Advances in Fuzzy Systems
ISSN , Volume x, Number x, pp. 1–4 (20xx)

Supplementary Files for the article: A Study of Some Properties of fuzzy Laplace Transform with their Applications in Solving The Second-Order Fuzzy Linear Partial Differential Equations

Elhassan Eljaoui[1]* and Said Melliani[2]
[1] University of Sultan Moulay Slimane, Higher School of Education and Training (ESEF), Beni Mellal, P.O. Box 591, Morocco, eljaouihass@gmail.com
[2] University of Sultan Moulay Slimane, Department of Mathematics FST, Beni Mellal, P.O. Box 523, Morocco

Please notice that we used the Python software to plot all the fourteen figures in our work. And bellow, you find the 14 Python programs gathered in this single file.

1 Headinf of all the 14 programs

```
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.datasets import load_iris
from matplotlib import cm
from mpl_toolkits.mplot3d.axes3d import get_test_data
```

2 Python program 1 to plot Figure 1

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def f(x, y):
return np.exp(-y)*np.sin(x)-0.5*np.exp(y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Lower solution 1 for alpha=0.5')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

3 Python program 2 to plot Figure 2

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def g(x, y):
return np.exp(-y)*np.sin(x)+0.5*np.exp(y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
W = g(X, Y)
ax.plot_surface(X, Y, W, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Upper solution 1 for alpha=0.5')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

4 Python program 3 to plot Figure 3

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def f1(x, y):
return np.exp(-y)*np.sin(x)-0.3*np.exp(y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z1 = f1(X, Y)
ax.plot_surface(X, Y, Z1, rstride=1, cstride=1,
```

cmap='jet', edgecolor='none')
ax.set_title('Lower solution 1 for alpha=0.7')
ax.set_xlabel("x")
ax.set_ylabel("t")

5 Python program 4 to plot Figure 4

set up the axes for the first plot
ax = plt.axes(projection='3d')
def f2(x, y):
return np.exp(-y)*np.sin(x)+0.3*np.exp(y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z2 = f2(X, Y)
ax.plot_surface(X, Y, Z2, rstride=1, cstride=1, cmap='jet', edgecolor='none')
ax.set_title('Upper solution 1 for alpha=0.7')
ax.set_xlabel("x")
ax.set_ylabel("t")

6 Python program 5 to plot Figure 5

set up the axes for the first plot
ax = plt.axes(projection='3d')
def f(x, y):
return np.exp(-y)*np.sin(x)-0.5*np.exp(-y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap='jet', edgecolor='none')
ax.set_title('Lower solution 2 for alpha=0.5')
ax.set_xlabel("x")
ax.set_ylabel("t")

7 Python program 6 to plot Figure 6

set up the axes for the first plot
ax = plt.axes(projection='3d')
def g(x, y):
return np.exp(-y)*np.sin(x)+0.5*np.exp(-y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
W = g(X, Y)
ax.plot_surface(X, Y, W, rstride=1, cstride=1, cmap='jet', edgecolor='none')
ax.set_title('Upper solution 2 for alpha=0.5')
ax.set_xlabel("x")
ax.set_ylabel("t")

8 Python program 7 to plot Figure 7

set up the axes for the first plot
ax = plt.axes(projection='3d')
def f1(x, y):
return np.exp(-y)*np.sin(x)-0.3*np.exp(-y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z1 = f1(X, Y)
ax.plot_surface(X, Y, Z1, rstride=1, cstride=1, cmap='jet', edgecolor='none')
ax.set_title('Lower solution 2 for alpha=0.7')
ax.set_xlabel("x")
ax.set_ylabel("t")

9 Python program 8 to plot Figure 8

set up the axes for the first plot
ax = plt.axes(projection='3d')
def f2(x, y):
return np.exp(-y)*np.sin(x)+0.3*np.exp(-y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)

```
X, Y = np.meshgrid(x, y)
Z2 = f2(X, Y)
ax.plot_surface(X, Y, Z2, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Upper solution 2 for alpha=0.7')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

10 Python program 9 to plot Figure 9

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def f(x, y):
return -0.5*x*y+x**2+y**2
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Lower solution 3 for alpha=0.5')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

11 Python program 10 to plot Figure 10

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def g(x, y):
return 0.5*x*y+x**2+y**2
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
W = g(X, Y)
ax.plot_surface(X, Y, W, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Upper solution 3 for alpha=0.5')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

12 Python program 11 to plot Figure 11

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def f2(x, y):
return -0.000001*x*y+x**2+y**2
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z2 = f2(X, Y)
ax.plot_surface(X, Y, Z2, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Lower solution 3 for al-
pha=0.=0.999999')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

13 Python program 12 to plot Figure 12

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def g2(x, y):
return 0.000001*x*y+x**2+y**2
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
W2 = g2(X, Y)
ax.plot_surface(X, Y, W2, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Upper solution 3 for alpha=0.999999')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

14 Python program 13 to plot Figure 13

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def f(x, y):
return np.exp(-y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
Z = f(X, Y)
```

```
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Crisp solution on [π/2, π]')
ax.set_xlabel("x")
ax.set_ylabel("t")
```

15 Python program 14 to plot Figure 14

```
set up the axes for the first plot
ax = plt.axes(projection='3d')
def g(x, y):
return np.exp(-y)*np.sin(x)
x = np.linspace(np.pi/2, np.pi, 50)
y = np.linspace(np.pi/2, np.pi, 50)
X, Y = np.meshgrid(x, y)
W = g(X, Y)
ax.plot_surface(X, Y, W, rstride=1, cstride=1,
cmap='jet', edgecolor='none')
ax.set_title('Crisp solution on [−π, π]')
ax.set_xlabel("x")
ax.set_ylabel("t")
```