ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

Ακαδ. Έτος: 2018-2019

Τετάρτη, 7/11/2018

Μάθημα: Μέθοδοι Επίλυσης με Η/Υ

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

Παραδείγματα για την 5η παράδοση – Βασικές εντολές - συμβολοσειρές

1. Υπολογισμός αριθμός ράβδων σε υποστύλωμα

Να συνταχθεί script ή συνάρτηση η οποία να διαβάζει από το πληκτρολόγιο τον οπλισμό As υποστυλώματος (cm^2) και να υπολογίζει την οικονομικότερη διάμετρο και το πλήθος των ράβδων που απαιτούνται. Οι διαθέσιμες διάμετροι ράβδων είναι (mm): 14, 16, 18, 20, 22, 25, 30
Για διευκόλυνση το πλήθος n των ράβδων διαμέτρου Φ (mm) για εμβαδό οπλισμού As (cm^2) δίνεται:

$$A_\Phi = \pi \frac{(\Phi/10)^2}{4} \quad , \quad n = [\frac{A_s}{A_\Phi}] \quad , \quad A_{act} = n\, A_\Phi$$

όπου οι αγκύλες συμβολίζουν στρογγύλευση σε ακέραιο προς τα πάνω και A_{act} είναι το πραγματικό εμβαδό οπλισμού που αντιστοιχεί στις ράβδους που τοποθετούνται.
Οι υπολογισμοί να γίνουν α) με δομή επανάληψης β) χωρίς δομή επανάληψης.

Ανάλυση
Από τους τύπους παρατηρούμε ότι ισχύει:

$$A_{act} \geq A_s$$

Συνεπώς, η διάμετρος ράβδου που θα δώσει το ελάχιστο A_{act} θα είναι η οικονομικότερη. Έτσι θα ελέγξουμε όλες τις διαμέτρους και θα προσδιορίσουμε ποια δίνει το ελάχιστο A_{act} .

α) Με δομή επανάληψης
Στο πρώτο ερώτημα έστω Amin το ελάχιστο A_{act} . Αρχικά στη μεταβλητή Amin θα θέσουμε μία τεράστια τιμή, έτσι ώστε η πρώτη διάμετρος να δίνει μικρότερο A_{act}. Στη συνέχεια θα υπολογίζουμε το A_{act} για κάθε ράβδο και αν είναι μικρότερο από το Amin θα το θέτουμε στο Amin.

Κώδικας
```
clear; clc;close all;
As = input('Οπλισμός σε cm2: ');
Amin = 1e100;
for d=[14:2:22 25 30]     % mm
    A = pi*(d/10)^2 / 4;
    n = ceil(As/A);
    A1 = n*A;
    if A1 < Amin
        Amin = A1;
        dmin = d;
    end
end
d = dmin;
A = pi*(d/10)^2 / 4;
n = ceil(As/A);
A1 = n*A;
t = ['As=' num2str(As) ' cm2    ' num2str(n) 'Φ' num2str(d) ...
    ' -> '  num2str(A1) ' cm2'];
disp(t);
```

Αποτελέσματα (για As=5.5 cm^2)

```
Οπλισμός σε cm2: 5.5
As=5.5 cm2  3Φ16 -> 6.0319 cm2
```

β) Χωρίς δομή επανάληψης

Για το δεύτερο ερώτημα θα κάνουμε όλους τους υπολογισμούς μητρωικά. Σημειώνεται ότι ο τελεστής πολλαπλασιασμού και διαίρεσης πρέπει να περιέχουν και την τελεία (.* και ./ αντίστοιχα).

Κώδικας

```
clear; clc;close all;
As = input('Οπλισμός σε cm2: ');
d = [14:2:22 25 30];     % mm
A = pi*(d/10).^2 ./ 4;
n = ceil(As./A);
A1 = n.*A;
[a1min, i] = min(A1);
t = ['As=' num2str(As) ' cm2   ' num2str(n(i)) 'Φ' num2str(d(i)) ...
    ' -> '  num2str(A1(i)) ' cm2'];
disp(t);
```

Αποτελέσματα (για As=5.5 cm^2)

```
Οπλισμός σε cm2: 5.5
As=5.5 cm2  3Φ16 -> 6.0319 cm2
```

Περαιτέρω ανάλυση

Συνήθως στα υποστυλώματα τοποθετείται συμμετρικός οπλισμός που σημαίνει ότι το πλήθος των ράβδων είναι πολλαπλάσιο του 4. Με αυτό τον περιορισμό ο κώδικας γίνεται:

Κώδικας

```
clear; clc;close all;
As = input('Οπλισμός σε cm2: ');
d = [14:2:22 25 30];     % mm
A = pi*(d/10).^2 ./ 4;
n = ceil(As./A);
n = ceil(n./4).*4;
A1 = n.*A;
[a1min, i] = min(A1);
t = ['As=' num2str(As) ' cm2   ' num2str(n(i)) 'Φ' num2str(d(i)) ...
    ' -> '  num2str(A1(i)) ' cm2'];
disp(t);
```

Αποτελέσματα (για As=5.5 cm^2)

```
Οπλισμός σε cm2: 5.5
As=5.5 cm2  4Φ14 -> 6.1575 cm2
```

2. Υπολογισμός πληθυσμού

Για το προσδιορισμό του ολικού πληθυσμού από το πλήθος των εργαζομένων, έγιναν οι εξής μετρήσεις σε n=5 νοικοκυριά μίας πόλης:

Νοικοκυριό	Πλήθος Μελών Yi	Πλήθος Εργαζομ. Xi
1	1	1
2	5	3
3	3	2
4	9	5
5	7	4

Υποθέτοντας γραμμική σχέση y=Ax+B, οι συντελεστές A, B υπολογίζονται:

$$A=\frac{n\sum x_i y_i-\left(\sum x_i\right)\left(\sum y_i\right)}{n\sum x_i^2-\left(\sum x_i\right)^2} \quad , \quad B=\frac{\sum y_i-A\sum x_i}{n} \quad , \quad i=1,2,\ldots,n$$

Το μέσο τετραγωνικό σφάλμα RMSE υπολογίζεται ως:

$$RMSE=\sqrt{\frac{\sum\left[y_i-\left(Ax_i+B\right)\right]^2}{n}} \quad , \quad i=1,2,\ldots,n$$

Να συντάξετε πρόγραμμα σε Matlab το οποίο να υπολογίζει και να τυπώνει στην οθόνη τους συντελεστές A, B και το μέσο τετραγωνικό σφάλμα για τις συγκεκριμένες μετρήσεις. Επίσης να τυπώνει το ολικό πληθυσμό (y) για 5 συνοικίες με πλήθος εργαζομένων (x) 1500, 1990, 2345, 325 και 589 αντίστοιχα.

Κώδικας

```
clear; clc; close all;
y = [1 5 3 9 7];
x = [1 3 2 5 4];
sx = sum(x);
sy = sum(y);
n = length(x);
a = (n*sum(x.*y) - sx*sy) / (n*sum(x.^2) - sx^2);
b = (sy - a*sx) / n;
rmse = sqrt(sum((y - (a.*x+b)).^2) / n);
disp([a b rmse]);
x = [1500 1900 2345 325 589];
disp(a.*x+b);
```

Αποτελέσματα (για As=5.5 cm^2)

```
     2    -1     0
        2999        3799        4689         649        1177
```

3. Υπολογισμός ροπής αδρανείας

Να υπολογιστεί η δευτεροβάθμια ροπή αδρανείας σε τριγωνική διατομή:

$$I_x = \int_0^b \int_0^{h-\frac{x}{b}h} y^2 dy\, dx$$

Να δοκιμάσετε το πρόγραμμα με b=30 cm και h=50 cm και να συγκρίνετε το αποτέλεσμα με την αναλυτική λύση:

$$I_x = \frac{bh^3}{12}$$

Για τον υπολογισμό των ολοκληρωμάτων να χρησιμοποιηθεί ο κανόνας του τραπεζίου:

$$\int_a^b f(x) dx = \Delta x [\frac{f(x_1)}{2} + f(x_2) + f(x_3) + \ldots + f(x_{N-1}) + f(x_N) + \frac{f(x_{N+1})}{2}] \ , \ x_1 = a \ , \ x_{N+1} = b$$

όπου a,b πεπερασμένοι πραγματικοί αριθμοί (a<b), η f(x) συνεχής στο διάστημα [a,b], και το διάστημα χωρίζεται σε N ίσα τμήματα μήκους Δx $(N \cdot \Delta x = b - a)$, δηλαδή $x_i = a + (i-1)\Delta x$.

