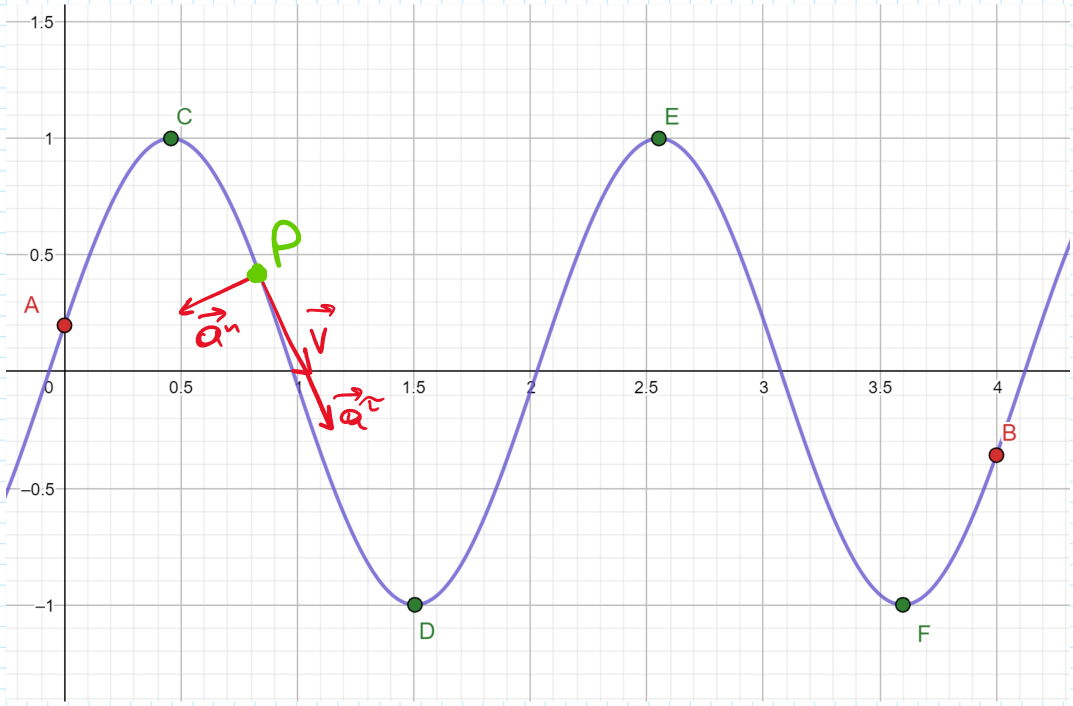
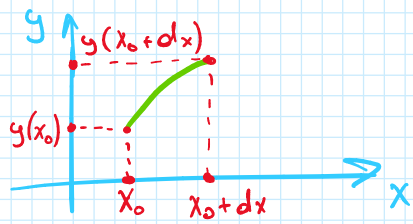
TM Big HW Anton Buguev B19-RO-01



For the solution let us use the following algorithm:

1) Divide the plot into a lot of sectors with small step along x axis, let us make it $\boxed{dx = 0{,}0001}$

2) So we will have the following graph:



Since dx is very small, we can calculate the length of this graph not using formula $\ell = \sqrt{1+\left(\frac{d}{dx}y(x)\right)^2}\,dx$ but using Eulidean distance because we can assume that this is a straigt line:

$$\boxed{\ell = \sqrt{(x_0-(x_0+dx))^2+(y(x)-y(x_0+dx))^2}}$$

3) Knowing the length of that piece, let us use formula $\boxed{v=\sqrt{2aS+v_0^2}}$. Using this formula we can find velocity and acceleration for each segment

Since we want our body to move as fast as possible, we set $a=a^{\tau}_{max}$ and v_0 is calculated for previous segment, so initially it is 0.

If calculated $v \le v_{max}$, we need to check that $a^n \le a^n_{max}$

4) $\boxed{a^n=\frac{v^2}{\rho}}$ where $\boxed{\rho = \frac{\sqrt{\left[1+\left(\frac{dy}{dx}\right)^2\right]^3}}{\frac{d^2y}{dx^2}}}$

If both conditions hold we save calculated v as velocity for this sector. But if at least one of the conditions fail, we try to decrease a^{τ} by small value and calculate everything again until we find appropriate values for v, a^{τ} and a^n.

However, it is may be that it is impossible to find appropriate values. In this case we need to go to the previous segment decrease values of v and a^{τ} and recalculate everything again.

5) Since we have a condition that in the end body must stop, we can consider last segments

as motion back, for example, from $x=4$ to $x=3,5$.

