Retour d'oral : Modélisation - Option B

Mohamed NASSIRI

Tirage :

☐ Un texte traitant d'EDO avec plusieurs ressorts modélisant une chaîne d'ADN. Mots clefs : Equations différentielles ordinaires. Propriétés qualitatives des solutions.
☒ Un texte traitant de la résolution d'EDP par des méthodes d'optimisation. Mots clefs : Optimisation. Algèbre linéaire. Méthodes itératives.

Spectateurs :

0

Préparation :

Alors moment de panique ... Je lis le texte sur les EDO et je me rends compte qu'il est vraiment horrible ! Tant pis, je regarde l'autre ... Pas mieux ... La partie qui consiste à résoudre l'équation de Laplace avec une méthode de minimisation, bah c'est un peu trop compliqué pour le comprendre et l'implémenter en 4h ... D'autant que je n'ai jamais bosser des textes sur l'optimisation (seulement les cours). Lueur d'espoir ! Je vois qu'un des théorèmes à démontrer c'est l'équivalence entre le problème $Ax = b$ et la minimisation d'une fonctionnelle quadratique avec A une matrice symétrique définie positive. Bon ça je sais faire ! Je n'avais jamais codé l'algorithme du gradient à pas optimal, mais il n'est pas trop dur ; j'ai su le coder. Et après je vois l'équation de Laplace, et celle-ci utilise les matrices symétriques définies positives dans le schéma : la matrice du Laplacien. Bon le truc qui me fait paniquer, c'est qu'on ne considère pas la conductivité thermique (notée a) constante. Du coup, on a des intrégrales dans nos schémas ... Super de l'intégration numérique en plus ... (*Appétissant hein !*). Je regarde parmi les questions. Jackpot ! " Proposer une autre méthode (numérique) pour résoudre l'équation de Laplace. On discutera selon les valeurs de a"! Je prends a constante et je dirais pour finir que pour un meilleur modèle on devrait ne pas la considérer constante ... (*Même si dans le texte, dès le début, il ne considère pas ça constant. Mais c'est soit ça, soit je fais rien.*)
Bon je tiens mon truc (*et je prie pour ne pas avoir fait de hors-sujet à partir de II)*).
J'ai donc fait le plan suivant (avec un titre, (*il semblerait que ce soit un plus*)) :

De l'importance des matrices symétriques définies positives. (*Je précise que je noterais SDP par la suite*)
I) Système linéaire et problème de minimisation
   1) Attention aux erreurs
   2) Equivalence de problèmes
   3) Algorithme du gradient à pas optimal
II) Application : Equation de Laplace (*Je précise que c'est une application des matrices symétriques définies positives*)
   1) Modélisation
   2) Schéma numérique et propriétés
   3) Algorithme
III) Vers un modèle plus réaliste
   1) a n'est plus constante (*a c'était la conductivité thermique d'une barre métallique*)

2) Dépendance du temps
3) Conditions aux bords
4) Deux dimensions
5) Eléments finis

Exposé :

J'expose rapidement le problème et j'écris le plan au tableau.
I) 1) Je commence avec un petit algorithme (*il semblerait que ce soit un plus aussi*) avec le problème des matrices perturbés avec le célèbre exemple dû à Wilson. J'explique qu'avec des erreurs type erreurs machine, nos solutions du système légèrement perturbé peuvent être loin du système de départ. Je dis également que notre matrice est sympathique : elle est symétrique, son déterminant vaut 1 et ses valeurs sont entières. Sa matrice inverse est tout aussi sympathique d'ailleurs. Dans mon algorithme, on calculait la norme de la différence et ça valait environ 12. Et, toujours dans mon algorithme, un *messagebox* affichait "C'est normal ! Le conditionnement vaut 2900!". Du coup, je donne un théorème suivant (il n'est pas dans le texte, et je précise que je ne le démontre pas là) sur le conditionnement :
Soit $A \in \mathrm{GL}_n(\mathbb{K})$ et soit u et $\tilde{u}$ les solutions des systèmes linéaires

$$\begin{aligned} &Au = b \\ &\tilde{A}\tilde{u} = b \quad \tilde{A} = A + \Delta A,\ \tilde{u} = u + \Delta u \end{aligned}$$

On suppose $b \neq 0$. Alors on a

$$\frac{||\Delta u||}{||\tilde{u}||} \leq \mathrm{cond}(A)\frac{||\Delta A||}{||A||}$$

