APPENDIX

NUMERICAL INTEGRATION OF THE *N*-BODY EQUATIONS OF MOTION

C

Without loss of generality we shall derive the equations of motion of the three-body system illustrated in Fig. C.1. The equations of motion for n bodies can easily be generalized from those of a three-body system.

Each mass of a three-body system experiences the force of gravitational attraction from the other members of the system. As shown in Fig. C.1, the forces exerted on body 1 by bodies 2 and 3 are $\mathbf{F}_{12}$ and $\mathbf{F}_{13}$, respectively. Likewise, body 2 experiences the forces $\mathbf{F}_{21}$ and $\mathbf{F}_{23}$ whereas the forces $\mathbf{F}_{31}$ and $\mathbf{F}_{32}$ act on body 3. These gravitational forces can be inferred from Eq. (2.9):

$$\mathbf{F}_{12} = -\mathbf{F}_{21} = \frac{Gm_1m_2(\mathbf{R}_2 - \mathbf{R}_1)}{\|\mathbf{R}_2 - \mathbf{R}_1\|^3} \tag{C.1a}$$

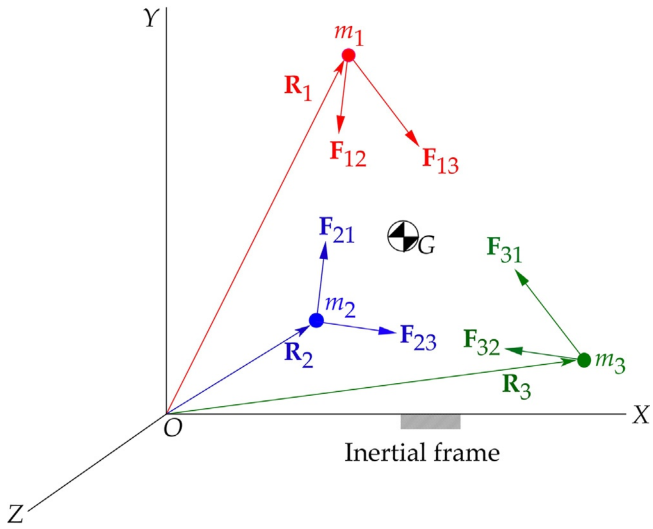

FIG. C.1

Three-body problem.

$$\mathbf{F}_{13} = -\mathbf{F}_{31} = \frac{Gm_1m_3(\mathbf{R}_3 - \mathbf{R}_1)}{\|\mathbf{R}_3 - \mathbf{R}_1\|^3} \tag{C.1b}$$

$$\mathbf{F}_{23} = -\mathbf{F}_{32} = \frac{Gm_2m_3(\mathbf{R}_3 - \mathbf{R}_2)}{\|\mathbf{R}_3 - \mathbf{R}_2\|^3} \tag{C.1c}$$

Relative to an inertial frame of reference the accelerations of the bodies are

$$\mathbf{a}_i = \ddot{\mathbf{R}}_i \quad i = 1,2,3$$

where $\mathbf{R}_i$ is the absolute position vector of body i. The equation of motion of body 1 is

$$\mathbf{F}_{12} + \mathbf{F}_{13} = m_1\mathbf{a}_1$$

Substituting Eqs. (C.1a) and (C.1b) yields

$$\mathbf{a}_1 = \frac{Gm_2(\mathbf{R}_2 - \mathbf{R}_1)}{\|\mathbf{R}_2 - \mathbf{R}_1\|^3} + \frac{Gm_3(\mathbf{R}_3 - \mathbf{R}_1)}{\|\mathbf{R}_3 - \mathbf{R}_1\|^3} \tag{C.2a}$$

For bodies 2 and 3 we find in a similar fashion that

$$\mathbf{a}_2 = \frac{Gm_1(\mathbf{R}_1 - \mathbf{R}_2)}{\|\mathbf{R}_1 - \mathbf{R}_2\|^3} + \frac{Gm_3(\mathbf{R}_3 - \mathbf{R}_2)}{\|\mathbf{R}_3 - \mathbf{R}_2\|^3} \tag{C.2b}$$

$$\mathbf{a}_3 = \frac{Gm_1(\mathbf{R}_1 - \mathbf{R}_3)}{\|\mathbf{R}_1 - \mathbf{R}_3\|^3} + \frac{Gm_2(\mathbf{R}_2 - \mathbf{R}_3)}{\|\mathbf{R}_2 - \mathbf{R}_3\|^3} \tag{C.2c}$$

The velocities are related to the accelerations by

$$\frac{d\mathbf{v}_i}{dt} = \mathbf{a}_i \quad i = 1,2,3 \tag{C.3}$$

and the position vectors are likewise related to the velocities,

$$\frac{d\mathbf{R}_i}{dt} = \mathbf{v}_i \quad i = 1,2,3 \tag{C.4}$$

Eqs. (C.2)–(C.4) constitute a system of ordinary differential equations (ODEs) in variable time.

