Αριθμητική Ανάλυση :

Τελική εξαμηνιαία εργασία

Σεπτέμβριος 2021

Στοιχεία Φοιτητή :

Γιοάνι Μπραούνι

Dit18131

Άσκηση 1
Μέρος I(i)

Μέθοδος της Διχοτόμησης(bisection)

Για να εφαρμοστεί η μέθοδος της διχοτόμησης πρέπει να ισχύουν τα εξής:

- πρέπει να είναι συνεχής στο κλειστό διάστημα [α,β]
- πρέπει να ισχύει $f(a)f(b)<0$

Άρα για την εξίσωση $x^3 - 8x^2 + 17x - 10 = 0$ μπορούμε να επιλέξουμε τα διαστήματα :

[0,1.5] , [1.7,3] , [4,5.5]

Διάστημα [0,1.5]

```
def f(x):
    return x**3-8*x**2+17*x-10
def bisection(a,b):
    x=(a+b)/2
    if f(x)*f(a)<0:
        b=x
    else:
        a=x
    return a,b,x
a,b=0,1.5
solution_list=[]#αρχικοποίηση
for i in range(20):
    temp_a,temp_b=a,b
    a,b,x=bisection(a,b)
    solution_list.append([i,temp_a,temp_b,x])
solution_list
```

```
[[0, 0, 1.5, 0.75],
 [1, 0.75, 1.5, 1.125],
 [2, 0.75, 1.125, 0.9375],
 [3, 0.9375, 1.125, 1.03125],
 [4, 0.9375, 1.03125, 0.984375],
 [5, 0.984375, 1.03125, 1.0078125],
 [6, 0.984375, 1.0078125, 0.99609375],
 [7, 0.99609375, 1.0078125, 1.001953125],
 [8, 0.99609375, 1.001953125, 0.9990234375],
 [9, 0.9990234375, 1.001953125, 1.00048828125],
 [10, 0.9990234375, 1.00048828125, 0.999755859375],
 [11, 0.999755859375, 1.00048828125, 1.0001220703125],
 [12, 0.999755859375, 1.0001220703125, 0.99993896484375],
 [13, 0.99993896484375, 1.0001220703125, 1.000030517578125],
 [14, 0.99993896484375, 1.000030517578125, 0.9999847412109375],
 [15, 0.9999847412109375, 1.000030517578125, 1.0000076293945312],
 [16, 0.9999847412109375, 1.0000076293945312, 0.9999961853027344],
 [17, 0.9999961853027344, 1.0000076293945312, 1.0000019073486328],
 [18, 0.9999961853027344, 1.0000019073486328, 0.9999990463256836],
 [19, 0.9999990463256836, 1.0000019073486328, 1.0000004768371582]]
```

F(0) = 0^3 –8*0^2 + 17*0 – 10 = -10

F(1.5) = 1.5^3 – *1.5^2 + 17*1.5 - 10 = 0.875

F(0)*F(1.5)<0 επίσης είναι συνεχής στο διάστημα [0,1.5]

Βλέπουμε ότι με προσέγγιση 6 δεκαδικών ψηφίων, η ρίζα δεν έχει απόλυτο και σχετικό σφάλμα

Διάστημα [1.7,3]

```python
def f(x):
    return x**3-8*x**2+17*x-10
def bisection(a,b):
    x=(a+b)/2
    if f(x)*f(a)<0:
        b=x
    else:
        a=x
    return a,b,x
a,b=1.7,3
solution_list=[]#αρχικοποίηση
for i in range(20):
    temp_a,temp_b=a,b
    a,b,x=bisection(a,b)
    solution_list.append([i,temp_a,temp_b,x])
solution_list
```

```
[[0, 1.7, 3, 2.35],
 [1, 1.7, 2.35, 2.025],
 [2, 1.7, 2.025, 1.8624999999999998],
 [3, 1.8624999999999998, 2.025, 1.9437499999999999],
 [4, 1.9437499999999999, 2.025, 1.984375],
 [5, 1.984375, 2.025, 2.0046875],
 [6, 1.984375, 2.0046875, 1.99453125],
 [7, 1.99453125, 2.0046875, 1.9996093750000001],
 [8, 1.9996093750000001, 2.0046875, 2.0021484375000003],
 [9, 1.9996093750000001, 2.0021484375000003, 2.00087890625],
 [10, 1.9996093750000001, 2.00087890625, 2.000244140625],
 [11, 1.9996093750000001, 2.000244140625, 1.9999267578125002],
 [12, 1.9999267578125002, 2.000244140625, 2.00008544921875],
 [13, 1.9999267578125002, 2.00008544921875, 2.0000061035156254],
 [14, 1.9999267578125002, 2.0000061035156254, 1.9999664306640628],
 [15, 1.9999664306640628, 2.0000061035156254, 1.999986267089844],
 [16, 1.999986267089844, 2.0000061035156254, 1.9999961853027348],
 [17, 1.9999961853027348, 2.0000061035156254, 2.00000114440918],
 [18, 1.9999961853027348, 2.00000114440918, 1.9999986648559573],
 [19, 1.9999986648559573, 2.00000114440918, 1.9999999046325687]]
```

F(1.7) = 1.7^3 –8*1.7^2 + 17*1.7– 10 = 0.693

F(3) = 3^3 – 8*3^2 + 17*3 -10 = -4

F(1.7)*F(3)<0 επίσης είναι συνεχής στο διάστημα [1.7,3]

Βλέπουμε ότι με προσέγγιση 6 δεκαδικών ψηφίων, η ρίζα έχει απόλυτο σφάλμα 0,000001 και σχετικό σφάλμα 5e-7

Διάστημα [4,5.5]

```
def f(x):
    return x**3-8*x**2+17*x-10
def bisection(a,b):
    x=(a+b)/2
    if f(x)*f(a)<0:
        b=x
    else:
        a=x
    return a,b,x
a,b=4,5.5
solution_list=[]#αρχικοποίηση
for i in range(20):
    temp_a,temp_b=a,b
    a,b,x=bisection(a,b)
    solution_list.append([i,temp_a,temp_b,x])
solution_list
```

```
[[0, 4, 5.5, 4.75],
 [1, 4.75, 5.5, 5.125],
 [2, 4.75, 5.125, 4.9375],
 [3, 4.9375, 5.125, 5.03125],
 [4, 4.9375, 5.03125, 4.984375],
 [5, 4.984375, 5.03125, 5.0078125],
 [6, 4.984375, 5.0078125, 4.99609375],
 [7, 4.99609375, 5.0078125, 5.001953125],
 [8, 4.99609375, 5.001953125, 4.9990234375],
 [9, 4.9990234375, 5.001953125, 5.00048828125],
 [10, 4.9990234375, 5.00048828125, 4.999755859375],
 [11, 4.999755859375, 5.00048828125, 5.0001220703125],
 [12, 4.999755859375, 5.0001220703125, 4.99993896484375],
 [13, 4.99993896484375, 5.0001220703125, 5.000030517578125],
 [14, 4.99993896484375, 5.000030517578125, 4.9999847412109375],
 [15, 4.9999847412109375, 5.000030517578125, 5.000007629394531],
 [16, 4.9999847412109375, 5.000007629394531, 4.999996185302734],
 [17, 4.999996185302734, 5.000007629394531, 5.000001907348633],
 [18, 4.999996185302734, 5.000001907348633, 4.999999046325684],
 [19, 4.999999046325684, 5.000001907348633, 5.000000476837158]]
```

