Driven N-Pendulum
MANUSCRIPT

William F. Barnes
[1]

February 23, 2023

Contents

Chapter 1

Driven N-Pendulum

Abstract

The equations of motion for the N-pendulum are derived from Lagrange mechanics and simulated using the RK4 approximation.

1.1 Problem Setup

1.1.1 N-Pendulum

The N-pendulum consists of N point masses $\{m_j\}$ connected by rigid rods $\{l_j\}$ such that the l_jth rod connects m_{j-1} to m_j, where $j = 1, 2, 3, \ldots, N$. The end of rod l_1 (opposite m_1) is represented by the massless point $(x_0(t), y_0(t))$. Each rod makes an angle θ_j with respect to the vertical line defined by $\theta = -\pi/2$. Depicted in Fig. 1.1 is an example system with $N = 4$.

1.1.2 Driving Effects

The point $(x_0(t), y_0(t))$ is externally manipulated to drive the system. The entire N-pendulum is also subject to a uniform gravitational field g, presumed downward along $\theta = -\pi/2$. Finally, we may also account for the effects of a damping fluid that acts to hinder motions in a prescribed fashion.

1.1.3 Evolution

As is the case for the double- or triple-pendulum, the N-pendulum is a classically *chaotic* system. This is to say (i) similar initial conditions don't

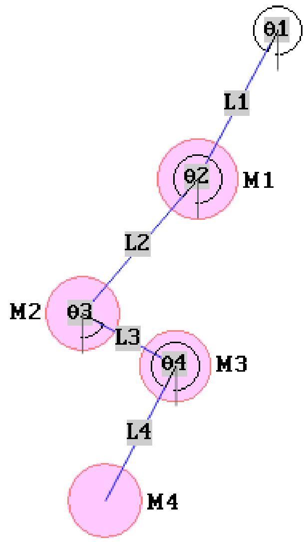

Figure 1.1: Four-body N-pendulum.

necessarily lead to similar outcomes, and (ii) evolution is generally 'unpredictable' when the system has sufficient energy.

1.2 Physical Setup

1.2.1 Positions

Using a polar coordinate representation, the position variables are represented by the angles θ_j.

$$x_j(t) = x_0(t) + \sum_{k=1}^{j} l_k \sin(\theta_k) \tag{1.1}$$

$$y_j(t) = y_0(t) - \sum_{k=1}^{j} l_k \cos(\theta_k) \tag{1.2}$$

1.2.2 Velocities

Differentiating (1.1)-(1.2) with respect to t yield the velocities $x_j(t)$, $y_j(t)$. The time derivative of any angle θ_j is the *angular velocity* ω_j:

$$\omega_j(t) = \frac{d}{dt}\theta_j = \dot{\theta}_j$$

$$\dot{x}_j(t) = \dot{x}_0(t) + \sum_{k=1}^{j} l_k \omega_k \cos(\theta_k)$$

$$\dot{y}_j(t) = \dot{y}_0(t) + \sum_{k=1}^{j} l_k \omega_k \sin(\theta_k)$$

Square of the Velocity

With the velocities $x_j(t)$, $y_j(t)$ written down, we may as well calculate the square of each:

$$\begin{aligned}
\dot{x}_j^2(t) &= \dot{x}_0^2(t) + 2\dot{x}_0(t)\sum_{k=1}^{j} l_k\omega_k\cos(\theta_k) + \left(\sum_{k=1}^{j} l_k\omega_k\cos(\theta_k)\right)^2 \\
&= \dot{x}_0^2(t) + 2\dot{x}_0(t)\sum_{k=1}^{j} l_k\omega_k\cos(\theta_k) + \sum_{k=1}^{j} l_k^2\omega_k^2\cos^2(\theta_k) \\
&\quad + 2\sum_{k,k'\neq j} l_k l_{k'}\omega_k\omega_{k'}\cos(\theta_k)\cos(\theta_{k'})
\end{aligned}$$

$$\begin{aligned}
\dot{y}_j^2(t) &= \dot{y}_0^2(t) + 2\dot{y}_0(t)\sum_{k=1}^{j} l_k\omega_k\sin(\theta_k) + \left(\sum_{k=1}^{j} l_k\omega_k\sin(\theta_k)\right)^2 \\
&= \dot{y}_0^2(t) + 2\dot{y}_0(t)\sum_{k=1}^{j} l_k\omega_k\sin(\theta_k) + \sum_{k=1}^{j} l_k^2\omega_k^2\sin^2(\theta_k) \\
&\quad + 2\sum_{k,k'\neq j} l_k l_{k'}\omega_k\omega_{k'}\sin(\theta_k)\sin(\theta_{k'})
\end{aligned}$$

The sum

$$v_j^2(t) = x_j^2(t) + y_j^2(t)$$

simplifies somewhat nicely from the above:

$$\begin{aligned} v_j^2(t) &= \dot{x}_0^2(t) + \dot{y}_0^2(t) \\ &\quad + 2\left(\dot{x}_0(t)\sum_{k=1}^{j} l_k\omega_k\cos(\theta_k) + \dot{y}_0(t)\sum_{k=1}^{j} l_k\omega_k\sin(\theta_k)\right) \\ &\quad + 2\sum_{k=1}^{j} l_k^2\omega_k^2 + \sum_{k,k'\neq j} l_k l_{k'}\omega_k\omega_{k'}\cos(\theta_k - \theta_{k'}) \end{aligned} \tag{1.3}$$

1.3 Energy

Now we express the total kinetic energy T and total potential U of the system. All constraints relative to the problem, i.e. masses, lengths, etc., must be built into the equations for T and U.

1.3.1 Kinetic Energy

Starting with the definition of the kinetic energy and plugging in the details, we write

$$T = \sum_{j=1}^{N}\frac{1}{2}m_j v_j^2(t) = \sum_{j=1}^{N}\frac{1}{2}m_j\left(\dot{x}_j^2(t) + \dot{y}_j^2(t)\right) ,$$

which, after substituting (1.3), develops into a very mean equation:

$$\begin{aligned} T &= \frac{1}{2}\left(\sum_{j=1}^{N} m_j\right)\left(\dot{x}_0^2(t) + \dot{y}_0^2(t)\right) \\ &\quad + \sum_{j=1}^{N}\left(m_j\dot{x}_0(t)\sum_{k=1}^{j} l_k\omega_k\cos(\theta_k) + m_j\dot{y}_0(t)\sum_{k=1}^{j} l_k\omega_k\sin(\theta_k)\right) \\ &\quad + \frac{1}{2}\sum_{j=1}^{N}\left(m_j\sum_{k=1}^{j} l_k^2\omega_k^2\right) + \sum_{j=1}^{N}\left(m_j\sum_{k,k'\neq j} l_k l_{k'}\omega_k\omega_{k'}\cos(\theta_k - \theta_{k'})\right) \end{aligned} \tag{1.4}$$

1.3.2 Potential Energy

Doing a similar exercise for the potential energy, we write

$$U = \sum_{j=1}^{N} m_j g y_j = \sum_{j=1}^{N} m_j g\left(y_0(t) - \sum_{k=1}^{j} l_k\cos(\theta_k)\right) ,$$

ending up at

$$U = g\left(\sum_{j=1}^{N} m_j\right) y_0\left(t\right) - g\sum_{j=1}^{N}\left(m_j\sum_{k=1}^{j} l_k\cos\left(\theta_k\right)\right) . \tag{1.5}$$

1.4 Equations of Motion

1.4.1 Lagrangian Formulation

In terms of the total kinetic energy T and total potential energy U, we write the *Lagrangian* of the system

$$L = T - U .$$

1.4.2 Derivative Exercise

To navigate for the minefield we've entered, let's do the intermediate exercise of calculating

$$\frac{\partial\tilde{T}}{\partial\omega_n} \qquad \text{given} \qquad \tilde{T} = \frac{1}{2}\sum_{j=1}^{N}\left(m_j\sum_{k=1}^{j} l_k^2\omega_k^2\right) .$$