Given the initial positions $\mathbf{R}_{i_0}$ and initial velocities $\mathbf{v}_{i_0}$, we must integrate Eq. (C.3) to find $\mathbf{v}_i$ as a function of time and substitute those results into Eq. (C.4) to obtain $\mathbf{R}_i$ as a function of time. The integrations must be done numerically.

To do this using MATLAB, we first resolve all the vectors into their three components along the *XYZ* axes of the inertial frame and write them as column vectors,

$$\mathbf{R}_1 = \begin{Bmatrix} X_1 \\ Y_1 \\ Z_1 \end{Bmatrix} \quad \mathbf{R}_2 = \begin{Bmatrix} X_2 \\ Y_2 \\ Z_2 \end{Bmatrix} \quad \mathbf{R}_3 = \begin{Bmatrix} X_3 \\ Y_3 \\ Z_3 \end{Bmatrix} \tag{C.5}$$

$$\mathbf{v}_1 = \begin{Bmatrix} \dot{X}_1 \\ \dot{Y}_1 \\ \dot{Z}_1 \end{Bmatrix} \quad \mathbf{v}_2 = \begin{Bmatrix} \dot{X}_2 \\ \dot{Y}_2 \\ \dot{Z}_2 \end{Bmatrix} \quad \mathbf{v}_3 = \begin{Bmatrix} \dot{X}_3 \\ \dot{Y}_3 \\ \dot{Z}_3 \end{Bmatrix} \tag{C.6}$$

According to Eqs. (C.2),

$$\mathbf{a}_1 = \begin{Bmatrix} \ddot{X}_1 \\ \ddot{Y}_1 \\ \ddot{Z}_1 \end{Bmatrix} = \begin{Bmatrix} \dfrac{Gm_2(X_2 - X_1)}{R_{12}{}^3} + \dfrac{Gm_3(X_3 - X_1)}{R_{13}{}^3} \\ \dfrac{Gm_2(Y_2 - Y_1)}{R_{12}{}^3} + \dfrac{Gm_3(Y_3 - Y_1)}{R_{13}{}^3} \\ \dfrac{Gm_2(Z_2 - Z_1)}{R_{12}{}^3} + \dfrac{Gm_3(Z_3 - Z_1)}{R_{13}{}^3} \end{Bmatrix} \tag{C.7a}$$

$$\mathbf{a}_2 = \begin{Bmatrix} \ddot{X}_2 \\ \ddot{Y}_2 \\ \ddot{Z}_2 \end{Bmatrix} = \begin{Bmatrix} \dfrac{Gm_1(X_1 - X_2)}{R_{12}{}^3} + \dfrac{Gm_3(X_3 - X_2)}{R_{13}{}^3} \\ \dfrac{Gm_1(Y_1 - Y_2)}{R_{12}{}^3} + \dfrac{Gm_3(Y_3 - Y_2)}{R_{13}{}^3} \\ \dfrac{Gm_1(Z_1 - Z_2)}{R_{12}{}^3} + \dfrac{Gm_3(Z_3 - Z_2)}{R_{13}{}^3} \end{Bmatrix} \tag{C.7b}$$

$$\mathbf{a}_3 = \begin{Bmatrix} \ddot{X}_3 \\ \ddot{Y}_3 \\ \ddot{Z}_3 \end{Bmatrix} = \begin{Bmatrix} \dfrac{Gm_1(X_1 - X_3)}{R_{12}{}^3} + \dfrac{Gm_2(X_2 - X_3)}{R_{13}{}^3} \\ \dfrac{Gm_1(Y_1 - Y_3)}{R_{12}{}^3} + \dfrac{Gm_2(Y_2 - Y_3)}{R_{13}{}^3} \\ \dfrac{Gm_1(Z_1 - Z_3)}{R_{12}{}^3} + \dfrac{Gm_2(Z_2 - Z_3)}{R_{13}{}^3} \end{Bmatrix} \tag{C.7c}$$

where

$$R_{12} = \|\mathbf{R}_2 - \mathbf{R}_1\| \quad R_{13} = \|\mathbf{R}_3 - \mathbf{R}_1\| \quad R_{23} = \|\mathbf{R}_3 - \mathbf{R}_2\| \tag{C.8}$$