F(4) = 4^3 –8*4^2 + 17*4– 10 = -6

F(5.5) = 5.5^3 – 8*5.5^2 + 17*5.5 -10 = 7.875

F(4)*F(5.5)<0 επίσης είναι συνεχής στο διάστημα [4,5.5]

Βλέπουμε ότι με προσέγγιση 6 δεκαδικών ψηφίων, η ρίζα δεν έχει απόλυτο και σχετικό σφάλμα

Άσκηση 1
Μέρος I(ii)

Οι διαφορετικές μορφές που μπορεί να πάρει η εξίσωση για να έρθει στη μορφή που απαιτεί η μέθοδος των διαδοχικών επαναλήψεων είναι :

1) g(x) = 10/(x^2 – 8*x + 17)

2) g(x) = (8*x^2 – 17*x + 10)^1/3

3) g(x) = ((x^3 + 17*x – 10)/8)^½

1) g(x) = 10/(x^2 – 8*x + 17)

Σύμφωνα με το θεώρημα διαδοχικών επαναλήψεων η συνάρτηση g(x) έχει μοναδικό σταθερό σημείο p στο διάστημα [0,1.5] , άρα η ακολουθία p_n με p_n+1 = g(p_n) και με τυχαίο p_0 E[0,1.5] συγκλίνει στον αριθμό p

Με αρχικη τιμη p = 0,75 :

1) g(x) = 10/(x^2 – 8*x + 17)

```
def g(x):
    return 10/(x**2-8*x+17)
solution_list=[]
def repetitive(x):
    temp=x
    x=g(x)
    return temp,x
x=0.75
for i in range(10):
    temp,x=repetitive(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(x,i)
        break
solution_list
```

```
[[0, 0.8648648648648649],
 [1, 0.9234401349072513],
 [2, 0.9555460299285096],
 [3, 0.9738331067695912],
 [4, 0.9844761824037364],
 [5, 0.9907480092438256],
 [6, 0.994470985575204],
 [7, 0.9966905233486281],
 [8, 0.9980171582184881],
 [9, 0.9988113164151762]]
```

Η ακολουθία συγκλίνει αργά στην ρίζα

Η ρίζα x της g(x) είναι 1 ενώ η θεωρητική ρίζα x* με προσέγγιση 3 δεκαδικών ψηφίων είναι 0.998 οπότε προκύπτει :

Απόλυτο σφάλμα : x* - x = -0,002

Σχετικό σφάλμα : 0,002

2) $g(x) = (8*x^2 – 17*x + 10)^1/3$

Σύμφωνα με το θεώρημα διαδοχικών επαναλήψεων η συνάρτηση g(x) έχει μοναδικό σταθερό σημείο p στο διάστημα [0,1.5] , άρα η ακολουθία p_n με p_n+1 = g(p_n) και με τυχαίο p_0 Ε[0,1.5] συγκλίνει στον αριθμό p

Με αρχικη τιμη p = 0,75 :

2) g(x) = (8*x^2 – 17*x + 10)^1/3

```python
def g(x):
    return (8*x**2 - 17*x + 10)**(1/3)
solution_list=[]
def repetitive(x):
    temp=x
    x=g(x)
    return temp,x
x=0.75
for i in range(10):
    temp,x=repetitive(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(x,i)
        break
solution_list
```

```
[[0, 1.205071132087615],
 [1, 1.0419988130120208],
 [2, 0.9906163523569721],
 [3, 1.0033514453949908],
 [4, 0.9989116198901223],
 [5, 1.0003658183872517],
 [6, 0.9998784026138374],
 [7, 1.0000405702452189],
 [8, 0.9999864807913449],
 [9, 1.0000045068699577]]
```

Η ακολουθία συγκλίνει στην ρίζα

Δεν υπάρχει απόλυτο και σχετικό σφάλμα

3) $g(x) = ((x^3 + 17*x - 10)/8)^{1/2}$

- Σύμφωνα με το θεώρημα διαδοχικών επαναλήψεων η συνάρτηση $g(x)$ έχει μοναδικό σταθερό σημείο p στο διάστημα $[1.7,2.2]$, άρα η ακολουθία p_n με $p_{n+1} = g(p_n)$ και με τυχαίο p_0 E$[1.7,2.2]$ συγκλίνει στον αριθμό p

Με αρχικη τιμη $p = 1.9$:

3) $g(x) = ((x^3 + 17*x – 10)/8)^{1/2}$

```
def g(x):
    return ((x**3 + 17*x - 10)/8)**(1/2)
solution_list=[]
def repetitive(x):
    temp=x
    x=g(x)
    return temp,x
x=1.9
for i in range(10):
    temp,x=repetitive(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(x,i)
        break
solution_list
```

```
[[0, 1.9091555724979565],
 [1, 1.917494398615474‌2],
 [2, 1.9250850297909456],
 [3, 1.9319911664768792],
 [4, 1.9382717973073977],
 [5, 1.9439813894623128],
 [6, 1.949170112240732],
 [7, 1.9538840802323119],
 [8, 1.958165605910240‌8],
 [9, 1.9620534541490438]]
```

Η ακολουθία συγκλίνει αργά στην ρίζα

Η ρίζα x της g(x) είναι 2 ενώ η θεωρητική ρίζα x* με προσέγγιση 3 δεκαδικών ψηφίων είναι 1.962 οπότε προκύπτει :

Απόλυτο σφάλμα : x* - x = 0,038

Σχετικό σφάλμα : 0,019

Άσκηση 1

Μέρος I(iii)

F(x) = x^3 – 8*x^2 + 17x – 10

F'(x) = 3*x^2 – 16*x + 17

Διάστημα [0,1.5] , [1.7,3] , [4,5.5]

Είναι προφανές ότι μια ρίζα της συνάρτησης βρίσκεται στο διάστημα

[0 ,1.5] , [1.7,3] , [4,5.5]

Είναι F(0) = -10 , F(1.5) = 0.875 , F(1.7) = 0.693 , F(3) = -4 , F(4) = -6 , F(5.5) = 7.875

και η συνάρτηση f είναι συνεχής στο

Κλειστό διάστημα [0 , 1.5] , [1.7,3] , [4,5.5] συνεπώς έχει μία τουλάχιστον

Πραγματική Ρίζα στο ανοικτό διάστημα (0,1.5) , (1.7,3) , (4,5.5)

Εφαρμόζοντας την μέθοδο του Newton έχω:

Διάστημα [0,1.5]

```
def f(x):
    return x**3-8*x**2+17*x-10,3*x**2-16*x+17
def newton_raphson(x):
    temp=x
    x=x-f(x)[0]/f(x)[1]
    return x,temp
solution_list=[]
x=0
for i in range(20):
    x,temp=newton_raphson(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(i,x)
        break
solution_list
```

```
6 0.9999999999999998
```

```
[[0, 0.5882352941176471],
 [1, 0.885538330855808],
 [2, 0.9867847908705343],
 [3, 0.9997875892974138],
 [4, 0.9999999436272622],
 [5, 0.9999999999999962],
 [6, 0.9999999999999998]]
```

Η ζητούμενη ρίζα x* δεν εχει απόλυτο και σχετικό σφάλμα

Διάστημα [1.7,3]

```python
def f(x):
    return x**3-8*x**2+17*x-10,3*x**2-16*x+17
def newton_raphson(x):
    temp=x
    x=x-f(x)[0]/f(x)[1]
    return x,temp
solution_list=[]
x=1.7
for i in range(20):
    x,temp=newton_raphson(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(i,x)
        break
solution_list
```

```
5 1.9999999999999998
```

```
[[0, 2.1529411764705886],
 [1, 2.0111890684052285],
 [2, 2.0000813265134902],
 [3, 2.000000004408497],
 [4, 2.000000000000001],
 [5, 1.9999999999999998]]
```