Proceeding with care, we find:

$$\begin{aligned}
\frac{\partial\tilde{T}}{\partial\omega_n} &= \frac{1}{2}\sum_{j=1}^{N}\left(m_j\sum_{k=1}^{j} l_k^2\frac{\partial\left(\omega_k^2\right)}{\partial\omega_k}\delta_{nk}\right) = \sum_{j=1}^{N}\left(m_j\sum_{k=1}^{j} l_k^2\omega_k\delta_{nk}\right) \\
\frac{\partial\tilde{T}}{\partial\omega_n} &= m_1\sum_{k=1}^{1} l_k^2\omega_k\delta_{nk} + m_2\sum_{k=1}^{2} l_k^2\omega_k\delta_{nk} + m_3\sum_{k=1}^{3} l_k^2\omega_k\delta_{nk} + \cdots \\
&\quad + m_n\sum_{k=1}^{n} l_k^2\omega_k\delta_{nk} + \cdots + m_N\sum_{k=1}^{N} l_k^2\omega_k\delta_{nk} \\
\frac{\partial\tilde{T}}{\partial\omega_n} &= \left(\sum_{j=n}^{N} m_j\right) l_n^2\omega_n
\end{aligned}$$

As expected, the dependence on ω is reduces from power 2 to power 1. The nontrivial part of this result is what happened to the sums: the k-sum vanished, and the j-sum has a new lower limit.

1.4.3 Kronecker Delta

The quantity

$$\delta_{nk} = \begin{cases} 1 & n = k \\ 0 & n \neq k \end{cases}$$

is called the *Kronecker delta*. The Kronecker delta is ultimately a convenience of notation, along with its counterpart:

$$\tilde{\delta}_{nk} = \begin{cases} 0 & n = k \\ 1 & n \neq k \end{cases}$$

1.4.4 Equations of Motion

Per standard prescription, the equations of motion of the system are calculated from L by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega_j}\right) - \frac{\partial L}{\partial \theta_j} = 0 \qquad\qquad j = 1, 2, 3, \ldots, N \tag{1.6}$$

Term 1 of 2

$$\begin{aligned}
\frac{\partial L}{\partial \omega_n} &= \frac{\partial T}{\partial \omega_n} - \cancel{\frac{\partial U}{\partial \omega_n}} \\
&= \sum_{j=n}^{N} \left(m_j \dot{x}_0(t)\, l_n \cos(\theta_n) + m_j \dot{y}_0(t)\, l_n \sin(\theta_n)\right) \\
&\quad + \sum_{j=n}^{N} \left(m_j l_n^2 \omega_n\right) + \sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} \tilde{\delta}_{nk} l_n l_k \omega_k \cos(\theta_n - \theta_k)\right)
\end{aligned}$$

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \omega_n}\right) &= \sum_{j=n}^{N} \left(m_j \ddot{x}_0(t)\, l_n \cos(\theta_n) - m_j \dot{x}_0(t)\, l_n \omega_n \sin(\theta_n)\right) \\
&\quad + \sum_{j=n}^{N} \left(m_j \ddot{y}_0(t)\, l_n \sin(\theta_n) + m_j \dot{y}_0(t)\, l_n \omega_n \cos(\theta_n)\right) \\
&\quad + \sum_{j=n}^{N} \left(m_j l_n^2 \dot{\omega}_n\right) + \sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} \tilde{\delta}_{nk} l_n l_k \dot{\omega}_k \cos(\theta_n - \theta_k)\right) \\
&\quad - \sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} l_n l_k \tilde{\delta}_{nk} \omega_k (\omega_n - \omega_k) \sin(\theta_n - \theta_k)\right)
\end{aligned}$$

Term 2 of 2

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_n} &= \frac{\partial T}{\partial \theta_n} - \frac{\partial U}{\partial \theta_n} \\
&= \sum_{j=n}^{N} \left(-m_j \dot{x}_0\left(t\right) l_n \omega_n \sin\left(\theta_n\right) + m_j \dot{y}_0\left(t\right) l_n \omega_n \cos\left(\theta_n\right)\right) \\
&\quad - \sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} \tilde{\delta}_{nk} l_n l_k \omega_n \omega_k \sin\left(\theta_n - \theta_k\right) \right) - g \sum_{j=n}^{N} m_j l_n \sin\left(\theta_n\right)
\end{aligned}
$$

