

Assignment #1

Course: Numerical methods of differential equation

Name: Student name Major: XXX Student ID: 123002584

Date: April 2, 2023

Problem 1. Here is a question.

Answer. Here is the answer of the question.

Problem 2. Here is a proof question.

Proof. Here is the proof of the problem. □

Definition 1. *This is a definition.*

Proposition 1. *This is a proposition.*

Lemma 1 (Lemma). *This is a lemma.*

Theorem 1 (Theorem). *This is a Theorem.*

Proof. This is the proof environment. □

Corollary 1. *This is a corollary.*

Proposition 2 (Proposition). *This is a proposition.*

Remark 1. *This is a remark.*

Example 1. *This is an example.*

Table

This is an example of table, such as Table 1

Table 1 Table name						
A	$N = 3$	$N = 5$	$N = 7$	$N = 9$	$N = 11$	$N = 13$
B	1.5789	1.3478	1.0645	0.8780	0.7222	0.5942
C	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
D	7.2632	14.3913	21.0323	27.3171	30.9630	34.0870

Figures

This is an example of figure, as shown in the Figure 2.

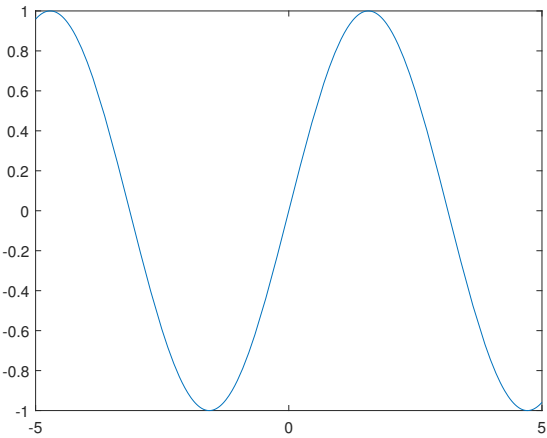


Figure 1 Image of function $y = \sin(x)$

Two pictures placed side by side, as shown in Figure 2 and Figure 3.

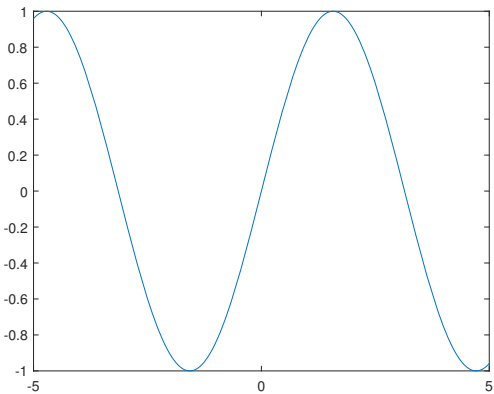


Figure 2 Description of image 1

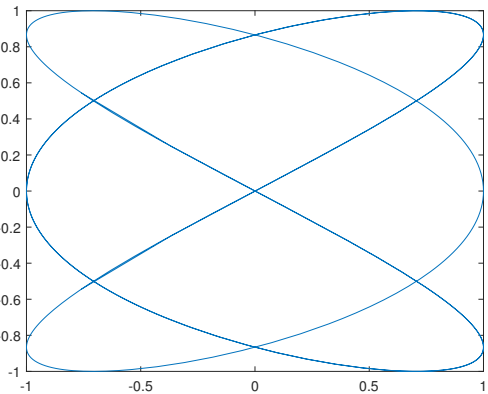


Figure 3 Description of image 2

Code highlighting

This is MATLAB code highlighting environment

MATLAB code

```
1 % Euler method for the ODE model
2 % u'(x)=x^2+x-u, x in [0,1]
3 % Initial condition: u(0)=0.
4 clear all; clf
5 h=0.1;
6 x=0:h:1;
7 N=length(x)-1;
8 u(1)=0;
9 fun=@(t,u) t.^2+t-u; % RHS
10 for n=1:N
11     u(n+1)=u(n)+h.*fun(x(n),u(n));
12 end
13 ue=-exp(-x)+x.^2-x+1; % exact solution
14 plot(x,ue,'b-',x,u,'r+', 'LineWidth',1)
15 legend('Exact','Numerical','location','North')
16 xlabel('x'), ylabel('u')
```

This is Python code highlighting environment

Python code

```
1 #PythonDraw.py
2 import turtle as t
3 t.setup(650, 350, 200, 200)
4 t.penup()
5 t.fd(-250)
6 t.pendown()
7 t.pensize(25)
8 t.pencolor("purple color")
9 t.seth(-40)
10 for i in range(4):
11     t.circle(40, 80)
12     t.circle(-40, 80)
13 t.circle(40, 80/2)
14 t.fd(40)
15 t.circle(16, 180)
16 t.fd(40 * 2/3)
17 t.done()
```

References

- [1] E. Tadmor. A review of numerical methods for nonlinear partial differential equations. *Bull. Amer. Math. Soc.*, 2012, 49(4): 507-554.
- [2] R. A. Adams, Fournier J J F. *Sobolev spaces*. Elsevier, 2003.
- [3] L. N. Trefethen, J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Rev.*, 2014, 56(3): 385-458.
- [4] J. Shen. Efficient spectral-Galerkin method I. Direct solvers of second- and fourth-order equations using Legendre polynomials. *SIAM J. Sci. Comput.*, 1994, 15(6): 1489-1505.