Free-Fall in 10 different Languages

C4dynamics

BACKGROUND

$$\dot{y} = g \cdot t$$

y: height

$\dot{y}$: falling rate

g: gravity acceleration, about 9.8

t: time

SOLUTION

$$y = y_0$$
$$y = y + g \cdot t \cdot dt$$

- A solution in the form of a numerical integration
- Not exact as analytical solution, but more flexible
- Let's go with it…

give me the gift!

1. PYTHON

```
while t <= t_max:
        y += g * t * dt
        t  += dt
```

(simple Euler integration)

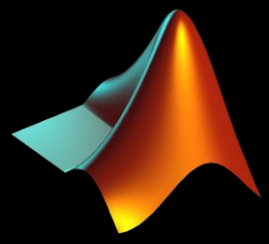
gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

2. MATLAB



```
while t <= t_max
        y = y + g * t * dt;
        t  = t + dt;
end
```

(dt is just the time step)

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

3. C

```
for (int i = 1; t < t_max; i++) {
        y += g * t * dt;
        t  += dt;
}
```

(remember that y is the height above ground)

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

4. C++

```
for (int i = 1; t < t_max; i++) {
        y += g * t * dt;
        t  += dt;
}
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

5. JAVA


```
for (int i = 1; t < t_max; i++)  {
        y += g * t * dt;
        t  += dt;
}
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

6. C#

```
for (int i = 1; i <= t_max / dt; i++) {
        y += g * t * dt;
        t  += dt;
}
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

7. JAVASCRIPT


```
for (let i = 1; t < t_max; i++)  {
        y += g * t * dt;
        t  += dt;
}
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

8. R

```
for (i in 1 : round(t_max / dt)) {
        y <- y + g * t  * dt
        t  <- t + dt
}
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

9. JULIA


```
while t <= t_max
        y += g * t * dt
        t  += dt
end
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

10. FORTRAN

```
do i = 1, int(t_max / dt)
        y = y + g * t * dt
        t  = t + dt
end do
```

gravity (acceleration, 9.8)

init with:
y = 1000, t = 0, dt = 0.01, t_max = 15
g = -9.8

give me the gift!

C4dynamics

SPECIAL BONUS!

An artistic python code doing a complete free fall model in 11 simple lines:

```
import C4dynamics as c4d

h0 = 1000
dt = 0.001
t  = 0
g  = -9.8

ball = c4d.datapoint(z = h0)

while ball.z >= 0:
    ball.run(dt, np.array([0, 0, g] * ball.m))
    ball.store(t)
    t += dt

ball.draw('z')

#
# for developers:
#   1) try to change the initial altitude, h0, and re-run the program.
#   2) draw also the trajectory in x axis: ball.draw('x')
#       what do you see? why there is no motion in x?
#   3) now, set initial velocity in x, change line 7 to: ball = c4d.datapoint(vx = 100,
#          z = -h0)
#       run again and draw the x trajectory.
#       what do you see now?
#       what does it do to the motion in z axis? why?
#   4) import C4dynamics as c4d in your project and use the object c4d.datapoint to
#          model your physics.
#   FAQ? contact zivmeri @ linkedin \ gmail, or C4dynamics at github.
##
```

C4dynamics

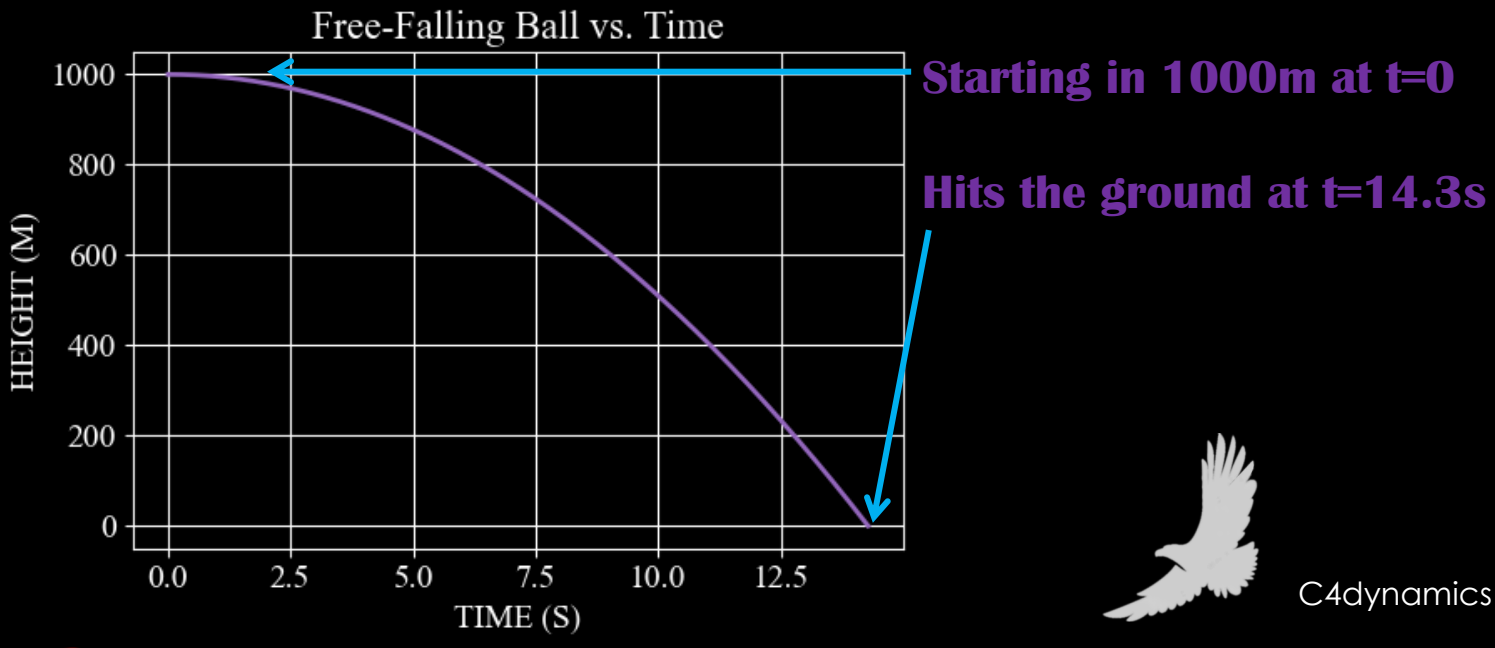
SPECIAL BONUS!

Want to make fun while developing cool algorithms for physical systems too?

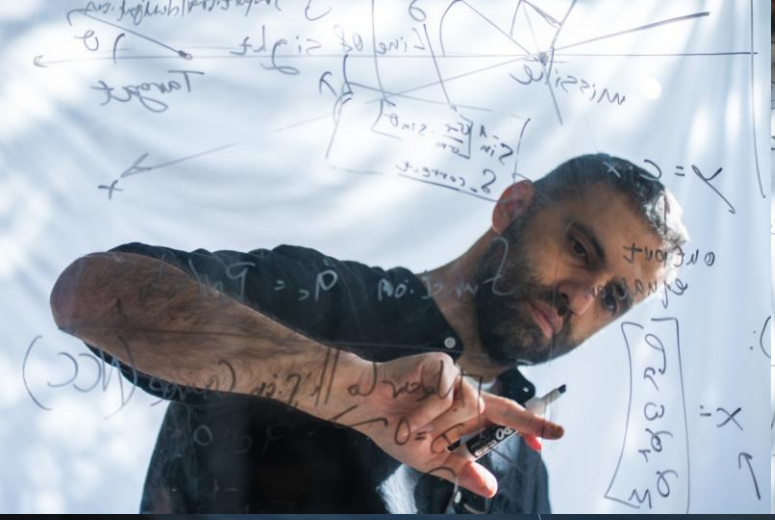
Download *now* C4dynamics and run freefall.py.
Follow the instructions there.

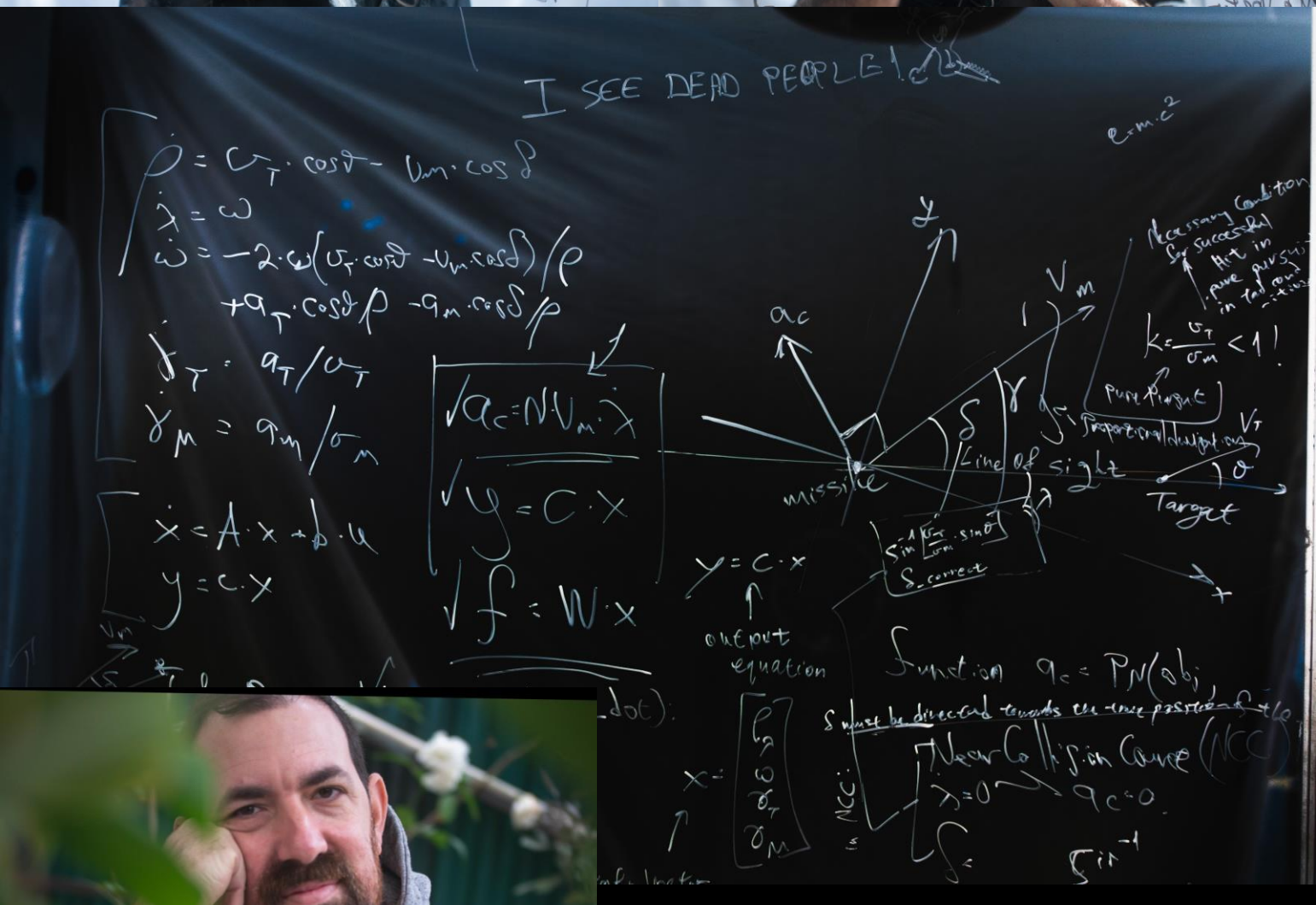
https://github.com/C4dynamics/C4dynamics/blob/main/examples/freefall.py

C4dynamics is a cutting-edge, high-standard algorithms development framework.



C4dynamics





Gavriel Weinberger

Visual Content Creator

C4dynamics