# *9. Προγραμματισμός Δυναμικής Ανάλυσης ΠΒΣ – Απευθείας Ολοκλήρωση*

**Εαρινό εξάμηνο 2024**

***Πέτρος Κωμοδρόμος***
**komodromos@ucy.ac.cy**

**http://www.eng.ucy.ac.cy/petros**

# Θέματα

- Δυναμική ανάλυση με απευθείας ολοκλήρωση των ΔΕ κίνησης
  - Μέθοδος Κεντρικής Διαφοράς για δυναμική ανάλυση ΠΒΣ
  - Μέθοδος *Newmark* για δυναμική ανάλυση ΠΒΣ

# Δυναμική Ανάλυση με απευθείας ολοκλήρωση των ΔΕ κίνησης

- Το σύστημα των ΔΕ που χαρακτηρίζουν την κίνηση ενός ΠΒΣ μπορεί να επιλυθεί αριθμητικά απευθείας χωρίς οποιοδήποτε μετασχηματισμό.
- Οι μέθοδοι που είχαμε δει για την αριθμητική ολοκλήρωση ΜΒΣ, μπορούν να γενικευτούν για την κατευθείαν αριθμητική ολοκλήρωση των ΔΕ κίνησης, για αυτό και ονομάζονται μέθοδοι απευθείας ολοκλήρωσης (*direct integration methods*) αφού κανένας μετασχηματισμός των εξισώσεων δεν χρησιμοποιείται.

# Μέθοδος κεντρικής διαφοράς για ΠΒΣ

- Χρησιμοποιώντας τις εξής σχέσεις για τις ταχύτητες και επιταχύνσεις:

$$\underline{\dot{u}}(t) = \frac{\underline{u}(t+\Delta t) - \underline{u}(t-\Delta t)}{2\cdot\Delta t}$$

$$\underline{\ddot{u}}(t) = \frac{\underline{u}(t+\Delta t) - 2\cdot\underline{u}(t) + \underline{u}(t-\Delta t)}{\Delta t^2}$$

- Αντικαθιστώντας τις πιο πάνω σχέσεις στο σύστημα ΔΕ:

$$\underline{M}\cdot\underline{\ddot{u}}(t) + \underline{C}\cdot\underline{\dot{u}}(t) + \underline{K}\cdot\underline{u}(t) = \underline{P}(t)$$

προκύπτει η πιο κάτω μητρωική σχέση:

$$\left(\frac{1}{\Delta t^2}\underline{M} + \frac{1}{2\Delta t}\underline{C}\right)u(t+\Delta t) = \underline{P}(t) - \left(\underline{K} - \frac{2}{\Delta t^2}\underline{M}\right)u(t) - \left(\frac{1}{\Delta t^2}\underline{M} - \frac{1}{2\Delta t}\underline{C}\right)u(t-\Delta t)$$

$$\left(\frac{1}{\Delta t^2}\underline{M}+\frac{1}{2\Delta t}\underline{C}\right)u(t+\Delta t)=\underline{P}(t)-\left(\underline{K}-\frac{2}{\Delta t^2}\underline{M}\right)u(t)-\left(\frac{1}{\Delta t^2}\underline{M}-\frac{1}{2\Delta t}\underline{C}\right)u(t-\Delta t)$$

- Οι μόνοι άγνωστοι στην πιο πάνω σχέση είναι οι μετακινήσεις $\underline{u}(t+\Delta t)$ οι οποίες μπορούν να προσδιοριστούν από την επίλυση του συστήματος γραμμικών εξισώσεων: $\hat{\underline{K}}\cdot\underline{u}(t+\Delta t)=\hat{\underline{P}}(t)$

- όπου $\hat{\underline{K}}=\frac{1}{\Delta t^2}\underline{M}+\frac{1}{2\cdot\Delta t}\underline{C}$

$$\hat{\underline{P}}(t)=\underline{P}(t)-\left(\underline{K}-\frac{2}{\Delta t^2}\underline{M}\right)\underline{u}(t)-\left(\frac{1}{\Delta t^2}\underline{M}-\frac{1}{2\Delta t}\underline{C}\right)\underline{u}(t-\Delta t)$$

- Για να μπορεί να ξεκινήσει η μέθοδος, πρέπει να υπολογιστούν οι μετακινήσεις στην χρονική στιγμή $(-\Delta t)$ η οποία μπορεί να αποδειχτεί, βάσει των παραδοχών που έγιναν ότι ισούται με:

$$\underline{u}(-\Delta t)=\underline{u}(0)-\Delta t\cdot\dot{\underline{u}}(0)+\frac{\Delta t^2}{2}\cdot\ddot{\underline{u}}(0)$$

