

7. Προγραμματισμός Μεθόδου Άμεσης Δυσκαμψίας για Δικτυώματα

Εαρινό εξάμηνο 2023

Πέτρος Κωμοδρόμος
komodromos@ucy.ac.cy

<http://www.eng.ucy.ac.cy/petros>

Δομή γενικού προγράμματος ανάλυσης κατασκευών

- Η δομή ενός γενικού προγράμματος ανάλυσης πρέπει να είναι τόσο γενική ώστε το πρόγραμμα ανάλυσης να μπορεί να χρησιμοποιηθεί για διαφορετικά είδη αναλύσεων και με δυνατότητες χρήσης στοιχείων διαφορετικών τύπων.
- Τα στάδια της εισαγωγής δεδομένων, ανάλυσης και επεξεργασίας αποτελεσμάτων είναι κοινά σε όλα τα γενικής σκοπιμότητας (general purpose) προγράμματα ανάλυσης.
- Τα δεδομένα ενός προγράμματος ανάλυσης εισάγονται από το χρήστη και καθορίζουν το φυσικό μοντέλο και τα χαρακτηριστικά του. Καθορίζονται τα γεωμετρικά και μηχανικά χαρακτηριστικά των μελών, οι ιδιότητες των υλικών, οι συνοριακές συνθήκες και τα φορτία, τα οποία μπορεί να είναι στατικά ή δυναμικά.
- Πρέπει επίσης να καθορισθεί ο τύπος της ανάλυσης τόσο όσο αφορά τους βαθμούς ελευθερίας (π.χ. επίπεδη ή τρισδιάστατη κατασκευή), όσο και κατά πόσο πρέπει να γίνει στατική ή δυναμική ανάλυση.

- Αφού σχηματιστούν οι εξισώσεις του μαθηματικού μοντέλου, χρησιμοποιώντας κατάλληλα στοιχεία και επιβάλλοντας κατάλληλα τα φορτία, το μοντέλο επιλύεται αριθμητικά.
- Τα αποτελέσματα (π.χ. μετακινήσεις κόμβων, διαγράμματα εντατικών μεγεθών, ισοτασικές καμπύλες κ.λπ.) μπορούν να τύχουν επεξεργασία στη συνέχεια και να παρουσιαστούν με τέτοιο τρόπο που να διευκολύνουν τη λήψη αποφάσεων για τη διαστασιολόγηση της κατασκευής, ή τυχόν αλλαγές στην αρχική μόρφωση της και επαναδιαστασιολόγηση.
- Τα προγράμματα γενικού σκοπού (general purpose programs) μπορούν να χρησιμοποιηθούν για τη στατική ή και τη δυναμική ανάλυση ενός μεγάλου εύρους κατασκευών.
- Σχεδόν όλα τα προγράμματα γενικού σκοπού βασίζονται στην γενικευμένη μορφή της Μεθόδου Άμεσης Δυσκαμψίας χρησιμοποιώντας πεπερασμένα στοιχεία, όπου τα γραμμικά στοιχεία είναι μια απλή ειδική περίπτωση.

Προγραμματισμός της Μεθόδου Άμεσης Δυσκαμψίας

- Η Μέθοδος Δυσκαμψίας μπορεί εύκολα να προγραμματιστεί αφού η διαδικασία είναι γενική, συστηματική και ισχύει τόσο για ισοστατικά όσο και για υπερστατικά δικτυώματα.
- Η πιο κάτω διαδικασία ισχύει για γραμμική ελαστική ανάλυση επίπεδων δικτυωμάτων, έχοντας σαν δεδομένο ότι τόσο οι παραμορφώσεις όσο και οι μετακινήσεις μπορούν να θεωρηθούν τόσο μικρές ώστε να είναι επιτρεπτή η χρήση της αρχικής γεωμετρίας του φορέα.
- Επίσης, για χάριν απλότητας, εδώ δεν λαμβάνονται υπόψη στη συγκεκριμένη διαδικασία τυχόν θερμοκρασιακές μεταβολές ή κατασκευαστικές ατέλειες, αν και είναι εύκολο να τροποποιηθεί η διαδικασία για να συμπεριλάβει και τέτοιες δυνατότητες ανάλυσης.
- Τέλος, θα περιοριστούμε στην περίπτωση των επίπεδων δικτυωμάτων, αν και η γενίκευση του προγράμματος σε χωρικά δικτυώματα είναι σχετικά απλή.

