1. Θεωρούμε την εξίσωση της μεταφοράς, $u_t + u_x = 0$, για $x \in [-1, 3]$ και $0 < t \leq 2.4$, με αρχική συνθήκη

$$u^0(x) = \begin{cases} \cos^2(\pi x) & |x| \leq \frac{1}{2}, \\ 0 & \text{διαφορετικά}, \end{cases}$$

και συνοριακή συνθήκη $u(-1, t) = 0$.

(α΄) Βρείτε και σχεδιάστε, πρόχειρα, την ακριβή λύση στις χρονικές στιγμές $t = 1.6$ και $t = 2.4$.

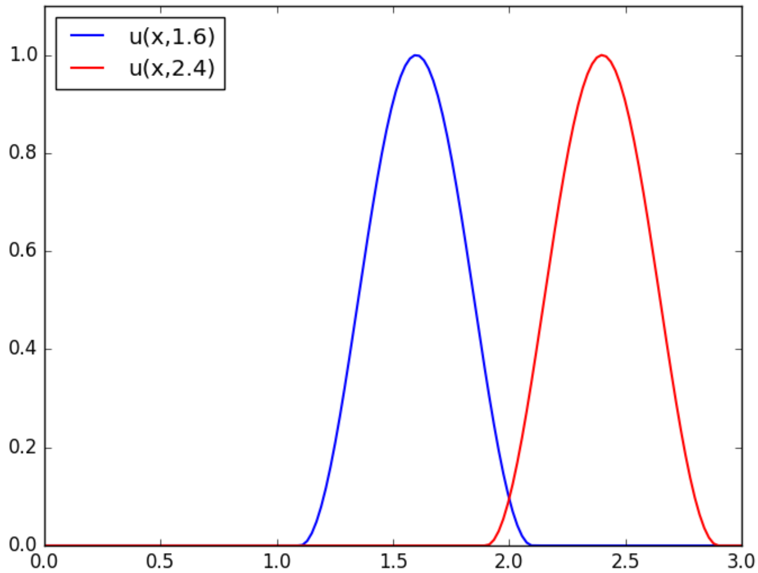
Απάντηση. Γνωρίζουμε ότι η λύση της εξίσωσης της μεταφοράς με τη δεδομένη αρχική και συνοριακή συνθήκη είναι

$$u(x,t) = \begin{cases} u^0(x-t) & x - t \geq -1, \\ 0 & \text{διαφορετικά}, \end{cases}$$

Ειδικότερα,

$$u(x, 1.6) = \begin{cases} \cos^2(\pi(x-1.6)) & 1.1 \leq x \leq 2.1, \\ 0 & \text{διαφορετικά}, \end{cases} \qquad u(x, 2.4) = \begin{cases} \cos^2(\pi(x-2.4)) & 1.9 \leq x \leq 2.9, \\ 0 & \text{διαφορετικά}, \end{cases}$$

οι γραφικές παραστάσεις των οποίων φαίνονται παρακάτω:



(β΄) Ποιά είναι η μέθοδος upwind για τη λύση του συγκεκριμένου προβλήματος;

Απάντηση. Με το συνηθισμένο συμβολισμό, η μέθοδος upwind για το συγκεκριμένο πρόβλημα έχει τη μορφή

$$\frac{U_j^{n+1} - U_j^n}{k} + \frac{U_j^n - U_{j-1}^n}{h} = 0, \qquad j = 1, \dots, J+1, \ n = 0, 1, \dots,$$

όπου $U_0^n = 0$ για $n \geq 0$ και $U_j^0 = u^0(x_j)$, $j = 0, \dots, J+1$. Αν θέσουμε $\lambda = \frac{k}{h}$, η μέθοδος γράφεται στη μορφή

$$U_j^{n+1} = U_j^n - \lambda(U_j^n - U_{j-1}^n), \qquad j = 1, \dots, J+1, \ n = 0, 1, \dots.$$

(γ΄) Ποιά προσέγγιση δίνει η μέθοδος upwind για την τιμή $u(1.44, 1.6)$; Για την τιμή $u(2.24, 2.4)$; Χρησιμοποιήστε $h = 1/100$ και $\lambda = 0.8$. Στρογγυλοποιήστε στο τέταρτο δεκαδικό ψηφίο. Μπορείτε να χρησιμοποιήσετε τη συνάρτηση

```
def u0(x): return np.where(abs(x) > 0.5, 0.0, np.cos(np.pi*x)**2)
```

για να υλοποιήσετε την αρχική συνθήκη.

Απάντηση. Η υλοποίηση της μεθόδου upwind για το συγκεκριμένο πρόβλημα φαίνεται παρακάτω:

```
from math import ceil
import numpy as np

def u0(x): return np.where(np.abs(x) > 0.5, 0, np.cos(np.pi*x)**2)

a = -1;  b = 3;  J = 399;  h = (b - a) / (J+1)
x = np.linspace(-1, 3, J+2)

lam = 0.8;  k = lam * h;  t = 1.6;  N = ceil(t/k)

U = u0(x);  V = np.zeros(J+2)

for n in range(N):
        for j in range(1,J+2): V[j] = U[j] - lam * ( U[j] - U[j-1] )
        for j in range(J+2): U[j] = V[j]
```

Στο τέλος του προγράμματος, το διάνυσμα `U` περιέχει την προσεγγιστική λύση στο χρόνο `t`. Για να προσεγγίσουμε την τιμή $u(1.44, 1.6)$ εκτελούμε τον παραπάνω κώδικα με τις τιμές των παραμέτρων `J = 399`, `t = 1.6` και εκτυπώνουμε την τιμή του διανύσματος `U` στη θέση `j` για την οποία ισχύει `-1 + j*h = 1.44`, δηλαδή για `j = 244`. Έτσι, λαμβάνουμε την προσέγγιση $u(1.44, 1.6) \approx 0.7512$. Για την προσέγγιση της τιμής $u(2.24, 2.4)$ τρέχουμε τον παραπάνω κώδικα με `J = 399`, `t = 2.4` και εκτυπώνουμε την τιμή του διανύσματος `U` στη θέση `j` για την οποία ισχύει `-1 + j*h = 2.24`, δηλαδή για `j = 324`. Έτσι, λαμβάνουμε την προσέγγιση $u(2.24, 2.4) \approx 0.7432$.