2) Je dis qu'une autre façon de résoudre c'est de considérer un problème de minimisation. Je note le thèorème suivant et je le démontre sans problème (c'était une des questions):
Soient A une matrice symétrique définie positive. Alors $x \in \mathbb{R}^n$ est solution de

$$Ax = b$$

si et seulement si

$$f(x) = \min_{y \in \mathbb{R}^n} f(y), \text{ avec } f(y) = \langle Ay, y\rangle - \langle b, y\rangle$$

3) J'explique la méthode de grandient à pas optimal. Je fais le dessin avec les lignes de niveaux, et je dis que l'on prend la direction du gradient d_k et que l'on va chercher un α_k qui minimise $f(x_{k+1})$ où $x_{k+1} = x_k + \alpha_k d_k$. Je note rapidement cette condition et je développe un peu pour dire que l'on trouve un bon vieux polynôme de degré 2 dont on connait le minimum et que c'est un minimum car A est SDP. Je donne la formule du minimum. De là, je dis que cela nous pousse à poser l'algorithme suivant :

$$\begin{cases} x^0 \in \mathbb{R}^n,\ r^0 = b - Ax^0 \\ \alpha_k = \frac{||r_k||^2}{\langle Ar^k, r^k\rangle} \\ x^{k+1} = x^k + \alpha_k r^k \\ r^{k+1} = r^k + \alpha_k A r^k \end{cases}$$

Et je montre mon super algorithme ! J'avais créé une matrice qui stockait mes itérés, mes résidus, j'ai utilisé la fonction $x(:, \$)$ pour récupérer le dernier vecteur, j'avais un *messagebox* qui affichait le temps d'exécution de l'algorithme, et la norme de la différence entre la solution avec l'algorithme du gradient à pas optimal et

avec la commande *backslash* ... Mais l'erreur était trop grande ... Et je leur dis.

II) 1) Je commence cette partie en disant qu'une bonne application des matrices symétriques définies positives, c'est l'équation de Laplace. Je propose une modélisation mais je me rends compte que je passe beaucoup trop vite dessus ... Et j'obient l'équation (avec un terme source)

$$-a\Delta u(x) = S(x)$$

2) J'explique ici le principe de la discrétisation et je calcule, pour une fonction v suffisamment régulière,

$$\begin{aligned} v_{j\pm1} = v(x_{j\pm1}) &= v(x_j \pm h) \\ &= v(x_j,t_n) \pm h\frac{\partial}{\partial x}v(x_j,t_n) + \frac{h^2}{2}\frac{\partial^2}{\partial x^2}v(x_j,t_n) \pm \frac{h^3}{6}\frac{\partial^2}{\partial x^2}v(x_j,t_n) + \mathcal{O}(h^4) \end{aligned}$$

et qu'ensuite on

$$\frac{v_{j+1} - 2v_j + v_{j-1}}{h^2} = \frac{1}{2}\frac{\partial^2}{\partial x^2}v(x_j,t_n) + \mathcal{O}(h^2)$$

et donc pour u solution de l'équation différentielle, on a

$$\frac{v_{j+1} - 2v_j + v_{j-1}}{h^2} - S(x_j) = \mathcal{O}(h^2)$$

(*J'ai perdu le coefficient $-a$ dans la bataille mais le jury ne m'a rien dit là dessus ...*)
et je dis qu'on pourrait employer des gros mots et dire que l'erreur de consistance est d'ordre 2. Cela nous pousse donc à considérer le schéma suivant

$$\frac{v_{j+1} - 2v_j + v_{j-1}}{h^2} = S_j, \quad \text{pour } 1 \le j \le J, \text{ avec } S_j = S(x_j)$$

et que matriciellement, cela revient à

$$AU = F \quad \text{avec } A = -\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}, \; U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{J-1} \\ u_J \end{pmatrix}, \; F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{J-1}) \\ f(x_J) \end{pmatrix}$$

(*Ne demandez pas pourquoi ma fonction S est devenue une fonction f.*)
Je donne ensuite le théorème suivant (*ce n'était pas un théorème du texte*) :

La matrice A est SDP.

