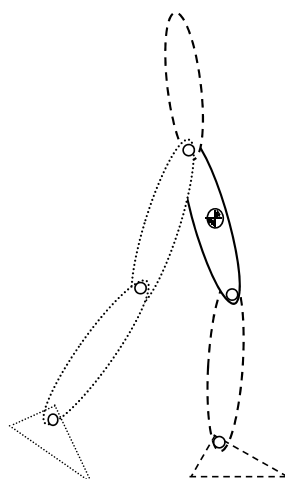


USER'S MANUAL

MUBOKINE

KINEMATIC ANALYSIS OF MULTIBODY SYSTEMS



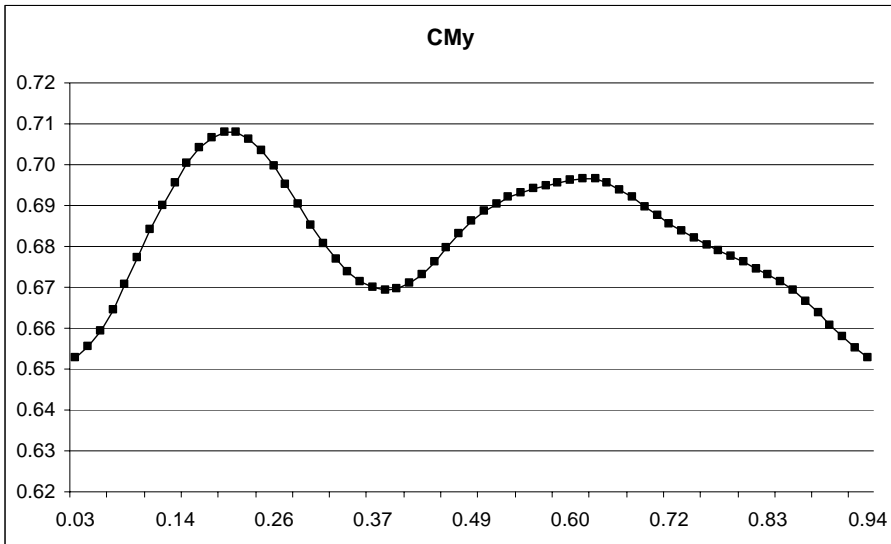
A FORTRAN Program for Kinematic Analysis of General Planar
Multibody Systems by

PAULO FLORES

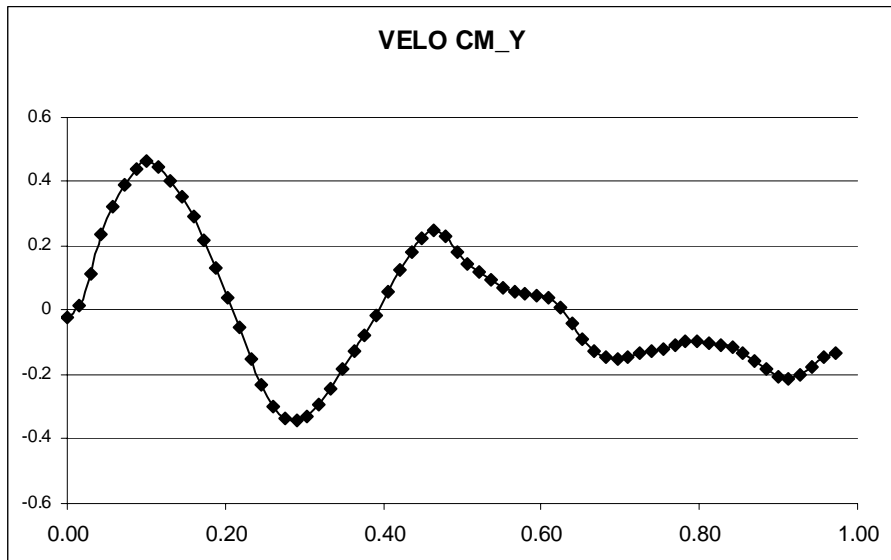
UNIVERSIDADE DO MINHO 2010

INDEX

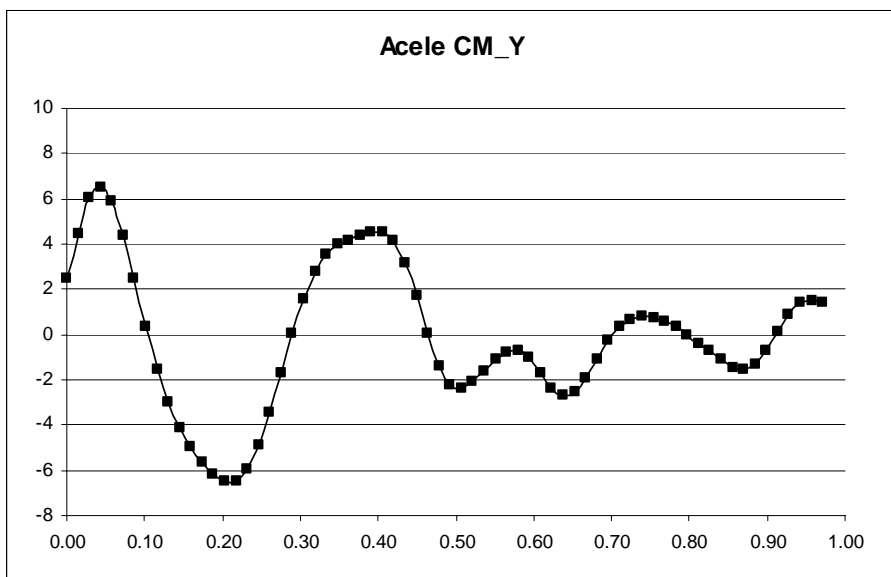
INDEX.....	1
1. INTRODUCTION.....	3
2. CONSTRAINT EQUATIONS.....	4
Revolute Joints.....	4
Guide Constraints.....	5
3. KINEMATIC ANALYSIS.....	6
4. AVAILABLE ELEMENT TYPES.....	8
5. INPUT FILE.....	9
Number of Elements.....	9
Rigid Bodies.....	10
Revolute Joints.....	10
Translational Joints.....	11
Ground Constraints.....	11
Simple Constraints.....	11
Driving Constraints.....	12
Points of Interest.....	12
Guide Constraints.....	13
Time Parameters.....	14
6. OUTPUT FILE.....	15
7. EXAMPLE APPLICATION: FOUR-BAR LINKAGE.....	16
8. EXAMPLE APPLICATION: UPPER LEG (THIGH) GUIDANCE.....	20



..... 22



..... 22



..... 22

9. FLOWCHART OF THE KABIOS PROGRAM..... 23

1. INTRODUCTION

Based on the methodology described in the reference textbook (P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988) a computer program designated by KABIOS (Acronym for Kinematic Analysis of Biomechanical Systems) was devised. This program is based on the Nikravesh's KAP with some important modifications.

The KABIOS performs kinematic analysis of two-dimensional biomodels, using multibody systems methodologies. The kinematic joints available in the program are revolute joint and translational joint. It is also possible to include simple kinematic constraints. The main difference from KAP code is the inclusion of guide constraints. These type of constraints impose bodies trajectories based on known CM linear and angular positions. This program also performs kinematic analysis of planar mechanical systems, as original Nikravesh's KAP.

