ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

Ακαδ. Έτος: 2018-2019

Μάθημα: Μέθοδοι Επίλυσης με Η/Υ

Τετάρτη, 16/1/2019

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

Παραδείγματα για την 12η παράδοση – Προσαρμογή δεδομένων

1. Υπολογισμός μέτρου ελαστικότητας

Στο αρχείο stress.txt (ιστοσελίδα mycourses) έχουν γραφεί ζεύγη μετρήσεων τάσης (MPa), παραμόρφωσης από δοκιμή θλίψης. Να υπολογιστεί το μέτρο ελαστικότητας του υλικού, λαμβάνοντας μόνο τις μετρήσεις που αντιστοιχούν στο γραμμικό τμήμα του διαγράμματος τάσεων/παραμορφώσεων.

Ανάλυση

Αρχικά γίνεται το γράφημα σ/ε που φαίνεται στο τέλος της άσκησης, και από αυτό βρίσκουμε ότι το γραμμικό τμήμα αποτελείται από τις 3 πρώτες μετρήσεις και οι οποίες θα ληφθούν υπόψη για τον υπολογισμό του μέτρου ελαστικότητας. Η σχέση τάσης/παραμόρφωσης δίνεται από τον τύπο:

$$\sigma = E \cdot \varepsilon$$

Τέτοια καμπύλη δεν υπάρχει έτοιμη στο Matlab. Η καμπύλη είναι γραμμική ως προς τους συντελεστές και με αυτόν τον τρόπο θα εισαχθεί στο Matlab.

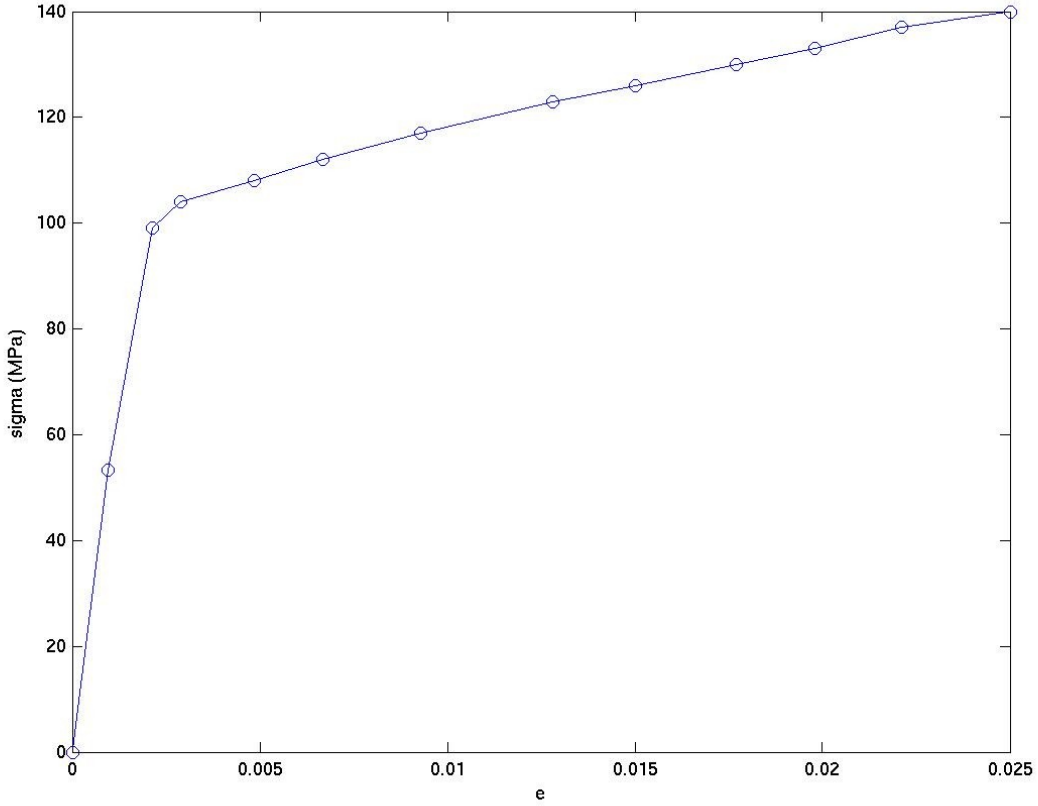
Κώδικας

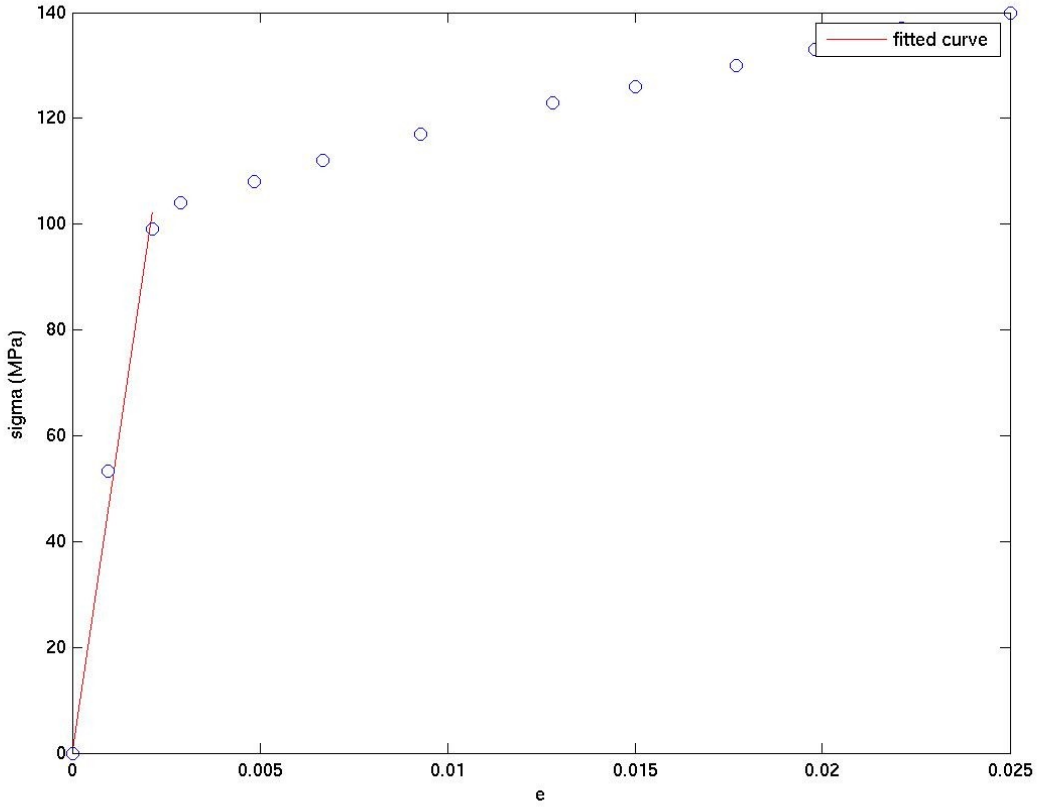
```
clear; clc; close all;
a=dlmread('stress.txt');
e = a(:, 1); s= a(:, 2); clear a;
figure();
plot(e, s, 'o-');      %Για εύρεση γραμμικού τμήματος
xlabel('e');
ylabel('sigma (Mpa)');

[xdata, ydata]=prepareCurveData(e(1:3), s(1:3));
ft=fittype({'x'});
[fr, er] = fit(xdata, ydata, ft);
figure();
plot(fr);
hold on;
plot(e, s, 'o');
xlabel('e');
ylabel('sigma (MPa)');
fprintf('Elasticity E=%.2f      RMSE=%.4f\n', fr.a, er.rmse);
```

Αποτελέσματα

```
Elasticity E=47955.34      RMSE=5.3794
```

Το δεύτερο γράφημα παρακάτω δείχνει τη προσαρμοσμένη καμπύλη μαζί με όλα τα δεδομένα.


sigma (MPa)
e
0
20
40
60
80
100
120
140
0
0.005
0.01
0.015
0.02
0.025


fitted curve
sigma (MPa)
e
0
20
40
60
80
100
120
140
0
0.005
0.01
0.015
0.02
0.025

2. Υπολογισμός εξίσωσης έντασης βροχής

Στο αρχείο diak.txt (ιστοσελίδα mycourses) έχουν γραφεί τριάδες μετρήσεων έντασης βροχής i (mm/hr), περιόδου επαναφοράς T (έτη) και διαρκείας (hr) από τον κοντινότερο βροχογραφικό σταθμό στον χείμαρρο Διακονιάρη Αχαΐας. Να προσδιοριστεί η απλοποιημένη εξίσωση έντασης της βροχής.

Ανάλυση

Η απλοποιημένη εξίσωση έντασης βροχής δίνεται από τον σχέση:

$$i=\frac{c\,T^{a}}{t^{b}}$$

Επειδή η σχέση είναι δύο μεταβλητών (T, t) θα χρησιμοποιηθεί προσαρμογή επιφάνειας. Παρόλο που το Matlab μπορεί να προσαρμόσει μη γραμμικές επιφάνειες σε δεδομένα, καλύτερα να γραμμικοποιηθεί η σχέση έτσι ώστε να μη χρειαστεί αρχική προσέγγιση των συντελεστών a, b, c. Παίρνοντας το λογάριθμο και των δύο μελών της σχέσης προκύπτει:

$$\ln i=\ln c+a\ln T-b\ln t \quad \text{ή} \quad z=p_{00}+p_{10}x+p_{01}y$$

όπου $z=\ln i$, $x=\ln T$, $y=-\ln t$, $p_{00}=\ln c$, $p_{10}=a$, $p_{01}=b$

Η σχέση έχει μετατραπεί σε πολυωνυμική επιφάνεια δύο μεταβλητών και είναι γραμμική και ως προς x και ως προς y ('poly11').