que je l'ai démontré, classiquement, en prenant un vecteur $(x_1, ..., x_J) \in \mathbb{R}^J$ et en écrivant ${}^t xAx$ comme une somme de carrés.
(*D'ailleurs, à ce moment, j'ai l'impression qu'il me reste à peine 2 minutes, donc je demande, et le jury me dit qu'il m'en reste encore 8 ...*)

3) Je présente donc mon premier algorithme. Je dis que j'ai pris une solution exacte que je connaissais, avec un terme source bien choisi. Il affiche un *plot* avec la solution exacte et la solution du schéma (avec des points rouges), une belle légende, les titres en LaTeX. Je dis également que l'on peut changer le pas de discrétisation, donc je donne une autre algorithme. En l'exécutant, ça affiche une bôite de commande où l'on peut sélectionner plusieurs discrétisations (j'avais mis 10, 25, 50, 100). J'explique que je stocke les erreurs dans un vecteur, que j'utilise la fonction *sparse* pour ne pas stocker les zéros inutiles de la matrice du

Laplacien, et que le *plot* qui s'affiche est celui qui donne la solution exacte et la solution du schéma pour la dernière discrétisation (ici, c'était 100). Je précise également que j'utilise le *backslash* et qu'en règle général, c'est pas bien mais qu'ici ce n'est pas grave en perte de coût algorithmique car ce qui ce cache derrière cette commande c'est une décomposition LU et que notre matrice est tridiagonale donc sa décomposition LU est peu coûteuse et que c'est encore sous la forme tridiagonale. Un *messagebox* affichait également la norme de l'erreur (toujours pour la dernière discrétisation). Mais je dis justement que l'erreur me semble un peu trop grande, c'était 0,5 environ. Et là je dis qu'il me manque un *plot* et que je suis bête car j'avais utilisé la commande *xclick* et que *Scilab* attendait donc que je clique ... (*Oui des fois, je suis vraiment débile ...*). Donc je clique et là s'affiche un *plot* avec l'erreur en fonction de la discrétisation en échelle $log - log$. Sauf que la pente ne vaut pas 2, et que normalement, c'est ce qu'on est sensé trouver. Je précise que dans le titre de mon *plot*, j'utilise la commande *reglin* pour chercher la pente de l'erreur en fonction de la discrétisation et que j'utilise la commande *string*() pour afficher cette pente dans le titre du plot. La pente valait 0.458 environ (*Il me semble, à une vache près ...*)

III) 1) Pour finir, je précise que l'on peut avoir un meilleur modèle en ne consiérant plus a constante.

2) On peut considérer une dépendance par rapport au temps et donc que l'on obtiendrait l'équation suivante :

$$\frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2} = 0$$

Je précise qu'il faut coder en implicite pour ne pas avoir de problème avec des conditions type CFL.

3) On pourrait changer les conditions aux bords : ne plus considérer les conditions de Dirichlet

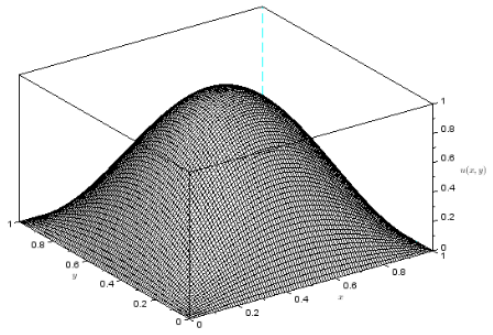
$$\begin{cases} u(0) = 0 \\ u(L) = 0 \end{cases} \quad , \text{ mais plutôt } \quad \begin{cases} u(0,t) = \alpha \\ u(L,t) = \beta \end{cases} \quad , \text{ ou même } \quad \begin{cases} u(0,t) = \alpha(t) \\ u(L,t) = \beta(t) \end{cases}$$

On peut même prendre des conditions de type Neumann du genre

$$\begin{cases} \frac{\partial u}{\partial t}(0,t) = 0 \\ \frac{\partial u}{\partial t}(L,t) = 0 \end{cases} \quad , \text{ mais plutôt } \quad \begin{cases} \frac{\partial u}{\partial t}(0,t) = \alpha \\ \frac{\partial u}{\partial t}(L,t) = \beta \end{cases}$$

où encore avoir des trucs horribles avec des dépendances par rapport au temps. Mais il faut prendre en compte que garder de bonnes propriétés pour la matrice A que l'on va obtenir quand on va discrétiser.