Η ζητούμενη ρίζα x* δεν εχει απόλυτο και σχετικό σφάλμα

Διάστημα [4,5.5]

```
def f(x):
    return x**3-8*x**2+17*x-10,3*x**2-16*x+17
def newton_raphson(x):
    temp=x
    x=x-f(x)[0]/f(x)[1]
    return x,temp
solution_list=[]
x=4
for i in range(20):
    x,temp=newton_raphson(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(i,x)
        break
solution_list
```

```
8 4.999999999999999

[[0, 10.0],
 [1, 7.707006369426752],
 [2, 6.265529178844164],
 [3, 5.442168945858322],
 [4, 5.082095185205586],
 [5, 5.003666329626171],
 [6, 5.000007815906755],
 [7, 5.000000000035634],
 [8, 4.999999999999999]]
```

Η ζητούμενη ρίζα x* δεν εχει απόλυτο και σχετικό σφάλμα

Άσκηση 1
Μέρος ΙΙ(i)

- $(x - 2a)(x - 3b)(x - c) = 0$

AM : 2022201800131

A = 1 + (31 mod 3) = 1 + 1 = 2

B = 1 + (800 mod 7) = 1 + 2 = 3

C = 1 + (2201 mod 11) = 1 + 1 =2

x^3 – 15*x^2 + 62*x - 72

Άσκηση 1
Μέρος II(i)

Για την εξίσωση x^3 – 15*x^2 + 62*x – 72 = 0 μπορούμε να επιλέξουμε τα διαστήματα :

[3.7,4.7] , [8.7,9.7] , [1.7,2.7]

Για x1 = 4 , f(3.7) = 2.703 και f(4.7) = -8.127 αρα f(3.7)*f(4.7) < 0

Για x2 = 9 , f(8.7) = -9.447 και f(9.7) = 30.723 αρα f(8.7)*f(9.7) < 0

Για x3 = 2 , f(1.7) = -5.037 και f(2.7) = 5783 αρα f(1.5)*f(2.5) < 0

Διάστημα [3.7,4.7]

```
def f(x):
    return x**3-15*x**2+62*x-72
def bisection(a,b):
    x=(a+b)/2
    if f(x)*f(a)<0:
        b=x
    else:
        a=x
    return a,b,x
a,b=3.7,4.7
solution_list=[]#αρχικοποίηση
for i in range(20):
    temp_a,temp_b=a,b
    a,b,x=bisection(a,b)
    solution_list.append([i+1,temp_a,temp_b,x])
solution_list
```

```
[[1, 3.7, 4.7, 4.2],
 [2, 3.7, 4.2, 3.95],
 [3, 3.95, 4.2, 4.075],
 [4, 3.95, 4.075, 4.0125],
 [5, 3.95, 4.0125, 3.98125],
 [6, 3.98125, 4.0125, 3.996875],
 [7, 3.996875, 4.0125, 4.0046875],
 [8, 3.996875, 4.0046875, 4.00078125],
 [9, 3.996875, 4.00078125, 3.998828125],
 [10, 3.998828125, 4.00078125, 3.9998046875],
 [11, 3.9998046875, 4.00078125, 4.00029296875],
 [12, 3.9998046875, 4.00029296875, 4.000048828125],
 [13, 3.9998046875, 4.000048828125, 3.9999267578125],
 [14, 3.9999267578125, 4.000048828125, 3.99998779296875],
 [15, 3.99998779296875, 4.000048828125, 4.000018310546875],
 [16, 3.99998779296875, 4.000018310546875, 4.000003051757813],
 [17, 3.99998779296875, 4.000003051757813, 3.9999954223632814],
 [18, 3.9999954223632814, 4.000003051757813, 3.999999237060547],
 [19, 3.999999237060547, 4.000003051757813, 4.00000114440918],
 [20, 3.999999237060547, 4.00000114440918, 4.0000001907348635]]
```

Βλέπουμε ότι με προσέγγιση 6 δεκαδικών ψηφίων, η ρίζα δεν έχει απόλυτο και σχετικό σφάλμα

Διάστημα [8.7,9.7]

```python
def f(x):
    return x**3-15*x**2+62*x-72
def bisection(a,b):
    x=(a+b)/2
    if f(x)*f(a)<0:
        b=x
    else:
        a=x
    return a,b,x
a,b=8.7,9.7
solution_list=[]#αρχικοποίηση
for i in range(20):
    temp_a,temp_b=a,b
    a,b,x=bisection(a,b)
    solution_list.append([i+1,temp_a,temp_b,x])
solution_list
```

```
[[1, 8.7, 9.7, 9.2],
 [2, 8.7, 9.2, 8.95],
 [3, 8.95, 9.2, 9.075],
 [4, 8.95, 9.075, 9.0125],
 [5, 8.95, 9.0125, 8.98125],
 [6, 8.98125, 9.0125, 8.996875],
 [7, 8.996875, 9.0125, 9.0046875],
 [8, 8.996875, 9.0046875, 9.00078125],
 [9, 8.996875, 9.00078125, 8.998828125],
 [10, 8.998828125, 9.00078125, 8.9998046875],
 [11, 8.9998046875, 9.00078125, 9.00029296875],
 [12, 8.9998046875, 9.00029296875, 9.000048828125],
 [13, 8.9998046875, 9.000048828125, 8.9999267578125],
 [14, 8.9999267578125, 9.000048828125, 8.99998779296875],
 [15, 8.99998779296875, 9.000048828125, 9.000018310546874],
 [16, 8.99998779296875, 9.000018310546874, 9.000003051757812],
 [17, 8.99998779296875, 9.000003051757812, 8.99999542236328],
 [18, 8.99999542236328, 9.000003051757812, 8.999999237060546],
 [19, 8.999999237060546, 9.000003051757812, 9.000001144409179],
 [20, 8.999999237060546, 9.000001144409179, 9.000000190734863]]
```

Βλέπουμε ότι με προσέγγιση 6 δεκαδικών ψηφίων, η ρίζα δεν έχει απόλυτο και σχετικό σφάλμα

Διάστημα [1.7,2.7]

```python
def f(x):
    return x**3-15*x**2+62*x-72
def bisection(a,b):
    x=(a+b)/2
    if f(x)*f(a)<0:
        b=x
    else:
        a=x
    return a,b,x
a,b=1.7,2.7
solution_list=[]#αρχικοποίηση
for i in range(20):
    temp_a,temp_b=a,b
    a,b,x=bisection(a,b)
    solution_list.append([i+1,temp_a,temp_b,x])
solution_list
```

```
[[1, 1.7, 2.7, 2.2],
 [2, 1.7, 2.2, 1.9500000000000002],
 [3, 1.9500000000000002, 2.2, 2.075],
 [4, 1.9500000000000002, 2.075, 2.0125],
 [5, 1.9500000000000002, 2.0125, 1.9812500000000002],
 [6, 1.9812500000000002, 2.0125, 1.9968750000000002],
 [7, 1.9968750000000002, 2.0125, 2.0046875],
 [8, 1.9968750000000002, 2.0046875, 2.00078125],
 [9, 1.9968750000000002, 2.00078125, 1.9988281250000002],
 [10, 1.9988281250000002, 2.00078125, 1.9998046875000002],
 [11, 1.9998046875000002, 2.00078125, 2.00029296875],
 [12, 1.9998046875000002, 2.00029296875, 2.000048828125],
 [13, 1.9998046875000002, 2.000048828125, 1.9999267578125002],
 [14, 1.9999267578125002, 2.000048828125, 1.9999877929687502],
 [15, 1.9999877929687502, 2.000048828125, 2.000018310546875],
 [16, 1.9999877929687502, 2.000018310546875, 2.0000030517578127],
 [17, 1.9999877929687502, 2.0000030517578127, 1.9999954223632814],
 [18, 1.9999954223632814, 2.0000030517578127, 1.999999237060547],
 [19, 1.999999237060547, 2.0000030517578127, 2.00000114440918],
 [20, 1.999999237060547, 2.00000114440918, 2.0000001907348635]]
```

