Post_processing

April 11, 2023

1 Question (4) (b):

1.1 Perform numerical experiments with a constant rd of 1/2 and 1/6(= 0.166667), and grid sizes N = {32, 64, 128, 256}. Here, rd is the stability parameter. In each simulation, evolve the solution to an end time $t_{end} = 0.4$.

2 Answer (4) (b):

3 Import the necessary libraries

[1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

4 Reading the name of the files generated

[2]:
```python
Output_files = pd.read_csv("Output_file_names.csv",delimiter=",",header=None).
    ↪to_numpy()
Output_files = np.squeeze(Output_files)
Output_files = Output_files.tolist()
```

4.1 Output files generated

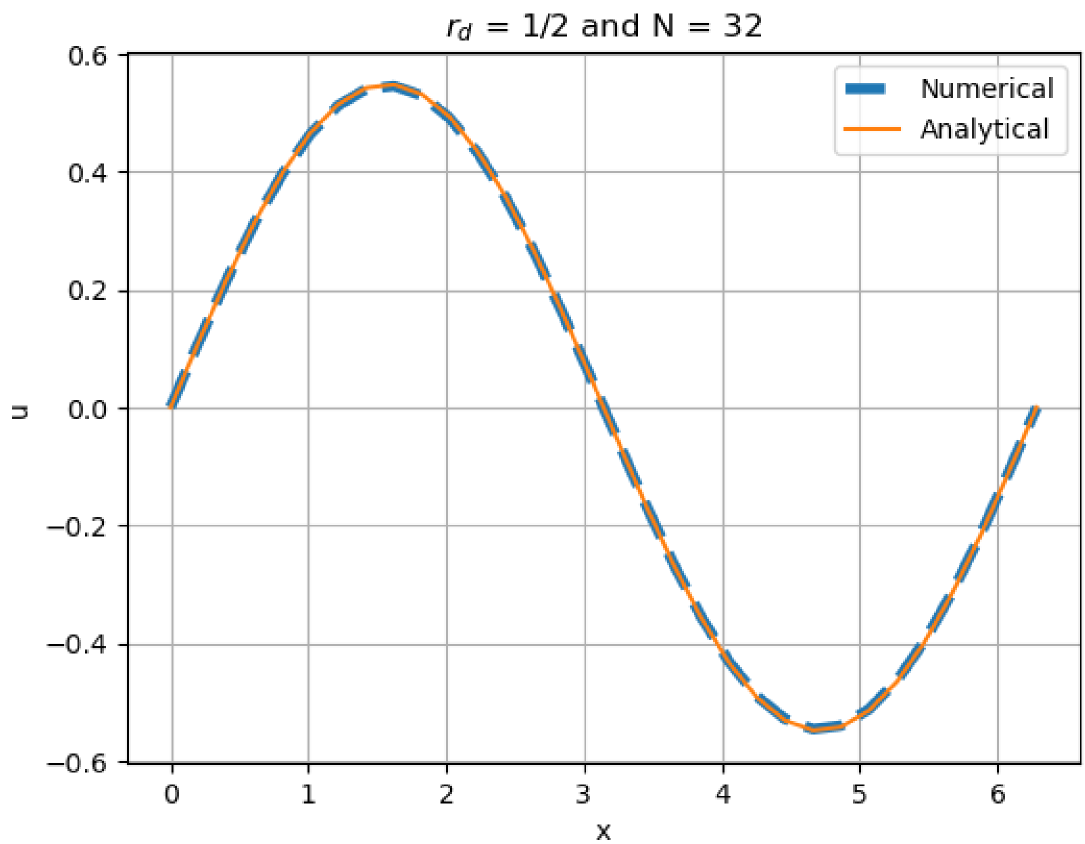
[3]:
```python
Output_files
```

[3]:
```
['Question_4_Explicit_u_n_t_0.400000_N_32_rd_0.500000_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_32_rd_0.500000_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_32_rd_0.500000_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_32_rd_0.166667_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_32_rd_0.166667_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_32_rd_0.166667_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_64_rd_0.500000_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_64_rd_0.500000_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_64_rd_0.500000_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_64_rd_0.166667_.csv',
```

```
 'Analytical_Solution_u_n_t_0.400000_N_64_rd_0.166667_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_64_rd_0.166667_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_128_rd_0.500000_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_128_rd_0.500000_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_128_rd_0.500000_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_128_rd_0.166667_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_128_rd_0.166667_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_128_rd_0.166667_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_256_rd_0.500000_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_256_rd_0.500000_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_256_rd_0.500000_.csv',
 'Question_4_Explicit_u_n_t_0.400000_N_256_rd_0.166667_.csv',
 'Analytical_Solution_u_n_t_0.400000_N_256_rd_0.166667_.csv',
 'Question_4_Explicit_Average_Error_u_n_t_0.400000_N_256_rd_0.166667_.csv']
```

