

Cahier d'Exercices : Vecteurs & Physique

Thème 1 : Vecteurs et Produits Scalaires

Ces exercices visent à maîtriser les outils mathématiques de base pour la détection et l'IA.

Exercice 1 : Le Garde et le Voleur.

Contexte : Dans un jeu d'infiltration, un garde regarde dans une direction donnée par le vecteur directeur \vec{D} . Un voleur se trouve à une position P dans la pièce. Le garde se trouve à la position G .

On considère que le garde a un champ de vision de 180° (il voit tout ce qui est devant lui).

Données :

- Position du Garde : $G(10, 10)$
- Direction du regard : $\vec{D} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ (Regarde vers l'Est)
- Position du Voleur : $P(5, 12)$

Question : En utilisant le produit scalaire, déterminer si le voleur est « Devant » ou « Derrière » le garde.

Exercice 2 : La Racing Line (Projections).

Contexte : Pour aider une IA à conduire, on définit la trajectoire idéale comme une ligne droite entre deux points A et B . La voiture se trouve actuellement au point V .

On souhaite connaitre la distance latérale entre la voiture et cette trajectoire idéale (l'écart de conduite).

Données :

- Départ de la ligne : $A(0, 0)$
- Fin de la ligne : $B(100, 0)$
- Position Voiture : $V(50, 5)$

Question :

1. Calculer le vecteur \overrightarrow{AV} .
2. Projeter ce vecteur sur la direction de la piste \overrightarrow{AB} .
3. En déduire la distance de la voiture à la ligne.

Exercice 3 : Le Radar de Vitesse (Vitesse Relative).

Contexte : Un radar mesure la vitesse de rapprochement sur sa ligne de visée. Pour savoir ce que le radar affiche, il faut projeter les vitesses des véhicules sur l'axe qui les relie.

Données (Positions et Vitesses) :

- Police (A) :
 - Position : $P_A = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
 - Vitesse : $\vec{v}_A = \begin{pmatrix} 0 \\ 100 \end{pmatrix}$ (Roule vers le Nord à 100 km/h)
- Cible (B) :
 - Position : $P_B = \begin{pmatrix} 0 \\ 500 \end{pmatrix}$ (Est située 500m au Nord de la police)
 - Vitesse : $\vec{v}_B = \begin{pmatrix} 85 \\ 85 \end{pmatrix}$ (Roule vers le Nord-Est à ≈ 120 km/h)

Questions :

1. Calculer le vecteur direction de visée $\vec{D} = P_B - P_A$.
2. En déduire la normale de visée \vec{n} (le vecteur unitaire).
3. Projeter les vitesses \vec{v}_A et \vec{v}_B sur \vec{n} (Produit Scalaire).
4. La police est-elle en train de rattraper la cible ? (Est-ce que v_A projeté > v_B projeté ?)

Thème 2 : Forces, Gravité et Frottements

Application des lois de Newton pour simuler des mouvements réalistes.

Exercice 4 : Le Drag Race (Code).

Contexte : Une voiture accélère en ligne droite.

- Force Moteur : $F_{\text{moteur}} = 5000 \text{ N}$ (Constante).
- Friction de l'air : $F_{\text{air}} = -k \cdot v^2$ (Opposée au mouvement).
- Masse : $m = 1000 \text{ kg}$, Coefficient $k = 0.8$.

Objectif Code : Implémenter ce système dans Three.js. Afficher la vitesse en temps réel. Observer la « Vitesse Terminale » : le moment où la voiture n'accélère plus car la friction compense exactement le moteur.

Exercice 5 : Le Sniper (Balistique). **Contexte :** On tire un projectile soumis uniquement à la gravité $\vec{g} = \begin{pmatrix} 0 \\ -9.81 \end{pmatrix}$.

Données :

- Vitesse initiale : $v_0 = 50 \text{ m/s}$.
- Angle de tir : $\theta = 45^\circ$.
- Hauteur initiale : $y_0 = 0$.

Objectif Code : Trouver ce résultat dans une simulation avec la méthode d'Euler.

Exercice 6 : Le Cube sur la Pente (Code). **Contexte :** Reprise de l'exercice théorique sur le plan incliné.

Mise en place : Créer un plan (sol) dans Three.js dont on peut modifier l'inclinaison (rotation X) via une interface (GUI). Poser un cube dessus.

Logique à coder :

- Calculer la force de gravité parallèle à la pente : $P_x = mg \sin(\theta)$.
- Calculer la friction statique max : $f_{s, \text{max}} = \mu_s mg \cos(\theta)$.
- **Condition :** Tant que $P_x < f_{s, \text{max}}$, la vitesse reste à 0. Sinon, le cube accélère.

Test : Vérifier si le cube se met à glisser exactement à l'angle prévu (ex: 30° pour $\mu_s = 0.57$).

Exercice 7 : L'Ascenseur Spatial (Ressorts). **Contexte :** Une boule est suspendue dans le vide, attachée à un point fixe par un ressort invisible.

Loi de Hooke : $\vec{F} = -k \cdot (L - L_0) \cdot \vec{n}$ (Où L est la longueur actuelle et L_0 la longueur au repos).

Objectif Code : Simuler ce ressort. Ajouter une force d'amortissement (Damping) $F = -b \cdot v$ pour que la boule finisse par se stabiliser et ne rebondisse pas à l'infini.

Définition 1 (Thème 3 : Collisions et Impulsions)

Gestion des chocs et conservation de la quantité de mouvement.

Exercice 8 : Le Bumper Car (Code). **Contexte :** Arène fermée avec 5 boules de masses identiques.

Objectif Code : Mettre en place la boucle de détection « Narrow Phase » : vérifier chaque paire de boules (i, j) . Si $\text{distance} < \text{radius} * 2$, résoudre la collision élastique ($e = 1$).

Observer le chaos et vérifier que les boules ne « fusionnent » pas (stabilité).

Exercice 9 : Le Matériau Mystère. **Contexte :** On lâche 3 boules de la même hauteur $H = 10m$. Elles tombent sur le sol (masse infinie).

- Boule A : Remonte à 10m.
- Boule B : Remonte à 5m.
- Boule C : Reste collée au sol.

Question : Déterminer le coefficient de restitution e pour chaque boule. Rappel : $v_{\text{après}} = -e \cdot v_{\text{avant}}$.

Exercice 10 : Le Rocket Jump. **Contexte :** Une explosion se produit à la position $E(0, 0, 0)$. Un joueur se trouve à la position $P(2, 0, 0)$.

On veut propulser le joueur avec une impulsion radiale.

Règle :

- Direction : Du centre de l'explosion vers le joueur.
- Intensité : $J = \frac{100}{\text{distance}^2}$.

Question : Calculer le vecteur Impulsion \vec{J} à appliquer au joueur.