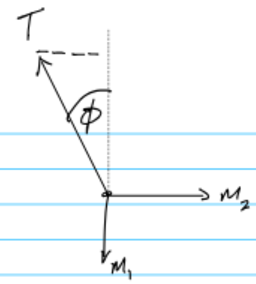
1

1.



$m_1 = 65g \pm 0.2g$ $T_1 = m_1 \cdot g$

$m_2 = 85g \pm 0.3g$ $T_2 = m_2 \cdot g$

2) $\phi = Tan^{-1}\left(\frac{-T_2}{T_1}\right)$ $T_3 = \sqrt{T_1^2+T_2^2}$

$$\delta T_3 = \sqrt{\left(\frac{\partial T_3}{\partial m_1}\, dm_1\right)^2 + \left(\frac{\partial T_3}{\partial m_2}\cdot dm_2\right)^2}$$

$$\delta \phi = \sqrt{\left(\frac{\partial \phi}{\partial T_1}\cdot dm_1\right)^2 + \left(\frac{\partial \phi}{\partial T_2}\cdot dm_2\right)^2}$$

where $\frac{\partial \phi}{\partial T_1} = \frac{-m_2(-m_1^{-2})}{1+\left(\frac{-m_2}{m_1}\right)^2}$ $\frac{\partial \phi}{\partial T_2} = -\frac{1}{m_1 + \frac{-m_2^2}{m_1}}$

$\frac{\partial T_3}{m_1} = \frac{m_1 g}{\sqrt{m_1^2+m_2^2}}$ $\frac{\partial T_3}{m_2} = \frac{m_2 g}{\sqrt{m_1^2+m_2^2}}$

b&c)

* See Mathematica *

1.b&c Mathematica code:

1.b&c)

```
In[ ]:= m1 = 0.065; dm1 = 0.0002; m2 = 0.085; dm2 = 0.0003; L1 = 0.025; dL1 = 0.005; L2 = 0.030; dL2 = 0.007; μ = 0.0031; dμ = 0.0002; g = 9.81;
t1 = m1 * g; t2 = m2 * g; T1 = (m1 + L1 * μ); T2 = (m2 + L2 * μ);

T[x_, y_] := Sqrt[(g * x)^2 + (g * y)^2];
ϕ[x_, y_] := ArcTan[y/x] * 180/π;

T[x, y] /. {x → m1, y → m2}
ϕ[x, y] /. {x → m1, y → m2}

Sqrt[((D[T[x, y], x] * dm1)^2 + (D[T[x, y], y] * dm2)^2)] /. {x → m1, y → m2}
Sqrt[((D[ϕ[x, y], x] * dm1)^2 + (D[ϕ[x, y], y] * dm2)^2)] /. {x → m1, y → m2}

Out[ ]= 1.04972

Out[ ]= 52.5946

Out[ ]= 0.00262406

Out[ ]= 0.129453
```

2.

$L = 1.25 \pm 0.002\,m$

$h = 0.037 \pm 0.001\,m$

a) A sphere rolls down distance $x = 0.85 \pm 0.003\,m$

Not accounting for moments of inertia

$$x = x_0 + v_0 t + \frac{1}{2} a t^2 \Rightarrow t = \sqrt{\frac{2x}{a}} \Rightarrow t = \sqrt{\frac{2xL}{gh}}$$

$$\delta t = \sqrt{\left(\frac{\partial t}{\partial x} \cdot dx\right)^2 + \left(\frac{\partial t}{\partial L} \cdot dL\right)^2 \left(\frac{\partial t}{\partial h} dh\right)^2}$$

$$\frac{\partial t}{\partial x} = \frac{\partial}{\partial x}\left(\frac{2xL}{gh}\right)^{1/2} = \frac{1}{2}\left(\frac{2xL}{gh}\right)^{-1/2} \cdot \frac{2L}{gh}$$

$$\frac{\partial t}{\partial L} = \frac{1}{2}\left(\frac{2xL}{gh}\right)^{-1/2} \cdot \frac{2x}{gh}$$

$$\frac{\partial t}{\partial h} = \frac{1}{2}\left(\frac{2xL}{gh}\right)^{-1/2} \cdot \frac{-2xh}{(gh)^2}$$

2.b

In [2]:

```
import sympy as sym
from numpy import sqrt

g=9.81

x,h,L=sym.symbols("x,h,L")

time=sym.sqrt(2*x*L/g/h)

dt_dx = sym.diff(time, x)
dt_dh = sym.diff(time, h)
dt_dL = sym.diff(time, L)

L,dL=1.25,0.002
h,dh=0.037,0.001
x,dx=0.85,0.003

sub = {'x': x,'h': h,'L': L,}

t=time.subs(sub).evalf(6)

dt=sqrt( (float(dt_dx.subs(sub).evalf(6))*dx)**2 + (float(dt_dh.subs(sub).evalf(6))*dh)**2 +(float(dt_dL.subs(sub)
print(f"Elasped time:{t:.2f} +- {dt:.2f} s")
```

```
Elasped time:2.42 +- 0.03 s
```

3.a

In [3]:
```
from numpy import pi
L=sym.symbols("L")

period=2*pi*sym.sqrt(L/g)

dT_dL=sym.diff(period,L)

L,dL=0.85,0.003
m,dm=0.250,0.002

sub={'L':L}

T=period.subs(sub).evalf(6)
dT=sqrt( (float(dT_dL.subs(sub).evalf(6))*dL)**2 )

print(f"Predicted period of pendulum without accounting for \nmoments of inertia: {T:.3f} +- {dT:.3f} s")
```

```
Predicted period of pendulum without accounting for
moments of inertia: 1.850 +- 0.003 s
```

3.b

In [4]:
```
L,T=sym.symbols("L T")

gravity=4*pi**2*L/T**2

dg_dL=sym.diff(gravity,L)
dg_dT=sym.diff(gravity,T)

L,dL=0.85,0.003
T,dT=1.83,0.06

sub={'L':L,'T':T}

g=gravity.subs(sub).evalf(6)

dg=sqrt( (float(dg_dL.subs(sub).evalf(6))*dL)**2 + (float(dg_dT.subs(sub).evalf(6))*dT)**2 )

print(f"Predicted gravitational constant without accounting for \nmoments of inertia: {g:.1f} +- {dg:.1f} m/s^2")
```

```
Predicted gravitational constant without accounting for
moments of inertia: 10.0 +- 0.7 m/s^2
```

3.c

In [13]:
```
# initialize which variables are symbolic
L_cm, T, m_b, r, L_s, L=sym.symbols("L_cm T m_b r L_s L")

L_string=L-r
dL_string_dL=sym.diff(L_string, L)
dL_string_dr=sym.diff(L_string, r)

r,dr=0.0281,0.0002
L,dL=0.85,0.003
m_s=0.0065
sub={'L':L,'r':r}
L_s1,dL_s1=L_string.subs(sub).evalf(6),sqrt( (float(dL_string_dL.subs(sub).evalf(6))*dL)**2 + (float(dL_string_dr.

L_cm, T, m_b, r, L, L_s=sym.symbols("L_cm T m_b r L L_s")

L_centermass=(m_s*L_s/2+m_b*L)/(m_s+m_b)
dL_centermass_dm_b=sym.diff(L_centermass,m_b)
dL_centermass_dL=sym.diff(L_centermass,L)
dL_centermass_dL_s=sym.diff(L_centermass,L_s)

m_b,dm_b=0.250, 0.002
r,dr=0.0281,0.0002
L,dL=0.85,0.003
m_s=0.0065

sub={'L':L,'r':r, 'L_s':L_s1, 'm_b':m_b,}
L_cm1,dL_cm1=L_centermass.subs(sub).evalf(6), sqrt( (float(dL_centermass_dm_b.subs(sub).evalf(6))*dm_b)**2 + (floa

L_cm, T, m_b, r, L, L_s=sym.symbols("L_cm T m_b r L L_s")

gravity=4*pi**2*(2*m_b*r**2/5+m_b*L**2+m_s*L_s**2/3)/T**2/(m_s+m_b)/L_cm

dg_dL_cm=sym.diff(gravity,L_cm)
dg_dT=sym.diff(gravity,T)
dg_dm_b=sym.diff(gravity,m_b)
dg_dL_s=sym.diff(gravity,L_s)
dg_dr=sym.diff(gravity,r)
dg_dL=sym.diff(gravity,L)

T,dT=1.83,0.06
m_b,dm_b=0.250, 0.002
r,dr=0.0281,0.0002
L,dL=0.85,0.003
m_s=0.0065
sub={'L_cm':L_cm1,'T':T,'m_b':m_b,'r':r, 'L_s': L_s1, 'L':L}

g=gravity.subs(sub).evalf(6)

dg=sqrt( (float(dg_dL.subs(sub).evalf(6))*dL)**2 + (float(dg_dL_cm.subs(sub).evalf(6))*dL_cm1)**2 + (float(dg_dT.s

print(f"Predicted gravitational constant with accounting for \nmoments of inertia: {g:.2f} +- {dg:.2f} m/s^2")
```

```
Predicted gravitational constant with accounting for
moments of inertia: 9.98 +- 0.66 m/s^2
```

In [16]:
```
g=9.8

m_b,r,L,L_s,L_cm = sym.symbols('m_b r L L_s L_cm')

time=2*pi*sym.sqrt( (2*m_b*r**2/5+m_b*L**2+m_s*L_s**2/3)/((m_s+m_b)*g*L_cm) )

dt_dm_b=sym.diff(time,m_b)
dt_dr=sym.diff(time,r)
dt_dL=sym.diff(time,L)
dt_dL_s=sym.diff(time,L_s)
dt_dL_cm=sym.diff(time,L_cm)

m_b,dm_b=0.250, 0.002
r,dr=0.0281,0.0002
L,dL=0.85,0.003

sub={'m_b':m_b,'r':r, 'L_cm':L_cm1,'T':T,'L_s': L_s1, 'L':L}

t=time.subs(sub).evalf(6)
dt=sqrt( (float(dt_dm_b.subs(sub).evalf(6))*dm_b)**2 + (float(dt_dr.subs(sub).evalf(6))*dr)**2 + (float(dt_dL.subs

print(f"Predicted period of pendulum with accounting for \nmoments of inertia: {t:.3f} +- {dt:.3f} s")
```

```
Predicted period of pendulum with accounting for
moments of inertia: 1.847 +- 0.007 s
```

Mathematica Section for bonus

Problem 3

Part a

```
ClearAll[L, dL, g, T, δT]
L := 0.85 (*string length in meters*)
dL := 0.003 (*Uncertainty of string length in meters*)
g := 9.8 (*m/s^2*)
T[L_, g_] := 2 π √(L / g)
T[L, g]
δT = √((D[T[x, g], x] * dL) ^2) /. x → L
```

Out[]= 1.85045

Out[]= 0.00326549

Part b

```
In[]:= ClearAll[g, T]
T := 1.83 (*Period in seconds*)
dT := 0.06 (*Uncertainty of period in seconds*)
```

$$g[L\_, T\_] := \frac{4 \pi^2 L}{T^2}$$

```
g[L, T]
```

$$\delta g = \sqrt{(D[g[x, T], x] * dL)^2 + (D[g[L, y], y] * dT)^2} \text{ /. } \{x \to L, y \to T\}$$

Out[]= 10.0202

Out[]= 0.658013

Part c

Redoing Part a

```
ClearAll[ms, mbb, dmb, rr, dr, g, Ls, Ib, Is, II, m, Lcm, T, δT, L, LL]
LL := 0.85 (*string length in meters*)
ms := 6.5 / 1000 (*string mass in kg*)
mbb := 250 / 1000 (*Ball mass in kg*)
dmb := 2 / 1000 (*Ball mass uncertainty in kg*)
rr := 0.0281 (*Ball radius in meters*)
dr := 0.0002 (*Ball radius uncertainty in meters*)
g := 9.8 (*m/s^2*)
Ls := L - r
Ib := (2 mb * r^2)/5
Is := (ms * Ls^2)/3
II := Ib + Is
m := ms + mb
Lcm := (ms * Ls/2 + mb * L)/(ms + mb)
T[II_, m_, Lcm_] := 2 π Sqrt[II/(g * m * Lcm)]
T[II, m, Lcm] /. {L → LL, mb → mbb, r → rr}
δT = √((D[T[II, m, Lcm], L] * dL)^2 + (D[T[II, m, Lcm], mb] * dmb)^2 + (D[T[II, m, Lcm], r] * dr)^2) /. {L → LL, mb → mbb, r → rr}
```

Out[•]= 0.169942

Out[•]= 0.000699246

Redoing Part b

```
ClearAll[ms, mbb, dmb, rr, dr, g, Ls, Ib, Is, II, m, Lcm, TT, δg, L, LL, T]
LL := 0.85 (*string length in meters*)
ms := 6.5 / 1000 (*string mass in kg*)
mbb := 250 / 1000 (*Ball mass in kg*)
dmb := 2 / 1000 (*Ball mass uncertainty in kg*)
rr := 0.0281 (*Ball radius in meters*)
dr := 0.0002 (*Ball radius uncertainty in meters*)
TT := 1.83 (*Period of oscillation in seconds*)
Ls := L - r
```

$$Ib := \frac{2\, mb * r\hat{}2}{5} + mb * L\hat{}2$$

$$Is := \frac{ms * Ls^2}{3}$$

```
II := Ib + Is
m := ms + mb
```

$$Lcm := \frac{ms * \frac{Ls}{2} + mb * L}{ms + mb}$$

$$g[L\_, mb\_, r\_, T\_] := \frac{4\pi^2\, II}{m * Lcm * T^2}$$

```
g[L, mb, r, T] /. {L → LL, mb → mbb, r → rr, T → TT}
```

$$\delta g = \sqrt{\left((D[g[L, mb, r, T], L] * dL)^2 + (D[g[L, mb, r, T], mb] * dmb)^2 + (D[g[L, mb, r, T], r] * dr)^2 + (D[g[L, mb, r, T], T] * dT)^2\right)} \;/.\; \{L \to LL, mb \to mbb, r \to rr, T \to TT\}$$

```
9.98032

0.655393
```

4

In [6]:

```python
import sympy as sym
from numpy import sqrt

R,C,t,v0=sym.symbols("R C t v0")

v=v0*(1-sym.exp(-t/R/C))

dv_dR = sym.diff(v,R)
dv_dC = sym.diff(v,C)
dv_dt = sym.diff(v,t)
dv_dv0 = sym.diff(v,v0)

R,dR=810,8.1
C,dC=2500e-6,125e-6
t,dt=3,0.1
v0,dv0=2.0,0.03


sub = {'R':R,'C':C,"t":t,'v0':v0}

v=v.subs(sub).evalf(6)

dv=sqrt( (float(dv_dv0.subs(sub).evalf(6))*dv0)**2 + (float(dv_dR.subs(sub).evalf(6))*dR)**2 + (float(dv_dC.subs(s
print(f"Estimated voltage at time t = {t:.2f}: \n{v:.2f} +- {dv:.3f} volts")
```

```
Estimated voltage at time t = 3.00:
1.55 +- 0.047 volts
```

Mathematica code for bonus

4)

In[116]:=
```
V = 2.0; dV = 0.03;
R = 810; dR = 8.1;
t = 3.0; dt = 0.1;
C1 = 2500 × 10^-6; dC = 2500 * 5 / 100 * 10^-6;


f[a_, b_, c_, d_] := a * (1 - Exp[-b/(c * d)]);

f[a, b, c, d] /. {a → V, da → dV, b → t, db → dt, c → C1,
    dc → dC, d → R, dd → dR} // N
dV =
 √((D[f[a, b, c, d], a] * da)^2 + (D[f[a, b, c, d], b] * db)^2 +
     (D[f[a, b, c, d], c] * dc)^2 + (D[f[a, b, c, d], d] * dd)^2) /.
  {a → V, da → dV, b → t, db → dt, c → C1, dc → dC, d → R, dd → dR}
```

Out[121]= 1.5454

Out[122]= 0.0471237

In [ ]: