DEPARTMENT OF ENGINEERING COURSEWORK

Term:	**Academic Year:**	**Module Code:**
Epiphany	2024-2025	ENGI 2211

Title:
Engineering Mathematics 2: Numerical Methods

Time Required:	It is expected that you should spend approximately **100** hours on this coursework assignment. This includes all learning related activities completed during the year (for example, attending lectures/workshops, completing Problem Sheets, etc).
Deadline(s) for submission:	Monday 20 January 2025 at 14:00hrs.
Date for feedback:	Monday 17 February 2025
Submission instructions:	• Your submission must be uploaded to Learn Ultra in advance of the deadline. • All submissions in the Department are electronic and no hard copy is required. • The maximum file size that can be accepted is 20 MB. • All submissions must be saved using the following naming convention: SURNAME-Firstname_ENGIXXXX.pdf E.g. **"BLOGGS-Joanne_ENGI2211_NM.pdf"**
Format:	• Reports should be submitted in PDF format. • Code files should be submitted in a zip file, together with the PDF report. • The report should not be longer than five sides of A4 and the minimum allowed font size is 12pt. Margins should be no smaller than 2 cm. Everything including tables and references has to be within the five sides of A4.
Penalties for non-compliance:	• All submissions must be received in the format specified by the coursework brief (this includes code files and data files). Submissions not in the correct file will not be marked. • Markers will mark the work up to the page limit, but any material submitted beyond the allowed page limit will not be marked.
Late submission:	In accordance with the Learning and Teaching Handbook : 6.2.5: Penalties for the Late Submission of Assessed Work - Durham University summative assessed work received late within five working days of the deadline will be capped at the module pass mark; work received more than five days of the deadline will not be marked and a mark of zero will be recorded.

Academic Integrity Guidance:	The Department of Engineering considers any attempt by a student to gain an unfair academic advantage through engaging in acts of academic misconduct ultimately diminishes the value of the degree sought and indicates a fundamental dishonestly which disrespects members of the learning community in the department. All potential incidents of suspected academic dishonesty shall be thoroughly investigated in accordance with the Learning and Teaching Handbook, Section 6.2.4: Academic Misconduct (sharepoint.com).
Use of Generative AI	Use of generative AI (gAI) and related technologies is only permitted for improving the readability of text and improving the efficiency of computer code. Use of generative AI (gAI) and related technologies should be compliant with our policy document entitled "Student Guidance on the Use of Generative Artificial Intelligence and Related Technology for the Department of Engineering". Any use of gAI or related technologies should be detailed in an acknowledgement section within your coursework submission document.
AHEP Learning Outcomes Assessed:	The Engineering Council sets the overall requirements for the Accreditation of Higher Education Programmes (AHEP) in engineering, in line with the UK Standard for Professional Engineering Competence (UK-SPEC). This assignment has been designed to assess the following AHEP Learning Outcomes: M1. Apply a comprehensive knowledge of mathematics, statistics, natural science and engineering principles to the solution of complex problems. M2. Formulate and analyse complex problems to reach substantiated conclusions. M3. Select and apply appropriate computational and analytical techniques to model complex problems, discussing the limitations of the techniques employed.
Instructions to Candidates:	You should not include a separate cover sheet. The assignment title and your full name must be presented at the top of the first page. Do not include your anonymous code. Your submission should be named using the following convention: SURNAME-firstname_ENGI2211_NM.pdf.

Revision:	

1 Introduction

In this assignment, you will learn more about quadrature rules and methods for hyperbolic problems.
This assignment consists of four MatLab codes that you have to implement and a report that you have to write.

2 Programs

In the next sections, the programs are described.

2.1 Gauss rules

The Gauss-Legendre rules are quadrature rules to integrate functions on the interval $[-1, 1]$. All Gauss-Legendre rules look like:

$$S = \sum_{i=1}^{n} f(x_i) w_i \ ,$$

where n is the number of points used in the rule, f is the function to integrate, x_i are the coordinates of the quadrature points and w_i are the weights.
For this assignment, you have to implement the first three rules with 2, 3 and 4 quadrature points respectively. The coordinates and the weights for the three rules are given in Tables 1, 2 and 3.

x_i	w_i
$-\sqrt{1/3}$	1
$+\sqrt{1/3}$	1

Table 1: Gauss-Legendre rule 1.

x_i	w_i
$-\sqrt{3/5}$	5/9
0	8/9
$\sqrt{3/5}$	5/9

Table 2: Gauss-Legendre rule 2.

x_i	w_i
$\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$
$-\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$
$\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18-\sqrt{30}}{36}$
$-\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18-\sqrt{30}}{36}$

Table 3: Gauss-Legendre rule 3.

The first task is to implement the Gauss-Legendre rules in MatLab. This requires you to write a MatLab function `gauss.m` with the following input-output format

```
function S = gauss( rule, f )
```

where `rule` is used to select the rule to use, only values 1 or 2 or 3 are acceptable, `f` is the function to integrate and `S` is the value of the quadrature rule. For example, suppose that we want to use `gauss.m` to approximate the integral of sin from -1 to 1 using the second rule, then the MatLab code should be:

```
S = gauss( 2, @sin )
```

Any change in the declaration of the function will result in zero marks being awarded for the corresponding part.

The second task is to complete Table 4 approximating the values of the integral $\int_{-1}^{1} f(x)\,dx$ using the Gauss-Legendre rules and also report their errors.

f	G-L rule 1	G-L rule 2	G-L rule 3	Error rule 1	Error rule 2	Error rule 3
x						
x^2						
x^3						
x^4						
x^5						
x^6						
x^7						
x^8						

Table 4: Comparison between different rules (answers to 3sf).

2.2 Double composite trapezoidal rule

In this task I would like that you implement the composite trapezoidal rule in Matlab to approximate the double integral:

$$\int_a^b \int_c^d f(x,y)\,dy\,dx \ .$$

In Chapter 6, we have seen the composite trapezoidal rule and on ULTRA you can find the corresponding code `trap_rule.m`. To extend the composite trapezoidal rule to double integrals, the shape of the panels changes from little segments to little rectangles.
This task requires you to write a MatLab function `double_trap_rule.m` with the following input-output format

```
function S = double_trap_rule( f, x, n, y, m )
```

where:

- `f` - Function to integrate
- `x` - Interval of integration for the x variable
- `n` - Number of panels in the x-direction
- `y` - Interval of integration for the y variable
- `m` - Number of panels in the y-direction
- `S` - Value of the quadrature rule

Any change in the declaration of the function will result in zero marks being awarded for the corresponding part.
For example, to approximate the integral

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

using 10 panels in each direction, the function `double_trap_rule.m` should be called as:

```
S = double_trap_rule( @(x,y)(sin(x)*sin(y)), [0,pi], 10, [0,pi], 10 )
```

Then using the code you have to fill in the Table 5 integrating the functions specified in the table on the domain $[0,1]^2$ for the values of n and m in the table.

$n=m$	$f(x,y)=1$	$f(x,y)=xy$	$f(x,y)=\sin(\pi x)\sin(\pi y)$
1			
5			
10			

Table 5: Integrals comouted using the double trapezoidal rule (answers to 3sf).

2.3 Analytical method for hyperbolic problems

In this task you have to use the method of separation of variables to find the expression for the analytical solution for the PDE problem:

$$u_{tt}(x,t) = c^2 u_{xx}(x,t), \quad x \in [0,L],\, t \geq 0\,,$$

with $L > 0$ and under the following boundary conditions

$$u(0,t) = 0, \quad u_x(L,t) = 0,$$

and with the initial conditions

$$u(x,0) = f(x), \quad u_t(x,0) = g(x),$$

with $f(x)$ and $g(x)$ functions left to the user to define.
The functions $f(x)$ and $g(x)$ are defined as trigonometric series of the form:

$$f(x) = \sum_{n=0}^{+\infty} A_n \sin(\lambda_n x),$$

$$g(x) = \sum_{n=0}^{+\infty} \lambda_n B_n \sin(\lambda_n x),$$

where the expression for λ_n can be found using the method of separation of variables. The values of the coefficients A_n and B_n are left to the user to define.
Then you have to write a MatLab function to compute the value of the solution at a given point (x, t) based on the expression of the analytical solution. The code must be named `hyperbolic_analytical.m` with the following input-output format

```
function u = hyperbolic_analytical(x, t, L, c, A, B)
```

where:

- `x` - Value for position variable x
- `t` - Value for time variable t
- `L` - Length of the domain
- `c` - Coefficient c
- `A` - Coefficient for the expansion of f
- `B` - Coefficient for the expansion of g
- `u` - Values of the computed solution at (x, t)

In practise, the series for $f(x)$ and $g(x)$ must be truncated in order to run the code. The function `hyperbolic_analytical` must be written in such a way that any finite number of coefficients for $f(x)$ and $g(x)$ can be specified.
Any change in the declaration of the function will result in zero marks being awarded for the corresponding part.
For example, to compute the solution at $(0.5, 1.0)$ for $c = 1$, $L = 1$ and with the following definitions for $f(x)$ and $g(x)$:

$$f(x) = 1\sin(\lambda_0 x) - 3\sin(\lambda_2 x),$$

$$g(x) = \lambda_0 1\sin(\lambda_0 x) + \lambda_1 2\sin(\lambda_1 x),$$

The function must be called in the following way:

```
u = hyperbolic_analytical(0.5, 1.0, 1.0, 1.0, [1, 0, -3], [1, 2])
```

Then you have to use the code to fill in Table 6:

t	x	u
0	0.2	
	0.5	
	0.9	
1	0.2	
	0.5	
	0.9	
2	0.2	
	0.5	
	0.9	

Table 6: Values of the solution of the hyperbolic problem (answers to 3sf).

To run the codes to fill in the Table 6, you must use the parameters listed in Table 7.

Parameter	Value
`L`	1
`c`	2
`T`	0.5
`A`	$[1, 2, 3]$
`B`	$[-1, 0, 2, -1]$

Table 7: Parameters for the codes.

2.4 Numerical method for hyperbolic problems

In this task, you have to implement the finite difference method for the hyperbolic problem:

$$u_{tt}(x,t) = c^2 u_{xx}(x,t), \quad x \in [a,b],\, t \in [0,T]\,,$$

with $T > 0$ and under the following boundary conditions

$$u(a,t) = 0, \quad u_x(b,t) = 0,$$

and with the initial boundary conditions

$$u(x,0) = f(x), \quad u_t(x,0) = g(x),$$

with $f(x)$ and $g(x)$ functions left to the user to define.
The code must be named `fdhyperbolic_neumann.m` with the following input-output format

```
function [x,t,u] = fdhyperbolic_neumann(a,b,n,T,m,c,f,g)
```

where:

- `a,b` - Extrema
- `n` - Number of subintervals in space
- `T` - Final time
- `m` - Number of steps in time
- `c` - Coefficient c
- `f` - Funciton f
- `g` - Function g
- `x` - Nodes of the partition
- `t` - Nodes in time
- `u` - Values of the computed solution at the nodes

Any change in the declaration of the function will result in zero marks being awarded for the corresponding part.
You can look at the code `fdhyperbolic.m` on ULTRA to help you to implement `fdhyperbolic_neumann.m`. For example, to compute the solution in the domain $[0,1] \times [0,0.5]$ with $n = 10$, $m = 20$, $c = 2$ and $f(x) = \sin(\pi x) + \sin(2\pi x)$ and $g(x) = 0$, the function must be called in the following way:

```
f = @(x)(sin(pi*x)+sin(2*pi*x));g = @(x)(0*x);
[x,t,u]=fdhyperbolic(0,1,10,0.5,20,2,f,g);
```

Then you have to use the codes `fdhyperbolic_neumann.m` and `hyperbolic_analytical` to fill in the Table 8, where the column u_{ref} is computed using `hyperbolic_analytical`, the column u is computed using `fdhyperbolic_neumann.m` and the column u_{err} is the error computed using the previous two column.

t	x	u_{ref}	u	u_{err}
0	0.2			
	0.5			
	0.9			
0.05	0.2			
	0.5			
	0.9			
0.5	0.2			
	0.5			
	0.9			

Table 8: Values of the solution and error of the hyperbolic problem (answers to 3sf).

To run the codes to fill in the Table 8, you must use the parameters listed in Table 9.

