Εργαστήριο MatLab

Μήτρες ϗ΄ διανύσματα:

1 Δημιουργία

`zeros(m,n)`
: μηδενική μήτρα $\mu \times \nu$

`ones(m,n)`
: μήτρα $\mu \times \nu$ με κάθε στοιχείο $\alpha_{ij} = 1$

`eye(n)`
: μοναδιαία τετραγωνική μήτρα $\nu \times \nu$

`eye(m,n)`
: οι μ γραμμές και ν στήλες της μοναδιαίας μήτρας $n \times n$, όπου $n = \{\mu, \nu\}$

`rand(m,n)`
: $\mu \times \nu$ μήτρα τυχαίων στοιχείων ομοιόμορφα κατανεμημένων στο διάστημα $(0, 1)$

`randn(m,n)`
: $\mu \times \nu$ μήτρα τυχαίων στοιχείων κανονικά κατανεμημένων με αριθμητικό μέσο $\mu = 0$ και διακύμανση $\sigma^2 = 1$

2 Διαχείριση

2.1 Πράξεις μεταξύ μητρών

πράξη	τελεστής	πράξη	τελεστής
πρόσθεση	+	αφαίρεση	−
πολλαπλασιασμός	*	διατεταγμένος πολλαπλασιασμός	.*
αναστροφή	′	διατεταγμένη δεξιά διαίρεση	./
ύψωση	^	διατεταγμένη ύψωση	.^

2.2 Χρήσιμες συναρτήσεις

`sum( )`	`diag( )`	`inv( )`	`reshape( )`
`length( )`	`size( )`	`numel( )`	`det( )`
`triu( )`	`tril( )`	`sin( )`	`cos( )`

2.3 Δεικτοδότηση μητρών

$A(row_{index}, column_{index})$

Γραμμική προσπέλαση στοιχείων

```
A=[1 2 3;4 5 6;7 8 9];
I=[1:1:9];
B=A(I)
B =

   1    4    7    2    5    8    3    6    9
```

Προσπέλαση πολλαπλών στοιχείων

```
A([1:row_index],column_index)
% προσπέλαση στοιχείων στήλης
A(row_index,[1:column_index])
% προσπέλαση στοιχείων γραμμής
A(row_index,[1:end]) ή A(row_index,:)
% προσπέλαση γραμμής
A([1:end],column_index) ή A(:,column_index)
% προσπέλαση στήλης
```

Ασκήσεις

1. Δημιουργήστε το διάνυσμα $\vec{X}$ με στοιχεία:
   α. $2, 4, 6, 8 \ldots, 20$
   ```
   x=2:2:20
   ```
   β. $10, 8, 6, 4, 2, 0, -2, -4$
   ```
   x=10:-2:-4
   ```
   γ. $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5} \ldots$
   ```
   x=ones(1,10)./[1:10]
   ```
   δ. $0, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5} \ldots$
   ```
   x=[0:9]./[1:10]
   ```
   ε. όλους τους περιττούς αριθμούς μεταξύ 31 και 75
   ```
   x=31:2:75
   ```

2. Αν $\vec{X} = \begin{bmatrix} 2 & 5 & 1 & 6 \end{bmatrix}$

```
x=[2 5 1 6];
```

   α. Προσθέστε τον 16 σε κάθε στοιχείο
   ```
   x+16
   ```
   β. Υπολογίστε την τετραγωνική ρίζα κάθε στοιχείου
   ```
   sqrt(x)
   ```
   γ. Υπολογίστε το τετράγωνο κάθε στοιχείου
   ```
   x.^2
   ```

3. Αν $\vec{X} = \begin{bmatrix} 3 \\ 2 \\ 6 \\ 8 \end{bmatrix}$ και $\vec{Y} = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 5 \end{bmatrix}$

```
x=[3 2 6 8]';
y=[4 1 3 5]';
```

   α. Προσθέστε τα στοιχεία του $\vec{X}$ στο $\vec{Y}$
   ```
   x+y
   ```
   β. Υψώστε κάθε στοιχείο του $\vec{X}$ στην δύναμη που προσδιορίζεται από το αντίστοιχο στοιχείο του $\vec{Y}$
   ```
   x.^y
   ```
   γ. Διαιρέστε κάθε στοιχείο του $\vec{Y}$ με το αντίστοιχο στοιχείο του $\vec{X}$
   ```
   y./x
   ```

