ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

Ακαδ. Έτος: 2018-2019

Τετάρτη, 31/10/2018

Μάθημα: Μέθοδοι Επίλυσης με Η/Υ

Διδάσκοντες: Ν.Δ. Λαγαρός (Αν. Καθηγητής), Αθ. Στάμος (ΕΔΙΠ), Χ. Φραγκουδάκης (ΕΔΙΠ)
Αμβ. Σαββίδης (ΥΔ)

Ασκήσεις για την 4η παράδοση – Δομές επανάληψης & Γραφήματα

1. Σύγκριση εμπειρικών σχέσεων βάθους/παροχής σε χείμαρρο

Σε χείμαρρο έγιναν οι εξής μετρήσεις βάθους h και παροχής q:

h_i	q_i
0.00000	0.00000
0.20000	6.81261
0.21000	8.16116
0.30000	8.36719
0.38000	8.63658
0.40000	9.03462
0.40500	9.66808
0.50000	11.20181
0.53000	11.88206
0.55000	11.82824
0.58000	12.02988
1.00000	15.74552

Από ανάλυση των μετρήσεων προέκυψαν τα παρακάτω μοντέλα:

α) $Q(h)=ah+bh^2, a=30.598, b=-15.297$

β) $Q(h)=c\sqrt{h}, c=15.591$

γ) $Q(h)=kh^m, k=15.432, m=0.491$

1.1. Να σχεδιαστούν σε 2D γράφημα οι μετρήσεις (ως σημεία) και τα τρία μοντέλα με διαφορετικά χρώματα. Τα μοντέλα θα σχεδιαστούν για h που κυμαίνεται από 0 έως 1 με βήμα 0.05. Ποιο από τα μοντέλα ταιριάζει καλύτερα στις μετρήσεις; Οι υπολογισμοί να γίνουν με δομές επανάληψης.

1.2. Να επιβεβαιώσετε την απάντησή σας υπολογίζοντας με δομές επανάληψης το μέσο τετραγωνικό σφάλμα για κάθε μοντέλο:

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(q_i-Q(h_i)\right)^2}$$

1.3. Να επαναληφθούν τα ερωτήματα 1.1 και 1.2 χωρίς δομές επανάληψης

Λύση

Για τα 3 διαφορετικά μοντέλα θα συνταχθούν 3 συναρτήσεις modela(), modelb() και modelc().

```
function q=modela(h)
    q = 30.598 .* h - 15.297 .* h.^2;
end
function q=modelb(h)
    q = 15.591 .* sqrt(h);
end
function q=modelc(h)
    q = 15.432 .* h .^ 0.491;
end
```

<u>Λύση με δομές επανάληψης</u>
Για τον υπολογισμό του σφάλματος θα συνταχθεί συνάρτηση rmse() η οποία θα έχει το όρισμα εισόδου modelx το οποίο είναι συνάρτηση, και θα καλεί αυτή τη συνάρτηση.

```
function er=rmse(h, q, modelx) %modelx είναι συνάρτηση
    n = length(h);
    s = 0.0;
    for i=1:n
        s = s + (q(i) – modelx(h(i))).^2;
    end
    er = sqrt(s/n);
end
```

Το script που επιλύει την άσκηση καλεί την rmse() 3 φορές μία με όρισμα την συνάρτηση modela(), μία με την modelb() και μία με την modlec(). Ο κανόνας του Matlab για την κλήση συνάρτησης με όρισμα πχ τη συνάρτηση modela(), είναι πριν από το όνομα της συνάρτησης να μπει ο χαρακτήρας @, δηλαδή @modela.