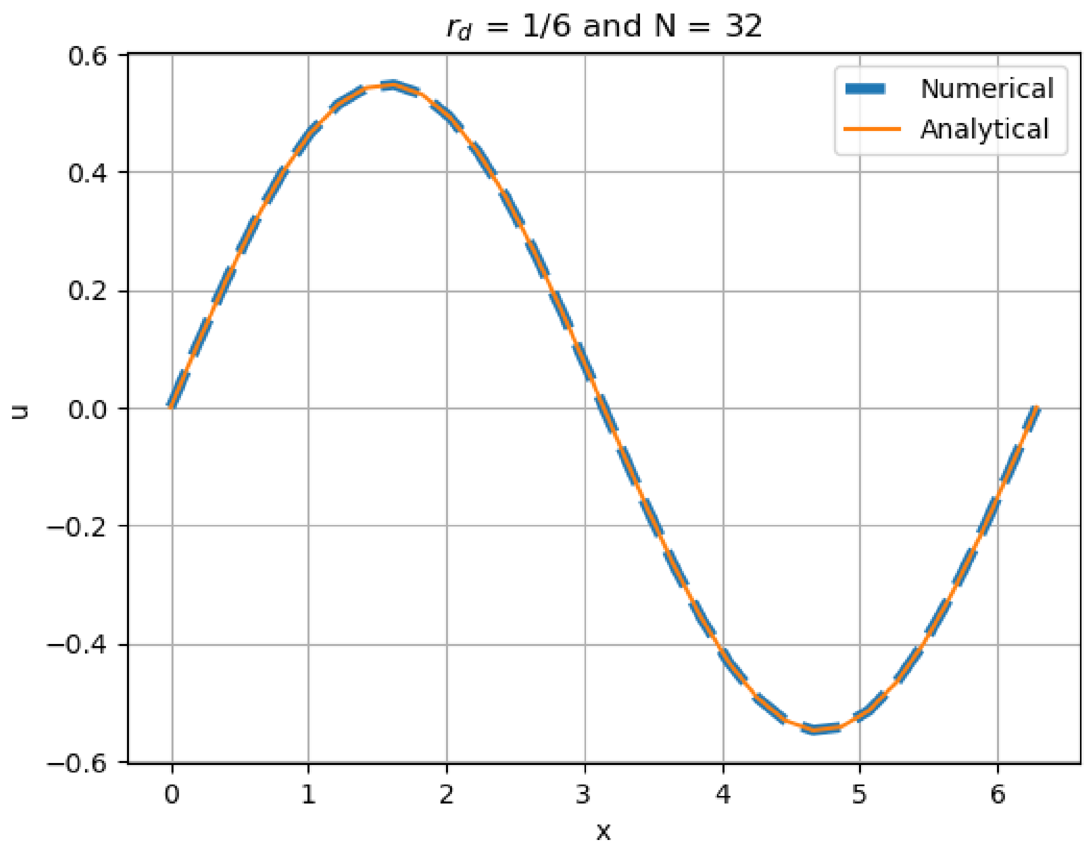
4.2 Plotting the results for rd = 1/2 and N = 32 at $t_{end} = 0.4$

[4]:
```
Numerical_Solution_N_32_rd_half = pd.read_csv(Output_files[0], delimiter =␣
 ↪",",header=None).to_numpy()
Analytical_Solution_N_32_rd_half = pd.read_csv(Output_files[1], delimiter =␣
 ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_32_rd_half.shape[0])
plt.plot(x,Numerical_Solution_N_32_rd_half,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_32_rd_half)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/2 and N = 32")
plt.grid()
plt.savefig("r_d_1_2_and_N_32.png",dpi = 1000)
plt.show()
```



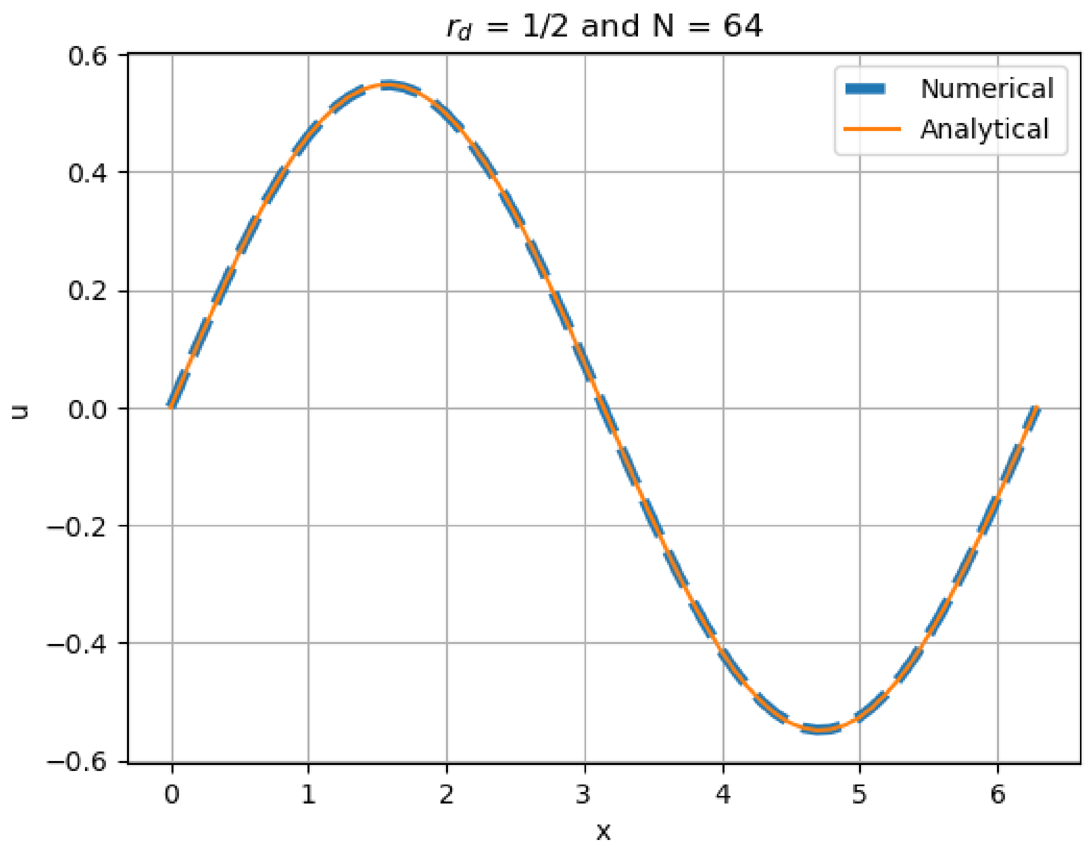
4.3 Plotting the results for rd = 1/6 and N = 32 at $t_{end} = 0.4$

[5]:
```
Numerical_Solution_N_32_rd_1_6 = pd.read_csv(Output_files[3], delimiter =␣
 ↪",",header=None).to_numpy()
Analytical_Solution_N_32_rd_1_6 = pd.read_csv(Output_files[4], delimiter =␣
 ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_32_rd_1_6.shape[0])
plt.plot(x,Numerical_Solution_N_32_rd_1_6,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_32_rd_1_6)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/6 and N = 32")
plt.grid()
plt.savefig("r_d_1_6_and_N_32.png",dpi = 1000)
plt.show()
```



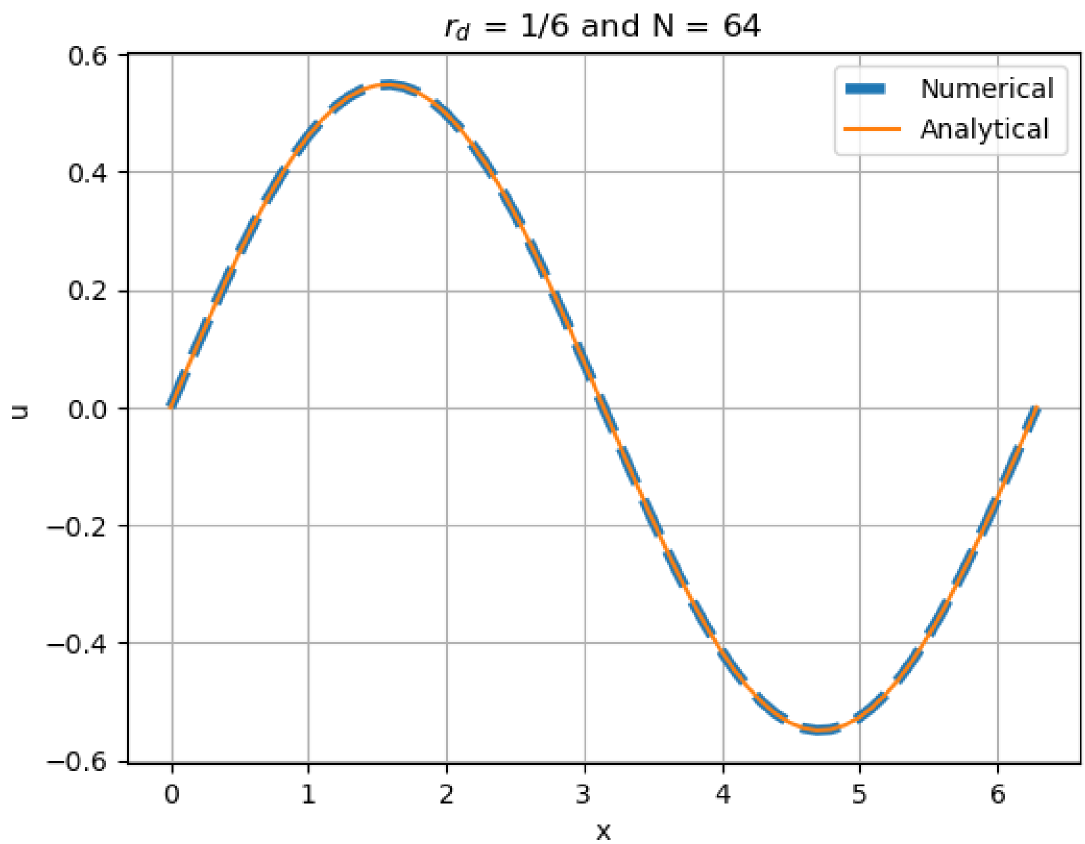
4.4 Plotting the results for rd = 1/2 and N = 64 at $t_{end} = 0.4$

[6]:
```
Numerical_Solution_N_64_rd_half = pd.read_csv(Output_files[6], delimiter =␣
 ↪",",header=None).to_numpy()
Analytical_Solution_N_64_rd_half = pd.read_csv(Output_files[7], delimiter =␣
 ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_64_rd_half.shape[0])
plt.plot(x,Numerical_Solution_N_64_rd_half,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_64_rd_half)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/2 and N = 64")
plt.grid()
plt.savefig("r_d_1_2_and_N_64.png",dpi = 1000)
plt.show()
```



4.5 Plotting the results for rd = 1/6 and N = 64 at $t_{end} = 0.4$

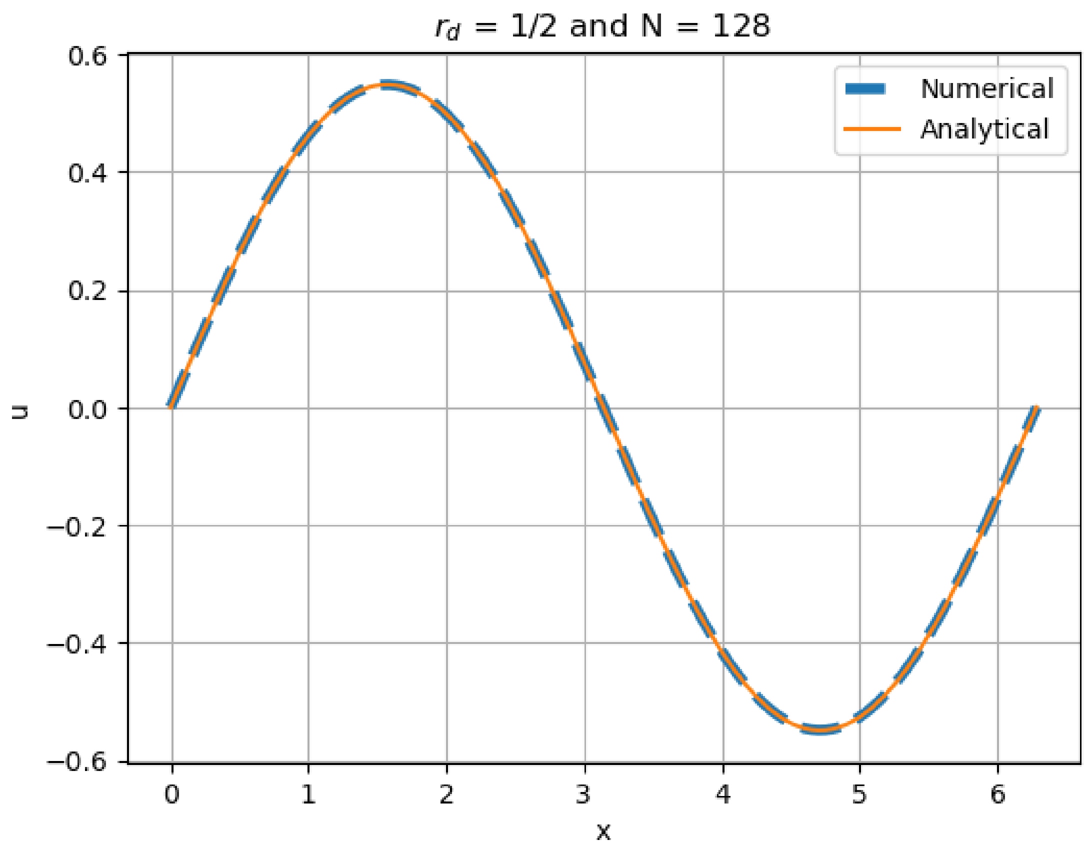
[7]:
```
Numerical_Solution_N_64_rd_1_6 = pd.read_csv(Output_files[9], delimiter =␣
  ↪",",header=None).to_numpy()
Analytical_Solution_N_64_rd_1_6 = pd.read_csv(Output_files[10], delimiter =␣
  ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_64_rd_1_6.shape[0])
plt.plot(x,Numerical_Solution_N_64_rd_1_6,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_64_rd_1_6)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/6 and N = 64")
plt.grid()
plt.savefig("r_d_1_6_and_N_64.png",dpi = 1000)
plt.show()
```



4.6 Plotting the results for rd = 1/2 and N = 128 at $t_{end} = 0.4$

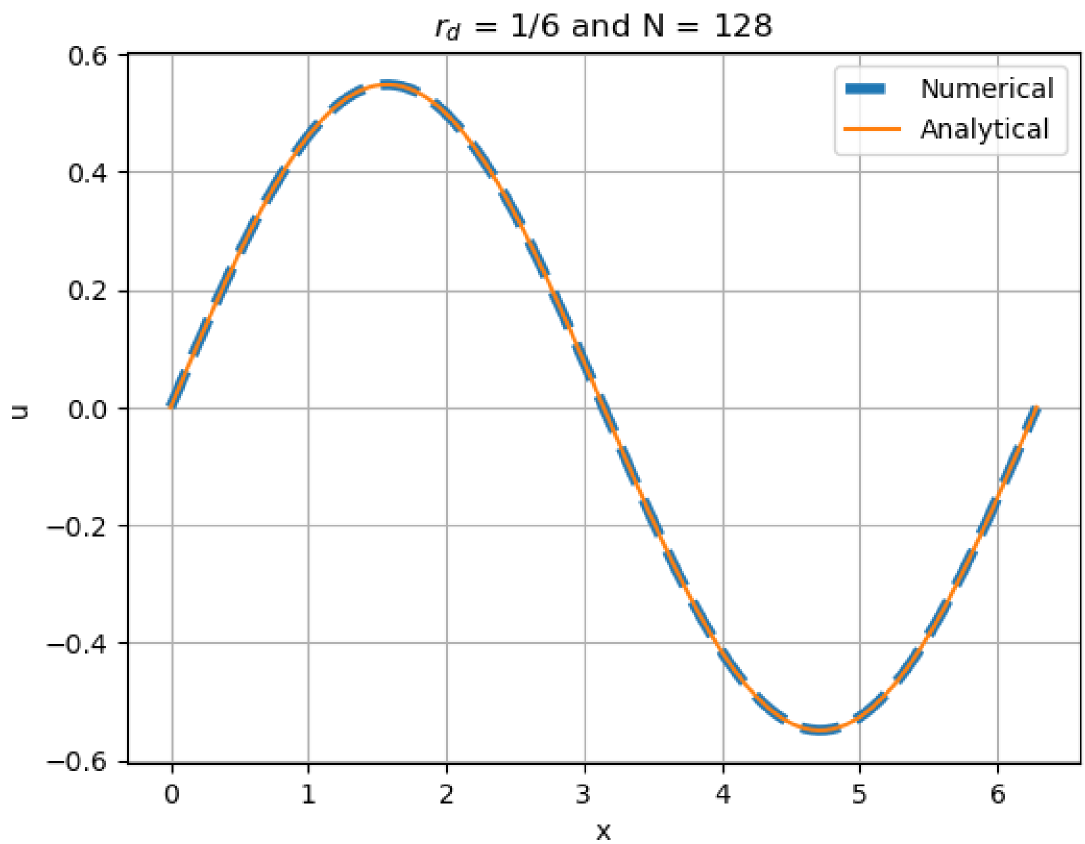
[8]:

```
Numerical_Solution_N_128_rd_half = pd.read_csv(Output_files[12], delimiter =␣
↪",",header=None).to_numpy()
Analytical_Solution_N_128_rd_half = pd.read_csv(Output_files[13], delimiter =␣
↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_128_rd_half.shape[0])
plt.plot(x,Numerical_Solution_N_128_rd_half,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_128_rd_half)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/2 and N = 128")
plt.grid()
plt.savefig("r_d_1_2_and_N_128.png",dpi = 1000)
plt.show()
```