- Οι επιταχύνσεις στην χρονική στιγμή $t = 0$ μπορούν να υπολογιστούν από την αντίστοιχη εξίσωση κίνησης σε εκείνη τη χρονική στιγμή με δεδομένες τις αρχικές μετακινήσεις και ταχύτητες:

$$\ddot{\underline{u}}(0) = \underline{M}^{-1} \cdot \left( \underline{P}(0) - \underline{C} \cdot \dot{\underline{u}}(0) - \underline{K} \cdot \underline{u}(0) \right)$$

- Για να είναι ευσταθής η μέθοδος, πρέπει το χρονικό βήμα $\Delta t$ της αριθμητικής ολοκλήρωσης να είναι μικρότερο από την κρίσιμη τιμή

$$\Delta t \le \Delta t_{cr} = \frac{T_{min}}{\pi} = \frac{2}{\omega_{max}}$$

όπου $T_{min}$ είναι η μικρότερη ιδιοπερίοδος και $\omega_{max}$ η μεγαλύτερη ιδιοσυχνότητα της κατασκευής.

**TABLE 9.1** *Step-by-step solution using central difference method (general mass and damping matrices)*

A. *Initial calculations:*

1. Form stiffness matrix $\mathbf{K}$, mass matrix $\mathbf{M}$, and damping matrix $\mathbf{C}$.
2. Initialize ${}^0\mathbf{U}$, ${}^0\dot{\mathbf{U}}$, and ${}^0\ddot{\mathbf{U}}$.
3. Select time step $\Delta t$, $\Delta t \le \Delta t_{cr}$, and calculate integration constants:

$$a_0 = \frac{1}{\Delta t^2}; \qquad a_1 = \frac{1}{2\,\Delta t}; \qquad a_2 = 2a_0; \qquad a_3 = \frac{1}{a_2}$$

4. Calculate ${}^{-\Delta t}\mathbf{U} = {}^0\mathbf{U} - \Delta t\,{}^0\dot{\mathbf{U}} + a_3\,{}^0\ddot{\mathbf{U}}$.
5. Form effective mass matrix $\hat{\mathbf{M}} = a_0\mathbf{M} + a_1\mathbf{C}$.
6. Triangularize $\hat{\mathbf{M}}$: $\hat{\mathbf{M}} = \mathbf{L}\mathbf{D}\mathbf{L}^T$.

B. *For each time step:*

1. Calculate effective loads at time $t$:

$${}^t\hat{\mathbf{R}} = {}^t\mathbf{R} - (\mathbf{K} - a_2\mathbf{M})\,{}^t\mathbf{U} - (a_0\mathbf{M} - a_1\mathbf{C})\,{}^{t-\Delta t}\mathbf{U}$$

2. Solve for displacements at time $t + \Delta t$:

$$\mathbf{L}\mathbf{D}\mathbf{L}^T\,{}^{t+\Delta t}\mathbf{U} = {}^t\hat{\mathbf{R}}$$

3. If required, evaluate accelerations and velocities at time $t$:

$${}^t\ddot{\mathbf{U}} = a_0({}^{t-\Delta t}\mathbf{U} - 2\,{}^t\mathbf{U} + {}^{t+\Delta t}\mathbf{U})$$

$${}^t\dot{\mathbf{U}} = a_1(-{}^{t-\Delta t}\mathbf{U} + {}^{t+\Delta t}\mathbf{U})$$

# Μέθοδος Newmark για ΠΒΣ

- Η αριθμητική ολοκλήρωση του συστήματος ΔΕ με τη μέθοδο *Newmark* βασίζεται στις πιο κάτω προσεγγιστικές σχέσεις για τις ταχύτητες και μετακινήσεις των δυναμικών ΒΕ:

$$\underline{\dot{u}}(t+\Delta t)=\underline{\dot{u}}(t)+\frac{\left(\underline{\ddot{u}}(t+\Delta t)+\underline{\ddot{u}}(t)\right)}{2}\cdot\Delta t$$

$$\underline{u}(t+\Delta t)=\underline{u}(t)+\underline{\dot{u}}(t)\cdot\Delta t+\frac{\underline{\ddot{u}}(t+\Delta t)+\underline{\ddot{u}}(t)}{2}\cdot\Delta t^2$$

- Με δεδομένη τη λύση μέχρι τη χρονική στιγμή $t$, λύνοντας τη 2η σχέση ως προς $\underline{\ddot{u}}(t+\Delta t)$, το οποίο στη συνέχεια αντικαθιστούμε στην 1η σχέση , έχουμε δύο σχέσεις οι οποίες εκφράζουν τις ταχύτητες $\underline{\dot{u}}(t+\Delta t)$ και επιταχύνσεις $\underline{\ddot{u}}(t+\Delta t)$, συναρτήσει του $\underline{u}(t+\Delta t)$ και άλλα γενικά μεγέθη.
- Αντικαθιστώντας τις σχέσεις για τις ταχύτητες $\underline{\dot{u}}(t+\Delta t)$ και επιταχύνσεις $\underline{\ddot{u}}(t+\Delta t)$ στο σύστημα ΔΕ κίνησης για τη χρονική στιγμή $t+\Delta t$ :

$$\underline{M}\cdot\underline{\ddot{u}}(t+\Delta t)+\underline{C}\cdot\underline{\dot{u}}(t+\Delta t)+\underline{K}\cdot\underline{u}(t+\Delta t)=\underline{P}(t+\Delta t)$$

