# Υπολογιστικός Ηλεκτρομαγνητισμός
# Άσκηση 1

Σεϊτανίδου Δήμητρα

6 Απριλίου 2020

## Αριθμητική λύση κυματικής εξίσωσης στην μία διάσταση

Η κυματική εξίσωση είναι μια υπερβολική διαφορική εξίσωση μερικών παραγώγων και μπορεί να βρεθεί η λύση της με αριθμητικές μεθόδους. Εμείς θα την λύσουμε με δύο μεθόδους, την explicit και την implicit. Η explicit μέθοδος αντικαθιστά τις μερικές παραγώγους με κεντρικές διαφορές και έτσι διακριτοποιείται η εξίσωση με τα παρακάτω βήματα:

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \Rightarrow$$
$$\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = \frac{1}{c^2}\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} \Rightarrow$$
$$u_i^{n+1} = (c\Delta t)^2\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \right) + 2u_i^n - u_i^{n-1} \tag{1}$$

Η μέθοδος αυτή έχει ακρίβεια δεύτερης τάξης γιατί οι κεντρικές διαφορές έχουν ακρίβεια δεύτερης τάξης. Η implicit μέθοδος κάνει μια αντίστοιχη διακριτοποίηση μόνο που αυτή τη φορά θα εκφράσουμε την λύση της εξίσωσης σε μορφή γραμμικού συστήματος:

$$u^{n+1} = 2[A]^{-1}[B]u^n - u^{n-1} \tag{2}$$

όπου $u^n$ ένα διάνυσμα που περιέχει όλα τα $i$ και

$$[A] = \beta[L] + \frac{\Delta x^2}{c\Delta t^2}[I]$$
$$[B] = \frac{2\beta - 1}{2}[L] + \frac{\Delta x^2}{c\Delta t^2}[I]$$

με $[I]$ ο μοναδιαίος πίνακας, $[L]$ ένας τριδιαγώνιος πίνακας με $L_{i,i} = 2$ και τα εκατέρωθεν στοιχεία ίσα με $-1$.Το $\beta = 1/4$ για έχουμε ακρίβεια δεύτερης τάξης.

Εφαρμόζουμε λοιπόν τις σχέσεις (1) και (2) για να λύσουμε την κυματική εξίσωση σε ένα χώρο με $N_x = 10$ χωρικά βήματα, για $N_t = 60$ χρονικά βήματα, χρησιμοποιώντας τις παρακάτω αρχικές και συνοριακές συνθήκες:

- $u_i^{-1} = \begin{cases} 1, i \geq 2 \\ 0, i < 2 \end{cases}$
- $u_i^{0} = \begin{cases} 1, i \geq 3 \\ 0, i < 3 \end{cases}$
- $u_0^n = u_{N_x}^n = 0$

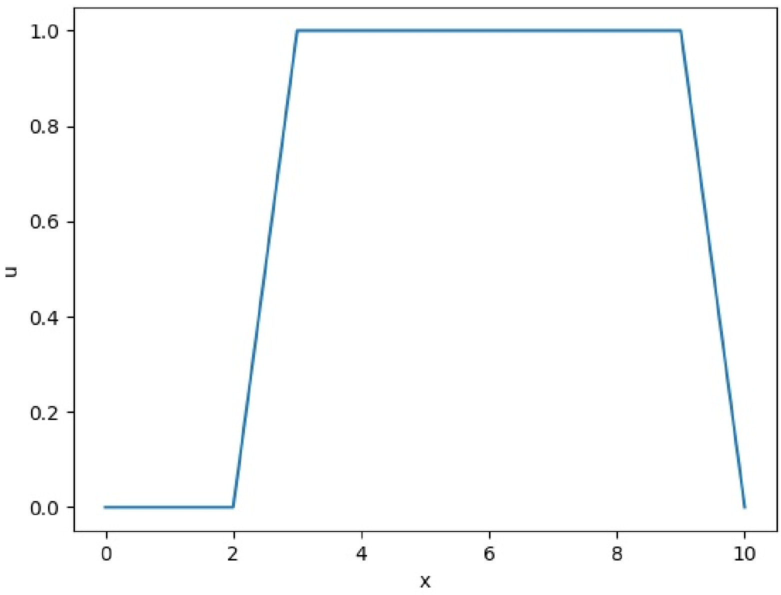
Κρατάμε "φωτογραφίες" του κύματος κάθε 10 βήματα ξεκινώντας από το τη χρονική στιγμή $N_t = 20$ για τρεις περιπτώσεις:

- $c\Delta t = 0.9\Delta x$
- $c\Delta t = \Delta x$
- $c\Delta t = 1.1\Delta x$

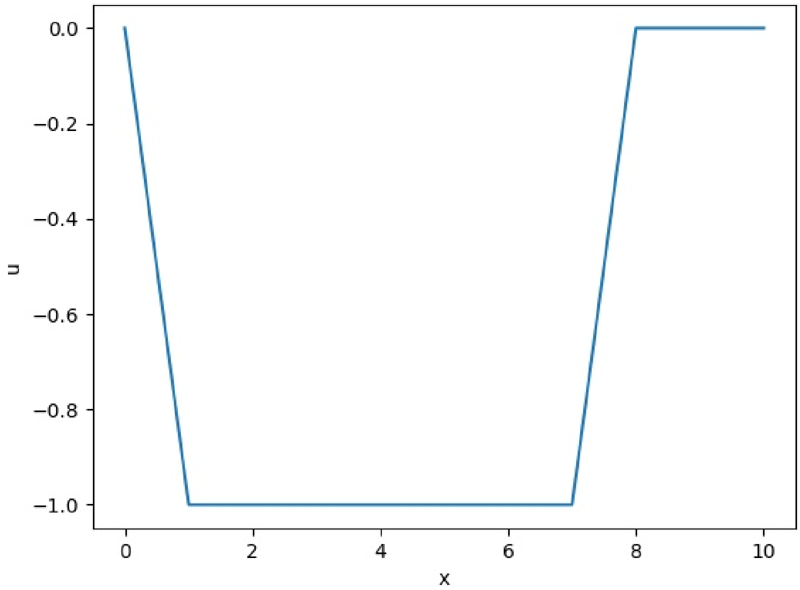
Υποθέτουμε ότι $c = 1$. Τα διαγράμματα δίνονται στην επόμενη παράγραφο.