4) On peut également considérer un problème en deux dimensions et on aura quelque chose dans ce genre là



(*Bien évidemment, mon dessin au tableau était plus dégueulasse*)
J'explique qu'en fonction du temps, la bosse "descend" et qu'ici, on a pris des conditions nulles au bord.

5) Je finis en disant que j'espère ne pas me tirer une balle dans le pied en parlant de ça, mais que l'on pourrait utiliser les éléments plutôt que les différences finies pour trouver une solution approchée.

Questions du jury :

- Jury 3 : C'est quoi le conditionnement ?
- Moi : Je note $\text{cond}(A) = |||A^{-1}|||.|||A|||$ avec une norme subordonnée. Ah oui, des fois je mets trois barres, des fois deux pour les normes matricielles ...
- Jury 2 : Oui pas de problème ! Du coup, j'aimerais revenir sur votre théorème avec le conditionnement. A quoi il sert ?
- Moi : Bah ça dit en gros que la solution perturbée est majorée par conditionnement. Donc plus, on a un grand conditionnement, moins on est sûr d'avoir des solutions perturbées proche de la vraie solution.
- Jury 3 : Mais il est bon votre théorème ?
- Moi : Euh bah oui ...
- Bah c'est quoi ce $\tilde{u}$?
- Moi : Je l'ai écrit, $\tilde{u} = u + \Delta u$, mais je peux le remplacer si vous voulez

$$\frac{||\Delta u||}{||u + \Delta u||} \leq \text{cond}(A) \frac{||\Delta A||}{||A||}$$

- Jury 3 : Hum ...
- Moi : Je peux vous le démontrer si vous voulez ?
- Jury 2 : Non non, c'est bon. Mais du coup, ré-expliquez moi à quoi il sert ?
- Moi : (*Je reformule différemment du coup*) Plus notre conditionnement est petit, plus les solutions perturbées seront proches de la solution exacte.
- Jury 2 : Ok ! Et pour quelle est la valeur minimale du conditionnement ?
- Moi : 1.
- Jury 2 : C'est atteint pour quelle(s) matrice(s) ?
- Moi : L'identité.
- Jury 2 : D'accord.

- Jury 2 : Quel est l'intérêt du coup de chercher une solution avec un problème de minimisation plutôt qu'avec une autre méthode ?
- Moi : (*Je comprends pas trop où ils veulent m'amener...*) C'est moins coûteux ...
- Jury 4 : Ah non, pas forcément ...
- Jury 2 : Non c'est pas vraiment ça. C'est plus au niveau des solutions.
- Moi : Ah ! On contrôle l'erreur et même le nombre d'étapes. Dans mon algorithme, j'ai deux conditions sur ma boucle, soit ϵ, soit N le nombre d'étapes.
- Jury 2 : Mouais ... C'est quoi l'inconvénient par rapport à la méthode de Gauss par exemple ?
- Moi : (*Je comprends pas toujours trop où ils veulent m'amener...*) Euh ...
- Jury 2 : Vous connaissez quoi comme autres méthodes ?
- Moi : En méthodes directes, on a LU, QR, ... En méthodes itératives, on a les décompositions régulières.
- Jury 2 : Bon voilà ! C'est quoi l'inconvénient des méthodes itératives par rapport aux méthodes directes au niveau des solutions ?
- Moi : Ah ! Ce sont des solutions approchées !
- Jury 2 : Oui voilà ! (*Très franchement, je n'avais pas saisi qu'ils voulaient cette réponse ...*)

- Jury 3 : A un moment, vous avez dit que l'on prenait la direction du gradient. Pourquoi ?
- Moi : Bah en fait, quand on dessine une courbe de niveau (*Je fais un dessin en même temps*), le gradient $\nabla f(x_k)$ est orthogonal en x_k à la courbe de niveau. Du coup, la façon la plus rapide de sortir de cette courbe de niveau, c'est de prendre cette direction.

- Jury 3 : D'accord.