Κώδικας

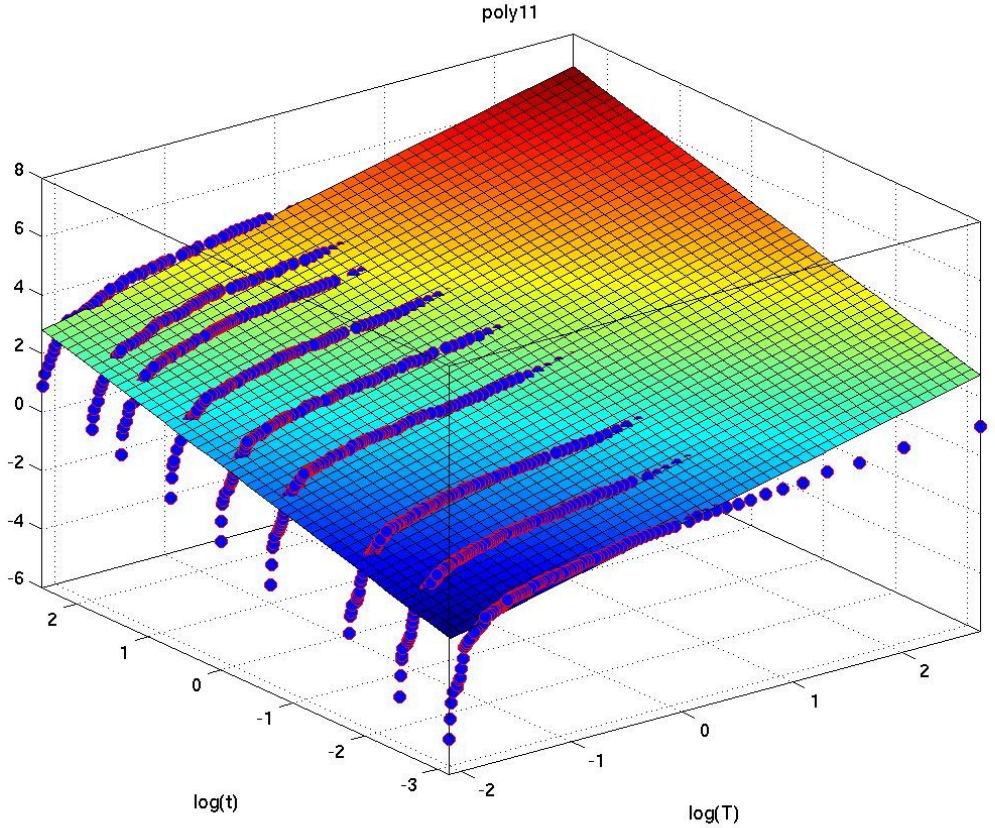
```
clear; clc; close all;
a = dlmread('diak.txt');
i = a(:, 1);
T = a(:, 2);
t = a(:, 3);
clear a;
z = log(i);
x = log(T);
y = -log(t);
ft = fittype('poly11');
[fr, er] = fit([x y], z, ft);
plot(fr); hold on;
plot3(x, y, z, 'ro', 'MarkerFaceColor', 'b');
xlabel('log(T)');
ylabel('log(t)');
title('poly11');
print('e1', '-djpeg');
fprintf('RMSE=%.3f    mean(z)=%.3f\n', er.rmse, mean(z));
a=fr.p10;
b=fr.p01;
c=exp(fr.p00);
fprintf('Coefficients  a=%.5f  b=%.5f  c=%.5f\n', a, b, c);
```

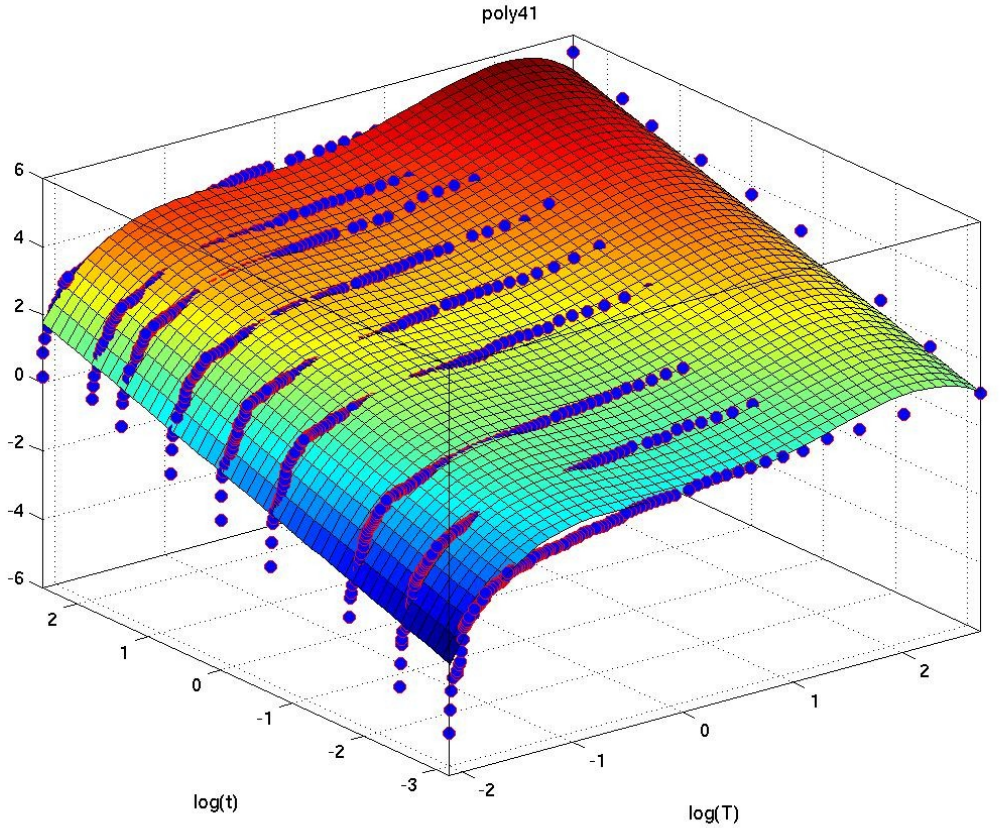
Αποτελέσματα

```
RMSE=0.642    mean(z)=1.662
Coefficients  a=0.85219  b=0.73073  c=16.17962
```

