BPhO Computational Challenge

Models of motion

Dr Andrew French.
December 2023.

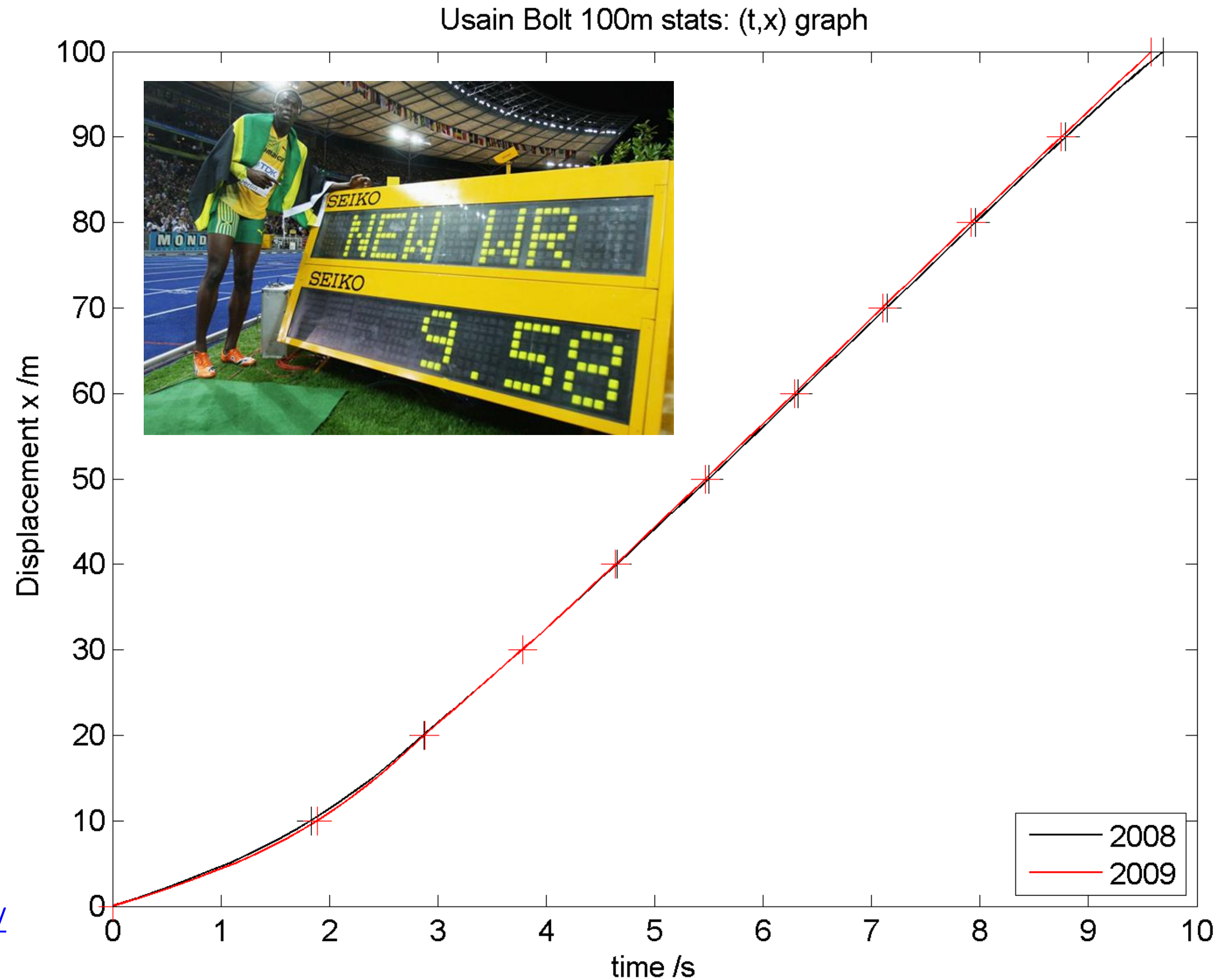
BPhO British Physics Olympiad

#1: The Kinematics of Usain Bolt

Two of Bolt's record-breaking 100m races. Time elapsed /s every 10m*

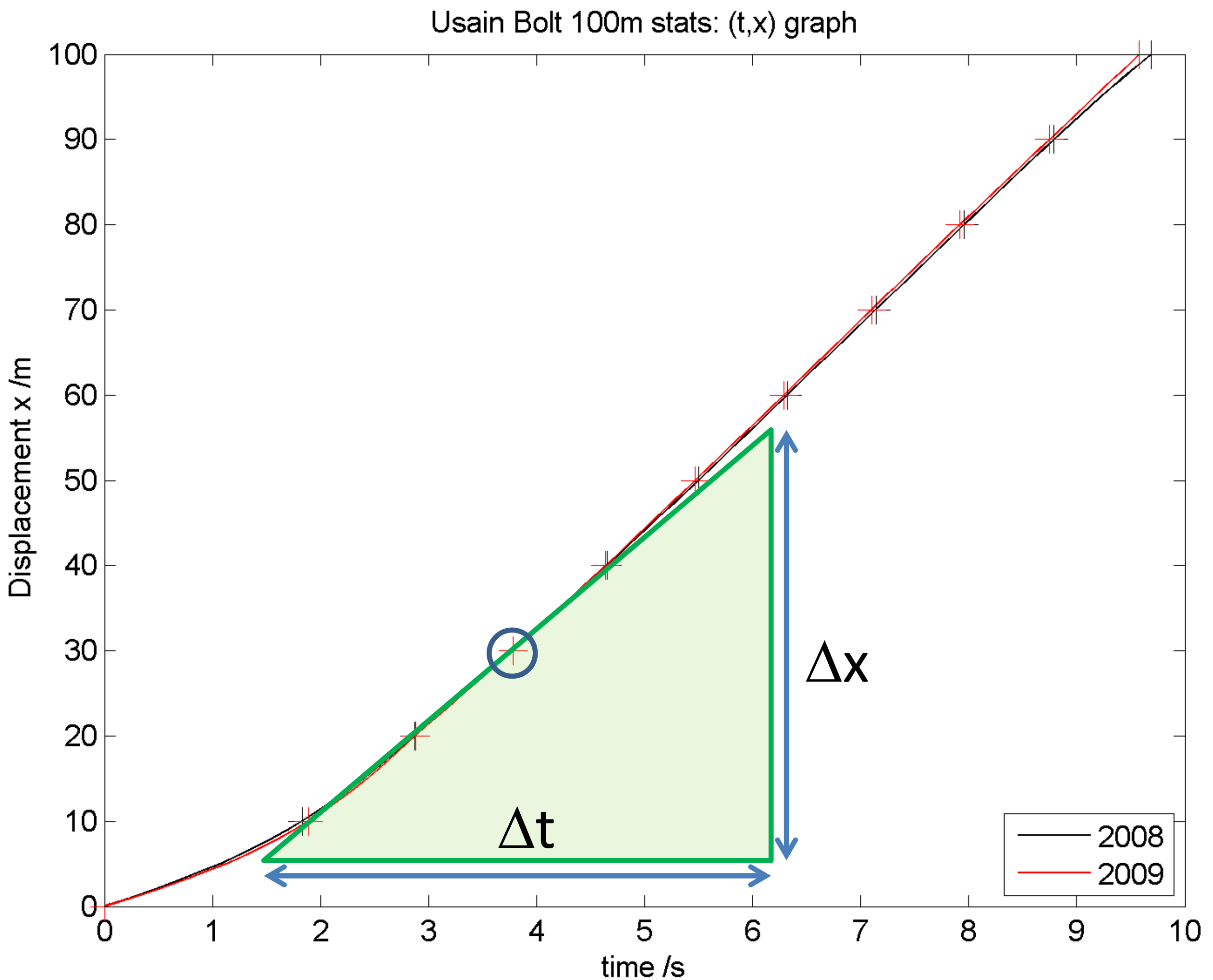
Bolt	10	20	30	40	50	60	70	80	90	100	
2008	1.83	2.87	3.78	4.65	5.5	6.32	7.14	7.96	8.79	9.69	Olympic final, Beijing
2009	1.89	2.88	3.78	4.64	5.47	6.29	7.10	7.92	8.75	9.58	World Champs, Berlin

Photo credit



*
http://rcuksportscience.wikispaces.com/file/view/Analysing+men+100m+Nspire.pdf

To find the **time, velocity graph** we could calculate the *gradient* of the (t,x) graph, at *different* times