From: *Finite Element Procedures*, K.J. Bathe

**TABLE 9.4** *Step-by-step solution using Newmark integration method*

A. *Initial calculations:*

1. Form stiffness matrix $\mathbf{K}$, mass matrix $\mathbf{M}$, and damping matrix $\mathbf{C}$.
2. Initialize ${}^0\mathbf{U}$, ${}^0\dot{\mathbf{U}}$, and ${}^0\ddot{\mathbf{U}}$.
3. Select time step $\Delta t$ and parameters $\alpha$ and $\delta$ and calculate integration constants:

$$\delta \geq 0.50; \qquad \alpha \geq 0.25(0.5 + \delta)^2$$

$$a_0 = \frac{1}{\alpha \Delta t^2}; \qquad a_1 = \frac{\delta}{\alpha \Delta t}; \qquad a_2 = \frac{1}{\alpha \Delta t}; \qquad a_3 = \frac{1}{2\alpha} - 1;$$

$$a_4 = \frac{\delta}{\alpha} - 1; \qquad a_5 = \frac{\Delta t}{2}\left(\frac{\delta}{\alpha} - 2\right); \qquad a_6 = \Delta t(1 - \delta); \qquad a_7 = \delta \Delta t$$

4. Form effective stiffness matrix $\hat{\mathbf{K}}$: $\hat{\mathbf{K}} = \mathbf{K} + a_0\mathbf{M} + a_1\mathbf{C}$.
5. Triangularize $\hat{\mathbf{K}}$: $\hat{\mathbf{K}} = \mathbf{L}\mathbf{D}\mathbf{L}^T$.

B. *For each time step:*

1. Calculate effective loads at time $t + \Delta t$:

$${}^{t+\Delta t}\hat{\mathbf{R}} = {}^{t+\Delta t}\mathbf{R} + \mathbf{M}(a_0\, {}^t\mathbf{U} + a_2\, {}^t\dot{\mathbf{U}} + a_3\, {}^t\ddot{\mathbf{U}}) + \mathbf{C}(a_1\, {}^t\mathbf{U} + a_4\, {}^t\dot{\mathbf{U}} + a_5\, {}^t\ddot{\mathbf{U}})$$

2. Solve for displacements at time $t + \Delta t$:

$$\mathbf{L}\mathbf{D}\mathbf{L}^T\, {}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\hat{\mathbf{R}}$$

3. Calculate accelerations and velocities at time $t + \Delta t$:

$${}^{t+\Delta t}\ddot{\mathbf{U}} = a_0({}^{t+\Delta t}\mathbf{U} - {}^t\mathbf{U}) - a_2\, {}^t\dot{\mathbf{U}} - a_3\, {}^t\ddot{\mathbf{U}}$$

$${}^{t+\Delta t}\dot{\mathbf{U}} = {}^t\dot{\mathbf{U}} + a_6\, {}^t\ddot{\mathbf{U}} + a_7\, {}^{t+\Delta t}\ddot{\mathbf{U}}$$

- Μετά από κάποιες πράξεις και διαχωρίζοντας τα άγνωστα μεγέθη από τα γνωστά, προκύπτει μια σχέση της μορφής: $\hat{\underline{K}} \cdot \underline{u}(t+\Delta t) = \hat{\underline{P}}(t+\Delta t)$

όπου
$$\hat{\underline{K}} = \underline{K} + a_0 \cdot \underline{M} + a_1 \cdot \underline{C}$$

$$\hat{\underline{P}}(t+\Delta t) = \underline{P}(t+\Delta t) + \underline{M} \cdot \left(a_0 \cdot \underline{u}(t) + a_2 \cdot \dot{\underline{u}}(t) + a_3 \cdot \ddot{\underline{u}}(t)\right)$$
$$+\underline{C} \cdot \left(a_1 \cdot \underline{u}(t) + a_4 \cdot \dot{\underline{u}}(t) + a_5 \cdot \ddot{\underline{u}}(t)\right)$$

όπου
$$a_0 = \frac{4}{\Delta t^2},\ a_1 = \frac{2}{\Delta t},\ a_2 = \frac{4}{\Delta t},\ \ a_3 = a_4 = 1,\ a_5 = 0,\ a_6 = a_7 = \frac{\Delta t}{2}$$