Βλέπουμε ότι με προσέγγιση 6 δεκαδικών ψηφίων, η ρίζα δεν έχει απόλυτο και σχετικό σφάλμα

Άσκηση 1
Μέρος ΙΙ(ii)

Οι διαφορετικές μορφές που μπορεί να πάρει η εξίσωση για να έρθει στη μορφή που απαιτεί η μέθοδος των διαδοχικών επαναλήψεων είναι :

1) g(x) = 72/(x^2 – 15*x + 62)

2) g(x) = (15*x^2 – 62*x + 72)^1/3

3) g(x) = ((x^3 + 62*x – 72)/15)^½

1) $g(x) = 72/(x^2 – 15*x + 62)$

Σύμφωνα με το θεώρημα διαδοχικών επαναλήψεων η συνάρτηση $g(x)$ έχει μοναδικό σταθερό σημείο p στο διάστημα [1.7,2.7] , άρα η ακολουθία p_n με $p_{n+1} = g(p_n)$ και με τυχαίο p_0 Ε[1.7,2.7] συγκλίνει στον αριθμό p

Με αρχικη τιμη p = 1,7 :

1) g(x) = 72/(x^2 – 15*x + 62)

```python
def g(x):
    return 72/(x**2-15*x+62)
solution_list=[]
def repetitive(x):
    temp=x
    x=g(x)
    return temp,x
x=1.7
for i in range(10):
    temp,x=repetitive(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(x,i)
        break
solution_list
```

```
[[0, 1.827875095201828],
 [1, 1.8985839259453297],
 [2, 1.9393489081208994],
 [3, 1.9634128612669308],
 [4, 1.977815657900825],
 [5, 1.9865072066759266],
 [6, 1.991778228169287],
 [7, 1.9949844380157906],
 [8, 1.9969382311957833],
 [9, 1.9981301480715814]]
```

Η ακολουθία συγκλίνει αργά στην ρίζα

Η ρίζα x της g(x) είναι 2 ενώ η θεωρητική ρίζα x* με προσέγγιση 3 δεκαδικών ψηφίων είναι 1.998 οπότε προκύπτει :

Απόλυτο σφάλμα : x* - x = 0,002

Σχετικό σφάλμα : 0,001

2) $g(x) = (15*x^2 – 62*x + 72)^1/3$

Σύμφωνα με το θεώρημα διαδοχικών επαναλήψεων η συνάρτηση $g(x)$ έχει μοναδικό σταθερό σημείο p στο διάστημα $[1.7,2.7]$, άρα η ακολουθία p_n με $p\_n+1 = g(p\_n)$ και με τυχαίο p_0 Ε$[1.7,2.7]$ συγκλίνει στον αριθμό p

Με αρχικη τιμη $p = 1,7$:

2) g(x) = (15*x^2 – 62*x + 72)^1/3

```
def g(x):
    return (15*x**2-62*x+72)**(1/3)
solution_list=[]
def repetitive(x):
    temp=x
    x=g(x)
    return temp,x
x=1.7
for i in range(10):
    temp,x=repetitive(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(x,i)
        break
solution_list
```

```
[[0, 2.150837964328351],
 [1, 2.003295022065583],
 [2, 1.9994642576216186],
 [3, 2.0000896451530794],
 [4, 1.9999850690750038],
 [5, 2.000002488763067],
 [6, 1.9999995852138122],
 [7, 2.000000069131244],
 [8, 1.9999999884781325],
 [9, 2.0000000019203111]]
```

Η ακολουθία συγκλίνει αργα στην ρίζα

Δεν υπάρχει απόλυτο και σχετικό σφάλμα

3) g(x) = ((x^3 + 62*x – 72)/15)^½

Σύμφωνα με το θεώρημα
διαδοχικών επαναλήψεων η συνάρτηση
g(x) έχει μοναδικό σταθερό σημείο p στο διάστημα
[3.7,4.7] , άρα η ακολουθία p_n με p_n+1 = g(p_n) και
με τυχαίο p_0 Ε[3.7,4.7] συγκλίνει στον αριθμό p

Με αρχικη τιμη p = 3.7 :

3) g(x) = ((x^3 + 62*x – 72)/15)^½

```
def g(x):
    return ((x**3+62*x-72)/15)**(1/2)
solution_list=[]
def repetitive(x):
    temp=x
    x=g(x)
    return temp,x
x=3.7
for i in range(10):
    temp,x=repetitive(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(x,i)
        break
solution_list
```

```
[[0, 3.724271740891097],
 [1, 3.74665399021363],
 [2, 3.7672791996387627],
 [3, 3.7862735919128876],
 [4, 3.8037567398279153],
 [5, 3.8198413647774117],
 [6, 3.834633301014337],
 [7, 3.848231582700703],
 [8, 3.860728619884842],
 [9, 3.872210436907464]]
```

Η ακολουθία συγκλίνει αργά στην ρίζα

Η ρίζα x της g(x) είναι 4 ενώ η θεωρητική ρίζα x* με προσέγγιση 3 δεκαδικών ψηφίων είναι 3.872 οπότε προκύπτει :

Απόλυτο σφάλμα : |x* - x| = |3.872 - 4 | = 0.128

Σχετικό σφάλμα : 0.032

Άσκηση 1

Μέρος II(iii)

f(x) = x^3 – 15*x^2 + 62*x – 72

f'(x) = 3*x^2 – 30*x + 62

Είναι προφανές ότι μια ρίζα της συνάρτησης βρίσκεται στα διαστήματα

[3.7,4.7] , [8.7,9.7] , [1.7,2.7]

Είναι f(3.7) = 2.703 και f(4.7) = -8.127 , f(8.7) = -9.447 και f(9.7) = 30.723 , f(1.7) = -5.037 και f(2.7) = 5.783

και η συνάρτηση f είναι συνεχής στο

Κλειστό διάστημα [3.7,4.7] , [8.7,9.7] , [1.7,2.7] συνεπώς έχει μία τουλάχιστον

Πραγματική Ρίζα στο ανοικτό διάστημα (3.7,4.7) , (8.7,9.7) , (1.7,2.7)

Εφαρμόζοντας την μέθοδο του Newton έχω:

Διάστημα [3.7,4.7]

Μέθοδος Newton-Raphson

```
def f(x):
    return x**3-15*x**2+62*x-72,3*x**2-30*x+62
def newton_raphson(x):
    temp=x
    x=x-f(x)[0]/f(x)[1]
    return x,temp
solution_list=[]
x=3.7
for i in range(20):
    x,temp=newton_raphson(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(i,x)
        break
solution_list
```

```
4 4.000000000000003
```

```
[[0, 4.040857503152586],
 [1, 4.0004757355218645],
 [2, 4.000000067856391],
 [3, 4.000000000000003],
 [4, 4.000000000000003]]
```