This program is based on the Method of Appended Driving Constraints. Additional constraint equations, called the driving constraints, equal in number of degrees of freedom (DOF) of the system, are appended to the original kinematic constraints. The driving constraints are equations representing each independent coordinate as a function of time. Guide constraints used in this program can be seen as special type of driving constraints.

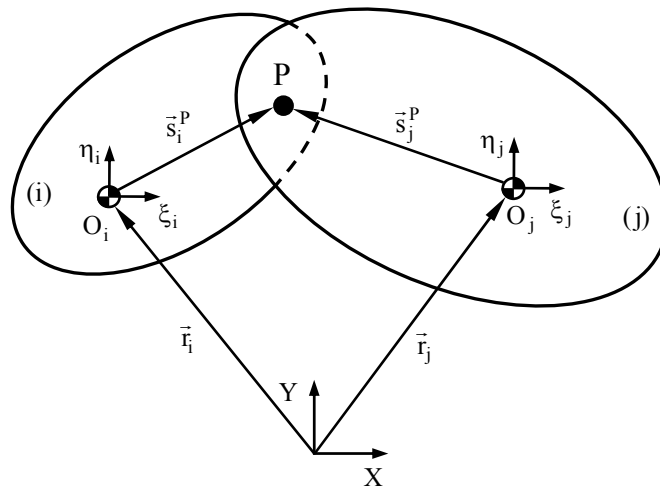
The program is written in FORTRAN language and is organized in a form that can be expanded to include other elements, such as other kinematic joints.

The user is free to use any desired unit system. It is user's responsibility to enter all the input data in a consistent system.

2. CONSTRAINT EQUATIONS

A kinematic joint imposes certain conditions on the relative motion between the adjacent bodies that it comprises. When these conditions are expressed in analytical form, they are called constraint equations. In a simple way, a constraint is any condition that reduces the number of degrees of freedom in a system. In what follows, the formulation for the planar revolute and translational joints is reviewed. For details on the formulation of other types of kinematic joints the interested reader is referred to the work of Nikravesh (1988).

Revolute Joints



Planar revolute joint connecting bodies i and j.

The revolute joint is a pin and bush type of joint that constrains the relative translation between the two bodies i and j , allowing only the relative rotations, as it is illustrated in figure above. The kinematic conditions for the revolute joint require that two and distinguish points, each one belonging to a different body, share the same position in space all the time. This means that the global position of a point P in body i is coincident with the global position of a point P in body j . Such condition is expressed by two algebraic equations¹ that can be obtained from the following vector loop equation,

$$\mathbf{r}_i + \mathbf{s}_i^P - \mathbf{r}_j - \mathbf{s}_j^P = \mathbf{0} \quad (1)$$

which is re-written as,

$$\Phi^{(r,2)} \equiv \mathbf{r}_i + \mathbf{A}_i \mathbf{s}_i^{\prime P} - \mathbf{r}_j - \mathbf{A}_j \mathbf{s}_j^{\prime P} = \mathbf{0} \quad (2)$$

Thus, there is only one relative DOF between two bodies that are connected by a planar revolute joint.

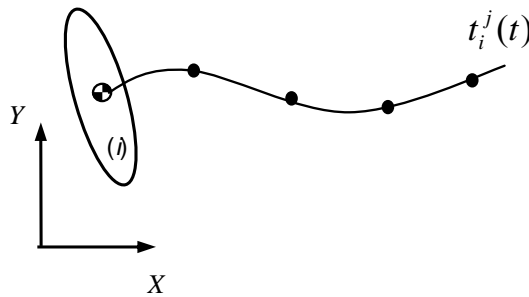
¹ To distinguish among the different constraint equations, each elementary set of constraints is identified by a superscript containing two parameters. The first parameter denotes the type of constraint while the second one defines the number of independent equations that it involves. For example $\Phi^{(r,2)}$ denotes the planar revolute joint constraint, which contains two equations.

Guide Constraints

In general, the motion of one or more bodies in a MBS is specified, i.e., the system is guided typically by rotational or translational actuators. Other type of guiding constraints are those associated with the known trajectories of the bodies CM, which can be obtained by experimental data acquisition, as it is illustrated in Figure below. The mathematical equation that represents the guiding constraint can be written as,

$$\Phi^{(g,3)} \equiv \begin{bmatrix} x_i - t_i^x(t) \\ y_i - t_i^y(t) \\ \phi_i - t_i^\phi(t) \end{bmatrix} = \mathbf{0} \quad (6)$$

where, $t_i^j(t)$ ($j = x, y, \phi$) represents the trajectory described by the CM of body i .



Body trajectory

In general, the bodies' trajectories are obtained through a digitization process in which the images collected by video cameras are used to reconstruct the two-dimensional coordinates of the anatomical points. From these anatomical points' coordinates, the CM(x,y) position are calculated, as well as the body orientation, ϕ .

The next step consists on the interpolation of the Cartesian coordinates' variation along time in order to achieve bodies trajectory curves ($t_i^j(t)$). Based on these curves, the analytical constraint equations, given by expression (6), can easily be obtained. This interpolation is performed employing cubic splines, since higher-order polynomials tend to swing through wild oscillations in the vicinity of an abrupt change, whereas cubic spline provides much more smooth transitions. Further, the use of cubic splines is useful to guarantee the continuity of the first and the second derivatives (velocity and acceleration respectively), property that is extremely important in the kinematic analysis.

This Program reads the input values of the Cartesian coordinates and performs the necessary calculations with the aid of specific routines for *cubic spline interpolation (cubic spline smoothing)* and *cubic spline derivative evaluation* available in the *ISML Fortran Numerical Libraries*.

3. KINEMATIC ANALYSIS

The kinematic analysis consists of study of the motion of a system independently of the forces that produce it. Since in the kinematic analysis the forces are not considered, the motion of the system is specified by driving elements that govern the system motion during the analysis, while the position, velocity and acceleration of the remaining elements are defined by kinematic constraint equations that describe the system topology.

It is clear that in the kinematic analysis, the number of driver constraints must be equal to the number of degrees of freedom of the multibody mechanical system. In short, the kinematic analysis is performed by solving a set of equations that result from the kinematic and driver constraints.

When the configuration of a MBS is described by N Cartesian coordinates \mathbf{q} , then a set of M algebraic kinematic independent holonomic constraints Φ can be written in a compact form as (Nikravesh, 1988),

$$\Phi(\mathbf{q}, t) = \mathbf{0} \quad (6)$$

where \mathbf{q} is the vector of generalized coordinates and t is the time variable, in general associated with the driving elements.

The velocities and accelerations of the system elements are evaluated using the velocity and acceleration constraint equations. Thus, the first time derivative with respect to time of equation (6) provides the velocity constraint equations,

$$\Phi_{\mathbf{q}} \dot{\mathbf{q}} = -\Phi_t \equiv \mathbf{v} \quad (7)$$

where $\Phi_{\mathbf{q}}$ is the Jacobian matrix of the constraint equations, that is, the matrix of the partial derivatives, $\partial \Phi / \partial \mathbf{q}$, $\dot{\mathbf{q}}$ is the vector of generalized velocities and \mathbf{v} is the right hand side of velocity equations, which contains the partial derivatives of Φ with respect to time, $\partial \Phi / \partial t$. It should be noticed that only rheonomic constraints, associated with driver equations, contribute with non-zero entries to the vector \mathbf{v} . Further, it is assumed that this vector does not present any dependency on the vector of coordinates.

