

Μέθοδοι Επίλυσης με Η/Υ

5^ο Μάθημα: *Βασικές εντολές - Συμβολοσειρές*

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

Το σημερινό μάθημα περιέχει:

- εντολές min, max, sort, find
- υπολογισμός ολοκληρωμάτων, παραγώγων
- Συμβολοσειρές
- Αποθήκευση εικόνων

Η εντολή find

Η εντολή find βρίσκει τους αριθμούς που ικανοποιούν μία συνθήκη

```
x = [46, 56, 1, 15, 30, 40]
y = [55, 60, 61, 57, 58, 59]

biggerThan30 = find(x>30) % ζητάμε τα μέλη του x που είναι μεγαλύτερα από 30

>> biggerThan30 = % μας απαντάει ότι είναι η 1η, η 2η και η 6η τιμή του x
    1
    2
    6

x(biggerThan30) % οι τις τιμές που είναι μεγαλύτερες από το 30

>> ans =
    46
    56
    40
```

Οι εντολές min/max

Οι εντολές min και max δίνουν τη μέγιστη και την ελάχιστη τιμή ενός διανύσματος:

`[i1,i2] = min(X)`

`[i1,i2] = max(X)`

i1 η μέγιστη/ελάχιστη τιμή και

i2 είναι η θέση του **X** που εμφανίζεται το μέγιστο/ελάχιστο

```
nn = rand(30,1); % 30 τυχαίοι αριθμοί στο διάστημα [0,1]
```

```
a = 5; b = 20;
```

```
Numbers = a + nn*(b-a); % 30 τυχαίοι αριθμοί στο διάστημα [a,b]
```

```
[elaxisto,thesiElaxistou]= min(Numbers)
```

```
[megisto,thesiMegistou]= max(Numbers)
```

```
elaxistoNew = Numbers(thesiElaxistou) % idio me to 'elaxisto'
```

```
megistoNew = Numbers(thesiMegistou)
```

Οι εντολές min/max

εναλλακτικά με εντολές for:

```
thesiElaxistou = 1;  
for i = 2:length(Numbers)  
    if Numbers(i) < Numbers(thesiElaxistou)  
        thesiElaxistou = i;  
    end  
end  
Elasisto = Numbers(thesiElaxistou)
```

εάν το X είναι μητρώο, το αποτέλεσμα είναι η ελάχιστη (ή μέγιστη) τιμή κάθε στήλης

```
X = [12  45  67  
     56  32  45  
     63   4  23]
```

```
mm = min(X)  
>>mm = [12  4  23]
```

Η εντολή sort

Η εντολή sort κατατάσσει σε αύξουσα (ή φθίνουσα) σειρά τους αριθμούς ενός διανύσματος:

```
nn = rand(30,1);  
a = 5; b = 20;  
NumUnsorted = a + nn*(b-a);  
  
NumSorted_auxousa = sort(NumUnsorted)  
NumSorted_fthinousa = sort(NumUnsorted, 'descend')
```

Εναλλακτικά :

```
x = [46, 56, 1, 15, 30, 40]  
y = [55, 60, 61, 57, 58, 59]  
  
[xSorted tt] = sort(x) % ταξινόμηση των x, το tt είναι η σειρά των στοιχείων του x  
                    % με βάση την σειρά τους  
ySorted = y(tt)       % ταξινόμηση των y, με βάση την αύξουσα σειρά των x που  
                    % αποθηκεύσαμε στο tt
```

sum και prod

Η εντολή `sum` προσθέτει τους αριθμούς ενός διανύσματος και η εντολή `prod` βρίσκει το γινόμενο τους

```
nn = rand(30,1);  
a = 5; b = 20;  
Numbers= a + nn*(b-a)  
  
athroisma = sum(Numbers)  
ginomeno = prod(Numbers)
```

εάν εναλλακτικά χρησιμοποιούσαμε εντολές `for`:

```
athroisma = 0;  
ginomeno = 1;  
for i = 1:length(Numbers)  
    athroisma = athroisma + Numbers(i);  
    ginomeno = ginomeno*Numbers(i);  
end  
athroisma  
ginomeno
```

sum και prod

Με την `sum` μπορούμε να υπολογίσουμε το άθροισμα των στηλών και των γραμμών ενός πίνακα:

```
X = [12  45  67  
     56  32  45  
     63   4  23]
```

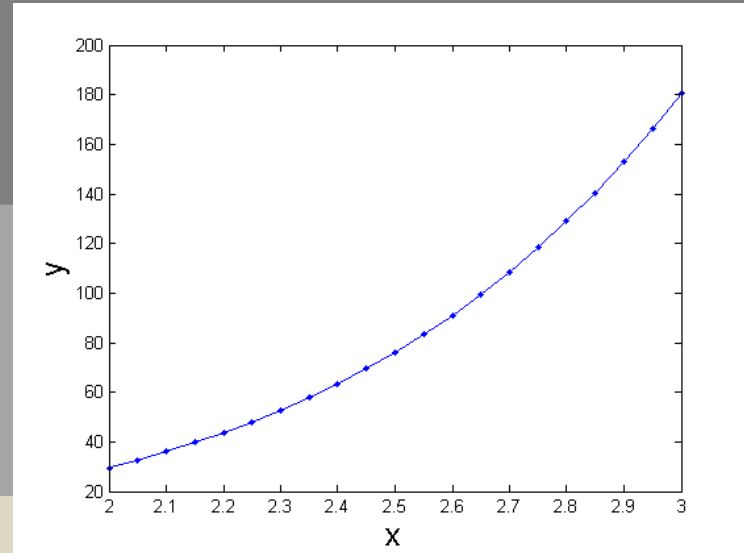
```
SumStiles = sum(X)  
SumGrammes = sum(X')
```

Προσπαθήστε να επαναλάβετε τον υπολογισμό με εντολές `for`.

Ολοκλήρωση συναρτήσεων

Ζητείται να ολοκληρωθεί η συνάρτηση
 $y = x^2 e^x$, στο διάστημα $x = 2,3$

```
dx = 0.05;  
x=2:dx:3;  
y = (x.^2).*exp(x);  
figure(34);  
plot(x,y,'b.-');  
xlabel('x','FontSize',18);  
ylabel('y','FontSize',18);  
  
olokl=0;  
for i=2:length(x)  
    olok1 = olok1 + 0.5*dx*(y(i-1)+y(i));  
end  
olokl
```



Χωρίζουμε το διάστημα x σε τραπέζια
ύψους dx και υπολογίζουμε το
εμβαδόν των τραπεζίων (κανόνας του
Τραπεζίου)

Ολοκλήρωση συναρτήσεων

Μπορείτε να εκτιμήσετε το σφάλμα του υπολογισμού;

(η αναλυτική λύση είναι: $\int x^2 e^x dx = x^2 e^x - 2x e^x + 2e^x + c$)

Να επαναληφθεί ο υπολογισμός χρησιμοποιώντας:

1. μικρότερο βήμα dx
2. τον κανόνα του Simpson

$$\begin{aligned} I &= \frac{dx}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 4f(x_{N-1}) + f(x_N)] = \\ &= \frac{dx}{3} \sum_{j=1}^{N/2} [f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})] \end{aligned}$$

χωρίζουμε το διάστημα σε N τμήματα μήκους dx , N άρτιος

Ολοκλήρωση συναρτήσεων

Το ολοκλήρωμα μπορεί επίσης να υπολογιστεί με την εντολή trapz:

$I = \text{trapz}(x,y)$

όπου x,y ζεύγη τιμών $x, f(x)$

```
dx = 0.05;  
x=2:dx:3;  
y = (x.^2) .*exp(x) ;  
  
Olokl = trapz(x,y)
```

Η εντολή trapz είναι συνάρτηση του Matlab.

Παραγωγή συναρτήσεων

Ζητείται το γράφημα της παραγώγου της συνάρτησης $y = \cos(x) + 2$

```
x=linspace(0,6*pi); % 100 αριθμοί (σε ίδια απόσταση) στο διάστημα [0,6π]  
y = cos(x)+2;
```

```
dx = diff(x) % υπολογίζει τη διαφορά  $x_{i+1} - x_i$ 
```

```
dy = diff(y) % υπολογίζει τη διαφορά  $y_{i+1} - y_i$ 
```

```
figure()
```

```
hold on; grid on;
```

```
plot(x,y,'b-') % η αρχική συνάρτηση
```

```
plot(x(1:end-1),dy./dx,'r-') % η παράγωγος  $f'(x) = dy/dx$ 
```

```
xlabel('x','FontSize',18)
```

```
ylabel('y or dy/dx','FontSize',18)
```

```
legend('y','dy/dx','FontSize',18)
```

Συμβολοσειρές

Έστω οι αλφαριθμητικές μεταβλητές (ή αλλιώς συμβολοσειρές ή **string**)

```
FirstName = 'Giwrgos'  
LastName = 'Georgiou'
```

οι μεταβλητές μπορούν να ενωθούν με την εντολή:

```
BestStudentName = [FirstName LastName]
```

Το αποτέλεσμα της παραπάνω εντολής είναι: GiwrgosGeorgiou

Για να διαχωρίσουμε όνομα και επίθετο με κενό, μπορούμε να γράψουμε:

```
BestStudentName = [FirstName ' ' LastName]
```

ή διαφορετικά:

```
BestStudentName2 = ['The best student is ' FirstName ' ' LastName]
```

Η παραπάνω συνένωση ονομάζεται concatenation και είναι μία πολύ χρήσιμη πράξη στο Matlab.

Συμβολοσειρές

Έστω η παρακάτω αλφαριθμητική μεταβλητή:

```
StudentName = 'Giwrgos Georgiou, born in 1992'
```

Για να αποθηκεύσουμε σε τρεις διαφορετικές μεταβλητές (α) το όνομα, (β) το επίθετο και (γ) την ηλικία του κ. Γιώργου Γεωργίου, χρησιμοποιούμε τις εντολές:

```
FirstName = StudentName(1:7)
LastName = StudentName(9:16)
Age = 2015 - str2num(StudentName(26:end))
```

Παρατηρήσεις:

- χρειαζόμαστε την εντολή `str2num` για να μετατρέψουμε μία συμβολοσειρά σε αριθμό, το αντίθετο επιτυγχάνεται με την εντολή `num2str`.
- οι εντολές `length(StudentName)` και `size(StudentName)` μας δίνουν το μήκος του string, δηλαδή:

```
>> length(StudentName)
ans =
    30
>> size(StudentName)
ans =
     1     30
```

Συμβολοσειρές

Η εντολή find είναι ιδιαίτερα χρήσιμη με τα strings:

```
StudentName = 'Giwrgos Georgiou, born in 1992'
```

```
ikeno = find(StudentName==' ')  
icomma = find(StudentName==',')
```

```
ikeno =  
      8      18      23      26
```

```
icomma =  
      17
```

μπορούμε έτσι να απομονώσουμε τμήματα ενός string:

```
kommati = StudentName(ikeno(1)+1:ikeno(2)-1)
```

```
kommati =  
Georgiou,          % το αποτέλεσμα είναι «Georgiou,»
```

(χρήσιμο αφού δεν έχουν όλα τα ονόματα των ανθρώπων το ίδιο μήκος χαρακτήρων)

Ερωτήσεις...