Ακρίβεια

Παρατηρούμε ότι το σφάλμα (RMSE) είναι μεγάλο σχέση με τη μέση ένταση βροχής (το λογάριθμο της έντασης). Αυτό φαίνεται και από το ακόλουθο γράφημα. Μετά από δοκιμές η πολυωνυμική καμπύλη 4ου βαθμού ως προς x γραμμική ως προς y ('poly41') δίνει το μικρότερο σφάλμα, όπως φαίνεται στο δεύτερο γράφημα παρακάτω. Όμως προτιμάται η πρώτη γραμμική επιφάνεια ('poly11') διότι για μεγάλες περιόδους επαναφοράς δίνει δυσμενέστερες (ασφαλέστερες) τιμές για την ένταση της βροχής, ενώ η δεύτερη επιφάνεια ('poly41') δίνει *μικρότερες* εντάσεις για τη μεγαλύτερη περίοδο, και επιπλέον φαίνεται να φθίνει για ακόμα μεγαλύτερες περιόδους ενώ θα έπρεπε σαφώς να κάνει το αντίθετο.


poly11
8
6
4
2
0
-2
-4
-6
2
1
0
-1
-2
-3
log(t)
-2
-1
0
1
2
log(T)


poly41
6
4
2
0
-2
-4
-6
2
1
0
-1
-2
-3
log(t)
-2
-1
0
1
2
log(T)

3. Υπολογισμός εξίσωσης μηχανικού ποσοστού οπλισμού δοκών

Στο αρχείο wm.txt (ιστοσελίδα mycourses) έχουν γραφεί ζεύγη μετρήσεων ανηγμένης ροπής μ_{sd} και μηχανικού ποσοστού ω_m που προέκυψαν από πολύπλοκους αναλυτικούς υπολογισμούς. Να βρεθεί ευκολομνημόνευτη προσεγγιστική σχέση που να δίνει αμελητέα μικρό σφάλμα.

Ανάλυση

Αρχικά γίνεται το γράφημα ω_m/μ_{sd} και από αυτό βρίσκουμε ότι η γραφική παράσταση είναι καμπύλη με μικρή καμπυλότητα. Συνεπώς θα χρησιμοποιηθεί η πιο απλή καμπύλη, η οποία είναι η πολυωνυμική δευτέρου βαθμού $y=ax^2 + bx + c$, όπου $y= \omega_m$ και $x= \mu_{sd}$.

Επειδή θα γίνουν διάφορες δοκιμές, συντάσσεται η συνάρτηση graf() που κάνει γράφημα της προσαρμογής της εκάστοτε καμπύλης στα δεδομένα.

Κώδικας 1

```
function []=graf(tit, xdata, ydata, fr, rmse, a, b, c)
figure();
xx = linspace(min(xdata), max(xdata));
plot(xx, fr(xx));
hold on;
plot(xdata, ydata, 'o');
legend('fitted', 'data', 'location', 'northwest');
xlabel('mu_{sd}');
ylabel('w_m');
title(tit);
fprintf('RMSE=%.4f   mean(wm)=%.4f\n', rmse, mean(ydata));
fprintf('coefficients  a=%f  b=%f  c=%f\n', a, b, c);
end

clear; clc; close all;
a = dlmread('wm.txt');
mu = a(:, 1);
wm = a(:, 2);
clear a;

[xdata, ydata] = prepareCurveData(mu, wm);
ft = fittype('poly2');
[fr, er] = fit(xdata, ydata, ft);
graf('a=free, b=free, c=free', xdata, ydata, fr,
er.rmse,fr.p1,fr.p2,fr.p3);
```

Αποτελέσματα κώδικα 1

```
RMSE=0.0018    mean(wm)=0.1788
coefficients  a=1.314529  b=0.903839  c=0.002495
```

Το σφάλμα είναι κάτω από 1% και είναι αμελητέο.

Περαιτέρω ανάλυση 2

Οι αριθμητικές τιμές των συντελεστών κάθε άλλο παρά ευκολομνημόνευτες είναι. Αν ο συντελεστής a στρογγυλευτεί σε a=1.31 το σφάλμα θα μεγαλώσει. Για να μη μεγαλώσει θα υπολογιστούν οι βέλτιστες τιμές των b, c με δεδομένο ότι a=1.31, δηλαδή η καμπύλη γίνεται $y=1.31x^2 + bx + c$.