At ◯

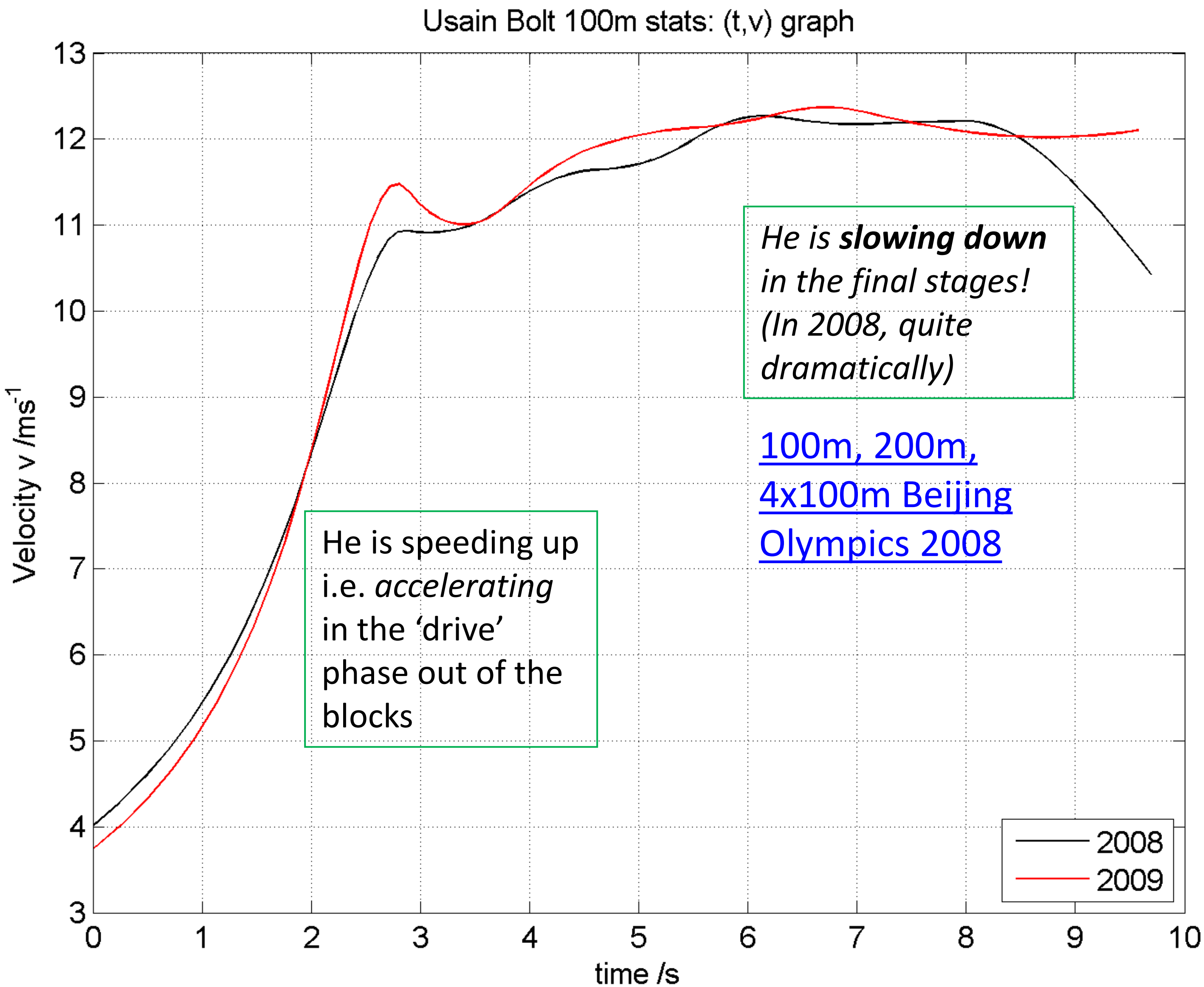
the *local gradient* is

$$\boxed{v = \frac{\Delta x}{\Delta t}}$$

$= 11.2 ms^{-1}$

THIS IS DONE BY FIRSTLY FITTING *CUBIC SPLINES* BETWEEN THE DATA POINTS

The graph below has been constructed from the *local gradients* calculated *every second* along a *smooth curve* drawn between the elapsed time data recorded at 10m intervals.

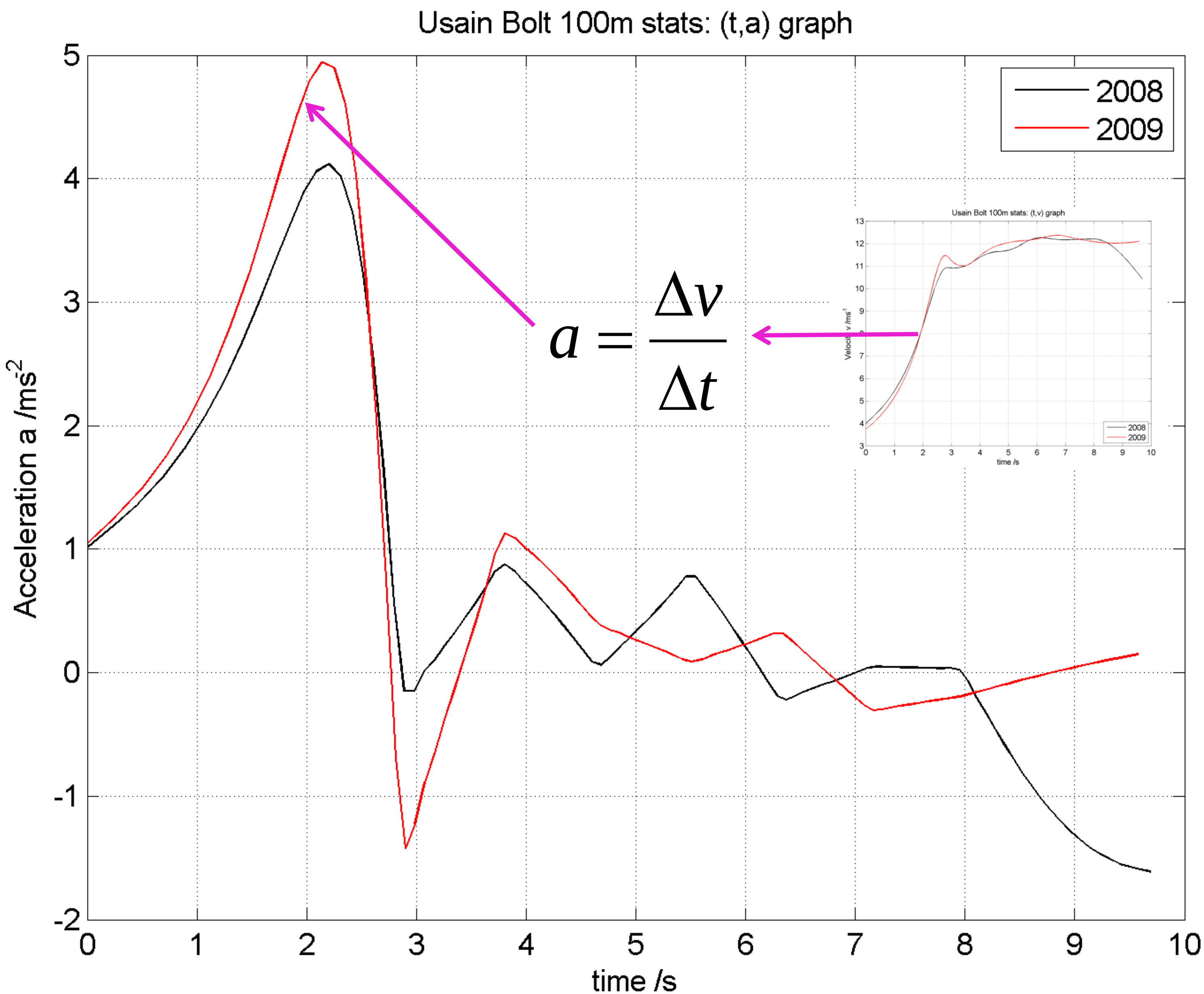


He is ***slowing down*** *in the final stages! (In 2008, quite dramatically)*

[100m, 200m, 4x100m Beijing Olympics 2008]

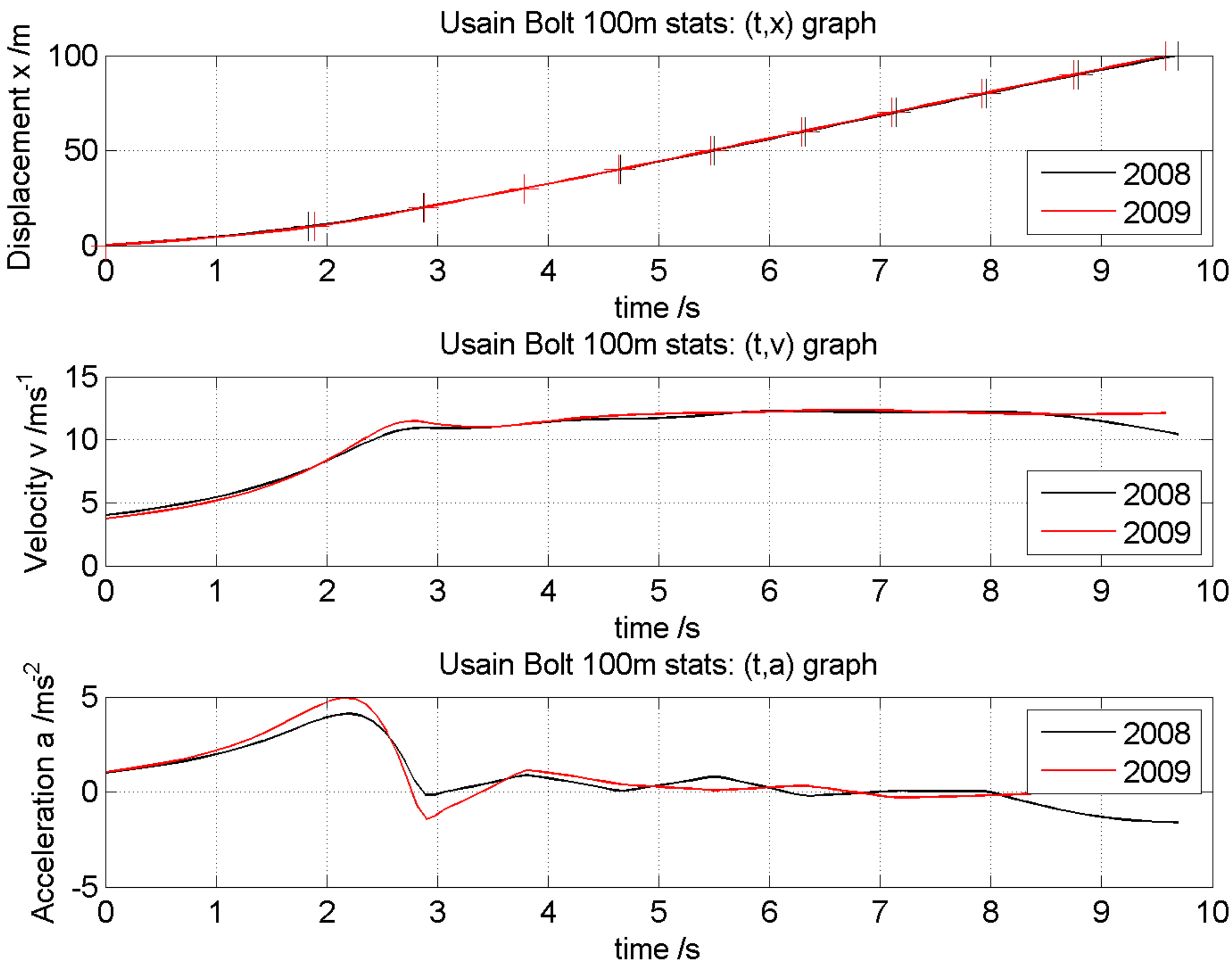
He is speeding up i.e. *accelerating* in the 'drive' phase out of the blocks

We can go one step further and find the graph of *acceleration vs time* by working out the local gradients of the (t,v) graph.



$$a = \frac{\Delta v}{\Delta t}$$

For a complete view we can compare (t,x), (t,v) and (t,a) traces. Note the time axis must be the *same scale* for each graph.