## Διαγράμματα

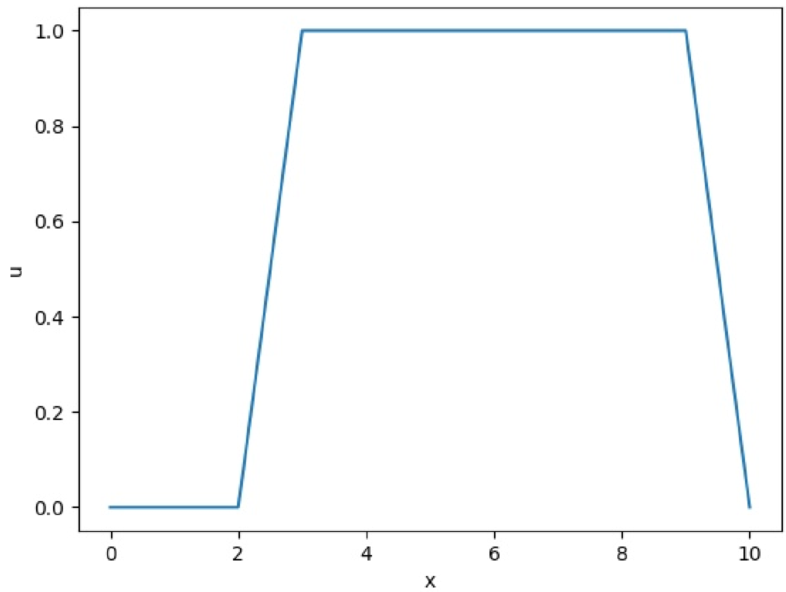
Σχήμα 1: Explicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 20$



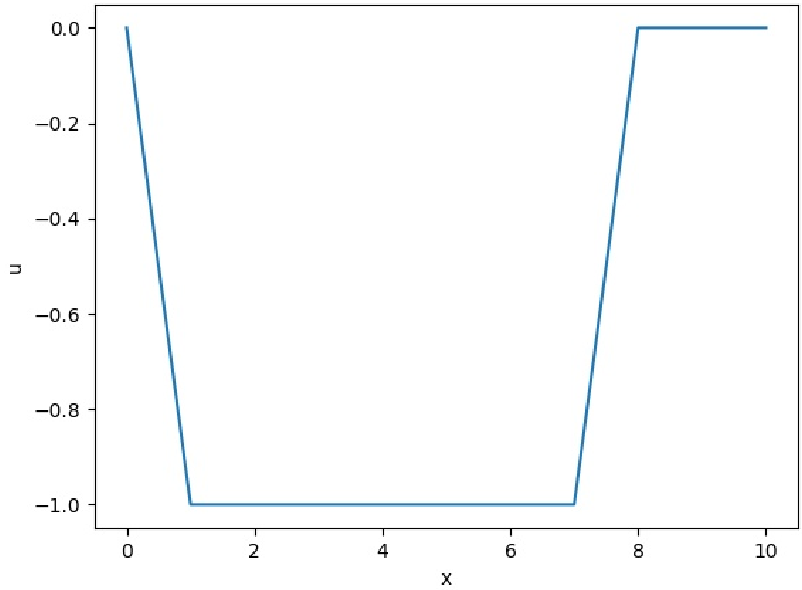
Σχήμα 2: Explicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 30$



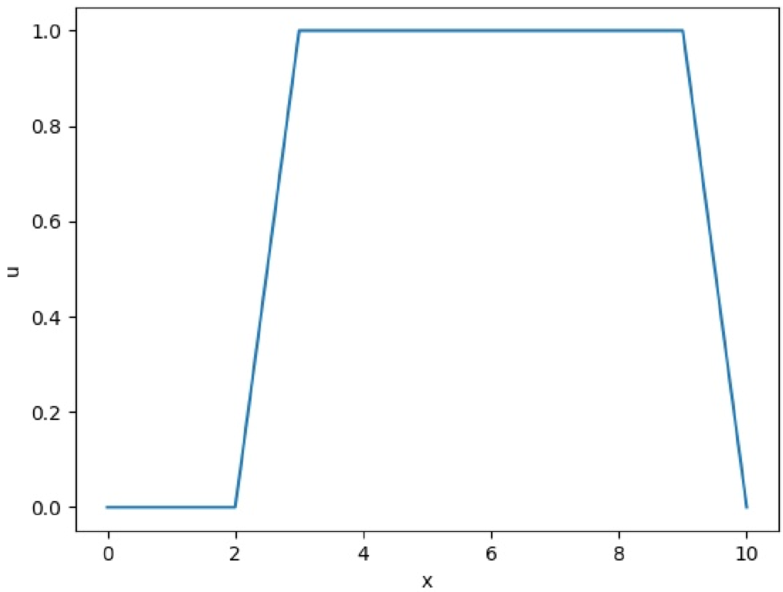

Σχήμα 3: Explicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 40$



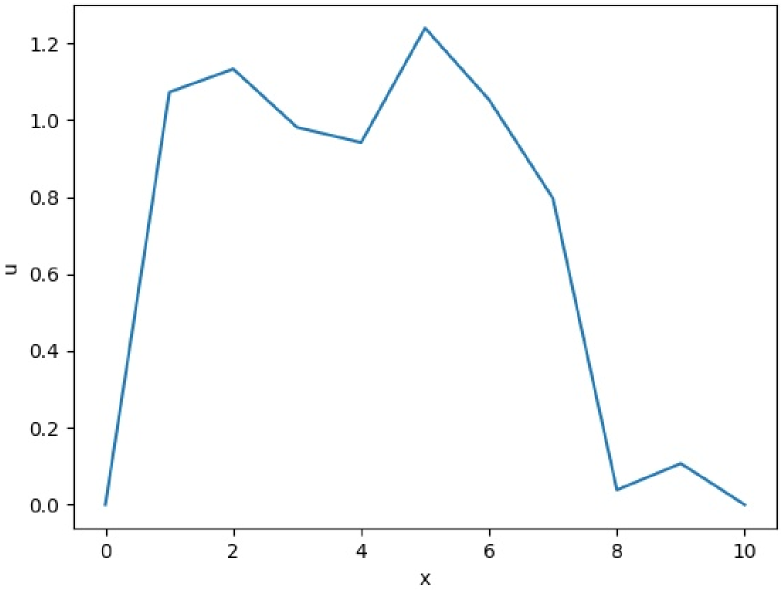
Σχήμα 4: Explicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 50$



