10

A FORTRAN Program for Analysis of Planar Dynamics

In this chapter a FORTRAN program for planar dynamic analysis is presented. The program employs several subroutines from the kinematic analysis program (KAP) in Chap. 5 without any modifications. This program can model constant forces, gravity, and translational elements consisting of a spring, a damper, and/or an actuator. The program is organized in a form that allows it to be expanded to include other types of force elements. The problems at the end of this chapter provide a pattern to use for expanding the program.

Numerical methods for solving a system of mixed algebraic and differential equations, such as the equations of motion given in Sec. 9.4, are discussed in detail in Chaps. 12 and 13. However, in order to show how the dynamic analysis program (DAP) listed in this chapter solves the equations of motion, a brief discussion is provided in Sec. 10.1.

The listed program can solve the equations of motion for the dynamic response of constrained systems. In addition, this program can solve static problems as formulated in Secs. 9.5 and 9.6.

10.1 SOLVING THE EQUATIONS OF MOTION

For an unconstrained mechanical system, the equations of motion are given by

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{g} \tag{10.1}$$

with initial conditions on the coordinates and velocities given as $\mathbf{q}^0$ and $\dot{\mathbf{q}}^0$. Since $\mathbf{M}$ is a constant diagonal matrix and $\mathbf{g}$ can be a function of $\mathbf{q}$ and $\dot{\mathbf{q}}$, Eq. 10.1 can be solved for the unknowns $\ddot{\mathbf{q}}^0$ at the initial time.

The numerical integration algorithms that will be discussed in Chap. 12, can integrate the velocity and acceleration vectors at a given time and obtain the position and velocity vectors at a new time step. If the position and velocity vectors are appended together as the vector

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{10.2}$$

then the velocity and acceleration vectors are represented in the vector

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \tag{10.3}$$

At time $t = t^i$, vector $\dot{\mathbf{y}}^{(i)}$ can be integrated numerically to obtain $\mathbf{y}^{(i+1)}$, where $t^{i+1} = t^i + \Delta t$; i.e.,

$$\dot{\mathbf{y}}^{(i)} \xrightarrow{\text{(integrate)}} \mathbf{y}^{(i+1)} \tag{10.4}$$

Initially, at $i = 0$, the initial conditions on $\mathbf{q}$ and $\dot{\mathbf{q}}$ are required to start the integration process.

For a constrained mechanical system, the equations of motion are, from Eq. 9.55,

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{10.5}$$

with initial conditions $\mathbf{q}^0$ and $\dot{\mathbf{q}}^0$. The Jacobian $\boldsymbol{\Phi}_{\mathbf{q}}$ is a function of $\mathbf{q}$, and $\mathbf{g}$ and $\boldsymbol{\gamma}$ are functions of $\mathbf{q}$ and $\dot{\mathbf{q}}$ that can be evaluated at the initial time. Hence, Eq. 10.5 can be solved for the unknowns at the initial time, i.e., $\ddot{\mathbf{q}}^0$ and $\boldsymbol{\lambda}^0$.

The initial conditions on $\mathbf{q}$ and $\dot{\mathbf{q}}$ for a constrained system cannot be specified arbitrarily. The initial conditions $\mathbf{q}^0$ and $\dot{\mathbf{q}}^0$ *must* satisfy the constraint equations; i.e.,

$$\boldsymbol{\Phi} = \mathbf{0} \qquad (\text{for } \mathbf{q} = \mathbf{q}^0) \tag{10.6}$$

and

$$\boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0} \qquad (\text{for } \mathbf{q} = \mathbf{q}^0 \text{ and } \dot{\mathbf{q}} = \dot{\mathbf{q}}^0) \tag{10.7}$$

For the constrained equations of motion, vectors $\mathbf{y}$ and $\dot{\mathbf{y}}$ are as defined in Eqs. 10.2 and 10.3, and a numerical integration algorithm is applied to process Eq. 10.4. This is a simple but *crude* method of solving the constrained equations of motion. The possible error accumulation associated with this method and the techniques for resolving the problem are discussed in Chap. 13.

10.2 DYNAMIC ANALYSIS PROGRAM (DAP)

The main routine of the dynamic analysis program performs three major tasks:

1. Reads input data, either directly or by making calls to other input subroutines
2. Splits the working arrays A and IA into smaller subarrays by defining pointers
3. Makes calls to subroutine RUNG4 for integrating the equations of motion, or subroutine STATIC for static analysis

A detailed explanation of these tasks follows.

- **Input/Output:** Same as for KAP, Sec. 5.1.
- **Working Arrays:** Same as for KAP, Sec. 5.1.
- **Number of Elements:** The first set of data the program requests is

```
ENTER NB, NR, NT, NG, NS, NSP, NP
```

which 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 to be) ground
NS	Number of simple constraints in the system
NSP	Number of translational spring, damper, or actuator elements
NP	Number of points of interest

The program computes the number of coordinates N and the total number of constraint equations M from the above information. If M is greater than N, then an error message is given. Otherwise, the program continues.

- **Subarrays:** The working arrays A and IA are divided into smaller subarrays, according to the number of elements in the problem. The subarrays and their corresponding pointers and lengths are shown in Fig. 10.1. The function of the subarrays is explained in Secs. 10.2.1, 10.2.2, and 10.2.4.

- **Input Data:** The main program makes calls to other subroutines to read additional information for the problem at hand. These subroutines are discussed in Sec. 10.2.1.

- **Time Parameters:** Same as for KAP, Sec. 5.1.

- **Static Analysis:** If N = M, then a call is made to subroutine STATIC. For static analysis, the time parameters are not used.

- **Dynamic Analysis:** If N > M, then a call to subroutine RUNG4 is made to start the integration process for dynamic analysis.

The main routine for DAP is as follows:

```
C.....DYNAMIC ANALYSIS/STATIC ANALYSIS.....
C
C.....Main Program.....
C
      COMMON /CONST / NRMAX,FEPS,EPSLU
      COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
```

```
      COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
     +                N17,N18,N19,N20,N21,N22,N23,N24
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      COMMON /TIME  / T0,TE,DT,T
      DIMENSION A(3000),IA(500)
C.....If more storage space in A and IA arrays are needed increase the
C.....dimension and update MAXA and MAXIA accordingly
      MAXA =3000
      MAXIA=500
C.....Read number of bodies, revolute joints, translational joints,
C.....grounded bodies, simple constraints, spring-damper-actuators,
C.....points of interest
   10 WRITE(1,200)
      READ (1,*  ) NB,NR,NT,NG,NS,NSP,NP
C.....Determine number of coordinates N and number of constraint M
      N=3*NB
      M=2*(NR+NT)+3*NG+NS
      NPM=N+M
      NC2=N+N
C.....N must be greater to M
      IF (M.LE.N) GOTO 20
      WRITE(1,210) N,M
      GOTO 10
C.....Define pointers and split A and IA into subarrays
C.....Refer to Figure 10.1
   20 N1=1
      N2=N1+4*NR
      N3=N2+7*NT
      N4=N3+3*NG
      N5=N4+  NS
      N6=N5+12*NSP
      N7=N6+7*NB
      M1=1
      M2=M1+2*NR
      M3=M2+2*NT
      M4=M3+6*NG
      M5=M4+2*NS
      M7=M5+2*NSP
      N10=N7+2*NP
      N11=N10+N
      N12=N11+N
      N13=N12+N
      N14=N13+M
      N15=N14+N*M
      N16=N15+M
      N17=N16+NPM
      N18=N17+N
      N19=N18+M
      N20=N19+NPM*NPM
      N21=N20+NC2
      N22=N21+NC2
      N23=N22+NC2
      NUSED=N23+NC2-1
      M10=M7+NP
      MUSED=M10+NPM-1
C.....Check for sufficient storage space in A and IA arrays
      IF(NUSED.LE.MAXA .AND. MUSED.LE.MAXIA)GOTO 30
      WRITE(1,220) NUSED,MUSED
      STOP
C.....Rigid body information
   30 CALL INBODY (A(N10),A(N11),A(N6),NB)
C.....Read revolute joints data
      IF (NR.GT.0) CALL INRVLT (A(N1),IA(M1),NR)
```

Working Array A				Working Array IA			
Pointer	Subarray	Length	Description	Pointer	Subarray	Length	Description
N1	RJ	4*NR	Revolute joints	M1	IRJ	2*NR	Revolute joints
N2	TJ	7*NT	Translational joints	M2	ITJ	2*NT	Translational joints
N3	GR	3*NG	Ground	M3	IGR	6*NG	Ground
N4	SM	NS	Simple constraints	M4	ISM	2*NS	Simple constraints
N5	SP	12*NSP	Springs, dampers, actuators	M5	ISP	2*NSP	Springs, dampers, actuators
N6	RB	7*NB	Rigid bodies	M6			
N7	PI	2*NP	Points of interest	M7	IPI	NP	Points of interest
N10	Q	N	$\mathbf{q}$	M10	ICOL	N	Column pointers for $\mathbf{\Phi}_q$
N11	QD	N	$\dot{\mathbf{q}}$				
N12	QDD	N	$\ddot{\mathbf{q}}$				
N13	EL	M	$\boldsymbol{\lambda}$				
N14	FQ	M*N	$\mathbf{\Phi}_q$				
N15	F	M	$\mathbf{\Phi}$ ($\dot{\mathbf{\Phi}}$ or $\ddot{\mathbf{\Phi}}$ if needed)				
N16	W	N + M	Work array				
N17	FRC	N	$\mathbf{g}$				
N18	RHS	M	$\boldsymbol{\gamma}$				
N19	EM	(N + M)*(N + M)	$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_q^T \\ \mathbf{\Phi}_q & \mathbf{0} \end{bmatrix}$				
N20	Y	2*N	Runge-Kutta				
N21	YD	2*N	Runge-Kutta				
N22	YS	2*N	Runge-Kutta				
N23	FTOT	2*N	Runge-Kutta				

Figure 10.1 Subarrays of A and IA with their corresponding pointers and lengths.

```
C.....Read translational joints data
      IF (NT.GT.0) CALL INTRAN (A(N2),IA(M2),NT,A(N10),NB)
C.....Read ground constraints data
      NG3=3*NG
      IF (NG.GT.0) CALL INGRND (A(N3),IA(M3),NG,A(N10),NG3,NB)
C.....Read simple constraints data
      IF (NS.GT.0) CALL INSMPL (A(N4),IA(M4),NS,A(N10),NB)
C.....Read spring-damper-actuator elements data
      IF (NSP.GT.0) CALL INSPRG (A(N5),IA(M5),NSP)
C.....Read special points of interest
      IF (NP.GT.0) CALL INPOIN (A(N7),IA(M7),NP)
C.....Read initial time, final time, and time increments
      WRITE(1,230)
      READ (1,*  ) T0,TE,DT
C.....End of input data
      EPSLU=0.00001
C.....Static analysis
      IF (M.EQ.N) CALL STATIC (A,IA,MAXA,MAXIA)
C.....Start dynamic analysis
C.....Transfer Q and QD to YS
      CALL TRANSF (A(N22),A(N10),NC2)
C.....Start numerical integration
      CALL RUNG4 (A,IA,A(N20),A(N21),A(N22),A(N23),MAXA,MAXIA)
      STOP
  200 FORMAT(5X,'ENTER NB,NR,NT,NG,NS,NSP,NP')
  210 FORMAT(5X,'***INPUT ERROR*** N =',I3,'  M =',I3)
  220 FORMAT(5X,'***DIMENSION ON A AND/OR IA ARRAYS NOT SUFFICIENT***',
     +     /,10X,'MINIMUM DIMENSION ON A  MUST BE',I5,
     +     /,10X,'MINIMUM DIMENSION ON IA MUST BE',I5)
  230 FORMAT(5X,'ENTER TSTART, TEND, AND STEP')
      END
```

10.2.1 Model Description Subroutines

The following subroutines are called by the main routine of DAP to read the description of the model.

