

Μέθοδοι Επίλυσης με Η/Υ

***12^ο Μάθημα:** Εργαλειοθήκες (toolboxes)*

Ν.Δ. Λαγαρός, Α. Στάμος, Χ. Φραγκουδάκης

Το σημερινό μάθημα περιέχει:

- 1) Προσαρμογή δεδομένων,
- 2) Παλινδρόμηση με πολυώνυμα,
- 3) Παλινδρόμηση με άλλες/αυθαίρετες συναρτήσεις
- 4) Μοντελοποίηση με κυβικά πολυώνυμα
- 5) Παλινδρόμηση 2 μεταβλητών (επιφάνειες)

Προσαρμογή δεδομένων (curve fitting)

Η **Προσαρμογή Καμπύλης** (Curve Fitting), είναι η προσαρμογή μίας ομάδας σημείων ενός διαγράμματος σε μια καμπύλη και η εύρεση της αντίστοιχης εξίσωσης. Το MatLab διαθέτει και συναρτήσεις που πραγματοποιούν **Πολυωνυμική Προσέγγιση** (Polynomial Fitting) και συναρτήσεις που πραγματοποιούν προσέγγιση με καμπύλες **Splines** (Cubic Splines).

Οι πολυωνυμικές είναι απλές και σίγουρες βάσεις για προσέγγιση συνεχών αλλά και τετραγωνικών ολοκληρώσιμων συναρτήσεων με διάφορες νόρμες. Οι βασικές συναρτήσεις του MatLab για Πολυωνυμική Προσέγγιση, είναι οι: **polyfit & polyval**.

Π.χ.:

Η εντολή **coef = polyfit(X,Y,d)** υπολογίζει τους συντελεστές coef του πολυωνύμου βαθμού d, που προσεγγίζει καλύτερα τα δεδομένα (X, Y).

Αντίστοιχα, η εντολή **Ybest = polyval(coef,X)** υπολογίζει μέσω του πολυωνύμου αυτού, τα νέα Y (Ybest) που αντιστοιχούν σε κάθε X.

Προσαρμογή δεδομένων (curve fitting)

Παράδειγμα:

```
x = 0:7; y = sin(x);

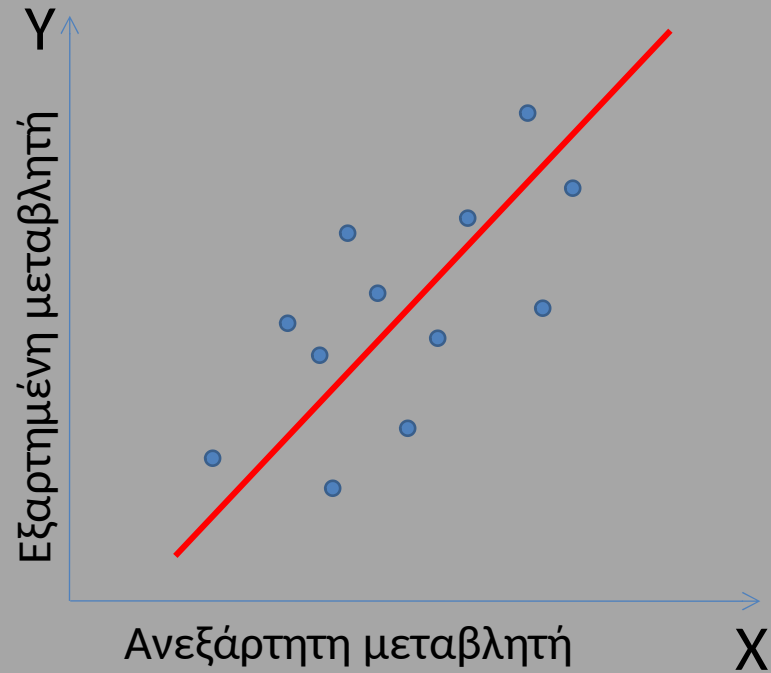
for d = 1:4
    coef = polyfit(x,y,d);
    ybest = polyval(coef,x);
    x1 = 0:.2:7; ybest1 = polyval(coef,x1);
    subplot(2,2,d)
    plot(x,y,'o',x,ybest,'+',x1,ybest1,'-')
    title(['d=' num2str(d,1)])
    axis([0 8 -1.5 1.5]),grid
end
```