- Jury 2 : Vous pourriez nous remontrer votre algorithme de gradient et nous dire ce qui ne va pas ?
- Moi : Oui ! Bah quand je résoud le système $Ax = b$ avec la fonction *backslash* et avec la méthode de gradient et que je calcule la norme de la différence, je trouve environ 15 ...
(*Il regarde mon algortihme et cherche l'erreur ... Et j'ajoute*)
- Moi : Pendant la préparation, j'ai fait plusieurs fois mon algorithme, en me rendant compte que j'avais pris une matrice aléatoire (*C'est un réflexe : pour avoir des tests plus convaincants*) et après j'ai pris une matrice symétrique définie positif mais ça marche toujours pas ...
- Jury 2 : (*Rigole*) Bah c'est évident en même temps ...
- Jury 3 : Mais vous prenez quelle norme ?
- Moi : Ah ! C'est peut-être ça ... Bah j'ai pris la norme par défaut de *Scilab*, et je ne sais pas si par défaut c'est la norme 2 ou la norme infini ...
- Jury 3 : Pourtant ça a l'air cohérent ce que vous avez écrit ... Ah mais c'est peut-être à cette ligne là ?
- Jury 2 : Non c'est bon ça ... Bon on va passer à autre chose.

- Jury 2 : Est-ce que vous pourriez revenir sur la modélisation de l'équation de Laplace ? Nous réexpliquer un peu comment vous la trouvez ? Parce que dans le texte il y a une aire et pas dans votre explication
- Moi : Oui oui ! Alors j'ai pas compris l'explication du texte pour la modélisation. Il y a un σ et je ne vois pas ce que c'est ...
- Jury 3 : Faites un dessin
- Moi : D'accord. *Je fais un dessin d'une barre metallique avec une surface noté σ puis je dis* Mais je ne comprends pas à quoi sert cette surface car après on travaille sur du $1D$.
- Jury 2 : Oui ça se simplifie en fait.
- Moi : Oui voilà ! Donc en faisant un bilan d'énergie entre x et $x + \delta x$, on a

$$q(u(x + \delta x) - u(x)) = f(x)$$

(*u c'était la température et q le flux*)
et en utilisant la loi de Fourier, on a bien l'équation en question.
- Jury 2 : Hum ... D'accord.

- Jury 4 : Moi j'aimerais revenir sur une phrase du texte. Il est écrit qu'il est évident de chercher une solution sous la forme $\sum_{j=1}^{+\infty} u_i \sin\left(\frac{2\pi j}{L}\right)$. Pourquoi c'est évident ? (*L était la longueur de la barre*)
- Moi : Euh ... Si on dérive deux fois cette série, on peut avoir des relations avec les coefficients u_j ...
- Jury 2 : Expliquez nous ?
- Moi : Par exemple, en considérant l'équation $u''(x) - u(x) = f(x)$, et en remplaçant par cette série, on aure des relations avec les coefficients ... Mais ici, on n'a pas $u(x)$ ... Et en plus, on sait résoudre directement ce genre d'équations ... (*Je me rends compte que je me suis enflammé et que c'est pas ça ...*) Mais pour moi, j'ai l'impression que c'est juste le fait que l'on sait résoudre des équations différentielles par les séries de Fourier.
- Jury 4 : Non mais c'est intrinsèque au problème ...
- Moi : Euh ... J'ai pas compris votre question ... Mais j'ai compris le mot intrinsèque.
- Jury 4 : (*Il sourit*) Non mais ça vous pensez à quoi ce genre de série ?
- Moi : Bah à une série de Fourier ou trigonométrique.
- Jury 4 : Oui, voilà ! Et ?
- Moi : Euh ... Très franchement, je ne vois pas ... C'est sûrement être évident mais je ne vois pas ...

- Jury 3 : On peut revenir sur votre algorithme ? Expliquez nous ce qui ne marche pas.
- Moi : Bah quand je trace l'erreur en échelle $log - log$, la pente n'est pas 2 alors qu'il faut trouver 2 ...
- Jury 3 : C'est quoi l'erreur ? Ecrivez le nous.
- Moi : $||e_j|| = ||u_j - u(x_j)||$, et je sais que pour les puristes, il y a un sens dans la soustraction mais je ne

retiens jamais. (*Jury 4 rigole*)
- Jury 3 : Et du coup, votre graphique c'est quoi en fonction de quoi ?
- Moi : C'est la norme de l'erreur en fonction de la discrétisation, les valeurs 10, 25, 50, 100 qui étaient entrées au début, en échelle $log - log$.
- Jury 3 : Et du coup, c'est quoi ce -0.458 ?
- Moi : Alors, c'est la pente que je récupère avec la commande *reglin*. Le moins n'est pas important.
- Jury 3 : Vous pouvez écrire une relation entre l'erreur et la discrétisation ?
- Moi : Alors je crois que c'est ça $||e_J|| \leq C^{2J}||e_0||$ (*Quand je passe au log, je vois qu'il y aura un problème ...*)
- Jury 3 : Partez directement de la formule avec le *log* et dites nous ce que vous vouliez ?
- Moi : Alors j'aimerais

$$\log(||e_J||) = a\log(N) + b$$

... Ah mais oui, c'est pas J que je dois mettre, c'est N. En fait, c'est pas logique sinon ...