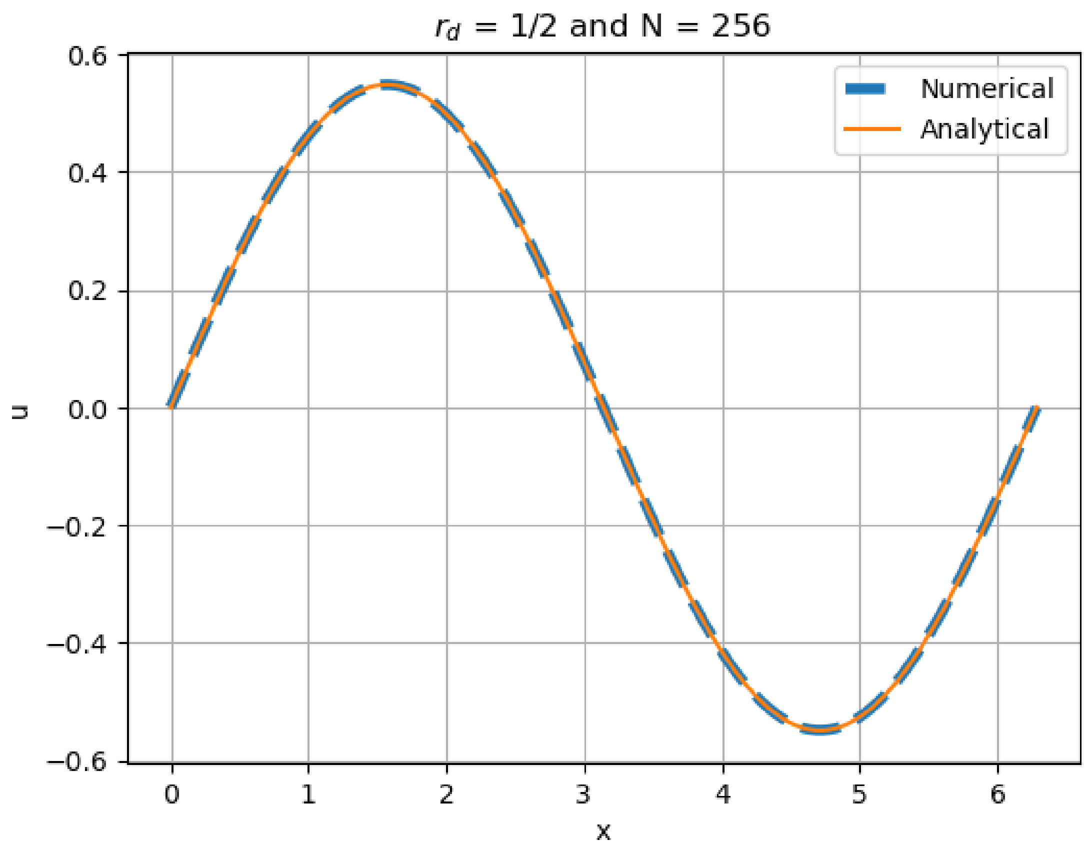
4.7 Plotting the results for rd = 1/6 and N = 128 at $t_{end} = 0.4$

[9]:
```
Numerical_Solution_N_128_rd_1_6 = pd.read_csv(Output_files[15], delimiter =␣
 ↪",",header=None).to_numpy()
Analytical_Solution_N_128_rd_1_6 = pd.read_csv(Output_files[16], delimiter =␣
 ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_128_rd_1_6.shape[0])
plt.plot(x,Numerical_Solution_N_128_rd_1_6,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_128_rd_1_6)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/6 and N = 128")
plt.grid()
plt.savefig("r_d_1_6_and_N_128.png",dpi = 1000)
plt.show()
```



