



## 1/ CONSIGNES GENERALES :

### a) Présentation du sujet

Le sujet portait sur la résolution de l'équation de la chaleur par des méthodes numériques. La première partie physique permettait d'établir les équations et les conditions aux limites sur lesquelles la seconde partie allait travailler pour trouver une solution approchée par des méthodes numériques. Deux méthodes classiques de résolution étaient proposées : une par méthode explicite, assez facile à mettre en œuvre mais nécessitant un pas de temps faible et une par méthode implicite, un peu plus délicate mais plus stable.

Le sujet était de difficulté moyenne et alliait des questions de physique, de mathématiques et de programmation. Un des objectifs du sujet était de faire réaliser par le candidat le programme dans son intégralité, morceau par morceau. En conséquence, certaines questions étaient redondantes.

L'énoncé était suffisamment clair puisque la plupart des candidats ont bien compris ce qui leur était demandé. Il était découpé en parties indépendantes et certains résultats intermédiaires étaient donnés, ce qui permettait aux candidats de ne pas être bloqués.

Le niveau de difficulté des questions était varié, ce qui a permis de classer les candidats.

Une annexe présentait les principales fonctions de Python et de Scilab utiles à la résolution de ce sujet, ce qui permettait d'aider les candidats aux connaissances informatiques fragiles et permettait de ne pas défavoriser les élèves 5/2.

### b) Problèmes constatés par les correcteurs

- De nombreux candidats ne savent pas faire correctement un développement limité.
- Très peu de candidats utilisent correctement l'instruction « print ».
- Beaucoup d'erreurs d'indices.
- Beaucoup d'étudiants oublient d'initialiser les matrices.
- Certains étudiants confondent les signes « == » et « = ».
- De nombreux étudiants font des fonctions sans nécessité.
- Certaines questions de programmation pouvaient être traitées de différentes manières, ce qui rendait la notation difficile.
- La plupart des candidats ont abordé le sujet avec sérieux, sauf quelques rares candidats qui n'ont pas du tout traité les questions de programmation.

Un nombre non négligeable de candidats ne font pas encore la différence entre l'écriture mathématique d'une expression et son écriture en langage de programmation (grandeurs ou fonctions non définies, non-respect de la syntaxe pour les opérations de multiplication, division, élévation à la puissance, etc.), ce qui a été difficile à corriger et à sanctionner, surtout quand la réponse était correcte.

Dans la plupart des copies, les indentations sont clairement visibles, soit en respectant un décalage constant et clair, soit en indiquant directement les indentations avec des lignes verticales (objets au même niveau d'indentation) ou horizontales (nombre d'indentation à chaque ligne). Toutefois, certaines copies présentaient des indentations « fluctuantes ». Cela nuit fortement à la lecture du code.

Les copies étaient globalement propres à de rares exceptions près. Il est important que les candidats mettent en valeur les résultats à l'issue de leur raisonnement et indique clairement la question qu'ils sont en train de traiter.

Les commentaires étaient trop peu présents. Il est important que le correcteur puisse comprendre la démarche des candidats surtout lorsque plusieurs solutions sont envisageables. Certains candidats ont utilisé des couleurs différentes pour les commentaires, c'est une très bonne idée pour les faire ressortir.

Les candidats ont pour beaucoup utilisé les syntaxes de liste plutôt que la syntaxe tableau de numpy. Ceci a induit, pour une partie des copies, des programmations 'lourdes' et des erreurs d'affectation dans les tableaux. Certains ont alors programmé des opérations telles que la multiplication par un scalaire, l'addition et la différence, ce qui alourdissait inutilement la programmation.

Lors de l'écriture d'un « while », surtout avec plusieurs conditions d'arrêt, les candidats seraient bien avisés d'écrire la négation mathématique. « Je veux m'arrêter si P » correspond à « je continue tant que non(P) » ; cela éviterait peut-être des erreurs du type « or » au lieu de « and » et « < » au lieu de « >= ».

Des élèves ont fait un usage inutile de la commande « break ».

Des affectations de variables sont réalisées dans le mauvais sens ( $2000 = \text{nbIter}$ ).

## 2/ REMARQUES SPECIFIQUES :

### PARTIE I : ETUDE PRELIMINAIRE

#### I.A. Equation gouvernant la température

I.A.1. Question souvent mal traitée par les candidats.

I.A.2. Question bien traitée. L'équation de la diffusion thermique est connue ainsi que l'expression du coefficient de diffusion en fonction des paramètres du solide.

#### I.B. Conditions aux limites

La condition aux limites de type flux nul a été rarement énoncée correctement.

#### I.C. Solutions en régime permanent

La résolution en régime permanent et le tracé des solutions sont dans l'ensemble très bien traités. Le tracé des profils intermédiaires est souvent bien réalisé : l'erreur la plus classique consiste en un tracé de droites où la température en  $x = e$  varie au cours du temps.

### PARTIE II : RESOLUTION NUMERIQUE

#### II.A. Equation à résoudre

II.A.1. Quelques erreurs.

II.A.2. Certains utilisent Text2 au lieu de Text1, mais ils n'ont pas été pénalisés.

## II.B. Méthode des différences finies

II.B.1.a. et b. Questions bien réussies.

II.B.2. Méthode utilisant un schéma explicite.

II.B.2.a. Beaucoup d'erreurs dans le développement limité DL : les coefficients étaient faux, le DL était fait au mauvais ordre, oubli du  $o(Dx^3)$ .

II.B.2.b. et c. Questions plutôt bien réussies. Le candidat a obtenu des points pour le raisonnement même s'il a fait des erreurs aux questions précédentes.

II.B.2.d. Question bien réussie. Quelques oublis du  $o(Dt)$ .

II.B.2.e. et f. Questions bien réussies.

II.B.2.g. et h. Questions plutôt bien réussies. Le candidat a obtenu des points pour le raisonnement même s'il a fait des erreurs aux questions précédentes.

II.B.2.i. Question bien réussie.

II.B.2.j.(i) Beaucoup oublient les variables "Tint" et "Text" en argument d'entrée.

II.B.2.j.(ii) Beaucoup d'élèves n'avertissent pas l'utilisateur de la mauvaise valeur de « r », contrairement à ce qui était demandé dans l'énoncé. Beaucoup ne semblent pas connaître la commande « print ».

II.B.2.j.(iii) Question bien traitée mais oubli fréquent de définir la valeur de « N ». Parfois, affectation inversée «  $2000=NbIter$  ».

II.B.2.j.(iv) Question généralement bien traitée.

II.B.2.j.(v) Des erreurs d'indice.

II.B.2.j.(vi) Question très bien réussie dans l'ensemble.

II.B.2.j.(vii) Des erreurs dans la boucle « while » (utilisation de « or » au lieu de « and », erreurs dans les signes d'inégalité). Des erreurs d'indice.

II.B.2.j.(viii) Certains oublient de donner une valeur à « nbIter » avant de la renvoyer.

II.B.3. Méthode utilisant un schéma implicite.

II.B.3.a. et b. Questions plutôt bien réussies. Le candidat a obtenu des points pour le raisonnement même s'il a fait des erreurs aux questions précédentes.

II.B.3.c. Question assez bien réussie. Matrice M globalement bien identifiée. Beaucoup d'erreurs sur le vecteur v.

I.B.3.d.(i) Beaucoup d'erreur d'indices. Les vecteurs n'ont souvent pas le bon nombre d'éléments. Dernière boucle rarement écrite correctement.

I.B.3.e.(i) Beaucoup oublient « Tint » et « Text » en argument d'entrée.

I.B.3.e.(ii) Question bien traitée mais oubli fréquent de définir la valeur de « N ». Parfois, affectation inversée «  $2000 = \text{NbIter}$  ».

I.B.3.e.(iii) Question généralement bien traitée.

I.B.3.e.(iv) Beaucoup d'erreur d'indice dans les boucles et mauvaises définitions des éléments aux bords de la matrice.

I.B.3.e.(v) Peu de candidats ont répondu à cette question. Question plutôt bien traitée mais certains oublient de rajouter «  $r*v$  » au deuxième argument d'entrée de « CalcTkp1. »

I.B.3.e.(vi) Des erreurs dans la boucle « while » (utilisation de « or » au lieu de « and », erreurs dans les signes d'inégalité). Des erreurs d'indice. Certains élèves n'appellent pas la fonction « CalcTkp1 ».

I.B.3.e.(vii) Certains oublient de donner une valeur à « nbIter » avant de la renvoyer.

## II.C. Programme principal

II.C.1.a. Certains confondent écriture mathématique et programmation et utilisent d'autres noms de variables que ceux donnés dans l'énoncé, avec des symboles grecs par exemple ou des indices/exposants dans les noms de variables. Certains se compliquent la tâche en utilisant des fonctions.

II.C.1.b. Question plutôt bien traitée sauf quand « Text2 » a été utilisé au lieu de « Text1 ».

II.C.1.c. Beaucoup d'erreurs dans cette question. Le vecteur n'a souvent pas la bonne taille ou ses indices sont faux.

II.C.1.d. Question bien réussie.

II.C.1.e. Certains confondent écriture mathématique et programmation.

II.C.2. Certains ne connaissent pas la commande « input ». Non renvoi de « T\_tous\_k » et « nbIter ».

II.C.3. Analyse du résultat

II.C.3.a. Les courbes étaient généralement présentes sur un graphique avec des labels aux axes, toutefois de trop nombreux candidats ont voulu tracer  $T(t)$  alors que  $T(x)$  était demandé. Certains oublient de faire afficher le graphique avec « pl.show() ».

II.C.3.b. La question a été très peu abordée alors qu'il ne s'agissait que d'une simple conversion nombre de pas de temps vers heure en utilisant  $Dt$ . Certains candidats ont oublié de diviser le temps par 3 600 pour obtenir un résultat en heures. Tous les candidats ne connaissent pas la commande « print ».

## **3/ CONCLUSIONS :**

Une immense majorité des candidats a utilisé le langage Python. Moins de 10 élèves ont traité l'épreuve en langage Scilab.

De nombreuses questions étaient très faciles, en particulier dans la partie physique et dans la résolution numérique par schéma implicite. Quelques questions, en particulier pour le schéma implicite, étaient plus délicates. Le sujet a donc permis de trier les candidats.

Les étudiants ayant fait preuve de sérieux dans les différentes matières ont rencontré peu de difficulté.