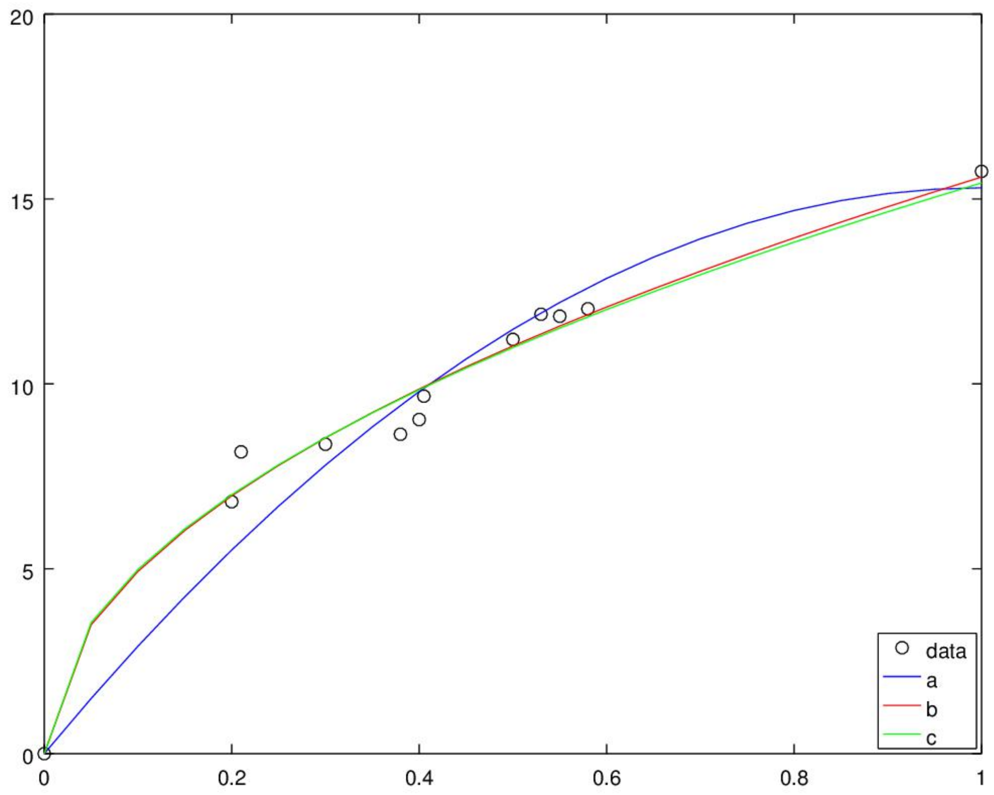
```
clear; clc; close all;
qd=[0.00000    0.00000
    0.20000    6.81261
    0.21000    8.16116
    0.30000    8.36719
    0.38000    8.63658
    0.40000    9.03462
    0.40500    9.66808
    0.50000   11.20181
    0.53000   11.88206
    0.55000   11.82824
    0.58000   12.02988
    1.00000   15.74552];

q = qd(:, 2);
h = qd(:, 1);
clear qd;

for i=1:21
   hh(i) = (i-1)*0.05;
   qqa(i) = modela(hh(i));
   qqb(i) = modelb(hh(i));
   qqc(i) = modelc(hh(i));
end
plot(h, q, 'ko');
hold on
plot(hh, qqa, 'b');
plot(hh, qqb, 'r');
plot(hh, qqc, 'g');
legend('data', 'a', 'b', 'c', 'location', 'southeast');
print('q1.jpg', '-djpeg');

disp('RMSEa');
disp(rmse(h, q, @modela));
disp('RMSEb');
disp(rmse(h, q, @modelb));
disp('RMSEc');
disp(rmse(h, q, @modelc));
```

Η εκτέλεση του προγράμματος δίνει:




```
RMSEa
0.90369
RMSEb
0.51781
RMSEc
 0.52521
```

Το δεύτερο και το τρίτο μοντέλο είναι πρακτικά τα ίδια. Συνεπώς είναι προτιμότερο εκείνο με τις λιγότερες παραμέτρους (το δεύτερο).

Λύση χωρίς δομές επανάληψης

Οι συναρτήσεις των μοντέλων δεν αλλάζουν. Η συνάρτηση rmse() και το script γίνονται:

```
function er=rmse2(h, q, modelx)  %modelx είναι συνάρτηση
    er = sqrt(sum( (q - modelx(h)).^2 ) / length(h));
end


clear; clc; close all;
qd=[0.00000    0.00000
    0.20000    6.81261
    0.21000    8.16116
    0.30000    8.36719
    0.38000    8.63658
    0.40000    9.03462
    0.40500    9.66808
    0.50000   11.20181
    0.53000   11.88206
    0.55000   11.82824
    0.58000   12.02988
    1.00000   15.74552];
```

```
q = qd(:, 2);
h = qd(:, 1);
clear qd;

hh = 0:0.05:1;
qqa = modela(hh);
qqb = modelb(hh);
qqc = modelc(hh);
plot(h, q, 'ko');
hold on
plot(hh, qqa, 'b');
plot(hh, qqb, 'r');
plot(hh, qqc, 'g');
legend('data', 'a', 'b', 'c', 'location', 'southeast');
print('q1.jpg', '-djpeg');

disp('RMSEa');
disp(rmse2(h, q, @modela));
disp('RMSEb');
disp(rmse2(h, q, @modelb));
disp('RMSEc');
disp(rmse2(h, q, @modelc));
```

Τα αποτελέσματα είναι τα ίδια.

2. Υπολογισμός ρίζας συνάρτησης

Η μέθοδος Newton-Raphson χρησιμοποιείται για την εύρεση της ρίζας μίας συνάρτησης. Η μέθοδος εκτελεί επαναλήψεις, όπου σε κάθε επανάληψη εκτιμά την ρίζα της συνάρτησης f(x) με την σχέση:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})}$$

Οι επαναλήψεις ολοκληρώνονται όταν οι τιμές του x συγκλίνουν στην ίδια τιμή, ή απλούστερα μετά από ένα επαρκές πλήθος επαναλήψεων.
Με την βοήθεια της μεθόδου Newton-Raphson να υπολογίσετε την ρίζα της συνάρτησης:

$$f(x) = 2\,sinx - \frac{x^2}{10}$$

Δίνεται η $1^{η}$ παράγωγος f'(x) = 2cosx – x/5. Οι επαναλήψεις να ξεκινήσουν από την τιμή x_0=1.8.
α) Να γίνουν 15 επαναλήψεις.
β) Να γίνουν όσες επαναλήψεις απαιτούνται για να επιτευχθεί ακρίβεια ε=0.0001. Να γίνουν τουλάχιστον 4 επαναλήψεις, και αν οι επαναλήψεις ξεπεράσουν τις 1000 το πρόγραμμα να σταματάει με μήνυμα λάθους.
γ) Να βρεθεί γραφικά η ρίζα της συνάρτηση με 2 δεκαδικά ακρίβεια.

Λύση

Οι συναρτήσεις f(x) και f'(x) θα είναι επίσης και συναρτήσεις του Matlab:

```
function y=f(x)
y = 2 .* sin(x) - x.^2./10;
end
function y=ft(x)
y = 2 .* cos(x) - x./5;
end
```

α) Με 15 επαναλήψεις

Η συνάρτηση newton υπολογίζει τον τύπο του θέματος για n φορές με αρχική τιμή xa. Ο τελευταίος υπολογισμός δίνει τη ρίζα.

```
function riza=newton(xa, n)
x(1) = xa;
for i=2:n+1
```

```
    x(i) = x(i-1) - f(x(i-1))/ft(x(i-1));
end
riza = x(end);
end
```

β) Με δεδομένη ακρίβεια

Οι επαναλήψεις θα συνεχίζονται μέχρι να ισχύσει η συνθήκη:

$$|x_i - x_{i-1}| < \varepsilon$$

Για να μην γίνεται σπατάλη μνήμης, θα κρατείται μόνο η τρέχουσα και η προηγούμενη διαδοχική προσέγγιση σε μεταβλητές:

```
function riza=newton2(xa, eps)
found = 0;
for i=1:1000
    riza = xa - f(xa)/ft(xa);
    if i >= 4
        if abs(riza-xa) < eps
            return;
        end
    end
    xa = riza;
end
error('Η μέθοδος δεν συγκλίνει');
end
```

