2Η ΕΡΓΑΣΙΑ ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΛΥΣΗ

ΑΘΑΝΑΣΙΟΣ ΞΑΝΘΟΠΟΥΛΟΣ  
ΑΕΜ: 2392

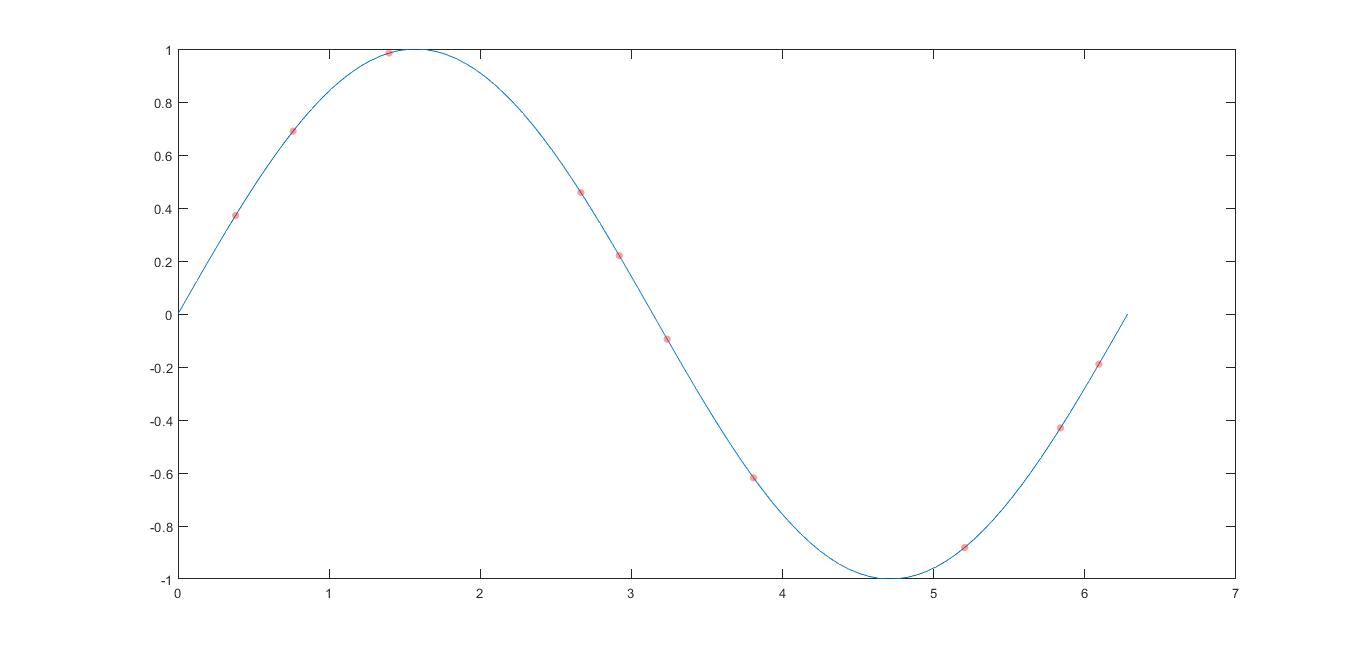
* **Άσκηση 5 (κώδικας στο αρχείο askisi5\_2392.py)**

**Τα σημεία που επέλεξα(στον οριζόντιο άξονα χ) για να δημιουργήσω την προσέγγιση της συνάρτησης του ημιτόνου είναι:**

x\_my\_sin\_points **=** **[**0.380799109526036**,** 0.761598219052071**,** 1.39626340159546**,** 2.66559376668225**,**

2.91945983969961**,** 3.2367924309713**,** 3.80799109526036**,**

5.20425449685582**,** 5.83891967939921**,** 6.09278575241657**]**

****

* + 1. **πολυωνυμική προσέγγιση: Υλοποιείται με την μέθοδο Newton στην συνάρτηση**

1. **def** polyonimiki\_prosegisi\_newton**(**x\_points**,** y\_points**):**
3. n **=** len**(**x\_points**)**
4. # Το πολυώνυμο 1x+0 (μεταβλητή x σε συμβατή μορφη)
5. x **=** numpy**.**poly1d**([**1**,** 0**])**
6. # Διαιρεμένες διαφορές υπολογισμένες στον πίνακα Dij
7. Dij **=** coef**(**x\_points**,** y\_points**)**
8. N **=** 0
9. **for** j **in** range**(**0**,** n**):**
10. nj **=** 1
11. **for** i **in** range**(**0**,** j**):**
12. nj **=** numpy**.**polymul**(**numpy**.**polyadd**(**x**,** **-**x\_points**[**i**]),** nj**)**
13. N **=** numpy**.**polyadd**(**numpy**.**polymul**(**Dij**[**j**],** nj**),** N**)**
14. **return** N

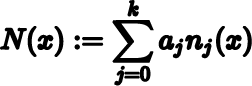
**Επιστρέφει το πολυώνυμο(προσέγγιση της συνάρτησης) που περνάει από τα σημεία που παίρνει σαν όρισμα, με την μέθοδο του Newton.**

**Η μορφή της επιστρεφόμενης μεταβλητής είναι: numpy.poly1d**

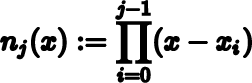
**όπου αποτελεί δομή δεδομένων που προσομοιάζει τα πολυώνυμα**

**Ο υπολογισμός των διαιρεμένων διαφορών γίνεται στην συνάρτηση coef.**

**Έπειτα από την γραμμή 13 και μετά υπολογίζω του όρους σύμφωνα με τον τύπο:**

****

**όπου**

****

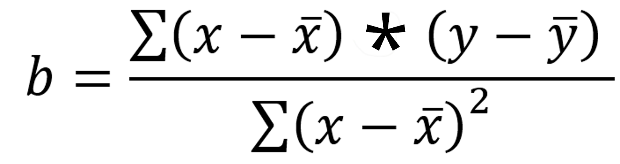
* + 1. **μέθοδος ελαχίστων τετραγώνων: Υλοποιείται με την βοήθεια της δομής πολυωνύμων από την βιβλιοθήκη numpy**

1. **def** methodos\_elaxistwn\_tetragwnwn**(**x\_points**,** y\_points**):**
2. # προσέγγιση των σημειών με πολυώνυμο 1ου βαθμού
3. # με την μέθοδο των ελαχίστων τετραγώνων
4. x\_meso **=** numpy**.**mean**(**x\_points**)**
5. y\_meso **=** numpy**.**mean**(**y\_points**)**
6. sum\_arith **=** 0
7. sum\_paron **=** 0
8. **for** i **in** range**(**len**(**x\_points**)):**
9. # Δημιουργούμε το άθροισμα του αριθμητή
10. sum\_arith **+=** **(**x\_points**[**i**]** **-** x\_meso**)** **\*** **(**y\_points**[**i**]** **-** y\_meso**)**
11. # Δημιουργούμε το άθροισμα του παρονομαστή
12. sum\_paron **+=** **(**x\_points**[**i**]** **-** x\_meso**)** **\*\*** 2
13. b **=** sum\_arith **/** sum\_paron
14. a **=** y\_meso **-** b **\*** x\_meso
15. eutheia\_proseggisis **=** numpy**.**poly1d**([**b**,** a**])**
16. **return** eutheia\_proseggisis

**Επιστρέφει το πολυώνυμο(προσέγγιση της συνάρτησης) που περνάει από τα σημεία που παίρνει σαν όρισμα, με την μέθοδο των ελάχιστων τετραγώνων.**

**Η μορφή της επιστρεφόμενης μεταβλητής είναι: numpy.poly1d**

**Υπολογίζω του όρους σύμφωνα με τον τύπο:**

****

**Όπου Y = a + bX**

**Μέσο σφάλμα πολυωνυμικής προσέγγισης Newton: 0.5047188139829841**

**Σφάλμα προσέγγισης με την μέθοδο των ελάχιστων τετραγώνων: 18.88863678035914**

* **Άσκηση 6 (κώδικας στο αρχείο askisi6\_2392.py)**