Subroutine INBODY. This subroutine reads initial conditions on x_i, y_i, and ϕ_i, initial velocities $\dot{x}_i$, $\dot{y}_i$, and $\dot{\phi}_i$, mass m_i, moment of inertia μ_i, constant external applied forces acting at the center of mass $f_{(x)_i}$ and $f_{(y)_i}$, and moments n_i. The prompt from this subroutine is repeated for each body i as follows:

```
FOR BODY i ENTER INITIAL COND. ON X, Y, PHI
    INITIAL CONDITIONS ON XD, YD, PHID
    MASS, MOMENT OF INERTIA
    CONSTANT FORCE-MOMENT FX, FY, N
```

The coordinates and velocities are stored in Q and QD (just as in KAP). The rest of the information is stored in array RB, dimensioned as RB(NB,7). For example, for body i,

$\sin \phi_i \rightarrow$ RB(I,1)
$\cos \phi_i \rightarrow$ RB(I,2)
$m_i \rightarrow$ RB(I,3)
$\mu_i \rightarrow$ RB(I,4)
$f_{(x)_i} \rightarrow$ RB(I,5)
$f_{(y)_i} - w_i \rightarrow$ RB(I,6)
$n_i \rightarrow$ RB(I,7)

The weight of each body is computed as $w_i = 9.81m_i$ (where the mass is in SI units) and is added in the negative y direction to $f_{(y)_i}$ (refer to Sec. 9.2.1).

Subroutine INBODY is as follows:

```
      SUBROUTINE INBODY (Q,QD,RB,NB)
      DIMENSION Q(3,NB),QD(3,NB),RB(NB,7)
      DO 10 I=1,NB
         WRITE(1,200) I
         READ (1,*  ) (Q(J,I),J=1,3),(QD(J,I),J=1,3),(RB(I,J),J=3,7)
   10    RB(I,6)=RB(I,6)-RB(I,3)*9.81
      RETURN
  200 FORMAT(5X,'FOR BODY',I4,' ENTER INITIAL COND. ON X, Y, PHI',/,
     +       10X,'INITIAL CONDITIONS ON XD, YD, PHID',/,
     +       10X,'MASS, MOMENT OF INERTIA',/,
     +       10X,'CONSTANT FORCE-MOMENT FX, FY, N')
      END
```

Subroutine INRVLT. Same as for KAP, Sec. 5.1.1.

Subroutine INTRAN. Same as for KAP, Sec. 5.1.1.

Subroutine INGRND. Same as for KAP, Sec. 5.1.1.

Subroutine INSMPL. Same as for KAP, Sec. 5.1.1.

Subroutine INSPRG. This subroutine is called if NSP > 0 to read information on spring, damper, and actuator elements (refer to Sec. 9.2.3, 9.2.4, and 9.2.5). An element can have one spring, one damper, and one actuator as long as the attachment points are shared. One possibility is that an element to contain only one spring and no damper or actuator. The prompt given by this subroutine is

```
FOR SPRING ELEMENT NO. k ENTER BODY NOS. I and J
   XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
   SPRING CONST., DAMPING COEF., ACTUATOR FORCE, UNDEFORMED SPRING
   LENGTH
```

This prompt is repeated for $k = 1, \ldots,$ NSP. The body numbers of bodies i and j, which are by definition those connected by the kth element, are stored in array ISP, dimensioned as ISP(NSP,2). The local coordinates of points P_i and P_j, the spring constant k, the damping coefficient d, the actuator force $f^{(a)}$, and the undeformed spring length l^0 are stored for each element in array SP, dimensioned as SP(NSP,12). For the kth element, the entries of SP are,

$$\xi_i^P \rightarrow \text{SP(K,1)} \qquad k \rightarrow \text{SP(K,5)}$$
$$\eta_i^P \rightarrow \text{SP(K,2)} \qquad d \rightarrow \text{SP(K,6)}$$
$$\xi_j^P \rightarrow \text{SP(K,3)} \qquad f^{(a)} \rightarrow \text{SP(K,7)}$$
$$\eta_j^P \rightarrow \text{SP(K,4)} \qquad l^0 \rightarrow \text{SP(K,8)}$$

The last four columns of SP, columns 9 through 12, are not used in this subroutine. They are used in subroutine SPRNG for storing the deformed length of the spring l, the time rate of change in length $\dot{l}$, the spring force, and the damper force. This information will be reported to the user by subroutine REPORT.

Subroutine INSPRG is as follows:

```
      SUBROUTINE INSPRG (SP,ISP,NSP)
      DIMENSION SP(NSP,12),ISP(NSP,2)
      DO 10 K=1,NSP
         WRITE(1,200) K
   10    READ (1,*  ) (ISP(K,L),L=1,2),(SP(K,L),L=1,8)
      RETURN
  200 FORMAT(5X,'FOR SPRING EL. NO.',I3,' ENTER BODY NOS. I AND J',/,
     +       10X,'XI-P-I,ETA-P-I,XI-P-J,ETA,P-J',/,
     +10X,'SPRING CONST., DAMPING COEF., ACTUATOR FORCE, SPRING LENGTH')
      END
```

Subroutine INPOIN. Same as for KAP, Sec. 5.1.1.

10.2.2 Dynamic Analysis

The dynamic analysis program (DAP) performs dynamic analysis by employing the methodology of Sec. 13.3. The N second-order differential equations of motion are converted into 2 * N first-order differential equations. A fourth-order Runge-Kutta algorithm is employed to solve the initial-value problem. A discussion on Runge-Kutta algorithms and other algorithms for solving initial value problems is provided in Chap. 12.

The Runge-Kutta algorithm uses the four arrays Y, YD, YS, and FTOT, each having a dimension of 2 * N. Vector $\mathbf{y}$ of Eq. 10.2 is stored in Y. At every time step, a copy of Y is saved in YS. Vector $\dot{\mathbf{y}}$ of Eq. 10.3 is stored in YD, and FTOT stores the sum of the functions evaluated by the algorithm.

Following the process of data input, the main program calls subroutine TRANSF to transfer $\mathbf{q}$ and $\dot{\mathbf{q}}$ from Y to YS. Then the main program calls subroutine RUNG4.

Subroutine TRANSF. Same as for KAP, Sec. 5.1.1.

Subroutine RUNG4. Subroutine RUNG4 evaluates $\dot{\mathbf{y}}$ four times in every time step Δt. This subroutine calls subroutine DIFEQN to evaluate $\dot{\mathbf{y}}$. The time variable T is incremented from TS to TE. At the beginning of every time step, a call is made to subroutine REPORT for reporting the result.

Subroutine RUNG4 is as follows:

```
      SUBROUTINE RUNG4 (A,IA,Y,YD,YS,FTOT,MAXA,MAXIA)
      COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
     +                N17,N18,N19,N20,N21,N22,N23,N24
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      COMMON /TIME  / T0,TE,DT,T
      DIMENSION A(MAXA),IA(MAXIA),Y(NC2),YD(NC2),YS(NC2),FTOT(NC2)
      T=T0
      DTH=0.5*DT
      TWODT=2.0*DT
      WRITE (1,200)
C.....Step 1.....
    1 TS=T
      DO 10 I=1,NC2
   10    Y(I)=YS(I)
      CALL DIFEQN (A,IA,MAXA,MAXIA)
```

```
      CALL REPORT (A,IA,A(N5),A(N6),A(N7),A(N10),A(N11),A(N12),
     +             IA(M7),T,MAXA,MAXIA)
      IF (T.GT.TE) RETURN
      DO 11 I=1,NC2
   11    FTOT(I)=DT*YD(I)
C.....Step 2......
      T=TS+DTH
      DO 20 I=1,NC2
   20    Y(I)=YS(I)+DTH*YD(I)
      CALL DIFEQN (A,IA,MAXA,MAXIA)
      DO 21 I=1,NC2
   21    FTOT(I)=FTOT(I)+TWODT*YD(I)
C.....Step 3......
      DO 30 I=1,NC2
   30    Y(I)=YS(I)+DTH*YD(I)
      CALL DIFEQN (A,IA,MAXA,MAXIA)
      DO 31 I=1,NC2
   31    FTOT(I)=FTOT(I)+TWODT*YD(I)
C.....Step 4......
      T=TS+DT
      DO 40 I=1,NC2
   40    Y(I)=YS(I)+DT*YD(I)
      CALL DIFEQN (A,IA,MAXA,MAXIA)
      DO 41 I=1,NC2
   41    FTOT(I)=FTOT(I)+DT*YD(I)
C.....Determine new values for Q and QD
      DO 50 I=1,NC2
   50    YS(I)=YS(I)+FTOT(I)/6.0
      GOTO 1
  200 FORMAT(///,10X,'***** DYNAMIC ANALYSIS  *****',//)
      END
```

Subroutine DIFEQN. This subroutine transfers the contents of Y to the arrays Q and QD prior to a call to subroutine DYNAM, by calling subroutine TRANSF. This transfer is necessary because subroutine RUNG4 modifies the contents of Y four times in every time step. Similarly, after the return from subroutine DYNAM, the contents of QD and QDD are transferred to YD.

Subroutine DIFEQN is as follows:

```
      SUBROUTINE DIFEQN (A,IA,MAXA,MAXIA)
      COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
     +                N17,N18,N19,N20,N21,N22,N23,N24
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      DIMENSION A(MAXA),IA(MAXIA)
C.....Transfer Y to Q and QD
      CALL TRANSF (A(N10),A(N20),NC2)
C.....Determine YD
      CALL DYNAM (A,IA,A(N15),A(N18),MAXA,MAXIA)
C.....Transfer QD and QDD to YD
      CALL TRANSF (A(N21),A(N11),NC2)
      RETURN
      END
```

Subroutine DYNAM. This subroutine can be considered the central station of DAP. The input to this subroutine is $\mathbf{q}$ and $\dot{\mathbf{q}}$ and the output is $\ddot{\mathbf{q}}$. This subroutine calls the subroutine FUNCT to evaluate the Jacobian matrix $\mathbf{\Phi}_\mathbf{q}$ and vector $\boldsymbol{\gamma}$, whereupon they are stored in arrays FQ and RHS, respectively.

A call to subroutine FORCE evaluates the vector of forces $\mathbf{g}$ which is stored in array FRC. Vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ are stored in FRC and RHS back to back, forming an N + M array in a form given by the right-hand side of Eq. 10.5.

A call to subroutine MASS generates the matrix at the left in Eq. 10.5. This matrix contains the diagonal matrix $\mathbf{M}$, along with the Jacobian matrix $\boldsymbol{\Phi}_\mathbf{q}$ and its transpose $\boldsymbol{\Phi}_\mathbf{q}^T$.

Finally, subroutine LINEAR is called to solve Eq. 10.5 for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. The results are transferred to arrays QDD and EL.

Subroutine DYNAM is as follows:

```
      SUBROUTINE DYNAM (A,IA,F,RHS,MAXA,MAXIA)
      COMMON /ANALYS/ JACOB,IFNCT
      COMMON /CONST / NRMAX,FEPS,EPSLU
      COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
     +                N17,N18,N19,N20,N21,N22,N23,N24
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      COMMON /TIME  / T0,TE,DT,T
      DIMENSION A(MAXA),IA(MAXIA),F(M),RHS(M)
C
C.....Calculate sine and cosine of rotational coordinates
      CALL TRIG (A(N6),A(N10),NB)
C.....Evaluate right-hand-side of acceleration equations.
      JACOB=1
      IFNCT=3
      CALL FUNCT (A,IA,MAXA,MAXIA,A(N10),A(N11),A(N14),A(N15),JACOB)
      DO 20 I=1,M
   20    RHS(I)=F(I)
C.....Evaluate forces
      CALL FORCE (A,IA,MAXA,MAXIA,A(N17),N)
C.....Evaluate mass matrix, Jacobian, Jacobian transpose
      CALL MASS  (A(N6),A(N14),A(N19),NB,N,M,NPM)
C.....Solve for accelerations and Lagrange multipliers
      CALL LINEAR (A(N19),A(N17),A(N16),IA(M10),NPM,1,EPSLU)
C.....Transfer accs. and Lag. mults. to QDD and EL
      CALL TRANSF (A(N12),A(N17),NPM)
      RETURN
      END
```

Subroutine TRIG. Same as in KAP, Sec. 5.1.2 except that the dimension of RB must be changed to RB(NB,7).

Subroutine MASS is as follows:

```
      SUBROUTINE MASS (RB,FQ,EM,NB,N,M,NPM)
      DIMENSION RB(NB,7),FQ(M,N),EM(NPM,NPM)
C.....Initialize mass matrix
      DO 10 I=1,NPM
         DO 10 J=1,NPM
   10       EM(I,J)=0.0
C.....Add mass and moment of inertia to diagonal
      DO 20 I=1,NB
         J=3*(I-1)+1
         EM(J  ,J  )= RB(I,3)
         EM(J+1,J+1)= RB(I,3)
   20    EM(J+2,J+2)= RB(I,4)
```

```
C.....Include Jacobian and Jacobian transpose
      DO 30 I=1,M
         II=N+I
         DO 30 J=1,N
            EM(II,J )= FQ(I,J)
30          EM(J ,II)= FQ(I,J)
      RETURN
      END
```

Subroutines LINEAR and LU. See Sec. 3.3.5.

