Fast & Furious

Misha DOPRE | Alexis DUPONT | Adem BRAHIM | Julian COTTALORDA

Mécanique

Livrable 3

Table des matières :

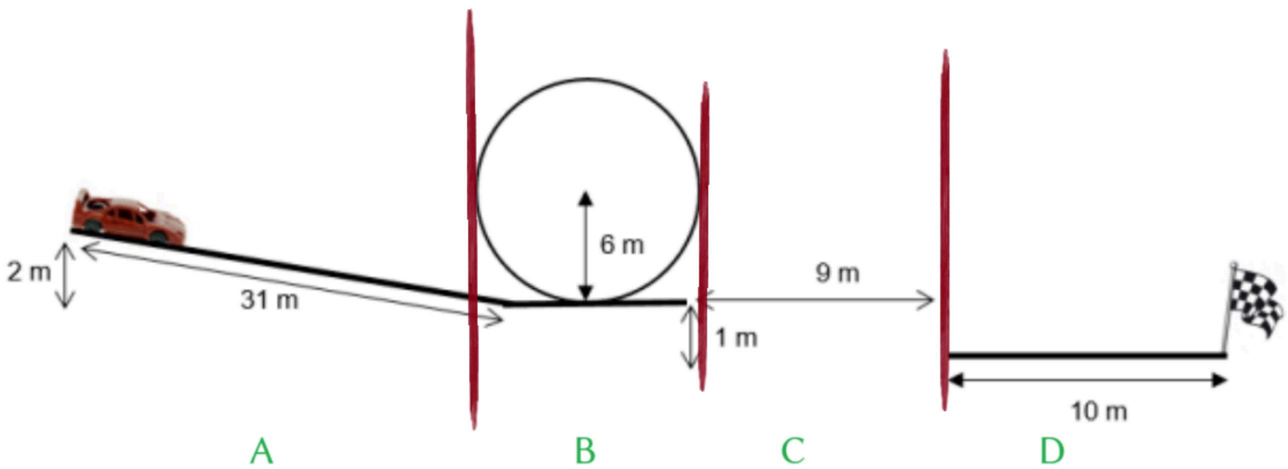
I) Contexte

- **Général :** Dom Toretto est défié par Owen Shaw dans une course comprenant plusieurs défis techniques : une piste d'élan (2 m de hauteur, 31 m de longueur), un looping (6 m de rayon), un saut au-dessus d'un ravin (9 m de large avec 1 m de dénivelé négatif), et une piste horizontale finale de 10 m. L'objectif est de compléter la course en moins de 8 secondes, sans endommager la voiture.

  Tej Parker, chargé de sélectionner le véhicule idéal dans la collection de Dom, doit réaliser une étude théorique des performances nécessaires pour chaque segment du circuit. Cela inclut les calculs de vitesse minimale pour réussir chaque étape, l'évaluation des contraintes mécaniques, et la simulation des performances des différents modèles. Une attention particulière sera portée à la Dodge préférée de Dom, avec la possibilité d'y apporter des modifications si nécessaire.

- **Dans ce <u>Livrable 3</u> :** Notre équipe se concentrera sur les repères et référentiels utilisés, les différentes équations du modèle des 4 parties du circuit (la pente, le looping, le ravin et la ligne droite), le Python, les vitesses minimales et les incertitudes.

