

# Informatique

## Présentation du sujet

Le sujet porte sur la simulation du mouvement des particules d'un gaz suivant le modèle des sphères dures. Dans ce modèle le rayon des particules n'est pas négligeable, les interactions des molécules entre elles et avec les parois sont des chocs élastiques. L'algorithme étudié présente une simulation à événements discrets, qui consiste essentiellement à déterminer les instants où les chocs ont lieu.

Le problème envisage successivement, en une ou plusieurs dimensions :

- le problème d'un placement aléatoire initial des sphères, sans chevauchement ;
- l'étude des lois de modification du mouvement après un choc ;
- l'étude des lois de prévision d'événements (chocs) futurs ;
- une structure de gestion des événements ;
- l'intégration des fonctions précédemment écrites en un algorithme global.

Une dernière partie propose enfin l'utilisation d'une base de données pour étendre l'étude au cas de particules de masses différentes.

## Analyse globale des résultats

L'épreuve d'informatique est une épreuve simple : les meilleurs candidats ont traité de façon très satisfaisante la totalité du sujet. Elle vise à valider un socle minimal de compétences informatiques que doit posséder le futur ingénieur, mises en œuvre sur un problème contextualisé.

De nombreux candidats produisent des copies remarquables, où chaque question reçoit une réponse soignée, concise et élégante. Le jury est alors enclin à pardonner les inévitables petites fautes inhérentes à l'informatique sur papier. Des points transversaux sont également attribués globalement, pour bonifier les copies où la syntaxe est rigoureuse, le code raisonnablement commenté, les noms de variables judicieusement choisis.

À contrario, certains cumulent les handicaps : souvent par manque de maîtrise de la syntaxe de base, ils proposent des solutions raturées, mal indentées, longues et confuses. Le jury fait en général l'effort de repérer les idées intéressantes, mais ne peut valoriser de telles rédactions à la hauteur des copies précédentes.

## Commentaires sur les réponses apportées et conseils aux futurs candidats

L'informatique permet, après une bonne analyse du résultat attendu, d'élaborer des stratégies très simples. Leur mise en œuvre est alors grandement facilitée et produit des programmes courts et clairs. Les bonnes copies sont toujours celles où les phases d'analyse ont été les plus soignées. Pour le même résultat, les solutions vont souvent de trois à vingt lignes.

Les commentaires sont bienvenus quand ils permettent de clarifier le rôle d'une variable ou d'une ligne qui le nécessite. Certains candidats en abusent cependant, en sur-commentant des blocs à la fonctionnalité évidente, voire chaque ligne. Un code épuré, bien indenté, avec des noms de variables judicieux est souvent préférable et suffisant.

L'indentation en python délimite les blocs d'instructions et doit apparaître clairement dans la rédaction ; bien souvent, un trait vertical marquant l'alignement du bloc d'instructions est suffisant. Un grand nombre de niveaux d'indentation rend la lecture délicate ; il vaut mieux dans ce cas introduire des fonctions intermédiaires.

Toute boucle doit être exécutée un nombre précis de fois. Une attention particulière doit être apportée à ce critère, car il signale en général une bonne compréhension de l'algorithme. Au contraire, quand il n'est pas respecté, le programme est en général inexact.

Les opérateurs booléens s'évaluent de gauche à droite avec arrêt de l'évaluation dès que possible. Ainsi `while i < n and t[i] < x` est correct pour un tableau de taille `n`, tandis que l'inversion des conditions peut conduire à un accès erroné à `t[n]`.

Lors d'un calcul de complexité, une justification minimale est attendue.

Une part non négligeable de la note tient compte de la clarté et la lisibilité de la copie. Sont prises en compte ici l'orthographe, le soin, la présence de schémas, le respect de l'ordre des questions, etc.

La structure de l'épreuve d'informatique n'est pas figée, mais comporte souvent plusieurs parties relativement indépendantes. Il est souvent opportun de s'organiser pour les aborder toutes.

## **I Initialisation**

La première partie vise à évaluer plusieurs stratégies d'initialisation de la simulation, d'abord en une, puis en trois dimensions.

Les questions 1 à 5 sont bien réussies. Quelques candidats détaillent en français les opérations élémentaires réalisées (multiplication, tirage aléatoire, etc.) alors qu'il était attendu une explication du sens de ces lignes pour le problème posé. Un schéma était souvent plus clair qu'un long discours pour les questions 3 et 4.