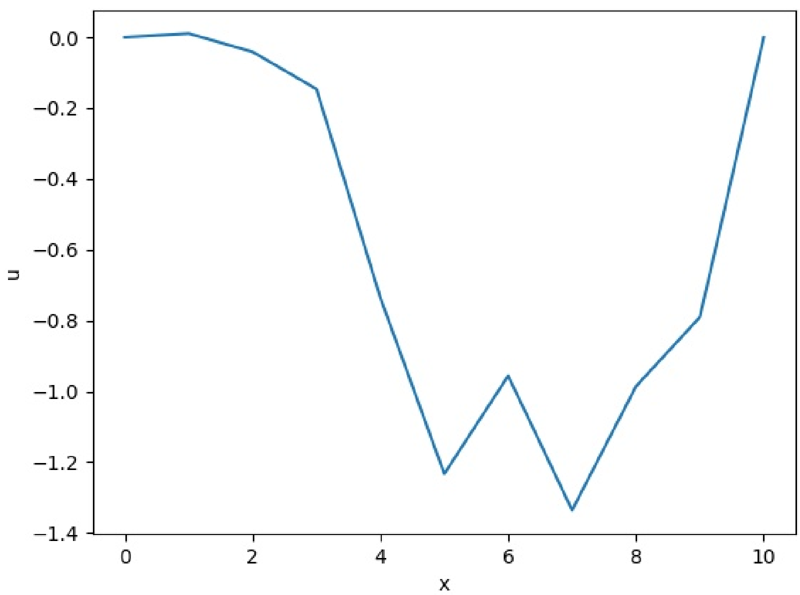

Σχήμα 5: Explicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 60$



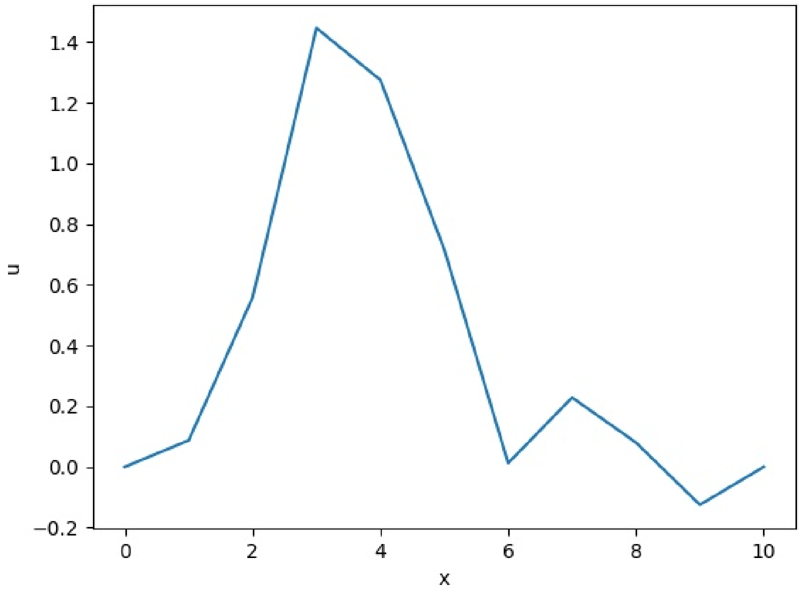
Σχήμα 6: Explicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 20$



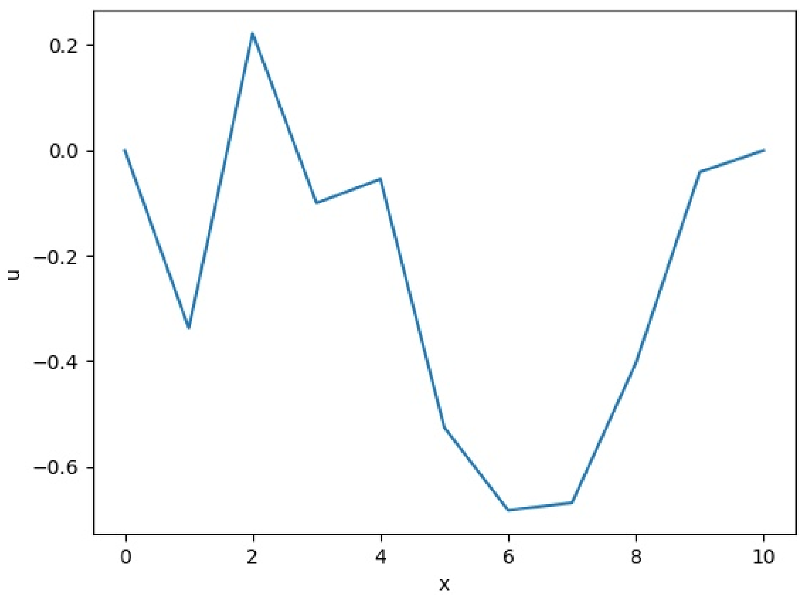

Σχήμα 7: Explicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 30$



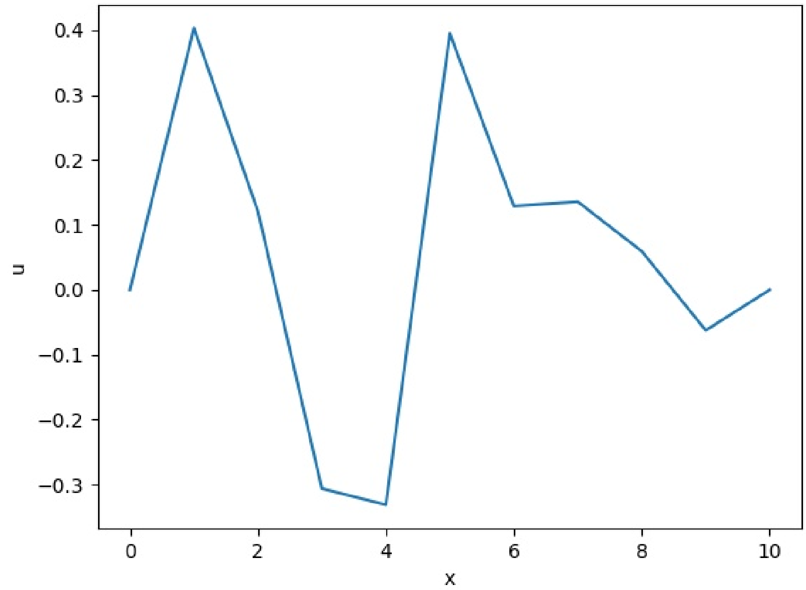
Σχήμα 8: Explicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 40$



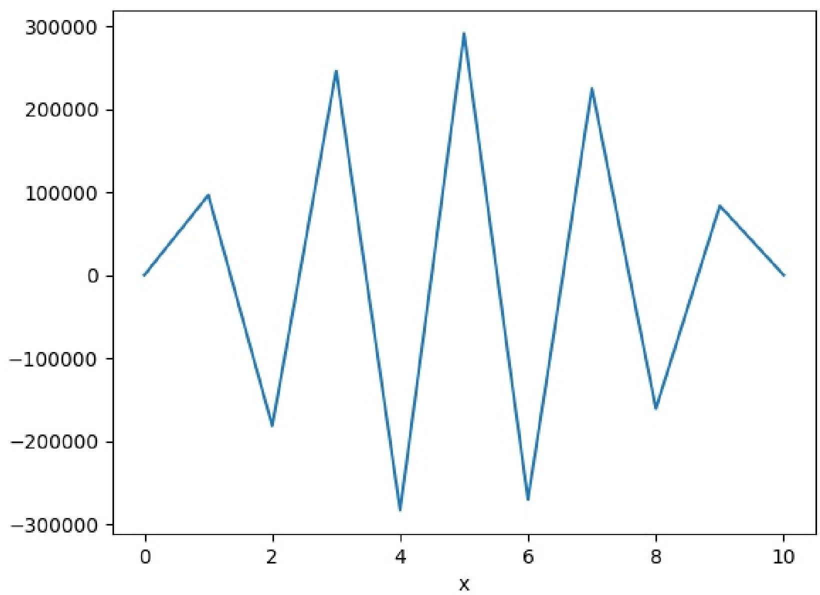

Σχήμα 9: Explicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 50$



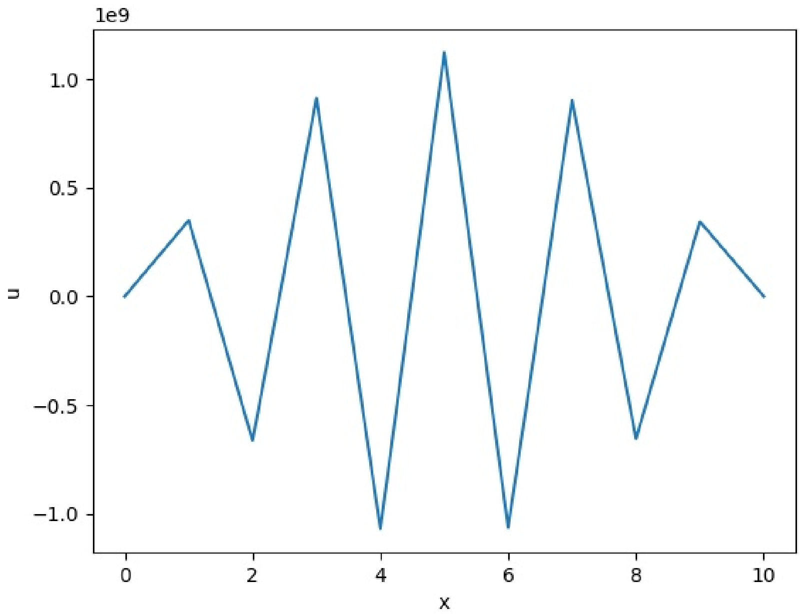
Σχήμα 10: Explicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 60$