Καθορισμός δεδομένων για ανάλυση επίπεδων δικτυωμάτων

Τα πιο κάτω βήματα μπορούν να ακολουθηθούν για να καθορισθούν τα απαραίτητα δεδομένα για την ανάλυση ενός επίπεδου δικτυώματος:

- Καθορισμός και αρίθμηση όλων των κόμβων της κατασκευής.
- Καθορισμός και αρίθμηση όλων των μελών της κατασκευής.
- Καθορισμός των τοπικών συστημάτων συντεταγμένων του κάθε μέλους της κατασκευής, δηλαδή προσδιορισμός των κόμβων αρχής (i) και τέλους (j).
- Καθορισμός ενός απόλυτου συστήματος συντεταγμένων.
- Προσδιορισμός των συντεταγμένων του κάθε κόμβου της κατασκευής χρησιμοποιώντας το απόλυτο σύστημα συντεταγμένων.
- Προσδιορισμός των συνοριακών συνθηκών στηρίξεων, δηλαδή των βαθμών ελευθερίας με δεδομένες μετακινήσεις, οι οποίες συνήθως είναι μηδενικές.
- Προσδιορισμός της συνδεσμολογίας των μελών της κατασκευής, δηλαδή των κόμβων αρχής (i) και τέλους (j) στα άκρα του κάθε μέλους.
- Καθορισμός των ιδιοτήτων του κάθε μέλους, δηλαδή για δικτυώματα της επιφάνειας διατομής A , και μέτρου ελαστικότητας E .

Αυτόματη διαδικασία ανάλυσης επίπεδων δικτυωμάτων:

- Υπολογισμός των συνημίτονων κατευθύνσεως του κάθε μέλους από τα τοπικά συστήματα συντεταγμένων στο απόλυτο σύστημα συντεταγμένων.
- Προσδιορισμός των μητρών δυσκαμψίας του κάθε μέλους, είτε κατευθείαν στο απόλυτο σύστημα συντεταγμένων, είτε χρησιμοποιώντας τα μητρώα δυσκαμψίας στα τοπικά συστήματα συντεταγμένων και τα αντίστοιχα μητρώα μετασχηματισμού.
- Έτσι, υπολογίζουμε το υπομητρώο δυσκαμψίας κάθε μέλους \underline{k}_m^{ii} , στο απόλυτο σύστημα συντεταγμένων:

$$\underline{k}_m^{ii} = \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix} \cdot \frac{AE}{L}$$

- Ξεκινώντας από ένα μητρώο δυσκαμψίας της κατασκευής, \underline{K} , με μηδενικά όλα του τα στοιχεία, για το κάθε μέλος πρέπει να προστεθούν τα εξής, λαμβάνοντας υπόψη την πιο πάνω αντιστοιχία των βαθμών ελευθερίας των μελών και της κατασκευής:
 - το μητρώο \underline{k}_m^{ii} στις γραμμές και στήλες που αντιστοιχούν στους βαθμούς ελευθερίας του κόμβου αρχής i
 - το μητρώο $\underline{k}_m^{ij} = -\underline{k}_m^{ii}$ στις γραμμές που αντιστοιχούν στους βαθμούς ελευθερίας του κόμβου αρχής i και στις στήλες που αντιστοιχούν στους βαθμούς ελευθερίας του κόμβου τέλους j
 - το μητρώο $\underline{k}_m^{ji} = -\underline{k}_m^{ii}$ στις γραμμές που αντιστοιχούν στους βαθμούς ελευθερίας του κόμβου τέλους j και στις στήλες που αντιστοιχούν στους βαθμούς ελευθερίας του κόμβου αρχής i
 - το μητρώο $\underline{k}_m^{jj} = \underline{k}_m^{ii}$ στις γραμμές και στήλες που αντιστοιχούν στους βαθμούς ελευθερίας του κόμβου τέλους j