$$\log(||e_N||) = a\log(N) + b$$

Et je devrais avoir $a = 2$ mais c'est pas le cas ... Pourtant, je ne comprends pas parce que quand je trace la solution exacte avec la solution du schéma numérique, ça marche. C'est très proche.
- Jury 2 : Mais du coup, c'est quoi votre deuxième droite ?
- Moi : C'est la droite qui est donné par $\log(N)$ en abscisse $a\log(N)$ en ordonnée.
- Jury 2 : D'accord.

Jury 3 : Je voulais juste revenir sur le minimum de la fonctionnelle. Pourquoi il existe ?
- Moi : Parce que f est convexe ...
- Jury 3 : D'accord.
(*J'ai hésité à dire "coervice" en plus et je pense que j'aurais dû car c'est faux sinon ! La fonction $x \to e^x$ est convexe mais n'a pas de minimum. Mais Jury 3 avait l'air d'acquiescer ...*)

- Jury 2 : Vous avez parlé d'éléments finis, vous pourriez expliquer ce que c'est ? Il reste deux minutes ...
- Moi : Euh oui bien sûr. Alors, on transforme notre équation différentielle en un problème dans un espace de Hilbert. C'était la bonne idée de Sobolev de ne plus considérer les éléments mais de changer d'espace. On va avoir un problème de la forme

$$a(u, v) = \mathcal{L}(v)$$

On obtient ça en réalisant ce qu'on appelle la formulation variationnelle de l'équation de départ. On prend une fonction v dans L^2 par exemple, et on aura

$$a(u, v) = \int \langle \nabla u, \nabla v \rangle \quad \text{et} \quad \mathcal{L}(v) = \int \nabla v f$$

On sait qu'il y a des solutions dans des espaces de Hilbert que je note H, par exemple, les espaces de Sobolev $\mathcal{H}^1$ et $\mathcal{H}^1_0$. On utilise le théorème de Riesz, et si on n'a pas de symétrie pour a, on utilise Lax-Milgram, à condition d'avoir de la coercivité.
Après l'idée, ça va être de considéré un sous-espace U_h de H, et on va chercher u_h dans U_h tel que

$$a(u_h, v) = \mathcal{L}(v) \quad \text{pour tout } v \in U_h$$

Après le lemme de Céa nous donnera des majorations entre u et u_h en fonction de la distance au sous-espace.
- Jury 2 : Pourquoi on cherche dans U_h ?
- Moi : Parce que c'est plus facile ?
- Jury 2 : Pourquoi ? Il a quoi comme particularité ?
- Moi : Ah ! Il est de dimension fini !

- Jury 2 : Et en quoi c'est bien ?
- Moi : Bah on va pouvoir prendre une base et décomposer nos éléments dessus et du coup pour les algorithmes, c'est mieux.
- Jury 2 : D'accord ! Bon c'est fini.

Ressenti :

Jury très agréable, souriant et très sympathique. Jury 1 n'a rien dit ... Peut-être était-il empaillé ...
J'ai juste peur d'avoir fait un hors-sujet par rapport aux mots-clefs "Optimisation".
J'ai pas très bien écrit et c'était un peu désorganisé sur la fin car j'ai parlé de plusieurs choses et ils ne voulaient pas trop que j'efface.
Un peu déçu que mes algorithmes n'aient pas donné de bons résultats, mais le jury a regardé avec moi les algorithmes et m'a dit qu'ils étaient cohérents pourtant et ne voyaient pas où pouvait être les erreurs ...
Je suis plutôt content d'avoir fini sur les éléments finis et d'avoir su répondre à la question.

Note :	**6.75/20**

Commentaire du jury :
Modélisation (option B), 27/80. Si le candidat possède des connaissances, elles ne sont pas exploitées au service du texte. La modélisation est escamotée et mal expliquée.