Μέθοδος Ελαχίστων Τετραγώνων

Συχνά συναντάμε ένα σύνολο πειραματικών δεδομένων που δεν βρίσκονται ακριβώς στην ίδια ευθεία, παρόλο που η σχέση που τα διέπει είναι γραμμική. Τότε χρειάζεται να χαράξουμε μια ευθεία γραμμή που να τα εκπροσωπεί όσο γίνεται καλύτερα. Ο υπολογισμός της ευθείας που προσεγγίζει καλύτερα τα σημεία μας γίνεται με τη μέθοδο των **Ελαχίστων Τετραγώνων** (Least Squares - LSM) και η διαδικασία λέγεται **Γραμμική Παλινδρόμηση** (Linear Regression).

Αντίθετα από άλλες μεθόδους προσαρμογής που πρέπει να διέρχονται από όλα τα σημεία του διαγράμματος, η βέλτιστη ευθεία συχνά δεν περιλαμβάνει κανένα από τα σημεία του διαγράμματος. Είναι όμως η μοναδική ευθεία που ικανοποιεί το κριτήριο των ελαχίστων τετραγώνων το οποίο απαιτεί: το άθροισμα των τετραγώνων των αποκλίσεων όλων των σημείων από αυτή να είναι το ελάχιστο δυνατό. Η Πολυωνυμική Προσέγγιση που είδαμε πιο πάνω βασίζεται σε αυτό το κριτήριο, και μάλιστα στη περίπτωση $d=1$ που το πολυώνυμο είναι 1ου βαθμού (γραμμή), υλοποιεί τη Γραμμική Παλινδρόμηση.

Γραμμική Παλινδρόμηση



Μέθοδος των ελαχίστων τετραγώνων

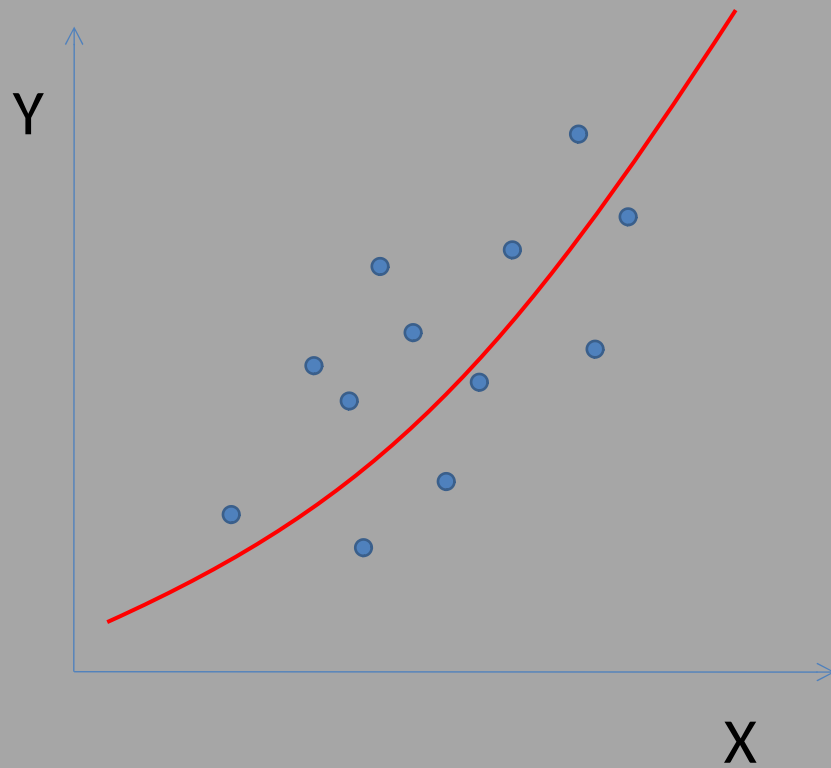
$$\hat{Y}_i = b_0 + b_1 X_i$$

$$\sum_{i=1}^n (\hat{Y}_i - Y_i)^2 = \text{minimum}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{n \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$