II) Le découpage du circuit

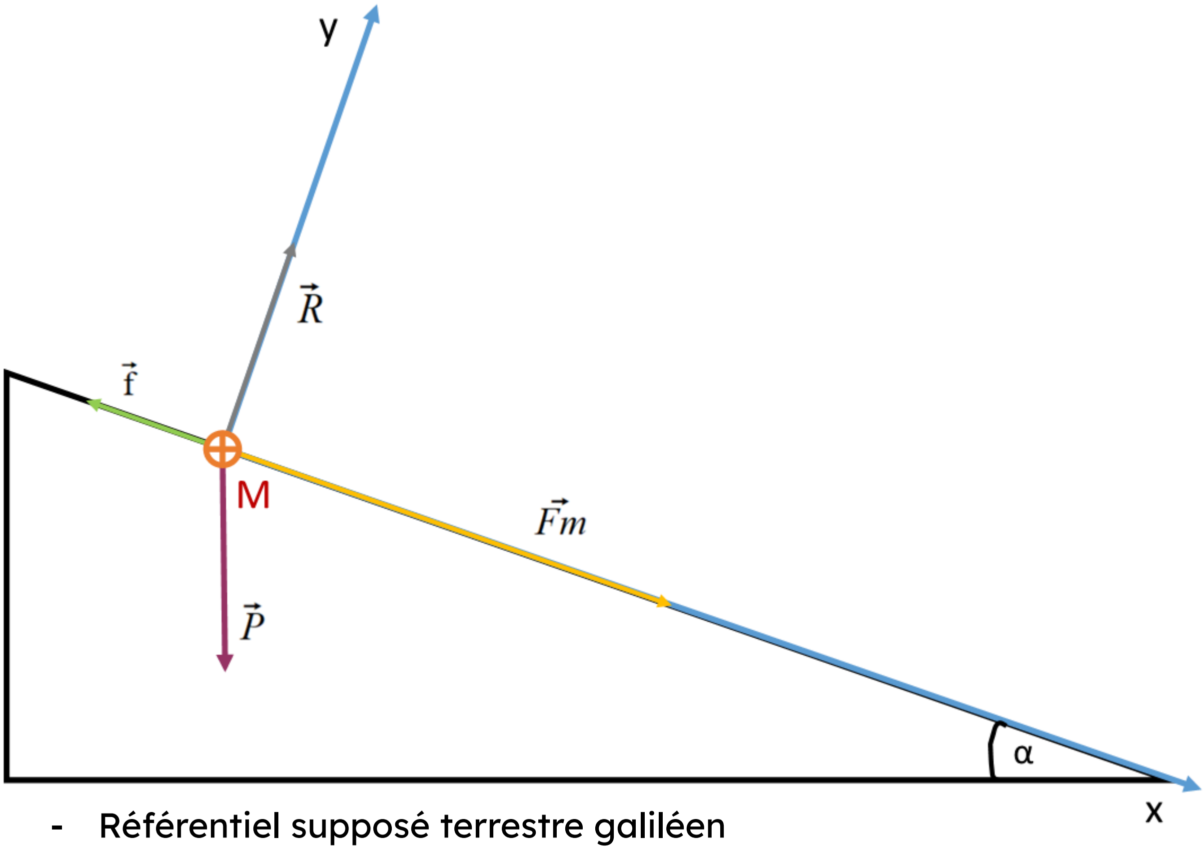


Le circuit est alors découpé en quatre parties :

- A) : La pente
- B) : Le looping
- C) : Le saut
- D) : La piste d'arrivée

III) Les calculs des étapes

A) La pente



- Référentiel supposé terrestre galiléen
- Repère cartésien (x; y)

Légendes :

- M : Centre de gravité de la voiture
- x : axe des abscisses
- y : axe des ordonnées
- α : angle de la pente

Vecteurs :

- Vecteur f : f = Fa + Fr (f = frottements aérodynamiques + frottements des roulements)
- Vecteur $\vec{P}$: P = m*g (poids = masse * cste de gravitation)
- Vecteur $\vec{R}$: Résistance normale
- Vecteur $\vec{Fm}$: Fm (Force motrice)

On prend en compte les frottements.

On néglige la portance dans la pente.
Fa : ½**Cx**p*S*v^2
S : Surface avant du véhicule
p : densité de l'air

Somme des forces :

- P(P sin (α); - P cos (α))
- Fa (- Fa; 0)
- Fr (- Fr; 0)
- Fm (Fm; 0)
- R (0; R)

2ème loi de Newton

$\Sigma\vec{F_{ext}} = m.\vec{a}$
$\vec{P} + \vec{Fa} + \vec{Fr} + \vec{Fm} + \vec{R} = m.\vec{a}$

Axe Ox

ma = Psin(α) - Fa - Fr + Fm

a = Psin(α) - Fa - Fr + Fm / m

Axe Oy

0 = -Pcos(α) + R

Calcul du vecteur accélération

$$\vec{a}\begin{cases} ax(t) = \dfrac{P*sin(\alpha)-Fa-Fr+Fm}{m} \\ ay(t) = 0 \end{cases}$$

$$\Leftrightarrow \quad \vec{a}\begin{cases} ax(t) = 0.5* \dfrac{m*g*sin(\alpha)-1/2*1.225*Cx*S*v^2-\mu*m*g*cos(\alpha)+Fm}{m} \\ ay(t) = 0 \end{cases}$$

On simplifie les m

$$\Leftrightarrow \quad \vec{a}\begin{cases} ax(t) = (\dfrac{-1/2*1.225*Cx*S*v^2+Fm}{m}) + g * sin(\alpha) - \mu * g * cos(\alpha) \\ ay(t) = 0 \end{cases}$$

