```
*****************************************************************
```

M A T L A B in D Y N A M I C S

```
*****************************************************************
```

Here we present some MATLAB programs which solve the second problem of Dynamics:

- Derive differential equations, resolve them analytically or numerically, and plot the relative graphics.
- There is no necessity to write a procedure-function for the derivatives! It is generated automatically.
- You can choose in-line the most proper Solver to your problem.
- All the data can be input from data files or in-line mode.
- You can run these programs repeatedly with different values of some parameters p1, p2, ..., and ini-conditions without their restart.
- The programs are applying in the laboratory tutorials with the students of Mechanical Engineering in Technical University – Gabrovo, BULGARIA.
- These programs are written in accordance with plans of SCOPES PROJECT No 7 IP 65642 "Establishing CSE in Bulgaria and Macedonia".

List Of The Programs

1. DTX   - rectilinear motion of a particle;
2. DTXY   - planar motion of a particle in Cartesian coordinates;
3. DTPC   - planar motion of a particle in polar coordinates;
4. ROT   - rotational motion of a body;
5. EKIN   - theorem of the kinetic energy in differential form;
6. LAGRE1 - Lagrange equation for systems with 1 degree of freedom;
7. LAGRE2 - Lagrange equations for systems with 2 degrees of freedom;
8. LAGREN - Lagrange equations for systems with N degrees of freedom.

```
% ****************************************************************
%                     P r o g r a m   DTX
% ****************************************************************
%
% PURPOSE:
% Resolve numerically differential equation of rectilinear motion
% of a particle
%                    m*d2x/dt2 = Fx(t,x,v)
% and plots graphics of coordinate, velocity and phase plane.
% If possible, the program could solve the problem analytically.
%
%  INPUT DATA:
%     m    - mass of the particle;
%     Fx   - projection of forces on axis x - Fx = Fx(t,x,v);
%     x0   - initial value of the coordinate ;
%     v0   - initial value of the velocity ;
%     Tend - upper bound of the integration ;
%     eps  - precision of the integration ;
%     np   - number of parameters .
%     P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
```

```
% NOTES:
%  1. The coordinate is designed by the symbol 'x' and velocity by 'v';
%  2. The physical names of the parameters are assigned to the
%     cells of the array P like this: P{1}='m', P{2}='c',...;
%  3. For analytical solution the values of Tend, eps, np and P are not
%     needed;
%  4. Initial values x0, v0 have to be entered as strings, even though
%     they represent numbers!
%  5. All the data can be input from file or in interactive mode.
%
% EXAMPLE of DATA FILE:
%  % Data for problem ...
%    m    = 'm';  ( or m = '5.3';)
%    Fx   = '-k*v - c*x'; % k, c - parameters
%    x0   = 'x0'; ( or x0 = '0.1';)
%    v0  = 'v0';  ( or v0 = '10';)
%    Tend = 20;
%    eps  = 1.e-8;
%    np   = 3;
%    P{1} = 'm';
%    P{2} = 'k';
%    P{3} = 'c';

% ****************************************************************
%                   P r o g r a m    DTXY
% ****************************************************************
%
% PURPOSE:
%  Resolve numerically differential equations of plane motion of a
%  particle      m*d2x/dt2 = Fx(t,x,y,xt,yt);
%                m*d2y/dt2 = Fy(t,x,y,xt,yt),
%  plots trajectory and graphics of coordinates and velocity.
%
% INPUT DATA:
%    m    - mass of the body ;
%    Fx   - sum of projections of forces on axis x ;
%    Fy   - sum of projections of forces on axis y ;
%    x0   - initial value of the coordinate x ;
%    y0   - initial value of the coordinate y ;
%    v0   - initial value of velocity ;
%    alfa - angle between v0 and horizontal plane;
%    Tend - upper bound of the integration ;
%    eps  - precision of the integration ;
%    np   - number of parameters .
%    P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
% NOTES:
%  1. The coordinates are designed by the symbols 'x', 'y'
%     and there first derivatives by 'xt', 'yt';
%  2. The physical names of the parameters are assigned to the
%     cells of the array P like this: P{1}='m', P{2}='c',...;