δ. Πολλαπλασιάστε κάθε στοιχείο του $\vec{X}$ με το αντίστοιχο στοιχείο του $\vec{Y}$, αποκαλώντας το αποτέλεσμα $\vec{Z}$.

```
z=x.*y
```

ε. Προσθέστε τα στοιχεία του $\vec{Z}$ και αναθέστε το αποτέλεσμα σε μια μεταβλητή $\vec{W}$

```
w=sum(z)
```

στ. Υπολογίστε το $\vec{X}'\vec{Y} - \vec{W}$.

```
x'*y-w
```

4. Δημιουργήστε ένα διάνυσμα $\vec{X}$ με στοιχεία $x_n = \dfrac{-1^{n+1}}{2n-1}$ για $n = [100]$ και εκχωρείστε το άθροισμα των στοιχείων του σε ένα νέο διάνυσμα $\vec{Y}$

```
n=1:100;
x=-1.^(n+1)./(2*n-1); % Προτεραιότητα τελεστών: ^ > /,*> -
y=sum(x)
```

5. Να πραγματοποιηθεί η αντιστροφή της σειράς των στοιχείων ενός δοσμένου διανύσματος $\vec{X}$.

```
x(length(x):-1:1)
```

6. Να κατασκευάσετε τις μήτρες

$A = [\alpha_{ij}], B = [\beta_{ij}] \in M_{15\times 15}$ όπου

$$\alpha_{ij} = \cos\left[(i+j)\cdot\pi + \frac{\pi}{2}\right]$$
$$\beta_{ij} = \sin\left[(i+j)\cdot\pi + \frac{\pi}{2}\right]$$

```
k=[1:15];
lk=length(k);
I=k'*ones(1,lk);
A=cos((I+I')*pi+pi/2);
B=sin((I+I')*pi+pi/2);
```

Να υλοποιηθεί script file που να υπολογίζει:

α. τη μήτρα $C = A^{-1}B$

β. τη μέση τιμή μ των στοιχείων των A, B

γ. τη διακύμανση σ^2 των στοιχείων του C

δ. τη μέση τιμή ανά στήλη του C

```
% α.
C=inv(A)*B; % C = A^{-1}B
% β.
```

```
% μ_A = (Σ_{i=1}^{k} Σ_{j=1}^{k} α_ij) / k
mA=sum(sum(A))/numel(A);
% ή mA=mean(mean(A));
mB=sum(sum(B))/numel(B);
% ή mB=mean(mean(B));
% γ.
%      σ^2 = (Σ_{i=1}^{k} Σ_{j=1}^{k} (c_ij − μ)^2) / k
% υπολογισμός του μ
mC=mean(mean(C));
% c_ij − μ
Cv=C-mC;
% (c_ij − μ)^2
Cv=Cv.^2;
S=sum(sum(Cv))/numel(Cv);
% ΛΑΘΟΣ: S=var(var(C));
% δ.
mc=mean(C);
```

7. Να δημιουργήσετε μήτρα $A \in M_{100\times 100}$ τα στοιχεία της οποίας είναι ομοιόμορφα κατανεμημένα στο διάστημα $[0, 10]$.

   α. Να υπολογισθεί η μέση τιμή των άρτιων γραμμών της μήτρας

   β. Να υπολογισθεί η μέση τιμή των περιττών στηλών της μήτρας

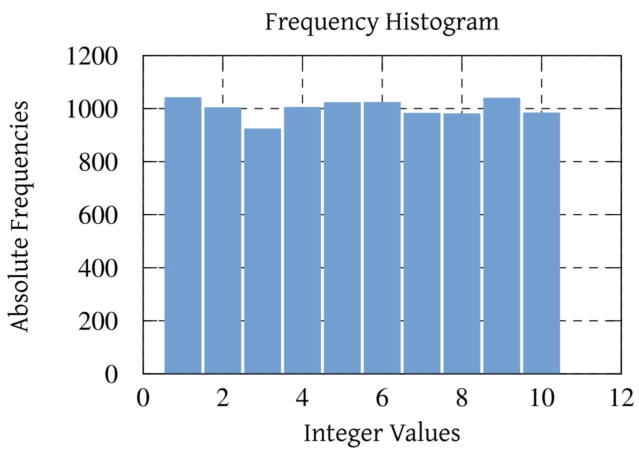
   γ. Να υπολογισθούν οι συχνότητες εμφάνισης καθενός εκ των στοιχείων του διαστήματος [0,10] και να αποθηκευθούν σε ένα νέο διάνυσμα $\vec{F}$