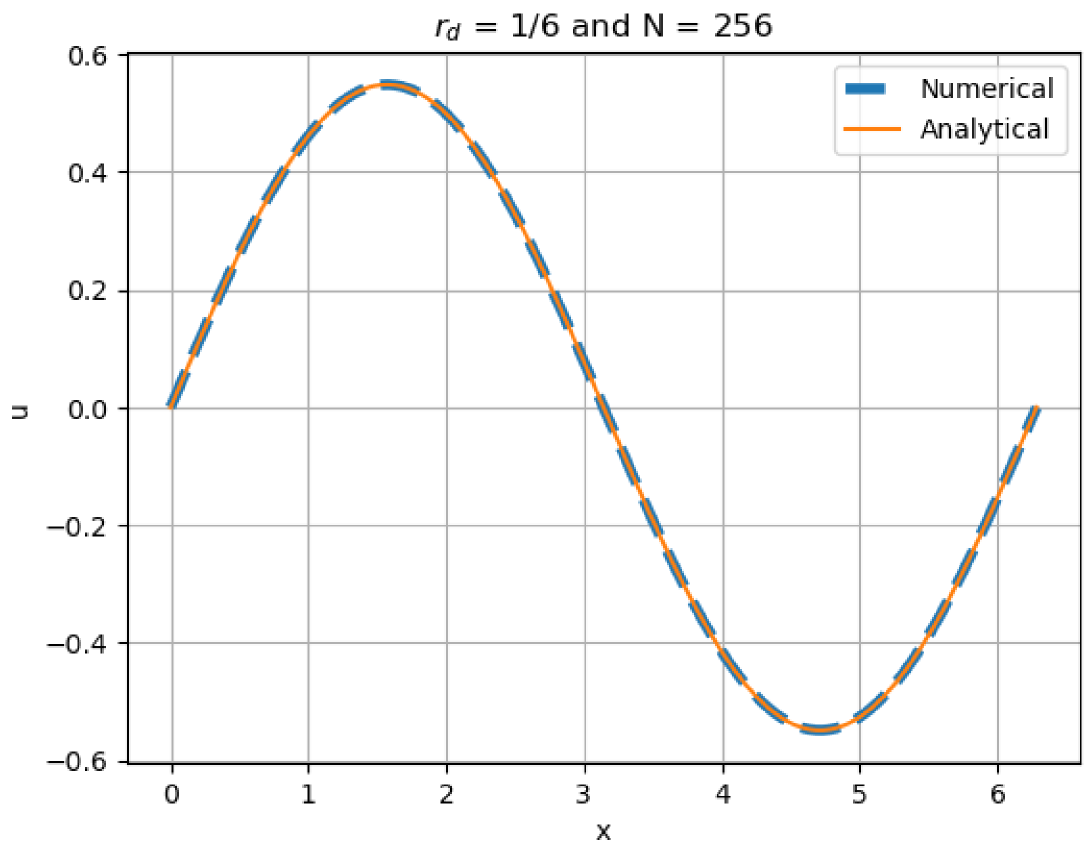
4.8 Plotting the results for rd = 1/2 and N = 256 at $t_{end} = 0.4$

[10]:
```
Numerical_Solution_N_256_rd_half = pd.read_csv(Output_files[18], delimiter =␣
  ↪",",header=None).to_numpy()
Analytical_Solution_N_256_rd_half = pd.read_csv(Output_files[19], delimiter =␣
  ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_256_rd_half.shape[0])
plt.plot(x,Numerical_Solution_N_256_rd_half,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_256_rd_half)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/2 and N = 256")
plt.grid()
plt.savefig("r_d_1_2_and_N_256.png",dpi = 1000)
plt.show()
```



4.9 Plotting the results for rd = 1/6 and N = 256 at $t_{end} = 0.4$

[11]:
```
Numerical_Solution_N_256_rd_1_6 = pd.read_csv(Output_files[21], delimiter =␣
  ↪",",header=None).to_numpy()
Analytical_Solution_N_256_rd_1_6 = pd.read_csv(Output_files[22], delimiter =␣
  ↪",",header=None).to_numpy()
x = np.linspace(0,2*np.pi,Numerical_Solution_N_256_rd_1_6.shape[0])
plt.plot(x,Numerical_Solution_N_256_rd_1_6,linestyle='dashed',linewidth=4)
plt.plot(x,Analytical_Solution_N_256_rd_1_6)
plt.xlabel("x")
plt.ylabel("u")
plt.legend(["Numerical","Analytical"])
plt.title(R"$r_d$ = 1/6 and N = 256")
plt.grid()
plt.savefig("r_d_1_6_and_N_256.png",dpi = 1000)
plt.show()
```



5 Question (4) (c):

5.1 Use the analytical solution to compute the average of absolute error (E(N)) in each simulation at t = tend.

6 Answer (4) (c):

6.1 The average of absolute error (E(N)) in each simulation at t = t_{end} is being calculated and stored in the following files.

[12]:
```
for file in Output_files:
    if "Average_Error" in file:
        print(file)
```

```
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_32_rd_0.500000_.csv
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_32_rd_0.166667_.csv
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_64_rd_0.500000_.csv
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_64_rd_0.166667_.csv
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_128_rd_0.500000_.csv
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_128_rd_0.166667_.csv
```

```
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_256_rd_0.500000_.csv
Question_4_Explicit_Average_Error_u_n_t_0.400000_N_256_rd_0.166667_.csv
```

7 Question (4) (d):

7.1 In a graph with logarithmic scale, plot N vs E(N) for rd = 1/2 and 1/6. Obtain the order of accuracy for the two rd cases.

