

# Intégration de quelques aspects en physiques et mathématiques dans le projet1A

## Introduction:

La cinématique est l'étude du **mouvement** de la particule ou des systèmes de particules sans la recherche des causes de ce mouvement.

Les causes des mouvement font partie de l'étude de la dynamique des objets.

## 1) Étude Cinématique: Mouvement rectiligne Uniforme

### a) Notion de vitesse moyenne et instantanée:

A vitesse constante, la distance  $\Delta x$ , parcourue entre deux instants quelconques  $t_1$  et  $t_2$  est proportionnelle à la durée  $\Delta t = t_2 - t_1$  d'ou  $\Delta x = v \cdot \Delta t$ .

Le nombre  $v$  est par définition la vitesse du mobile.

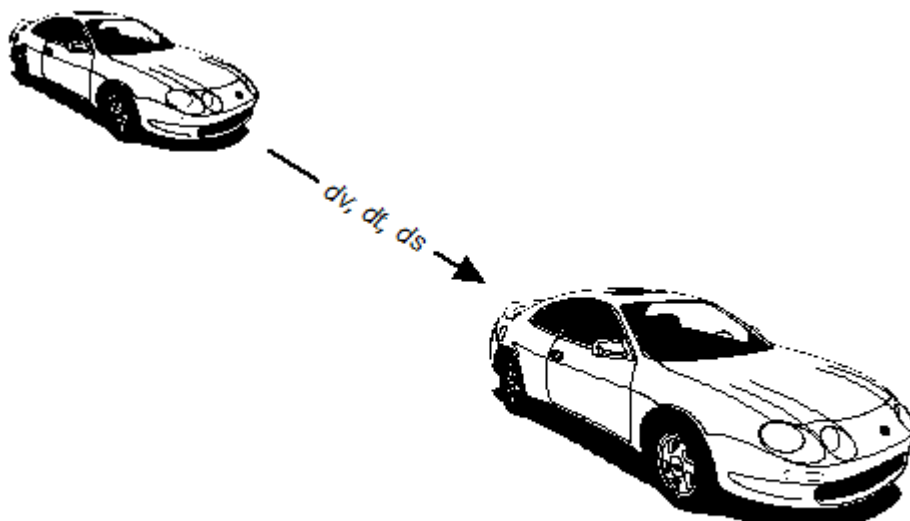
On désigne alors par  $x(t)$  l'abscisse du mobile à l'instant  $t$ ,  $x(t + \Delta t)$  son abscisse à l'instant  $t + \Delta t$  et  $\Delta x = x(t + \Delta t) - x(t)$ .

La vitesse moyenne du mobile entre les instants  $t$  et  $t + \Delta t$  est :

$$V_{\text{moy}} = \frac{\Delta x}{\Delta t} = \frac{x(t + \Delta t) - x(t)}{\Delta t}$$

On définit sa vitesse instantanée à l'instant  $t$ , notée  $v(t)$ , comme la limite de ce rapport quand  $\Delta t$  tend vers 0.

$$v(t) = \lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} = \frac{dx(t)}{dt}$$



### a) Notion d'accélération Moyenne et instantanée:

L'accélération est le taux de variation instantanée de la vitesse. Une accélération positive indique une vitesse croissante, une accélération nulle indique une vitesse constante et une accélération négative indique une vitesse décroissante. L'accélération est la dérivée de la vitesse, et donc la dérivée seconde de la distance. Elle s'exprime en  $km/s^2$

- **Accélération Moyenne:**

Si le point **M** se situe en **M1** à l'instant **t1** et qu'il possède une vitesse instantanée **v1**; s'il passe à l'instant **t2** en **M2** à la vitesse **v2**, son accélération moyenne entre **t1** et **t2** vaut :

$$a_{moy} = \frac{v_2 - v_1}{t_2 - t_1} = \frac{\Delta v}{\Delta t}$$

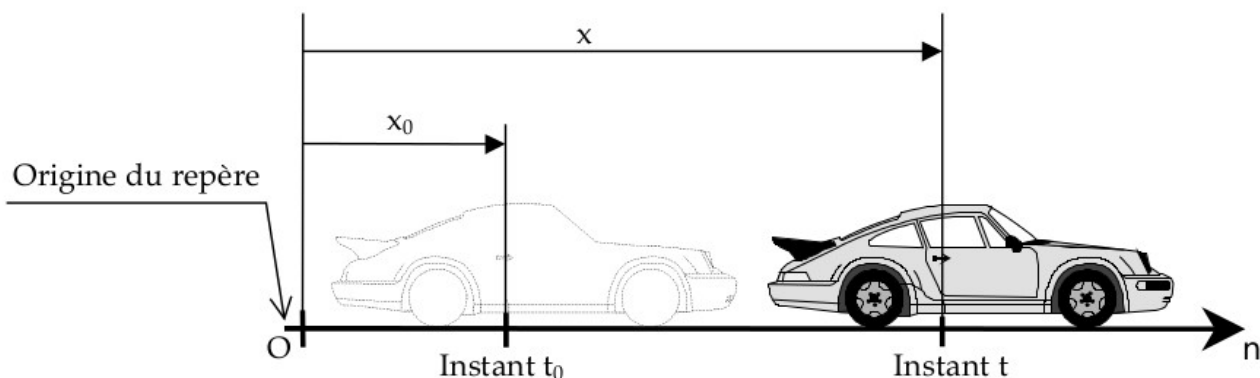
- **Accélération instantanée:**

A l'instant **t** quelconque, l'accélération instantanée correspond à la limite du rapport

$$\frac{\Delta v}{\Delta t} \text{ lorsque } \Delta t \text{ tend vers } 0 : a(t) = \lim_{\Delta t \rightarrow 0} \frac{\Delta v}{\Delta t} = \frac{dv}{dt}, \text{ or } v(t) = \frac{dx}{dt}, \text{ d'où}$$

$$a(t) = \frac{d^2 x(t)}{dt^2}$$

### b) Étude de cas: Mouvement rectiligne uniforme:



Dans ce cas de figure l'engin avance sans que le conducteur accélère ( $a=0$ ) d'où la vitesse de mouvement est constante par rapport au temps. (tout en supposant que les forces de frottements sont nulles).

## ✓ Équations du mouvement:

soient:

$t_0$ : instant initial:  $t_0=0s$

$x_0$ : le déplacement initial, à  $t=t_0$

$v_0$ : la vitesse initiale à  $t=t_0$

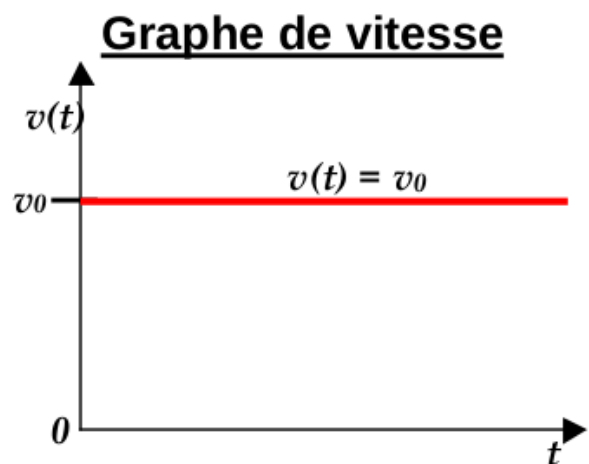
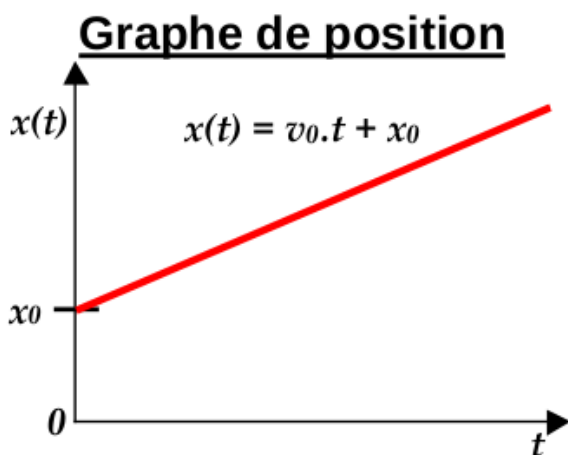
$x(t)$ : le déplacement à l'instant  $t$

$v(t)$ : la vitesse à l'instant  $t$

Les équations relatives à l'accélération, la vitesse et la position instantané d'une entité en mouvement rectiligne uniforme sont:

$$a(t)=0 \quad , \quad v(t)=v_0=\text{constante} \quad , \quad x(t)=\int v(t) \cdot dt \quad \text{d'où}$$
$$x(t)=v_0 \cdot t + x_0$$

Pour un déplacement très petit :  $dx = v_0 \cdot dt$  (1)



## Travail à Faire(Partie 1): (7pts)

1. Récupérer le code source du TP (dossier nommé TPMathPhysics/Part1/)
2. Lancer un terminal, accéder au dossier en question ensuite compiler le code source.
3. Exécuter le binaire résultant.

4. Dans la **Game-Loop** du jeu , ajouter le code permettant de calculer la valeur de **dt** à chaque itération.

```
Uint32 t_prev,dt=1; //dans la partie déclaration

while(!done) {

    t_prev = SDL_GetTicks(); //au début de la game loop

    ...

    dt=SDL_GetTicks()-t_prev; //à la fin de la game loop

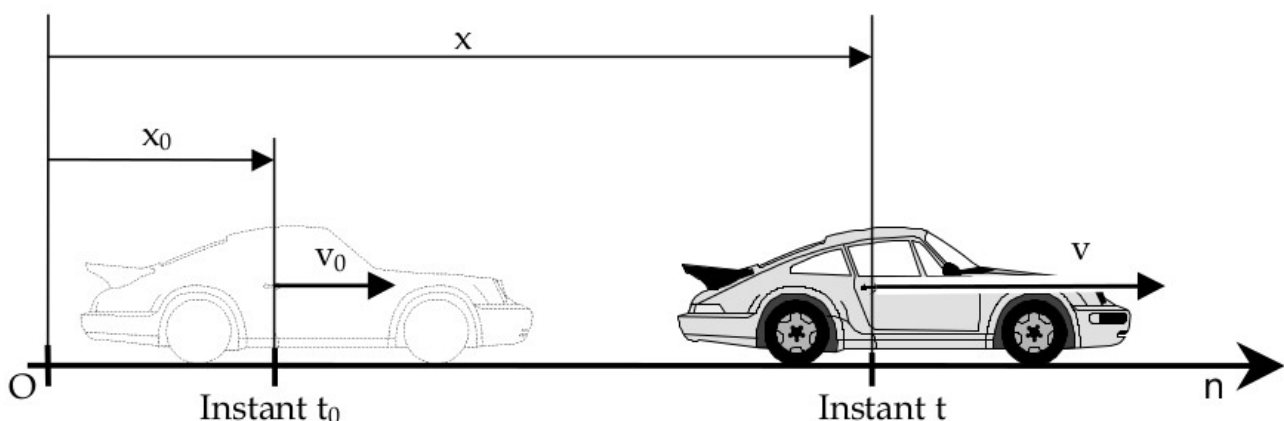
    if(1000/FPS > dt)
        SDL_Delay(1000/FPS - dt); //pour avoir un FPS constant
}
```

5. Tout en sachant qu'à chaque nouvelle itération de la boucle du jeu, la position  $x_n = x_{n-1} + dx$  , Implémenter le déplacement de la voiture moyennant la relation entre **dx** et **dt** (voir (1)).
6. Recompiler le code source et ré-exécuter le nouveau binaire résultant.

### c) Étude de cas: Mouvement rectiligne uniformément accéléré/décéléré:

Dans ce cas de figure l'accélération reste constante au cours du temps. On peut distinguer deux mouvements:

- Mouvement accélérés ( $a > 0$ )
- Mouvement décélérés ( $a < 0$ )



## ✓ Équations du mouvement:

soient:

$t_0$ : instant initial:  $t_0=0s$

$x_0$ : le déplacement initial, à  $t=t_0$

$a_0$ : l'accélération initiale

$v_0$ : la vitesse initiale à  $t=t_0$

$x(t)$ : le déplacement à l'instant  $t$

$v(t)$ : la vitesse à l'instant  $t$

Les équations relatives à l'accélération, la vitesse et la position instantané d'une entité en mouvement rectiligne uniformément accélérés sont:

$$a(t) = a_0 = \text{constante}$$

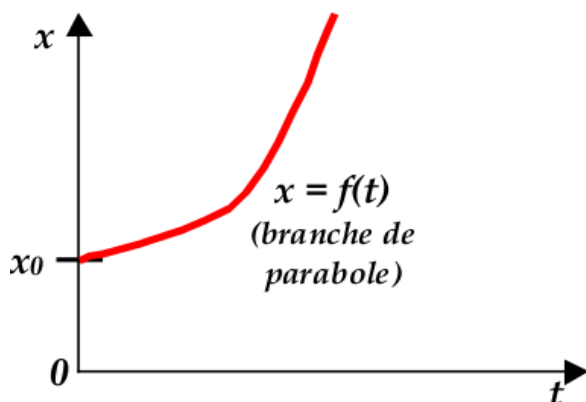
$$x(t) = \int (v(t) \cdot dt) = \int (a_0 \cdot t + v_0) \cdot dt = \frac{1}{2} a_0 t^2 + v_0 \cdot t + x_0$$

$$v(t) = a_0 \cdot t + v_0 \quad \text{et} \quad x(t) = \frac{1}{2} a_0 \cdot t^2 + v_0 \cdot t + x_0$$

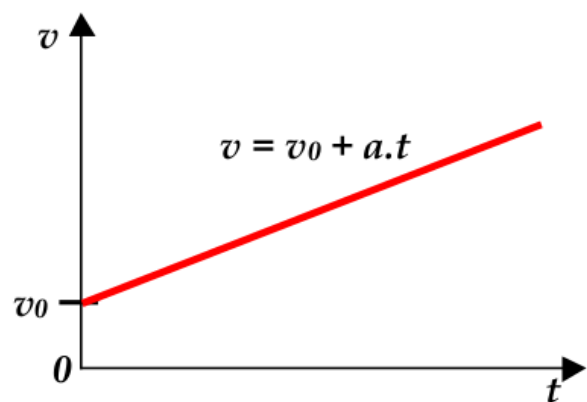
Pour un déplacement minime  $dx$ :

$$dx = \frac{1}{2} a_0 \cdot dt^2 + v_0 \cdot dt \quad (2)$$

### Graphe de position



### Graphe de vitesse

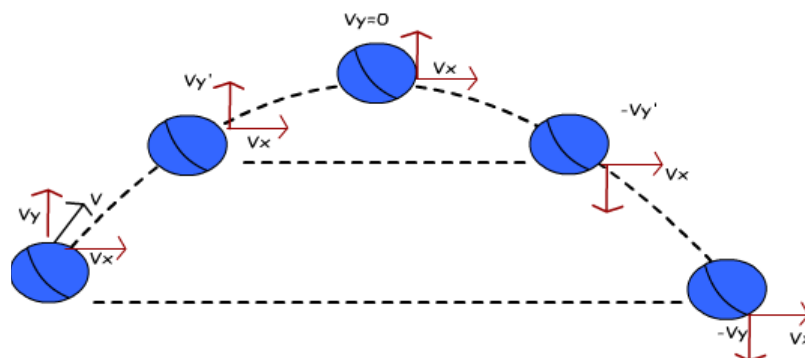


## Travail à Faire (Partie 2): (7pts)

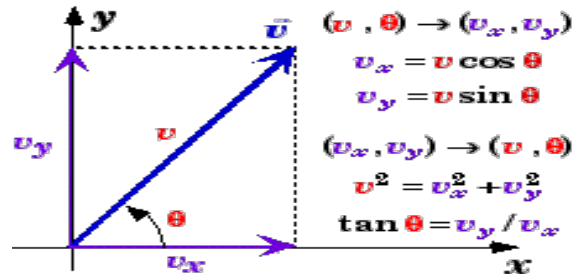
1. Copier le contenu du dossier **Part1** dans le dossier TPMathPhysics/Part2/
2. Lancer l'exécution du binaire **jeu (déjà existant dans le dossier Part2)**.
3. Tester l'exécution du binaire en question:
  - Accélérer l'avancement de la voiture en appuyant sur la touche (espace) du clavier.
  - Laisser la voiture avancer toute seule sans l'appui sur la touche d'accélération, la voiture s'arrêtera tout seule sous l'effet de la décélération causé par un certain facteur de frottement
  - décélérer l'avancement de la voiture ne appuyant sur la touche (retour chariot)
4. Au niveau de la **Game Loop** programmer l'événement de l'accélération(lors de l'appuie sur la touche espace) avec un cumul d'une constante **car.acceleration+=0.005** lors de l'appui sur espace.
5. Au niveau de la **Game Loop** programmer l'événement de décélération (lors de l'appuie sur la touche retour chariot) avec un cumul d'une constante **car.acceleration-=0.01** lors de l'appui sur espace.
6. Au niveau de la **Game Loop** ajouter une décélération avec un cumul de la constante **car.acceleration-=0.001** (A chaque itération et sans événement préalable)
7. Tout en sachant qu'a chaque nouvelle itération du Game Loop :  $x_n = x_{n-1} + dx$  , implémenter le déplacement de la voiture (fonction **deplacerVoiture** moyennant la variation de l'accélération et de la vitesse (voir (2)).
8. Compiler a nouveau, ensuite exécuter le binaire résultant.

## 2) Étude Cinématique: Mouvement Curviligne (dans un plan 2D):

Nous avons vus précédemment le cas d'un mouvement uni-dimensionnel. Maintenant, nous allons essayer d'expliquer le mouvement en deux dimensions qui est exactement appelés "mouvement de projectile". Dans ce type de mouvement, la gravité est le seul facteur agissant sur les objets en question.



Les mouvements verticaux et horizontaux de tout projectile sont indépendants les uns des autres. Cela signifie que tout problème de projectile peut être résolu en deux composantes: une composante verticale et une autre horizontale qui peuvent être résolues indépendamment les uns des autres.



Les équations génériques fondamentaux qui s'appliquent à tout problème de mouvement de projectile sont:

Composante Horizontale	Composante Verticale
Le Mouvement horizontal d'un projectile est équivalent à un corps se déplaçant à une vitesse constante de <b><math>v_{0x} = v_0 \cos(\theta_0)</math></b> dans une ligne droite horizontale. <b>(3)</b>	Le Mouvement vertical d'un projectile est équivalent à un corps lancé verticalement avec une vitesse initiale: <b><math>v_{0y} = v_0 \sin(\theta_0)</math></b> et subissant la gravité de la chute libre. En outre, la balle a une accélération vers le bas constante égale à <b><math>g</math></b> . <b>(3')</b>
$ax=0$ , $x(t)=v_x \cdot t + x_0$ et $v_x(t)=\text{constante}=v_{0x}=v_0 \cdot \cos(\theta_0)$ <b>(4)</b>	$ay=-g$ , $y(t)=\frac{-1}{2}g \cdot t^2 + v_0 \cdot t + y_0$ et $v_y(t)=-g \cdot t + v_{0y} = -g \cdot t + v_0 \cdot \sin(\theta_0)$ <b>(4')</b>

## Travail à Faire (Partie 3): (7pts)

1. Accéder au répertoire Part3/
  2. Exécuter le binaire **jeu**.
  3. Compiler le code source et exécuter le binaire résultant.
  4. Appuyer sur:
    - La touche **espace** afin de déplacer le tank
    - La touche **Right-Shift** afin de décélérer
    - La touche **Retour Chariot** afin de lancer un bullet.
- ➔ Le **bullet** devrait ne pas avancer correctement.

- ➔ A partir de la fonction **deplacerBullet**, vous devriez convertir l'angle du lancet en radian.
- ➔ Modifier la valeur de magnitude du **bullet** afin que le déplacement soit plus rapide.
- 5. Éditer le fichier **vector.h** dans lequel est défini le vecteur déplacement qui a pour composantes la **magnitude** et l'**angle** de direction du bullet.
- 6. Ajouter dans le même fichier (**vector.h**) la définition de deux vecteurs:
  - **Acceleration**: contenant deux composantes:  $a_x$  et  $a_y$
  - **Vitesse**: contenant deux composantes:  $v_x$  et  $v_y$
- 7. Ajouter dans la définition du **bullet (struct Bullet)** deux champs:
  - **struct Acceleration acc;** //qui définit l'accélération du bullet dans un plan 2D
  - **struct Vitesse vit;** //qui définit la vitesse du bullet dans un plan 2D
- 8. Pour  $t=0s$  (instant du lancet du bullet), Initialiser les valeurs de  $v_x, v_y, a_y$  et  $a_x$  à partir de la fonction **initBullet** (voir **(3)** et **(3')**) (pour info  $g=9.8$ )
- 9. pour  $t \neq 0s$  (Le Bullet est déjà lancé), Modifier la fonction **deplacerBullet** afin d'appliquer l'accélération due à la force de gravité. (voir **(4)** et **(4')** ).