Τέτοια καμπύλη δεν υπάρχει στο Matlab και έτσι συντάσσεται συνάρτηση f2(). Οι αρχικές τιμές για τους συντελεστές είναι οι συντελεστές που υπολογίστηκαν προηγουμένως. Ο κώδικας γίνεται:

Κώδικας 2

```
function [ y ] = f2(x, b, c)
y = 1.31*x.^2+b.*x+c;
end

clear; clc; close all;
a = dlmread('wm.txt');
mu = a(:, 1);
wm = a(:, 2);
```

```
clear a;

xdata, ydata] = prepareCurveData(mu, wm);
ft = fittype('f2(x, b, c)');
[fr, er] = fit(xdata, ydata, ft, 'StartPoint', [0.904 0.002495]);
graf('a=1.31, b=free, c=free', xdata, ydata, fr, er.rmse, 1.31,fr.b,fr.c);
```

Αποτελέσματα κώδικα 2

```
RMSE=0.0018     mean(wm)=0.1788
coefficients  a=1.310000  b=0.905198  c=0.002432
```

Το σφάλμα παραμένει κάτω από 1% και είναι αμελητέο.

Περαιτέρω ανάλυση 3

Ο συντελεστής a είναι τώρα ευκολομνημόνευτος αλλά ο b όχι. Αν ο συντελεστής b στρογγυλευτεί σε b=0.9 το σφάλμα θα μεγαλώσει. Για να μη μεγαλώσει θα υπολογιστεί η βέλτιστη τιμή του c με δεδομένο ότι a=1.31 και b=0.9, δηλαδή η καμπύλη γίνεται $y=1.31x^2 + 0.9x + c$.
Τέτοια καμπύλη δεν υπάρχει στο Matlab και έτσι συντάσσεται συνάρτηση f3(). Η αρχική τιμή για το συντελεστή είναι ο συντελεστής που υπολογίστηκε προηγουμένως. Ο κώδικας γίνεται:

Κώδικας 3

```
function [ y ] = f3(x, c)
y = 1.31*x.^2+0.9.*x+c;
end

clear; clc; close all;
a = dlmread('wm.txt');
mu = a(:, 1);
wm = a(:, 2);
clear a;

[xdata, ydata] = prepareCurveData(mu, wm);
ft = fittype('f3(x, c)');
[fr, er] = fit(xdata, ydata, ft, 'StartPoint', [0.002432]);
graf('a=1.31, b=0.9, c=free', xdata, ydata, fr, er.rmse, 1.31,0.9,fr.c);
```

Αποτελέσματα κώδικα 3

```
RMSE=0.0018     mean(wm)=0.1788
coefficients  a=1.310000  b=0.900000  c=0.003211
```

Το σφάλμα παραμένει κάτω από 1% και είναι αμελητέο.

Περαιτέρω ανάλυση 4

Οι συντελεστές a και b είναι τώρα ευκολομνημόνευτοι αλλά ο c όχι. Ο συντελεστής c στρογγυλεύεται σε c=0.0032, δηλαδή η καμπύλη γίνεται $y=1.31x^2 + 0.9x + 0.0032$. Δεν μπορεί να γίνει περαιτέρω προσαρμογή καμπύλης (όλοι οι συντελεστές έχουν προσδιοριστεί), αλλά χρειάζεται να υπολογιστεί το σφάλμα. Έτσι συντάσσεται συνάρτηση f4() και με αυτή υπολογίζεται το σφάλμα. Ο κώδικας γίνεται:

Κώδικας 4

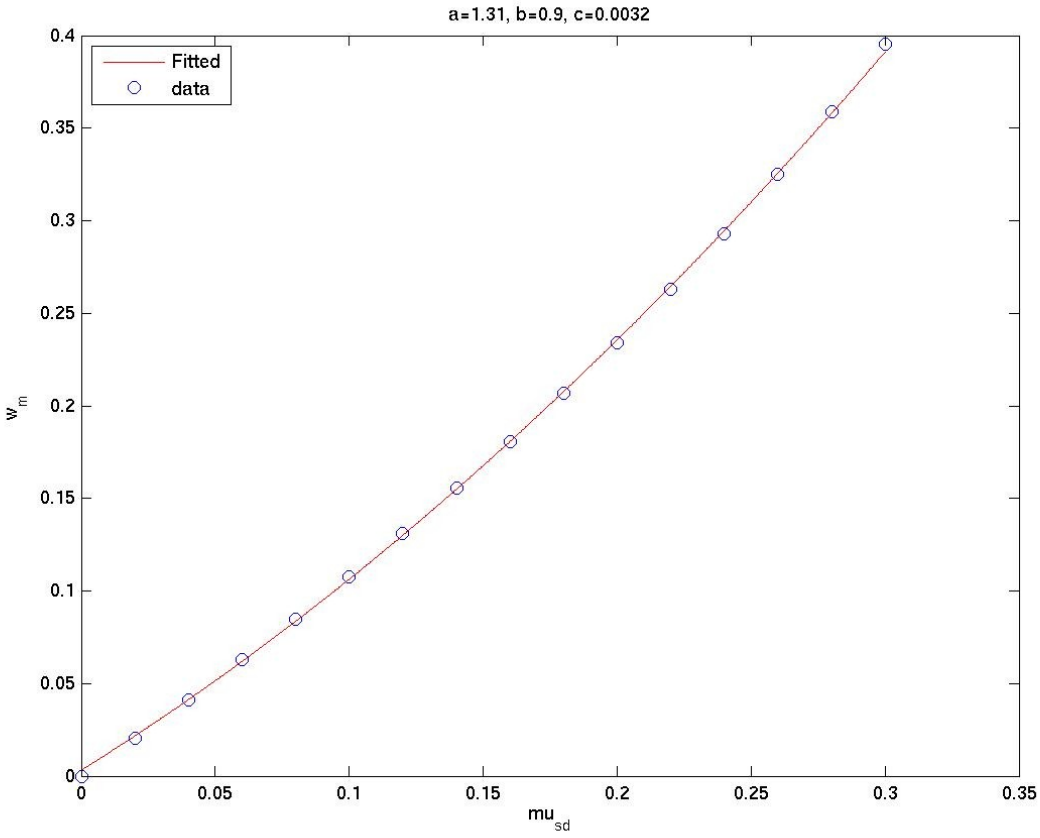
```
function [ y ] = f4(x)
y = 1.31*x.^2+0.9.*x+0.0032;
end

clear; clc; close all;
a = dlmread('wm.txt');
mu = a(:, 1);
wm = a(:, 2);
clear a;

[xdata, ydata] = prepareCurveData(mu, wm);
y2=f4(xdata);
rmse = sqrt(sum((ydata-y2).^2)/length(ydata));
```

```
graf('a=1.31, b=0.9, c=0.0032',xdata,ydata,@f4,rmse,1.31,0.9,0.0032);
```

Αποτελέσματα κώδικα 4

```
RMSE=0.0017     mean(wm)=0.1788
coefficients  a=1.310000  b=0.900000  c=0.003200
```

Το σφάλμα παραμένει κάτω από 1% και είναι αμελητέο. Τώρα ολόκληρη η σχέση που δίνει το μηχανικό ποσοστό οπλισμού είναι ευκολομνημόνευτη και το σφάλμα που έχει είναι αμελητέο. Παρακάτω δίνεται το γράφημα της τελικής σχέσης.



4. Αριθμητικός υπολογισμός παραγώγου

Να υπολογιστεί αριθμητικά η παράγωγος αυθαίρετης συνάρτησης. Δοκιμάστε το πρόγραμμά σας με τις συναρτήσεις sin(x), sqrt(x) και e^x για x=1.

Λύση

Ο αριθμητικός υπολογισμός παραγώγου με μεγάλη ακρίβεια είναι απροσδόκητα δύσκολος. Η παράγωγος ορίζεται:

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

Η παράγωγος μπορεί να προσεγγιστεί από το κλάσμα αν το h είναι πολύ μικρό. Θεωρητικά όσο το h γίνεται μικρότερο, η ακρίβεια θα μεγαλώνει. Δυστυχώς αριθμητική ανάλυση έχει αποδείξει ότι αν το h γίνει μικρότερο από την τετραγωνική ρίζα της ακρίβειας των πραγματικών αριθμών (~10^{-14}) το σφάλμα αποκοπής καθιστά το κλάσμα αναξιόπιστο. Στην πράξη το h πρέπει να είναι μεγαλύτερο από 10^{-7} διότι εξαρτάται και από τη συνάρτηση που παραγωγίζεται.

Για να αυξηθεί ακρίβεια καταρχήν χρησιμοποιείται η συμμετρική σχέση:

$$f'(x)=\lim_{h\to 0}\frac{f(x+h)-f(x-h)}{2h}$$

Στη συνέχεια για σταθερό x ορίζεται η συνάρτηση:

$$g(h)=\frac{f(x+h)-f(x-h)}{2h}$$

Από τον πιο πάνω ορισμό συνάγεται:

$$f'(x)=g(0)$$

Η συνάρτηση g(h) δεν είναι γνωστή, αλλά μπορεί να προσεγγιστεί με πολυώνυμο από μερικές τιμές της, πχ για h=0.001, 0.0005, 0.0001. Ας σημειωθεί ότι αυτές οι τιμές είναι πολύ πιο μεγάλες από 10^{-7} για να αποφευχθεί το σφάλμα αποκοπής. Επίσης η συνάρτηση f(x+h) είναι πρακτικά γραμμική για πολύ μικρά h, και έτσι το πολυώνυμο δεν χρειάζεται να είναι μεγάλου βαθμού. Έτσι επιλέγεται πολυώνυμο δευτέρου βαθμού.