Sum of Terms

$$
\begin{aligned}
0 &= \frac{d}{dt}\left(\frac{\partial L}{\partial \omega_n}\right) - \frac{\partial L}{\partial \theta_n} \\
0 &= \sum_{j=n}^{N} \Big(m_j \ddot{x}_0\left(t\right) l_n \cos\left(\theta_n\right) - \cancel{m_j \dot{x}_0\left(t\right) l_n \omega_n \sin\left(\theta_n\right)} \Big) \\
&\quad + \sum_{j=n}^{N} \Big(m_j \ddot{y}_0\left(t\right) l_n \sin\left(\theta_n\right) + \cancel{m_j \dot{y}_0\left(t\right) l_n \omega_n \cos\left(\theta_n\right)} \Big) \\
&\quad + \sum_{j=n}^{N} \left(m_j l_n^2 \dot{\omega}_n \right) + \sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} l_n l_k \dot{\omega}_k \cos\left(\theta_n - \theta_k\right) \right) \\
&\quad - \sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} l_n l_k \omega_k \left(\cancel{\omega_n} - \omega_k\right) \sin\left(\theta_n - \theta_k\right) \right) \\
&\quad - \sum_{j=n}^{N} \Big(\cancel{-m_j \dot{x}_0\left(t\right) l_n \omega_n \sin\left(\theta_n\right)} + \cancel{m_j \dot{y}_0\left(t\right) l_n \omega_n \cos\left(\theta_n\right)} \Big) \\
&\quad + \cancel{\sum_{j=n}^{N} \left(m_j \sum_{k=1}^{j} l_n l_k \omega_n \omega_k \sin\left(\theta_n - \theta_k\right) \right)} - g \sum_{j=n}^{N} m_j l_n \sin\left(\theta_n\right)
\end{aligned}
$$

Result

$$
\begin{aligned}
0 = &\left(\sum_{j=n}^{N} m_j\right)\left(\ddot{x}_0\left(t\right) l_n \cos\left(\theta_n\right) + \ddot{y}_0\left(t\right) l_n \sin\left(\theta_n\right)\right) \\
&+ \sum_{j=n}^{N}\left(m_j \sum_{k=1}^{j} \tilde{\delta}_{nk} l_n l_k \left(\dot{\omega}_k \cos\left(\theta_n - \theta_k\right) + \omega_k^2 \sin\left(\theta_n - \theta_k\right)\right)\right) \\
&+ \left(\sum_{j=n}^{N} m_j\right) l_n^2 \dot{\omega}_n - g\left(\sum_{j=n}^{N} m_j\right) l_n \sin\left(\theta_n\right)
\end{aligned} \tag{1.7}
$$

Amazingly, the entirety of our efforts so far have boiled down to (1.7).

1.5 Matrix Setup

1.5.1 Acceleration Matrix

The equations of motion (1.7) are clearly a hopeless tangle of θ_j's, ω_j's, along with all constraints (masses, lengths), and external forces on the N-pendulum. Nonetheless, the set of equations generated by the above can be described conveniently with an 'acceleration matrix' as follows:

$$
\begin{bmatrix}
A_{11} & A_{12} & A_{13} & \cdots & A_{1j} & \cdots & A_{1N} \\
A_{21} & A_{22} & A_{23} & \cdots & \cdots & \cdots & \cdots \\
A_{31} & A_{32} & A_{33} & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
A_{j1} & \cdots & \cdots & \cdots & A_{jj} & \cdots & A_{jN} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
A_{N1} & \cdots & \cdots & \cdots & A_{Nj} & \cdots & A_{NN}
\end{bmatrix}
\begin{bmatrix}
\dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \\ \cdots \\ \dot{\omega}_j \\ \cdots \\ \dot{\omega}_N
\end{bmatrix}
=
\begin{bmatrix}
b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_j \\ \cdots \\ b_N
\end{bmatrix}
$$

In more condensed notation, the above reads

$$
A\frac{d}{dt}\vec{\omega} = \vec{b}\,, \tag{1.8}
$$

reminiscent of Newton's second law.