Thus we divided our model into 2 parts:

1 - Motion on $x \in [0; 3,5)$ with $v_0 = 0$ for the smallest time

2 - Motion on $4 \geq x \geq 3,5$ with $v_0 = 0$ for the smallest ime

6) Pseudocode:

1. create lists for required data
2. $dx = 0,0001$
   $da = 0,1$
3. $i = 0$

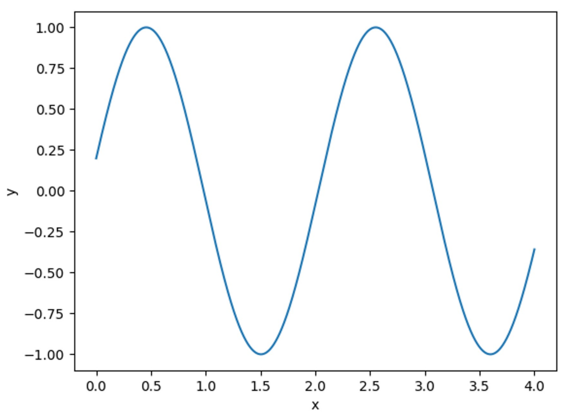
```
while(i < 3,5·dx):
    find ℓ       // Eulidean dist.
    V₀ = V_data[i]
    v = √(2ℓa^τ + v₀²)
    if (v ≤ v_max and a^n ≤ a^n_max):
        if (a^τ < -a^τ_max):
            i--
        else:
            a^τ -= da
        continue
    else:
        save v, a^τ, a^n
```

```
        save v, a^τ, a^n
    i++
4. j = 4 / dx
   while(j ≥ 3,5·dx):
    find l      // Eulidean dist.
    V0 = V_data[j]
    V = sqrt(2 l a^τ + V0^2)
    if(v ≤ v_max and a^n ≤ a^n_max):
        if(a^τ < -a^τ_max):
            j+1
        else:
            a^τ -= da
        continue
    else:
        save v, a^τ, a^n
    j--
5. t=0
   for i in len(dist_data):
      t += dist_data[i] / vel_data[i]
```

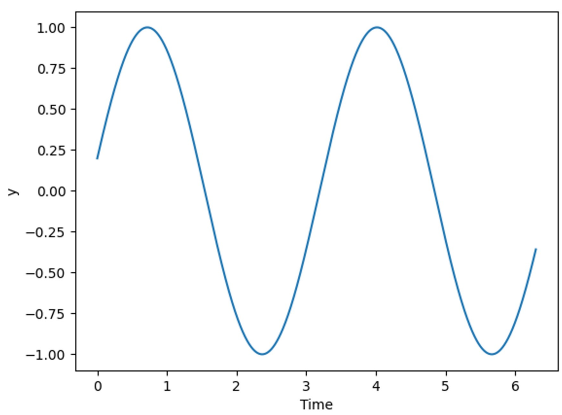
7) The best time is approx. $\boxed{t = 6{,}3\ s}$

8) Plots

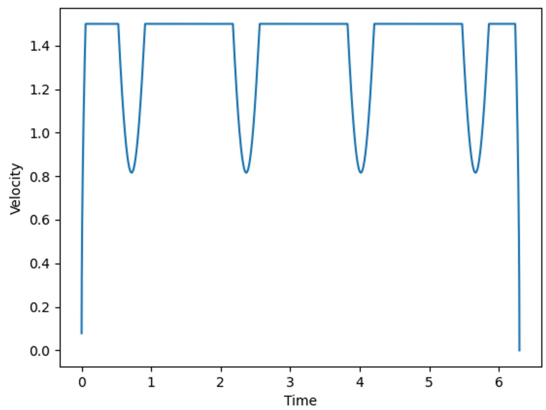
$y(x)$:



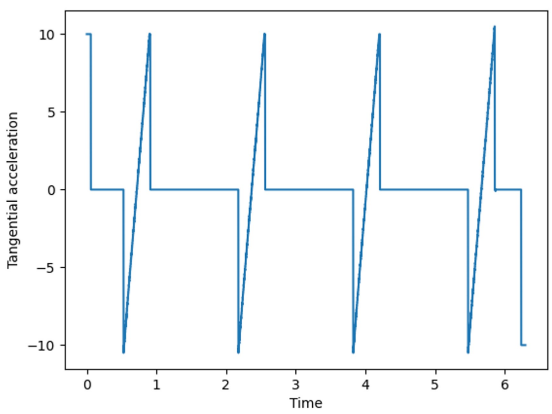
$y(t)$:



$v(t)$:



$a^{\tau}(t)$:



$a^{n}(t)$: