.677128	39.770906

- Function file trapezoidal.m:

```

1 function [I] = trapezoidal(f, a, b, n)
2 %TRAPEZOIDAL: Trapezoidal Rule
3 % Syntax: [I] = trapezoidal(f, x, h, M)
4 % Inputs:
5 %   f = a function handle, the function of which the derivative will be evaluated
6 %   a = a scalar, the lower limit
7 %   b = a scalar, the upper limit
8 %   n = a scalar, the number of points
9 % Outputs:
10 %   I = a scalar, the integral of f over a to b using n points
11 %
12 % Author: first_name last_name
13 % Date: 04/20/2020
14
15 x = linspace(a, b, n);
16 dx = x(2) - x(1);
17 N = length(x) - 1;
18 I = 0;
19 for i = 1:N
20     I = I + dx / 2 * (f(x(i)) + f(x(i + 1))));
21 end
22
23 end

```

- Function file simpson.m:

```

1 function [I] = simpson(f, a, b, n)
2 %SIMPSON: Simpson's Rule
3 % Syntax: [I] = simpson(f, a, b, n)
4 % Inputs:
5 %   f = a function handle, the function of which the derivative will be evaluated
6 %   a = a scalar, the lower limit
7 %   b = a scalar, the upper limit
8 %   n = a scalar, the number of points
9 % Outputs:
10 %   I = a scalar, the integral of f over a to b using n points
11 %
12 % Author: first_name last_name
13 % Date: 04/20/2020
14
15 x = linspace(a, b, n);
16 dx = x(2) - x(1);
17 N = length(x) - 2;

```

```

18 I = 0;
19 for i = 1:2:N
20     I = I + dx / 3 * (f(x(i)) + 4 * f(x(i + 1)) + f(x(i + 2)));
21 end
22
23 end

```

- Script file hw7_p1.m:

```

1 % MATH 3340, Fall 2020
2 % Homework 7, Problem 1
3 % Author: Libao Jin
4 % Date: 11/11/2020
5
6 clear; close all; clc;
7 % Change default text interpreter to LaTeX
8 set(groot, 'defaultTextInterpreter', 'latex');
9 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
10 set(groot, 'defaultLegendInterpreter', 'latex')
11
12
13 % PUT YOUR CODE HERE
14 test = 0;
15 if test == 1
16     f = @(x) x.^3 .* (cos(x).^3);
17     a = 0;
18     b = 4 * pi;
19 else
20     f = @(x) x.^2 .* (sin(x).^2);
21     a = 0;
22     b = 2 * pi;
23 end
24
25 n = 5;
26 I_t = trapezoidal(f, a, b, n);
27 I_s = simpson(f, a, b, n);
28 I_e = integral(f, a, b);
29
30 fprintf('%5s %12s %12s %12s %12s %12s\n', 'n', 'a', 'b', 'Trapezoidal', 'Simpson', 'Exact');
31 fprintf('%5d %12.6f %12.6f %12.6f %12.6f %12.6f\n', n, a, b, I_t, I_s, I_e);

```

□

2 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 $N = 4$:

□

3 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:

```

1 Summary for f(x) = (x^3 - 1) * e^(-x^2):
2   N   I_gauss   I_exact   error
3   1   0.256419   0.227879   0.028540
4   2   0.269340   0.227879   0.041461
5   3   0.223064   0.227879   0.004815
6   4   0.227980   0.227879   0.000101
7   5   0.227895   0.227879   0.000015

```

- Function file richardson.m:

```

1 function [I] = gauss_quad(f, a, b, N)
2 %gauss_quad: Gauss Quadrature on a general interval [a, b]
3 % Syntax: [I] = gauss_quad(f, a, b, N)
4 % Inputs:
5 %   f = the integrand (function handle)
6 %   a = lower limit of the integral (scalar)
7 %   b = upper limit of the integral (scalar)
8 %   N = the number of Gauss nodes (scalar)
9 % Outputs:
10 %   I = the values of the integration (scalar)
11 %
12 % Author: Libao Jin
13 % Date: 11/11/2020
14
15 % PUT YOUR CODE HERE
16 g = @(x) f((b - a) / 2 * x + (b + a) / 2) * (b - a) / 2;
17 [x, w] = legendre_pair(N);
18 I = dot(w, g(x));
19
20 end

```

- Script file hw7_p3.m:

```

1 % MATH 3340, Fall 2020
2 % Homework 7, Problem 3
3 % Author: first_name last_name
4 % Date: 11/11/2020
5
6 clear; close all; clc;
7 % Change default text interpreter to LaTeX
8 set(groot, 'defaultTextInterpreter', 'latex');
9 set(groot, 'defaultAxesTickLabelInterpreter', 'latex');
10 set(groot, 'defaultLegendInterpreter', 'latex')
11
12 % PUT YOUR CODE HERE
13 test = 0;
14 if test == 1

```

```
15     f = @(x) x.^2 + cos(x);
16     a = -2;
17     b = 2;
18     N = 5;
19 else
20     f = @(x) (x.^3 - 1) .* exp(-x.^2);
21     a = 1;
22     b = 3;
23     N = 5;
24 end
25
26 n = 1:N;
27
28 for i = 1:length(n)
29     I(i) = gauss_quad(f, a, b, n(i));
30     I_exact(i) = integral(f, a, b);
31     error(i) = abs(I(n(i)) - I_exact(n(i)));
32 end
33
34 if test == 1
35     fprintf('Summary for f(x) = x^2 + cos(x): (REPLACE THIS FILE WITH YOUR OWN)\n');
36     fprintf('%5s %10s\n', 'N', 'I_gauss');
37     for i = 1:length(n)
38         fprintf('%5d %10.6f\n', n(i), I(i));
39     end
40 else
41     fprintf('Summary for f(x) = (x^3 - 1) * e^(-x^2):\n');
42     fprintf('%5s %10s %10s %10s\n', 'N', 'I_gauss', 'I_exact', 'error');
43     for i = 1:length(n)
44         fprintf('%5d %10.6f %10.6f %10.6f\n', n(i), I(i), I_exact(i), error(i));
45     end
46 end
```