Η ζητούμενη ρίζα x* δεν εχει απόλυτο και σχετικό σφάλμα

Διάστημα [8.7,9.7]

```
def f(x):
    return x**3-15*x**2+62*x-72,3*x**2-30*x+62
def newton_raphson(x):
    temp=x
    x=x-f(x)[0]/f(x)[1]
    return x,temp
solution_list=[]
x=8.7
for i in range(20):
    x,temp=newton_raphson(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(i,x)
        break
solution_list
```

```
4 9.0
```

```
[[0, 9.0365514784467366],
 [1, 9.000449532726945],
 [2, 9.0000000692683],
 [3, 9.0],
 [4, 9.0]]
```

Η ζητούμενη ρίζα x* δεν εχει απόλυτο και σχετικό σφάλμα

Διάστημα [1.7,2.7]

```
def f(x):
    return x**3-15*x**2+62*x-72,3*x**2-30*x+62
def newton_raphson(x):
    temp=x
    x=x-f(x)[0]/f(x)[1]
    return x,temp
solution_list=[]
x=1.7
for i in range(20):
    x,temp=newton_raphson(x)
    solution_list.append([i,x])
    if abs(x-temp)<10**(-9):
        print(i,x)
        break
solution_list
```

```
4 1.9999999999999991
```

```
[[0, 1.956075241484494],
 [1, 1.9988149877589596],
 [2, 1.9999990984014913],
 [3, 1.9999999999994773],
 [4, 1.9999999999999991]]
```

Η ζητούμενη ρίζα x* δεν εχει απόλυτο και σχετικό σφάλμα

Άσκηση 2

Μέρος I(i) - Διαιρεμένες διαφορές

A = (0.3, 1.1), B = (0.7, 1.8), C = (1.2, 1.9), D = (2, 2.2), E = (5, 6.5)

- Διαιρεμένες διαφορές(για τα σημεία A , B)

Έστω x0 = 0.3 , f(x0) = 1.1 και x1 = 0.7 , f(x1) = 1.8

Τότε διαδοχικά έχουμε :

f[x0] = f(0.3) = 1.1 και f[x1] = f(0.7) = 1.8

f[x0,x1] = (f(x1) - f(x0))/(x1 – x0) = (1.8 - 1.1)/(0.7 - 0.3) = 0.7/0.4

Επομένως το πολυώνυμο παρεμβολής είναι:

P(x) = f[x0] + f[x0,x1](x-x0) = 1.1 + 1.75(x – 0.3) = 1.75x + 0.575

Άσκηση 2

Μέρος Ι(i) - Προσέγγιση ευθείας ελαχίστων τετραγώνων

- Προσέγγιση ευθείας ελαχίστων τετραγώνων

Για τα σημεία Α,Β έχω:

x0 = 0.3 , y0 = 1.1

X1 = 0.7 , y1 = 1.8

i	Xi	Yi	Xi * Yi	(Xi)^2
1	0.3	1.1	0.33	0.09
2	0.7	1.8	1.26	0.49
	1.0	2.9	1.59	0.58

Προσέγγιση ευθείας ελαχίστων τετραγώνων

Για n = 2

A = (2*1.59 - 1*2.9)/(2*0.58 - 1.0) = (3.18 - 2.9) / (1.16 - 1.0) = 0.28/0.16 = 1.75

B = (0.58*2.9 - 1.59*1) / (2*0.58 - 1^2) = (1.682 - 1.59) / (1.16 - 1) = 0.092 / 0.16 = 0.575

Άρα το ζητούμενο πολυώνυμο είναι
P(x) = 1.75*x + 0.575

Σφάλμα της προσέγγισης

$E = e_1^2 + e_2^2 = [y_1 - (a*x_1 + b)]^2 + [y_2 -(a*x_2 + b)]^2$
$= [1.1 - (1.75 * 0.3 + 0.575)]^2 + [1.8 - (1.75 * 0.7 + 0.575)] ^ 2$

$= 0 + 0 = 0$

Άσκηση 2

Μέρος I(ii) - Παρεμβολή Lagrange

Πολυωνυμική Παρεμβολή 2ης τάξης

A = (0.3, 1.1), B = (0.7, 1.8), C = (1.2, 1.9)

(x0, f(x0)) = (0.3 ,1.1)

(x1, f(x1)) = (0.7 ,1.8)

(x2, f(x2)) = (1.2 ,1.9)

P(x) = L0(x)f(x0) + L1(x)f(x1) + L2(x)f (x2)

Άσκηση 2

Μέρος I(ii) - Συνέχεια

$$L(x0,x) = (x - x1) * (x - x2) / (x0 - x1) * (x0 - x2)$$

$$= (x - 0.7) * (x - 1.2) / (0.3 - 0.7) * (0.3 - 1.2)$$

$$= (x^2 - 1,2*x - 0,7*x + 0,84) / (0.4 * 0.9)$$

$$= (x^2 - 1.9*x + 0.84) / (0.36)$$

$$L(x1,x) = (x - x0) * (x - x2) / (x1 - x0) * (x1 - x2)$$

$$= (x - 0.3) * (x - 1.2) / (0.7 - 0.4) * (0.7 - 1.2)$$

$$= (x^2 - 1.2*x - 0.3*x + 0.36) / (0.4 * (-0.5))$$

$$= (x^2 - 1.5*x + 0.36) / (- 0.2)$$

Άσκηση 2
Μέρος I(ii) - Συνέχεια

L(x2,x) = (x – x0) * (x – x1) / (x2 – x0) * (x2 – x1)

= (x – 0.3) * (x – 0.7) / (1.2 - 0.3) * (1.2 - 0.7)

= (x^2 – 0.7*x – 0.3*x + 0.21) / (0.9 * 0.5)

= (x^2 – 1.0*x + 0.21) / (0.45)

P(x) = L0(x)f(x0) + L1(x)f(x1) + L2(x)f (x2)

= ((x^2 – 1.9*x + 0.84) / (0.36)) * 1.1 + ((x^2 – 1.5*x + 0.36) /-(0.2)) * 1.8 +

((x^2 – 1.0*x + 0.21) / (0.45)) * 1.9

= (x^2 – 1.9*x + 0.84) * 3.05555 - (x^2 – 1.5*x + 0.36) * 9 + (x^2 – 1.0*x
+ 0.21) * 4.22222

Άσκηση 2

Μέρος I(ii) - Συνέχεια

1) $3.05555*x^2 – 5.80554*x + 2.56666$

2) $-9*x^2 + 13.5*x – 3.24$

3) $4.22222*x^2 – 4.22222*x + 0.88666$

1) + 2) + 3) = $-1.7223*x^2 + 3.47224*x – 0.46002$

Άσκηση 2 Μέρος I(ii) - Συνέχεια Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

Επειδή ο αριθμός των σημείων είναι 3 , σύμφωνα με την συνθήκη $m < n - 1$

Ο μεγαλύτερος δυνατός βαθμός m του πολυωνύμου είναι : $m < 3 - 1$ δηλαδή

$m=1$

Έστω $P(x) = a0 + a1*x$ το ζητούμενο πολυώνυμο . Τότε το σύστημα γράφεται ως εξής :