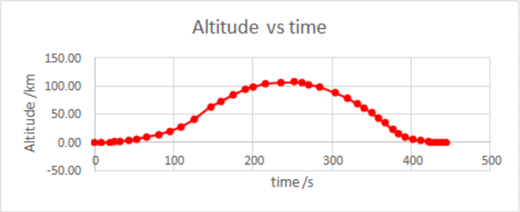


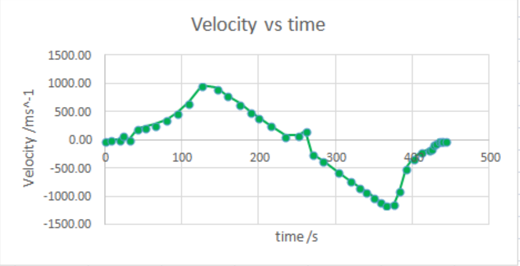
Youtube link 1 mile = 1609.3m 1ft = 0.3048m

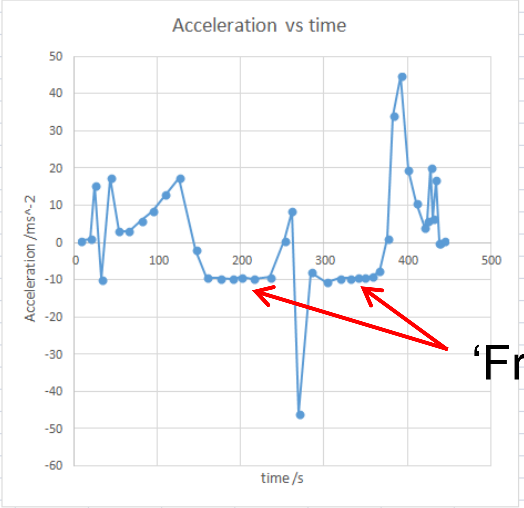
Kinematics of Blue Origin (New Shephard) launch. Data for rocket booster (which separates from capsule).

20/July/2020. Geoff Bezos (Amazon founder) + three other space tourists!

Time min	Time s	t /s	Altitude /ft	Velocity /mph	x /km	v /ms^-1	a /ms^-2
0	0	0	-2	0	0.00	0.00	
0	8	8	6	10	0.00	4.47	0.56
0	19	19	1110	35	0.34	15.65	1.02
0	24	24	2536	208	0.77	92.98	15.47
0	32	32	5647	31	1.72	13.86	-9.89
0	43	43	11707	458	3.57	204.74	17.35
0	53	53	19302	526	5.88	235.14	3.04
1	5	65	28769	609	8.77	272.24	3.09
1	20	80	44494	806	13.56	360.30	5.87
1	34	94	63223	1069	19.27	477.87	8.40
1	49	109	91965	1500	28.03	670.54	12.84
2	6	126	137204	2159	41.82	965.13	17.33
2	26	146	203398	2070	62.00	925.35	-1.99
2	39	159	239985	1794	73.15	801.97	-9.49
2	55	175	278503	1449	84.89	647.74	-9.64
3	10	190	307615	1122	93.76	501.57	-9.75
3	20	200	322077	916	98.17	409.48	-9.21
3	35	215	339091	591	103.35	264.19	-9.69
3	55	235	350085	179	106.71	80.02	-9.21
4	12	252	351210	197	107.05	88.06	0.47
4	21	261	346519	368	105.62	164.51	8.49
4	30	270	336308	-558	102.51	-249.44	-45.99
4	44	284	324430	-808	98.89	-361.20	-7.98
5	3	303	292043	-1261	89.01	-563.70	-10.66
5	19	319	258169	-1606	78.69	-717.93	-9.64
5	31	331	227370	-1865	69.30	-833.71	-9.65
5	40	340	201843	-2054	61.52	-918.20	-9.39
5	49	349	173458	-2245	52.87	-1003.58	-9.49
5	58	358	142876	-2425	43.55	-1084.04	-8.94
6	6	366	113551	-2561	34.61	-1144.84	-7.60
6	16	376	76387	-2537	23.28	-1134.11	1.07
6	23	383	49662	-2004	15.14	-895.84	34.04
6	32	392	30292	-1104	9.23	-493.52	44.70
6	41	401	18569	-711	5.66	-317.84	19.52
6	51	411	10145	-472	3.09	-211.00	10.68
7	1	421	3708	-383	1.13	-171.21	3.98
7	4	424	2092	-343	0.64	-153.33	5.96
7	8	428	539	-162	0.16	-72.42	20.23
7	11	431	313	-118	0.10	-52.75	6.56
7	14	434	66	-5	0.02	-2.24	16.84
7	18	438	33	-5	0.01	-2.24	0.00
7	19	439	12	-5	0.00	-2.24	0.00
7	24	444	0	0	0.00	0.00	0.45





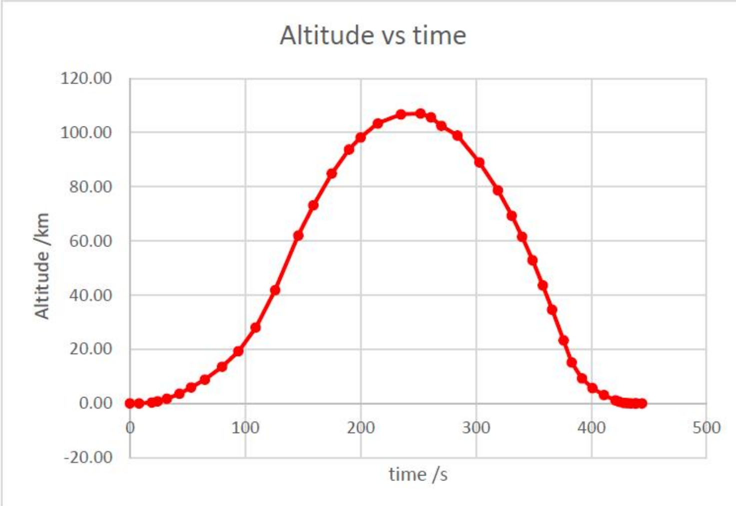


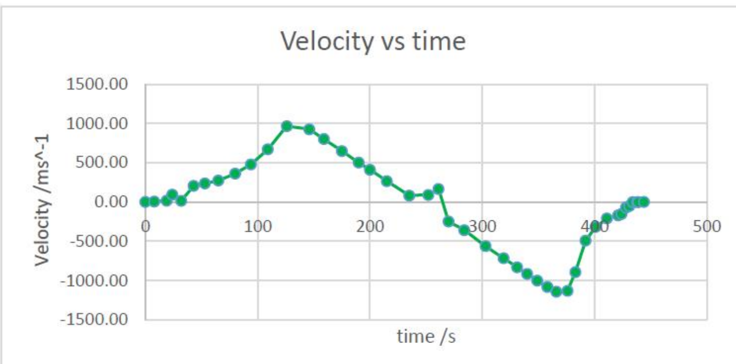
‘Free fall’ at -9.81m/s^2

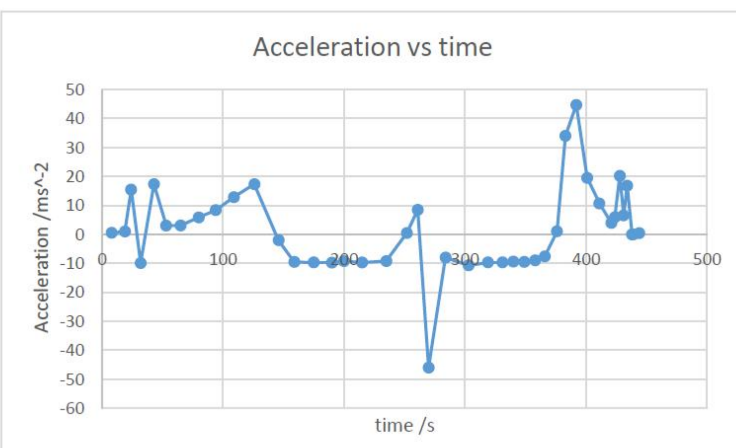
#2: Kinematics of Blue Origin’s *New Shepherd* launch on 20-July-2021