```

```
%   3. For analytical solution the values of Tend, eps, np and P are not
%      needed.
%   4. Initial values x0, y0, v0, alfa must to be entered as strings,
%      even though they represent numbers!
%   5. All the data can be input from file or in interactive mode.
%
%  EXAMPLE of DATA FILE:
% % Dynamics of a projectile, launched
% % with initial velocity v0 under
% % angle "alfa'
%   m =  'm';
%   Fx = '-k*xt';
%   Fy = '-k*yt - m*9.81';
%   x0 = '0';
%   y0 = '0';
%   v0 = 'v0';
%   alfa = 'alfa';
%   Tend = 10;
%   eps = 1.e-10;
%   np = 2;
%   P{1} = 'm';
%   P{2} = 'k';

% *****************************************************************
%                     P r o g r a m    DTPC
% *****************************************************************
%
%  PURPOSE:
%   Resolve numerically differential equations of plane motion of a
%   particle in polar coordinates
%       m*(rtt-r*ft^2)    = Fr(t,r,f,rt,ft,rtt,ftt),
%       m*(r*ftt+2*rt*ft) = Ff(t,r,f,rt,rt,rtt,ftt)
%   plots trajectory and graphics of coordinates and velocity.
%
%  INPUT DATA:
%     m    - mass of the body ;
%     Fr   - sum of projections of forces on axis [r] ;
%     Ff   - sum of projections of forces on axis [f] ;
%     r0   - initial value of the coordinate r ;
%     f0   - initial value of the coordinate f ;
%     v0   - initial value of velocity ;
%     alfa - angle between v0 and polar axis p;
%     Tend - upper bound of the integration ;
%     eps  - precision of the integration ;
%     np   - number of parameters .
%     P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
%  NOTES:
%   1. The coordinates are designed by the symbols 'r'- radius,
%      'f'-polar angle. There first derivatives by 'rt', 'ft' and
%      second by 'rtt', 'ftt';
```

```
%   2. If the second derivatives of the coordinates are present in the
%      right sides of DE's, the problem can be solved only if 'rtt' and
%      'ftt' are present linearly!
%   3. The physical names of the parameters are assigned to the
%      cells of the array P like this: P{1}='m', P{2}='c',...;
%   4. All the data can be input from file or in interactive mode.
%   5. Even the most sample problems can't be solved analytically
%      in polar coordinates!
%
%  EXAMPLE of DATA FILE:
% % Dynamics of a particle, moving
% % on a horizontal plane under action
% % of an elastic and an resistance force.
%   m =  'm';
%   Fr = '-c*r - k*rt';
%   Ff = '-k*r*ft';
%   r0 = 'r0';
%   f0 = 'f0';
%   v0 = 'v0';
%   alfa = 'alfa';
%   Tend = 20;
%   eps = 1.e-8;
%   np = 3;
%   P{1} = 'm';
%   P{2} = 'c';
%   P{3} = 'k';

% **************************************************************
%                   P r o g r a m   ROT
% **************************************************************
%
% PURPOSE:
% Resolve numerically differential equation of rotational motion
% of a body
%                  Jz*d2f/dt2 = Mz(t,f,w)
% and plots graphics of coordinate, velocity and phase plane.
% If possible, the program could solve the problem analytically.
%
%  INPUT DATA:
%     Jz   - moment of inertia of the body ;
%     Mz   - rotational moment  Mz = Mz(t,f,w);
%     f0  - initial value of the coordinate ;
%     w0   - initial value of the angular velocity ;
%     Tend - upper bound of the integration ;
%     eps  - precision of the integration ;
%     np   - number of parameters .
%     P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
%  NOTES:
%   1. The coordinate is designed by the symbol 'f' and velocity by 'w';
%   2. The physical names of the parameters are assigned to the
```

```
%      cells of the array P like this: P{1}='Jz', P{2}='c',...;
%   3. For analytical solution the values of Tend, eps, np and P are not
%      needed.
%   4. The parameters Jz, f0 and w0 have to be entered only as
%      strings, even though they represent numbers!
%   5. All the data can be input from file or in interactive mode.
%
%  EXAMPLE of DATA FILE:
%   % Data File for problem ...
%     Jz   = 'Jz'; ( or Jz = '0.15';)
%     Mz   = '-k*w - c*f'; % k, c - parameters
%     f0  = 'f0';  ( or f0 = '0.33';)
%     w0   = 'w0'; ( or w0 = '7';)
%     Tend = 20;
%     eps  = 1.e-8;
%     np   = 3;
%     P{1} = 'Jz';
%     P{2} = 'k';
%     P{3} = 'c';

% ***************************************************************
%                   P r o g r a m   EKIN
% ***************************************************************
%
% PURPOSE:
%   Derive differential equation of motion of a mechanical
%   system with one degree of freedom by means of the theo-
%   rem of Kinetic Energy     dEk/dt = N(t,q,qt)
%   Resolve the equation numerically and plots the graphics
%   of the coordinate, velocity and phase plane.
%   If possible, the program could solve the problem analytically.
%
% INPUT DATA:
%   Ek   - expression of the kinetic energy Ek = Ek(q,qt);
%   N    - power of the forces and moments N = N(t,q,qt);
%   q0   - initial value of the coordinate;
%   qt0  - initial value of the velocity;
%   Tend - upper bound of the integration;
%   eps  - precision of the calculations;
%   np   - number of parameters .
%   P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
%  NOTES:
%   1. The coordinate is designed by the symbol 'q' and velocity by 'qt';
%   2. The physical names of the parameters are assigned to the
%      cells of the array P like this: P{1}='m', P{2}='c',...;
%   3. For analytical solution the values of Tend, eps, np and P are not
%      needed.
%   4. Initial values q0, qt0 must to be entered as strings, even though
%      they represent numbers!
%   5. All the data can be input from file or in interactive mode.
```

```
%
%  EXAMPLE of DATA FILE:
%   % Data for problem ...
%     Ek   = '1/2*a*qt^2';
%     N    = '-(k*qt + 9.81*sin(q))*qt';
%     q0   = 'q0';  ( or q0  = '0.12';)
%     qt0  = 'qt0'; ( or qt0 = '7.5';)
%     Tend = 20;
%     eps  = 1.e-8;
%     np   = 2;
%     P{1} = 'a';
%     P{2} = 'k';

% ***************************************************************
%                  P r o g r a m   LAGRE1
% ***************************************************************
%
% PURPOSE:
%   Derive differential equation of motion of a mechanical
%   system with one degree of freedom by means of Lagrange
%   equation   d/dt(dL/dqt) - dL/dq = QN(t,q,qt)
%   Resolve the equation numerically and plots the graphics
%   of the coordinate, velocity and phase plane.
%   If possible, the program could solve the problem analytically.
%
%  INPUT DATA:
%    L    - expression of the Lagrangian L = L(q,qt);
%    QN   - generalized not potential force QN = QN(t,q,qt);
%    q0   - initial value of the coordinate;
%    qt0  - initial value of the velocity;
%    Tend - upper bound of the integration;
%    eps  - precision of the calculations;
%    np   - number of parameters .
%    P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
%  NOTES:
%   1. The coordinate is designed by the symbol 'q' and velocity by 'qt';
%   2. The physical names of the parameters are assigned to the
%      cells of the array P like this: P{1}='m', P{2}='c',...;
%   3. For analytical solution the values of Tend, eps, np and P are not
%      needed.
%   4. Initial values q0, qt0 must to be entered as strings, even though
%      they represent numbers!
%   5. All the data can be input from file or in interactive mode.
%
%  EXAMPLE of DATA FILE:
%   % Data for problem ...
%    L    = '1/2*a*qt^2 + 9.81*l*cos(q)';
%    QN   = '-k*qt';
%    q0   = '0.5'; ( or q0 = 'q0';)
%    qt0  = '10';  ( or qt0 = 'qt0';)
```

```
%    Tend = 20;
%    eps  = 1.e-7;
%    np   = 3;
%    P{1} = 'a';
%    P{2} = 'l';
%    P{3} = 'k';

% *******************************************************************
%                      P r o g r a m    LAGRE2
% *******************************************************************
%
% PURPOSE:
%   Derive differential equations of motion of a mechanical system
%   with two degree of freedom by means of Lagrange equations
%        d/dt(dL/dqt1) - dL/dq1 = QN1(t,q1,q2,qt1,qt2);
%        d/dt(dL/dqt2) - dL/dq2 = QN2(t,q1,q2,qt1,qt2);
%   Resolve the equations numerically and plots the graphics
%   of the coordinates, velocities and phase planes.
%
% INPUT DATA:
%    L    - Expression of the Lagrangian  L = L(t, q1, q2, qt1, qt2);
%    QN1  - generalized non potential force QN1 = QN1(t,q1,q2,qt1,qt2);
%    QN2  - generalized non potential force QN2 = QN2(t,q1,q2,qt1,qt2);
%    qj0  - vector initial values of the coordinates;
%    qtj0 - vector initial values of the velocities;
%    Tend - upper bound of the integration;
%    eps  - precision of the calculations;