A0Sum(n=1 to 3) of xi_0 + a1*Sum(n=1 to 3) of xi_1 = Sum(n=1 to 3) of yi*xi_0

A0Sum(n=1 to 3) of xi_1 + a1*Sum(n=1 to 3) of xi_2 = Sum(n=1 to 3) of yi*xi_1

Άσκηση 2 Μέρος Ι(ii) - Συνέχεια Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

i	xi	xi^2	yi	xi*yi
0	0.3	0.09	1.1	0.33
1	0.7	0.49	1.8	1.26
2	1.2	1.44	1.9	2.28
	2.2	2.02	**4.8**	3.87

Άσκηση 2 Μέρος I(ii) - Συνέχεια
Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

Σύμφωνα με τον πίνακα της προηγούμενης διαφάνειας έχουμε:

$4a0 + 2.2a1 = 4.8$ (1)

$2.2a0 + 2.02a1 = 3.87$ (2)

Από (1)

$a0 + 0.55\alpha1 = 1.2 \Rightarrow \alpha0 = -0.55\alpha1 + 1.2$ (3)

Άρα έχω για την (2)

$2.2(-0.55\alpha1 + 1.2) + 2.02a1 = 3.87 \Rightarrow -1.21a1 + 2.64 + 2.02a1 = 3.87 \Rightarrow$

$0.81a1 = 1.23 \Rightarrow a1 = 1.48192$ (4)

Άσκηση 2 Μέρος I(ii) - Συνέχεια Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξ ης

Από (4) + (3)

A0 = -0.55*1.48192 +1.2 = -0.81506 + 1.2 = 0.38494

Το ζητούμενο πολυώνυμο που προκύπτει είναι :

P(x) = 0.38494 + 1.48192*x

Άσκηση 2

Μέρος Ι(iii) - Διαιρεμένες διαφορές

A = (0.3, 1.1), B = (0.7, 1.8), C = (1.2, 1.9), D = (2, 2.2), E = (5, 6.5)

Διαιρεμένες διαφορές(για τα σημεία A , B, C , D)

Έστω $x0 = 0.3$, $f(x0) = 1.1$ και $x1 = 0.7$, $f(x1) = 1.8$

$x2 = 1.2$, $f(x2) = 1.9$ και $x3 = 2$, $f(x3) = 2.2$

Άσκηση 2
Μέρος I(iii) - Συνέχεια

Πινακάκι διαιρεμένης διαφοράς 3ής τάξης

x	y	1^{st} order	2^{nd} order	3^{rd} order
0.3	1.1			
		1.75		
0.7	1.8		-1.7222	
		0.2		1.0923
1.2	1.9		0.1346	
		0.375		
2	2.2			

Άσκηση 2
Μέρος I(iii) - Συνέχεια

$$f(x) = y_0 + \left(x - x_0\right)f\left[x_0, x_1\right] + \left(x - x_0\right)\left(x - x_1\right)f\left[x_0, x_1, x_2\right] + \left(x - x_0\right)\left(x - x_1\right)\left(x - x_2\right)f\left[x_0, x_1, x_2, x_3\right]$$

$$f(x) = 1.1 + (x - 0.3) \times 1.75 + (x - 0.3)(x - 0.7) \times \ - 1.7222 + (x - 0.3)(x - 0.7)(x - 1.2) \times 1.0923$$

$$f(x) = 1.1 + (x - 0.3) \times 1.75 + \left(x^2 - x + 0.21\right) \times \ - 1.7222 + \left(x^3 - 2.2x^2 + 1.41x - 0.252\right) \times 1.0923$$

$$f(x) = 1.1 + (1.75x - 0.525) + \left(- 1.7222x^2 + 1.7222x - 0.3617\right) + \left(1.0923x^3 - 2.403x^2 + 1.5401x - 0.2752\right)$$

$$f(x) = 1.0923x^3 - 4.1252x^2 + 5.0123x - 0.0619$$

Άσκηση 2

Μέρος I(iii) - προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

```
def least_squares(x,y):
    n=len(y)
    Sxi=sum([xi for xi in x])
    Sxi2=sum([xi**2 for xi in x])
    Syi=sum([yi for yi in y])
    #Sxiyi=sum([xi*yi for xi,yi in zip(x,y)])
    Sxiyi=0
    for i in range(n):
        Sxiyi+=x[i]*y[i]
    a=(n*Sxiyi-Sxi*Syi)/(n*Sxi2-(Sxi)**2)
    b=(Sxi2*Syi-Sxiyi*Sxi)/(n*Sxi2-(Sxi)**2)
    return a,b
```

```
least_squares([0.3, 0.7 , 1.2 , 2 , 5] , [1.1, 1.8, 1.9, 2.2, 6.5])
```

```
(1.130428611978427, 0.6200113539596939)
```

Άρα το πολυώνυμο που προκύπτει είναι P(x) = 1.13042*x + 0.62001

Άσκηση 2
Μέρος II(i)

- AM : 2022201800131
- A = 1 + (31 mod 3) = 1 + 1 = 2
- B = 1 + (800 mod 7) = 1 + 2 = 3
- C = 1 + (2201 mod 11) = 1 + 1 =2

A = (a, 1.2*b), B = (a*2 , 3*c), C = (4 + a, b + 2c)

A = (2 , 3.6) , B = (4 , 6) , C = (6 , 7)

Άσκηση 2
Μέρος II(i) - Διαιρεμένες διαφορές

Διαιρεμένες διαφορές(για τα σημεία Α , Β)

- Έστω $x0 = 2$, $f(x0) = 3.6$ και $x1 = 4$, $f(x1) = 6$

- Τότε διαδοχικά έχουμε :
- $f[x0] = f(2) = 3.6$ και $f[x1] = f(4) = 6$

- $f[x0,x1] = (f(x1) - f(x0))/(x1 – x0) = (6-3.6)/ (4 – 2) = 2.4 / 2 = 1.2$

- Επομένως το πολυώνυμο παρεμβολής είναι:
- $P(x) = f[x0] + f[x0,x1](x-x0) = 3.6 + 1.2(x – 2) = 1.2x + 1.2$

Άσκηση 2 Μέρος ΙΙ(i) - Προσέγγιση ευθείας ελαχίστων τετραγώνων

- Για τα σημεία Α,Β έχω:

x0 = 2 , y0 = 3.6

X1 = 4 , y1 = 6

i	Xi	Yi	Xi * Yi	(Xi)^2
1	2	3.6	7.2	4
2	4	6	24	16
	6	9.6	31.2	20

Άσκηση 2 Μέρος II(i) - Προσέγγιση ευθείας ελαχίστων τετραγώνων

Για n = 2

A = (2*31.2 - 6*9.6)/(2*20 - (6)^2) = (62.4 - 57.6) / (40 - 36) = 4.8/ 4 = 1.2

B = (20*9.6 - 31.2*6) / (2*20 - 6^2) = (192 – 187.2) / (40 - 36) = 4.8 / 4 = 1.2

Άρα το ζητούμενο πολυώνυμο είναι

P(x) = 1.2*x + 1.2

Σφάλμα της προσέγγισης

- E = e_1^2 + e_2^2

  = [y_1 - (a*x_1 + b)]^2 + [y_2 -(a*x_2 + b)]^2

  = [3.6 - (1.2 * 2 + 1.2)]^2 + [6 - (1.2 * 4 + 1.2)] ^ 2

  = 0 + 0 = 0

Άσκηση 2
Μέρος II(ii) - Παρεμβολή Lagrange

A = (2 , 3.6) , B = (4 , 6) , C = (6 , 7)