   δ. Να παραστήσετε γραφικά το ιστόγραμμα συχνοτήτων που αντιστοιχεί στο διάνυσμα $\vec{F}$

   ε. Να υπολογιστούν οι συχνότητες εμφάνισης των άρτιων και περιττών στοιχείων της μήτρας A

```
clear all
% διαγραφή όλων των ενεργών μεταβλητών
clc
% καθαρισμός της οθόνης
A=ceil(10*rand(100,100));
% β.
EvenRows=A([2:2:end],:);
EvenRowsMean=mean(mean(EvenRows));
% ή EvenRowsMean=sum(sum(EvenRows))/numel(EvenRows);
OddColumns=A(:,[1:2:end]);
```

```
OddColumnsMean=sum(sum(OddColumns))/numel(OddColumns);
% ή OddColumnsMean=mean(mean(OddColumns));
% γ.
V=reshape(A,1,numel(A));
% μετατροπή της μήτρας A σε διάνυσμα V⃗
F=hist(V,[1:1:10]);
% υπολογισμός του ιστογράμματος συχνοτήτων
% δ.
figure('Name','Frequency Histogram')
% απόδοση τίτλου στο διάγραμμα
bar([1:1:10],F,'b')
% κατασκευή ιστογράμματος
xlabel('Integer Values')
ylabel('Absolute Frequencies')
% απόδοση τίτλων στον οριζόντιο και κατακόρυφο άξονα
grid on
% ε.
Feven_mean=mean(F(2:2:10));
Fodd_mean=mean(F(1:2:9));
```



8. Να υπολογισθεί η μέση τιμή του αθροίσματος $\sum_{\mu=1}^{100}\sum_{\nu=1}^{100} \mu^2+\nu^2$

Έστω ότι $\mu, \nu = 3$ τότε για τον υπολογισμό του αθροίσματος θα έχουμε:

$$\sum_{\mu=1}^{3}\sum_{\nu=1}^{3} \mu^2+\nu^2 = \sum_{\mu=1}^{3} \mu^2+1^2 + \sum_{\mu=1}^{3} \mu^2+2^2 + \sum_{\mu=1}^{3} \mu^2+3^2 =$$

$$=(1^2+1^2)+(2^2+1^2)+(3^2+1^2)+(1^2+2^2)+(2^2+2^2)+ (3^2+2^2)+(1^2+3^2)+(2^2+3^2)+(3^2+3^2)=84$$

```
clc
clear all
N=100;
% 1η εκδοχή: χρήση δομής επανάληψης
tic
S =0;
for m=1:1:N
        for n=1:1:N
                S = S + m^2 + n^2;
        end;
end;
toc
S
% 2η εκδοχή: κώδικας διανυσμάτων
tic
S=0;
I=[1:1:N]'*ones(1,N);
W=I.^2 +(I').^2;
S=sum(sum(W));
toc
S
```

9. Θεωρούμε ένα γραμμικό χρονικά αναλλοίωτο σύστημα διακριτού χρόνου με κρουστική απόκριση $h[n]$, διάρκειας N, ίση με την περίοδο του σήματος εισόδου $x[n]$. Να υλοποιήσετε μια συνάρτηση MatLab *myconv.m* η οποία δεχόμενη ως είσοδο τα διανύσματα $\vec{x}$ και $\vec{h}$, μήκους N, θα επιστρέφει την έξοδο του συστήματος $\vec{y}$. Με χρήση της συνάρτησης που θα υλοποιήσετε, να υπολογίσετε το σήμα εξόδου που αντιστοιχεί στο σήμα εισόδου

$$x[n] = \{\cos\left(\pi + \frac{\pi}{64}n\right), 0 \leq n \leq N-1\}$$

με κρουστική απόκριση

$$h[n] = \{e^{\frac{n}{100}}, 0 \leq n \leq N-1\}$$

για $N = 128$
Γνωρίζουμε από τη θεωρία σημάτων και συστημάτων ότι η απόκριση ενός Γ.Χ.Α. συστήματος δίνεται από την σχέση:

$$y[n] = \sum_{k=0}^{N} x[k] \cdot h[n-k]$$

Με βάση την παραπάνω, έχουμε:

$$\begin{aligned}
y[0] &= x[0] \cdot h[0] \\
y[1] &= x[0] \cdot h[1] + x[1] \cdot h[0] \\
&\vdots \\
y[N-2] &= x[0] \cdot h[N-2] + \ldots + x[N-2] \cdot h[0] \\
y[N-1] &= x[0] \cdot h[N-1] + x[1] \cdot h[N-2] + \ldots + x[N-1] \cdot h[0]
\end{aligned}$$

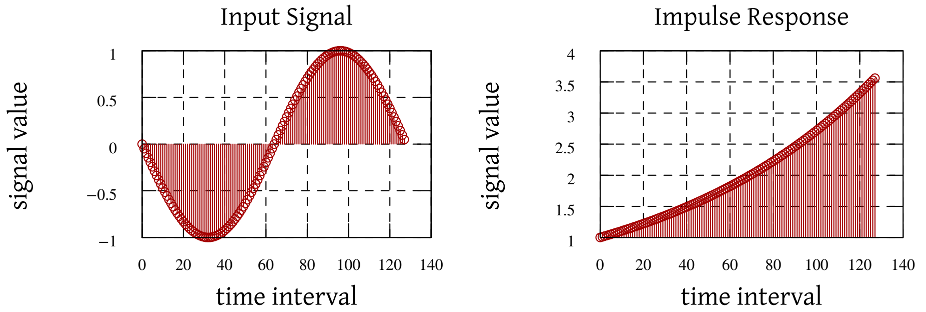
αρκεί η κατασκευή της μήτρας

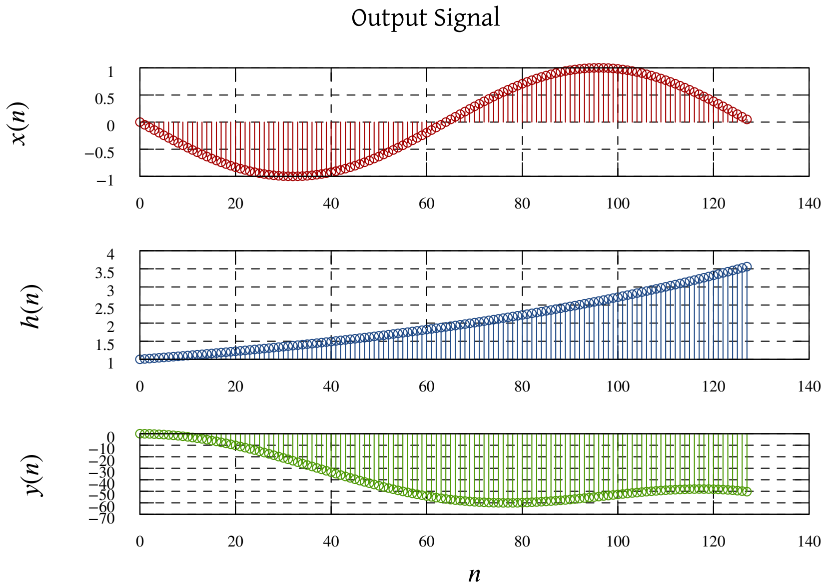
$$I = \begin{bmatrix} 1 & & & & \\ 2 & 1 & & & \\ \vdots & \vdots & \ddots & & \\ N-1 & N-2 & \cdots & 1 & \\ N & N-1 & \cdots & \cdots & 1 \end{bmatrix}$$

για τη δεικτοδότηση του διανύσματος $\vec{h}$

```
function y=myconv(x,h)
N=length(x);
n=[0:1:N-1];
tic
y=zeros(N);
% ο ορισμός της μηδενικής NxN μήτρας έξω από το βρόχο
% επιταχύνει τη διαδικασία
for m=1:N
    for k=1:m
        y(m,k)=x(k)*h(1+m-k);
    end
end
y=sum(y');
toc
% διαφορετική προσέγγιση: χρήση της μήτρας I για δεικτοδότηση
tic
I=ones(N,1)*n;
I=I'+1-I+triu(I,1);
X=ones(N,1)*x;
X=tril(X);
H=ones(N,1)*h;
H=H';
H=tril(H(I));
Y=X.*H;
y=sum(Y');
toc
```

```
% ορισμός του δείγματος
N=128;
% ορισμός του χρονικού διαστήματος
n=[0:1:N-1];
% ορισμός του σήματος εισόδου και της κρουστικής απόκρισης
x=sin(pi+(pi/64)*n);
h=exp(0.01*n);
% γραφική παράσταση του σήματος εισόδου
figure('Name','Input Signal')
stem(n,x,'-r','LineWidth',1.5);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
% γραφική παράσταση της κρουστικής απόκρισης
figure('Name','Impulse Response')
stem(n,h,'-r','LineWidth',1.8);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
% κλήση της συνάρτησης για τον υπολογισμό του σήματος εξόδου
y=myconv(x,h)
% γραφικες παραστάσεις του σήματος εισόδου, της κρουστικής
% απόκρισης και του σήματος εξόδου σε διάταξη 3x1
figure('Name','Output Signal')
subplot(3,1,1)
stem(n,x,'-r','LineWidth',1.5);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
subplot(3,1,2)
stem(n,h,'-b','LineWidth',1.8);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
subplot(3,1,3)
stem(n,y,'-g','LineWidth',1.8);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
```