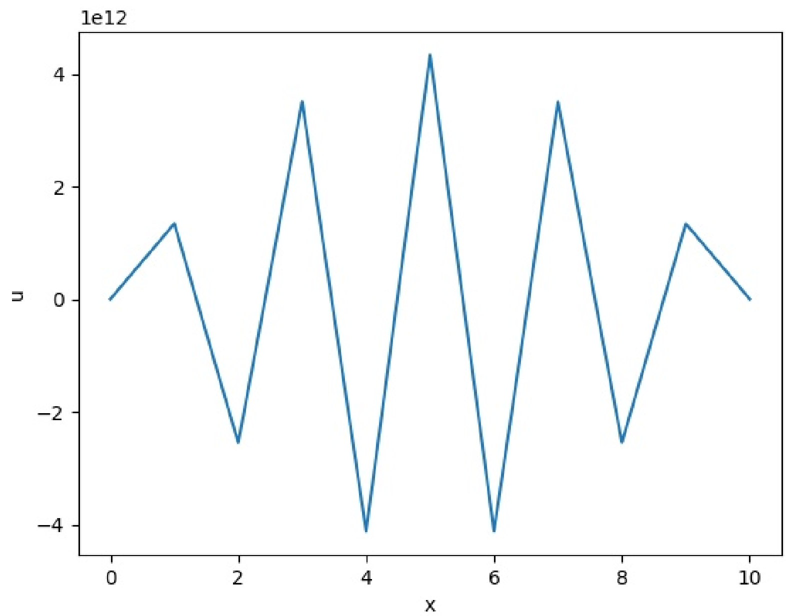

Σχήμα 11: Explicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 20$



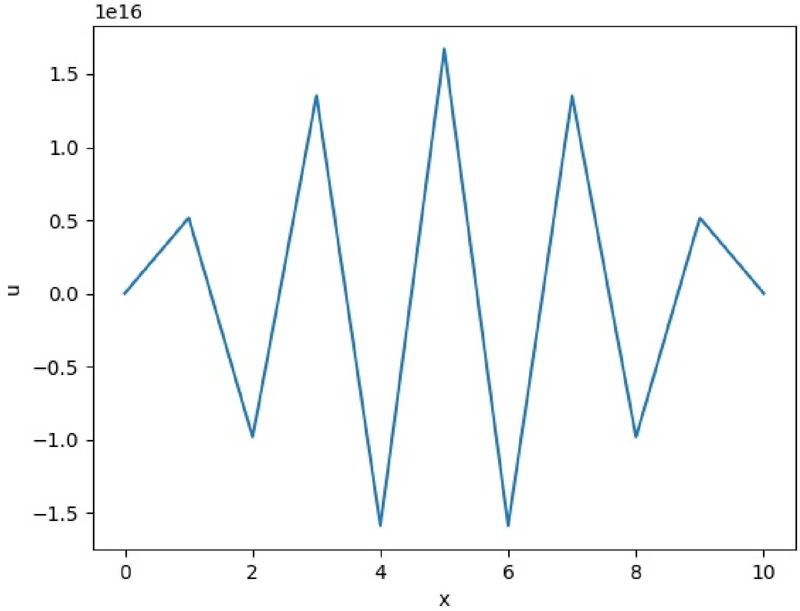
Σχήμα 12: Explicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 30$



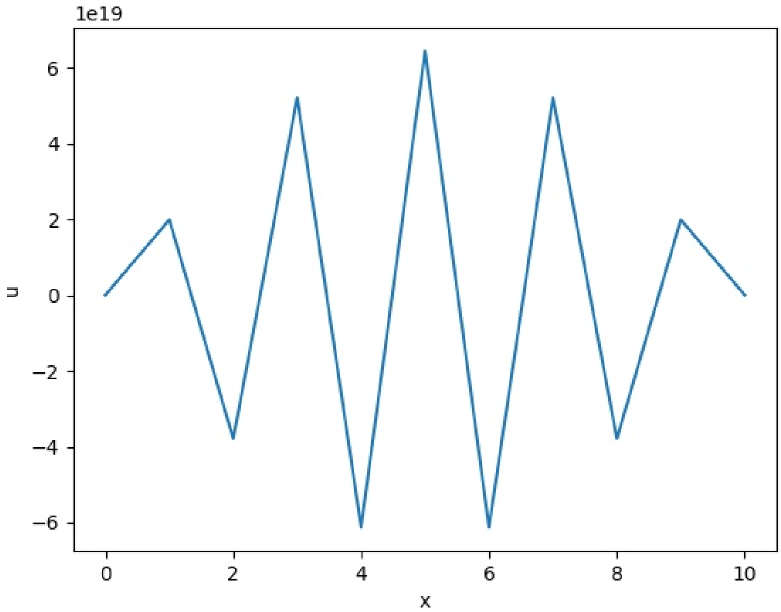

Σχήμα 13: Explicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 40$



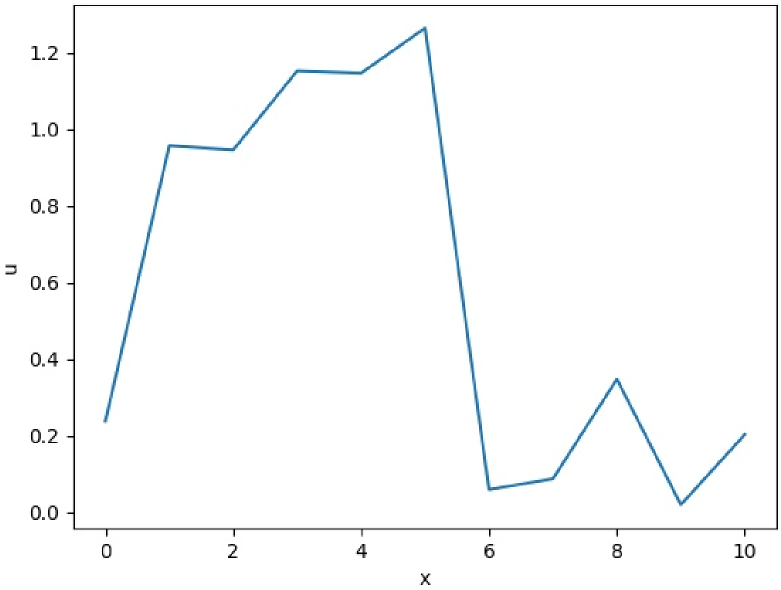
Σχήμα 14: Explicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 50$



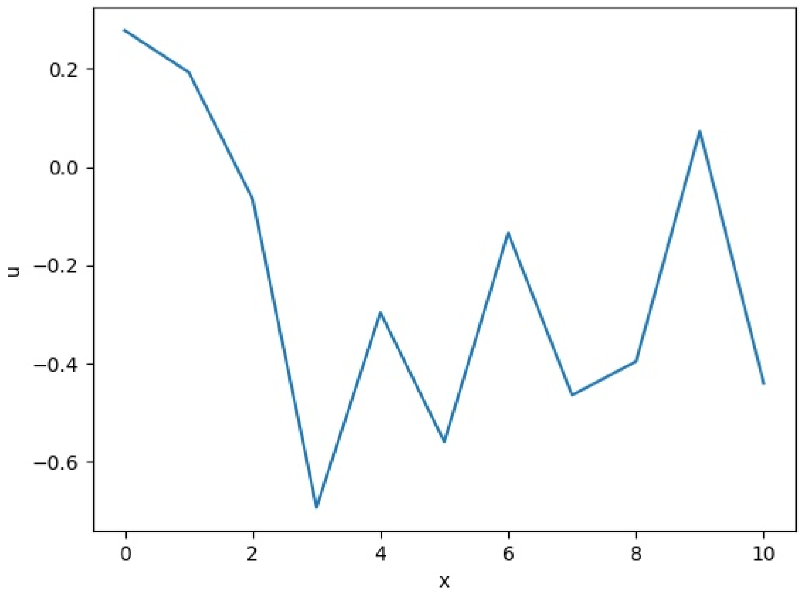

Σχήμα 15: Explicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 60$



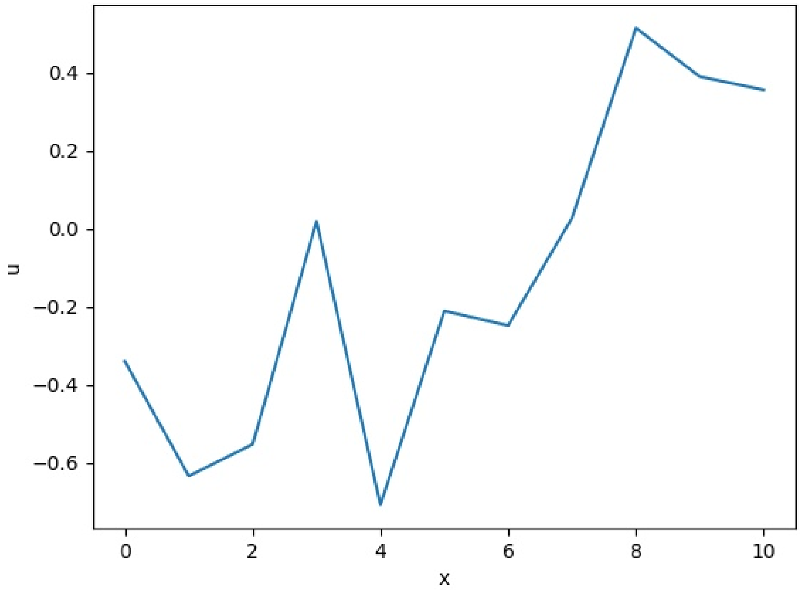
Σχήμα 16: Implicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 20$



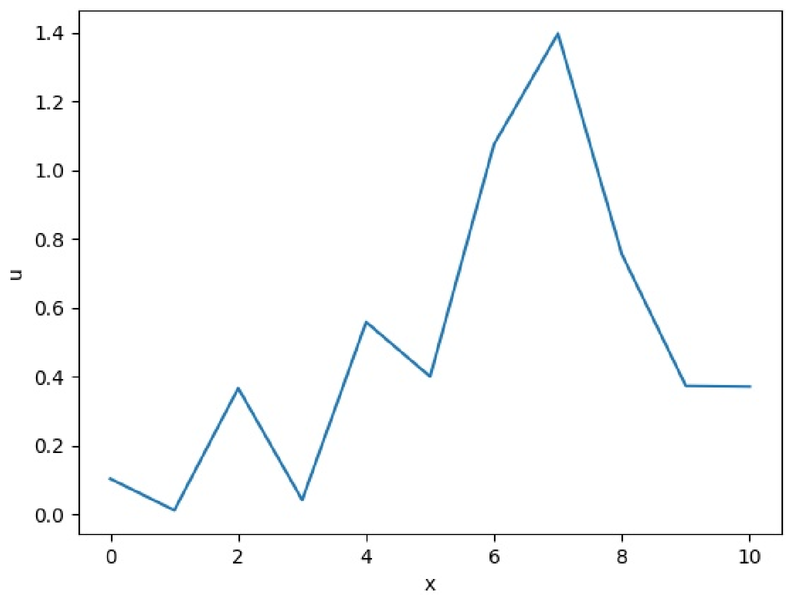

Σχήμα 17: Implicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 30$



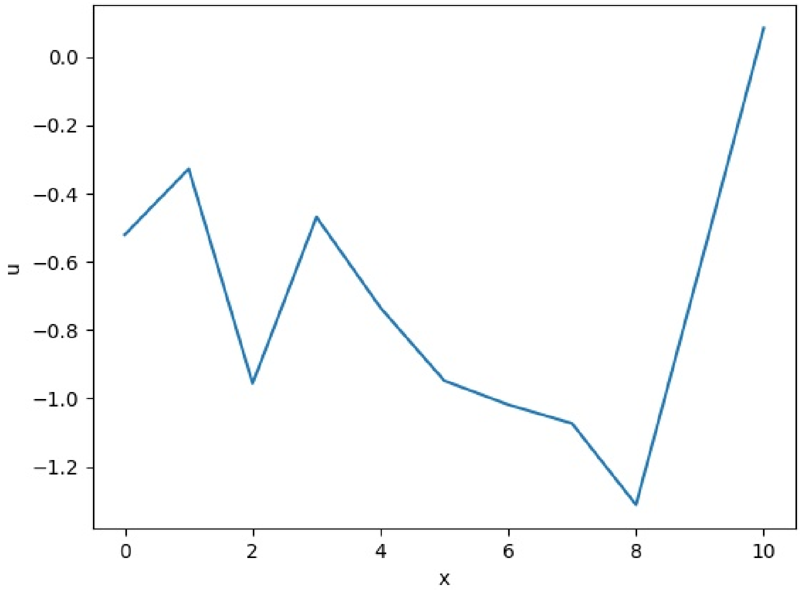
Σχήμα 18: Implicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 40$