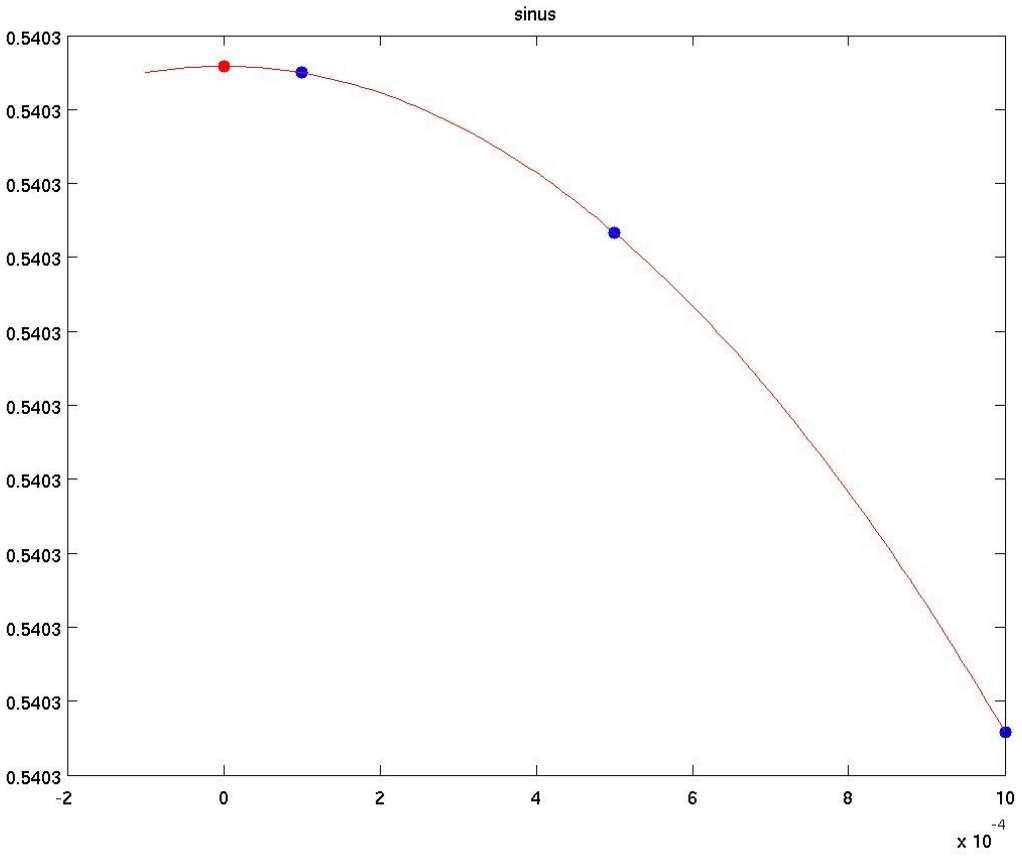
Για να υπολογίσουμε την παράγωγο 3 διαφορετικών συναρτήσεων, χρησιμοποιούμε ανώνυμες συναρτήσεις που αποθηκεύονται σε ανομοιογενές μητρώο:

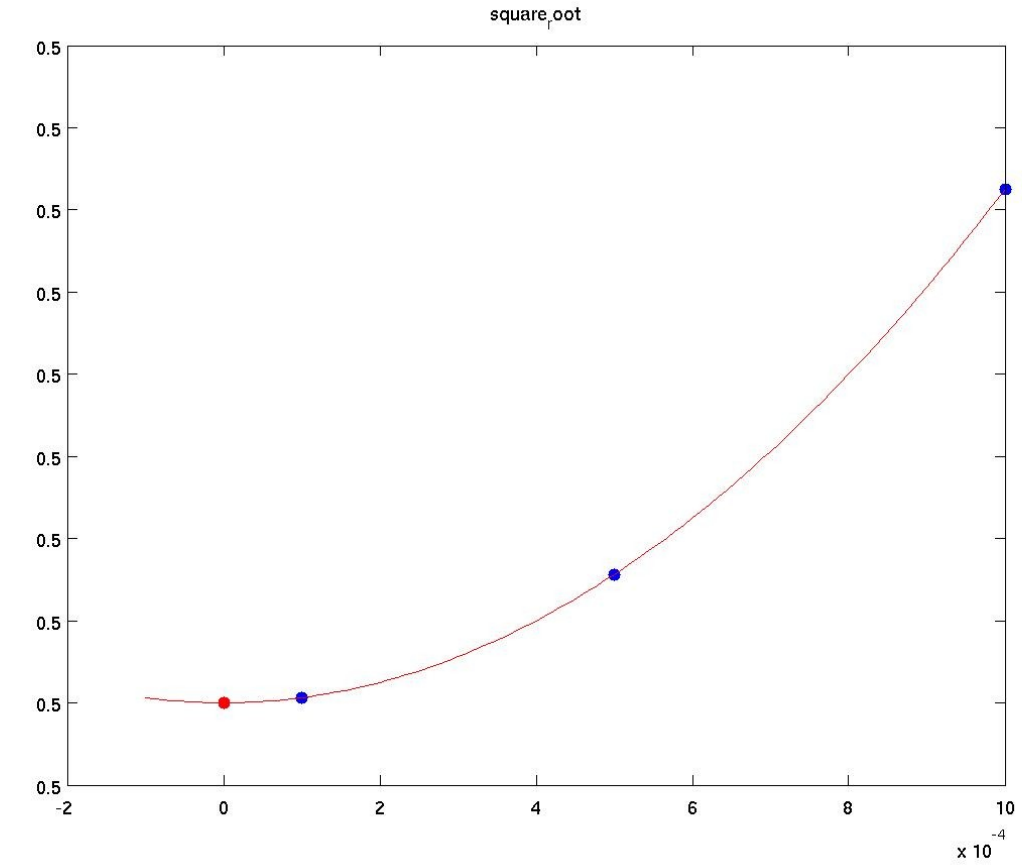
```
clear; clc; close all;
x = 1;
funs = {'sinus',       @(x) sin(x),  @(x) cos(x)
        'square root', @(x) sqrt(x), @(x) 1./(2*sqrt(x))
        'exponential', @(x) exp(x),  @(x) exp(x)         };

h3 = [0.001 0.0005 0.0001];
for i=1:length(funs)
    name = funs{i, 1};
    f = funs{i, 2};
    df = funs{i, 3};
    dy3 = (f(x+h3)-f(x-h3)) ./ (2*h3);
    p = polyfit(h3, dy3, 2);
    parag = polyval(p, 0);

    h = linspace(-0.0001, 0.001, 100);
    dy = polyval(p, h);
    figure(i);
    plot(h3, dy3, 'b.', 'markersize', 20);
    hold on;
    plot(h, dy, 'r');
    plot(0, parag, 'r.', 'markersize', 20);
    title(name);
    fprintf('%s function:\n', name);
    fprintf('    numerical  derivative=%.10f\n', parag);
    fprintf('    analytical derivative=%.10f\n', df(x));
end
```