Λύση

Το ολοκλήρωμα μπορεί να γραφεί:

$$I_x = \int_0^b f(x) dx \quad , \quad f(x) = \int_0^{h-\frac{x}{b}h} g(y) dy \quad , \quad g(y) = y^2$$

Και τα δύο ολοκληρώματα θα υπολογιστούν με το κανόνα του τραπεζίου. Έτσι θα φτιαχτεί συνάρτηση η οποία από τα x_i και y_i=f(x_i) θα υπολογίζει το ολοκλήρωμα:

```
function [olok] = oloktrapez(x, y)
    %Ολοκλήρωση με κανόνα τραπεζίου
    n = length(x);
    if n < 2
        error('Τουλάχιστον 2 σημεία απαιτούνται');
    end
    dx = x(2)-x(1);
    olok = sum(y) - y(1)/2 - y(end)/2;
    olok = dx*olok;
end
```

Η συνάρτηση f(x), η οποία εξαρτάται μόνο από το x, μπορεί να υπολογιστεί άμεσα με τον κανόνα του τραπεζίου. Χρησιμοποιούμε τη συνάρτηση linspace(a, b, n) του matlab που επιστρέφει ένα μητρώο με n τιμές ομοιόμορφα κατανεμημένες μεταξύ a και b (συμπεριλαμβανομένων των a και b):

```
function [ olok ] = f(b, h, x)
% y = 0:(h-x/b*h)/50:h-x/b*h;   Δεν λειτουργεί αν h-x/b*h = 0 !!!
y = linspace(0, h-x/b*h, 50);
gg = y.^2;
%olok = trapz(y, gg);
olok = oloktrapez(y, gg);
end
```

Με γνωστή τη συνάρτηση f(x) μπορεί να υπολογιστεί το ολοκλήρωμα $I_x = \int_0^b f(x) dx$ περίπου με τον ίδιο τρόπο:

```
function [ Ix ] = adran(b, h)
x = linspace(0, b, 50);
ff = zeros(1, length(x));
for i=1:length(x)
    ff(i) = f(b, h, x(i));
end
%Ix = trapz(x, ff);
Ix = oloktrapez(x, ff);
end
```

Ας σημειωθεί ότι δεν γίνεται να καλέσουμε τη συνάρτηση f(a,b,x) όπου το x είναι μητρώο, διότι η linspace() δεν δέχεται μητρώα και τα ορίσματα της oloktrapez() είναι ήδη μητρώα. Έτσι χρησιμοποιούμε δομή επανάληψης for.

Τέλος με ένα script ελέγχουμε το πρόγραμμα:

```
clear; clc; close all;
b = 30;
h = 50;
Ix = adran(b, h)
Ixanal = b.*h.^3./12
```

Το πρόγραμμα δίνει αποτελέσματα:

```
Ix =   3.1270e+05
Ixanal =      312500
```

4. Ολοκλήρωση με τον κανόνα του Simpson

4.1 Να συνταχθεί μία συνάρτηση που να υπολογίζει αριθμητικά το ολοκλήρωμα της f(x) στο διάστημα [a,b]:

$$I=\int_a^b f(x)dx$$

Όπου a,b πεπερασμένοι πραγματικοί αριθμοί (a<b) και η f(x) συνεχής στο διάστημα [a,b]. Αν το διάστημα χωριστεί σε N ίσα τμήματα μήκους h $(N \cdot h=b-a)$ και το N είναι άρτιος αριθμός, τότε το ολοκλήρωμα I δίνεται κατά προσέγγιση από την σχέση:

$$I=\frac{h}{3}[f(x_1)+4f(x_2)+2f(x_3)+\ldots+2f(x_{N-1})+4f(x_N)+f(x_{N+1})] \quad , \quad x_1=a \quad , \quad x_{N+1}=b$$

4.2 Για τη συνάρτηση f(x) = $x^4 - 3x^3 + 16x$, να υπολογιστεί διαδοχικά το ολοκλήρωμα για a=6, b=9 και αριθμό διαστημάτων N, 2N, 3N, ..., jN, (j+1)N, ... μέχρις ότου επιτευχθεί η ακρίβεια:

$$\frac{I_{(j+1)N}-I_{jN}}{I_{jN}}<\varepsilon$$

Θεωρείστε N=2, ε=10-6.

4.3 Να γίνει γραφική απεικόνιση της συνάρτησης f(x) για το διάστημα [a,b].

4.4 Να γίνει γραφική απεικόνιση της προσέγγισης.