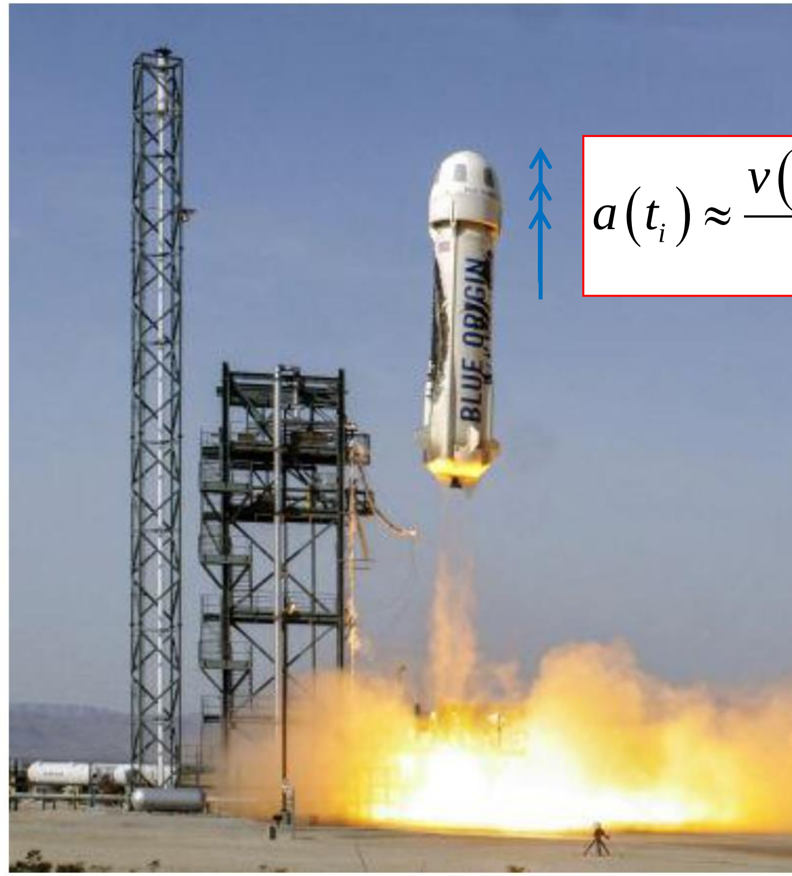
Acceleration estimated from velocity

$$a(t_i) \approx \frac{v(t_i) - v(t_{i-1})}{t_i - t_{i-1}}$$



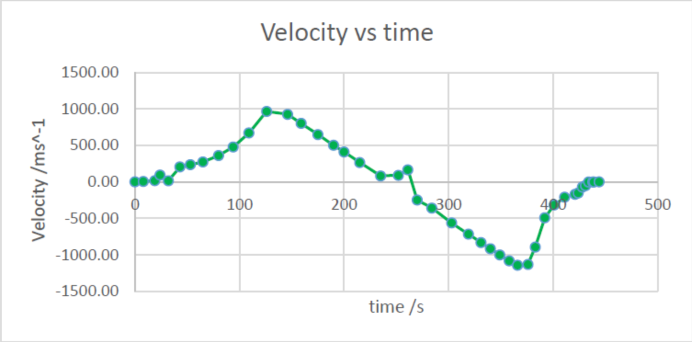




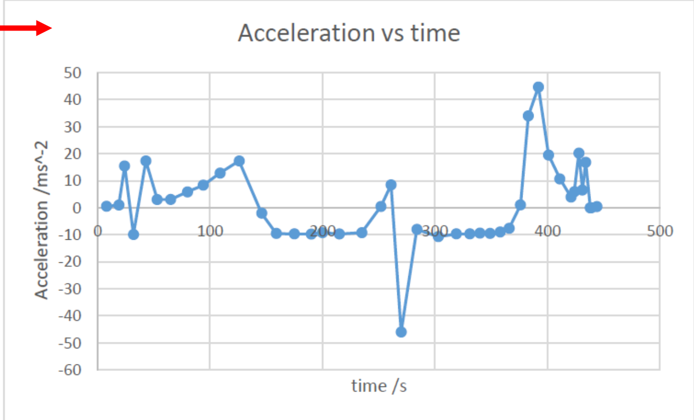


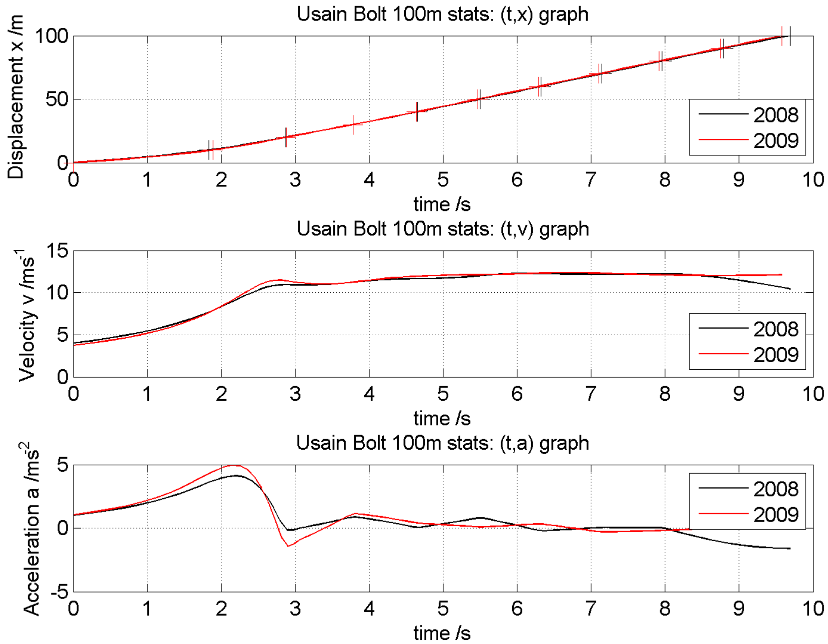
$$a(t_i) \approx \frac{v(t_i) - v(t_{i-1})}{t_i - t_{i-1}}$$

In summary: Kinematics provides a really good reason to wish to know *velocity* or *acceleration* from *displacement* measurements. **i.e. the gradient of the 'underlying curve.'** We may *not know* the functional form of the curve however – we may have to estimate it from the data.



#2: Bezos example: Assume a straight line between velocity measurements.

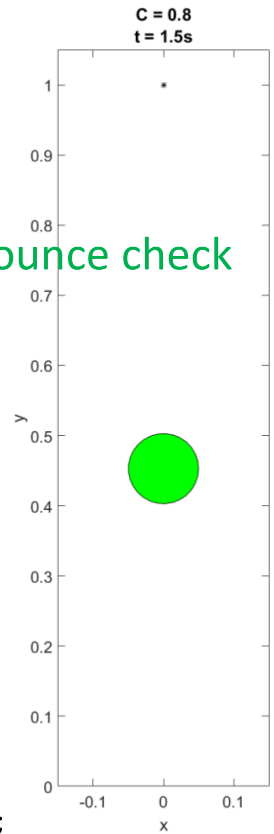




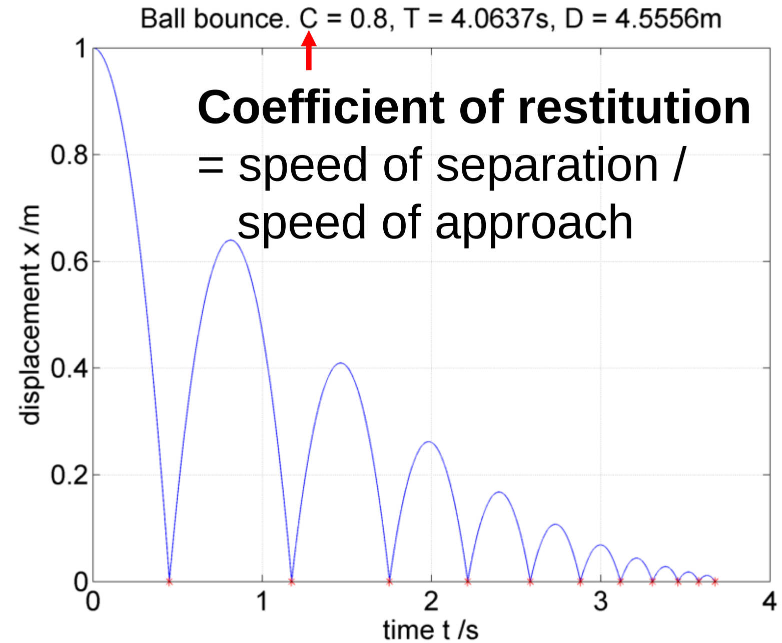
#1: Bolt example: Fit a *cubic spline* between a rolling set of four data points. We can then **differentiate** the cubic (yielding a quadratic).

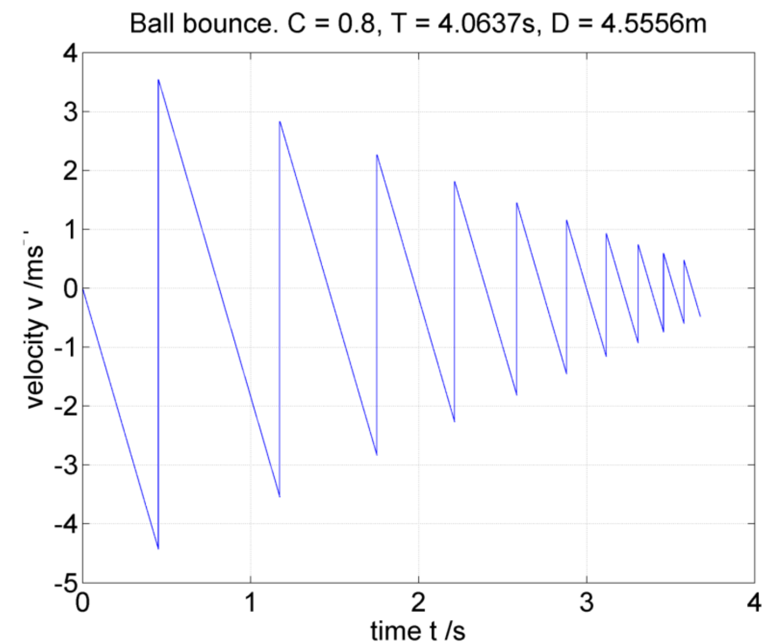
#3: Bouncing ball sim

```
n = 1; x = 1; v = 0; t = 0; g = 9.81; dt = 0.001;  C = 0.8;
max_bounce = 50; bounce = 0;

while bounce < max_bounce
   % Use constant acceleration motion between
   % time steps dt
   x(n+1) = x(n) - v(n)*dt - 0.5*g*dt^2;
   v(n+1) = v(n) + g*dt;
   t(n+1) = t(n) + dt;
   if ( x(n+1) < 0 ) && ( v(n+1) > 0 )   % Bounce check
      bounce = bounce + 1;
      v(n+1) = -C*v(n);
   end
   n = n + 1;
end

%Time to stop bouncing
T = 2*sqrt( 2*x(1)/g )*( 1/(1-C) - 0.5 );
```



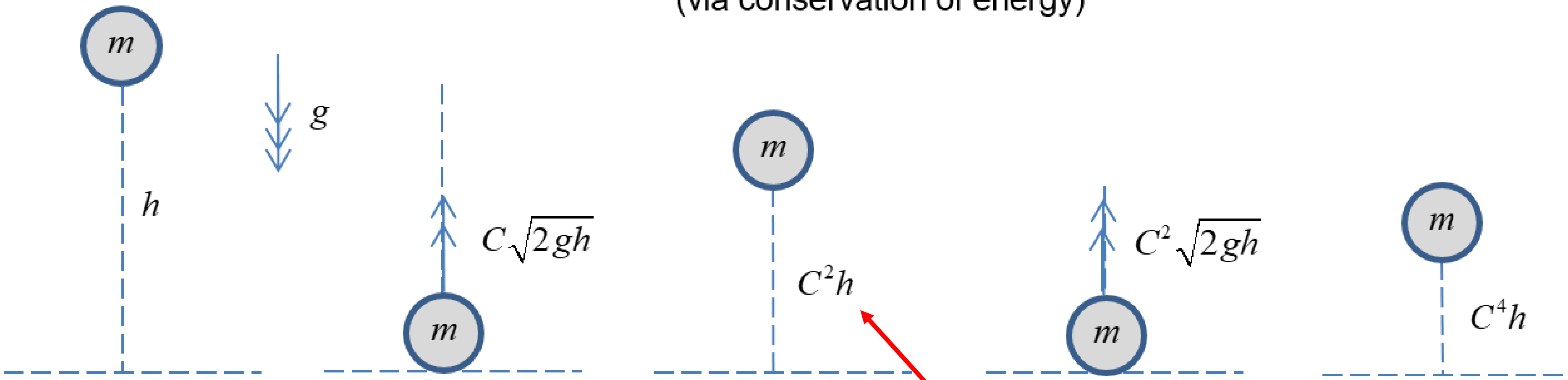

```
subplot(2,1,1); plot(t,x); xlabel('t /s'); ylabel('x (m)'); grid on;
title(['Ball Bounce. C=',num2str(C),', g=',num2str(g),' ms^{-2}, T=',num2str(T),' s.']);
subplot(2,1,2); plot(t,v); xlabel('t /s'); ylabel('v (m/s)'); grid on;
print(gcf,'bounce.png','-dpng','-r300’); close(gcf);
```