Παλινδρόμηση



$$y = b_0 + b_1x + b_2x^2$$

$$y = b_0 + b_1x + b_2x^2 \dots \dots \dots b_nx^n$$

Παλινδρόμηση με το Matlab

`Polyfit` - υπολογισμός καμπύλης παλινδρόμησης

`p = polyfit(x,y,n)`, υπολογίζει για τα δεδομένα x, y τους συντελεστές του πολυωνύμου p που έχει βαθμό n και προσαρμόζεται στα δεδομένα

$$p(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1}$$

Γραμμική Παλινδρόμηση

```
>> x = (0: 0.1: 2.5)';
```

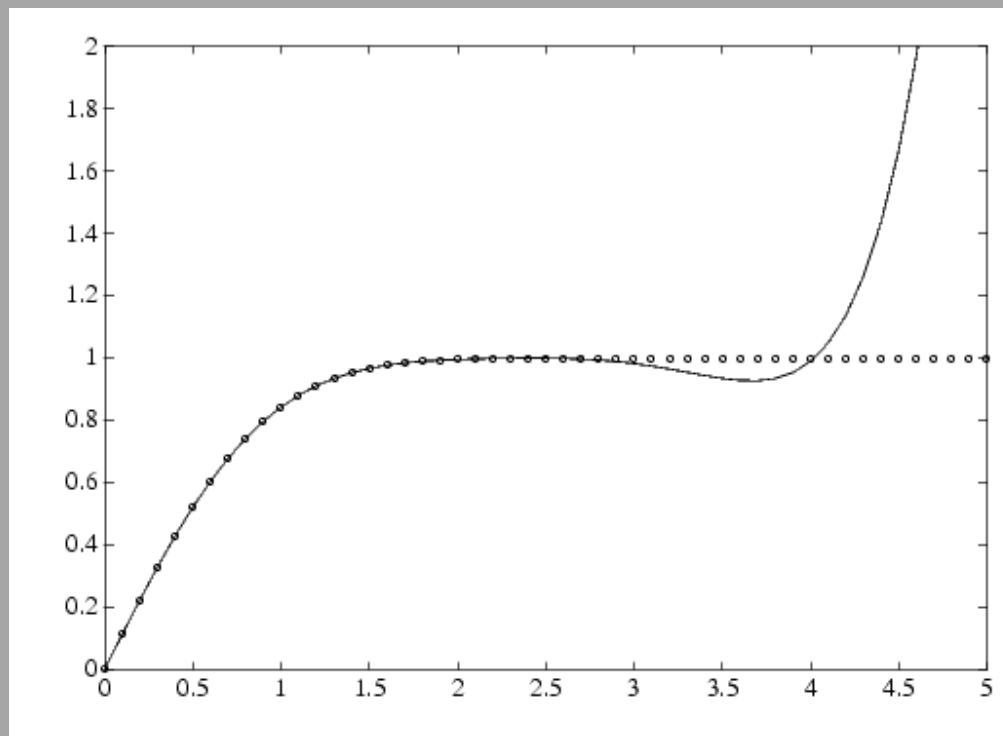
```
>> y = erf(x);
```

```
>> p = polyfit(x,y,6)
```

```
p =
```

```
0.0084 -0.0983 0.4217 -0.7435 0.1471 1.1064 0.0004
```

$$0.0084x^6 - 0.0983x^5 + 0.4217x^4 - 0.7435x^3 + 0.1471x^2 + 1.1064x + 0.0004$$



Γραμμική Παλινδρόμηση

$y = \text{polyval}(p, x)$ Υπολογίζει την τιμή ενός πολυωνύμου σε ένα σημείο x .

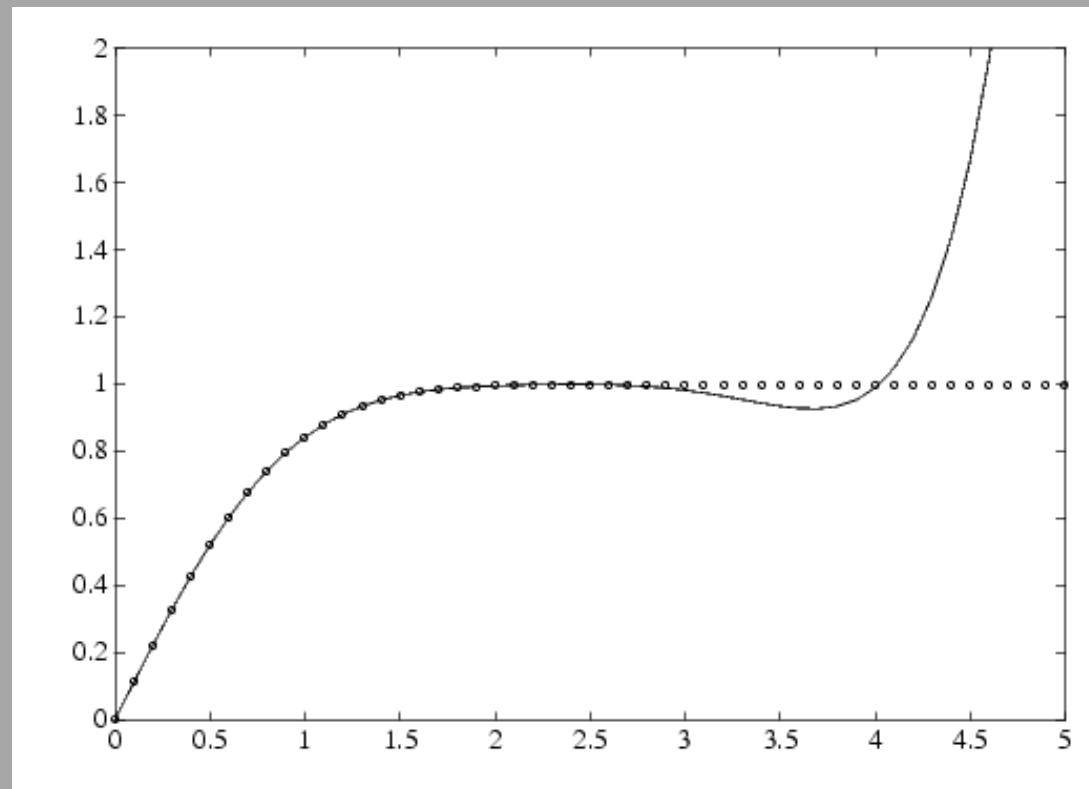
Είσοδος: οι συντελεστές του πολυωνύμου p

Έξοδος : οι τιμές που αντιστοιχούν στα σημεία x

```
p = polyfit(x,y,6)
```

```
f = polyval(p,x);
```

```
plot(x,y,'o',x,f,'-')
```



Παλινδρόμηση με άλλες συναρτήσεις

Έστω τα δεδομένα βάθους/παροχής:

```
X=[0 0.2 0.21 0.3 0.38 0.4 0.405 0.5 0.53 0.55 0.58  
1];  
y=[0 6.81261 8.16116 8.36719 8.63658 9.03462 9.66808  
11.20181 11.88206 11.82824 12.02988 15.74552];
```

Έλεγχος αν τα δεδομένα είναι σε μορφή στηλών, έχουν το ίδιο πλήθος στοιχείων, είναι πραγματικοί αριθμοί κλπ:

```
>>[xdata, ydata] = prepareCurveData(x, y)
```

Ορισμός τύπου καμπύλης (εδώ πολυώνυμο πρώτου βαθμού – γραμμικό):

```
>>ft = fittype('poly1')
```

Υπολογισμός συντελεστών και RMSE:

```
[fitresult, er] = fit(xdata, ydata, ft);
```

Η καμπύλη ως συνάρτηση: fitresult

Συντελεστές: fitresult.p1, fitresult.p2

RMSE: er.rmse

Παλινδρόμηση με άλλες συναρτήσεις

Για οπτικό έλεγχο σφάλματος γίνεται γράφημα (το plot αναγνωρίζει το fitresult):

```
plot(fitresult);  
hold on;  
plot(xdata, ydata, 'o');
```