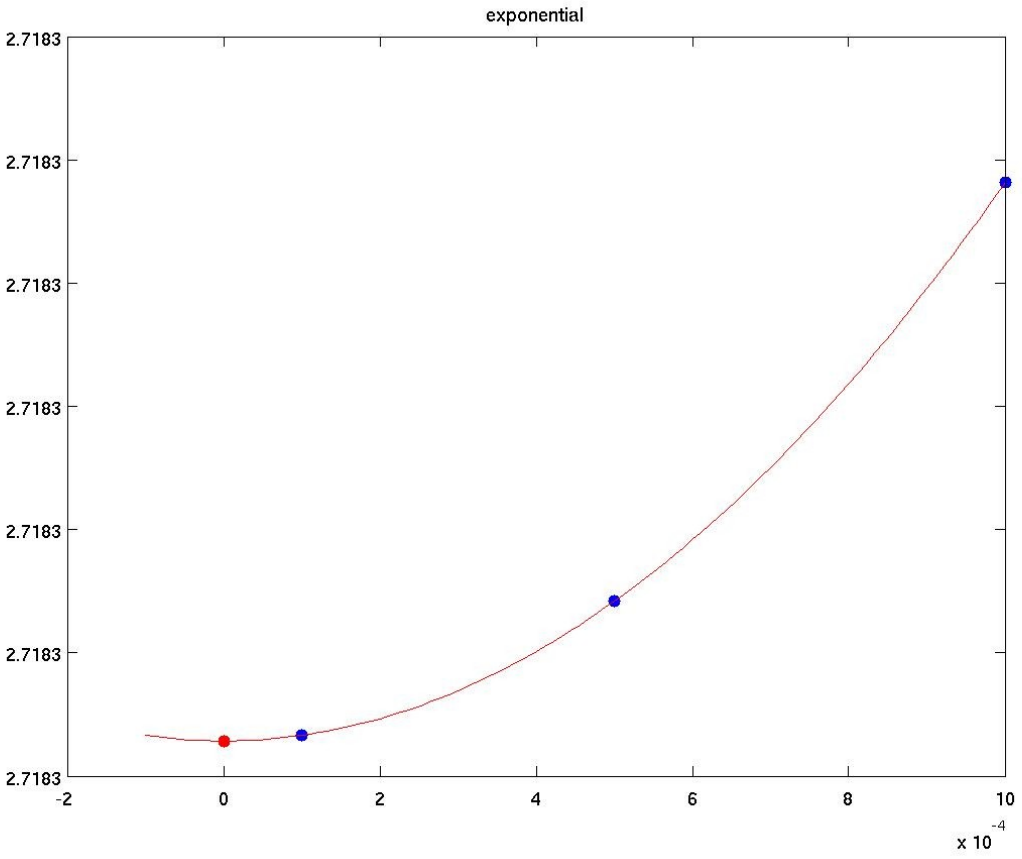
Το πρόγραμμα δίνει αποτελέσματα:

```
sinus function:
    numerical  derivative=0.5403023059
    analytical derivative=0.5403023059
square root function:
    numerical  derivative=0.5000000000
    analytical derivative=0.5000000000
exponential function:
    numerical  derivative=2.7182818285
    analytical derivative=2.7182818285
```

και τα γραφήματα:




square_oot
0.5
0.5
0.5
0.5
0.5
0.5
0.5
0.5
0.5
-2
0
2
4
6
8
10
x 10^-4


exponential
2.7183
2.7183
2.7183
2.7183
2.7183
2.7183
2.7183
-2
0
2
4
6
8
10
x 10^-4