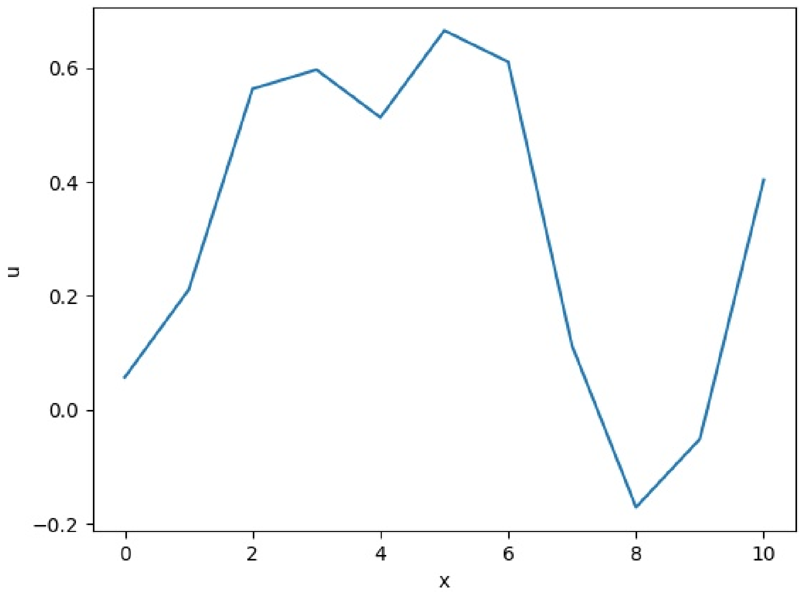

Σχήμα 19: Implicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 50$



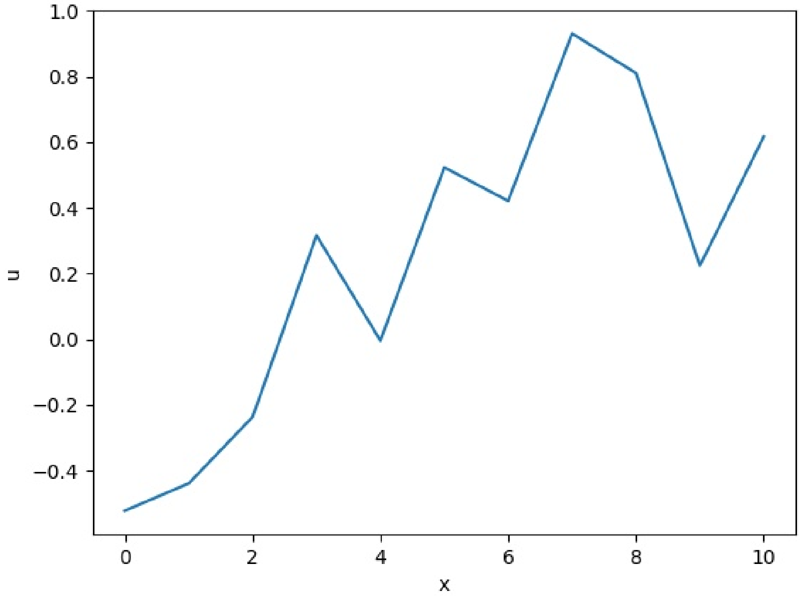
Σχήμα 20: Implicit μέθοδος με $c\Delta t = \Delta x$ για $N_t = 60$



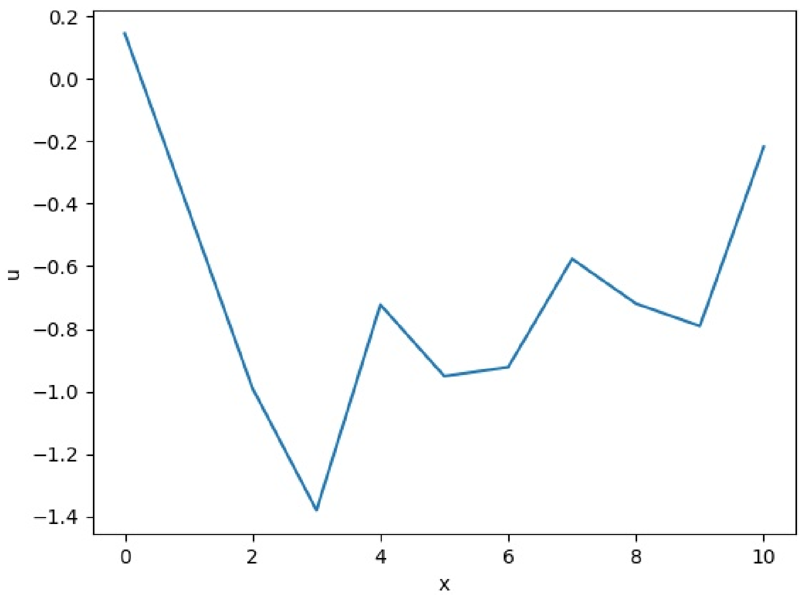

Σχήμα 21: Implicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 20$



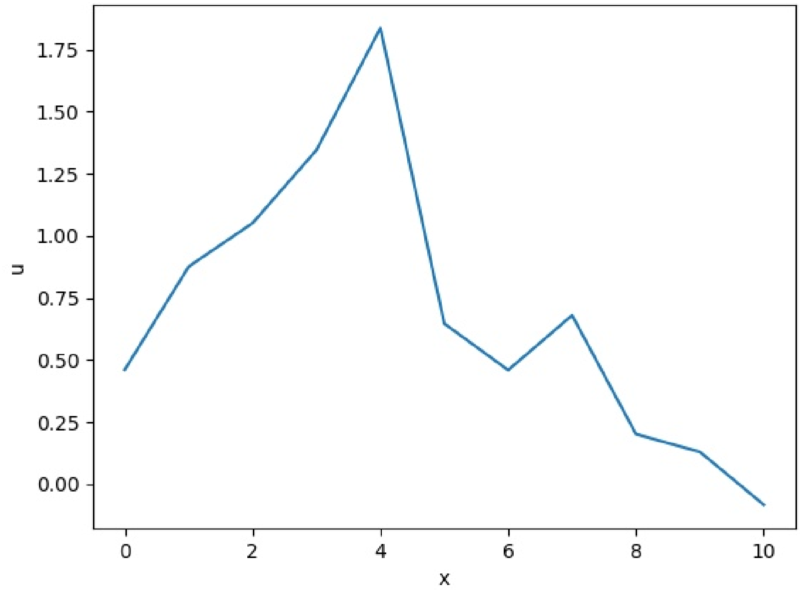
Σχήμα 22: Implicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 30$