A second differentiation of equation (6) with respect to time leads to the acceleration constraint equations, obtained as,

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}} \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} - 2\Phi_{\mathbf{q}t} \dot{\mathbf{q}} - \Phi_{tt} \equiv \boldsymbol{\gamma} \quad (8)$$

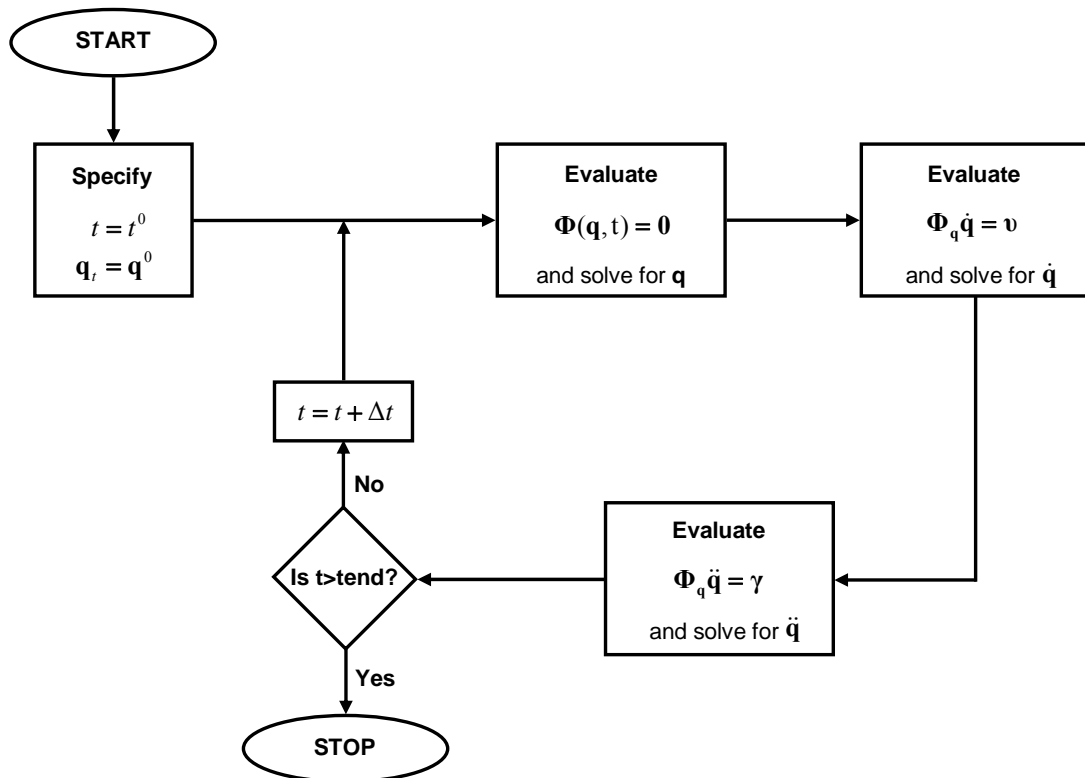
where $\ddot{\mathbf{q}}$ is the acceleration vector and $\boldsymbol{\gamma}$ is the right hand side of acceleration equations, *i.e.*, the vector of quadratic velocity terms, which contains the terms that are exclusively function of velocity, position and time. In the case of holonomic scleronomic constraints, that is, Φ is not explicitly dependent on the time, the term Φ_t in equation (7) and the $\Phi_{\mathbf{q}t}$ and Φ_{tt} terms in equation (8) vanish.

The constraint equations represented by equation (6) are, in general, non-linear in terms of \mathbf{q} and are usually solved by employing the Newton-Raphson method. Equations (7) and (8)

are linear in terms of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively, and are solved by any solver of systems of linear equations.

The kinematic analysis of a multibody system can be carried out by solving the set of equations (6), (7) and (8). The necessary steps to perform this analysis are summarized as sketched in figure below, and described as,

- i) Specify the initial conditions for positions \mathbf{q}^0 and initialize the time counter t^0 .
- ii) Evaluate the position constraint equations (6) and solve them for positions, \mathbf{q} .
- iii) Evaluate the velocity constraint equations (7) and solve them for velocities, $\dot{\mathbf{q}}$.
- iv) Evaluate the acceleration constraint equations (8) and solve them for accelerations, $\ddot{\mathbf{q}}$.
- v) Increment the time. If the time is smaller than final time, go to step ii), otherwise stop the kinematic analysis.



Flowchart of computational procedure for kinematic analysis of a MBS.

4. AVAILABLE ELEMENT TYPES

The available element types in the KABIOS are the following

Rigid bodies

Kinematic joints

 Revolute

 Translational

Ground constraints

Simple constraints

Driving constraints

Guide constraints

5. INPUT FILE

A multibody system model must be provided to KABIOS as an input file. The input file is the file that contains the model description, and must be a **text file**. The elements of a multibody model are described in data lines.

A list of all the prompts given by the program KABIOS is presented below. The prompts are labeled for easy reference, from (a) to through (i). The prompts given are as follows:

LABEL	PROMPT
(a)	ENTER NB, NR, NT, NG, NS, ND, NP,NGUI,NDATA
(b)	FOR BODY k ENTER INITIAL EST. ON X, Y, PHI
(c)	FOR REVOLUTE JOINT NO. k ENTER BODY NOS. I AND J XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
(d)	FOR TRAN. JOINTS NO. k ENTER BODY NOS. I AND J XI-P-I, ETA-P-I, XI-Q-I, ETA-Q-I, XI-P-J, ETA-P-J
(e)	ENTER BODY NO. FOR NO. k GROUND BODY
(f)	FOR SIMPLE CONSTRAINT NO. k ENTER BODY NO. AND 1, 2, OR 3 FOR X, Y, OR PHI CONSTRAINT DIRECTION
(g)	FOR DRIVER NO. k ENTER BODY NO. I 1, 2, OR 3 FOR X, Y, OR PHI INITIAL POSITION, VELOCITY, AND ACCELERATION
(h)	POINT OF INTEREST NO. k ENTER BODY NO. XI-P-I, ETA-P-I COORDINATES
(i)	FOR GUIDE NO. k ENTER BODY NO. I 1, 2, OR 3 FOR X, Y, OR PHI
(j)	ENTER STARTING TIME, FINAL TIME, AND TIME INCREMENT

Number of Elements

The first set of data the program requests is

ENTER NB, NR, NT, NG, NS, ND, NP, NGUI, NDATA

These data are defined as follows:

NB	Number of bodies in the system, including ground
NR	Number of revolute joints in the system
NT	Number of translational joints in the system
NG	Number of bodies that are attached to (or considered as) ground
NS	Number of simple constraints in the system

ND	Number of driving constraints (driving links)
NP	Number of points of interest
NGUI	Number of guide constraints
NDATA	Number of input data, associated to guide constraints, used in the cubic spline interpolation

The program computes the number of coordinates N and the total number of constraint equations M from the above information. N is equal to $3 \cdot NB$, and M is equal to $2 \cdot (NR + NT) + 3 \cdot NG + NS + ND + NGUI$. N must be equal to M . If M is greater than N , then an error message is given. Otherwise, the program continues.