10. Να γραφεί ρουτίνα MatLab η οποία θα κατασκευάζει τον τετραγωνικό πίνακα $A \in M_{n\times n}$ με στοιχεία που δίνονται από τη σχέση:
$\alpha_{rc} = (r-1)\cdot 10n + 10c$

```
N=2; %N=3;
I=ones(N,1)*[1:N];
A=(I'-1)*10*N+10*I
```

Στη συνέχεια, η ρουτίνα θα πρέπει να αντιμεταθέτει μεταξύ τους τα στοιχεία του άνω τριγωνικού και κάτω τριγωνικού υποπίνακα εξαιρώντας τα στοιχεία της κύριας διαγωνίου.

```
U=find(triu(A,1))
L=find(tril(A,-1))
A([U L])=A([L U])
```

Η ρουτίνα που περιγράφεται παρακάτω, πραγματοποιεί την αντιμετάθεση των στοιχείων του πίνακα με άξονα συμμετρίας την κύρια διαγώνιο; (όχι, π.χ. για $n = 4$)
για $n = 2$ θα πρέπει να πραγματοποιείται η εξής αντιστροφή:

$$\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} \rightarrow \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$$

για $n = 3$:

$$\begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix} \rightarrow \begin{bmatrix} 10 & 40 & 70 \\ 20 & 50 & 80 \\ 30 & 60 & 90 \end{bmatrix}$$

```
% Clear screen.
clc
% Clear all variables in the working space.
clear all
% Set the dimensionality of the square matrix.
n=3;
% Compute the number of elements in the corresponding square matrix
N=n*n;
% Set the elements of matrix M, where M is a n x n matrix whose
% elements are given by the following equation:
%              M(r,c) = (r − 1) · 10n + 10c
% Matrix M is initialy defined as a row vector.
M1=[10:10:10*N];
% Reshape the row vector M in a corresponding n x n square matrix
M1=reshape(M1,n,n)'
M2=M1;
% Get the main diagonal of matrix M.
Diag=diag(M2);
% Get the positions of the main diagonal elements in the original
% matrix M. Keep in mind that the intersect routine requires a row
% vector version  of the matrix M, thus the reshape operation is
% used in order to internally trasform matrix M into a row vector
[Diagonal,DiagonalIndices]=intersect(reshape(M2,1,N),Diag);
% Replace the main diagonal elements with zeros;
M2(DiagonalIndices)=0;
% Get the upper and lower triangle matrix corresponding to the
% original matrix M.
UpperTriangle=triu(M2);
LowerTriangle=tril(M2);
% Get the non-zero elements positions of the upper and lower
% triangle matrices.
UpperTriangleNonZeroIndices=find(UpperTriangle~=0);
LowerTriangleNonZeroIndices=find(LowerTriangle~=0);
% Get the non-zero elements of the upper and lower triangle matrices
```

```
NonZeroUpperTriangle=UpperTriangle(UpperTriangleNonZeroIndices)
NonZeroLowerTriangle=LowerTriangle(LowerTriangleNonZeroIndices)
M2(UpperTriangleNonZeroIndices)=NonZeroLowerTriangle;
M2(LowerTriangleNonZeroIndices)=NonZeroUpperTriangle;
M2(DiagonalIndices)=Diagonal;
M2
```

11. Να γραφεί συνάρτηση MatLab η οποία θα πραγματοποιεί το k fold cross validation διαμερισμό ενός δοσμένου συνόλου δεικτών.

```
function[TrainIndices,TestIndices]=kfoldIndices(N,K)
Indices=[1:1:N];
M=N/K;
if(mod(N,K)~=0)
        error('The number of elements within vector Indices must be
        fully divided by K');
else
        TrainIndices=cell(1,K);
        TestIndices=cell(1,K);
        for k=1:1:K
                test_indices=[(k-1)*M+1:1:k*M]
                train_indices=setdiff(Indices,test_indices);
                TrainIndices{k}=train_indices;
                TestIndices{k}=test_indices;
        end
end
```