10.2.3 Function Evaluation

- **Subroutine FUNCT:** Same as in KAP, Sec. 5.2.3, but the call to subroutine DRVR is deleted.
- **Subroutine RVLT:** Same as in KAP, Sec. 5.2.3.
- **Subroutine TRAN:** Same as in KAP, Sec. 5.2.3.
- **Subroutine SMPL:** Same as in KAP, Sec. 5.2.3.

Note that in DAP the control flag IFNCT is never set to 1 or 2 (in contrast to the case for KAP). Therefore, the constraint equations and the right side of the velocity equations are never evaluated in the three subroutines just named. Only vector γ is evaluated when IFNCT is set to 3 in subroutine DYNAM. Several methods are discussed in Chap. 13 for controlling the accumulation of numerical errors during the numerical integration process. If DAP is modified to incorporate such a technique as the constraint violation stabilization method (refer to Chap. 13), then IFNCT = 1 may be used in order to provide the violation in the constraints for the algorithm.

10.2.4 Force Evaluation

Subroutine FORCE. The vector of forces is evaluated by a call to subroutine FORCE. All of the external and internal forces and moments acting on the bodies are stored in array FRC, dimensioned as FRC(3,NB). For example, the forces acting on body i are stored as follows:

$$f_{(x)_i} \rightarrow \text{FRC(1,I)}$$
$$f_{(y)_i} \rightarrow \text{FRC(2,I)}$$
$$n_i \rightarrow \text{FRC(3,I)}$$

This subroutine calls subroutines BODYF and SPRNG to evaluate these forces.

Subroutine FORCE is as follows:

```
SUBROUTINE FORCE (A,IA,MAXA,MAXIA,FRC,N)
COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
+               N17,N18,N19,N20,N21,N22,N23,N24
DIMENSION A(MAXA),IA(MAXIA),FRC(N)
          CALL BODYF (A(N6),FRC,NB)
IF (NSP.GT.0) CALL SPRNG (A(N5),A(N6),A(N10),A(N11),A(N17),
+                         IA(M5),NB,NSP)
RETURN
END
```

Subroutine BODYF. This subroutine transfers the constant external forces and moments, including the gravitational force, from RB to FRC, for all of the bodies.

Subroutine BODYF is as follows:

```
      SUBROUTINE BODYF (RB,FRC,NB)
      DIMENSION RB(NB,7),FRC(3,NB)
C.....Add constant forces and weights to FRC
      DO 10 I=1,NB
         FRC(1,I)= RB(I,5)
         FRC(2,I)= RB(I,6)
   10    FRC(3,I)= RB(I,7)
      RETURN
      END
```

Subroutine SPRNG. This subroutine computes the forces of spring, damper, and actuator elements between pairs of bodies. For each element, the spring constant k, damping coefficient d, actuator force $f^{(a)}$, and undeformed spring length l^0 are obtained from array SP. The body numbers connected by each element are found in array ISP. The element forces are calculated from the equations of Sec. 9.2.3, 9.2.4, and 9.2.5. The element forces and moments acting on each body are entered in array FRC.

This subroutine saves the values of l, $\dot{l}$, $f^{(s)}$, and $f^{(d)}$ for each element in the last four entries of array SP. This information is reported by subroutine REPORT at each time step.

Subroutine SPRNG is as follows:

```
      SUBROUTINE SPRNG(SP,RB,Q,QD,FRC,ISP,NB,NSP)
      DIMENSION SP(NSP,12),ISP(NSP,2),RB(NB,7),FRC(3,NB),Q(3,NB),
     +          QD(3,NB)
      DO 10 K=1,NSP
         I=ISP(K,1)
         J=ISP(K,2)
         XPIMXI=SP(K,1)*RB(I,2)-SP(K,2)*RB(I,1)
         YPIMYI=SP(K,1)*RB(I,1)+SP(K,2)*RB(I,2)
         XPJMXJ=SP(K,3)*RB(J,2)-SP(K,4)*RB(J,1)
         YPJMYJ=SP(K,3)*RB(J,1)+SP(K,4)*RB(J,2)
C.....Current spring length and change of length
         ELX   =Q(1,J)+XPJMXJ-Q(1,I)-XPIMXI
         ELY   =Q(2,J)+YPJMYJ-Q(2,I)-YPIMYI
         EL = SQRT(ELX**2 +ELY**2)
         DELEL =EL-SP(K,8)
         IF(ABS(EL).LT.1.E-10) EL=1.E-10
C......Unit vector
         UX    =ELX/EL
         UY    =ELY/EL
C.....Spring velocity
         ELXD=QD(1,J)-YPJMYJ*QD(3,J)-QD(1,I)+YPIMYI*QD(3,I)
         ELYD=QD(2,J)+XPJMXJ*QD(3,J)-QD(2,I)-XPIMXI*QD(3,I)
         ELD = (ELX*ELXD + ELY*ELYD)/EL
C.....Element forces
         FS=SP(K,5)*DELEL
         FD=SP(K,6)*ELD
         FF=FS+FD+SP(K,7)
         FX= UX*FF
         FY= UY*FF
C.....Save element length, vel., spr. force, damp. force for output
         SP(K, 9)=EL
         SP(K,10)=ELD
         SP(K,11)=FS
         SP(K,12)=FD
```

```
C.....Add forces to the vector of forces
      FRC(1,I)=FRC(1,I)+FX
      FRC(2,I)=FRC(2,I)+FY
      FRC(3,I)=FRC(3,I)-YPIMYI*FX+XPIMXI*FY
      FRC(1,J)=FRC(1,J)-FX
      FRC(2,J)=FRC(2,J)-FY
10    FRC(3,J)=FRC(3,J)+YPJMYJ*FX-XPJMXJ*FY
      RETURN
      END
```

10.2.5 Reporting

For reporting the result of the dynamic analysis at the beginning of each time step, subroutine RUNG4 calls subroutine REPORT.

Subroutine REPORT. This subroutine is similar to subroutine REPORT in KAP. It reports the coordinates, velocities, and accelerations of the bodies and points of interest. In addition, if there are any spring, damper, or actuator elements in the system, the contents of the last four columns of the SP array are reported. This subroutine calls subroutine REACT to calculate and report the reaction forces at the kinematic joints.

Subroutine REPORT is as follows:

```
      SUBROUTINE REPORT (A,IA,SP,RB,PI,Q,QD,QDD,IPI,T,MAXA,MAXIA)
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      DIMENSION SP(NSP,12),Q(3,NB),QD(3,NB),QDD(3,NB),PI(NP,2),
     +          IPI(NP),RB(NB,7),A(MAXA),IA(MAXIA)
      WRITE(1,200) T
      DO 10 I=1,NB
   10 WRITE(1,210)I,(Q(J,I),J=1,3),(QD(J,I),J=1,3),(QDD(J,I),J=1,3)
      IF (NP.EQ.0) GO TO 30
      WRITE(1,220)
      DO 20 K=1,NP
         I=IPI(K)
         XPMX=PI(K,1)*RB(I,2)-PI(K,2)*RB(I,1)
         YPMY=PI(K,1)*RB(I,1)+PI(K,2)*RB(I,2)
         XP  =Q(1,I)+XPMX
         YP  =Q(2,I)+YPMY
         XDP =QD(1,I)-YPMY*QD(3,I)
         YDP =QD(2,I)+XPMX*QD(3,I)
         XDDP=QDD(1,I)-XPMX*QD(3,I)**2-YPMY*QDD(3,I)
         YDDP=QDD(2,I)-YPMY*QD(3,I)**2+XPMX*QDD(3,I)
   20    WRITE(1,260) K,XP,YP,XDP,YDP,XDDP,YDDP
   30 IF(NSP.EQ.0) GO TO 50
      WRITE(1,240)
      DO 40 K=1,NSP
   40 WRITE(1,250)K,SP(K,9),SP(K,10),SP(K,11),SP(K,12)
   50 IF (M.GT.0) CALL REACT(A,IA,MAXA,MAXIA)
      RETURN
  200 FORMAT(/,' TIME =',F10.4,/,' ----------------',/,
     +       ' BODY',5X,'X',7X,'Y',5X,'PHI',6X,'XD',6X,'YD',4X,'PHID',6X,
     +       'XDD',6X,'YDD',4X,'PHIDD')
  210  FORMAT(I3,6F8.3,3F9.3)
  220 FORMAT(' POINTS OF INTEREST',/,'  NO.',6X,'X',7X,'Y',6X,
     +       'XD',6X,'YD',6X,'XDD',6X,'YDD')
  240 FORMAT(' TRANSLATIONAL SPRING-DAMPER-ACTUATOR',/,
     +       '  NO.  LENGTH  d(EL)/dt',6X,'f(s)',6X,'f(d)')
  250 FORMAT(I3,4F10.3)
  260 FORMAT(I3,4F8.3,2F9.3)
      END
```

Subroutines REACT and RFORCE. Subroutine REACT calls subroutine RFORCE for computing the product $\Phi_q^T \boldsymbol{\lambda}$. This product is computed separately for each kinematic joint. Some of the variables in the argument list of subroutine RFORCE are dependent on the type of joint. For example, for a revolute joint, NEQ, which represents the number of equations, is set to 2, and NBJ, which represents the number of bodies associated with the joint, is also set to 2.

Subroutine RFORCE reports the reaction forces acting on each body that are due to that body's kinematic joints. A reaction force is reported in terms of its x and y components and its moment with respect to the centroid of the body. The reported reaction forces are labeled with REV. for a revolute joint, TRA. for a translational joint, and SMP. for a simple constraint.

Subroutines REACT and RFORCE are as follows:

```
      SUBROUTINE REACT(A,IA,MAXA,MAXIA)
      COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,
     +                N15,N16,N17,N18,N19,N20,N21,N22,N23,N24
      DIMENSION A(MAXA),IA(MAXIA)
      JR=0
      WRITE(1,200)
      IF(NR.GT.0) CALL RFORCE(A(N13),A(N14),IA(M1),NR,2,2,JR,'REV.')
      IF(NT.GT.0) CALL RFORCE(A(N13),A(N14),IA(M2),NT,2,2,JR,'TRA.')
      JR=JR+NG3
      IF(NS.GT.0) CALL RFORCE(A(N13),A(N14),IA(M4),NS,1,1,JR,'SMP.')
      RETURN
  200 FORMAT(' REACTION FORCES',/,2X,'JOINT NO. I ',5X,'FX-I',6X,
     +       'FY-I',7X,'N-I',3X,'J ',5X,'FX-J',6X,'FY-J',7X,'N-J')
      END

      SUBROUTINE RFORCE(EL,FQ,IJ,NJ,NEQ,NBJ,JR,NAME)
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      CHARACTER*4 NAME
      DIMENSION FQ(M,N),EL(M),IJ(NJ,NBJ),F(3,2)
      DO 20 K=1,NJ
         DO 10 L=1,NBJ
            I =IJ(K,L)
            IC=(I-1)*3
            DO 10 J=1,3
               F(J,L)=0.0
               DO 10 MM=1,NEQ
   10             F(J,L)=F(J,L)-FQ(JR+MM,IC+J)*EL(JR+MM)
         WRITE(1,200) NAME,K,(IJ(K,L),(F(J,L),J=1,3),L=1,NBJ)
   20    JR=JR+NEQ
      RETURN
  200 FORMAT(2X,A4,I3,I4,3F10.3,I4,3F10.3)
      END
```

10.2.6 Static Analysis

When the number of constraint equations is equal to the number of coordinates, the main program calls subroutine STATIC to perform static analysis. This subroutine calls subroutine FUNCT to evaluate the Jacobian matrix Φ_q, calls subroutine FORCE to evaluate the vector of forces $\mathbf{g}$, and then calls subroutine JACTRN to transpose the Jacobian matrix. A call to subroutine LINEAR solves for the Lagrange multipliers according to Eq. 9.57. The constraint reaction forces are then reported by a call to subroutine REPORTS.

