4η Άσκηση

Ζητούμενο είναι να προεγγιστούν οι συναρτήσεις $f_1(x) = x^2 - 2x + 3$, $f_2(x) = \cos(\pi x)$ και $f_3(x) = e^{-x}$ με πολυώνυμο δευτέρου βαθμού, οι συντελεστές των όρων του οποίου υπολογίζονται με χρήση της μεθόδου ελαχίστων τετραγώνων.

Έστω ότι το πολυώνυμο που προκύπτει από την μέθοδο ελαχίστων τετραγώνων δίνεται από την εξίσωση 1, όπου $\{\phi_n(x)\}$ οικογένεια πολυωνύμων που θα χρησιμοποιηθούν σαν βάση για την προσέγγιση. Εν προκειμένω, ζητείται ο βαθμός του πολυωνύμου να είναι $n = 2$.

$$p_n(x) = a_0 + a_1\phi_1(x) + \cdots + a_n\phi_n(x) \tag{1}$$

Υπάρχουν διάφορες επιλογές για την οικογένεια των πολυωνύμων δευτέρου βαθμού. Η πιο απλή επιλογή είναι τα μονώνυμα $\{1, x, x^2\}$ αλλά υπάρχουν και ορθογώνια πολυώνυμα όπως τα πολυώνυμα Legendre $\{1, x, x^2 - \dfrac{1}{3}\}$. Η μέθοδος ελαχίστων τετραγώνων ελαχιστοποιεί το σφάλμα (2).

$$E = \int_a^b [f(x) - P_2(x)]^2\, dx \tag{2}$$

Η προσέγγιση με μονώνυμα για το διάστημα $[a, b]$ απαιτεί επίλυση του συστήματος (3), όπου $s_i = \int_a^b x^i dx$ και $b_i = \int_a^b x^i f(x) dx$.

$$\begin{bmatrix} s_0 & s_1 & s_2 \\ s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \tag{3}$$

Για $[a, b] = [0, 1]$ ο πίνακας είναι ο (4), ο πίνακας Hilbert, ο οποίος είναι αντιστρέψιμος για $n = 3$, για μεγαλύτερα n οδηγεί σε ασταθή συστήματα.

$$S = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix} \tag{4}$$

Για το λόγο αυτό, σε μεγαλύτερα n χρησιμοποιούνται ορθογώνια πολυώνυμα, όπως τα Legendre. Τα ορθογώνια πολυώνυμα μηδενίζουν όλα τα μη διαγώνια στοιχεία του Πίνακα 3 και η επίλυσή του είναι άμεση. Συγκεκριμένα, είναι $s_i = \int_a^b w(x)\phi_i^2(x)dx$ και $b_i = \int_a^b w(x)\phi_i(x)f(x)dx$, όπου $w(x) = 1$ κατάληλλη συνάρτηση βάρους.

Σε αντίθεση με την περίπτωση των μονώνυμων, η ολοκλήρωση πρέπει να γίνει στο διάστημα $[-1, 1]$, για να ισχύει η σχέση ορθογωνιότητας (5). Το ζητούμενο διάστημα προσέγγισης εξακολουθεί να είναι το $[0, 1]$ καθώς είναι υποσύνολο του $[-1, 1]$.

$$\int_{-1}^{1} \phi_i(x)\phi_j(x)dx = 0 \tag{5}$$

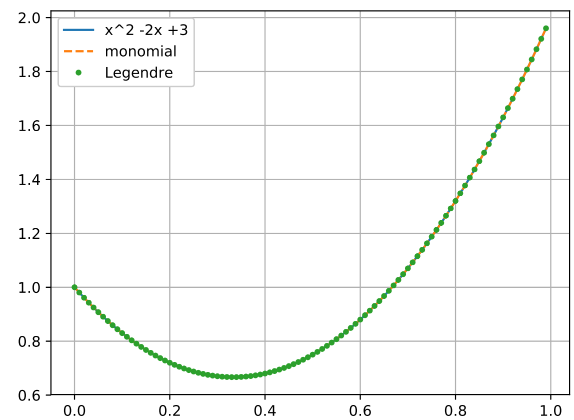
Για σύγκριση, το πρόβλημα λύνεται υπολογιστικά και με τις δύο προσεγγίσεις. Και στις δύο περιπτώσεις, αναμένεται για κάθε συνάρτηση $f_k(x)$ να παραχθεί ένα πολυώνυμο δευτέρου βαθμού. Ειδικότερα, η $f_1(x)$ είναι ήδη ένα πολυώνυμο δευτέρου βαθμού και επομένως αναμένεται η μέθοδος ελαχίστων τετραγώνων να παράγει σαν πολυώνυμο την ίδια την $f_1(x)$.

Στα Σχήματα 1, 2 και 3 φαίνονται τα αποτελέσματα της προσέγγισης. Όπως ήταν αναμενόμενο, η μέθοδος ελαχίστων τετραγώνων παράγει την ίδια την $f_1(x)$ στην πρώτη περίπτωση. Επίσης, στην δεύτερη περίπτωση, η λύση με μονώνυμα είναι μια ευθεία σε αντίθεση με την λύση με πολυώνυμα Legendre, που είναι δευτέρου βαθμού.

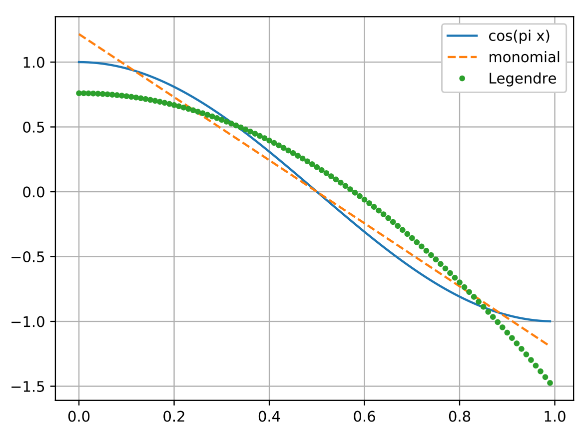
Παρακάτω ακολουθεί ο κώδικας που γράφτηκε σε Python και έγινε χρήση της βιβλιοθήκης Numpy. Οι ρουτίνες ελαχίστων τετραγώνων με μονώνυμα και με πολυώνυμα Legendre δίνονται στο Παράρτημα.

ex4.py

```
import numpy as np
import matplotlib.pyplot as plt
import interpolation


def f1(x):
    """Evaluates x^2-2x+3"""
    return np.polyval([3, -2, 1], x)


```



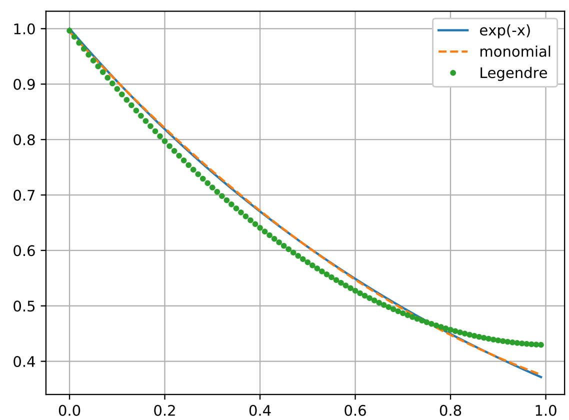
Σχήμα 1: Ελάχιστα Τετράγωνα για $x^2 - 2x + 3$



Σχήμα 2: Ελάχιστα Τετράγωνα για $\cos(\pi x)$

```python
def f2(x):
    """Evaluates cos(pi * x)"""
    return np.cos(np.pi*x)


def f3(x):
    """Evaluates e^(-x)"""
    return np.exp(-x)


def Legendre(x, n):
    """Evaluates Legendre polynomial
    of n degree"""
    c = np.zeros(n + 1)
    c[n] = 1.
    return np.polynomial.legendre.legval(x, c)


def wLegendre(x):
    """Weight function for Legendre
    polynomials used in Least Squares"""
    return np.ones(x)


a = 0
b = 1
order = 2

x = np.arange(a, b, 0.01)

plt.close('all')

p1 = interpolation.lSquaresMonomial(f1, order, a, b)
a1 = interpolation.lSquaresOrthogonal(f1,
                                      wLegendre,
                                      Legendre,
                                      3,
                                      -1,
                                      1)  # Base Legendre
    polynomials
```

```
plt.figure(1)
plt.plot(x, f1(x), label='x^2 -2x +3')
plt.plot(x, p1(x), '--', label='monomial')
plt.plot(x, np.polynomial.legendre.legval(x, a1), '.', label='Legendre')
plt.legend()
plt.grid()

p2 = interpolation.lSquaresMonomial(f2, order, a, b)
a2 = interpolation.lSquaresOrthogonal(f2,
                                      wLegendre,
                                      Legendre,
                                      3,
                                      -1,
                                      1)  # Base Legendre polynomials
plt.figure(2)
plt.plot(x, f2(x), label='cos(pi x)')
plt.plot(x, p2(x), '--', label='monomial')
plt.plot(x, np.polynomial.legendre.legval(x, a2), '.', label='Legendre')
plt.legend()
plt.grid()

p3 = interpolation.lSquaresMonomial(f3, order, a, b)
a3 = interpolation.lSquaresOrthogonal(f3,
                                      wLegendre,
                                      Legendre,
                                      3,
                                      -1,
                                      1)  # Base Legendre polynomials
plt.figure(3)
plt.plot(x, f3(x), label='exp(-x)')
plt.plot(x, p3(x), '--', label='monomial')
plt.plot(x, np.polynomial.legendre.legval(x, a3), '.', label='Legendre')
plt.legend()
plt.grid()
```



Σχήμα 3: Ελάχιστα Τετράγωνα για e^{-x}