

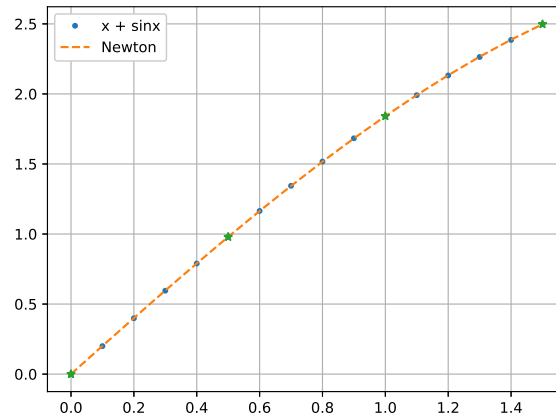
2η Άσκηση

Το πολυώνυμο παρεμβολής Newton μπορεί να βρεθεί με χρήση του αλγορίθμου διηρημένων διαφορών. Συγκεκριμένα το πολυώνυμο που προκύπτει είναι της μορφής (1), όπου οι συντελεστές a_i δίνονται από τις διηρημένες διαφορές(2) και $x_i, i = 0, 1, \dots, n$ τα σημεία παρεμβολής.

$$P_n(x) = a_0 + a_1(x - x_0) + \dots + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (1)$$

$$a_i = f[x_0, x_1, \dots, x_i] \quad (2)$$

Το αποτέλεσμα της παρεμβολής δίνεται στο Σχήμα 1.



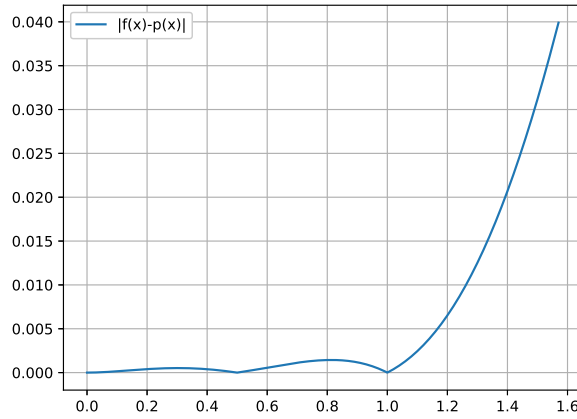
Σχήμα 1: Παρεμβολή με πολυώνυμο Newton

Ζητείται επίσης μια προσέγγιση για το άνω σφάλμα προσέγγισης $|P(x) - f(x)|$ στο $x \in [0, \frac{\pi}{2}]$. Γενικότερα, το σφάλμα στην πολυωνυμική παρεμβολή αποδεικνύεται ότι δίνεται από την εξίσωση (3), με $\xi \in (a, b)$, όπου $a = \min[x_0, \dots, x_n]$ και $b = \max[x_0, \dots, x_n]$.

$$f(x) - p_n(x) = \frac{(x - x_0) \cdots (x - x_n)}{(n + 1)!} f^{(n+1)}(\xi) \quad (3)$$

Λαμβάνοντας την απόλυτη τιμή της (3) και για την δοσμένη $f(x)$ προκύπτει η (4). Στο Σχήμα 2, δίνεται η γραφική παράσταση του σφάλματος στο υπό εξέταση διάστημα.

$$\begin{aligned} |f(x) - p_n(x)| &= \left| \frac{x(x - \frac{1}{2})(x - 1)}{4!} \sin \xi \right| \rightarrow \\ &\leq \frac{|x||x - \frac{1}{2}||x - 1|}{4!} \rightarrow \\ &\leq \frac{\frac{\pi}{2}(\frac{\pi}{2} - \frac{1}{2})(\frac{\pi}{2} - 1)}{4!} \rightarrow \\ &= \frac{\pi^3 - 3\pi^2 + 2\pi}{192} \approx 0.04 \end{aligned} \quad (4)$$



Σχήμα 2: Παρεμβολή με πολυώνυμο Lagrange

Παρακάτω ακολουθεί ο κώδικας που γράφτηκε σε Python και έγινε χρήση

της βιβλιοθήκης Numpy. Ο αλγόριθμος διηρημένων διαφορών και φωλιασμένου υπολογισμού της τιμής του πολυωνύμου δίνεται στο Παράρτημα.

ex2.py

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

6 def f(x):
    return x + np.sin(x)

def er(x):
11     return x * (x - .5) * (x - 1) * np.sin(x) / np.math.
        factorial(4)

xValues = np.array([0, 0.5, 1, 1.5])
yValues = f(xValues)
16
coefficients = interpolation.dividedDifferenceTable(xValues,
    yValues)
p = interpolation.NestedMultiplication(0, xValues, coefficients)

plt.close('all')
21 x = np.arange(0, 1.6, 0.1)
y = interpolation.NestedMultiplication(x, xValues, coefficients)
plt.figure(1)
plt.plot(x, f(x), '. ', label='x + sinx')
plt.plot(x, y, '—', label='Newton')
26 plt.plot(xValues, yValues, '*')
plt.legend()
plt.grid()

31 x = np.arange(0, np.pi/2, 0.01)
err = np.abs(er(x))
plt.figure(2)
plt.plot(x, err, label='| f(x)-p(x) |')
```

```
plt.legend()
36 plt.grid()
error = (np.power(np.pi, 3) - 3 * np.power(np.pi, 2) + 2 * np.pi
        ) / 192
print('Theoretical Error:', error)
print('Graphical Error:', max(err))
```