Subroutines STATIC, JACTRN, and REPORTS are as follows:

```
      SUBROUTINE STATIC (A,IA,MAXA,MAXIA)
      COMMON /ANALYS/ JACOB,IFNCT
      COMMON /CONST / NRMAX,FEPS,EPSLU
      COMMON /MPNTR / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /NPNTR / N1,N2,N3,N4,N5,N6,N7,N10,N11,N12,N13,N14,N15,N16,
     +                N17,N18,N19,N20,N21,N22,N23,N24
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      DIMENSION A(MAXA),IA(MAXIA)
C.....Calculate sine and cosine of rotational coordinates
      CALL TRIG (A(N6),A(N10),NB)
C.....Evaluate Jacobian matrix
      JACOB=1
      IFNCT=0
      CALL FUNCT (A,IA,MAXA,MAXIA,A(N10),A(N11),A(N14),A(N15),JACOB)
C.....Evaluate forces
      CALL FORCE (A,IA,MAXA,MAXIA,A(N17),N)
C.....Jacobian transpose
      CALL JACTRN(A(N14),A(N19),M)
C.....Solve for Lagrange multipliers
      CALL LINEAR (A(N19),A(N17),A(N16),IA(M10),M,1,EPSLU)
C.....Transfer Lagrange multipliers to EL
      CALL TRANSF (A(N13),A(N17),M)
C.....Output the result
      CALL REPORTS (A,IA,A(N10),MAXA,MAXIA)
      STOP
      END


      SUBROUTINE JACTRN (FQ,CM,M)
      DIMENSION FQ(M,M),CM(M,M)
      DO 10 I=1,M
         DO 10 J=1,M
   10       CM(I,J)=FQ(J,I)
      RETURN
      END


      SUBROUTINE REPORTS (A,IA,Q,MAXA,MAXIA)
      COMMON /NELMNT/ NB,NR,NT,NG,NG3,NS,NSP,NP
      COMMON /ROWCOL/ IR,IC,M,N,NPM,NC2
      DIMENSION Q(3,NB),A(MAXA),IA(MAXIA)
      WRITE (1,200)
      DO 10 I=1,NB
   10    WRITE(1,210)I,(Q(J,I),J=1,3)
      CALL REACT(A,IA,MAXA,MAXIA)
      RETURN
  210 FORMAT(I3,3F8.3)
  200 FORMAT(///,10X,'*****  STATIC ANALYSIS  *****',//,
     +           ' BODY',5X,'X',7X,'Y',5X,'PHI')
      END
```

10.2.7 Input Prompts

A list of all the prompts given by the program DAP is given here. The prompts are labeled for easy reference, from (a) through (i). In the examples that follow, each prompt is referred to by its corresponding label. The parameter k in the prompts is problem-dependent.

The prompts given are as follows:

(a) ENTER NB, NR, NT, NG, NS, NSP, NP

(b) FOR BODY k ENTER INITIAL COND. ON X, Y, PHI
INITIAL CONDITIONS ON XD, YD, PHID
MASS, MOMENT OF INERTIA
CONSTANT FORCE-MOMENT FX, FY, N

(c) FOR REV. JOINT NO. k ENTER BODY NOS. I AND J
XI-P-I, ETA-P-I, XI-P-J, ETA-P-J

(d) FOR TRAN. JOINT 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 GROUNDED BODY

(f) FOR SIMPLE CONSTRAINT NO. k ENTER BODY NO.
AND 1, 2, OR 3 FOR X, Y, OR PHI CONSTRAINT DIRECTION

(g) FOR SPRING EL. NO. k ENTER BODY NOS. I AND J
XI-P-I, ETA-P-I, XI-P-J, ETA-P-J
SPRING CONST., DAMPING COEF., ACTUATOR FORCE, UNDEFORMED
SPRING LENGTH

(h) FOR POINT OF INTEREST NO. k ENTER BODY NO.
XI-P AND ETA-P COORDINATES

(i) ENTER STARTING TIME, FINAL TIME, AND TIME INCREMENT

10.3 SIMPLE EXAMPLES

In Secs. 10.3.1 through 10.3.3 several simple examples are presented. The steps needed to set up a model for each mechanical system are explained. Input data for the dynamic analysis program (DAP) are listed for each example. Similar steps can be followed to analyze many other mechanical systems by means of this program.

10.3.1 Four-Bar Linkage

The four-bar linkage of Sec. 5.2.1 is considered here. The mechanism is released from an initial position, where all of the initial velocities are zero, and falls under its own weight. The mass and moment of inertia for the moving bodies are:

$$m_2 = 1, \qquad m_3 = 2.25, \qquad m_4 = 2$$
$$\mu_2 = 0.3, \qquad \mu_3 = 2, \qquad \mu_4 = 1.35$$

For body 1, the nonmoving body, any arbitrary value can be assigned for the mass and moment of inertia.

When DAP is executed, the following sequence of prompts is given by the program; each prompt must be followed by input from the user to describe the model:

Prompt	k	Input
(a)		4, 4, 0, 1, 0, 0, 1
(b)	1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
(b)	2	.5, .866, 1.047, 0, 0, 0, 1, .3, 0, 0, 0
(b)	3	2.824, 2.553, .423, 0, 0, 0, 2.25, 2, 0, 0, 0
(b)	4	3.574, 1.687, 1.004, 0, 0, 0, 2, 1.35, 0, 0, 0
(c)	1	1, 2, 0, −1, 0
(c)	2	2, 3, 1, 0, −2, 0
(c)	3	3, 4, 2, 0, 2, 0
(c)	4	4, 1, −2, 0, 2.5, 0
(e)	1	1
(h)	1	3, 0.5, 1.5
(i)		0.0, 0.25, 0.025

The initial conditions for the coordinates satisfy the constraint equations as stated by Eq. 10.6. This is assured, since these coordinates are taken from the output of KAP at $t = 0$ for the same linkage, as given in Sec. 5.2.1. The initial conditions for the velocities do not violate the velocity equations, since $\dot{\mathbf{q}} = \mathbf{0}$ automatically satisfies Eq. 10.7.

The output of DAP for the first time step is as follows:

```
***** DYNAMIC ANALYSIS  *****

TIME =      .0000
----------------
 BODY      X         Y       PHI       XD        YD       PHID      XDD       YDD       PHIDD
  1      .000      .000      .000      .000      .000      .000      .000      .000      .000
  2      .500      .866     1.047      .000      .000      .000     2.544    -1.470    -2.938
  3     2.824     2.553      .423      .000      .000      .000     5.183    -3.149     -.115
  4     3.574     1.687     1.004      .000      .000      .000     2.639    -1.679    -1.564
 POINTS OF INTEREST
  NO.       X         Y        XD        YD       XDD       YDD
  1     2.664     4.126      .000      .000     5.364    -3.131
 REACTION FORCES
  JOINT NO.  I       FX-I       FY-I        N-I    J       FX-J       FY-J        N-J
  REV.   1   1     -7.242    -15.387       .000    2      7.242     15.387     -1.425
  REV.   2   2     -4.698     -7.046       .543    3      4.698      7.046     -8.994
  REV.   3   3      6.964      7.941      8.764    4     -6.964     -7.941      3.223
  REV.   4   4     12.242     24.202     -5.334    1    -12.242    -24.202    -60.504
```

The output gives the coordinates, velocity, and acceleration of each body's centroid and of each point of interest. The reaction forces corresponding to each joint are also printed; these are shown in Fig. 10.2.

10.3.2 Horizontal Platform

The horizontal platform shown in Fig. 10.3(a) is connected to ground by four slender legs through eight revolute joints. The mass and moment of inertia of the platform

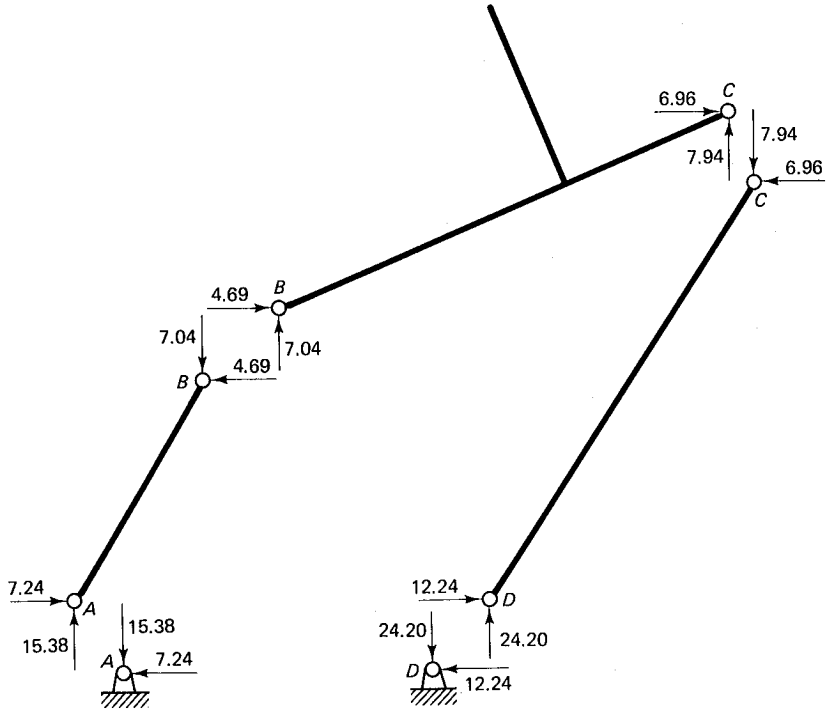


Figure 10.2 Reaction forces acting on each body at $t = 0$.

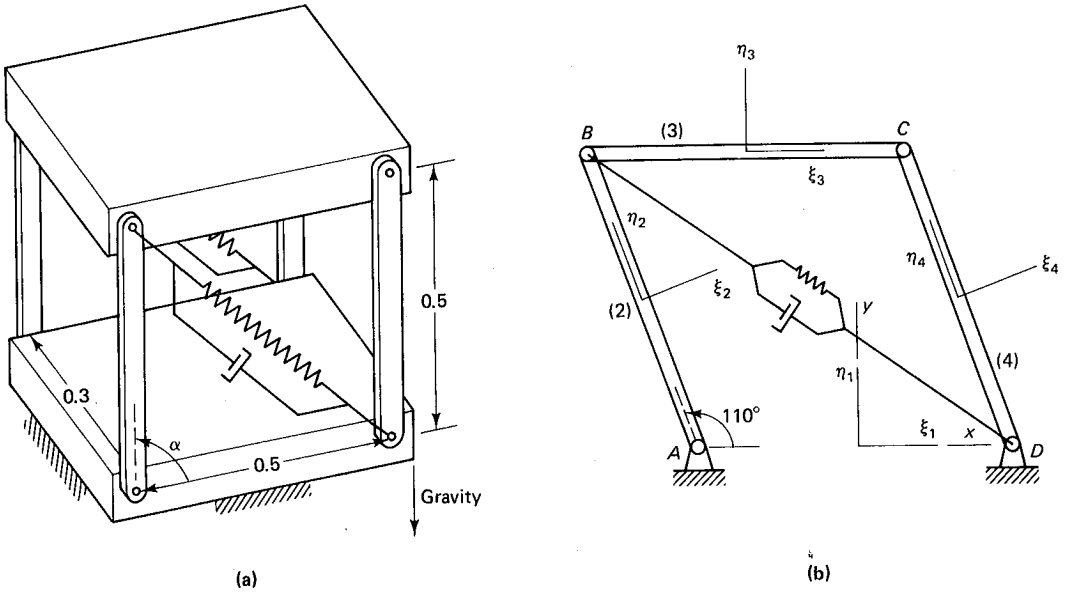


Figure 10.3 (a) A horizontal platform and (b) its corresponding planar model.

(about the ζ-axis out of the plane of motion) are 1.5 and 0.2, respectively. The mass and moment of inertia of each leg are 0.3 and 0.05, respectively. There are two parallel spring-damper elements in the system, as shown. For each element, $k = 350$, $l^0 = 0.6$, and $d = 25$. When the angle of each leg with the horizontal is $\alpha = 110°$, the platform is released with an upward velocity of $\dot{y} = 0.12$.