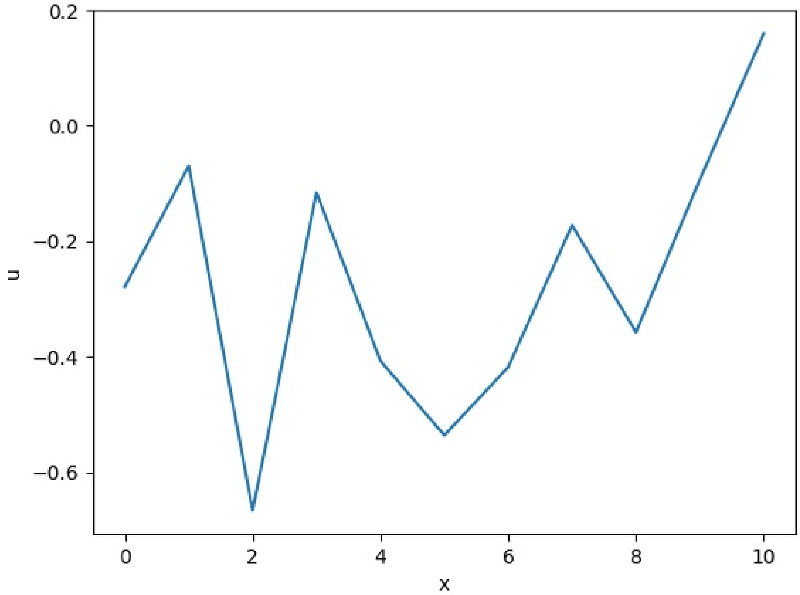

Σχήμα 23: Implicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 40$



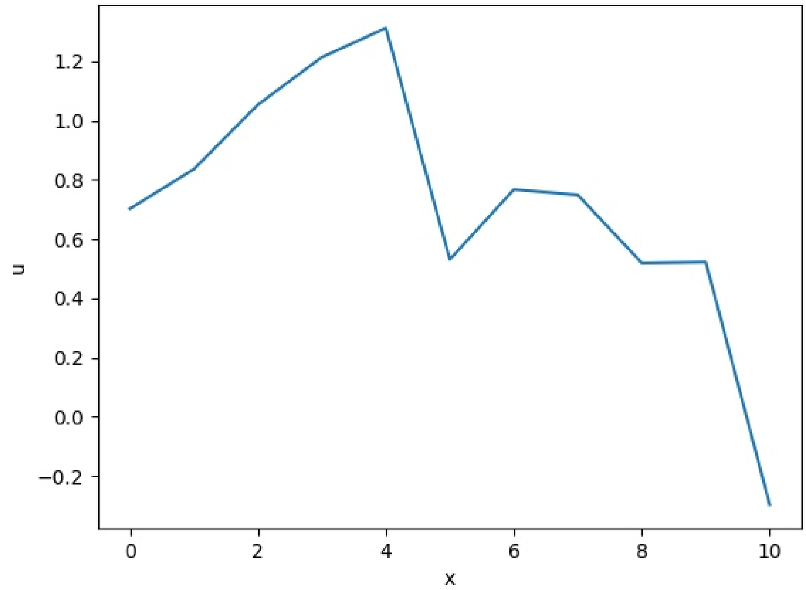
Σχήμα 24: Implicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 50$



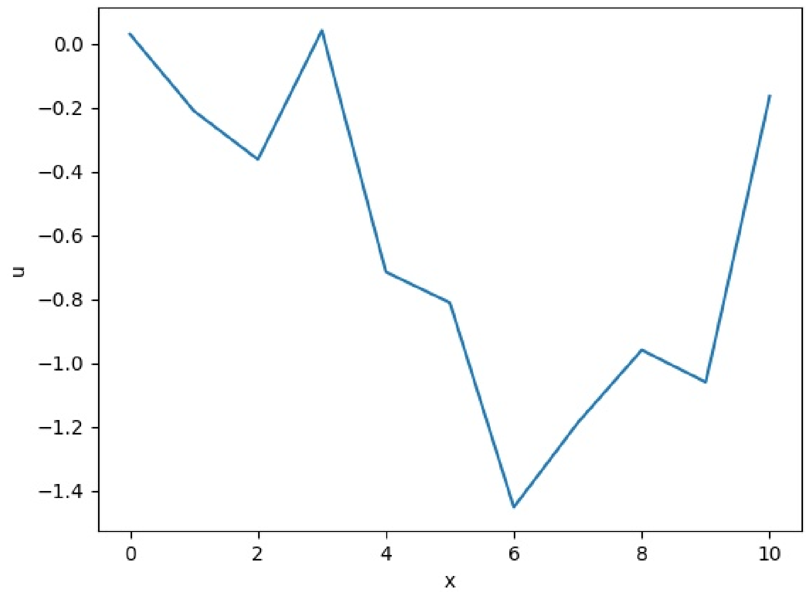

Σχήμα 25: Implicit μέθοδος με $c\Delta t = 0.9\Delta x$ για $N_t = 60$



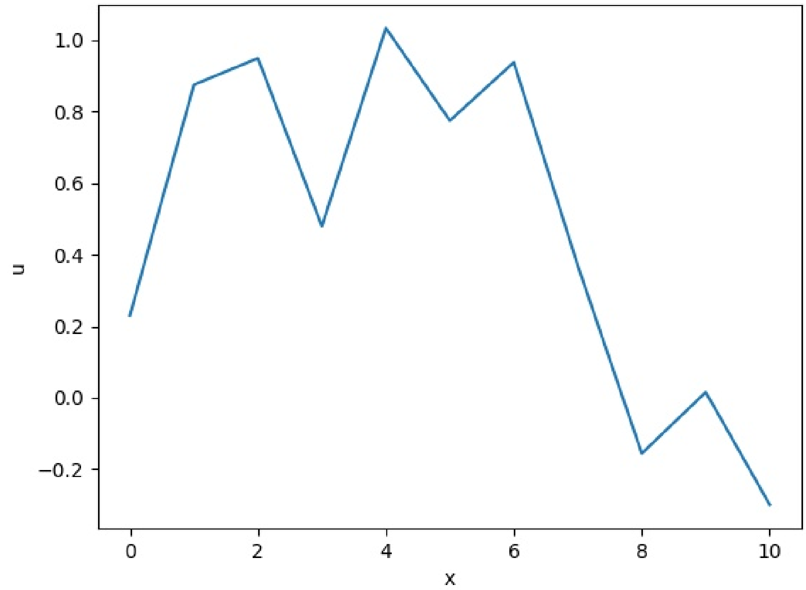
Σχήμα 26: Implicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 20$