P(x) = L0(x)f(x0) + L1(x)f(x1) + L2(x)f (x2)

- L(x0,x) =(x – x1) * (x – x2) / (x0 – x1) * (x0 – x2)
  =(x – 4)* (x – 6) / (2 - 4) * (2 - 6)
  =(x^2 – 6*x – 4*x + 24) / ((-2) * (-4))
  =(x^2 – 10*x + 24) / 8

- L(x1,x) =(x – x0) * (x – x2) / (x1 – x0) * (x1 – x2)
  =(x – 2) * (x – 6) / (4 - 2) * (4 - 6)
  =(x^2 – 6*x – 2*x + 12) / (2 * (-2))
  =(x^2 – 8*x + 12) / (- 4)

Άσκηση 2

Μέρος ΙΙ(ii) - Συνέχεια

- L(x2,x) = (x – x0) * (x – x1) / (x2 – x0) * (x2 – x1)

  = (x – 2) * (x – 4) / (6 - 2) * (6 - 4)

  = (x^2 – 4*x – 2*x + 8) / (4 * 2)

  = (x^2 – 6*x + 8) / (8)

- P(x) = L0(x)f(x0) + L1(x)f(x1) + L2(x)f (x2)

  = ((x^2 – 10*x + 24) / 8)) * 3.6 + ((x^2 – 8*x + 12) / (- 4)) * 6 + ((x^2 – 6*x +8) / (8) * 7

  = 8*((x^2 – 10*x + 24) / 8)) * 3.6 + 8*((x^2 – 8*x + 12) / (- 4)) * 6 + 8*((x^2 – 6*x +8) / (8) * 7

Άσκηση 2
Μέρος ΙΙ(ii) - Συνέχεια

- $P(x) = (x^2 – 10*x + 24) * 3.6 + (x^2 – 8*x + 12) * 3 + (x^2 – 6*x +8) * 7$

  $= 3.6*x^2 - 36$

  $= 13.6*x^2 – 102*x + 178.4$

Άσκηση 2 Μέρος II(ii) - Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

Επειδή ο αριθμός των σημείων είναι 3 , σύμφωνα με την συνθήκη $m < n - 1$

Ο μεγαλύτερος δυνατός βαθμός m του πολυωνύμου είναι : $m < 3 - 1$ δηλαδή

$m=1$

Άσκηση 2 Μέρος ΙΙ(ii) - Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

- A = (2 , 3.6) , B = (4 , 6) , C = (6 , 7)

i	Xi	Yi	Xi * Yi	(Xi)^2
1	2	3.6	7.2	4
2	4	6	24	16
	6	7	42	36
	12	16.6	73.2	**56**

Άσκηση 2 Μέρος ΙΙ(ii) - Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

A = (3 * 73.2 - 12 * 16.6) / (3 * 56 – 12^2) = (219.6 - 199.2) / (168 – 144)
= 20.4 / 24 = 0.85

B = (56 * 16.6 - 73.2 * 12) / (3 * 56 – 12^2) = (929.6 - 878.4) / (168 – 144)
= 51.2 / 24 = 2.13

Άρα το ζητούμενο πολυώνυμο είναι
P(x) = 0.85*x + 2.13

Άσκηση 2 Μέρος ΙΙ(ii) - Προσέγγιση ελαχίστων τετραγώνων δεύτερης τάξης

Σφάλμα της προσέγγισης

E = e_0^2 + e_1^2 + e_2^2= [y_0 - (a*x_0 + b)]^2 + [y_1 - (a*x_1 + b)]^2 + [y_2 -(a*x_2 + b)]^2 = [3.6 - (0.85 * 2 + 2.13)]^2 + [6 - (0.85 * 4 + 2.13)] ^ 2 + [7 - (0.85 * 6 + 2.13)] ^ 2

= 0.0529 + 0.0529 + 0.2209 = 0.3267

Άσκηση 3

Μέρος I(i) - Πολυώνυμο Taylor

Το πολυώνυμο Taylor τρίτης τάξης της συνάρτησης γύρω από το x = 3 είναι :

- $f(x) = x^2 e^{-x^2}$

  $f(3) = \frac{9}{e^9}$

- $f'(x) = 2x(1-x^2)e^{-x^2}$

  $f'(3) = -\frac{48}{e^9}$

- $f''(x) = 2(2x^4 - 5x^2 + 1)e^{-x^2}$

  $f''(3) = \frac{236}{e^9}$

- $f'''(x) = (-8x^5 + 36x^3 - 24x)e^{-x^2}$

  $f'''(3) = -\frac{1044}{e^9}$

Άσκηση 3

Μέρος I(i) - Πολυώνυμο Taylor

$$f(x) = \frac{\frac{9}{e^9}}{0!}(x-(3))^0 + \frac{-\frac{48}{e^9}}{1!}(x-(3))^1 +$$

$$\frac{\frac{236}{e^9}}{2!}(x-(3))^2 + \frac{\frac{-1044}{e^9}}{3!}(x-(3))^3$$

$$= \frac{9}{e^9} - \frac{48}{e^9}(x-3) + \frac{118}{e^9}$$

Άσκηση 3

Μέρος I(ii) - Σύνθετος κανόνας τραπεζίου

- Για h = 0.1

```
trapez(f,20,0,2)
```

```
0.422541941281371
```

Για h = 0.01

```
trapez(f,200,0,2)
```

```
0.4227232249326587
```

- Για h = 0.05

```
trapez(f,40,0,2)
```

```
0.4226792699465189
```

Για h = 0.001

```
trapez(f,2000,0,2)
```

```
0.4227250381768392
```

h= (b-a)/n

Αλγόριθμος

```
import math
def trapez(f,N,a,b):
    h=(b-a)/N
    S0=f(a)+f(b)
    S1=0
    x0=a
    for i in range(1,N):
        x=x0+i*h
        S1=S1+f(x)
    I=0.5*h*(S0+2*S1)
    return I
def f(x):
    return x**2*math.exp(-x**2)
```

Άσκηση 3
Μέρος Ι(ii) - Σύνθετος κανόνας τραπεζίου

h	Αποτελεσμα	Απολυτο Σφαλμα	Σχετικο Σφαλμα
0.1	0.422541	0.000184	0.00043527115
0.05	0.422679	0.000046	0.00010881778
0.01	0.422723	0.000002	0.0000047312
0.001	**0.422725**	**0**	**0**

Θεωρητικη τιμη : 0.422725

Άσκηση 3
Μέρος I(ii) - Σύνθετος κανόνας Simpson

- Για h = 0.1

```
simpson(f,10,0,2)
```

```
0.4227248840627754
```

- Για h = 0.05

```
simpson(f,20,0,2)
```

```
0.42272504616823503
```

Για h = 0.01

```
simpson(f,100,0,2)
```

```
0.4227250564761865
```

Για h = 0.001

```
simpson(f,1000,0,2)
```

```
0.4227250564924748
```

- h= (b-a)/2n

Αλγόριθμος

```
def simpson(f,N,a,b):
    n=2*N
    h=(b-a)/n
    S0=f(a)+f(b)
    S1=0
    S2=0
    x0=a
    for i in range(1,n):
        x=x0+i*h
        if i%2==0:
            S2+=f(x)
        else:
            S1+=f(x)
    I=(h/3)*(S0+4*S1+2*S2)
    return I
def f(x):
    return x**2*math.exp(-x**2)
```