The number of each element tells the program how many data lines should exist. A data line describes the characteristics of each element. The information on a data line can be separated by a comma or by one or several spaces.

Rigid Bodies

The rigid bodies data line must be provide the initial conditions on the positions. The initial conditions on the coordinates do not need to be correct.

The data lines for rigid bodies are repeated NB times, and are as follows:

i X Y PHI

where,

i Body number
 X X-coordinate of the body
 Y Y-coordinate of the body
 PHI ϕ -coordinate of the body

Revolute Joints

Refer to section 4.2.1 of the textbook, "Computer-Aided Analysis of Mechanical Systems," by Nikravesh 1988, for detailed description of this joint.

The data lines for revolute joints are repeated NR times, and are as follows:

i j XI-P-I ETA-P-I XI-P-J ETA-P-J

where,

i Index number of body i
 j Index number of body j
 XI-P-I ξ -coordinate of point P on body i
 ETA-P-I η -coordinate of point P on body i

XI-P-J ξ -coordinate of point P on body j
ETA-P-J η -coordinate of point P on body j

Translational Joints

Refer to section 4.2.1 of the textbook, "Computer-Aided Analysis of Mechanical Systems," by Nikravesh 1988, for detailed description of this joint.

The data lines for translational joints are repeated NT times, and are as follows:

i j XI-P-I ETA-P-I XI-Q-I ETA-Q-I XI-P-J ETA-P-J

where,

i Index number of body i
j Index number of body j
XI-P-I ξ -coordinate of point P on body i
ETA-P-I η -coordinate of point P on body i
XI-Q-I ξ -coordinate of point Q on body i
ETA-Q-I η -coordinate of point Q on body i
XI-P-J ξ -coordinate of point P on body j
ETA-P-J η -coordinate of point P on body j

Ground Constraints

Refer to section 4.2.7 of the textbook, "Computer-Aided Analysis of Mechanical Systems," by Nikravesh 1988, for detailed description of this kinematic constraint.

The data lines for ground constraints are repeated NG times, and are as follows:

i

where,

i Index number of body i constrained to the ground

Simple Constraints

Refer to section 4.2.7 of the textbook, "Computer-Aided Analysis of Mechanical Systems," by Nikravesh, for detailed description of this kinematic constraint.

The data lines for simple constraints are repeated NS times, and are as follows:

i k

where,

i	Index number of body <i>i</i>
k	Index of the coordinate to be constrained: k=1: constrains X-coordinate (Eq. 4.31) k=2: constrains Y-coordinate (Eq. 4.32) k=3: constrains ϕ -coordinate (Eq. 4.33)

Driving Constraints

Refer to section 4.2.8 of the textbook, "Computer-Aided Analysis of Mechanical Systems," by Nikravesh, for detailed description of this kinematic constraint.

In most problems there is only one driver. KABIOS can provide a driver expression on one of the three coordinates of a body:

$$\begin{array}{ll} 1 & x_i = x_i^0 + v_i^0 t + \frac{1}{2} a t^2 \\ 2 & y_i = y_i^0 + v_i^0 t + \frac{1}{2} a t^2 \\ 3 & \phi_i = \phi_i^0 + \omega_i^0 t + \frac{1}{2} \alpha t^2 \end{array}$$

The data lines for diving constraints are repeated ND times, and are as follows:

i k [X, Y, or PHI] [XD, YD, or PHID] [XDD, YDD, or PHIDD]

where,

i	Index number of body <i>i</i>
k	Index of the coordinate to be constrained: k=1: constrains X-coordinate k=2: constrains Y-coordinate k=3: constrains ϕ -coordinate)
[X, Y, or PHI]	Initial linear/angular position
[XD, YD, or PHID]	Initial linear/angular velocity
[XDD, YDD, or PHIDD]	Initial linear/angular acceleration

Points of Interest

This is not a typical data line like a joint. In this data line the program is instructed to output the coordinates, velocities, and accelerations of one or more points of interest attached to one or more rigid bodies. It should be noted that KABIOS automatically outputs the coordinates, velocities, and accelerations of the origins of the bodies.

The data lines for points of interest are repeated NP times, and are as follows:

i XI-P-I ETA-P-I

where,

i Index number of body *i*
 XI-P-I ξ -coordinate of point *P* on body *i*
 ETA-P-I η -coordinate of point *P* on body *i*

Guide Constraints

In General the number of guide constraints must be equal to number of bodies plus 2 (NGUI=NB+2). This means that in pure biomechanical systems, the CM(x,y) need to be constrained only for the main body, and ϕ must be constrained for all bodies.

The data lines for diving constraints are repeated NGUI times, and are as follows:

i k

where,

i Index number of body *i*
 k Index of the coordinate to be constrained:
 k=1: constrains *X*-coordinate
 k=2: constrains *Y*-coordinate
 k=3: constrains ϕ -coordinate

KABIOS provide a guide expression on one of the three coordinates of a body:

$$\begin{array}{l} 1 \quad x_i - t_i^x(t) = 0 \\ 2 \quad y_i - t_i^y(t) = 0 \\ 3 \quad \phi_i - t_i^\phi(t) = 0 \end{array}$$

where $t_i(t)$ is introduced as a set of points as (NGUI+1 columns, NDATA rows),

Ti	Xi	Yi	PHIi
T0
...
TE

Time Parameters

It is important to select the time step adequately. A very small time step yields a more accurate result and, of course, it requires more computation time. A large time step is computationally more efficient but it may cause the numerical problems. The selection of the time step must be based on the highest frequency of the system response. Therefore, time step selection is not an easy task.

In rotating bodies, such as the slider crank mechanism, a reasonable choose for time step is a value that allows the crank not rotate more than 10 degrees in one time step.

In KABIOS the time parameters for kinematic analysis of a multibody system is given as follows:

T0 TE DT

where,

T0	Starting time
TE	Final time
DT	Time increment

6. OUTPUT FILE

The output results are written on the screen or on a file specified by the user. The results are reported with specified format.

KABIOS reports the values of coordinates, velocities, and accelerations for all bodies in the system at the end of each time step. In addition, the global coordinates, velocities, and accelerations of the special points of interest are computed and reported.