A planar model for this system is shown in Fig. 10.3(b). In this model, the masses and moments of inertia are:

$$m_2 = m_4 = 0.6, \qquad m_3 = 1.5$$
$$\mu_2 = \mu_4 = 0.1, \qquad \mu_3 = 0.2$$

For the spring-damper element:

$$k = 700, \qquad l^0 = 0.6, \qquad d = 50$$

Since the initial conditions for $\mathbf{q}$ and $\dot{\mathbf{q}}$ are not available, a simulation on KAP for $t = 0$ can be performed initially. In this simulation the known initial conditions are the vertical coordinate and velocity of the platform. From $\alpha = 110°$, it is found that $y_3 = 0.46895$, and it is given that $\dot{y}_3 = 0.12$. Therefore, a driver constraint can be specified on y_3. The input to KAP is

Prompt (KAP)	k	Input
(a)		4, 4, 0, 1, 0, 1, 0
(b)	1	0, 0, 0
(b)	2	−0.35, 0.2, 0.3
(b)	3	−0.1, 0.46895, 0
(b)	4	0.15, 0.2, 0.3
(c)	1	1, 2, −0.25, 0, 0, −0.25
(c)	2	2, 3, 0, 0.25, −0.25, 0
(c)	3	3, 4, 0.25, 0, 0, 0.25
(c)	4	4, 1, 0, −0.25, 0.25, 0
(e)	1	1
(g)	1	3, 2, 0.46895, 0.12, 0
(i)		0.0, 0.0, 0.1

The output from the kinematic analysis at $t = 0$ is

```
***** KINEMATIC ANALYSIS *****

TIME =     .0000
---------------
 BODY      X        Y       PHI       XD       YD      PHID      XDD      YDD     PHIDD
   1     .000     .000     .000     .000     .000     .000     .000     .000      .000
   2    -.335     .235     .349     .165     .060    -.702     .360     .000     -1.353
   3    -.171     .470     .000     .330     .120     .000     .720     .000      .000
   4     .165     .235     .349     .165     .060    -.702     .360     .000     -1.353
```

The initial conditions for $\mathbf{q}$ and $\dot{\mathbf{q}}$ are taken from this output to generate the input to DAP:

Prompt (DAP)	k	Input
(a)		4, 4, 0, 1, 0, 1, 0
(b)	1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
(b)	2	−.335, .235, .349, .165, .060, −.702, .6, .1, 0, 0, 0
(b)	3	−.171, .46895, .000, .330, .120, .000, 1.5, .2, 0, 0, 0
(b)	4	.165, .235, .349, .165, .060, −.702, .6, .1, 0, 0, 0
(c)	1	1, 2, −.25, 0, 0, −.25
(c)	2	2, 3, 0, .25, −.25, 0
(c)	3	3, 4, .25, 0, 0, .25
(c)	4	4, 1, 0, −.25, .25, 0
(e)	1	1
(g)	1	1, 3, 0.25, 0, −0.25, 0, 700, 50, 0, 0.6
(i)		0, 3, 0.01

The result of this simulation for $t = 0$ and $t = 3$ s is as follows:

***** DYNAMIC ANALYSIS *****

TIME = .0000

BODY	X	Y	PHI	XD	YD	PHID	XDD	YDD	PHIDD
1	.000	.000	.000	.000	.000	.000	.000	.000	.000
2	-.335	.235	.349	.165	.060	-.702	13.630	4.829	-57.838
3	-.171	.470	.000	.330	.120	.000	27.260	9.658	.000
4	.165	.235	.349	.165	.060	-.702	13.630	4.829	-57.838

TRANSLATIONAL SPRING-DAMPER-ACTUATOR

NO.	LENGTH	f(s)	f(d)
1	.819	-.201	153.462 -10.072

REACTION FORCES

JOINT NO.	I	FX-I	FY-I	N-I	J	FX-J	FY-J	N-J
REV. 1	1	45.067	-105.648	26.412	2	-45.067	105.648	-1.556
REV. 2	2	53.245	-96.865	-4.228	3	-53.245	96.865	-24.216
REV. 3	3	-23.310	14.601	3.650	4	23.310	-14.601	-4.228
REV. 4	4	-15.132	23.384	-1.556	1	15.132	-23.384	-5.846

TIME = 3.0000

BODY	X	Y	PHI	XD	YD	PHID	XDD	YDD	PHIDD
1	.000	.000	.000	.000	.000	.000	.000	.000	.000
2	-.173	.238	-.311	.000	.000	.000	.000	.000	.000
3	.153	.476	.000	.000	.000	.000	.000	.000	.000
4	.327	.238	-.311	.000	.000	.000	.000	.000	.000

TRANSLATIONAL SPRING-DAMPER-ACTUATOR

NO.	LENGTH	f(s)	f(d)
1	.589	.000	-7.805 .000

REACTION FORCES

JOINT NO.	I	FX-I	FY-I	N-I	J	FX-J	FY-J	N-J
REV. 1	1	-1.283	-6.933	1.733	2	1.283	6.933	-.225
REV. 2	2	-1.283	-1.047	.225	3	1.283	1.047	-.262
REV. 3	3	3.311	7.358	1.839	4	-3.311	-7.358	.225
REV. 4	4	3.311	13.244	-.225	1	-3.311	-13.244	-3.311

At $t = 3$ s, all of the velocities and accelerations are zero. This indicates that the system reaches the state of static equilibrium within 3 seconds as a result of the presence of dampers in the system. Figure 10.4 shows the plots of y_3, $\dot{y}_3$, spring force, and damper force as a function of time.

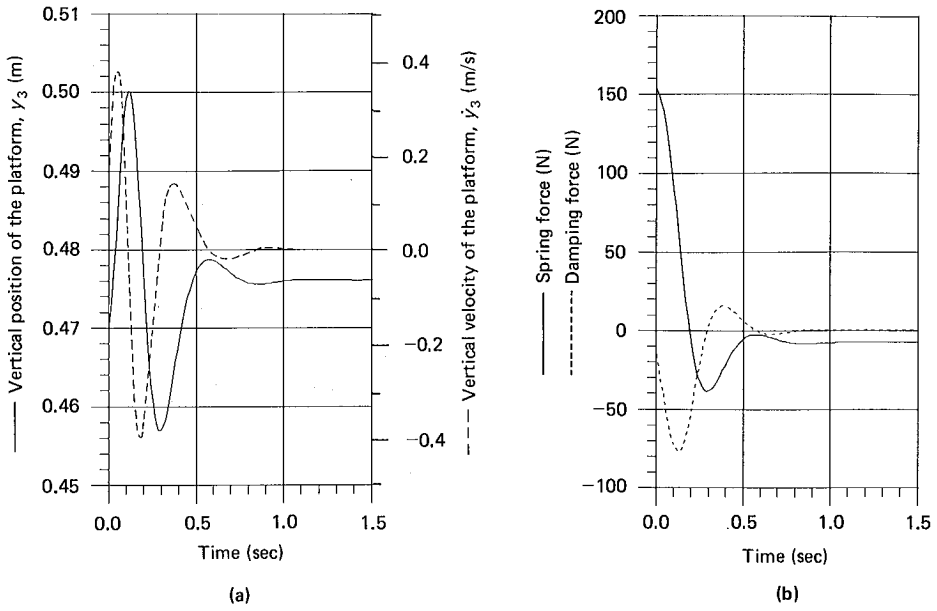
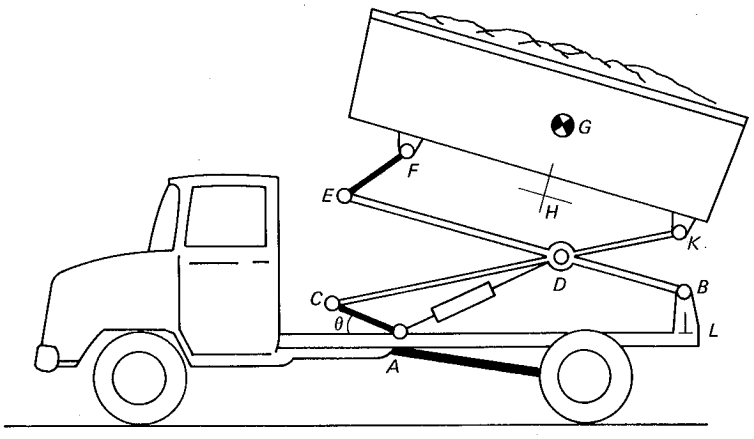


Figure 10.4 The dynamic response for the horizontal platform. (a) Plots of y displacement and velocity of the platform, and (b) spring and damper forces versus time.

10.3.3 Dump Truck

A hydraulic actuator controls the unloading process of a dump truck. For the configuration shown in Fig. 10.5 ($\theta = 0.38$ rad), find the force that the actuator must apply be-



$AL = 1.6$
$AC = EF = GH = 0.4$
$BL = 0.2$
$CD = DE = 1.3$
$CK = BE = 2$
$FH = HK = 0.8$

Figure 10.5 A dump truck in unloading configuration.

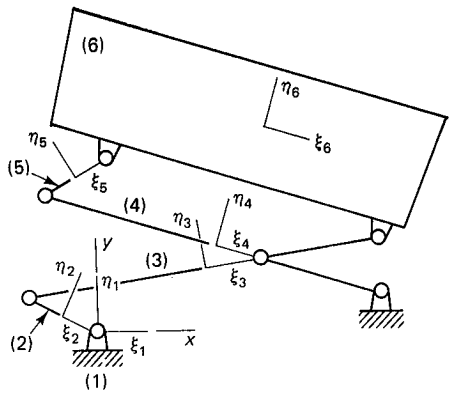
tween points A and D in order to keep the system in equilibrium. Masses and moments of inertia for the bodies are:

$$m_{EF} = m_{AC} = 0.4, \qquad m_{CK} = m_{BE} = 2, \qquad m_{\text{load}} = 100$$
$$\mu_{EF} = \mu_{AC} = 0.005, \qquad \mu_{CK} = \mu_{BE} = 0.7, \qquad \mu_{\text{load}} = 27$$

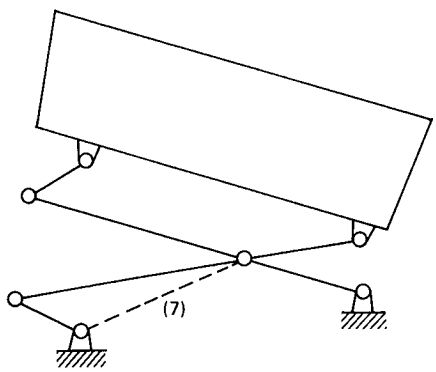
Assume that the center of mass of each body is at its geometric center, and that the center of mass of the load is at G.

First, as shown in Fig. 10.6(a), a model is set up for KAP to determine the correct initial conditions on the coordinates. This model is executed on KAP for $t = 0$, with the known value of ϕ_2, which is found from the specified θ. Since the system is not in motion, a constraint on ϕ_2 can be stated in the form of either a simple constraint or a driver constraint. The input to KAP is as follows:

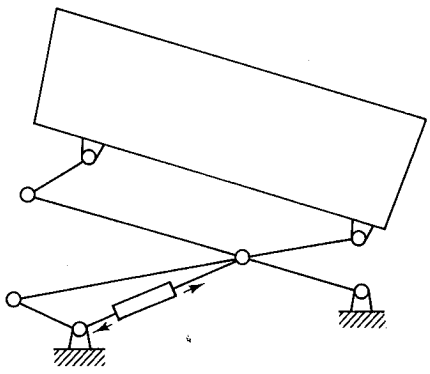
Prompt (KAP)	k	Input
(a)		6, 7, 0, 1, 1, 0, 1
(b)	1	0, 0, 0



(a)



(b)



(c)

Figure 10.6 A model for the dump truck for (a) kinematic analysis, (b) static analysis, and (c) dynamic analysis.

