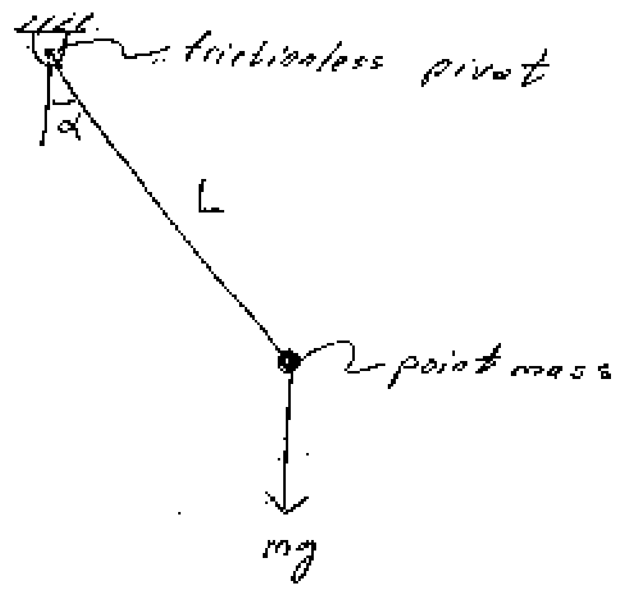
① Pendulum Dynamics

9/25/02
NJD

x) Point Mass Pendulum with Large Angle of oscillation



Assumptions

x) Mass is concentrated at single point (no rotational inertia)

x) No frictional losses

x) constant gravitational field

x) pendulum is a rigid body

Notation

$L \Rightarrow$ length of pendulum

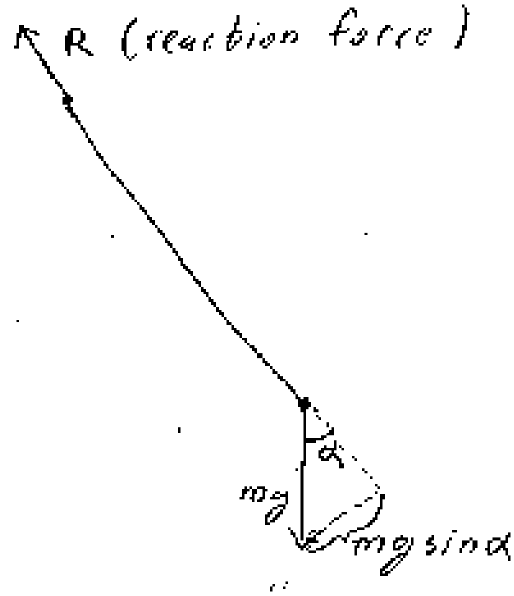
$m \Rightarrow$ mass of pendulum

$g \Rightarrow$ gravity

$\alpha \Rightarrow$ angle of pendulum

Free Body Diagram

(show forces on body)



②

Equation of Motion

Rotation of a rigid body about a pivot

$\tau = I\ddot{\alpha}$ $\qquad$ $\tau \Rightarrow$ torque about pivot

$I \Rightarrow$ rotational inertia $I = \int r^2 dm$

$\ddot{\alpha} \Rightarrow$ angular acceleration

For Point Mass Pendulum

$\tau = -mg\sin\alpha L$ (Note right hand rule counterclockwise as positive)

$I = mL^2$

Equation of motion:

$$-mg\sin\alpha\, L = mL^2\ddot{\alpha} \qquad \Big/ \times \frac{1}{mL^2}$$

$$\boxed{\ddot{\alpha} + \frac{g}{L}\sin\alpha = 0}$$

If we could assume small angles of oscillation

Then $\sin\alpha \approx \alpha$ (see excel sheet graph)

For small angles $\ddot{\alpha} + \frac{g}{L}\alpha \approx 0$

Guess a solution in the form

$$\alpha(t) = A\cos(\omega t) \qquad \left(\text{more general form} \quad \alpha(t) = A\cos(\omega t + \theta)\right)$$

(3)

$$\frac{d\alpha}{dt} = \dot{\alpha} = -wA\sin(wt)$$

$$\frac{d^2\alpha}{dt^2} = \ddot{\alpha} = -w^2 A\cos(wt)$$

substituted "guess" back into equation of motion

$$-w^2 A \cancel{\cos(wt)} + \frac{g}{L} A \cancel{\cos(wt)} = 0$$

$$-w^2 = \frac{g}{L}$$

$$\boxed{w = \sqrt{\frac{g}{L}}}$$

natural frequency for small vibration

For a pendulum released at α_0 with a zero initial velocity, and small α

$$A = \alpha_0$$

$$\alpha(t) = \alpha_0 \cos\left(\sqrt{\frac{g}{L}}\, t\right)$$

(4) Solve Differential Equation to Simulate Motion

Euler method ⇒ simple, to implement but not most efficient

Based upon the definition of a derivative

$$\frac{dx}{dt}(t) \approx \frac{x(t+\delta t) - x(t)}{\delta t} \quad \text{for small } \delta t$$

$$\rightarrow \quad x(t+\delta t) \approx x(t) + \frac{dx}{dt}(t)\,\delta t$$

Numerical Implementation

$$x_{i+1} = x_i + \frac{dx_i}{dt}\,\delta t$$

Implement in Pendulum Equation of Motion

Initial Conditions are given: $\alpha(t=0) = \alpha_0$

$\dot{\alpha}(t=0) = \dot{\alpha}_0$

$$\alpha_{i+1} = \alpha_i + \dot{\alpha}_i\,\delta t$$

$$\dot{\alpha}_{i+1} = \dot{\alpha}_i + \ddot{\alpha}_i\,\delta t \quad (\text{need } \ddot{\alpha}_i)$$

$$\ddot{\alpha}_i = -\frac{g}{L}\sin\alpha_i \quad (\text{from equation of motion})$$

→ No small angle approximation!

(5) Code Implementation

```
% initial conditions
t(1) = 0
α(1) = α_0
α̇(1) = α̇_0
tstep => define small time step

for i=1:n

    t(i+1) = t(i) + tstep
    α(i+1) = α(i) + α̇(i) tstep
    α̈(i) = -(g/L) sin(α_i)
    α̇(i+1) = α̇(i) + α̈(i) tstep
end
```

⑥

Comments About the Euler Method

⇒ Not numerically efficient

⇒ Requires small step size to achieve accuracy and avoid an unstable solution

⇒ Most ODE solvers in CAE packages use recursive algorithms like the Runga-Kutta

⇒ But I still use the Euler method every once in a while to verify simulations

⇒ Some codes (e.g. Working Model) let one select the Euler method for the same reason.

(10) Programing Tips:

% Meaningful variable names (doesn't cost more)

% concise documentation written during coding.

% function calls & modularity