Parameter	Value
`a`	0
`b`	1
`L`	1
`n`	1000
`m`	1000
`c`	2
`T`	0.5
`f`	$\sin((\pi/2)x)$
`g`	$(2\pi/2)\sin((\pi/2)x))$

Table 9: Parameters for the codes.

3 Report

The report should not be longer than five sides of A4 and the minimum allowed font size is 12pt. Margins should be no smaller than 2 cm. Everything including tables and references has to be within the five sides of A4. The report should be in pdf format and named using the following convention: `SURNAME-Firstname_ENGI2211_NM.pdf`. All the files have to be submitted in one Zip file. The Zip file should contain the report in pdf format and the codes. The Zip file should be named using

the following convention: `SURNAME-Firstname_ENGI2211_NM.zip`. In the report, you have to include the tables described below that you can fill in using the results from your programs. **For the tables, report your answers to 3 significant figures.**
You can use the report to explain the methods you have implemented. In particular, you must include the following points in your report:

1. Explain your implementation of the codes.
2. Explain any technique used to make the code run faster.
3. Discuss engineering applications of the methods discussed.

Feel free to discuss any other aspect of your work that you consider interesting.

Durham University
Department of Engineering

DEPARTMENT OF ENGINEERING UG MARKING MATRIX

Honours %	Class I 86 - 100	 70 - 85	Class II (i) 60 – 69	Class II (ii) 50 - 59	Class III 40 - 49	Not Honours Standard (Fail) 30 – 39	 0 – 29
Gauss Legendre (5%)	Exceptional: well commented fast code with no warnings, delivering correct results.	Excellent: fast code with no warnings, delivering correct results.	Good: code with no warnings, delivering correct results.	Adequate: code with some warnings, delivering correct results.	Limited contribution: running code, but the results are not correct.	Very limited contribution: the code runs only in certain cases.	Unacceptable: the code does not run.
Double Trapezoidal (10%)	Exceptional: well commented fast code with no warnings, delivering correct results.	Excellent: fast code with no warnings, delivering correct results.	Good: code with no warnings, delivering correct results.	Adequate: code with some warnings, delivering correct results.	Limited contribution: running code, but the results are not correct.	Very limited contribution: the code runs only in certain cases.	Unacceptable: the code does not run.
Hyperbolic Analytical (10%)	Exceptional: well commented fast code with no warnings, delivering correct results.	Excellent: fast code with no warnings, delivering correct results.	Good: code with no warnings, delivering correct results.	Adequate: code with some warnings, delivering correct results.	Limited contribution: running code, but the results are not correct.	Very limited contribution: the code runs only in certain cases.	Unacceptable: the code does not run.
Hyperbolic Finite Differences (20%)	Exceptional: well commented fast code with no warnings, delivering correct results.	Excellent: fast code with no warnings, delivering correct results.	Good: code with no warnings, delivering correct results.	Adequate: code with some warnings, delivering correct results.	Limited contribution: running code, but the results are not correct.	Very limited contribution: the code runs only in certain cases.	Unacceptable: the code does not run.
Tables (15%)	Exceptionally clear and accurate table. Results rounded up to the requested digit. Clear caption.	Excellent: very clear and accurate table. Results rounded up to the requested digit.	Good: complete and accurate table. Results rounded up to the requested digit.	Adequate: complete and accurate table. Not all results rounded up not to the requested digit.	Limited contribution: some wrong values. *or* incomplete table.	Very limited contribution: tables with many wrong values or mostly incomplete.	Unacceptable: no table submitted or all results in the submitted table are incorrect.
Interpretation	Exceptional and	Excellent and complete	Good	Adequate	Limited	Very limited	Unacceptable

of results (40%)	complete interpretation capturing all of the required information. Good references to tables. Excellent use of figures and diagrams. The discussion is well supported using equations and mathematical expressions. Equation presented using an excellent style.	interpretation capturing all of the required information. Good references to tables. Very good use of figures and diagrams. Equation presented using a very good style	interpretation capturing the majority of the information. Good use of figures and diagrams. Equation presented using a good style	interpretation. Superficial analysis.	contribution: very superficial interpretation.	contribution: report containing little information.	interpretation.