8 Answer (4) (d):

9 Reading the average of absolute error from the files

[13]:
```
rd = []
for i in range(2,len(Output_files),3):
    temp = pd.read_csv(Output_files[i], delimiter = ",",header=None).to_numpy().
 ↪squeeze()
    rd.append(temp.item())
rd_half = []
for i in range(0,len(rd),2):
    temp = rd[i]
    rd_half.append(temp)
rd_one_sixth = []
for i in range(1,len(rd),2):
    temp = rd[i]
    rd_one_sixth.append(temp)
```

9.1 Average of absolute error for rd = 1/2

[14]:
```
rd_half
```

[14]:
```
[0.00138949, 0.000341756, 8.48383e-05, 2.11276e-05]
```

9.2 Average of absolute error for rd = 1/6

[15]:
```
rd_one_sixth
```

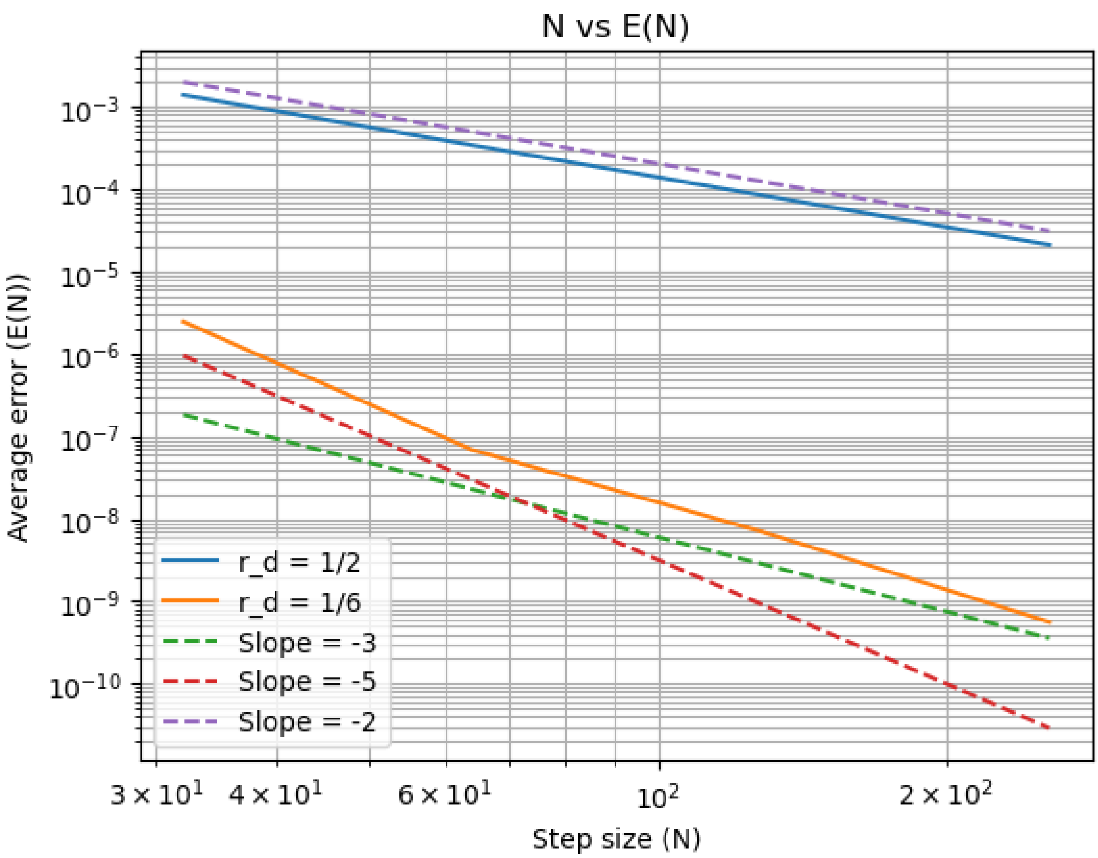
[15]:
```
[2.47444e-06, 6.89476e-08, 7.08266e-09, 5.61005e-10]
```

9.3 Values of N

[16]:
```
N = [32,64,128,256]
```

10 Average error vs step size for different values of rd in log-log scale

[17]:
```
fig = plt.figure()
plt.loglog(N,rd_half)
plt.loglog(N,rd_one_sixth)
plt.plot(N,(5.5*np.array(N))**(-3),"--")
plt.plot(N,(0.5*np.array(N))**(-5),"--")
plt.plot(N,(0.70*np.array(N))**(-2),"--")
plt.legend(["r_d = 1/2", "r_d = 1/6", "Slope = -3","Slope = -5","Slope = -2"])
plt.title("N vs E(N)")
plt.xlabel("Step size (N)")
plt.ylabel("Average error (E(N))")
#plt.xticks(ticks=Step_size_num,rotation=90)
plt.grid(which='both')
plt.show()
fig.savefig("E_N_vs_N.png",dpi = 500, bbox_inches="tight")
```



[ ]:
```
```