1.5.2 Matrix Components

Another way of expressing the matrix equation (1.8) is using index notation, namely

$$
\sum_{k=1}^{N} A_{nk}\dot{\omega}_k\tilde{\delta}_{nk} + \sum_{k=1}^{N} A_{nk}\delta_{nk}\dot{\omega}_k = b_n\,.
$$

Comparing this to the equations of motion (1.7), we may write

$$\sum_{j=n}^{N}\left(m_j\sum_{k=1}^{j}\tilde{\delta}_{nk}l_nl_k\left(\dot{\omega}_k\cos\left(\theta_n-\theta_k\right)\right)\right)+\left(\sum_{j=n}^{N}m_j\right)l_n^2\dot{\omega}_n=b_n\ .$$

b-Vector

The nth component of the vector $\vec{b}$ is already evident:

$$\begin{aligned}b_n=&-\left(\sum_{j=n}^{N}m_j\right)\left(\ddot{x}_0\left(t\right)l_n\cos\left(\theta_n\right)+\ddot{y}_0\left(t\right)l_n\sin\left(\theta_n\right)\right)\\&+g\left(\sum_{j=n}^{N}m_j\right)l_n\sin\left(\theta_n\right)\\&-\sum_{j=n}^{N}\left(m_j\sum_{k=1}^{j}\tilde{\delta}_{nk}l_nl_k\left(\omega_k^2\sin\left(\theta_n-\theta_k\right)\right)\right)\end{aligned}\tag{1.9}$$

Off-Diagonal Components

Introducing the Kronecker-delta-like relation

$$\sigma_{kj}=\begin{cases}0 & k>j\\1 & k\leq j\end{cases}\quad ,$$

the off-diagonal components of A come from:

$$\begin{aligned}\sum_{k=1}^{N}A_{nk}\dot{\omega}_k\tilde{\delta}_{nk}&=\sum_{j=n}^{N}\left(m_j\sum_{k=1}^{j}\tilde{\delta}_{nk}l_nl_k\left(\dot{\omega}_k\cos\left(\theta_n-\theta_k\right)\right)\right)\\\sum_{k=1}^{N}A_{nk}\dot{\omega}_k\tilde{\delta}_{nk}&=\sum_{k=1}^{N}\sum_{j=n}^{N}m_j\sigma_{kj}\tilde{\delta}_{nk}l_nl_k\left(\dot{\omega}_k\cos\left(\theta_n-\theta_k\right)\right)\\A_{nk}&=\sum_{j=n}^{N}m_j\sigma_{kj}l_nl_k\left(\cos\left(\theta_n-\theta_k\right)\right)\qquad n\neq k\end{aligned}\tag{1.10}$$

Diagonal Components

$$\sum_{k=1}^{N} A_{nk}\delta_{nk}\dot{\omega}_k = \left(\sum_{j=n}^{N} m_j\right) l_n^2 \dot{\omega}_n$$

$$\sum_{k=1}^{N} A_{nk}\delta_{nk}\dot{\omega}_k = \sum_{k=1}^{N}\left(\sum_{j=n}^{N} m_j l_k^2\right)\delta_{nk}\dot{\omega}_k$$

$$A_{nn} = \left(\sum_{j=n}^{N} m_j\right) l_n^2 \tag{1.11}$$

1.5.3 Inverse Matrix

Given the matrix formulation (1.8) of the equations of motion (1.7) for the N-pendulum, the task on hand is to 'dig out' the vector $d\vec{\omega}/dt$ from the middle of the equation. This is done using the inverse of the matrix A, denoted A^{-1}, such that:

$$A\frac{d}{dt}\vec{\omega} = \vec{b}$$

$$\left(A^{-1}A\right)\frac{d}{dt}\vec{\omega} = A^{-1}\vec{b}$$

$$\frac{d}{dt}\vec{\omega} = A^{-1}\vec{b} \tag{1.12}$$

Of course, the above is contingent on the inverse A^{-1} being easily attained, which can in practice be computationally expensive.

1.5.4 First-Order System

After attaining N equations of motion from the above, the total system is represented by $2N$ linear differential equations:

$$\frac{d}{dt}\begin{bmatrix}\theta_1\\ \omega_1\\ \theta_2\\ \omega_2\\ \cdots\\ \theta_N\\ \omega_N\end{bmatrix} = \begin{bmatrix}\omega_1\\ p_1\\ \omega_2\\ p_2\\ \cdots\\ \omega_N\\ p_N\end{bmatrix}$$

The jth angular frequency ω_j depends on some function p_j based on known quantities. In order to condense the above, we deploy vector notation to equivalently write

$$\frac{d}{dt}\vec{q} = \vec{p}\,(\vec{q}\,(t)) = \vec{p}\,.$$

Note that the vector $\vec{p}$ is equivalent to $A^{-1}\vec{b}$.

1.6 Computational Setup

1.6.1 Augmented First-Order System

As part of the first-order system characterizing the N-pendulum, let us augment the *time* variable onto $d\vec{q}/dt = \vec{p}$ such that

$$\frac{d}{dt}\begin{bmatrix} \theta_1 \\ \omega_1 \\ \theta_2 \\ \omega_2 \\ \cdots \\ \theta_N \\ \omega_N \\ t \end{bmatrix} = \begin{bmatrix} \omega_1 \\ p_1 \\ \omega_2 \\ p_2 \\ \cdots \\ \omega_N \\ p_N \\ 1 \end{bmatrix} .$$

To be clear, the quantities θ_j, ω_j relate to q_j via

$$q_j = \begin{cases} \omega_k & k = 2j \\ \theta_k & k = 2j - 1 \\ t & j = 2N + 1 \end{cases} .$$

To proceed, one imagines a 'small' value Δt such that if $\vec{q}\,(t)$ represents the system at time t, then $\vec{q}\,(t + \Delta t)$ represents the system at a later time with all coordinates updated.

1.6.2 Runge-Kutta Approximation

It's possible to show (in a variety of ways) that a straightforward Euler's method-like attack on the above system doesn't yield accurate results. Rather, it turns out that a fourth-order method called the *Runge-Kutta* approximation happens to perfectly approximate the N-pendulum. This approximation is done by updating the q_jth coordinate such that:

$$q_j\,(t_0 + \Delta t) = q_j\,(t_0) + \frac{1}{6}\,(f\,(1, j) + 2f\,(2, j) + 2f\,(3, j) + f\,(4, j))$$

1.6.3 Intermediate Calculations

The Runge-Kutta approximation depends on four calculations to evaluate the intermediate functions $f(1,j)$, …, $f(4,j)$. These must be calculated in series:
Step 1:

$$q_j^{(1)}(t_0) = q_j(t_0) \qquad\qquad f(1,j) = \Delta t \cdot p\left(q_j^{(1)}(t_0)\right)$$

Step 2:

$$q_j^{(2)}(t_0) = q_j(t_0) + (1/2)\, f(1,j) \qquad\qquad f(2,j) = \Delta t \cdot p\left(q_j^{(2)}(t_0)\right)$$

Step 3:

$$q_j^{(3)}(t_0) = q_j(t_0) + (1/2)\, f(2,j) \qquad\qquad f(3,j) = \Delta t \cdot p\left(q_j^{(3)}(t_0)\right)$$

Step 4:

$$q_j^{(4)}(t_0) = q_j(t_0) + f(3,j) \qquad\qquad f(4,j) = \Delta t \cdot p\left(q_j^{(4)}(t_0)\right)$$

1.6.4 Pseudocode Implementation

```
For (j = 1 To 2N+1) { qtmp(j) = q(j)              }
Update vector p(qtmp)
For (j = 1 To 2N+1) { f(1, j) = dt * p(j)         }

For (j = 1 To 2N+1) { qtmp(j) = q(j) + f(1, j) / 2 }
Update vector p(qtmp)
For (j = 1 To 2N+1) { f(2, j) = dt * p(j)         }

For (j = 1 To 2N+1) { qtmp(j) = q(j) + f(2, j) / 2 }
Update vector p(qtmp)
For (j = 1 To 2N+1) { f(3, j) = dt * p(j)         }

For (j = 1 To 2N+1) { qtmp(j) = q(j) + f(3, j)     }
Update vector p(qtmp)
For (j = 1 To 2N+1) { f(4, j) = dt * p(j)         }
```

```
For (j = 1 To 2N+1) {
    q(j) = q(j) + (f(1, j) + (2 * f(2, j))
                           + (2 * f(3, j))
                           + f(4, j)) / 6
}
```