Ball bouncing on a horizontal surface

A ball is dropped from rest from vertical height h onto a horizontal floor. The impact velocity is $\sqrt{2gh}$ (via conservation of energy)



The **distance** travelled after *n* bounces is

$$D = h + 2C^2h + 2C^4h + ... + 2\left(C^2\right)^{n-1}h$$

$$\frac{D}{2h} + \tfrac{1}{2} = 1 + C^2 + \left(C^2\right)^2 + ... + \left(C^2\right)^{n-1}$$

Geometric progression

$$\frac{D}{2h} + \tfrac{1}{2} = \frac{1-C^{2n}}{1-C^2}$$

$$D = 2h\left(\frac{1-C^{2n}}{1-C^2} - \tfrac{1}{2}\right) \qquad \therefore \boxed{D_\infty = 2h\left(\frac{1}{1-C^2} - \tfrac{1}{2}\right)}$$

$$\tfrac{1}{2}mv^2 = mgH$$

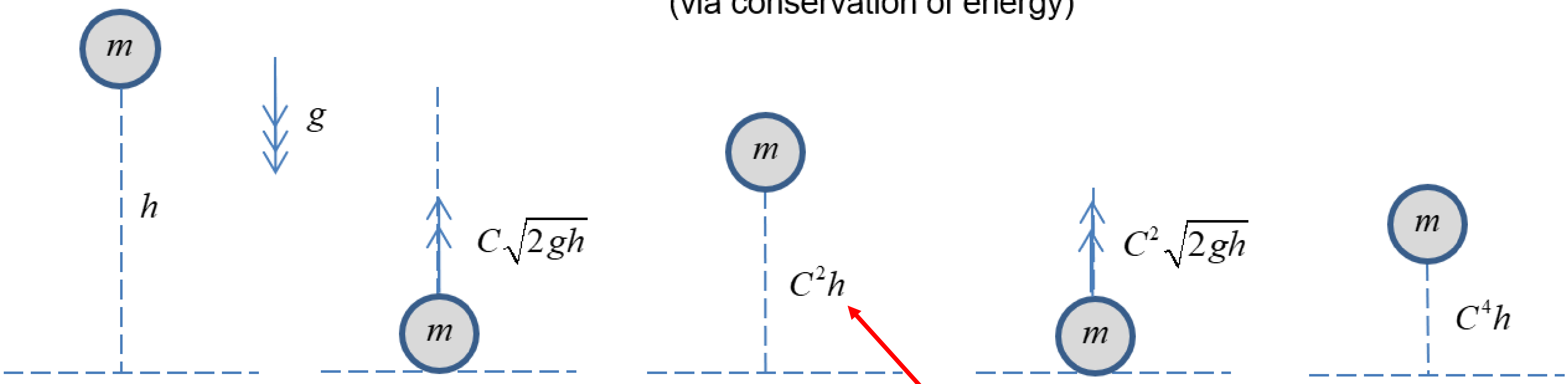
$$\therefore H \propto v^2$$

$$v^2 = 2gH$$

$$\frac{\left(C\sqrt{2gh}\right)^2}{2g} = C^2h$$

Ball bouncing on a horizontal surface

A ball is dropped from rest from vertical height h onto a horizontal floor. The impact velocity is $\sqrt{2gh}$ (via conservation of energy)



The **time** travelled after n bounces is

$$T=\sqrt{\frac{2h}{g}}+2C\sqrt{\frac{2h}{g}}+2C^2\sqrt{\frac{2h}{g}}+....+2C^{n-1}\sqrt{\frac{2h}{g}}$$

$$\frac{T}{2}\sqrt{\frac{g}{2h}}+\tfrac{1}{2}=1+C+C^2+...+C^{n-1}$$

Geometric progression

$$\frac{T}{2}\sqrt{\frac{g}{2h}}+\tfrac{1}{2}=\frac{1-C^n}{1-C}$$

$$T=2\sqrt{\frac{2h}{g}}\left(\frac{1-C^n}{1-C}-\tfrac{1}{2}\right)\qquad\therefore\ \boxed{T_\infty=2\sqrt{\frac{2h}{g}}\left(\frac{1}{1-C}-\tfrac{1}{2}\right)}$$

$$\tfrac{1}{2}mv^2=mgH$$

$$\therefore H\propto v^2$$

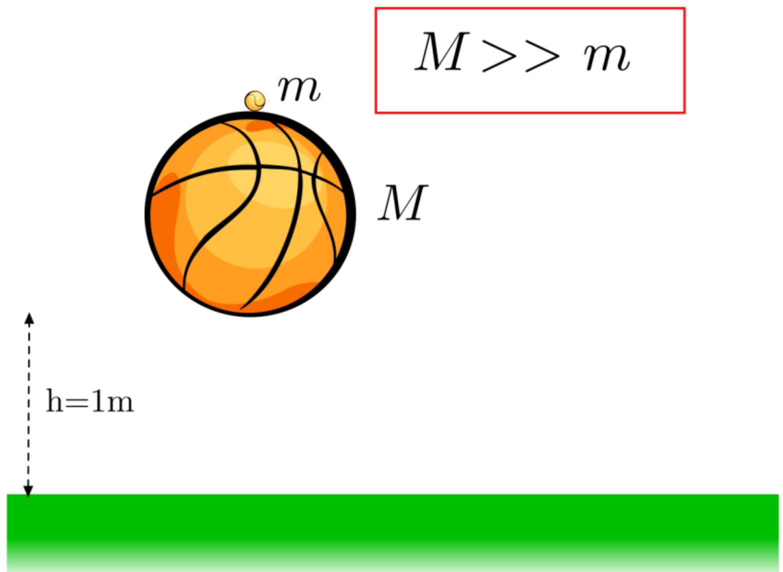
$$v^2=2gH$$

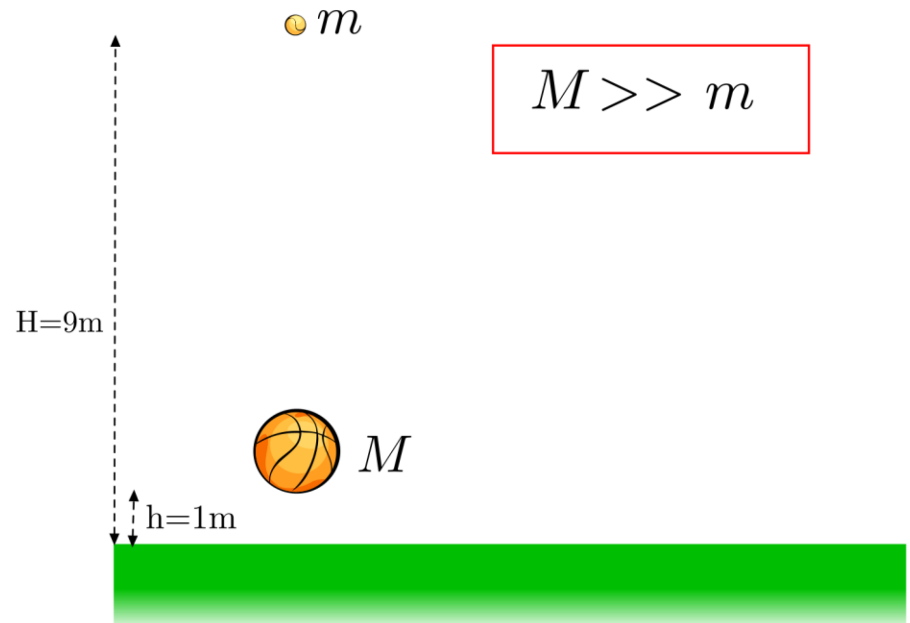
$$\frac{\left(C\sqrt{2gh}\right)^2}{2g}=C^2h$$

#4: Double ball bounce

Following collision, the smaller mass rises up to *nine times* the distance fallen!

Balls are dropped from rest





$$\boxed{\frac{H}{h} = \frac{v^2}{u^2} = \left(\frac{C_1 - \frac{m}{M} + C_2 C_1 + C_2}{1 + \frac{m}{M}}\right)^2}$$

All elastic collisions $C_1 = C_2 = 1 \Rightarrow$ $\boxed{\frac{H}{h} < \left(\frac{3 - \frac{m}{M}}{1 + \frac{m}{M}}\right)^2}$

You have a model … now make a calculator using a spreadsheet

DOUBLE BALL BOUNCE CALCULATOR

M /g	595	basket ball mass
m /g	57	tennis ball mass

Basket ball rise height / drop height	0.74
Tennis ball rise height /drop height	0.49

C1	0.9	coefficient of restitution: basket ball and ground
C2	0.7	coefficient of restitution: basket ball and tennis ball