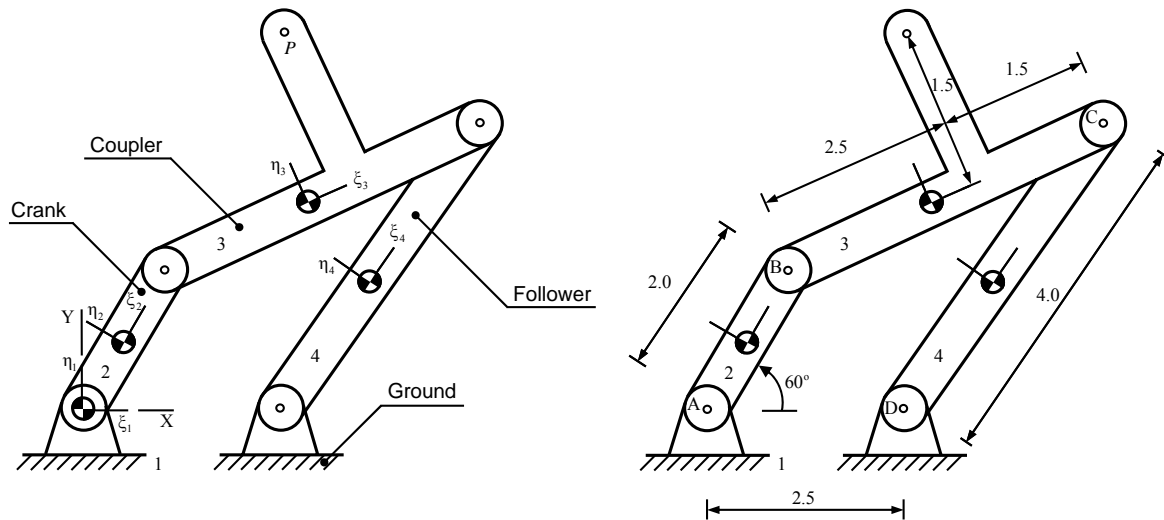
7. EXAMPLE APPLICATION: FOUR-BAR LINKAGE

A four-bar mechanism is considered for kinematic analysis (Nikravesh, 1988, pp.135-137). Figure below illustrates schematically the configuration of the four-bar linkage, which consists of four bodies, including ground, and four revolute joints.

It should be noticed that the kinematic constraints considered here are assumed to be holonomic and time invariant.

The body numbers and their corresponding coordinate systems are shown in the figure below.

Point P on body 3 is considered as point of interest.



Kinematic configuration of the four bar linkage.

The initial estimates for the coordinates are as follows:

$$\begin{array}{lll}
 x_1 = 0.0 & y_1 = 0.0 & \phi_1 = 0.0 \\
 x_2 = 0.5 & y_2 = 0.8 & \phi_2 = 60.0^\circ \\
 x_3 = 2.6 & y_3 = 2.6 & \phi_3 = 30.0^\circ \\
 x_4 = 3.5 & y_4 = 1.8 & \phi_4 = 60.0^\circ
 \end{array}$$

The local coordinates for the revolute joints are:

$$\begin{array}{llll}
 \xi_1^A = 0.0 & \eta_1^A = 0.0 & \xi_2^A = -1.0 & \eta_2^A = 0.0 \\
 \xi_2^B = 1.0 & \eta_2^B = 0.0 & \xi_3^B = -2.0 & \eta_3^B = 0.0 \\
 \xi_3^C = 2.0 & \eta_3^C = 0.0 & \xi_4^C = 2.0 & \eta_4^C = 0.0 \\
 \xi_4^D = -2.0 & \eta_4^D = 0.0 & \xi_1^D = 2.5 & \eta_1^D = 0.0
 \end{array}$$

Body 2 is the driver link, and its corresponding driving variable is ϕ_2 , with $\phi_2^0 = 1.047$ rad (60.0°) and a constant angular velocity of $\omega = 2\pi$ rad/s.

For one complete revolution of the crank, one second of simulation time is required. A time increment of 0.025s results in increments of 9° in ϕ_2 .

The point of interest (point P) is located on body 3. The local coordinates of this point are:

$$\xi_3^P = 0.5 \quad \eta_3^P = 1.5$$

In short, the KABIOS input file that describes the four-bar model is as follows:

```

4 4 0 1 0 1 1 0 0 (a)
0.0000 0.0000 0.0000 (b)
0.5000 0.8000 1.0470 (b)
2.6000 2.6000 0.5000 (b)
3.5000 1.8000 1.0000 (b)
1 2 0.0000 0.0000 -1.0000 0.0000 (c)
2 3 1.0000 0.0000 -2.0000 0.0000 (c)
3 4 2.0000 0.0000 2.0000 0.0000 (c)
4 1 -2.0000 0.0000 2.5000 0.0000 (c)
1 (e)
2 3 1.0472 6.2832 0.0000 (g)
3 0.5000 1.5000 (h)
0.0000 1.0000 0.0250 (i)

```

The output results for the two first time steps are presented below.

```

TIME = 0.0000
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
1  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
2  0.500  0.866  1.047  -5.441  3.142  6.283  -19.739  -34.190  0.000
3  2.824  2.553  0.423  -11.085  6.732  0.246  -52.441  -39.898  15.646
4  3.574  1.687  1.004  -5.644  3.590  3.344  -32.702  -5.709  12.264
POINTS OF INTEREST
NO.      X      Y      XD      YD      XDD      YDD
1  2.663  4.126  -11.472  6.692  -77.042  -42.500

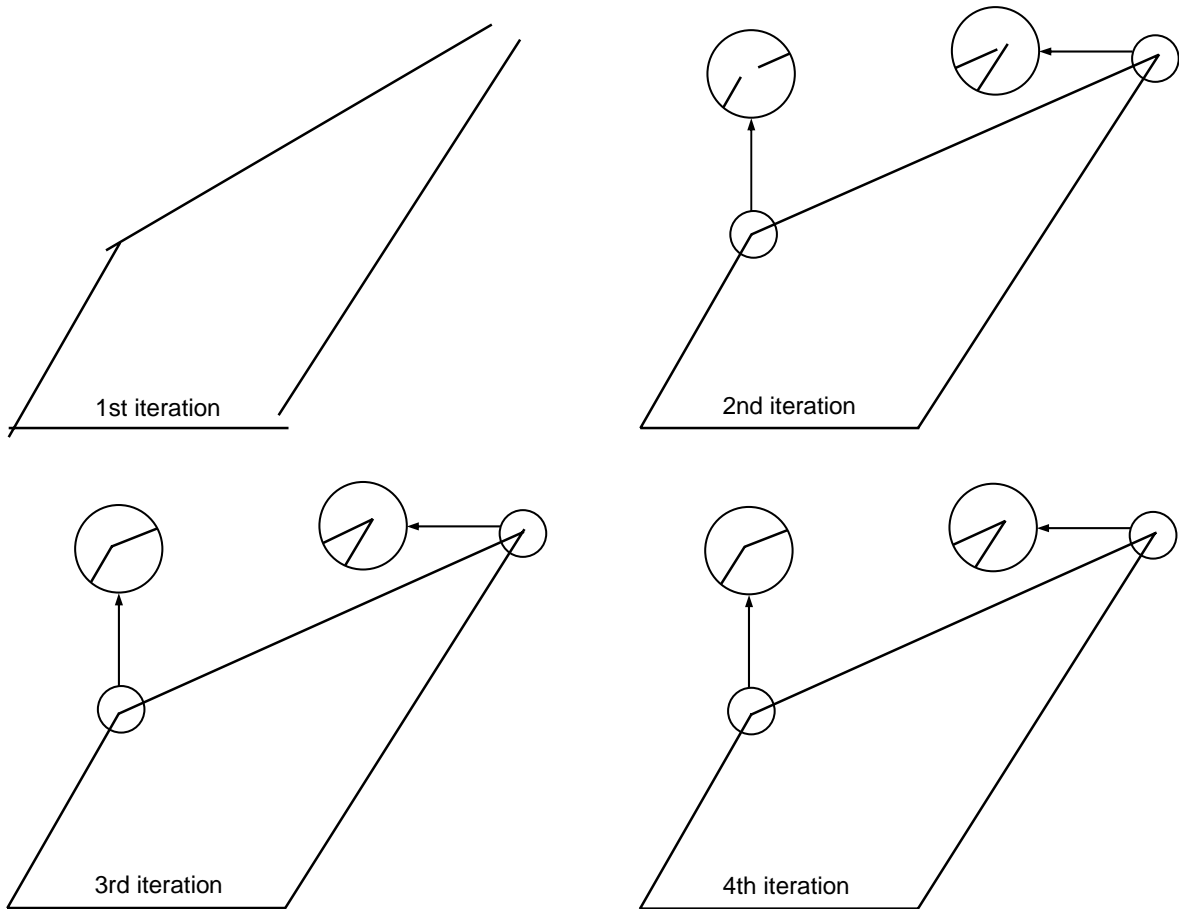
TIME = 0.0250
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
1  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
2  0.358  0.934  1.204  -5.866  2.252  6.283  -14.148  -36.856  0.000
3  2.531  2.708  0.434  -12.220  5.558  0.581  -38.613  -53.046  11.545
4  3.423  1.774  1.091  -6.354  3.306  3.581  -24.465  -16.189  7.116
POINTS OF INTEREST
NO.      X      Y      XD      YD      XDD      YDD
1  2.355  4.279  -13.133  5.455  -56.693  -55.617

```

It can be observed that at $t=0$, the KABIOS program corrects the initial estimates on the coordinates. The values of ϕ_2 is kept constant according to the value given with the driving constraint.

Also, the coordinates of body 1 remain unchanged, since body 1 is the ground. At each time step, the body numbers and the corresponding coordinates, velocities and accelerations of their points of interest are reported.

Figure below shows how the Newton-Raphson method works in order to keep the constraint violations under the value defined by FEPS variable. FEPS is the variable in the KABIOS program that the maximum error for the constraint violation.



The chart below shows the variation of number of iteration with the FEPS. NRMAX is the KABIOS variable that defines the maximum number of iterations allowed for the Newton-Raphson algorithm at each time step. NRMAX is specified as 20.

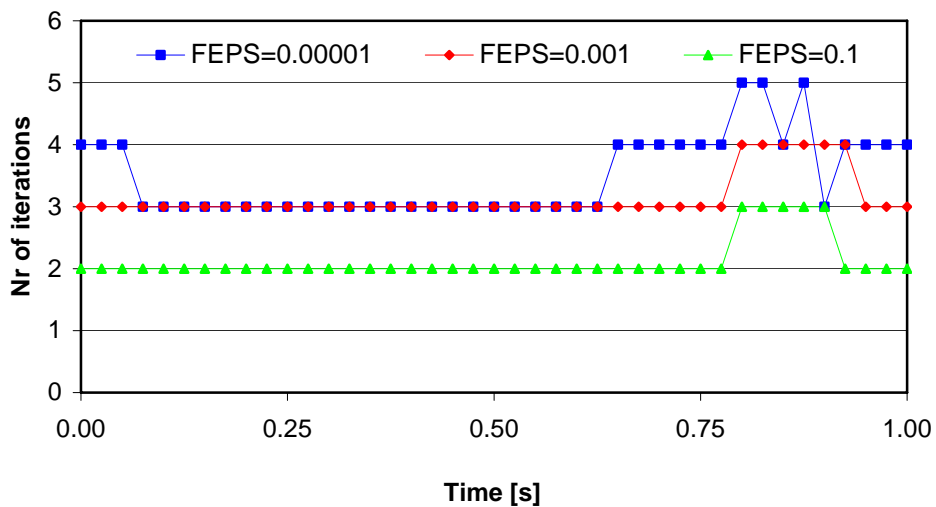
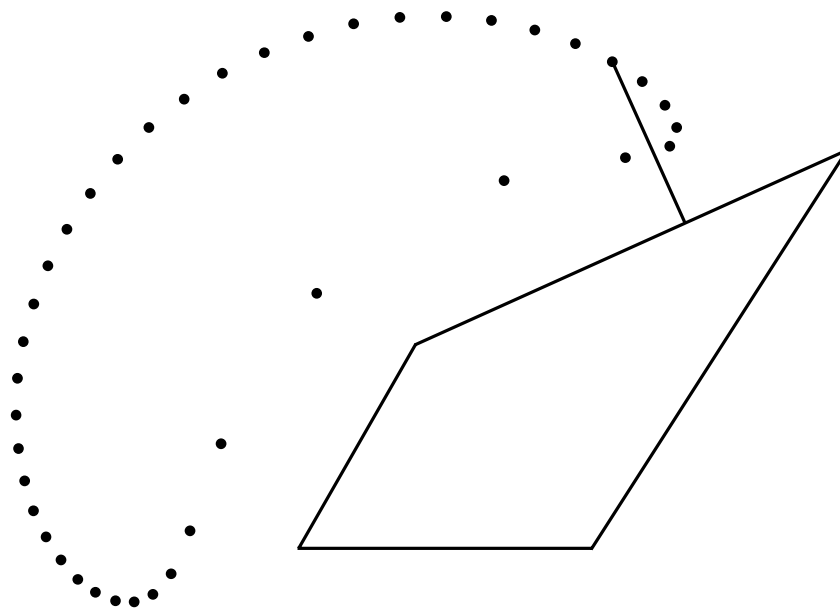


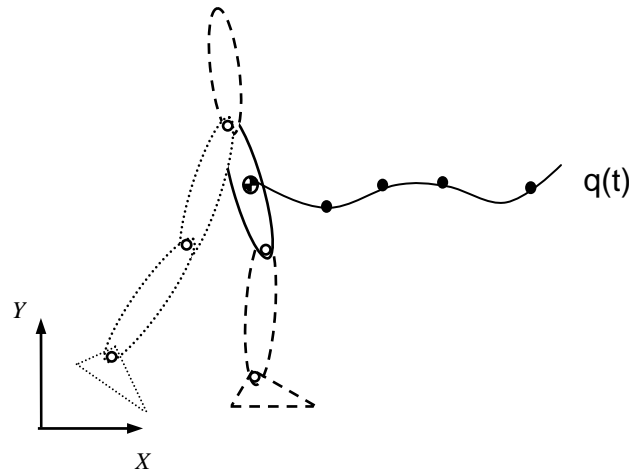
Figure below shows the path taken by point P for one revolution of the crank.



Path covered by point of interest P .

8. EXAMPLE APPLICATION: UPPER LEG (THIGH) GUIDANCE

A simple body that is a representation of the Upper Right Leg is considered for kinematic analysis. The used data is from a publication report by Winter(2005). The output of KABIOS is compared with Winter's results. Figure below illustrates schematically the configuration of the body, within the representation of a complete biomechanical model to perform gait analysis.



Kinematic configuration of the Upper Leg (Thigh).

The initial estimates for the coordinates are as follows:

$$x_1 = 0.4300. \quad y_1 = 0.6520 \quad \phi_1 = 1.4434$$

For one complete gait cycle, 0.972 seconds of simulation time is required. A time increment of 0.0145s is used, which represents the double of the input frequency data.

In short, the KABIOS input file that describes the four-bar model is as follows:

1	0	0	0	0	0	0	3	35	(a)
0.4300		0.6520		1.4434					(b)
1	1								(i)
1	2								(i)
1	3								(i)
0.0000	0.430123	0.652077		1.443387					
0.0290	0.489423	0.652808		1.546362					
0.0570	0.545594	0.659141		1.659808					
0.0860	0.598219	0.670233		1.769764					
0.1140	0.648436	0.683091		1.865757					
0.1430	0.696805	0.694952		1.940806					
0.1720	0.743154	0.703639		1.989675					
0.2000	0.787385	0.707748		2.017601					
0.2290	0.829305	0.706757		2.022837					
0.2570	0.869559	0.700867		2.010619					
0.2860	0.909368	0.691555		1.984439					
0.3150	0.950274	0.681953		1.956514					
0.3430	0.993673	0.674714		1.933825					
0.3720	1.039768	0.670524		1.919862					
0.4000	1.086796	0.669464		1.907645					
0.4290	1.132480	0.672231		1.895428					
0.4580	1.175514	0.678265		1.888446					
0.4860	1.214432	0.684909		1.877974					

0.5150	1.248995	0.689604	1.848304
0.5430	1.279970	0.692636	1.795944
0.5720	1.307300	0.694514	1.733112
0.6010	1.331829	0.695975	1.673771
0.6290	1.355112	0.696672	1.619666
0.6580	1.378287	0.694929	1.570796
0.6860	1.401978	0.691245	1.525418
0.7150	1.427097	0.686995	1.476549
0.7440	1.453355	0.683180	1.425934
0.7720	1.480774	0.679986	1.375319
0.8010	1.510731	0.677247	1.331686
0.8290	1.544018	0.674333	1.300270
0.8580	1.582550	0.670896	1.281072
0.8870	1.628757	0.666206	1.275836
0.9150	1.683475	0.660464	1.293289
0.9440	1.743471	0.654759	1.333432
0.9720	1.803705	0.650697	1.399754
0.0000	0.9720	0.0145	

(j)

The output results for the two first time steps are presented below.

```

TIME =      0.0000
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
  1  0.430  0.652  1.443  2.055 -0.020  3.400  0.000  0.000  0.000

TIME =      0.0145
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
  1  0.460  0.652  1.493  2.048  0.014  3.513 -0.987  4.665 15.627

TIME =      0.0290
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
  1  0.490  0.653  1.546  2.026  0.115  3.853 -1.973  9.330 31.255

...

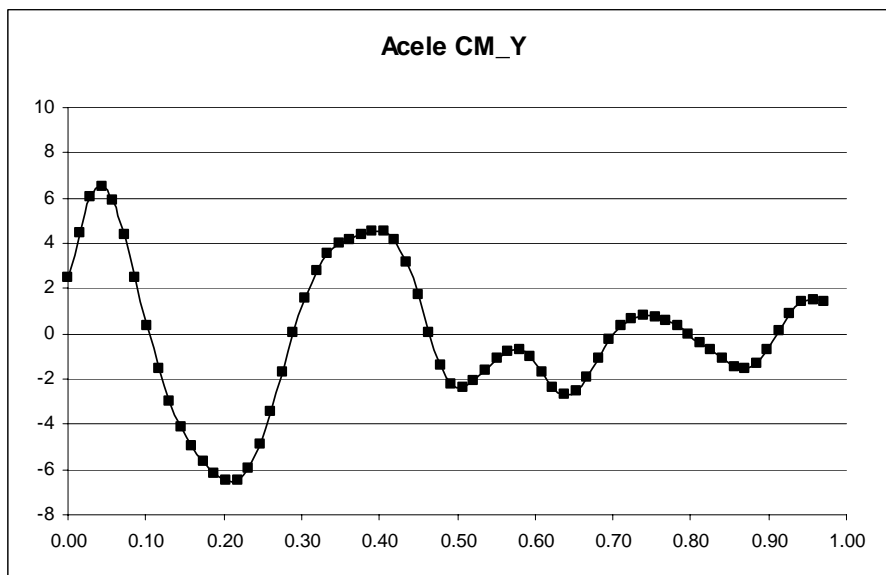
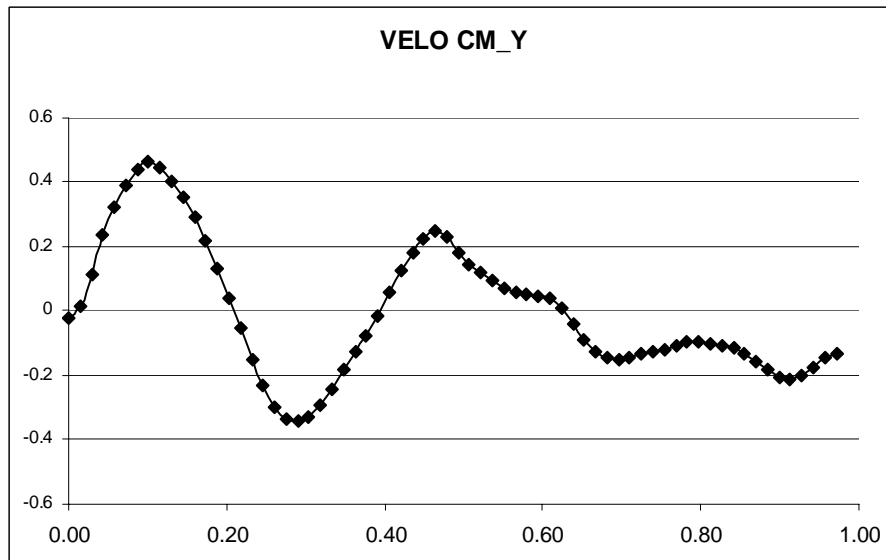
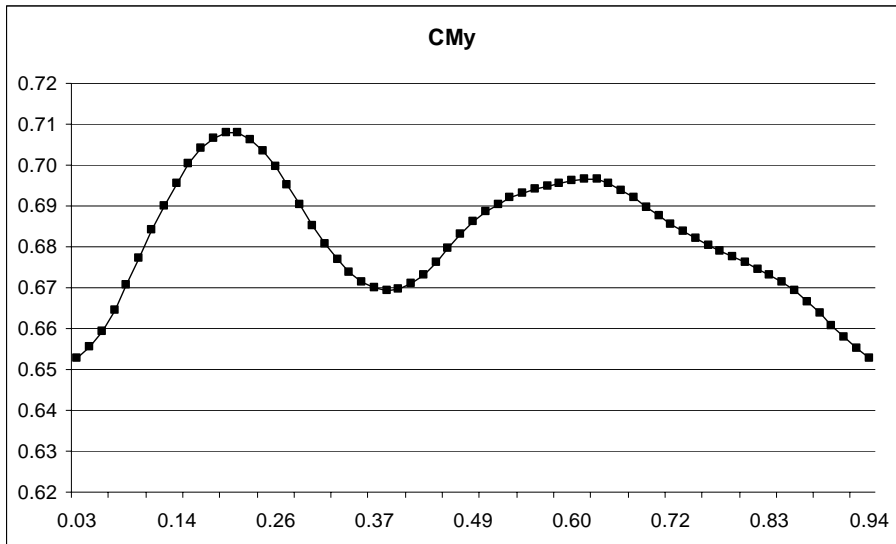
TIME =      0.9570
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
  1  1.771  0.653  1.362  2.148 -0.143  2.399  1.205  1.425 25.075

TIME =      0.9715
-----
BODY      X      Y      PHI      XD      YD      PHID      XDD      YDD      PHIDD
  1  1.802  0.651  1.398  2.157 -0.133  2.587  0.040  0.048  0.836

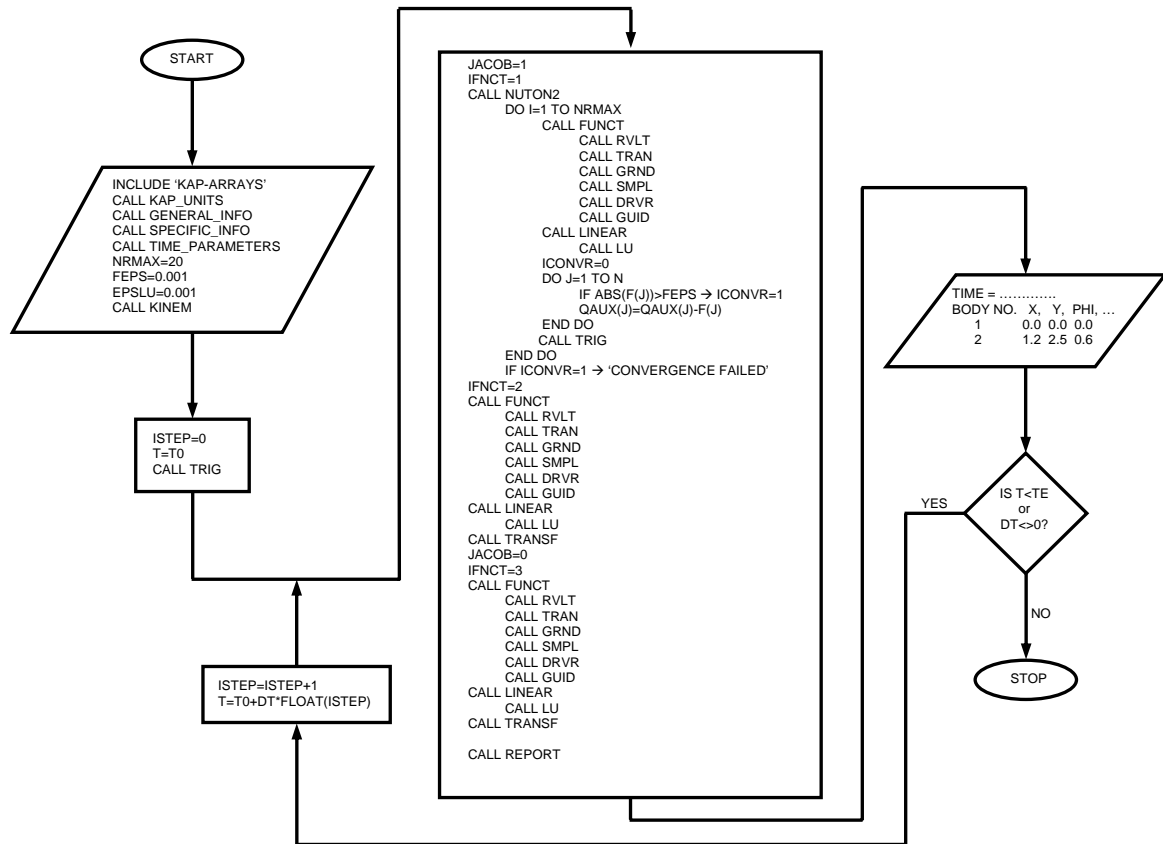
```