Άσκηση 3
Μέρος Ι(ii) - Σύνθετος κανόνας Simpson

h	Αποτελεσμα	Απολυτο Σφαλμα	Σχετικο Σφαλμα
0.1	**0.422724**	0.000001	0.0000023656
0.05	**0.422725**	0	0
0.01	**0.422725**	0	0
0.001	**0.422725**	**0**	**0**

Θεωρητικη τιμη : 0.422725

Άσκηση 3

Μέρος I(ii) - Σύνθετος κανόνας Simpson 3/8

- Για h = 0.1

```
simpson_38(f,7,0,2 )
```

```
0.4227247223977927
```

Για h = 0.01

```
simpson_38(f,13,0,2 )
```

```
0.42272503037467324
```

- Για h = 0.05

```
simpson_38(f,67,0,2 )
```

```
0.42272505645652747
```

Για h = 0.001

```
simpson_38(f,667,0,2 )
```

```
0.42272505649247283
```

- h= (b-a)/3n

Αλγόριθμος

```
def simpson_38(f,N,a,b):
    n=3*N
    h=(b-a)/n
    S0=f(a)+f(b)
    S1=0
    S2=0
    S3=0
    x0=a
    for i in range(1,n):
        x=x0+i*h
        if i%3==0:
            S3+=f(x)
        elif i%3==2:
            S2+=f(x)
        elif i%3==1:
            S1+=f(x)
    I=(3*h/8)*(S0+3*S1+3*S2+2*S3)
    return I
def f(x):
    return x**2*math.exp(-x**2)
```

Άσκηση 3
Μέρος I(ii) - Σύνθετος κανόνας Simpson 3/8

h	Αποτελεσμα	Απολυτο Σφαλμα	Σχετικο Σφαλμα
0.1	**0.422724**	0.000001	0.0000023656
0.05	**0.422725**	0	0
0.01	**0.422725**	0	0
0.001	**0.422725**	**0**	**0**

Θεωρητικη τιμη : 0.422725

Άσκηση 3

Μέρος II(i) - Taylor

- $f(x) = 2xe^{-3x^2}$

  $f(3) = \frac{6}{e^{27}}$

- $f'(x) = 2(1-6x^2)e^{-3x^2}$

  $f'(3) = -\frac{106}{e^{27}}$

- $f''(x) = (72x^3 - 36x)e^{-3x^2}$

  $f''(3) = \frac{1836}{e^{27}}$

- $f'''(x) = 36(-12x^4+12x^2-1)e^{-3x^2}$

  $f'''(3) = -\frac{31140}{e^{27}}$

$$f(x) = \frac{\frac{6}{e^{27}}}{0!}(x-3)^0 + \frac{-\frac{106}{e^{27}}}{1!}(x-3)^1 + \frac{\frac{1836}{e^{27}}}{2!}(x-3)^2 + \frac{-\frac{31140}{e^{27}}}{3!}(x-3)^3$$

$$= \frac{6}{e^{27}} - \frac{106}{e^{27}}(x-3) + \frac{918}{e^{27}}(x-3)^2 - \frac{5190}{e^{27}}(x-3)^3$$

Άσκηση 3
Μέρος II(ii) - Σύνθετος κανόνας τραπεζίου

A = 2 , B = 3

- Για h = 0.1

```
trapez(f,20,0,2)
```

```
0.33165935133761276
```

- Για h = 0.05

```
trapez(f,40,0,2)
```

```
0.33291424693132754
```

Για h = 0.01

```
trapez(f,200,0,2)
```

```
0.33331461574101057
```

Για h = 0.001

```
trapez(f,2000,0,2)
```

```
0.33333111857227954
```

Αλγόριθμος

```
import math

def trapez(f,N,a,b):
    h=(b-a)/N
    S0=f(a)+f(b)
    S1=0
    x0=a
    for i in range(1,N):
        x=x0+i*h
        S1=S1+f(x)
    I=0.5*h*(S0+2*S1)
    return I
def f(x):
    return 2*x*math.exp(-3*x**2)
```

Άσκηση 3
Μέρος ΙΙ(ii) - Σύνθετος κανόνας τραπεζίου

h	Αποτελεσμα	Απολυτο Σφαλμα	Σχετικο Σφαλμα
0.1	0.422541	0.000184	0.00043527115
0.05	0.422679	0.000046	0.00010881778
0.01	0.422723	0.000002	0.0000047312
0.001	**0.422725**	**0**	**0**

Θεωρητικη τιμη : 0.333331

Άσκηση 3

Μέρος ΙΙ(ii) - Σύνθετος κανόνας Simpson

- A = 2 , B = 3
- Για h = 0.1

```
simpson(f,10,0,2)
```

```
0.3333520168218347
```

Για h = 0.01

```
simpson(f,100,0,2)
```

```
0.33333128726148353
```

- Για h = 0.05

```
simpson(f,20,0,2)
```

```
0.33333254546256585
```

Για h = 0.001

```
simpson(f,1000,0,2)
```

```
0.3333312852627488
```

Αλγόριθμος

```
def simpson(f,N,a,b):
    n=2*N
    h=(b-a)/n
    S0=f(a)+f(b)
    S1=0
    S2=0
    x0=a
    for i in range(1,n):
        x=x0+i*h
        if i%2==0:
            S2+=f(x)
        else:
            S1+=f(x)
    I=(h/3)*(S0+4*S1+2*S2)
    return I
def f(x):
    return 2*x*math.exp(-3*x**2)
```

Άσκηση 3
Μέρος II(ii) - Σύνθετος κανόνας Simpson

h	Αποτελεσμα	Απολυτο Σφαλμα	Σχετικο Σφαλμα
0.1	**0.333352**	0.000021	0.00006300044
0.05	**0.333332**	0.000001	0.00000300002
0.01	**0.333331**	0	0
0.001	**0.333331**	**0**	**0**

Θεωρητικη τιμη : 0.333331

Άσκηση 3

Μέρος II(ii) - Σύνθετος κανόνας Simpson 3/8

- A = 2 , B = 3
- Για h = 0.1

```
simpson_38(f,7,0,2)
```

```
0.333370912750731
```

Για h = 0.01

```
simpson_38(f,67,0,2)
```

```
0.3333312896728633
```

- Για h = 0.05

```
simpson_38(f,13,0,2)
```

```
0.3333344548626922
```

Για h = 0.001

```
simpson_38(f,667,0,2)
```

```
0.3333312852629976
```

Αλγόριθμος

```
def simpson_38(f,N,a,b):
    n=3*N
    h=(b-a)/n
    S0=f(a)+f(b)
    S1=0
    S2=0
    S3=0
    x0=a
    for i in range(1,n):
        x=x0+i*h
        if i%3==0:
            S3+=f(x)
        elif i%3==2:
            S2+=f(x)
        elif i%3==1:
            S1+=f(x)
    I=(3*h/8)*(S0+3*S1+3*S2+2*S3)
    return I
def f(x):
    return 2*x*math.exp(-3*x**2)
```

Άσκηση 3
Μέρος II(ii) - Σύνθετος κανόνας Simpson 3/8

h	Αποτελεσμα	Απολυτο Σφαλμα	Σχετικο Σφαλμα
0.1	**0.333370**	0.000039	0.00011700081
0.05	**0.333334**	0.000003	0.00000900006
0.01	**0.333331**	0	0
0.001	**0.333331**	**0**	**0**

Θεωρητικη τιμη : 0.333331