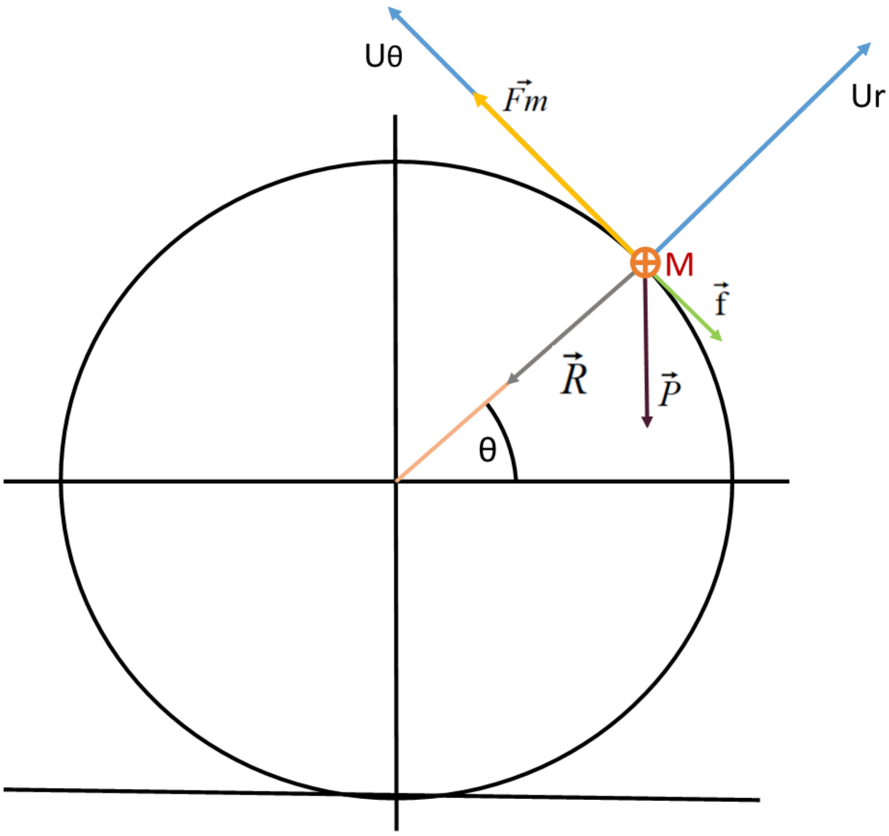
Calcul du vecteur vitesse

$$\vec{v}\begin{cases} vx(t) = [(\dfrac{-1/2*1.225*Cx*S*v^2+Fm}{m}) + g * sin(\alpha)-\mu * g * cos(\alpha)]*t+v0*cos(\alpha) \\ vy(t) = 0 \end{cases}$$

Calcul du vecteur position

$$\vec{OM}\begin{cases} x(t) = ½*((\frac{-1/2*1.225*Cx*S*v^2+Fm}{m})+g*\sin(\alpha)-\mu*g*\cos(\alpha))*t^2+(v0*\cos(\alpha))*t+x0 \\ y(t) = 0 \end{cases}$$

B) Le looping



- Référentiel supposé terrestre galiléen
- Repère polaire

Légendes :

- M : Centre de gravité de la voiture
- Ur : axe radial au looping

- $U\theta$: axe tangent au looping
- θ : angle par rapport à l'axe de référence

Vecteurs :

- Vecteur $\vec{f}$: f = Fa + Fr (f = frottements aérodynamiques + frottements des roulements)
- Vecteur $\vec{P}$: P = m*g (poids = masse * cste de gravitation)
- Vecteur $\vec{R}$: Résistance normale
- Vecteur $\vec{Fm}$: Fm (Force motrice)

⚠ **Les frottements :**

On suppose que les frottements sur une même distance à une altitude de départ et d'arrivée similaires sont égaux.

P = 2*π*r
P = 2*π*r
P ≃ 37.7 m

Donc calculer les frottements du looping revient à calculer les frottements d'une ligne droite de 37.7 m car l'énergie mécanique se conserve (par des variations d'énergie cinétique et potentielle).

On cherche l'accélération en coordonnées polaires

$Ur = i\cos(\theta) + j\sin(\theta)$

$U\theta = -i\sin(\theta) + j\cos(\theta)$

Vitesse : $R * (-i*\dot{\theta}*\sin(\theta)+j*\dot{\theta}+\cos(\theta)) = \vec{v}$

$\vec{v} = r*\dot{\theta}*(-i*\sin(\theta)+j*\cos(\theta))$
$\vec{v} = r*\dot{\theta}*U\theta$
$\vec{v} = r*\dot{\theta}*(-i\sin(\theta) + j\cos(\theta))$

On dérive donc :

$\vec{a}$ = r*($\theta^{\bullet\bullet}$(-i*sin(θ)+j*cos(θ))+$\theta^{\bullet}$*(-i*$\theta^{\bullet}$*cos(θ)-j*$\theta^{\bullet}$*sin(θ)))
$\vec{a}$ = r*($\theta^{\bullet\bullet}$*Uθ+$\theta^{\bullet 2}$*(-Ur))
$\vec{a}$ = r*$\theta^{\bullet\bullet}$*Uθ-r*$\theta^{\bullet 2}$*Ur

Passons au PFD :

P(-Pcos(θ);-Psin(θ));
R(-R; 0);
f(0; -f);
Fm(0; Fm);