La question 10 demande d'écrire un algorithme décrit dans l'énoncé. Les examinateurs ont accepté tout algorithme qui permettait de répartir sans interférence les boules, même lorsque la répartition n'est pas équiprobable. Beaucoup de propositions oublient que la liste n'est pas triée après tirage aléatoire, ce qui rend les décalages inopérants. La question 12 est souvent correctement traitée pour  $N=1$  et  $N=5$ . Le cas  $N=2$  nécessitait une analyse plus fine brillamment trouvée par quelques candidats.

La question 13 permettait d'envisager une reformulation du code donné pour une simulation en trois dimensions. Le critère de non chevauchement a rarement été correctement perçu.

## **II Mouvement des particules**

La deuxième partie s'attache à décrire le mouvement (vol libre, rebond sur la paroi ou choc entre particules).

L'analyse des lois physiques a posé peu de difficultés aux candidats, bien que le jury constate que la notion de mouvement rectiligne uniforme ne soit pas toujours claire. Les fonctions à écrire étaient très simples, mais il fallait savoir modifier le paramètre mutable fourni en argument.

## **III Inventaires des évènements**

La troisième partie étudie la prévision des événements futurs et élabore une structure de données mémorisant ces événements dans un catalogue.

Beaucoup de candidats ont eu des difficultés à traduire informatiquement les conditions booléennes de collision entre particules et paroi pour les fonctions `tr` et `tc`. Peu de tests bien ordonnés suffisent ; un manque d'analyse conduit beaucoup de candidats à envisager une pléthore de cas redondants et souvent non exhaustive.

L'ajout d'un évènement nécessitait d'insérer un élément dans une liste triée, algorithme explicitement au programme. Trop souvent le cas où l'élément à insérer se place au début ou à la fin de la liste est mal pris en compte.

L'ajout de tous les évènements relatifs à une particule est relativement bien abordé, mais beaucoup de candidats oublient de vérifier que les fonctions `tr` et `tc` ne renvoient pas `None` avant d'ajouter le résultat au catalogue. Certains ont cherché à sélectionner l'évènement le plus proche, contrairement à ce que demande le sujet.

L'initialisation du catalogue n'a pas posé de difficulté, mais le calcul de complexité nécessitait de remarquer que la taille du catalogue n'est pas  $O(N)$ , ce que très peu de candidats ont vu. Un calcul de complexité ne se limite pas toujours à compter les boucles for imbriquées.

#### IV Simulation

Cette partie est la synthèse des précédentes et ne met pas en valeur les mêmes qualités. Aucune difficulté algorithmique ici, mais il faut bien comprendre l'articulation des différentes parties du problème entre elles.

La fonction étape permettait de tenir compte du déplacement rectiligne uniforme des particules entre deux évènements (vol), et de traiter l'évènement (le rebond ou le choc). Certains candidats oublient la phase de vol, soit pour toutes les particules, ou pour les particules subissant l'évènement.

La mise à jour du catalogue est souvent incomplète, en particulier pour la gestion des dates. De même, la gestion correcte des évènements invalides et du temps dans la simulation est rarement juste.

#### V Exploitation des résultats

La cinquième partie demandait d'élaborer trois requêtes SQL d'extraction de résultats. Bien qu'elles soient en fin de sujet, ces questions sont la plupart du temps abordées, avec des résultats plutôt corrects. L'instruction `GROUP BY`, relative aux fonctions d'agrégation, n'est pas toujours connue. Certaines requêtes utilisent des jointures inutiles.

#### Conclusion

Le jury se satisfait d'une préparation de bonne qualité à cette épreuve, la plupart des candidats témoignent d'un niveau de maîtrise satisfaisant. Les outils numériques (mumpy, ...) sont de mieux en mieux utilisés, certains candidats faisant même preuve d'une dextérité étonnante.

Le sujet aborde une large partie du programme d'informatique commune. Le choix d'un sujet s'appuyant la simulation d'un phénomène physique par une approche numérique, impliquant une part d'algorithmique, assure une cohérence avec la formation d'ingénieurs. Cette approche sera reconduite sur des problématiques de simulation ou d'algorithmique en informatique, à partir du programme des trois semestres d'informatique.