- Δοκιμή με πολυώνυμο 2^{ου} βαθμού ('poly2') δίνει μικρότερο σφάλμα.
- Δοκιμή με πολυώνυμο 3^{ου} βαθμού ('poly3') δίνει μικρότερο σφάλμα, ΑΛΛΑ παρουσιάζει καμπή που δεν έχει παρατηρηθεί σε χείμαρρους και απορρίπτεται.
- Δοκιμή με εκθετική καμπύλη ('exp1': $y = a e^{bx}$) δίνει μεγαλύτερο σφάλμα

Ψηλότερος βαθμός πολυωνύμου (ή άλλης καμπύλης) δεν συνεπάγεται απαραίτητα μεγαλύτερη ακρίβεια. Συχνά συμβαίνει το αντίθετο (πχ η καμπύλη παρουσιάζει ακρότατο που δεν συμφωνεί με τη θεωρία).

Άλλες έτοιμες καμπύλες είναι: 'fourier1', 'fourier2', 'gauss1', 'gauss2', 'power1', 'power2', 'ratij' κλπ.

Παλινδρόμηση με αυθαίρετη συνάρτηση

Μια ειδική μορφή καμπύλης με ευρεία χρήση είναι μια (αυθαίρετη) μη γραμμική καμπύλη, αλλά γραμμική ως προς τους συντελεστές, η οποία ορίζεται από το χρήστη:

$$f(x)=a x^2+b \sin(2x)+c$$

```
>>ft = fitttype({'x^2', 'sin(2*x)', '1'})
```

Το ίδιο μπορεί να γίνει συντάσσοντας τυχαία συνάρτηση για να χρησιμοποιηθεί ως καμπύλη:

```
function [ y ] = myfun(x, a, b, c)
y=a*x.^2 + b*sin(2*x) + c;
end
>>ft = fitttype('myfun(x,a,b,c)')
```

Σε αυτή την περίπτωση χρειάζονται και αρχικές τιμές των παραμέτρων:

```
>>[fitresult, er] = fit(xdata,ydata,ft, 'StartPoint',
[5 10 1.5])
```

Αν οι αρχικές τιμές δεν είναι αρκετά κοντά στις τελικές, τότε η LSM μπορεί να μην συγκλίνει, ή να συγκλίνει σε τιμές που δεν είναι οι βέλτιστες.

Μοντελοποίηση δεδομένων με κυβικά πολυώνυμα (splines)

Τα κυβικά πολυώνυμα επειδή έχουν μεταβλητή καμπυλότητα (δεύτερη παράγωγο) μπορούν να μοντελοποιήσουν πολλές μορφές δεδομένων:

```
>>x=linspace(0,pi/2,100); y=sin(x);  
[x,y]=prepareCurveData(x, y);  
>>ft=fitttype('poly3'); [fr,er]=fit(x,y,ft);
```

Η ακρίβεια γίνεται μικρότερη όταν τα δεδομένα έχουν πιο πολλές μεταβολές:

```
>>x=linspace(0,2*pi,100); y=sin(x);  
[fr,er]=fit(x,y,ft);
```

Η ακρίβεια χάνεται όταν τα δεδομένα έχουν ακόμα πιο πολλές μεταβολές:

```
>>x=linspace(0,8*pi,100); y=sin(x);  
[fr,er]=fit(x,y,ft);
```

Μοντελοποίηση δεδομένων με κυβικά πολυώνυμα (splines)

Η ακρίβεια διορθώνεται χρησιμοποιώντας διαφορετικό κυβικό πολυώνυμο ανά δύο σημεία, έτσι ώστε δύο διαδοχικά πολυώνυμα να έχουν ίδια τιμή και παράγωγο στο κοινό τους σημείο (spline). Έχει την ιδιότητα να περνάει ακριβώς από όλα τα σημεία:

```
>>x=linspace(0,8*pi,100); y=sin(x);  
[x,y]=prepareCurveData(x, y);  
>>ft=fitttype('cubicspline');  
>>[fr,er]=fit(x,y,ft);
```

Η μοντελοποίηση με spline, που είναι το πολυώνυμο χαμηλότερου βαθμού με την ως άνω ιδιότητα, είναι πολύ διαδεδομένη.