At each time step, the body numbers and the corresponding coordinates, velocities and accelerations of their points of interest are reported.

The three figures below illustrate respectively, the variation of position, velocity and acceleration for the Cartesian Coordinate Y along the time analysis.



9. FLOWCHART OF THE KABIOS PROGRAM



Flowchart for the KABIOS FORTRAN code.

10. ARRAYS USED IN THE KABIOS PROGRAM

$$Q = \begin{bmatrix} x_1 & x_2 & \dots & x_{NB} \\ y_1 & y_2 & \dots & y_{NB} \\ \phi_1 & \phi_2 & \dots & \phi_{NB} \end{bmatrix}_{(3 \times NB)} \quad QD = \begin{bmatrix} \dot{x}_1 & \dots \\ \dot{y}_1 & \dots \\ \dot{\phi}_1 & \dots \end{bmatrix}_{(3 \times NB)} \quad QDD = \begin{bmatrix} \ddot{x}_1 & \dots \\ \ddot{y}_1 & \dots \\ \ddot{\phi}_1 & \dots \end{bmatrix}_{(3 \times NB)}$$

$$RJ = \begin{bmatrix} \xi_i^P & \dots \\ \eta_i^P & \dots \\ \xi_j^P & \dots \\ \eta_j^P & \dots \end{bmatrix}_{(4 \times NR)} \quad IRJ = \begin{bmatrix} i & \dots \\ j & \dots \end{bmatrix}_{(2 \times NR)}$$

$$RT = \begin{bmatrix} \xi_i^P & \dots \\ \eta_i^P & \dots \\ \alpha_i & \dots \\ \beta_i & \dots \\ \xi_j^P & \dots \\ \eta_j^P & \dots \\ \phi_i^0 - \phi_j^0 & \dots \end{bmatrix}_{(7 \times NT)} \quad IRT = \begin{bmatrix} i & \dots \\ j & \dots \end{bmatrix}_{(2 \times NT)}$$

$$GR = [x_i \quad y_i \quad \phi_i \quad \dots]_{(3 \times NG)} \quad IGR = \begin{bmatrix} i & \dots \\ 1' & \dots \end{bmatrix}_{(2 \times NG)}$$

$$SM = [\dots \quad \dots \quad \dots \quad \dots]_{(NS)} \quad ISM = \begin{bmatrix} \dots \\ \dots \end{bmatrix}_{(2 \times NS)}$$

$$DR = \begin{bmatrix} \phi_i & \dots \\ \omega_i & \dots \\ \alpha_i & \dots \end{bmatrix}_{(3 \times ND)} \quad IDR = \begin{bmatrix} i & \dots \\ 1' & \dots \end{bmatrix}_{(2 \times ND)}$$

$$GUI = \begin{bmatrix} t_i & x_i & y_i & \phi_i & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}_{(NDATA \times NGUI + 1)} \quad IGUI = \begin{bmatrix} i & \dots \\ 1' & \dots \end{bmatrix}_{(2 \times NGUI)}$$

$$PI = \begin{bmatrix} \xi_i^P & \dots \\ \eta_i^P & \dots \end{bmatrix}_{(2 \times NP)} \quad IPI = [i \quad \dots]_{(NP)}$$

$$RB = \begin{bmatrix} \sin \phi_1 & \dots \\ \cos \phi_1 & \dots \end{bmatrix}_{(2 \times NB)}$$