Drop height /m: 1.00

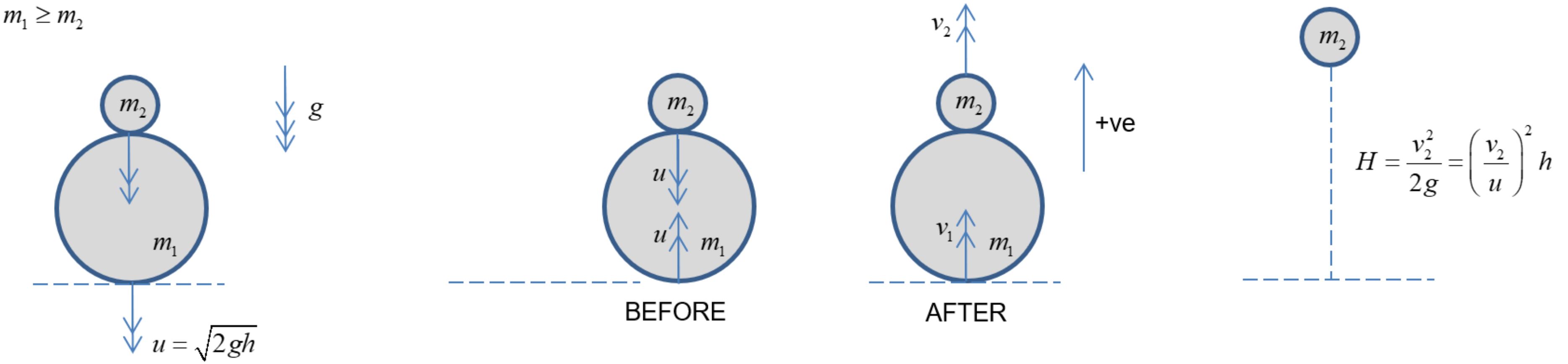
Predicted rise height of tennis ball /m: 3.56

$$\frac{H}{h} = \frac{v^2}{u^2} = \left(\frac{C_1 - \frac{m}{M} + C_2 C_1 + C_2}{1 + \frac{m}{M}} \right)^2$$

https://www.youtube.com/watch?v=Vk8YIuFTQ3g

Interesting scenario: two balls dropped together

$m_1 \geq m_2$



Both balls are dropped from height h. The lower (and more massive one) collides elastically with a hard floor

The balls then collide

Upper ball rises to height H

$$m_1 u - m_2 u = m_1 v_1 + m_2 v_2$$

Momentum conservation

$$C = \frac{v_2 - v_1}{2u} \qquad \therefore v_1 = v_2 - 2uC$$

Restitution

$$m_1 u - m_2 u = m_1 (v_2 - 2uC) + m_2 v_2$$

$$m_1 u - m_2 u + 2C m_1 u = v_2 (m_1 + m_2)$$

$$v_2 = \frac{(2C+1)m_1 - m_2}{m_1 + m_2} u$$

$$v_2 = \frac{(2C+1) - \frac{m_2}{m_1}}{1 + \frac{m_2}{m_1}} u \qquad H = \frac{v_2^2}{2g} = \left(\frac{v_2}{u}\right)^2 h$$

$$\boxed{\therefore H = \left(\frac{(2C+1) - \frac{m_2}{m_1}}{1 + \frac{m_2}{m_1}}\right)^2 h}$$

All collisions elastic

$$m_1 \gg m_2$$

$$v_2 = 3u$$

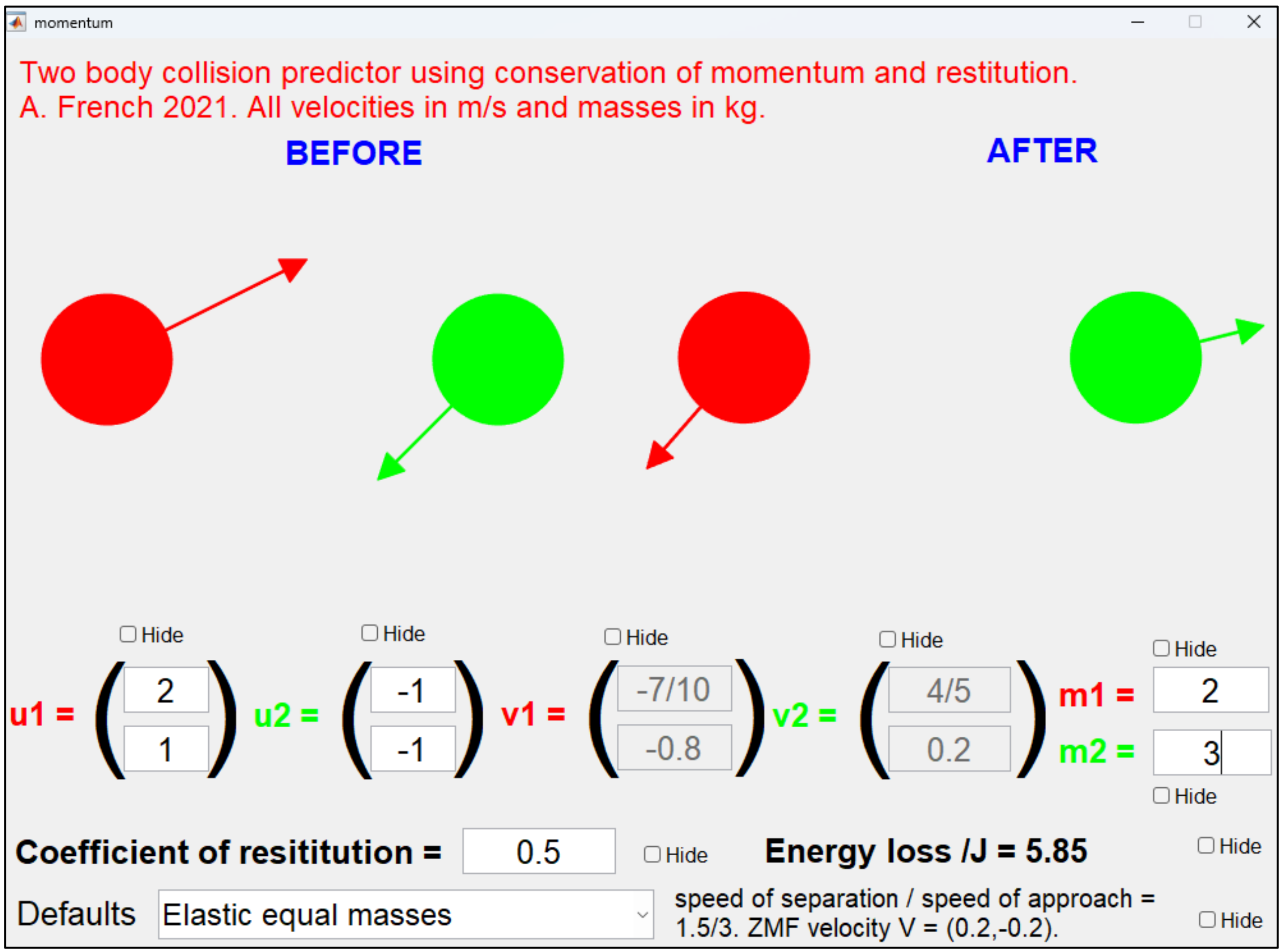
$$H = 9h$$

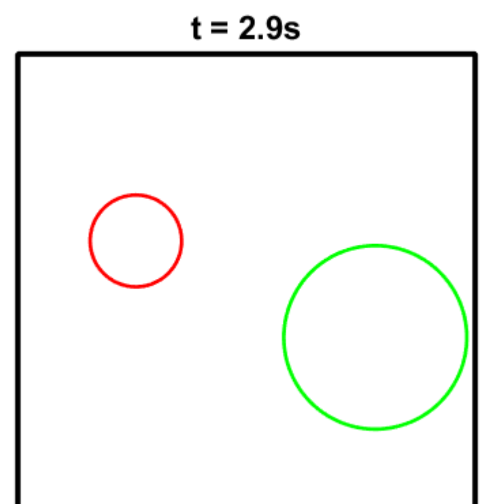
$$m_1(\mathbf{u}_1 - \mathbf{V}) + m_2(\mathbf{u}_2 - \mathbf{V}) = \mathbf{0} \qquad \therefore \mathbf{V} = \frac{m_1\mathbf{u}_1 + m_2\mathbf{u}_2}{m_1 + m_2}$$

After transforming back to the 'laboratory frame' from the 'Zero Momentum Frame' (ZMF), the resulting velocities $\mathbf{v}_{1,2}$ following collision are:

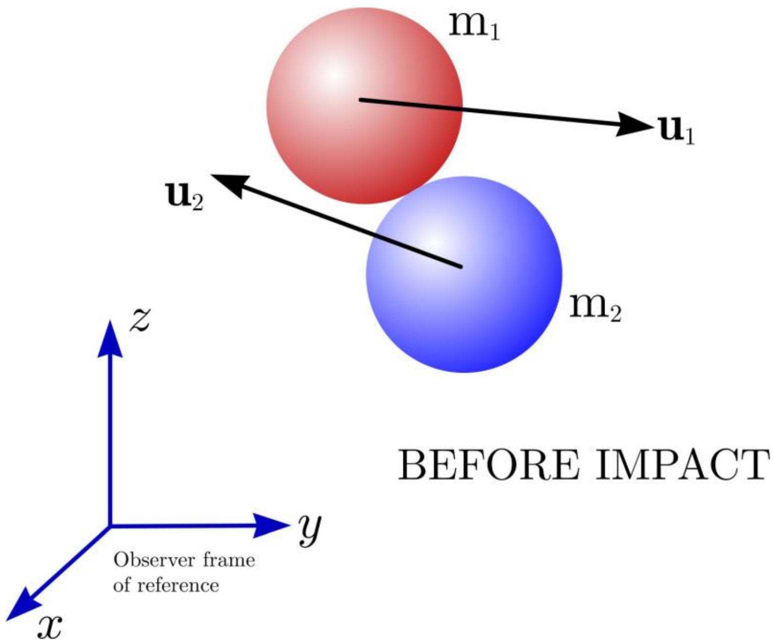
$$\mathbf{v}_{1,2} = -C(\mathbf{u}_{1,2} - \mathbf{V}) + \mathbf{V}$$