Τέλος με ένα script ελέγχουμε τις συναρτήσεις.

```
clear; clc; close all;
riza = newton(1.8, 15)
riza2 = newton2(1.8, 0.0001)
```

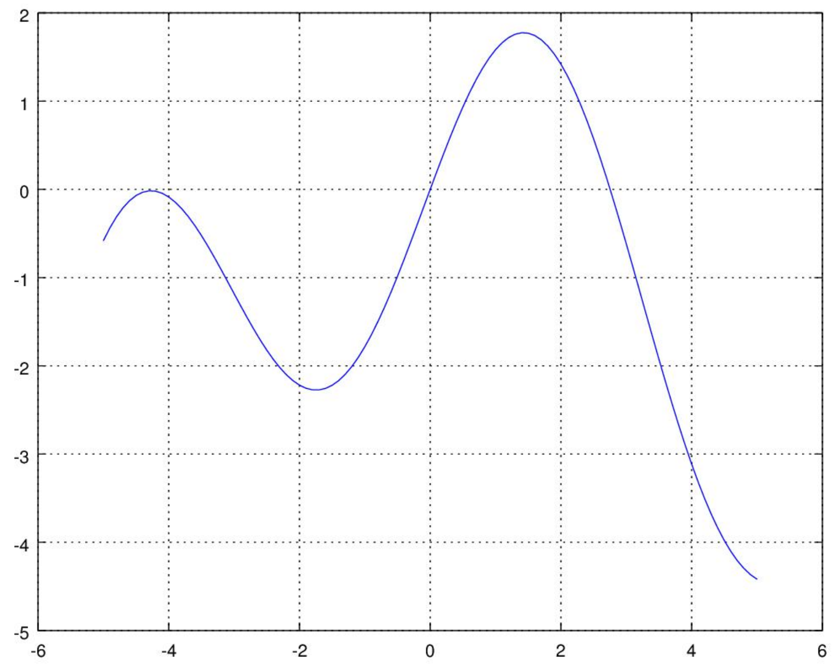
Το πρόγραμμα δίνει αποτελέσματα:

```
riza =  2.7529
riza2 =  2.7529
```

γ) Γραφική επίλυση

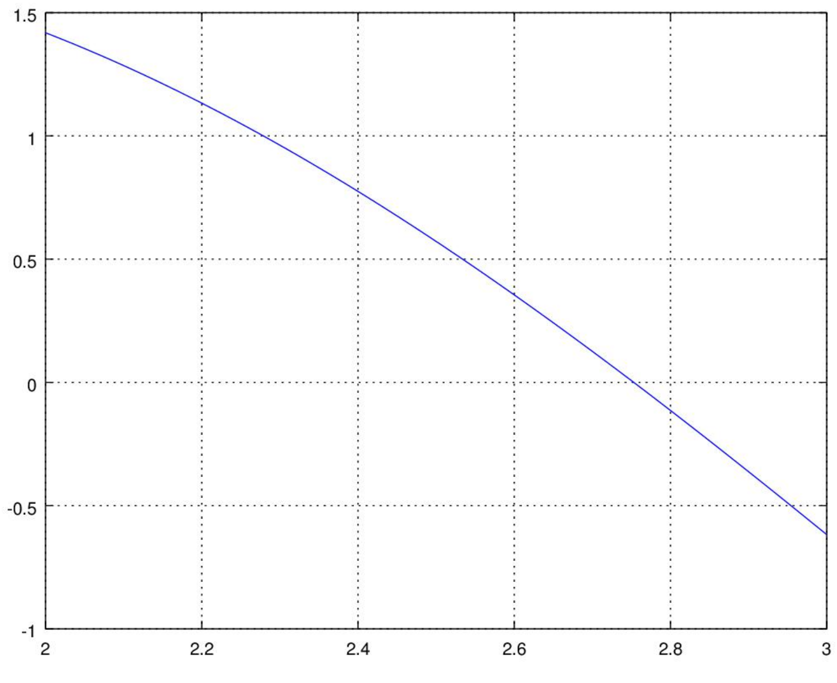
Κάνουμε το διάγραμμα της συνάρτησης για x που κυμαίνεται από -5 έως 5:

```
clear; clc; close all;
x = -5:0.1:5;
plot(x, f(x));
grid on;
print('ga.jpg', '-djpeg');
```



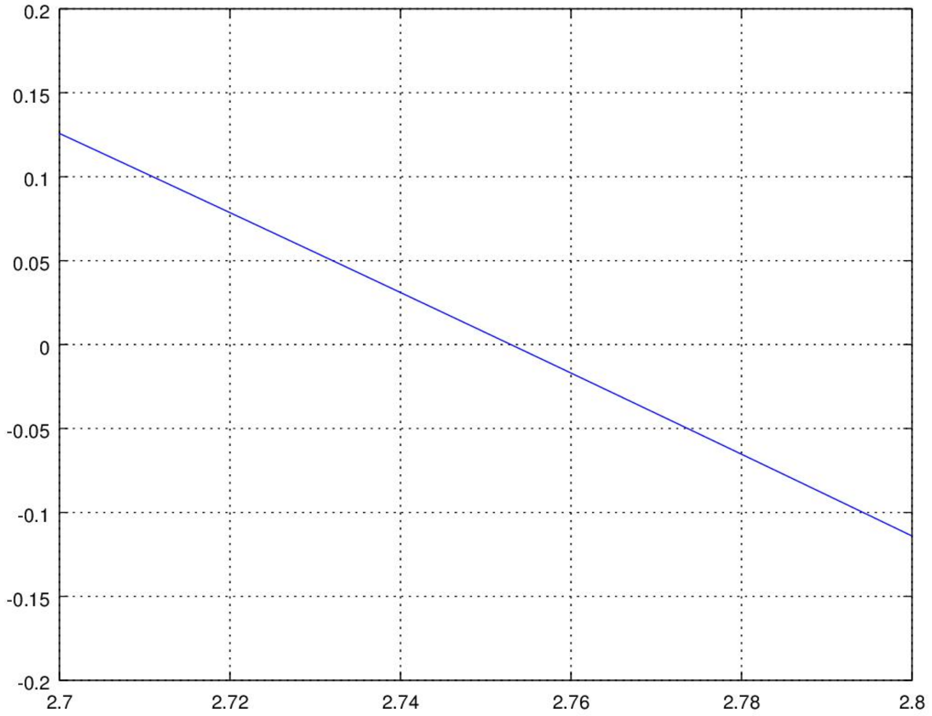
Παρατηρούμε ότι το γράφημα της συνάρτησης τέμνει τον άξονα x στο 0 (όπως εύκολα επαληθεύεται με αντικατάσταση στη συνάρτηση), και επίσης μεταξύ 2 και 3. Κάνουμε λεπτομερέστερο γράφημα μεταξύ 2 και 3:

```
clear; clc; close all;
x = 2:0.01:3;
plot(x, f(x));
grid on;
print('gb.jpg', '-djpeg');
```



Παρατηρούμε ότι το γράφημα της συνάρτησης τέμνει τον άξονα x για x μεταξύ 2.7 και 2.8. Κάνουμε λεπτομερέστερο γράφημα:

```
clear; clc; close all;
x = 2.7:0.005:2.8;
plot(x, f(x));
grid on;
print('gc.jpg', '-djpeg');
```



Η ρίζα είναι μεταξύ 2.75 και 2.76, και πιο κοντά στο 2.75. Συνεπώς με 2 δεκαδικά η ρίζα είναι 2.75.