Παλινδρόμηση δύο μεταβλητών - επιφάνειες

Υπόθεση καμπύλης επιφάνειας (γραμμική, πολυωνυμική, ρητή κλπ):

$$f(x,y)=p00 + p10 x + p01 y + p20 x^2 + p11 x y + p02 y^2$$

Δεδομένα της μορφής (x, y, z) :

(0,0,0), (3,4,5), (6,8,10), ...

Στο Matlab ορισμός τύπου επιφάνειας (εδώ πολυώνυμο δύο μεταβλητών δευτέρου βαθμού και ως προς τις δύο μεταβλητές):

```
>>x = rand(100, 1); y = rand(100, 1); z = sqrt(x.^2 +  
y.^2);  
ft = fittype('poly22');  
[fr, er]=fit([x y], z, ft);  
plot(fr);  
hold on; plot3(x, y, z, 'ro', 'MarkerFaceColor', 'b');
```

Άλλη έτοιμη επιφάνεια είναι η: 'lowess'

Παλινδρόμηση με αυθαίρετη επιφάνεια

Μια ειδική μορφή επιφάνειας με ευρεία χρήση είναι μια (αυθαίρετη) μη γραμμική επιφάνεια, αλλά γραμμική ως προς τους συντελεστές, η οποία ορίζεται από το χρήστη:

$$f(x)=a x + b y + c x y + d$$

```
>>ft=fitttype({'x', 'y', 'x*y', '1'}, 'independent',  
{ 'x', 'y'}, ...  
             'dependent', 'z');
```

Στις επιφάνειες είναι απαραίτητο να οριστούν οι ανεξάρτητες και η εξηρημένη μεταβλητές.

Παλινδρόμηση με αυθαίρετη επιφάνεια

Το ίδιο μπορεί με σύνταξη τυχαίας συνάρτησης για να χρησιμοποιηθεί ως επιφάνεια:

```
function [ z ] = mysurf(x, y, a, b, c, d)
z=a*x + b*y + c*x.*y + d;
End
>>ft=fitttype('mysurf(x,y,a,b,c,d)', 'independent',
{'x', 'y'}, ...
'dependent', 'z');
```

Σε αυτή την περίπτωση χρειάζονται και αρχικές τιμές των παραμέτρων:

```
>>[fp, er] = fit([x y], z, ft, 'StartPoint', [1 1 0
0])
```

Αν οι αρχικές τιμές δεν είναι αρκετά κοντά στις τελικές, τότε η LSM μπορεί να μην συγκλίνει, ή να συγκλίνει σε τιμές που δεν είναι οι βέλτιστες.

Χρήσεις μοντελοποίησης (ή προσαρμογής καμπύλης)

Εύρεση άγνωστου μοντέλου που συνδέει 2 ή 3 φυσικά μεγέθη, από πειραματικά δεδομένα.

Πρόβλεψη της τιμής φυσικού μεγέθους σε μελλοντικό χρόνο από τιμές του παρελθόντος.

Παρεμβολή σε πίνακα τιμών, με πιο εύχρηστο τρόπο.

Εξομάλυνση δεδομένων με θόρυβο, σύνδεση σημείων με ομαλή καμπύλη.

Χρήσεις μοντελοποίησης (ή προσαρμογής καμπύλης)

- Αντοχή υλικών: Σχέση μεταξύ τάσης και παραμόρφωσης, υπολογισμός μέτρου ελαστικότητας
- Σκυρόδεμα: υπολογισμός μηχανικού ποσοστού οπλισμού σε δοκούς
- Οικοδομική: Μέγιστη επιτρεπόμενη θερμοπερατότητα σε κτίρια
- Υδραυλική: Υπολογισμός σχέσης στάθμης/παροχής σε χείμαρρο και σε υπερχειλιστή
- Πολεοδομία: πρόβλεψη πληθυσμού, πρόβλεψη συντελεστή δόμησης
- Τοπογραφία: Ψηφιακά μοντέλα εδάφους, εξομάλυνση ισοϋψών
- Οδοποιία: Σύνδεση σημείων άξονα ή οριογραμμών με ομαλή καμπύλη

Ερωτήσεις...