%    np   - number of parameters .
%    P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%
% NOTES:
%   1. The coordinates are designed by the symbols 'q1', 'q2'
%      and velocities by 'qt1', 'qt2';
%   2. The physical names of the parameters are assigned to the
%      cells of the array P like this: P{1}='m', P{2}='c',...;
%   3. For analytical solution the values of Tend, eps, np and P are not
%      needed.
%   4. Initial values q0, qt0 must to be entered as strings, even though
%      they represent numbers!
%   5. All the data can be input from file or in interactive mode.
%
% EXAMPLE of DATA FILE:
%  % Problem: Elliptical Pendulum
%    L = ['1/2*(m1+m2)*qt1^2 + 1/2*m2*l^2*qt2^2 + ',...
%            'm2*l*qt1*qt2*cos(q2) - 1/2*c*q1^2 + ',...
%            '9.81*m2*l*cos(q2)']; % Lagrangian
%    QN{1}  = '-alfa*qt1'; % Generalized
%    QN{2}  = '-k*qt2';    % non potential forces
%    qj0  = [0.02, 0];     % Initial coordinates
%    qtj0 = [0.1, 0];      % Initial velocities
%    Tend = 20;            % Upper bound of integration
```

```
%    eps  = 1.e-8;          % Precision of computations
%    np = 6;                % Number of parameters
%    P{1} = 'm1';           % Name assignation of the
%    P{2} = 'm2';           % physical parameters
%    P{3} = 'l';
%    P{4} = 'c';
%    P{5} = 'alfa';
%    P{6} = 'k';

% *****************************************************************
%                    Universal Program LAGREN
% *****************************************************************
%
% PURPOSE:
%   - Derives differential equations of motion of a mechanical
%     system  with arbitrary degree of freedom 's' by means of
%     LAGRANGE equations:
%        d/dt(dL/dqtj)- dL/dqj = QNj, j = 1, 2, ..., s ;
%   - Generate automatic the file-function, which describes right-
%     hand sides of canonical differential equations system;
%   - Integrates the equations numerically and plots the graphics
%     of the coordinates, velocities and phase planes.
%     For systems with one degree of freedom, could find analytical
%     solution, if possible.
%
% INPUT DATE:
%   s – degree  of freedom of the system;
%   L = L(t,q1,q2,...,qs,qt1,qt2,...,qts)- Lagrangian;
%   QN{j} = F(t,q1,q2,...,qs,qt1,qt2,...,qts)- generalized
%   non potential forces (array of cells), j = 1,2,...,s ;
%   np - number of parameters;
%   P{1}, P{2}, ..., P{np} - names of the parameters (array of cells);
%   qj0  = [q10,q20,...,qs0] - vector initial coordinates;
%   qtj0 = [qt10,qt20,...,qts0] - vector initial velocities;
%   Tend - upper bound of the integration;
%   eps - precision of the computations.
%
% NOTES:
%   1. The coordinates are designed by the symbols q1,q2,...,qs;
%   2. The velocities are designed by the symbols qt1,qt2,...,qts;
%   3. The physical names of the parameters are assigned to the
%      cells of the array P like this: P{1}='m1', P{2}='alfa',...;
%   4. All the data can be input from file or in interactive mode;
%   5. If, for a mechanical system with 1 degree of freedom, you think
%      to try an analytical solution, you have to enter initial
%      coordinate qj0 and initial velocity qtj0 as strings like this:
%         qj0  = 'q0';  ( or qj0  = '0.1';)
%         qtj0 = 'qt0'; ( or qtj0 = '7.0';)
%      After you have got the analytical solution and save it in a file,
%      you can immediately continue with the numerical solution. Than,
%      if qj0 and qtj0 have been symbols, you will be prompted to enter
```

```
%       them as numbers!
%
% EXAMPLE of DATA FILE:
%  % Problem: Elliptical Pendulum
%    s = 2;                   % Degree of freedom
%    L = ['1/2*(m1+m2)*qt1^2 + 1/2*m2*l^2*qt2^2 + ',...
%            'm2*l*qt1*qt2*cos(q2) - 1/2*c*q1^2 + ',...
%            '9.81*m2*l*cos(q2)']; % Lagrangian
%    QN{1}  = '-alfa*qt1'; % Generalized
%    QN{2}  = '-k*qt2';    % non potential forces
%    qj0  = [0.02, 0];     % Initial coordinates
%    qtj0 = [0.1, 0];      % Initial velocities
%    Tend = 20;            % Upper bound of integration
%    eps  = 1.e-8;         % Precision of computations
%    np = 6;               % Number of parameters
%    P{1} = 'm1';          % Name assignation of the
%    P{2} = 'm2';          % physical parameters
%    P{3} = 'l';
%    P{4} = 'c';
%    P{5} = 'alfa';
%    P{6} = 'k';
```

###

You can start anyone of the programs and get HELP information about it in two different ways:

1. Enter the command
   \>> dinp
and answer the questions on the command window.

2. Start directly a program entering its name
   \>> *name_of_program*
or get HELP by the command
\>> help *name_of_program*

###

For each one of the programs are presented DATA files, demonstrating how to resolve some problems of Dynamics. In parenthesis are given exemplary values of the parameters.

Description of DATA files:

DTX → dtx1 – Free damped vibrations of a particle.
Parameters:
m - mass of the particle (1 - 10);
k - coefficient of resistance (0 – 2);
c - stiffness of the spring (100 – 500).

DTXY → dtxy1 – Dynamics of a particle, thrown with initial velocity v0 under angle "alfa' to the horizon.

Parameters:

- m - mass of the particle (5 – 20);
- k - koefiicient of resistance (0 – 1) .

dtxy2 - Motion of a particle on a horizontal plane under action of an elastic and a resistance force.

Parameters:

- m - mass of the particle (5 – 10);
- c - spring stiffness (100 – 500);
- k - coefficient of damping (0 – 2).

dtxy3 - Motion of a particle on a horizontal plane under action of elastic and resistance forces and force of dry friction.

Parameters:

- m - mass of the particle (5 – 10);
- c - spring stiffness (100 – 500);
- k - coefficient of resistance (0 – 0.5);
- mu - coefficient of dry friction (0 – 0.3).

DTPC → dtpc1 - Dynamics of a particle, moving on a horizontal plane under action of an elastic and an resistance force.

Parameters:

- m - mass of the particle (5 – 10);
- c - spring stiffness (100 – 500);
- k - coefficient of resistance (0 – 2).

dtpc3 - Dynamics of a particle, moving on a horizontal plane under action of a body, bound up to the particle by a non elastic thread. The body can move in vertical direction.