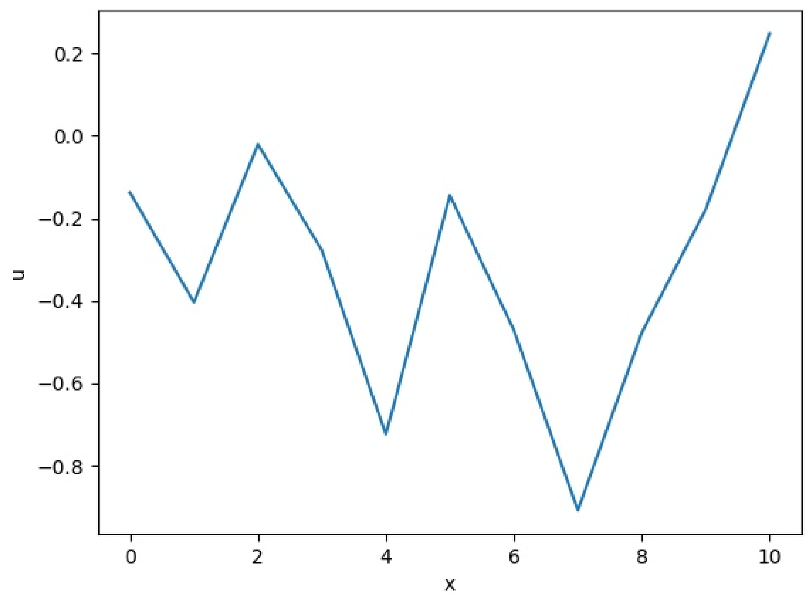

Σχήμα 27: Implicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 30$



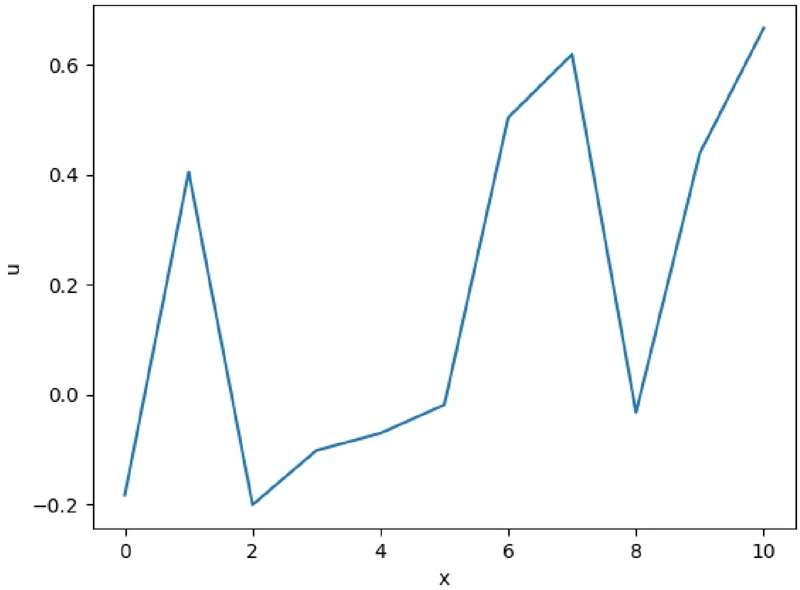
Σχήμα 28: Implicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 40$




Σχήμα 29: Implicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 50$



Σχήμα 30: Implicit μέθοδος με $c\Delta t = 1.1\Delta x$ για $N_t = 60$




## Συμπεράσματα

Βλέπουμε ότι μόνο στην περίπτωση της explicit μεθόδου για $c\Delta t = \Delta x$ έχουμε τον τετραγωνικό παλμό που περιμέναμε να δούμε σύμφωνα με τις αρχικές συνθήκες που επιλέξαμε. Αυτό συμβαίνει γιατί η αριθμητική λύση της κυματικής εξίσωσης δίνει την πραγματική λύση μόνο για $c\Delta t = \Delta x$, το λεγόμενο magic step. Σε όλες τις άλλες περιπτώσεις η λύση είναι απλά μια προσέγγιση και το σχήμα δεν είναι αυτό που περιμέναμε. Πράγματι στην περίπτωση $c\Delta t = 0.9\Delta x$ μπορούμε να δούμε ότι πάει να σχηματιστεί ο τετραγωνικός παλμός αλλά υπάρχουν διαταραχές στο σχήμα και τελικά το αποτέλεσμα είναι κάτι ακανόνιστο. Στην περίπτωση $c\Delta t = 1.1\Delta x$ βλέπουμε συνέχεια το ίδιο σχήμα που σε καμία περίπτωση δεν θυμίζει τετραγωνικό παλμό. Αυτό μπορεί να οφείλεται στο αν $\Delta t > \frac{\Delta x}{c}$ τότε η αριθμητική ταχύτητα επίλυσης είναι μικρότερη από την ταχύτητα διάδοσης του κύματος και επομένως δεν μπορούμε να λύσουμε την κυματική εξίσωση.

Με την implicit μέθοδο και στις τρεις περιπτώσεις παίρνουμε ένα παρόμοιο αποτέλεσμα που είναι αντίστοιχο με αυτό της explicit μεθόδου για $c\Delta t = 0.9\Delta x$, δηλαδή έχουμε ένα σχήμα που μοιάζει με τετραγωνικό παλμό αλλά δεν είναι ακριβώς. Αυτό μας λέει ότι η implicit μέθοδος δεν δίνει την πραγματική λύση ούτε για το magic step και ότι δεν επηρεάζεται από περιορισμούς στην ταχύτητα επίλυσης. Αυτό μπορεί να οφείλεται στο γεγονός ότι λύνουμε την κυματική εξίσωση σαν γραμμικό σύστημα. Στην επόμενη ενότητα δίνεται ο κώδικας που χρησιμοποιήσαμε.

## Κώδικας

Ο κώδικας είναι αρκετά απλός. Αρχικά ορίζουμε τους πίνακες που θα αποθηκεύσουμε την λύση για τις δύο μεθόδους ($ue$ και $ui$). Ορίζουμε τις αρχικές και οριακές συνθήκες σύμφωνα με όσα είπαμε παραπάνω και προχωράμε στην επίλυση, υπολογίζοντας τη λύση μέσω της σχέσης (1) για όλα τα χρονικά βήματα $N_t$ και όλα τα χωρικά βήματα $N_x$. Αντίστοιχα για την επίλυση μέσω της σχέσης (2) υπολογίζουμε τους πίνακες που χρειαζόμαστε και μετά την λύση για κάθε χρονική στιγμή. Τέλος φτιάχνουμε τα διαγράμματα για $N_t = 20, 30, 40, 50, 60$.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import diags

Nx = 11     #space values
Nt = 61     #time values

ue = np.zeros((Nx,Nt))  #explicit solution
ui = np.zeros((Nx,Nt))      #implicit solution
c = 1     #light speed
dx = 1    #space step
dt = 1.1  #time step

#initial values
for i in range(3,Nx):
    if i==10:
        ue[i,0] = 1
        ue[i,1] = 1
        ui[i,0] = 1
        ui[i,0] = 1
        continue
    ue[i,0] = 1
    ue[i+1,1] = 1
    ui[i,0] = 1
    ui[i+1,1] = 1

#dirichlet boundary conditions
for n in range(0,Nt):
    ue[0,n] = 0
    ue[Nx-1,n] = 0
    ui[0,n] = 0
    ui[Nx-1,n] = 0

# explicit scheme
for n in range(1,Nt-1):
    for i in range(1,Nx-1):
        ue[i,n+1] = (c*dt)**2*(ue[i+1,n]-2*ue[i,n]+ue[i-1,n])/dx**2 +2*ue[i,n]-ue[i,n-1]

#implicit scheme
beta = 0.25
I = np.identity(Nx)   #identity matrix
L = diags([-1, 2, -1], [-1, 0, 1], shape=(Nx, Nx)).toarray()  #tridiagonal matrix
A = beta*L + dx**2/(c*dt)**2*I
B = (2*beta-1)/2*L + dx**2/(c*dt)**2*I
AA = np.linalg.inv(A)
AB = np.matmul(AA,B)

for n in range(1,Nt-1):
    ui[:,n+1] = 2*AB.dot(ui[:,n])-ui[:,n-1]

#plot snapshot of the wave
x = np.arange(0,11,1) # x-axis values
for n in [20,30,40,50,60]:
    plt.plot(x,ue[:,n])
    plt.xlabel('x')
```

```
    plt.ylabel('u')
    plt.savefig('ue_'+str(n)+'_dt11.jpeg')
    plt.clf()
    plt.plot(x,ui[:,n])
    plt.xlabel('x')
    plt.ylabel('u')
    plt.savefig('ui_'+str(n)+'_dt11.jpeg')
    plt.clf()
```