Pour Ur : $\Sigma \vec{Fext}$ **=** m*$\vec{a}$

donc -P*cos(θ)-R = m*$\vec{a}$

m*(-r*$\theta^{\bullet 2}$) = P*cos(θ)-R

Pour Uθ : $\Sigma \vec{Fext}$ = m*$\vec{a}$

m*$\vec{a}$ = -P*sin(θ)-F

m*r*$\theta^{\bullet\bullet}$ = -P*sin(θ)-F

Accélération :

$$m\vec{a}\begin{cases} \mathbf{m}\ \vec{a}\ Ur = m*(-r*\theta^{\bullet 2}) = -P*\cos(\theta)-R \\ \mathbf{m}\ \vec{a}\ U\theta = m*r*\theta^{\bullet\bullet} = -P*\sin(\theta)-\mu*R+Fm \end{cases}$$

Donc R = - P*cos(θ) -m*r*$\theta^{\bullet 2}$
Ainsi, R = m (-g*cos(θ) - r *$\theta^{\bullet 2}$)

$$m*r*\ddot{\theta} = -m*g*\sin(\theta)-\mu*R+Fm$$

$$m*r*\ddot{\theta} = -m*g*\sin(\theta)-\mu*m(-g*\cos(\theta)-r*\theta^2)+Fm$$

$$r\ddot{\theta} = -g*\sin(\theta)-\mu*(-g*\cos(\theta)-r*\theta^2)+\frac{Fm}{m}$$

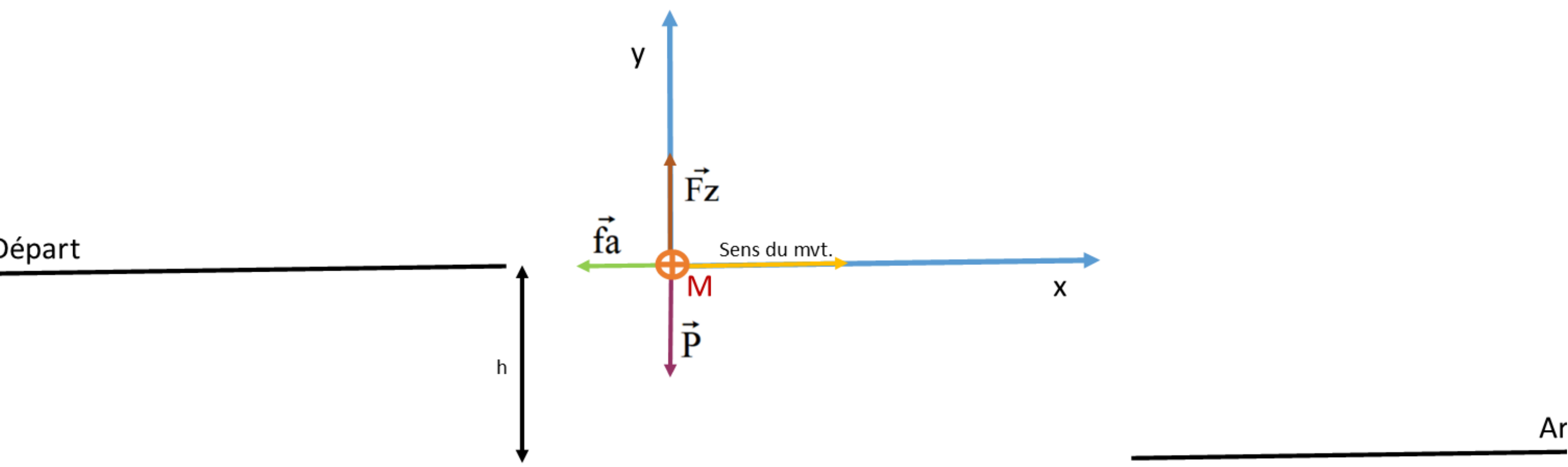
$$\ddot{\theta} = \frac{-g*\sin(\theta)}{r} + \frac{\mu*g*\cos(\theta)}{r} + \mu\dot{\theta}^{2} + \frac{Fm}{rm}$$

Sachant que :

- $\ddot{\theta}$ = accélération polaire
- $\dot{\theta}$ = vitesse polaire
- θ = coordonnée polaire

Donc pour $\dot{\theta}$: On le fera à l'aide de python (livrable 3), grâce à la méthode d'Euler.

C) Le saut



- Référentiel supposé terrestre galiléen
- Repère cartésien

Légendes :

- M : Centre de gravité de la voiture
- x : axe des abscisses
- y : axe des ordonnées
- Sens du mouvement (même sens que la vitesse afin de déterminer la portance)

Vecteurs :
- Vecteur $\vec{Fa}$: Frottements aérodynamiques (On ne prend pas en compte les frottements des roulements)
- Vecteur $\vec{P}$: P = m*g (poids = masse * cste de gravitation)
- Vecteur $\vec{Fz}$: Force de portance

En x on ne garde que les frottements car la force motrice est nulle du fait qu'il n'y ait plus de contact avec la route il ne reste donc en abscisse que les frottements de l'air. Et en y $\Sigma \vec{Fext} = \cancel{m}*\vec{a} = \cancel{m}*g$