- Έτσι οι μετακινήσεις στη χρονική στιγμή $t+\Delta t$ μπορούν να υπολογιστούν σαν:

$$\underline{u}(t+\Delta t) = \hat{\underline{K}}^{-1} \cdot \hat{\underline{P}}(t+\Delta t)$$

Στη συνέχεια οι επιταχύνσεις και ταχύτητες σε χρόνο μπορούν να προσδιοριστούν από τις πιο κάτω σχέσεις:

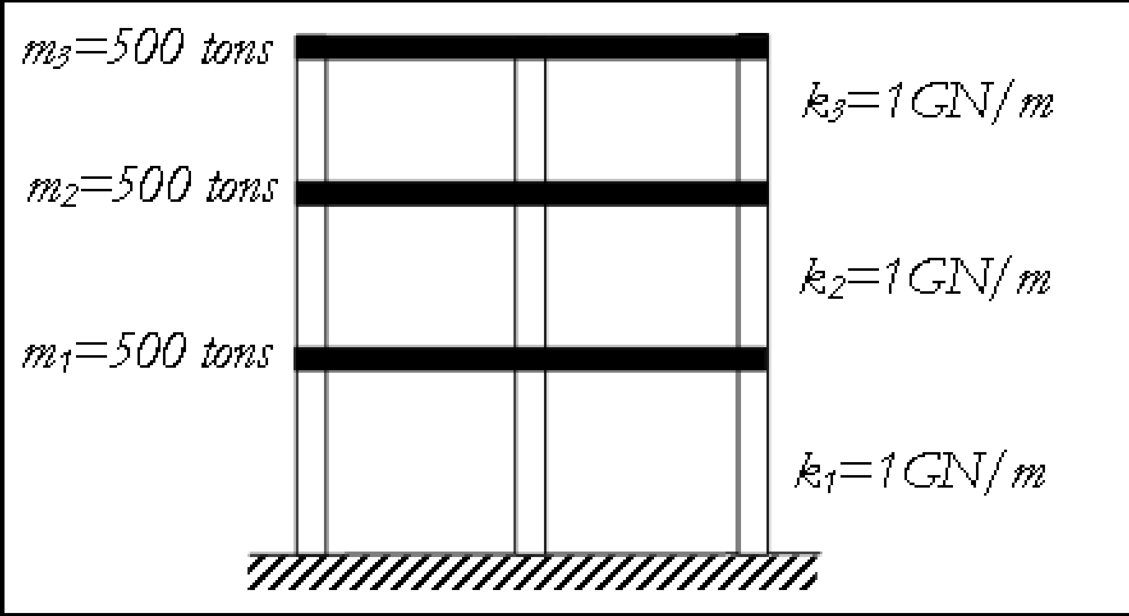
$$\ddot{\underline{u}}(t+\Delta t) = a_0 \cdot \left(\underline{u}(t+\Delta t) - \underline{u}(t)\right) - a_2 \cdot \dot{\underline{u}}(t) - a_3 \cdot \ddot{\underline{u}}(t)$$

$$\dot{\underline{u}}(t+\Delta t) = \dot{\underline{u}}(t) + a_6 \cdot \ddot{\underline{u}}(t) + a_7 \cdot \ddot{\underline{u}}(t+\Delta t)$$

# Προγραμματισμός απευθείας αριθμητικής ολοκλήρωσης

- Με την απευθείας αριθμητική ολοκλήρωση, είτε με τη Μέθοδο Κεντρικής Διαφοράς είτε με τη Μέθοδο *Newmark*, εκτελείται δυναμική ανάλυση ενός ΠΒΣ για το οποίο έχουμε εισάγει δεδομένα.

- Υπολογίζονται για την κάθε μάζα, δηλαδή τον κάθε όροφο, οι μέγιστες σχετικές μετακινήσεις (*relative displacement*s), οι μέγιστες διαφορικές μετακινήσεις (*interstory deflections*) μεταξύ των ορόφων, οι μέγιστες απόλυτες επιταχύνσεις των ορόφων, καθώς και οι μέγιστες τέμνουσες ορόφων.



```
clear

% Fortwsh epitaxynsewn edafoys
load AthensAccel
t =  AthensAccel(:,1);
ag = AthensAccel(:,2);

% k(i) stiffness of i-story
k(1)=1000e6;
k(2)=1000e6;
k(3)=1000e6;

% m(i) mass of i-floor
m(1)= 500e3 ;
m(2)= 500e3 ;
m(3)= 500e3 ;

% damping ratio for first and last modes
zFirst= 0.05;
zLast= 0.05;

% N number of dynamic DOFs
NDOF = length(k);
```

## ApeutheiasOloklirosi.m

```
clear
Input_Data
fprintf('\n Number of dynamic DOF: %d \n', NDOF)

%%%%%%%%%%%%%% Setting the stiffness and mass matrices %%%%%%%%%%%%%%%%%%%%%
K = zeros(NDOF,NDOF);
M = zeros(NDOF,NDOF);
for i=1:NDOF
  K(i,i) = k(i);
  M(i,i) = m(i);
  if i>1
      K(i,i-1) = K(i,i-1) - k(i);
      K(i-1,i) = K(i-1,i) - k(i);
      K(i-1,i-1) = K(i-1,i-1) + k(i);
  end
end

printMatrix(M,'Mass')
printMatrix(K,'Stiffness')
```

```
%%%%%%%%%  Ypologismos idiotimwn kai idiomorfwn  %%%%%%%%%%%%
[V,D] = eig(K,M);

for i=1:NDOF
   w(i) = sqrt(D(i,i));
   T(i) = 2*pi/w(i);
   modes(:,i) = V(:,i);
end

rayleighCoeff = inv( [ 1/w(1)  w(1)
                1/w(NDOF)  w(NDOF) ] ) * [2*zFirst ; 2*zLast];
C = rayleighCoeff(1)*M + rayleighCoeff(2) * K;
printMatrix(C,'Damping')

fprintf('\n\n Mode   Eigenperiod     Eigenfrequency   Cyclic Eigenfrequency')
for i=1:NDOF
   fprintf('\n   %d    %.4f sec     %.3f rad/sec       %.3f Hz', i, T(i), w(i), 1/T(i));
   modes(:,i) = V(:,i);
end
```

```
fprintf('\n\n Select the method to use:')
fprintf('\n    [1] Central Difference Method')
fprintf('\n    [2] Newmark Method \n')
selection = input('    Method: ');

%%%%%%%%%%%% Computing PGA %%%%%%%%%%%%%%%%%%%%
dt = t(2)-t(1);
numberPoints = length(t);
g=9.81;
pga=abs(ag(1));
tPGA=t(1);

for i=2:numberPoints
    if pga < abs(ag(i))
      pga=abs(ag(i));
      tPGA=t(i);
    end
 end
fprintf('\n Exoyn diabastei %d times epitaxynsewn edafoys ana %.3f sec',
                                                            numberPoints, dt)
fprintf('\n Megisth epitaxynsh edafoys = %.2f m/sec^2 = %.3f g sta %.3f sec\n', pga,
                                                                   pga/g, tPGA)
```

```
u0 = zeros(NDOF,1);
v0 = zeros(NDOF,1);
P = -M * ones(NDOF,1) * ag';

%%%%%%%%%%%% Method Selection %%%%%%%%%%%%%%%%%%%%%%
if selection==1
   fprintf('\n Using the Central Difference Method')
   [U,V,A] = CDM(numberPoints, dt, t, P, K, M, C, u0, v0, w(NDOF), NDOF);
else if selection==2
        fprintf('\n Using the Newmark Method')
        [U,V,A] =NEWMARK(numberPoints,dt,t,P,K,M,C,u0, v0);


    end
end


xMin = 0;
xMax = 30;
yMin = -0.20;
yMax = 0.20;
```

```
%%%%%%%%%%%%%%   Plot displacements  %%%%%%%%%%%%%%%%%%%%%%%%
figureNumber=1;
figure(figureNumber)
clf
orient tall
for floor=1:NDOF
    subplot(NDOF,1,NDOF+1-floor)
    plot(t,U(floor,:))
    grid on
    xlabel('Time, t [sec]')
    s = sprintf('U%d [m]',floor);
    ylabel(s)
    axis([xMin xMax yMin yMax])
end
```

```
%%%%%%%%%%%%  Plot interstory deflections  %%%%%%%%%%%%%%%%%%%
figureNumber=figureNumber+1;
figure(figureNumber)
clf
orient tall
for floor=1:NDOF
    subplot(NDOF,1,NDOF+1-floor)
    if(floor>1)
       dU(floor,:) = U(floor,:)-U(floor-1,:);
    else
       dU(floor,:) = U(floor,:);
    end
     plot(t,dU(floor,:))
    grid on
    xlabel('Time, t [sec]')
    s = sprintf('dU%d [m]',floor);
    ylabel(s)
    axis([xMin xMax yMin/2 yMax/2])
end
```

```
%%%%%%%%%%%%%   Plot accelogram   %%%%%%%%%%%%%%%%%%%%
figureNumber=figureNumber+1;
figure(figureNumber)
clf
orient tall
subplot(NDOF+1,1,NDOF+1)
plot(t,ag)
grid on
xlim([0 30])
xlabel('Time, t [sec]')
ylabel('Ag [m/s^2]')
```

```
%%%%%%%%%%%   Plot Absolute Accelerations  %%%%%%%%%%%%%%
for floor=1:NDOF
    subplot(NDOF+1,1,NDOF+1-floor)
    plot(t,A(floor,:)'+ag)
    hold on
    grid on
    xlabel('Time, t [sec]')
    s = sprintf('A%dtot [m/s^2]',floor);
    ylabel(s)
    xlim([0 30])
end

for floor=1:NDOF
    shearForces(floor,:) = k(floor) * dU(floor,:);
end
maxShearForce = max(max(abs(shearForces)));
```

```
%%%%%%%%%%%%%   Plot shear forces   %%%%%%%%%%%%%%%%%%%%%%
figureNumber=figureNumber+1;
figure(figureNumber)
clf
orient tall
for floor=1:NDOF
    subplot(NDOF,1,NDOF+1-floor)
    plot(t,shearForces(floor,:)/1e6)
    grid on
    xlabel('Time, t [sec]')
    s = sprintf('V%d [MN]',floor);
    ylabel(s)
    axis([xMin xMax -maxShearForce/1e6 maxShearForce/1e6])
end

fprintf('\n\n *********  Response Peak Values **********');
fprintf('\n Floor  Displacements   Interstory deflections    Total accelerations    Shear forces');
```

```
for floor=1:NDOF
    uMax = 100*max(abs(U(floor,:)));
    duMax = 100*max(abs(dU(floor,:)));
    aTotMax = max(abs(A(floor,:)'+ag));
    shearMax = max(abs(shearForces(floor,:)))/1e6;
    fprintf('\n  %d       %.2f cm          %.3f cm           %9.4f m/sec^2       %.3f MN',
                                                   floor, uMax, duMax, aTotMax, shearMax);
end
fprintf('\n')
```

CDM.m

```
function [U,V,A]=CDM(nPoints, dt, t, P, K, M, C, u0, v0, wN, NDOF)
% Me8odos Kentrikhs diaforas


  checkStability(dt,wN)


  % Arxikes syn8hkes
  U(:,1) = u0;
  V(:,1) = v0;
  a0 = inv(M) * (P(:,1) - C*v0 - K*u0);
  A(:,1) = a0;


  % Ypologismos syntelestwn
  c0 = 1/dt/dt;
  c1 = 0.5/dt;
  c2 = 2*c0;
  c3 = 1/c2;


  kEff = c0*M + c1*C;
  pEff_1 = K - c2*M;
  pEff_2 = c0*M - c1*C;
```

```
up = u0;
upp = up - dt*v0 + (0.5*dt*dt)*a0;

for i=1:nPoints-1
  pEff = P(:,i) - pEff_1*up - pEff_2*upp;
  u = inv(kEff)*pEff;
  U(:,i+1) = u;
  A(:,i) = c0 * (u - 2*up + upp);
  V(:,i) = c1 * (u - upp) ;
  upp = up;
  up = u;
end

pEff = P(nPoints) - pEff_1*U(:,i+1) - pEff_2*U(:,i);
uNext = inv(kEff) * pEff;
A(:,nPoints) = c0 * (uNext-2*up+upp);
V(:,nPoints) = c1 * (uNext-upp);

return
```

# NEWMARK.m

```
function [U,V,A]=NEWMARK(n,dt,t,P,K,M,C,u0,v0)
 % Me8odos Newmark


a=0.25;
d=0.5;


 % Arxikes syn8hkes
U(:,1)=u0;
V(:,1)=v0;
A(:,1)=inv(M)*(P(:,1)-C*V(:,1)-K*U(:,1));

% Ypologismos syntelestwn alpha
a0=1/(a*dt^2);
a1=d/(a*dt);
a2=1/(a*dt);
a3=(1/(2*a))-1;
a4=(d/a)-1;
a5=dt/2*((d/a)-2);
a6=dt*(1-d);
a7=d*dt;
```

```
K_hat=K+a0*M+a1*C;


for i=1:n-1
    Phat=P(:,i+1)+M*(a0*U(:,i)+a2*V(:,i)+a3*A(:,i))+C*(a1*U(:,i)+a4*V(:,i)+a5*A(:,i));
    U(:,i+1)=inv(K_hat)*Phat;
    A(:,i+1)=a0*(U(:,i+1)-U(:,i))-a2*V(:,i)-a3*A(:,i);
    V(:,i+1)=V(:,i)+a6*A(:,i)+a7*A(:,i+1);
end
```