- Με την εφαρμογή των συνοριακών συνθηκών, \underline{U}_s , ο φορέας μπορεί να γίνει σταθερός και να επιλυθεί για τα επικόμβια φορτία, \underline{R}_f :

$$\begin{bmatrix} \underline{R}_f \\ \underline{R}_s \end{bmatrix} = \begin{bmatrix} \underline{K}_{ff} & \underline{K}_{fs} \\ \underline{K}_{sf} & \underline{K}_{ss} \end{bmatrix} \cdot \begin{bmatrix} \underline{U}_f \\ \underline{U}_s \end{bmatrix}$$

- Με δεδομένες μετακινήσεις στις συνοριακές συνθήκες $\underline{U}_s = \underline{U}_s^*$ μπορούν να υπολογισθούν οι άγνωστες μετακινήσεις των κόμβων \underline{U}_f :

$$\underline{U}_s = \underline{U}_s^* \quad \Leftrightarrow \quad \underline{R}_f = \underline{K}_{ff} \cdot \underline{U}_f + \underline{K}_{fs} \cdot \underline{U}_s^* \quad \Leftrightarrow \quad \underline{U}_f = \underline{K}_{ff}^{-1} (\underline{R}_f - \underline{K}_{fs} \cdot \underline{U}_s^*)$$

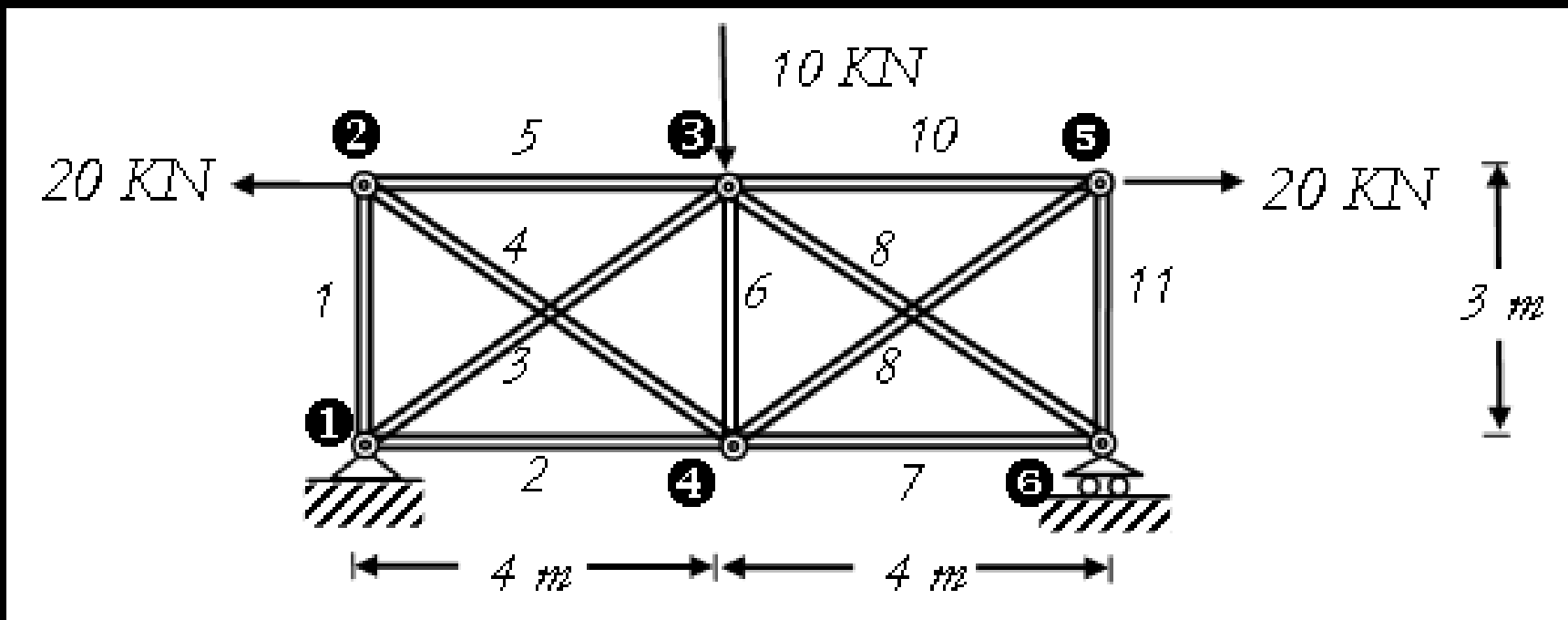
- Έχοντας υπολογίσει τις άγνωστες μετακινήσεις των αδέσμευτων βαθμών ελευθερίας μπορούν στη συνέχεια να υπολογιστούν οι άγνωστες αντιδράσεις στους βαθμούς ελευθερίας που αντιστοιχούν στις συνοριακές συνθήκες:

$$\underline{R}_s = \underline{K}_{sf} \cdot \underline{U}_f + \underline{K}_{ss} \cdot \underline{U}_s^*$$

- Στη συνέχεια μπορούν να υπολογιστούν τα εντατικά μεγέθη των μελών:

$$\underline{s}'_m = \underline{k}'_m \cdot \underline{u}'_m$$

Δεδομένα



```

%          NODES INFORMATION
% NODES(node,x(n),Y(n)) : Nodal coordinates table
% LOADS(node, magnitude, 1/2=X/Y direction) : Nodal loads table
% SUPPORTS = [node, 1/2=X/Y direction, prescribed displ/ment)
%=====

% NODAL COORDINATES
NODES = [
%  NODE      X-COORD      Y-COORD
%  n        X(n)        Y(n)
%  1          0.0          0.0
%  2          0.0          3.0
%  3          4.0          3.0
%  4          4.0          0.0
%  5          8.0          3.0
%  6          8.0          0.0  ];

```

```
LOADS = [
```

```
% NODE      LOAD      DIRECTION
```

```
%  n        (magnitude) (1:X/2:Y)
```

```
2      -20000      1
```

```
3      -10000      2
```

```
5      20000       1 ];
```

```
SUPPORTED_DOF = [
```

```
% NODE      DIRECTION      PRESCRIPED
```

```
%  n        (1:X, 2:Y)      DISPLACEMENT
```

```
1      1      0.0
```

```
1      2      0.0
```

```
6      2      0.0      ];
```

```
%=====
%
%          MEMBERS INFORMATION
% MEMBERS: Connectivity and properties table
% m: member
% [i] / m(-): start node
% [j] / m(+): end node
% E(m) : Modulus of elasticity
% A(m) : Cross section area
%=====
```

% MEMBER CONNECTIVITY TABLE

MEMBERS = [

% MEMBER START-NODE END-NODE Elasticity Modulus Cross Area

<i>%</i>	<i>m</i>	<i>[i] / m_z(-)</i>	<i>[j] / m_z(+)</i>	<i>E(m)</i>	<i>A(m)</i>	
	1	2	1	200e9	0.05	
	2	1	4	200e9	0.05	
	3	1	3	200e9	0.05	
	4	2	4	200e9	0.05	
	5	2	3	200e9	0.05	
	6	4	3	200e9	0.05	
	7	4	6	200e9	0.05	
	8	4	5	200e9	0.05	
	9	3	6	200e9	0.05	
	10	3	5	200e9	0.05	
	11	6	5	200e9	0.05] ;

Πρόγραμμα

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           2-D TRUSS ANALYSIS PROGRAM                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%=====
%           GENERAL INFORMATION
% NumNodes : Number of nodes
% NumMembers : Number of members
% NumSupportedDOF: Number of prescribed displacements
% NumLoads : Number of nodal loads
%=====

```

```

%=====
%           INFORMATION FOR NODES
% - NODES: Nodal coordinates table [NumNodes x 3]
% NODE   X-COORD   Y-COORD
%  n     X(n)     y(n)

```

```

% - LOADS: Nodal loads table [NumLoads x 3]
% NODE   LOAD     DIRECTION
%  n     (magnitude) (1:X/2:Y)

```

```

% - SUPPORTS = Supported DOFs table [NumSupportedDOF x 3]
% NODE DIRECTION PRESCRIPED
%  n     (1:X, 2:Y) DISPLACEMENT

```

```

%=====
%          MEMBERS INFORMATION
% - MEMBERS: Connectivity and properties table [NumMembers x 5]
%
% MEMBER START-NODE END-NODE Elasticity Modulus Cross Area
%   m      [i] / m(-)      [j] / m(+)      E(m)      A(m)
%=====

% CLEAR VARIABLES FROM MEMORY:
clear

% GET THE NAME OF THE INPUT FILE:
inputFile = input('Input file:','s');
fprintf('Reading data from the file: %s', inputFile)

% READING FROM INPUT FILE:
eval(inputFile)

sz = size(NODES);
NumNodes = sz(1);

sz = size(MEMBERS);
NumMembers = sz(1);

sz = size(SUPPORTED_DOF);
NumSupportedDOF = sz(1);
NumFreeDOF = 2 * NumNodes - NumSupportedDOF;

```



```

sz = size(LOADS);
NumLoads = sz(1);

fprintf('\n INPUT STATISTICS')
fprintf('\n -----')
fprintf('\n %d nodes have been read', NumNodes)
fprintf('\n %d members have been read', NumMembers)
fprintf('\n %d loads have been read', NumLoads)
fprintf('\n %d supported DOF have been read', NumSupportedDOF)
fprintf('\n --> %d free DOFs', NumFreeDOF)

% NODE DATA: Extract the provided nodal coordinates:
X = NODES(:,2) ;
Y = NODES(:,3) ;

% PLOT THE TRUSS:
figure(1)
clf
hold on
grid on

% Define the axis parameters to properly fit the truss
dx = abs(max(X)-min(X));
dy = abs(max(Y)-min(Y));
axis equal
axis([min(X)-dx/5 max(X)+dx/5 min(Y)-dy/5 max(Y)+dy/5])
axis off

```

```

% Print and plot nodes
fprintf('\n\n Data for the %d nodes that have been read:', NumNodes)
fprintf('\n   Coordinates\n Node   X   Y ')
for n=1:NumNodes
    fprintf('\n %d   %.3f %.3f', NODES(n), X(n), Y(n))
    plot(X(n), Y(n), 'o')
end

```

```

% MEMBER DATA:

```

```

fprintf('\n\n Data for the %d members that have been read:', NumMembers)
fprintf('\n M i j Length Area Elast.Mod. cos(Q) sin(Q)')

```