Parameters:

- m - mass of the particle (0.05 - 0.08);
- G - weight of the body (40 – 60);
- mu - coefficient of dry friction (0 – 0.4).

ROT → rotd - Rotational motion of a body.

Parameters:

- Jz - moment of inertia (0.05 - 0.5);
- k - coefficient of resistance (0 – 1.5);
- c - spring stiffness (50 – 200).

EKIN → ekin1 - Oscillations of a mathematical pendulum.

Parameters:

- m - mass of the pendulum (1 - 10);
- k - coefficient of resistance (0 – 0.8).

pilz1 - Forced damped vibrations of a mechanical system with one degree of freedom, contained a reverse pendulum.

Parameters:

- c - spring stiffness (100 – 300);
- k - coefficient of damping (0.3 - 0.6);
- p - disturbance frequency (5 – 10).

LAGRE1 → mpend - Mathematical Pendulum .

Parameters:

m - mass of the particle (1 – 10);
l - length of the pendulum (0.1 – 0.5);
k - coefficient of resistance (0 – 1).

pilz2 - Forced damped vibrations of a mechanical system with one degree of freedom, contained a reverse pendulum (the same as **pilz1**).

LAGRE2 → dmah2 - Dynamics of an elliptical pendulum consisting of a slider with spring and a hinged to the slider homogeneous rod.

Parameters:

m1 - mass of the slider (1 – 10);
m2 - mass of the rod (1 – 10);
l - length of the rod (0.2 – 1);
c - spring stiffness (100 – 500);
alfa - coefficient of damping of the slider (0 – 0.7);
k - coefficient of damping of the rod (0 – 0.7).

LAGREN → lan - Free linear vibrations of a system with one degree of freedom .

Parameters:

a - generalized coefficient of inertia (5 – 10);
b - generalized coefficient of resistance (0 – 1);
c - generalized coefficient of stiffness (100 – 500).

pilz3 - Forced damped vibrations of a mechanical system with one degree of freedom, contained a reverse pendulum (the same as **pilz1**).

dmahn - dynamics of an elliptical pendulum (the same as **dmah2**).

mah3 - dynamics of a triple pendulum – three hinged each to other rods, moving in a vertical plane.

Parameters:

m1, m2, m3 - masses of the rods (1 - 5);
l1, l2, l3 - lengths of the rods (0.1 – 0.5);
k1, k2, k3 - coefficients of damping.

##

!!! **NOTE :**

1. The programs are tested and work properly with MATLAB ver. 6.0 and higher . Before running you have to had installed Extended Symbolic Math Toolbox.
2. You have to set PATH to the directories "P-Codes" and "HELP Files".
3. DATA files must be only in the current directory !
4. Everyone of the programs, after generating the file function, open it by the system editor medit. You can look at it, correct if needed and close. After closing, the program continue with numerical computations.

##

List of the Files

Programs (in p-code): dtx.p, dtxy.p, dtpc.p, rot.p, ekin.p, lagre1.p, lagre2.p, lagren.p

HELP Files: dtx.m, dtxy.m, dtpc.m, rot.m, ekin.m, lagre1.m, lagre2.m, lagren.m

DATA Files: dtx1.m, dtxy1.m, dtxy2.m, dtxy3.m, dtpc1.m, dtpc3.m, rotd.m, ekin1.m, pilz1.m, mpend.m, pilz2.m, dmah2.m, lan.m, pilz3.m, dmahn.m, mah3.m

Start File : dinp.m

This File: readme.doc .

##

Author : Assoc. Prof. Jordan Tonchev Jordanov
dep. of "Technical Mechanics"
Technical University - Gabrovo
Gabrovo 5300
BULGARIA
Tel.: ++ 359/66/801 226
E-mail : tonchev@tugab.bg

Author would be glad to receive any suggestions for improvements of the programs!

Thank You in advance !

Jordan Tonchev

18.09.2003
Gabrovo

##