*Output:*

```
>> ApeutheiasOloklirosi

 Number of dynamic DOF: 3

********** Mass Matrix ****************
  5e+005  0  0
  0  5e+005  0
  0  0  5e+005

********** Stiffness Matrix ****************
  2e+009  -1e+009  0
  -1e+009  2e+009  -1e+009
  0  -1e+009  1e+009

********** Damping Matrix ****************
  2.79e+006  -9.95e+005  0
  -9.95e+005  2.79e+006  -9.95e+005
  0  -9.95e+005  1.79e+006
```

| Mode | Eigenperiod | Eigenfrequency | Cyclic Eigenfrequency |
|---|---|---|---|
| 1 | 0.3157 sec | 19.903 rad/sec | 3.168 Hz |
| 2 | 0.1127 sec | 55.767 rad/sec | 8.876 Hz |
| 3 | 0.0780 sec | 80.585 rad/sec | 12.826 Hz |

Select the method to use:

[1] Central Difference Method

[2] Newmark Method

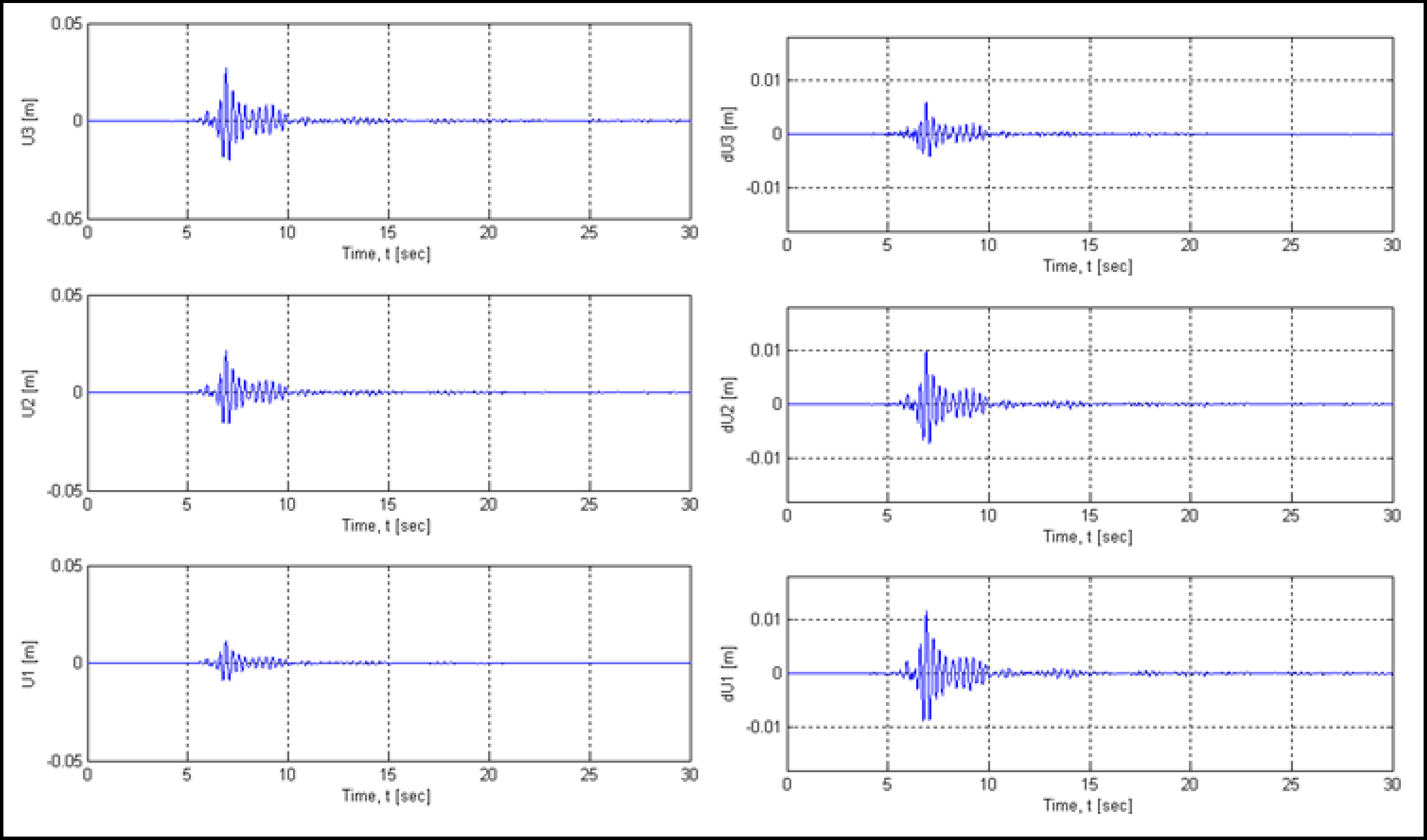
Method: 1

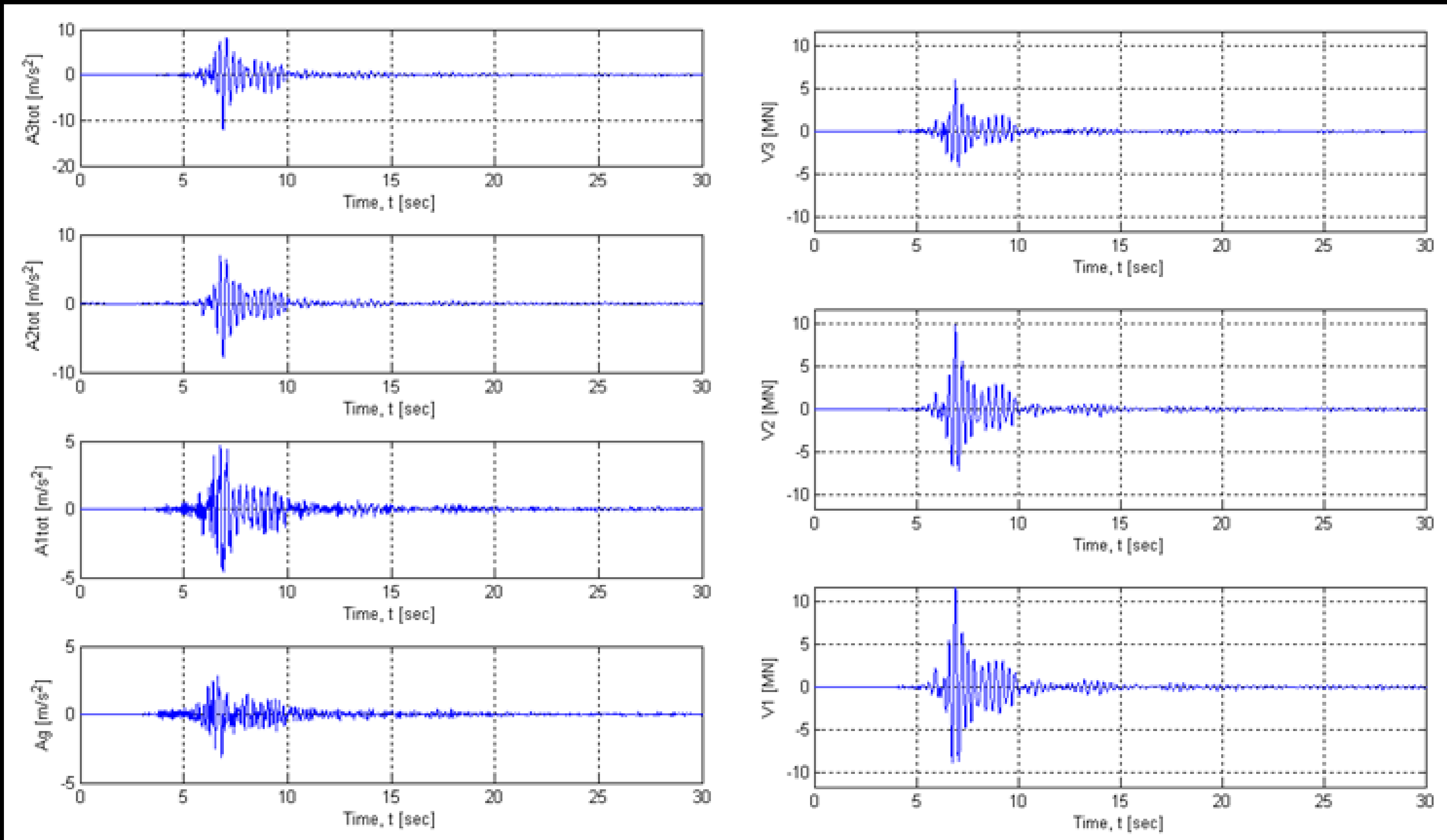
Exoyn diabastei 4603 times epitaxynsewn edafoys ana 0.010 sec
Megisth epitaxynsh edafoys = 3.20 m/sec^2 = 0.326 g sta 6.820 sec

Using the Central Difference Method

********* Response Peak Values **********

| Floor | Displacements | Interstory deflections | Total accelerations | Shear forces |
|---|---|---|---|---|
| 1 | 1.16 cm | 1.162 cm | 4.7155 m/sec^2 | 11.617 MN |
| 2 | 2.14 cm | 0.976 cm | 7.9564 m/sec^2 | 9.765 MN |
| 3 | 2.72 cm | 0.594 cm | 12.0674 m/sec^2 | 5.942 MN |


U3 [m]
Time, t [sec]
dU3 [m]
Time, t [sec]
U2 [m]
Time, t [sec]
dU2 [m]
Time, t [sec]
U1 [m]
Time, t [sec]
dU1 [m]
Time, t [sec]


A3tot [m/s$^2$]
A2tot [m/s$^2$]
A1tot [m/s$^2$]
Ag [m/s$^2$]
V3 [MN]
V2 [MN]
V1 [MN]
Time, t [sec]