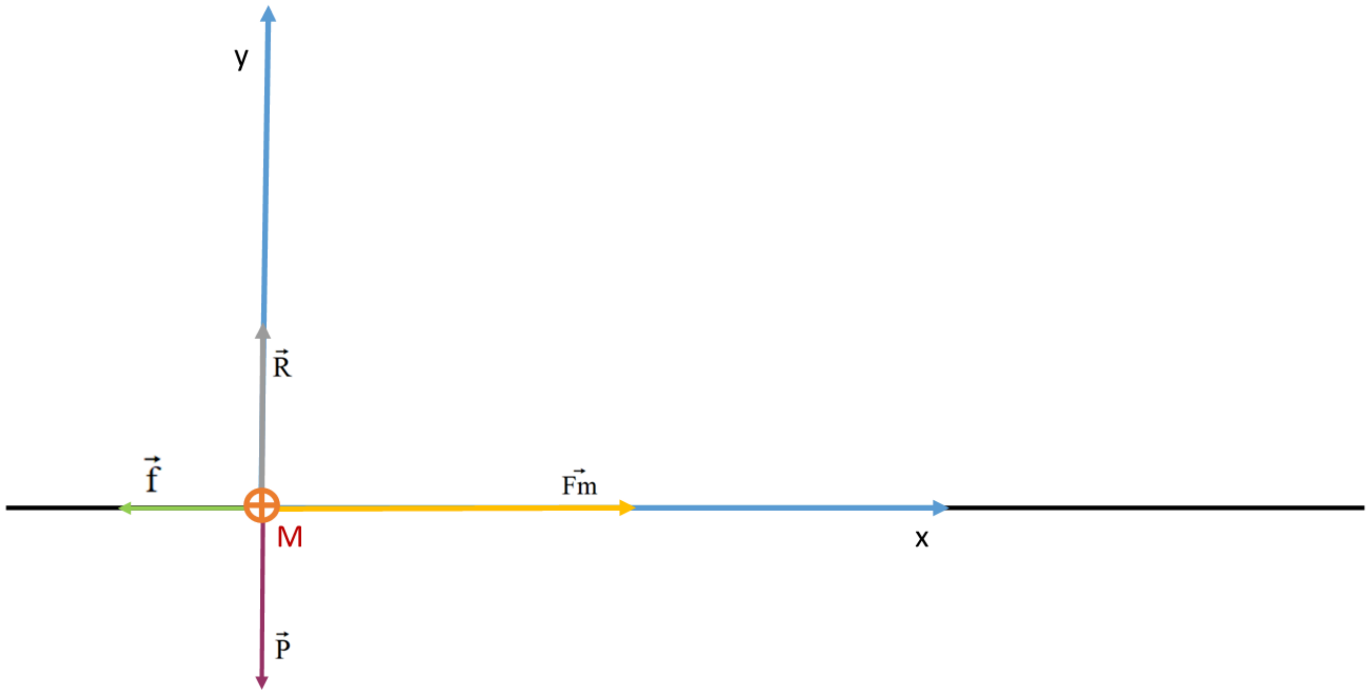
$$\vec{a} \begin{cases} ax(t) = -\frac{1}{2}*\rho\ air*Cx*Surface\ avant\ voiture*v^2 \quad = \text{traînée} \\ ay(t) = -g + \frac{1}{2}*\rho\ air*Cy*Surface\ dessous\ voiture*v^2 \quad = \text{portance} \end{cases}$$

On suppose ici que la voiture reste à plat pour simplifier les calculs de la portance sachant que la différence sera très faible et que l'on pourrait multiplier la portance par cos (angle moyen de la voiture) $\simeq$ 0.99 = 99% de la portance donc c'est négligeable

$$\vec{v} \begin{cases} vx(t) = (-1/2 *1.225*Cx*S\ avant\ voiture*v^2)*t+v0*\cos(\alpha) \\ vy(t) = (-g+\frac{1}{2}*1.225*Cy*S\ dessous\ voiture*v^2)*t+v0*\sin(\alpha) \end{cases}$$

$$\vec{OM} \begin{cases} x(t) = \frac{1}{2}*(-\frac{1}{2}*1.225*Cx*S \text{ avant voiture}*v^2)*t^2+v0*t+x0 \\ y(t) = \frac{1}{2}*(-g+\frac{1}{2}*1.225*Cz*S \text{ dessous voiture}*v^2)*t^2+y0 \end{cases}$$

D) La piste d'arrivée



- Référentiel supposé terrestre galiléen
- Repère cartésien

Légendes :

- M : Centre de gravité de la voiture
- x : axe des abscisses
- y : axe des ordonnées

Vecteurs :

- Vecteur f : f = Fa + Fr (f = frottements aérodynamiques + frottements des roulements)

- Vecteur $\vec{P}$: P = m*g (poids = masse * cste de gravitation)
- Vecteur $\vec{R}$: Résistance normale
- Vecteur $\vec{Fm}$: Fm (Force motrice)

On utilise la 2nd loi de Newton :

$\Sigma \vec{F}$ = m*$\vec{a}$ Donc $\vec{a} = \frac{\Sigma \vec{F}}{m}$

On va primitiver $\vec{a}$ (t) pour trouver $\vec{v}$ (t) puis la position OM.

Donc $\vec{a}$ (t) $\begin{cases} ax(t) = \dfrac{Fm - Fa - Fr}{m} \\ ay(t) = 0 \end{cases}$

L'accélération est nulle en y.

$\vec{a}$ (t) $\begin{cases} ax(t) = \dfrac{\vec{a}\ voiture - 1/2 *1.225*Cx*S*v^2 - \mu*m*g}{m} \\ ay(t) = 0 \end{cases}$

$\vec{v}$ (t) $\begin{cases} vx(t) = \left(\dfrac{\vec{a}\ voiture - 1/2*1.225*Cx*S*v^2 - \mu*m*g}{m}\right)*t+v0*\cos(\alpha) \\ vy(t) = 0 \end{cases}$

$\vec{OM}$ (t) $\begin{cases} x(t) = ½*\left(\dfrac{\vec{a}\ voiture - 1/2*1.225*Cx*S*v^2 - \mu*m*g}{m}\right) \\ \quad *t^2+(v0*\cos(\alpha))*t+x0 \\ y(t) = 0 \end{cases}$

IV) Le Python

On utilise donc notre code Python afin de résoudre certaines équations impossibles à effectuer de façon manuelle et également pour gagner du temps dans nos calculs. Dans notre code Python, nous retrouvons donc toutes nos équations du mouvement retranscrites.

Voir fichier Python joint pour l'intégralité du code

Exemple avec une voiture (dodge sans amélioration) du fonctionnement du code :

```
Quelles voitures choisissez-vous ? (séparées par des virgules parmi dodge, supra, camaro, rx_7, skyline, lancer) dodge
Vous avez choisi : Dodge
Sur quelle portion activer le boost (A, B, C, D ou N pour aucun) ? N
Voulez-vous monter les ailes sur le bolide ? (oui/non) non
Pas d'ailes ajoutées.
Voulez-vous monter la jupe sur le bolide ? (oui/non) non
Pas de jupe ajoutée.
```

```
Résultats pour Dodge :
Temps final : 3.6074074074074076 s
Position finale : x = 31.04415431474152 m
Vitesse finale : vx = 17.273610698076677 m/s

------------Etape-2-------------


Résultats pour Dodge :
Temps final : 6.477777777777778 s
Position finale : x = 37.733896659397395 m
Vitesse finale : vx = 19.236084608315473 m/s
```

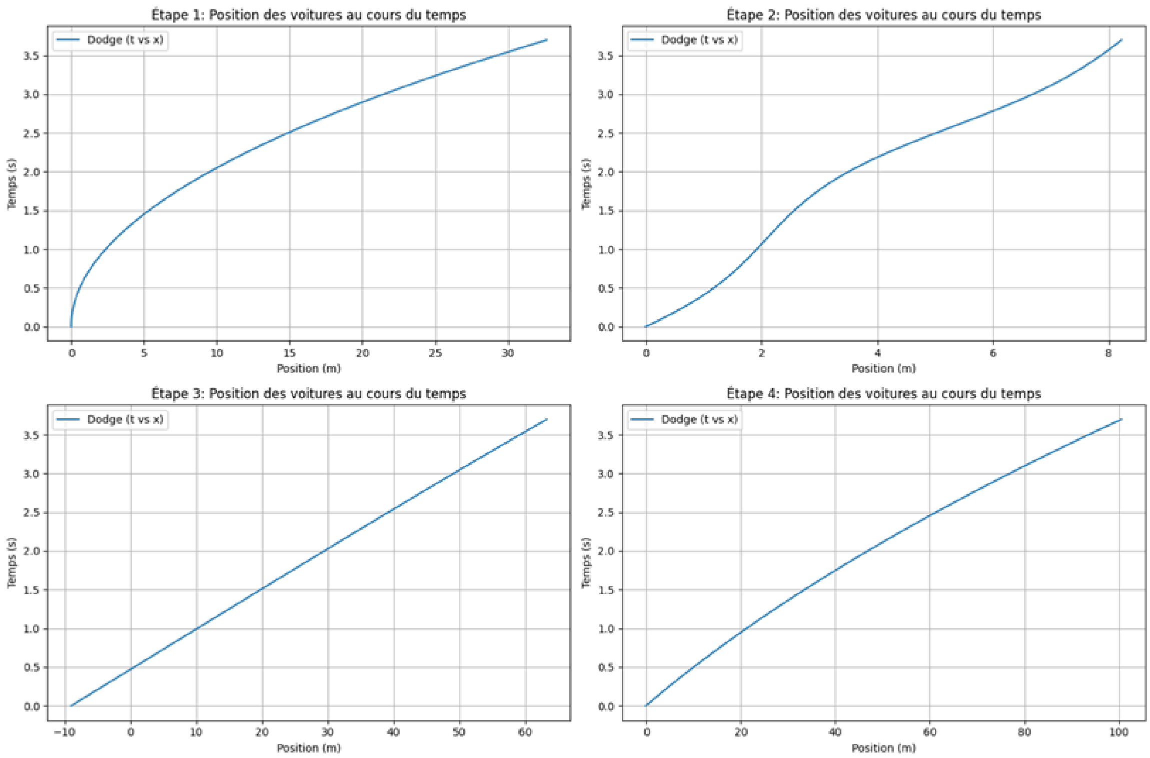
```
-------------Etape-3-------------


Résultats pour Dodge :
Temps final : 6.940740740740741 s
Position finale : x = -0.10500884564596974 m
Vitesse finale : vx = 19.19714951310107 m/s

-------------Etape-4-------------


Résultats pour Dodge :
Temps final : 7.440740740740741 s
Position finale : x = 10.02569570096348 m
Vitesse finale : vx = 21.32817401352327 m/s
```

Graphiques obtenus :



V) Les vitesses minimales

Calculer nos vitesses minimales avant le looping et avant le saut nous permet une comparaison avec nos valeurs de vitesses obtenues avec le Python. Et ainsi, l'on pourra déterminer si un modèle de voiture est capable de passer un obstacle lorsque sa vitesse est supérieure ou égale à la valeur minimum.

A) Vitesse minimale au départ du looping

On utilise le théorème de l'énergie cinétique et la somme des travaux des forces exercées sur la voiture.

$\Delta EC_{AB} = \Sigma W_{AB^{\to}}$

$\frac{1}{2} * \cancel{m} * vb^2 - \frac{1}{2} * \cancel{m} * va^2 = \cancel{m} * g(\underbrace{ha}_{0} - \underbrace{hb}_{-2R})$

$\frac{1}{2} * vb^2 - \frac{1}{2} * va^2 = -2gR$

$\cancel{1/2} * vb^2 + 2gB = \cancel{1/2} * va^2$

$va^2 = \boxed{vb^2} + 4gR$

$va^2 = gR + 4gR$

$va = \sqrt{5gR}$

$\boxed{v_{Amin} = 17.2 \text{ m/s}} = 61.92$ km/h

v_B : $\Sigma F = m * a$ donc $\vec{ac} = \frac{\Sigma F}{m}$ et $\vec{ac} = \frac{v^2}{R}$

$\frac{v^2}{R} = \frac{P + Rn}{m} \Rightarrow Rn = \frac{\cancel{m}v^2}{R} - \cancel{m}g > 0 \Rightarrow v_B = \sqrt{g * R}$

Rn > 0 : cela fonctionne
Rn = Réaction normale au support (pour que la voiture ne tombe pas)

B) Vitesse minimale au départ du saut

La voiture part avec un angle $\theta = 0°$ et une vitesse initiale que l'on cherche à déterminer pour passer le ravin.

$a_y(t) = P = -m * g$

$v_y(t) = (-m * g) * t$

$y(t) = -\frac{1}{2} * (m * g) * t^2$

et y = - 1 mètre de dénivelé = hauteur = h

$h = -\frac{1}{2} * g * t^2$

$t^2 = -\frac{2h}{g}$

$t = \sqrt{\frac{-2*(-1)}{9.81}}$

t = 0.425 secondes

Et on sait que $v_0 = \frac{d}{t}$ et d = 9 mètres

Donc $v_0 = \frac{9}{0.425}$

$\boxed{v_0 \simeq 20\, m/s}$

VI) Le choix de la voiture / Améliorations

A)Le choix de la voiture

Voitures	Looping	Ravin	
Dodge Charger R/T	1	0	0
Toyota Supra Mark IV	0	0	0
Chevrolet Yenko Camaro	1	0	0
Mazda RX-7 FD	1	0	0
Nissan Skyline GTR-R34	1	1	1
Mitsubishi Lancer Evolution VII	0	0	0

En comparant les valeurs obtenues dans notre code Python avec les valeurs minimales pour passer les obstacles, on obtient ce tableau récapitulatif. On en déduit donc que seule la **Nissan Skyline est capable de réaliser le parcours** dans l'état actuel, tandis que la **Dodge Charger ne peut pas passer le ravin**.

B) Les améliorations

Sachant que Dom Toretto préfère utiliser sa Dodge Charger même en faisant des améliorations, nous allons étudier cette possibilité.

Voici une **liste des améliorations possibles** :

- **Booster :** +30% d’accélération en moyenne sur une portion du circuit
- **Ailes :** S = 0.8 m² et m = 30 kg, permet de gagner 10% sur le coefficient de portance de la voiture
- **Jupe :** m = 15 kg, permet de diminuer de 5% le coefficient de traînée de la voiture

Améliorations	Vitesse avant le ravin (m/s)	Résultat
Rien	19,236	0
Ailes	19,241	0
Jupe	19,254	0
Booster (B)	20,876	1
Ailes + Booster (B)	20,836	1
Ailes + Jupe	19,259	0
Jupe + Booster (B)	20,846	1
Ailes + Booster (B) + Jupe	20,849	1

Il apparaît alors en effectuant nos simulations sur Python :
- La voiture ne peut pas passer les obstacles uniquement avec les ailes ou la jupe ou avec les deux.
- La voiture peut passer les obstacles avec un booster (B) *(on utilise le booster sur la section juste avant le ravin)* ainsi qu’avec un booster (B) et un ou plusieurs accessoires suivant les combinaisons.

On remarque :
- La voiture est la plus performante uniquement avec un booster (B).
- Les ailes ou la jupe ou les deux ont pour effet de ralentir la voiture quand on utilise le booster (B) : cela peut s’expliquer à cause d’une augmentation de la masse.

On en déduit :
- Il est préférable d’utiliser uniquement le booster (B) afin de réaliser notre parcours.
- Il reste important de noter que la présence des ailes en plus permettrait d’assurer une meilleure stabilité pendant le saut *(cela*

reste à vérifier en faisant des tests car cela apparaît négligeable dans nos calculs).

VII) Les résultats

Voici les meilleurs temps que l'on peut estimer permettant de réaliser notre circuit avec certaines contraintes :

- **Meilleure voiture sans améliorations :** Nissan Skyline | 6.77 secondes
- **Voiture préférée avec améliorations :** Dodge Charger | 7.06 secondes *(amélioration : booster (B))*
- **Meilleure voiture avec améliorations :** Nissan Skyline | 5.96 secondes *(amélioration : booster (A))*

Ces résultats sont donc inférieurs à 8 secondes, l'on répond donc aux conditions, il reste à Dom Toretto de choisir selon ses préférences.

VIII) Les incertitudes

Notre équipe a réalisé un fichier excel *(voir annexe)* à l'aide des valeurs mesurées afin de calculer les écarts types et donc les incertitudes.

	A	B	C	D	E	F	G	H	I
1	Mesure vitesse fin pente (m/s	Mesure vitesse fin looping (m/s)							
2	17,50	19	0,20	1,175056		Moyenne pente	17,948	N =	50
3	17,90	20,8	0,00	0,512656		Moyenne looping	20,084		
4	17,20	19,3	0,56	0,614656					
5	17,80	20,1	0,02	0,000256					
6	17,50	20,3	0,20	0,046656					
7	18,00	20,8	0,00	0,512656					
8	17,30	19,1	0,42	0,968256					
9	18,50	19,1	0,30	0,968256		Ecart Type Pente	0,5296283		
10	18,60	21	0,43	0,839056		Ecart Type looping	0,656001		
11	17,40	19,1	0,30	0,968256					
12	18,20	20,3	0,06	0,046656		Urep (V pente)	0,0749008		
13	17,30	19,1	0,42	0,968256		Urep (V looping)	0,0927725		
14	18,00	20,5	0,00	0,173056					
15	18,10	21	0,02	0,839056		Ures (Vpente)	1,4433757		
16	17,90	20,5	0,00	0,173056		Ures (V looping)	1,4433757		
17	17,20	20,5	0,56	0,173056					
18	17,30	20,4	0,42	0,099856		Uc (V pente)	1,4453178		
19	18,80	19,2	0,73	0,781456		Uc (V looping)	1,4463541		
20	17,90	20,5	0,00	0,173056					
21	18,60	19,7	0,43	0,147456					
22	17,30	19,2	0,42	0,781456					
23	17,60	20,8	0,12	0,512656					
24	18,30	20,9	0,12	0,665856					
25	17,80	20,7	0,02	0,379456					
26	17,20	20,2	0,56	0,013456					
27	17,90	19,6	0,00	0,234256					
28	17,70	19,3	0,06	0,614656					
29	18,10	19,2	0,02	0,781456					
30	19,00	19,2	1,11	0,781456					

$$Uc = \sqrt{U_{res}^{\ 2} + U_{rep}^{\ 2} + U_{lect}^{\ 2}}$$

$$U_{rep} = \frac{\sigma}{\sqrt{N}}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i}(x_i - \overline{x})^2}$$

Cela nous permet donc d'en déduire que nos calculs théoriques sont à prendre en compte avec précaution. En effet, les résultats sur circuit peuvent avoir une marge de différence, nécessitant d'adapter nos résultats avec une certaine incertitude. De plus, des essais sur une maquette apparaissent comme judicieux avant de passer sur le circuit réel.

IX) Conclusion

D'après nos calculs, la Nissan Skyline et la Dodge (avec des améliorations) sont les deux voitures permettant à Dom Toretto de gagner cette course en ayant un temps inférieur à 8 secondes. En revanche, il est important de prendre en compte une imprécision dans nos résultats.