Λύση

Η συνάρτηση f(x) θα είναι επίσης και συνάρτηση του Matlab.

```
function y=f(x)
y = x.^4 - 3 .* x.^3 + 16 .* x;
end
```

Παρατηρούμε ότι στη μέθοδο Simpson οι άρτιοι όροι πολλαπλασιάζονται με 4 και οι περιττοί επί 2 εκτός του πρώτου και τελευταίου. Συνεπώς υπολογίζουμε αρχικά όλους τους όρους, αθροίζουμε τους άρτιους από 2 έως n και το άθροισμα πολλαπλασιάζουμε επί 4, αθροίζουμε τους περιττούς από 3 έως n-1 και το άθροισμα πολλαπλασιάζουμε επί 2, και τέλος προσθέτουμε τον πρώτο και τον τελευταίο όρο.

```
function olok=simpson(a, b, n)
x=linspace(a, b, n+1);
dx = x(2)-x(1);
y = f(x);
olok = dx/3*(y(1) + 4*sum(y(2:2:n)) + 2*sum(y(3:2:n-1)) + y(n+1));
end
```

Με τη βοήθεια της Simpson υπολογίζουμε το ολοκλήρωμα για με 1N, 2N, 3N, ... όρους. Ορίζεται έτσι μία ακολουθία I_1, I_2, I_3, ... η οποία συγκλίνει προς την ακριβή τιμή του ολοκληρώματος. Θεωρούμε ότι έχει συγκλίνει όταν ισχύει η συνθήκη του θέματος.

```
clear;clc;close all;
a = 6;
b = 9;
n = 2;
eps = 1e-6;

j = 1;
olok(j) = simpson(a, b, n);
while 1
    olok(j+1) = simpson(a, b, (j+1)*n);
    if abs((olok(j+1)-olok(j)) / olok(j)) < eps
```

```
        break;
    end
    j = j + 1
end
j = j + 1;
disp(['the integral is ' num2str(olok(j))]);

x=linspace(a, b);
y=f(x);
figure(1);
plot(x, y);
xlabel('x');
ylabel('f(x)');
title('Function');

figure(2);
x = n:n:j.*n;
plot(x, olok);
xlabel('Number of steps');
ylabel('Integral');
title('Approximations');
```

5. Υπολογισμός ρίζας συνάρτησης

Η μέθοδος Monte-Carlo χρησιμοποιείται για την εύρεση της ρίζας μίας συνάρτησης εντός ενός διαστήματος [a, b]. Η μέθοδος υπολογίζει τη συνάρτηση σε τυχαία x μεταξύ a και b, και ως λύση λαμβάνει το x που δίνει f(x) που είναι κοντά στο μηδέν (κατ' απόλυτη τιμή).

α) Με την βοήθεια της μεθόδου Monte-Carlo να υπολογίσετε την ρίζα της συνάρτησης:

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

στο διάστημα [1, 4].

β) Να σχεδιάσετε την συνάρτηση f(x) και να δείξετε την τιμή της ρίζας που υπολογίσατε στο προηγούμενο ερώτημα με μια κόκκινη κουκκίδα.

Λύση

α) Η συνάρτηση f(x) θα είναι επίσης και συνάρτηση του Matlab:

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

Η συνάρτηση rand() του Matlab δίνει μητρώα με τυχαίους αριθμούς y από 0 έως 1. Για να υπολογίσουμε τυχαίους αριθμούς w από a έως b, χρησιμοποιούμε τον τύπο:

w = a + (b-a) .* y

Για την εφαρμογή της μεθόδου χρησιμοποιούμε 10000 τυχαίους αριθμούς:

```
clear; clc; close all;
a = 1;
b = 4;
x = a + rand(10000, 1) .* (b-a);
y = abs(f(x));
[ymin, i] = min(y);
disp([ 'riza=' num2str(x(i)) ]);
disp([ 'f(riza)=' num2str(ymin) ]);
```

β) Στη γραφική παράσταση της συνάρτησης, για να κάνουμε μία κουκκίδα (κύκλο) θα κάνουμε γράφημα δύο μητρώων u και v που το καθένα θα έχει ένα στοιχείο. Το u θα έχει μόνο τη ρίζα και το v μόνο την τιμής της f(x) για x=ρίζα (που πρέπει να είναι μηδέν).

```
xx = linspace(a, b, 50);
u = x(i);
v = ymin;
figure(1);
plot(xx, f(xx));
```

```
hold on;
plot(u, v, 'ro', 'MarkerSize', 10);
grid on;
xlabel('x');
ylabel('f(x)');
title('Function');
legend('f(x)', 'riza');
```

Το πρόγραμμα δίνει αποτελέσματα:

```
riza=2.7538
f(riza)=0.0021396
```