Next, we form the 18-component column vector

$$\mathbf{y} = \lfloor \mathbf{R}_1 \;\; \mathbf{R}_2 \;\; \mathbf{R}_3 \;\; \mathbf{v}_1 \;\; \mathbf{v}_2 \;\; \mathbf{v}_3 \rfloor^T \tag{C.9}$$

The first derivatives of the components of this vector comprise the column vector

$$\dot{\mathbf{y}} = \mathbf{f} = \lfloor \mathbf{v}_1 \;\; \mathbf{v}_2 \;\; \mathbf{v}_3 \;\; \mathbf{a}_1 \;\; \mathbf{a}_2 \;\; \mathbf{a}_3 \rfloor^T \tag{C.10}$$

According to Eqs. (C.8), the accelerations are functions of $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$. Hence, Eq. (C.10) is of the form

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}) \tag{C.11}$$

given in Eq. (1.95), although in this case time t does not appear explicitly. Eq. (C.11) can be employed in procedures, such as Algorithms 1.1, 1.2, or 1.3, to obtain a numerical solution for $\mathbf{R}_1(t)$, $\mathbf{R}_2(t)$, and $\mathbf{R}_3(t)$. We shall choose MATLAB's *ode45* Runge-Kutta solver.

For simplicity, we will solve the three-body problem in the plane. That is, we will restrict ourselves to only the XY components of the vectors $\mathbf{R}$, $\mathbf{v}$, and $\mathbf{a}$. The reader can use these scripts as a starting point for investigating more complex n-body problems.

The MATLAB function *threebody.m* contains the subfunction *rates*, which computes the accelerations given above in Eqs. (C.7). That information together with the initial conditions are passed to *ode45*, which integrates the system given by Eq. (C.11) and finally plots the solutions. The results of this program were used to create Figs. 2.4 and 2.5.

Function file `threebody.m`

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function threebody
% ~~~~~~~~~~~~~~~~~~~~
%{
  This program presents the graphical solution of the motion of three
  bodies in the plane for data provided in the input definitions below.

  MATLAB's ode45 Runge-Kutta solver is used.

  G                         - gravitational constant (km^3/kg/s^2)
  t0, tf                    - initial and final times (s)
  m1, m2, m3                - masses of the three bodies (kg)
  m                         - total mass (kg)
  X1,Y1; X2,Y2; X3,Y3       - coordinates of the three masses (km)
  VX1,VY1; VX2,VY2; VX3,VY3 - velocity components of the three
                              masses (km/s)
  XG, YG                    - coordinates of the center of mass (km)
  y0                        - column vector of the initial conditions
  t                         - column vector of times at which the solution
                              was computed
  y                         - matrix, the columns of which contain the
                              position and velocity components evaluated at
                              the times t(:):
                                y(:,1) , y(:, 2) =  X1(:), Y1(:)
                                y(:,3) , y(:, 4) =  X2(:), Y2(:)
                                y(:,5) , y(:, 6) =  X3(:), Y3(:)

                                y(:,7) , y(:, 8) = VX1(:), VY1(:)
                                y(:,9) , y(:,10) = VX2(:), VY2(:)
                                y(:,11), y(:,12) = VX3(:), VY3(:)

  User M-functions required:  none
  User subfunctions required: rates, plotit
%}
% ----------------------------------------------------------------------------

clear all
close all
clc

G = 6.67259e-20;

%...Input data:
m1 = 1.e29; m2 = 1.e29; m3 = 1.e29;
```

```
t0 = 0; tf = 67000;

X1 = 0;      Y1 = 0;
X2 = 300000; Y2 = 0;
X3 = 2*X2;   Y3 = 0;

VX1 = 0;    VY1 = 0;
VX2 = 250;  VY2 = 250;
VX3 = 0;    VY3 = 0;
%...End input data

m  = m1 + m2 + m3;
y0 = [X1 Y1  X2 Y2  X3 Y3  VX1 VY1  VX2 VY2  VX3 VY3]';

%...Pass the initial conditions and time interval to ode45, which
%   calculates the position and velocity of each particle at discrete
%   times t, returning the solution in the column vector y. ode45 uses
%   the subfunction 'rates' below to evaluate the accelerations at each
%   integration time step.
[t,y] = ode45(@rates, [t0 tf], y0);

X1 = y(:,1); Y1 = y(:,2);
X2 = y(:,3); Y2 = y(:,4);
X3 = y(:,5); Y3 = y(:,6);

%...Locate the center of mass at each time step:
XG = []; YG = [];
for i = 1:length(t)
    XG = [XG; (m1*X1(i) + m2*X2(i) + m3*X3(i))/m];
    YG = [YG; (m1*Y1(i) + m2*Y2(i) + m3*Y3(i))/m];
end

%...Coordinates of each particle relative to the center of mass:
X1G = X1 - XG; Y1G = Y1 - YG;
X2G = X2 - XG; Y2G = Y2 - YG;
X3G = X3 - XG; Y3G = Y3 - YG;

plotit

return

% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  function dydt = rates(t,y)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
  This function evaluates the acceleration of each member of a planar
  3-body system at time t from their positions and velocities
  at that time.
```

```
  t                          - time (s)
  y                          - column vector containing the position and
                               velocity components of the three masses
                               at time t
  R12                        - cube of the distance between m1 and m2 (km^3)
  R13                        - cube of the distance between m1 and m3 (km^3)
  R23                        - cube of the distance between m2 and m3 (km^3)
  AX1,AY1; AX2,AY2; AX3,AY3 - acceleration components of each mass (km/s^2)
  dydt                       - column vector containing the velocity and
                               acceleration components of the three
                               masses at time t

%}
% ------------------------------------------------------------------------------

X1  = y( 1);
Y1  = y( 2);

X2  = y( 3);
Y2  = y( 4);

X3  = y( 5);
Y3  = y( 6);

VX1 = y( 7);
VY1 = y( 8);

VX2 = y( 9);
VY2 = y(10);

VX3 = y(11);
VY3 = y(12);

%...Equations C.8:
R12 = norm([X2 - X1, Y2 - Y1])^3;
R13 = norm([X3 - X1, Y3 - Y1])^3;
R23 = norm([X3 - X2, Y3 - Y2])^3;

%...Equations C.7:
AX1 = G*m2*(X2 - X1)/R12 + G*m3*(X3 - X1)/R13;
AY1 = G*m2*(Y2 - Y1)/R12 + G*m3*(Y3 - Y1)/R13;
AX2 = G*m1*(X1 - X2)/R12 + G*m3*(X3 - X2)/R23;
AY2 = G*m1*(Y1 - Y2)/R12 + G*m3*(Y3 - Y2)/R23;
AX3 = G*m1*(X1 - X3)/R13 + G*m2*(X2 - X3)/R23;
AY3 = G*m1*(Y1 - Y3)/R13 + G*m2*(Y2 - Y3)/R23;

dydt = [VX1 VY1 VX2 VY2 VX3 VY3 AX1 AY1 AX2 AY2 AX3 AY3]';
```

```
end %rates
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% ~~~~~~~~~~~~~~~~
function plotit
% -------------

%...Plot the motions relative to the inertial frame (Figure 2.4):
figure(1)
title('Figure 2.4: Motion relative to the inertial frame', ...
      'Fontweight', 'bold', 'FontSize', 12)
hold on
plot(XG, YG, '--k', 'LineWidth', 0.25)
plot(X1, Y1,   'r', 'LineWidth', 0.5)
plot(X2, Y2,   'g', 'LineWidth', 0.75)
plot(X3, Y3,   'b', 'LineWidth', 1.00)
xlabel('X(km)'); ylabel('Y(km)')
grid on
axis('equal')

%...Plot the motions relative to the center of mass (Figure 2.5):
figure(2)
title('Figure 2.5: Motion relative to the center of mass', ...
      'Fontweight', 'bold', 'FontSize', 12)
hold on
plot(X1G, Y1G,   'r', 'LineWidth', 0.5)
plot(X2G, Y2G, '--g', 'LineWidth', 0.75)
plot(X3G, Y3G,   'b', 'LineWidth', 1.00)
xlabel('X(km)'); ylabel('Y(km)')
grid on
axis('equal')
end %plotit
% ~~~~~~~~~~~~~~~~

end %threebody
```