```

for m=1:NumMembers

```

```

    I_Node(m) = MEMBERS(m,2);

```

```

    J_Node(m) = MEMBERS(m,3);

```

```

    Xi = X(I_Node(m));

```

```

    Xj = X(J_Node(m));

```

```

    Yi = Y(I_Node(m));

```

```

    Yj = Y(J_Node(m));

```

```

    E(m) = MEMBERS(m,4);

```

```

    A(m) = MEMBERS(m,5);

```

```

    L(m) = sqrt((Xi-Xj)^2+(Yj-Yi)^2);

```

```

    DIR_COS(m) = (Xj-Xi) / L(m);

```

```

    DIR_SIN(m) = (Yj-Yi) / L(m);

```

```

    fprintf('\n %d %d %d %.2f %.3f %.2e %.3f %.3f', m, I_Node(m),

```

```

                                                J_Node(m), L(m), A(m), E(m),DIR_COS(m),DIR_SIN(m) )

```

```

    plot([Xi Xj], [Yi Yj])

```

```
% GENERATION OF GLOBAL STIFFNESS MATRIX :
```

```
% KK : Global stiffness matrix
```

```
% K : Member stiffness matrix
```

```
K = zeros(2*NumNodes,2*NumNodes) ;
```

```
for m=1:NumMembers
```

```
    Km = A(m)*E(m)/L(m)*[DIR_COS(m)*DIR_COS(m) DIR_COS(m)*DIR_SIN(m)  
                          DIR_COS(m)*DIR_SIN(m) DIR_SIN(m)*DIR_SIN(m) ];
```

```
    i = (I_Node(m)-1)*2+1;
```

```
    j = (J_Node(m)-1)*2+1;
```

```
    K(i:i+1,i:i+1) = K(i:i+1,i:i+1) + Km;
```

```
    K(i:i+1,j:j+1) = K(i:i+1,j:j+1) - Km;
```

```
    K(j:j+1,i:i+1) = K(j:j+1,i:i+1) - Km;
```

```
    K(j:j+1,j:j+1) = K(j:j+1,j:j+1) + Km;
```

```
end
```

```
fprintf('\n\n GLOBAL STIFFNESS MATRIX: K = \n ');
```

```
disp(K)
```

```
% REDUCTION OF GLOBAL STIFFNESS MATRIX (Based on the given constraints):
```

```
% NumFreeDOF: Number of free DOF
```

```
% NumSupportedDOF: Number of DOF with prescribed displacements
```

```
% --> 2 * NumNodes = NumFreeDOF + NumSupportedDOF
```

```
freeDOF = zeros(NumFreeDOF,1);
```

```
supportedDOF = zeros(NumSupportedDOF,1);
```

```

Us=zeros(NumSupportedDOF,1);
U = zeros(2*NumNodes,1);
for s=1:NumSupportedDOF
    supportedDOF(s) = 2*(SUPPORTED_DOF(s,1)-1) + SUPPORTED_DOF(s,2);
    Us(s) = SUPPORTED_DOF(s,3);
    U(supportedDOF(s)) = Us(s);
end

f = 1;
for n=1:2*NumNodes
    found=false;
    for s=1:NumSupportedDOF
        if(supportedDOF(s)==n)
            found=true;
            continue
        end
    end
    if(found==false)
        freeDOF(f) = n;
        f = f+1;
    end
end

fprintf('\nSupported degrees of freedom: ');
fprintf('\n DOF \t Prescribed displacements ');
for s=1:NumSupportedDOF
    fprintf('\n %d \t\t %.2f', supportedDOF(s), Us(s) );
end

```

```

fprintf('\n\nFree degrees of freedom: \n');
for r=1:NumFreeDOF
    fprintf(' %d \n',freeDOF(r));
end

%FORM THE Kff, Kfs, Ksf, Kss
Kff=K(freeDOF,freeDOF)
Kfs=K(freeDOF,supportedDOF)
Ksf=K(supportedDOF,freeDOF)
Kss=K(supportedDOF,supportedDOF)

R = zeros(2*NumNodes,1);

for load=1:NumLoads
    R(2*(LOADS(load,1)-1)+LOADS(load,3)) = LOADS(load,2);
end

Rf=zeros(NumFreeDOF,1);
for f=1:NumFreeDOF
    Rf(f) = R(freeDOF(f));
end

fprintf('\n Nodal loads: ');
fprintf('\n DOF \t Load [KN] ');
for f=1:NumFreeDOF
    fprintf('\n %d \t\t %.2f', freeDOF(f), Rf(f)/1000 );
end

```

```

% Computation of nodal displacements
Uf = inv(Kff) * (Rf - Kfs*Us);
fprintf('\n\n Computed nodal displacements: ');
fprintf('\n DOF \t displacement [mm] ');
for f=1:NumFreeDOF
    fprintf('\n %d \t\t %.5f', freeDOF(f), Uf(f)*1000 );
    U(freeDOF(f))= Uf(f);
end

%=====
%      BACKSUBSTITUTION PHASE
% In this stage the program will compute
% the reactions at the supports and the
% member forces from the computed nodal
% displacements.

% Computation of reaction forces
Rs = Ksf*Uf + Kss*Us;
fprintf('\n\n Computed reaction forces: ');
fprintf('\n DOF \t Force [KN] ');
for s=1:NumSupportedDOF
    fprintf('\n %d \t %.2f', supportedDOF(s), Rs(s)/1000 );
end

```

```

% Member Forces
fprintf('\n\n Computed axial member forces: ');
fprintf('\n Member      Forces [kN] ');
fprintf('\n      Start-node  End-node ');
for m=1:NumMembers
    i = (I_Node(m)-1)*2+1;
    j = (J_Node(m)-1)*2+1;
    Um = [ U(i) ; U(i+1) ; U(j) ; U(j+1) ];
    Km = [ 1 -1
           -1 1 ] * A(m)*E(m)/L(m);
    Tm = [ DIR_COS(m) DIR_SIN(m) 0 0
           0 0 DIR_COS(m) DIR_SIN(m) ];
    AxialForces(m,1:2) = Km * Tm * Um;
    fprintf('\n %d \t\t %.2f \t\t %.2f', m, AxialForces(m,1)/1000,
           AxialForces(m,2)/1000);
end

scale=100000;
for n=1:NumNodes
    plot(X(n), Y(n), 'o')
    plot(X(n)+scale*U(2*(n-1)+1), Y(n)+scale*U(2*(n-1)+2), 'or')
end

for m=1:NumMembers
    I_Node(m) = MEMBERS(m,2);
    J_Node(m) = MEMBERS(m,3);
    Xi = X(I_Node(m))+scale*U(2*(I_Node(m)-1)+1);
    Xj = X(J_Node(m))+scale*U(2*(J_Node(m)-1)+1);
    Yi = Y(I_Node(m))+scale*U(2*(I_Node(m)-1)+2);
    Yj = Y(J_Node(m))+scale*U(2*(J_Node(m)-1)+2);
    plot([X(I_Node(m)) X(J_Node(m))], [Y(I_Node(m)) Y(J_Node(m))], 'l--')
    plot([Xi Xj], [Yi Yj], 'r')
end

fprintf('\n\n Program terminates normally....\n')

```

Αποτελέσματα


```
>> trussAnalysis
```

```
Input file:truss1
```

```
Reading data from the file: truss1
```

INPUT STATISTICS

```
6 nodes have been read
```

```
11 members have been read
```

```
3 loads have been read
```

```
3 supported DOF have been read
```

```
--> 9 free DOFs
```

```
Data for the 6 nodes that have been read:
```

Coordinates

Node	X	Y
1	0.000	0.000
2	0.000	3.000
3	4.000	3.000
4	4.000	0.000
5	8.000	3.000
6	8.000	0.000

```
Data for the 11 members that have been read:
```

M	i	j	Length	Area	Elast.Mod.	cos(Q)	sin(Q)
1	2	1	3.00	0.050	2.00e+011	0.000	-1.000
2	1	4	4.00	0.050	2.00e+011	1.000	0.000
3	1	3	5.00	0.050	2.00e+011	0.800	0.600
4	2	4	5.00	0.050	2.00e+011	0.800	-0.600
5	2	3	4.00	0.050	2.00e+011	1.000	0.000
6	4	3	3.00	0.050	2.00e+011	0.000	1.000
7	4	6	4.00	0.050	2.00e+011	1.000	0.000
8	4	5	5.00	0.050	2.00e+011	0.800	0.600
9	3	6	5.00	0.050	2.00e+011	0.800	-0.600
10	3	5	4.00	0.050	2.00e+011	1.000	0.000
11	6	5	3.00	0.050	2.00e+011	0.000	1.000

```
GLOBAL STIFFNESS MATRIX: K =
```

```
1.0e+009 *
```

3.7800	0.9600	0	0	-1.2800	-0.9600	-2.5000	0	0	0	0	0
0.9600	4.0533	0	-3.3333	-0.9600	-0.7200	0	0	0	0	0	0
0	0	3.7800	-0.9600	-2.5000	0	-1.2800	0.9600	0	0	0	0
0	-3.3333	-0.9600	4.0533	0	0	0.9600	-0.7200	0	0	0	0
-1.2800	-0.9600	-2.5000	0	7.5600	0	0	0	-2.5000	0	-1.2800	0.9600
-0.9600	-0.7200	0	0	0	4.7733	0	-3.3333	0	0	0.9600	-0.7200
-2.5000	0	-1.2800	0.9600	0	0	7.5600	0	-1.2800	-0.9600	-2.5000	0
0	0	0.9600	-0.7200	0	-3.3333	0	4.7733	-0.9600	-0.7200	0	0
0	0	0	0	-2.5000	0	-1.2800	-0.9600	3.7800	0.9600	0	0
0	0	0	0	0	0	-0.9600	-0.7200	0.9600	4.0533	0	-3.3333
0	0	0	0	-1.2800	0.9600	-2.5000	0	0	0	3.7800	-0.9600
0	0	0	0	0.9600	-0.7200	0	0	0	-3.3333	-0.9600	4.0533

```
Supported degrees of freedom:
```

DOF	Prescribed displacements
1	0.00
2	0.00
12	0.00

```
Free degrees of freedom:
```

3
4
5
6
7
8
9
10
11

$K_{ff} =$

$1.0e+009 *$

3.7800	-0.9600	-2.5000	0	-1.2800	0.9600	0	0	0
-0.9600	4.0533	0	0	0.9600	-0.7200	0	0	0
-2.5000	0	7.5600	0	0	0	-2.5000	0	-1.2800
0	0	0	4.7733	0	-3.3333	0	0	0.9600
-1.2800	0.9600	0	0	7.5600	0	-1.2800	-0.9600	-2.5000
0.9600	-0.7200	0	-3.3333	0	4.7733	-0.9600	-0.7200	0
0	0	-2.5000	0	-1.2800	-0.9600	3.7800	0.9600	0
0	0	0	0	-0.9600	-0.7200	0.9600	4.0533	0
0	0	-1.2800	0.9600	-2.5000	0	0	0	3.7800

$K_{fs} =$

$1.0e+009 *$

0	0	0
0	-3.3333	0
-1.2800	-0.9600	0.9600
-0.9600	-0.7200	-0.7200
-2.5000	0	0
0	0	0
0	0	0
0	0	-3.3333
0	0	-0.9600

$K_{sf} =$

$1.0e+009 *$

0	0	-1.2800	-0.9600	-2.5000	0	0	0	0
0	-3.3333	-0.9600	-0.7200	0	0	0	0	0
0	0	0.9600	-0.7200	0	0	0	-3.3333	-0.9600

$K_{ss} =$

$1.0e+009 *$

3.7800	0.9600	0
0.9600	4.0533	0
0	0	4.0533

Nodal loads:

DOF	Load [kN]
3	-20.00
4	0.00
5	0.00
6	-10.00
7	0.00
8	0.00
9	20.00
10	0.00
11	0.00

Computed nodal displacements:

DOF	displacement [mm]
3	-0.00533
4	-0.00125
5	0.00045
6	-0.00178
7	0.00045
8	0.00071
9	0.00624
10	-0.00125
11	0.00091

Computed reaction forces:

DOF	Force [KN]
1	-0.00
2	5.00
12	5.00

Computed axial member forces:

Member	Forces [KN]	
	Start-node	End-node
1	4.15	-4.15
2	-1.13	1.13
3	1.42	-1.42
4	-6.92	6.92
5	-14.47	14.47
6	8.30	-8.30
7	-1.13	1.13
8	-6.92	6.92
9	1.42	-1.42
10	-14.47	14.47
11	4.15	-4.15

Program terminates normally....