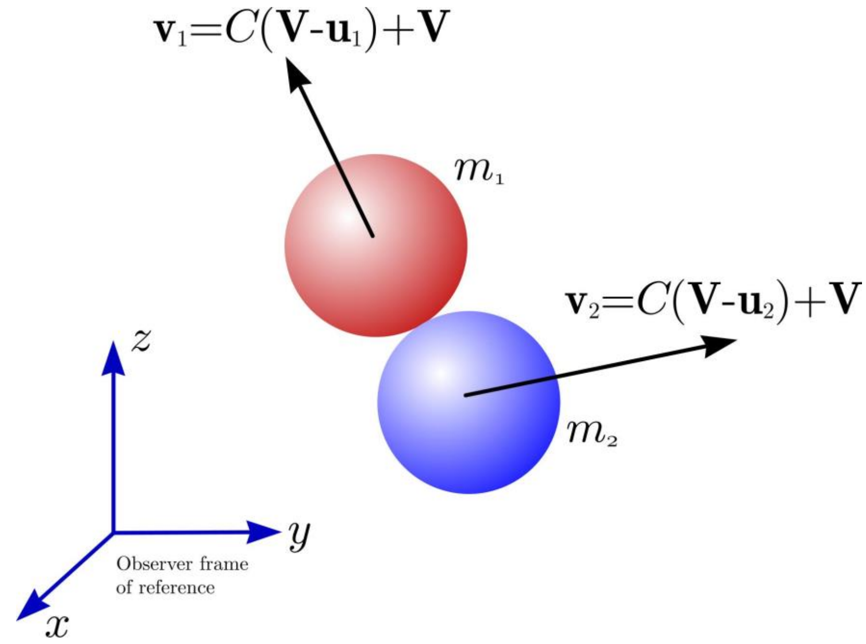
#5: Collisions and the ZMF



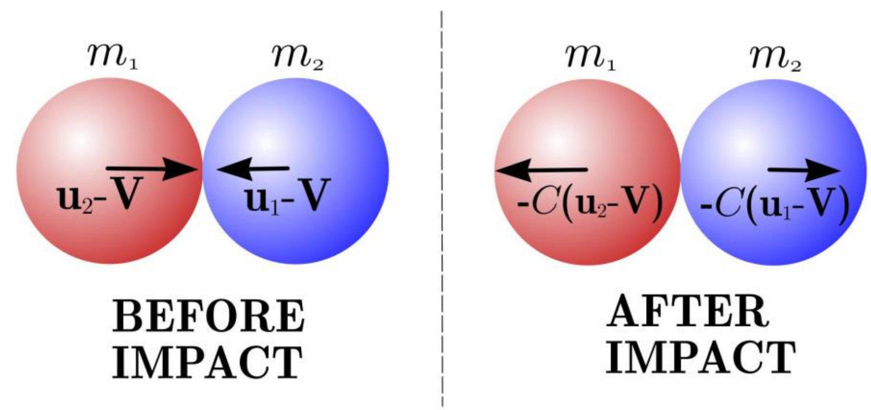


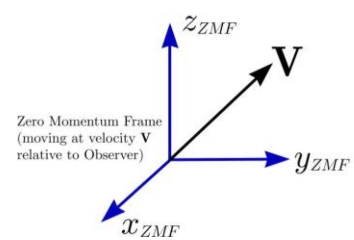
Ball bounce simulation





Zero Momentum Frame



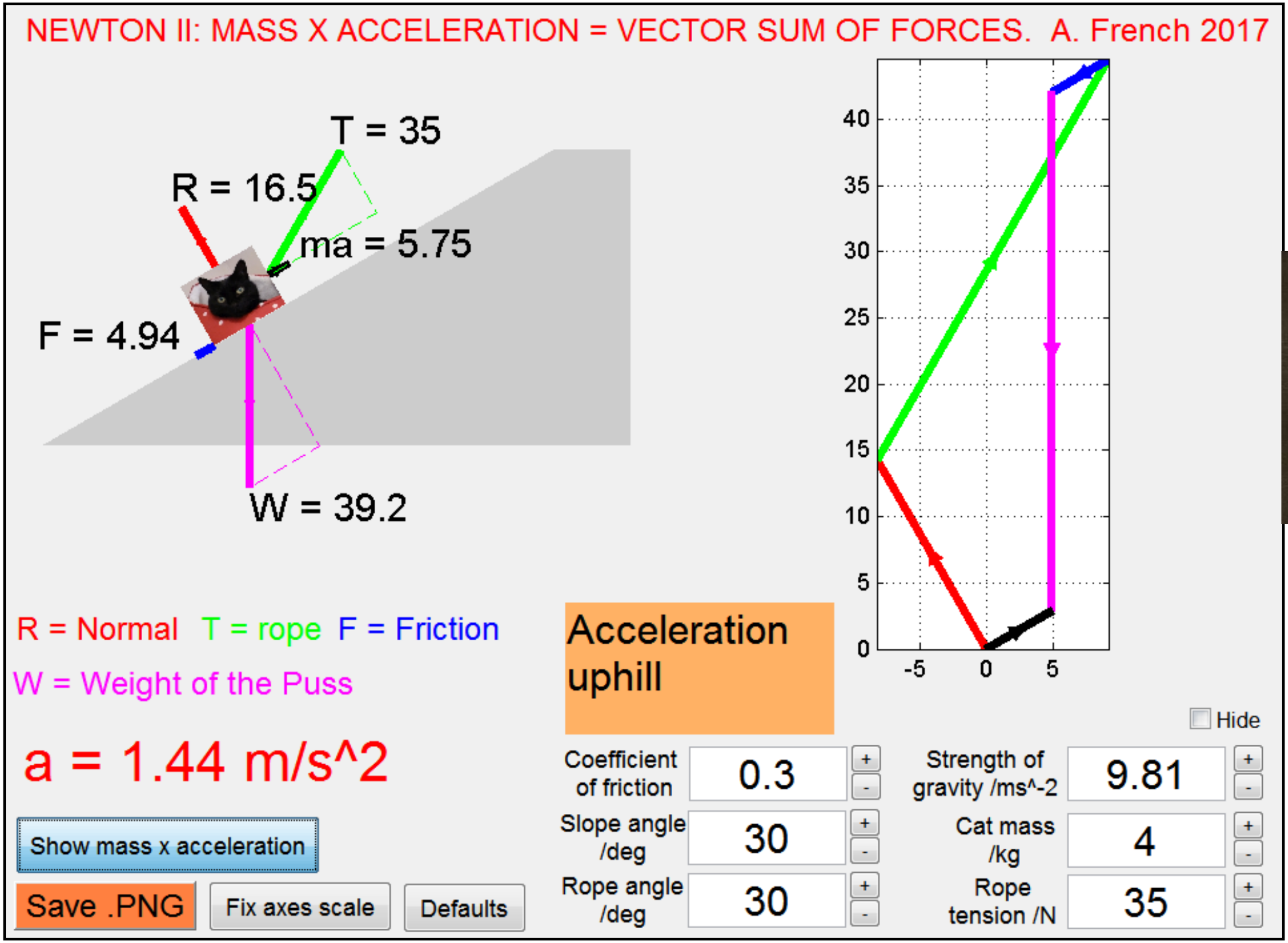


$$\boxed{\mathbf{V} = \frac{m_1\mathbf{u}_1 + m_2\mathbf{u}_2}{m_1 + m_2}}$$

C is the coefficient of restitution

#6: MASS x ACCELERATION = VECTOR SUM OF FORCES

Newton's second law



Isaac Newton
(1643-1727)

```
m_r = 57.5/1000;    %Dry mass of rocket /kg
m_w = 0.3;          %Initial water mass /kg
r = 0.05;           %Radius of bottle end /m
cD = 0.4;           %Drag coefficient
rho_air = 1.225;    %Air density /kgm^-3
g = 9.81;           %Strength of gravity (N/kg)
T = 9/30;           %Water ejection time /s
C = 60;             %Water ejection speed relative to rocket (m/s)
mu = m_w/T;         %Average mass ejection rate (kg/s)
m = m_r + m_w;      %Initial mass of rocket plus water /kg
A = pi*(r^2);       %Cross sectional area of nosecone (m^2)
dt = 0.0001;        %Timestep /s

%Initialize height x (m), time t (s), velocity v (m/s) and acceleration
%(m/s^2) vectors.
x = 0; t = 0; v = 0; a = 0;  hits_ground = 0;  thrust_phase = 1;  n=1;

%Determine trajectory
while  hits_ground == 0

  %Determine acceleration by Newton's Second Law weight(n) = m(n)*g;
  drag(n) = 0.5*cD*rho_air * A * v(n) * abs( v(n) );
  if thrust_phase == 1;
    thrust(n) = mu*C;
    m(n+1) = m(n) - mu*dt;  %Reduce mass of rocket
  else
    thrust(n) = 0;  %No more thrust - all water ejected
    m(n+1) = m(n);  %Mass of rocket remains the same
  end
  a(n) = ( thrust(n) - weight(n) - drag(n) )/m(n);

  %Determine new speeds and distance via 'constant acceleration within a
  %time step' (Verlet method)
  v(n+1) = v(n) + a(n)*dt; x(n+1) = x(n) + v(n)*dt + 0.5*a(n)*dt^2;
  t(n+1) = t(n) + dt;

  %Check if thrust phase is over
  if mu*t(n) > m_w; thrust_phase = 0; end

  %Check if rocket has landed
  if ( v(n+1) < 0 ) && ( x(n+1) < 0 ); hits_ground = 1; end

  %Increment step counter
  n = n + 1;
end
t(end) = []; x(end) = []; v(end) = []; m(end) = [];  %Remove last values to make variable arrays the same length

%Plot x vs t, v vs t and a vs t
subplot(2,2,1); plot(t,x); xlabel('t/s'); ylabel('altitude x (m)');
title(['ROCKET MODEL: ToF=',num2str(t(end)),'s, xmax=',num2str( max(x) ),'m']); axis tight; grid on;
subplot(2,2,2); plot(t,v); xlabel('t/s'); ylabel('velocity v (m/s)');
title(['vmax=',num2str( max(v) ),'m/s']); axis tight; grid on;
subplot(2,2,3); plot(t,a); xlabel('t/s'); ylabel('acceleration a (m/s^2)');
title(['amax=',num2str( max(a) ),'m/s^2']); axis tight; grid on; axis tight; grid on;
subplot(2,2,4); plot(t,thrust,t,weight,t,drag); xlabel('t/s'); ylabel('Forces /N');
title(['max thrust=',num2str( max(thrust) ),'N']); axis tight; grid on; axis tight; grid on; axis tight; grid on;
legend({'thrust','weight','drag'}); print(gcf,'rocket_model.png','-dpng','-r300');
```

$$m\frac{dv}{dt} = \underbrace{T}_{\text{Thrust}} - \underbrace{mg}_{\text{Weight}} - \underbrace{\tfrac{1}{2}c_D A v|v|}_{\text{Drag}}$$

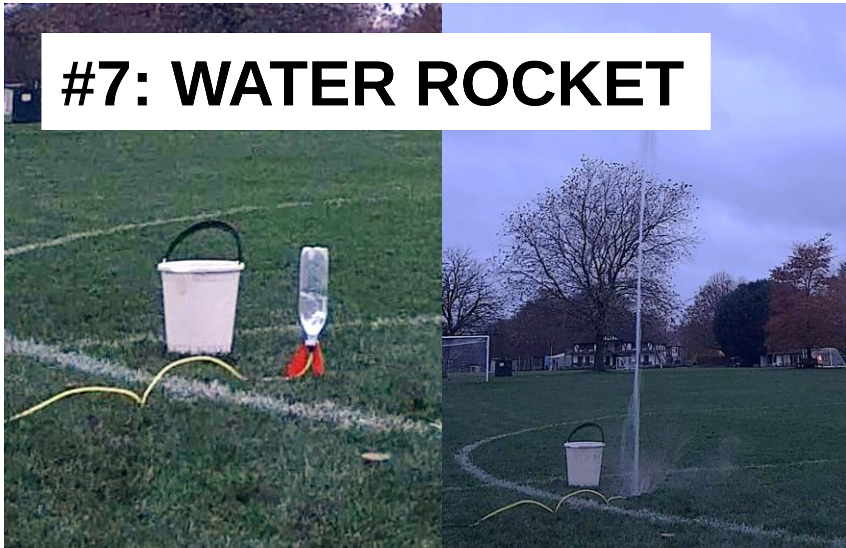
Newton II

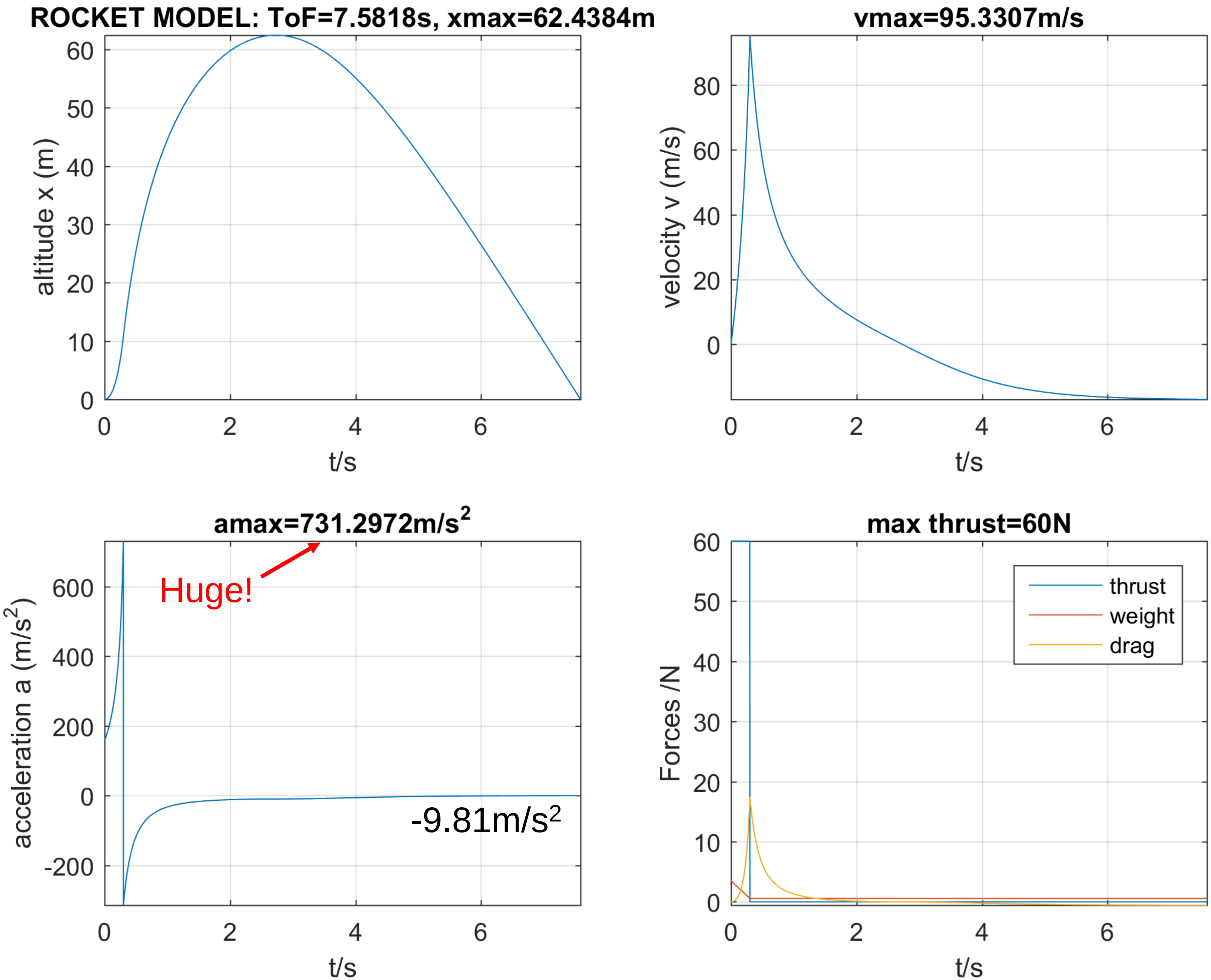
Thrust

$$T = \begin{cases} \mu C & 0 < t < \dfrac{m_w}{\mu} \\ 0 & \dfrac{m_w}{\mu} < t < t_{impact} \end{cases}$$

Mass

$$m = \begin{cases} m_w + m_r - \mu t & 0 < t < \dfrac{m_w}{\mu} \\ m_r & \dfrac{m_w}{\mu} < t < t_{impact} \end{cases}$$




ROCKET MODEL: ToF=7.5818s, xmax=62.4384m
altitude x (m)
t/s
vmax=95.3307m/s
velocity v (m/s)
t/s
amax=731.2972m/s^2
Huge!
acceleration a (m/s^2)
-9.81m/s^2
t/s
max thrust=60N
thrust
weight
drag
Forces /N
t/s

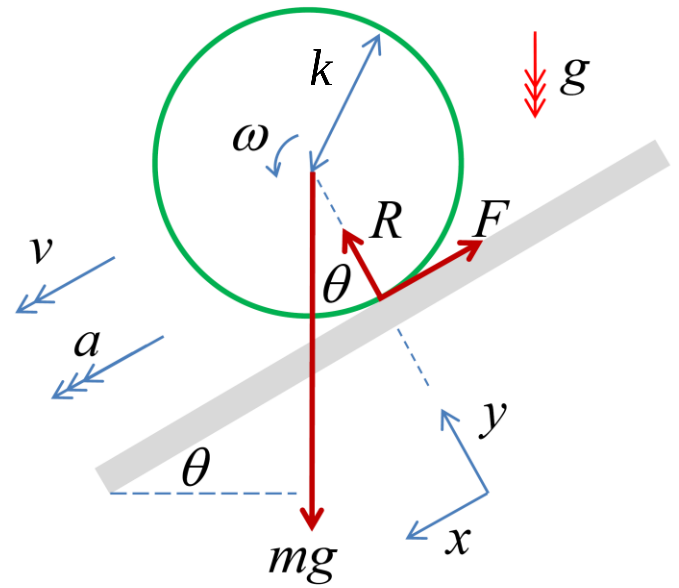
#8: ROLL THE LOOP

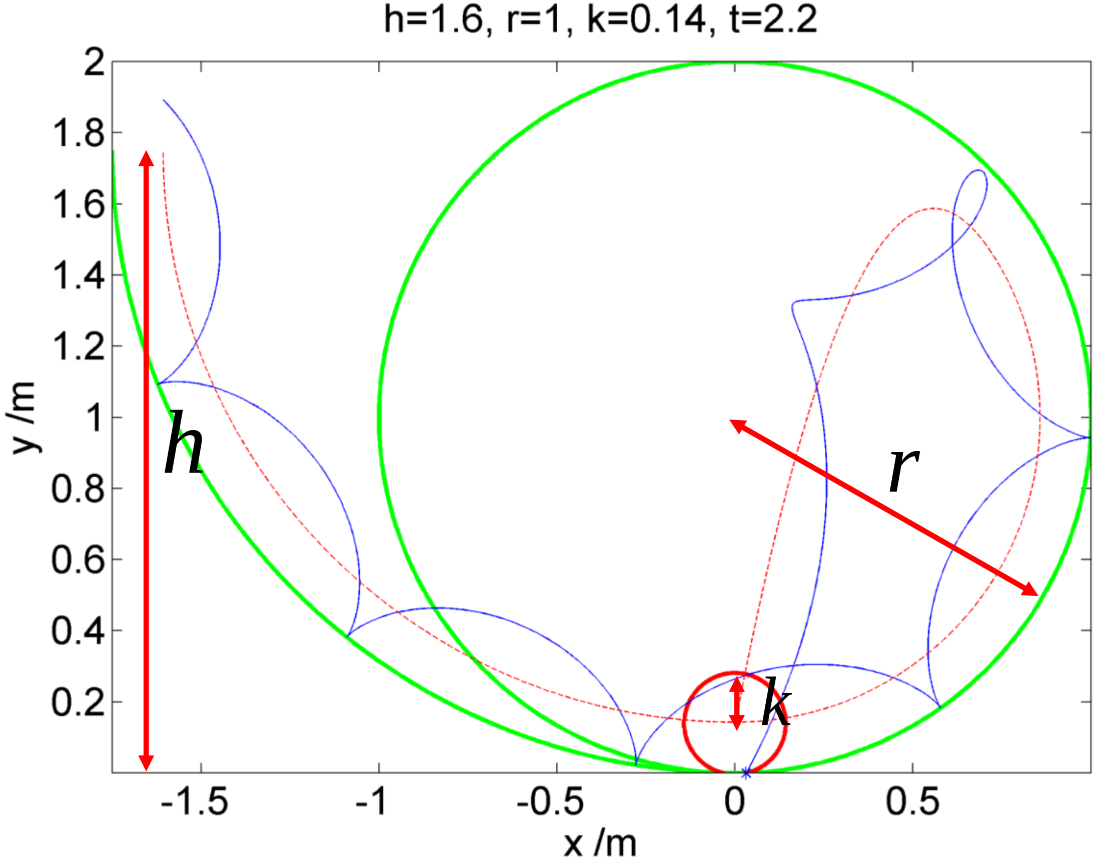
$$\boxed{h > \tfrac{1}{2}(5+\rho\sigma)(r-k)}$$

All the way round without falling off

$\rho = 1$ rolling $\qquad \sigma = 1/2$ cylinder

$\rho = 0$ sliding $\qquad \sigma = 2/5$ sphere





To fall off and land at (0,0)

$$\boxed{h > \tfrac{1}{4}(7+\rho\sigma)(r-k)}$$