(b)	2	−.2, .08, −0.38
(b)	3	.6, .3, 0.15
(b)	4	.6, .4, −0.2
(b)	5	−.15, .6, 0.25
(b)	6	.9, .95, −0.15
(c)	1	1, 2, 0, 0, .2, 0
(c)	2	1, 4, 1.6, .2, 1, 0
(c)	3	2, 3, −.2, 0, −1, 0
(c)	4	3, 4, .3, 0, .3, 0
(c)	5	3, 6, 1, 0, .8, −.4
(c)	6	4, 5, −1, 0, −.2, 0
(c)	7	5, 6, .2, 0, −.8, −.4
(e)	1	1
(g)	1	2, 3, −.38
(h)	1	3, .3, 0
(i)		0, 0, 0.1

In this model, there are six bodies (one grounded), and seven revolute joints, and point D is given as a special point of interest on body 3 (or body 4). The output from this simulation is:

```
***** KINEMATIC ANALYSIS *****

TIME =      .0000
---------------
 BODY      X        Y       PHI       XD       YD      PHID      XDD      YDD     PHIDD
   1     .000     .000      .000     .000     .000      .000     .000     .000      .000
   2    -.186     .074     -.380     .000     .000      .000     .000     .000      .000
   3     .618     .296      .148     .000     .000      .000     .000     .000      .000
   4     .620     .401     -.202     .000     .000      .000     .000     .000      .000
   5    -.166     .652      .256     .000     .000      .000     .000     .000      .000
   6     .882     .968     -.163     .000     .000      .000     .000     .000      .000
 POINTS OF INTEREST
  NO.      X        Y       XD       YD      XDD      YDD
   1     .914     .341      .000     .000     .000     .000
```

According to Sec. 9.6, in order to determine the force required by the actuator for keeping the system in equilibrium, the actuator may be replaced by a revolute-revolute joint. The reaction forces exerted at A and D on this revolute-revolute joint give the required actuator force. However, since the formulation for the revolute-revolute joint is not included in the present version of DAP, the actuator can be replaced with a fictitious body and two revolute joints, one at A and one at D. This model is shown in Fig. 10.6(b), and the input to DAP is as follows:

Prompt (DAP)	k	Input
(a)		7, 9, 0, 1, 0, 0, 0
(b)	1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

(b)	2	−.1857, .0742, −0.380, 0, 0, 0, .4, .005, 0, 0, 0
(b)	3	.6175, .2962, 0.148, 0, 0, 0, 2, .7, 0, 0, 0
(b)	4	.6204, .4008, −0.202, 0, 0, 0, 2, .7, 0, 0, 0
(b)	5	−.1658, .6522, 0.256, 0, 0, 0, .4, .005, 0, 0, 0
(b)	6	.8818, .9682, −0.163, 0, 0, 0, 100, 27., 0, 0, 0
(b)	7	.4571, .1703, 0.357, 0, 0, 0, .001, .001, 0, 0, 0
(c)	1	1, 2, 0, 0, .2, 0
(c)	2	1, 4, 1.6, .2, 1, 0
(c)	3	2, 3, −.2, 0, −1, 0
(c)	4	3, 4, .3, 0, .3, 0
(c)	5	3, 6, 1, 0, .8, −.4
(c)	6	4, 5, −1, 0, −.2, 0
(c)	7	5, 6, .2, 0, −.8, −.4
(c)	8	1, 7, 0, 0, −.4878, 0
(c)	9	3, 7, .3, 0, .4878, 0
(e)	1	1
(i)		0, 0, 0.1

In this model, there are seven bodies and nine revolute joints. The coordinates of point D from the first simulation, with the known coordinates of A, are used to calculate the correct coordinates of body 7. Since this is a 0-degree of freedom system, DAP performs static analysis and the output is as follows:

```
      *****  STATIC ANALYSIS  *****

BODY      X        Y       PHI
 1      .000     .000     .000
 2    -.186     .074    -.380
 3      .618     .296     .148
 4      .620     .401    -.202
 5    -.166     .652     .256
 6      .882     .968    -.163
 7      .457     .170     .357
REACTION FORCES
 JOINT NO.  I      FX-I       FY-I       N-I   J      FX-J       FY-J       N-J
 REV.   1   1  -835.818    331.874      .000   2   835.818   -331.874      .365
 REV.   2   1  4505.031      8.582  -887.275   4 -4505.031     -8.582  -912.248
 REV.   3   2  -835.817    335.798     -.364   3   835.817   -335.798   455.377
 REV.   4   3 -5565.859   -307.829   154.884   4  5565.859    307.829   425.473
 REV.   5   3  1060.831   -705.297  -854.017   6 -1060.831    705.297   -45.452
 REV.   6   4 -1060.831   -279.627   486.775   5  1060.831    279.627     -.380
 REV.   7   5 -1060.831   -275.703      .380   6  1060.831    275.703    45.452
 REV.   8   1 -3669.213  -1368.553      .000   7  3669.213   1368.553     -.002
 REV.   9   3  3669.213   1368.544   243.756   7 -3669.213  -1368.544      .002
```

This output yields the reaction forces acting at A and D on the fictitious body 7. These force components indicate a compressive force on body 7, having a magnitude

$$f = (3.669.2^2 + 1368.5^2)^{1/2} = 3916$$

If body 7 is replaced by an actuator, the force of the actuator must be $f^{(a)} = -3916$ to keep the system in equilibrium. The negative sign is assigned to the actuator force, ac-

cording to the sign convention of Sec. 9.2.3. Note that the force that the actuator must apply on the system is in the direction opposite that of the force that the system applies on the actuator (or the fictitious body 7).

For dynamic analysis, another model can be set up by replacing the fictitious body with an actuator, as shown in Fig. 10.6(c). If the force of the actuator is specified as $f^{(a)} = -3916$, then the system remains in equilibrium. For simulating the unloading process, the actuator force is increased slightly to -3955 and the input to DAP is as follows:

Prompt (DAP)	k	Input
(a)		6, 7, 0, 1, 0, 1, 0
(b)	1	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
(b)	2	−.1857, .0742, −0.380, 0, 0, 0, .4, .005, 0, 0, 0
(b)	3	.6175, .2962, 0.148, 0, 0, 0, 2, .7, 0, 0, 0
(b)	4	.6204, .4008, −0.202, 0, 0, 0, 2, .7, 0, 0, 0
(b)	5	−.1658, .6522, 0.256, 0, 0, 0, , .4, .005, 0, 0, 0
(b)	6	.8818, .9682, −0.163, 0, 0, 0, 100, 27, 0, 0, 0
(c)	1	1, 2, 0, 0, .2, 0
(c)	2	1, 4, 1.6, .2, 1, 0
(c)	3	2, 3, −.2, 0, −1, 0
(c)	4	3, 4, .3, 0, .3, 0
(c)	5	3, 6, 1, 0, .8, −.4
(c)	6	4, 5, −1, 0, −.2, 0
(c)	7	5, 6, .2, 0, −.8, −.4
(e)	1	1
(g)	1	1, 4, 0, 0, .3, 0, 0, 500, −3955., 0
(i)		0, 1.5, 0.05

The result of this simulation is shown in Fig. 10.7 for several intermediate stages.

In the dynamic simulation of this system, or other systems, a problem may arise. In certain kinematic configurations, the Jacobian matrix $\mathbf{\Phi}_\mathbf{q}$ may lose rank—one or more of the constraint equations may become redundant. In this case, the matrix at the left in Eq. 10.5 becomes singular. Therefore, the present version of subroutine LINEAR cannot solve Eq. 10.5 for the accelerations and Lagrange multipliers. For the dump truck model, this situation occurs when the linkage system is completely stretched open or when it is folded. In these two cases, some of the bodies align in a way that causes kinematic redundancy in the constraint equations.

10.4 TIME STEP SELECTION

One of the most crucial problems in using a constant-step numerical integration algorithm, such as the Runge-Kutta algorithm, is the selection of a proper step size. A large step size may cause erroneous results. A step size too small may yield accurate results while increasing the computation time unreasonably. Therefore, it is important to choose a reasonably small step size to obtain accurate results without unnecessarily in-

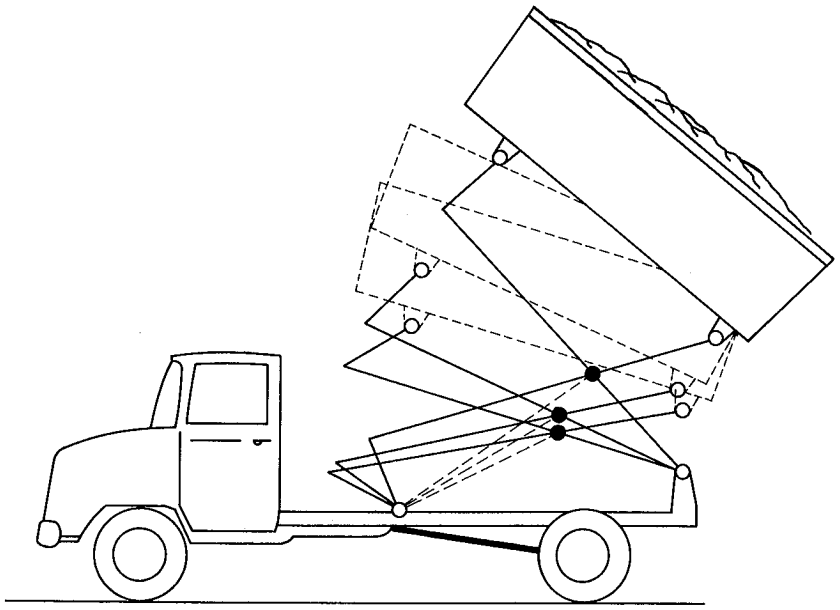


Figure 10.7 Unloading process of the dump truck.

creasing the computation time. A thorough discussion on the subject of time-step selection is outside the scope of this book. The interested reader may refer to other textbooks on the subject of numerical solution of differential equations. In this section only two highly simple examples are presented to familiarize the reader with this important point. Since these examples deal with vibratory motion, which has not been covered in this book, the reader may refer to any textbook on the subject of mechanical vibration for more detail.

The simplest form of vibratory motion is a *simple harmonic motion* which is described by the differential equation

$$\ddot{y} + p^2 y = 0 \tag{10.8}$$

where p is a real number. The frequency of oscillation of this single degree-of-freedom system is

$$f = \frac{p}{2\pi} \tag{10.9}$$

The period of the oscillation is

$$\tau = \frac{2\pi}{p} \tag{10.10}$$

If Eq. 10.8 is solved numerically, the step size Δt must be much smaller than the period τ.

As an example, consider the one-dimensional motion of the mass-spring system shown in Fig. 10.8(a). The only external force acting on the system is gravity. The equation of motion for this system, in the y direction, is given by

$$m\ddot{y} = -mg - k(y - l^0)$$

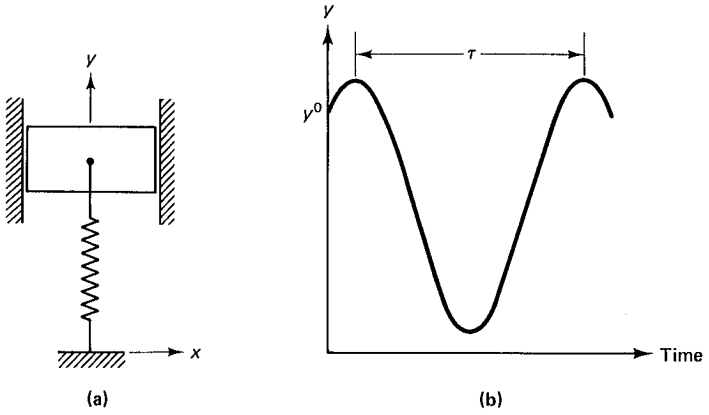


Figure 10.8 (a) A one-dimensional vibrating mass-spring system, and (b) a full cycle of the response denoted by τ.

or

$$\ddot{y} + \frac{k}{m}y = -g + \frac{k}{m}l^0 \qquad (a)$$

For a system in *free vibration,* if the left sides of Eq. 10.10 and Eq. *a* are compared, it is found that

$$p^2 = \frac{k}{m}$$

or

$$\tau = 2\pi\sqrt{\frac{m}{k}} \qquad (b)$$

Given the values of m and k, the time period τ can be calculated. For numerical integration, a reasonable value for Δt can be $\Delta t \simeq \tau/20$.

As another example, consider the single pendulum shown in Fig. 10.9. A single equation of motion for this system in terms of the angular coordinate can be written

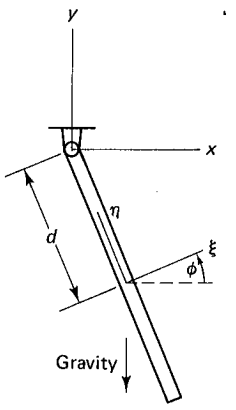


Figure 10.9 A single pendulum.

directly. However, here the equation is derived from the Cartesian equations of motion. Considering gravity to be the only external force, the equations of motion are

$$\begin{aligned} m\ddot{x} - \lambda_1 &= 0 \\ m\ddot{y} - \lambda_2 &= -mg \\ \mu\ddot{\phi} + d\cos\phi\lambda_1 + d\sin\phi\lambda_2 &= 0 \end{aligned} \tag{c}$$

where λ_1 and λ_2 are two Lagrange multipliers associated with the constraints for the revolute joint,

$$\begin{aligned} x - d\sin\phi &= 0 \\ y + d\cos\phi &= 0 \end{aligned}$$

The kinematic acceleration equations are,

$$\begin{aligned} \ddot{x} - d\cos\phi\,\ddot{\phi} + d\sin\phi\,\dot{\phi}^2 &= 0 \\ \ddot{y} - d\sin\phi\,\ddot{\phi} - d\cos\phi\,\dot{\phi}^2 &= 0 \end{aligned} \tag{d}$$

Elimination of λ_1, λ_2, $\ddot{x}$, and $\ddot{y}$ in Eqs. c and d results in one equation:

$$(\mu + md^2)\ddot{\phi} + mdg\sin\phi = 0$$

For oscillations of small amplitude, $\sin\phi$ can be replaced by ϕ; this last equation is then linearized as follows:

$$(\mu + md^2)\ddot{\phi} + mdg\phi = 0 \tag{e}$$

Comparing Eq. 10.8 and Eq. e yields

$$p^2 = \frac{mdg}{\mu + md^2}$$

or

$$\tau = 2\pi\sqrt{\frac{\mu + md^2}{mdg}} \tag{f}$$

The preceding examples show how the time period of the oscillation can be calculated. For more complicated systems, the calculation of natural frequencies will not be that simple. For systems with several interconnected moving bodies, the linearization of the equations of motion in explicit form can be rather cumbersome. For systems with more than 1 degree of freedom, there will be more than one natural frequency. For such systems, the highest frequency must be found, and a step size much smaller than the period of the highest frequency must be selected. Since the equations of motion are generally nonlinear in terms of the coordinates, linearization of these equations yields a time step which is valid only in the individual configuration. In a different configuration for the same system, the linearization process may yield a different time period, and consequently a different time step size.

To avoid the difficulties associated with the selection of a proper time step, it is strongly suggested that a variable–step size algorithm be used for dynamic analysis. Many well-developed algorithms of this sort are available. Most of these algorithms determine a proper time step and will adjust the time step automatically during the simulation.

PROBLEMS

The following problems provide examples that can be simulated by using a dynamic analysis program such as DAP. Many of the problems can be simulated on the existing listed version of DAP. Other problems may require some modifications or extensions to the program. Guidelines for improving the versatility and increasing the capability of DAP are included for some of those problems.

10.1 Refer to Probs. 5.8 through 5.11 and include similar modifications in DAP for input-output versatility.

10.2 Modify DAP to accept data in different but consistent units.

10.3 Formulate additional force elements such as the rotational spring-damper-actuator element in DAP.

10.4 Refer to Probs. 5.13 and 5.16 through 5.19. Include similar changes in DAP.

10.5 Refer to Prob. 9.5 and include this capability in DAP.

10.6 An external force acting on a body can be time-dependent. Modify DAP to accept time-dependent external forces in the following forms:

(a) A closed-form expression

(b) A table of data points

10.7 Provide the user with the option of including or excluding gravitational forces in a simulation.

10.8 Refer to Prob. 5.18. A dummy subroutine USRFRC, similar to the user-supplied subroutine USRCON, can be called from subroutine UFORCE before the RETURN statement. This subroutine may be used for nonstandard forces defined by the user.

10.9 The present version of DAP applies L-U factorization to the coefficient matrix in Eq. 10.5 to solve for $\ddot{\mathbf{q}}$ and λ. This process can be made more efficient by the following modification:

(a) From Eq. 10.5 it is found that,

$$\mathbf{B}\boldsymbol{\lambda} = \boldsymbol{\gamma} - \mathbf{C}\mathbf{g} \tag{a.1}$$

$$\ddot{\mathbf{q}} = \mathbf{C}^T\boldsymbol{\lambda} + \mathbf{M}^{-1}\mathbf{g} \tag{a.2}$$

where $\mathbf{C} = \boldsymbol{\Phi}_\mathbf{q}\mathbf{M}^{-1}$ and $\mathbf{B} = \mathbf{C}\boldsymbol{\Phi}_\mathbf{q}^T$. Matrix $\mathbf{B}$ is an $m \times m$ symmetric matrix.

(b) Perform L-U factorization on $\mathbf{B}$, and then solve Eq. a.1 for $\boldsymbol{\lambda}$.

(c) Substitute the result for $\boldsymbol{\lambda}$ in Eq. a.2 to find $\ddot{\mathbf{q}}$.

Note that this process requires $\mathbf{M}^{-1}$ to be evaluated only once, since $\mathbf{M}$ is a constant diagonal matrix. If the mass or the moment of inertia of a body is specified as zero, then the inversion cannot be performed. This is usually the case for nonmoving (grounded) bodies. For these bodies, the mass and moment of inertia can be set to any nonzero value; e.g., $m = 1$ and $\mu = 1$.

10.10 Matrix $\mathbf{B}$ in Prob. 10.9, for a well-posed problem, is symmetric and positive-definite. For these types of matrices, special-purpose matrix factorization algorithms are available. These algorithms are more efficient than the standard L-U factorization algorithms. Replace subroutine LU with a subroutine utilizing such an algorithm.

10.11 Evaluation of matrices $\mathbf{C}$ and $\mathbf{B}$ in Prob. 10.9 requires matrix multiplication involving $\boldsymbol{\Phi}_\mathbf{q}$. Since $\boldsymbol{\Phi}_\mathbf{q}$ is a sparse matrix, a large number of multiplications by zero are performed in the process of evaluating $\mathbf{C}$ and $\mathbf{B}$. In order to eliminate an unnecessary multiplication by zero, store the row and column numbers of the nonzero elements of $\boldsymbol{\Phi}_\mathbf{q}$ in two index arrays IRN

and ICN. The elements of these arrays, plus the fact that $\mathbf{M}^{-1}$ is diagonal, can minimize the number of multiplications in the evaluation of **C** and **B**.

10.12 For very small integration time steps and long simulation time periods, the output of DAP can be extensive. Modify DAP to report the result at every nth time step, where n is an integer to be specified by the user.

10.13 If the initial conditions on coordinates and velocities violate the constraints and their time derivatives respectively, they must be corrected before the start of dynamic analysis. Perform a kinematic position and velocity analysis at $t = 0$ to correct the initial conditions. For this process, additional constraints equal to the number of degrees of freedom must be introduced. The corrected initial conditions are then used to start the dynamic analysis.

10.14 Replace the Runge-Kutta algorithm in DAP with a predictor-corrector algorithm in order to make the integration process more efficient (refer to Chap. 12). You may find such an algorithm in the library of your computer.

10.15 Use a variable step/order predictor-corrector algorithm in DAP instead of subrouting RUNG4. This can be the *most important modification* to DAP for minimizing the numerical error in the computation (refer to Prob. 12.7).

10.16 Refer to the constraint violation stabilization method in Sec. 13.3.1. Modify vector $\boldsymbol{\gamma}$, according to Eq. 13.18, to include the feedback terms $-2\alpha\dot{\boldsymbol{\Phi}}$ and $-\beta^2\boldsymbol{\Phi}$. Before evaluating the forces in subroutine DYNAM, perform the following tasks:

(a) Call FUNCT, with IFNCT = 1 and JACOB = 0, to obtain the constraint violations in F. Add $-\beta^2$F(I) to RHS(I) for I = 1, . . . , M.

(b) Since the Jacobian matrix is already available, add $-2\alpha\boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}$ to RHS(I) for I = 1, . . . , M. The user must specify the parameters α and β. Simulate different problems and experiment with different values of these parameters.

10.17 Modify DAP in order to handle both unconstrained and constrained systems. If M = 0, there are no constraints. Therefore, the program should solve the equations of motion stated in Eq. 10.1.

10.18 In the dynamic analysis of mechanical systems, it might be necessary to include aerodynamic forces in vector **g**. A simple formula for calculating aerodynamic forces is given as

$$f^{(\text{aero})} = \tfrac{1}{2} c_d \rho A v^2$$

where

c_d = drag coefficient; e.g., $c_d = 0.5$
ρ = air density
v = velocity of the body
A = cross-sectional area of the body perpendicular to the direction of velocity

Included this capability in DAP as another force element.

10.19 Simulate the dynamics of the chain shown in Fig. P.10.19. The links are connected by revolute joints and the external force is gravity. Start the dynamic analysis from two different initial positions:

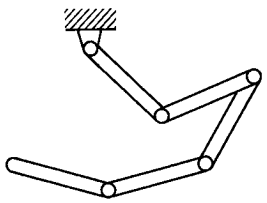


Figure P. 10.19

(a) The links are open and the chain is stretched.

(b) The links are folded on top of one another.

10.20 The front landing gear of an aircraft is designed in such a way that it avoids hitting a protruding pod (this might be an extra fuel tank) while retracting. The opening or retracting of the landing gear is controlled by a hydraulic actuator between points P and Q as shown in Fig. P.10.20. Assume $AB = 1.2$, $CD = 1.62$, $BC = 0.5$, $BO = 1.32$, $AE = 0.97$, $DE = 0.14$, and the wheel radius $\rho = 0.2$.

(a) Assign values to the mass and moment of inertia of each moving body.

(b) Simulate the retracting process by applying a constant force (or a variable force, if DAP is modified to accommodate one) to the actuator.

(c) Determine the path of the wheel with respect to the aircraft.

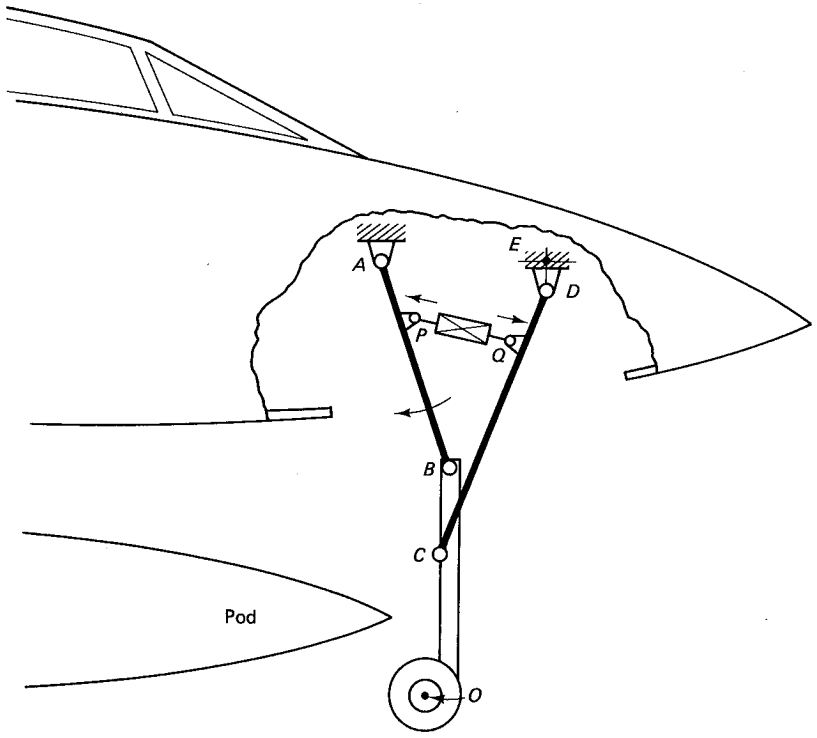


Figure P. 10.20

10.21 Simulate the bouncing of a rubber ball against the ground. Assume that the ground surface is flat at $y = 0$, as shown in Fig. P.10.21. Write a nonstandard force subroutine (refer to Prob. 10.8) to determine the reaction force between the ball and the ground during contact. Monitor the deformation $\Delta l = \rho - y_2$. If $\Delta l < 0$, then there is no contact; hence, there is no reaction force. If $\Delta l > 0$, calculate the reaction force as $f = f^{(s)} + f^{(d)}$ where

$$f^{(s)} = k\,\Delta l$$

$$f^{(d)} = \begin{cases} -d\dot{y}_2 & \text{for } \dot{y}_2 < 0 \\ 0 & \text{for } \dot{y}_2 > 0 \end{cases}$$

It is assumed that k and d are the stiffness and damping coefficient of the ball.

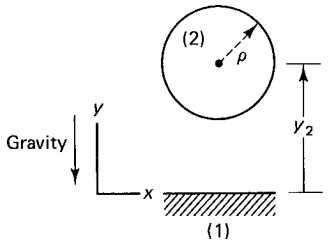


Figure P. 10.21

10.22 Repeat Prob. 10.21 and assume that the ground surface makes an angle with the horizontal direction.

10.23 Two rubber balls are constrained to move in the vertical direction inside a frictionless cylinder as shown in Fig. P.10.23. Write a subroutine to calculate the reaction forces between the balls and the ground. Simulate the motion of the balls using DAP.

10.24 The apparatus shown in Fig. P.10.24 consists of five pendulums terminating in rubber balls which can be modeled as five moving bodies and five revolute joints. The interaction between the balls can be modeled by unilateral spring elements, as can that in Prob. 10.23. Move one ball (or two) from the equilibrium state and then release it (or them). Simulate the motion of the balls using DAP.

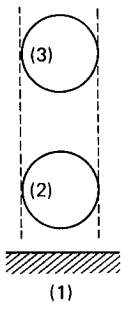


Figure P. 10.23

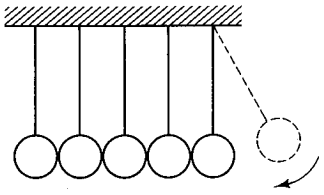


Figure P. 10.24

10.25 The motion of the articulated bulldozer shown in Fig. P.10.25 is controlled by two hydraulic actuators. The centers of mass of the moving bodies are shown as $G_1, \ldots, G_4$. Take measurements from the figure and assume $m_1 = 190$, $m_2 = 52$, $m_3 = 12$, $m_4 = 3$, $\mu_1 = 45$, $\mu_2 = 0.9$, $\mu_3 = 0.3$, and $\mu_4 = 0.1$.

(a) Find a correct set of initial conditions for the coordinates.

(b) Determine the actuator forces in the equilibrium state.

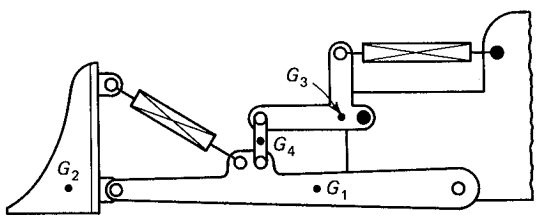


Figure P. 10.25

10.26 A hydraulic excavator is shown in Fig. P.10.26. Set up a model for this vehicle by taking direct measurements from the figure and assuming reasonable values for the mass and moment of inertia of each body.

(a) For different orientations, find the correct set of initial conditions.

(b) For each set of initial conditions, find the necessary force for each actuator to keep the system in equilibrium.

(c) Perform dynamic analysis by controlling the force of each actuator.

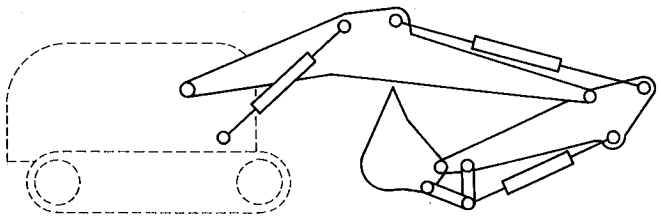


Figure P. 10.26

10.27 Modify the revolute-revolute joint constraint formulation of Eq. 4.13 by assuming that the length of the link varies as a function of time; i.e.,

$$\boldsymbol{l}^T\boldsymbol{l} - [d(t)]^2 = 0$$

For this constraint, $\dot{d}$ and $\ddot{d}$ must be included in the velocity and acceleration equations.

(a) Include this formulation in DAP.

(b) Repeat Prob. 10.26 and employ this variable-length revolute-revolute joint constraint to represent the actuators. Specify a time function for each $d(t)$ and monitor the reaction forces associated with each actuator.

This process is an inverse approach for determining the required force of each actuator as a function of time in order to generate a particular motion for the system.

10.28 A fork-lift mechanism is shown in Fig. P. 10.28. Rotation of the crank provides an upward or a downward motion of the fork. Set up a model for this vehicle by taking direct measurements from the figure and assigning values to the mass and moment of inertia of each body.

(a) For different orientations, find the correct set of initial conditions.

(b) For each set of initial conditions, find the torque on the crank that keeps the system in equilibrium.

(c) Perform a dynamic analysis by changing the applied torque. Monitor the path of the fork for a complete revolution of the crank.

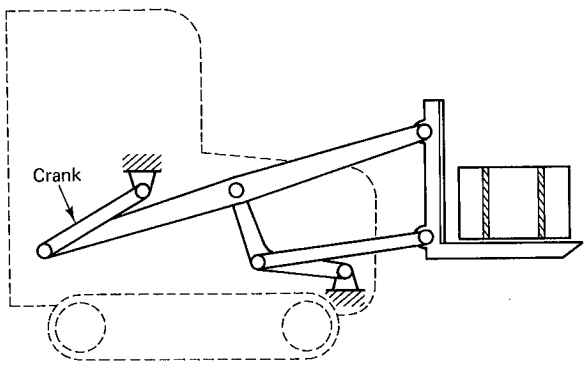


Figure P. 10.28

10.29 The mechanism shown in Fig. P. 10.29 is a centrifugal brake system. The braking mechanism is designed so that when the drive shaft exceeds a certain angular velocity, the three pistons are forced against the hub wall. The contacting surfaces of the pistons are covered by brake pads, which provide the friction force opposing the motion, hence reducing the angular velocity. When the shaft is not rotating, the distance between the shaft axis and the contact surface of each pad is equal to the inner radius of the hub. The contact interface beween the pads and the hub wall can be modeled by unilateral spring elements. The spring force $f^{(r)} = k_2\,\Delta l$ constitutes the reaction force, where k_2 is the pad stiffness and Δl is the pad deformation. A friction force $f^{(f)} = \mu_d f^{(r)}$ is applied to the pad at the contacting surface, where μ_d is the kinetic coefficient of friction. Assume $m_2 = 10$, $m_3 = 3$, $\mu_2 = 10$, $\mu_3 = 3$, $k_1 = 150$, $d_1 = 10$, $k_2 = 10{,}000$, and $\mu_d = 0.5$.

(a) Write a user-supplied subroutine for the unilateral spring, reaction force, and friction force (repeat for each pad).

(b) For a specified initial rotational velocity of body 2, determine the initial velocity vector for the system that will satisfy the constraints.

(c) Apply a constant input torque to the shaft and simulate the response.

(d) Determine the equilibrium rotational velocity of the shaft.

Note: Since the three braking elements are identical, you may model only one element and multiply the resisting force (moment) by 3.

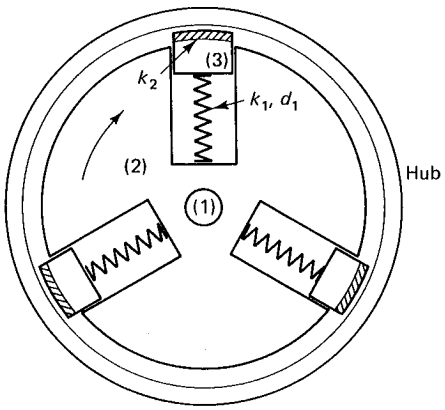


Figure P. 10.29

10.30 The pressure-type altimeter shown in Fig. P. 10.30 utilizes the difference between the pressure at sea level and the ambient pressure to displace a pointer which indicates the altitude. This is accomplished by sealing air at sea-level pressure inside a bellows. When the outside pressure is different from the air pressure inside the bellows, the bellows expands or contracts until the pressure in the bellows is equal to the outside pressure.

(a) Write a user-supplied subroutine to model the bellows and convert the pressure difference to a force acting on the attached piston.

(b) Include some damping (friction) force acting on the piston.

(c) Simulate the dynamics of the system and show whether or not the displacement of the pointer is a linear function of the pressure change.

10.31 A simple model of a vehicle can be set up by assuming that it consists of three bodies and two revolute joints as shown in Fig. P.10.31. Body 1 is the chassis, and bodies 2 and 3 are the wheels. Let $m_1 = 600$, $m_2 = m_3 = 20$, $\mu_1 = 1200$, and $\mu_2 = \mu_3 = 5$. The axial de-

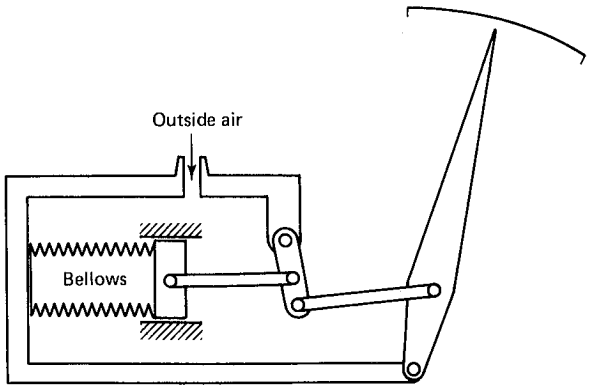


Figure P. 10.30

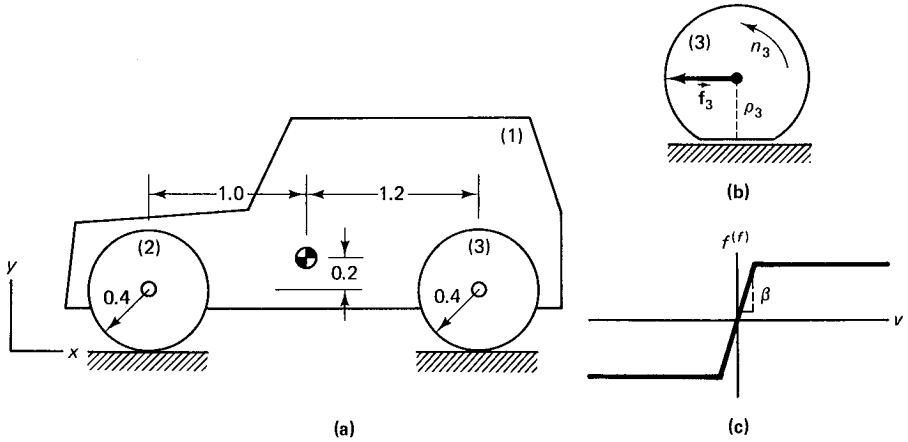


Figure P. 10.31

formation of each tire can be modeled by unilateral spring-damper elements, where $k = 175{,}000$ and $d = 7000$.

(a) Use DAP to determine the static equilibrium state of the vehicle.

(b) Perform a dynamic analysis, starting from the static equilibrium state, by assuming that the wheels slide (i.e., do not rotate). Assign an initial velocity to the vehicle. Refer to Prob. 9.10 and write a user-supplied subroutine for different terrains and obstacles.

(c) For a more realistic dynamic simulation, the rotation of the wheels must be included in the analysis. Assume that the rear wheel (body 3) is the driving wheel. A moment n_3 is applied to this wheel by the engine. If there is no slipping between the wheel and the ground, then a force $f_3 = -n_3/\rho_3$, where ρ_3 is the deformed radius of the wheel, is applied to the wheel in the direction shown in Fig. P. 10.31(b).

(d) If the results of part (c) are studied, it is observed that the rotational and the translational velocity of body 3 do not satisfy the no-slip assumption; i.e., $x_3 \neq -\rho_3\dot{\phi}_3$. Therefore, a friction model must be included in the simulation. If $v = \dot{x}_3 + \rho_3\dot{\phi}_3$ is nonzero, then the wheel is slipping. A friction force can be calculated according to the friction curve shown in Fig. P.10.31(c). In this curve β is a coefficient based on the tire characteristics (for simulation, assume values of β between 1000 and 5000), and

the upper bound for the friction force is $f^{(l)} = \mu_d f^{(r)}$, where μ_d is the kinetic coefficient of friction and $f^{(r)}$ is the normal reaction force. $f^{(r)}$ is available from the unilateral spring-damper element representing the tire deformation.

(e) Assume that the front wheel (body 2) is the driven wheel. Include a no-slip friction model for this wheel and perform a dynamic simulation.

10.32 Improve the vehicle model of Prob. 10.31 by adding a suspension system to the front and rear wheels as shown in Fig. P.10.32. Assume that the front wheel is attached to an extra body (body 4) by a revolute joint, and that body 4 is attached to the chassis by a translational joint. Also, assume that the rear wheel is attached to an extra body (body 5) by a revolute joint, and that body 5 is attached to the chassis by another revolute joint. Let $m_4 = m_5 = 4$, $\mu_4 = 1$, and $\mu_5 = 2$. The spring-damper characteristics for the front suspension system are $k_1 = 90{,}000$ and $d_1 = 5000$, and for the rear suspension system $k_2 = 60{,}000$ and $d_2 = 5000$. The tire characteristics are the same as in Prob. 10.31.

(a) Find the static equilibrium state for the vehicle. Adjust the attachment points or the undeformed lengths of the suspension springs in such a way that the main chassis and the axis of body 5 remain horizontal when the vehicle is in static equilibrium.

(b) Include friction models for a no-slip condition of the wheels.

(c) Perform a variety of simulations for different terrains.

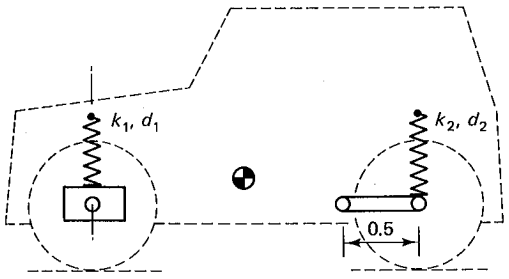


Figure P. 10.32

10.33 Include an additional capability in DAP to perform a static equilibrium analysis prior to dynamic analysis when necessary. Refer to the algorithms in Chap. 14 (the iterative method of Eq. 14.3 requires a minimum amount of programming).