Mathématiques 2

Présentation du sujet

L'épreuve consiste en un exercice unique. Le candidat dispose d'une demi-heure de préparation pendant laquelle il a un accès libre à Maple ou Mathématica. Pendant la demi-heure suivante, les résultats obtenus sur ordinateur sont discutés, tandis que la résolution des questions théoriques se fait au tableau. Signalons qu'il n'est nullement nécessaire de résoudre l'exercice en totalité pour obtenir une note excellente.

L'usage du logiciel est une étape imposée, et une question est systématiquement placée vers le début de l'énoncé à cet effet. Le candidat peut cependant librement choisir de l'utiliser même lorsque l'énoncé n'y fait pas explicitement référence. Le refus d'utiliser le logiciel, ou l'absence de résultats pertinents expose le candidat à une note décevante, quand bien même l'exercice serait résolu entièrement. En particulier, une procédure qui ne fonctionne pas n'a pas la moindre valeur.

Outre les connaissances théoriques et la maitrise du logiciel, l'examinateur prend en compte dans son évaluation l'autonomie et la communication du candidat. Mentionnons pour finir qu'il est le seul à décider du déroulement de l'oral.

Analyse globale des résultats

Le jury a pu constater comme les années précédentes qu'une majorité de candidats se trouve à l'aise dans l'usage du logiciel de calcul formel. Lorsque la préparation a permis la conjecture d'une ou plusieurs propriétés, et que celle-ci se conjugue avec une avancée significative dans l'exercice, que ce soit de manière autonome ou avec une bonne réactivité aux indications, les examinateurs ont étés ravis de mettre d'excellentes notes. De telles prestations ont eu lieu fréquemment.

Signalons cependant qu'une faible proportion de candidats reste réfractaire à l'usage du logiciel. Notons également que bon nombre de candidats, sans être étrangers aux outils proposés, ne présentent guère de combativité sur la machine, prennent le moindre message d'erreur du logiciel pour une fatalité inéluctable, et ne comptent alors que sur le bon vouloir de l'examinateur pour avancer, relayant ce dernier à la tâche ingrate de debuggeur informatique. Il est donc attendu d'un candidat qu'il teste et corrige lui même ses commandes, et ne présente pas un demi-travail inachevé.

Pour le reste, si les connaissances théoriques sont globalement satisfaisantes, le jury a constaté une grande disparité en ce qui concerne la rigueur. « Il suffit d'appliquer tel théorème » est un raccourci auquel nous avons encore trop souvent droit. Le nom d'un théorème ne suffit pas pour répondre à une question, il faut aussi détailler toutes les hypothèses et leur vérification. Pour finir, la qualité de la communication reste très variable : un certain nombre de candidats se contente toujours de leur tableau comme seul interlocuteur, ce qui est très pénalisant en terme d'évaluation. Rappelons qu'une épreuve orale reste un échange et qu'il convient de s'exprimer le plus souvent de vive voix en regardant son examinateur, le tableau servant de support essentiellement pour les détails techniques.

Commentaires et conseils aux candidats

À propos du logiciel de calcul formel

Comme il est dit plus haut, on constate avec satisfaction que beaucoup de candidats sont vraiment bien familiarisés avec le logiciel de calcul formel. *A contrario*, on reconnait très rapidement ceux qui en sont restés à quelques lointains souvenirs de première année. On a vu par exemple que certains ne savent même pas calculer un produit matriciel avec le logiciel, et d'autres qui ouvrent simultanément les deux packages **linalg** et **LinearAlgebra** (de Maple) montrant ainsi leur méconnaissance de l'un et l'autre et mélangeant joyeusement les commandes au gré de leur lecture de l'aide en ligne ! Passons en revue quelques points qui posent souvent problème.

Un abus trop fréquent de l'usage de procédures

Il est parfois demandé explicitement d'écrire une procédure (toujours courte et simple) parce que l'objet créé sera manipulé dans d'autres questions au cours de l'épreuve. Mais hors cette situation, il faut savoir que c'est rarement une nécessité. Pourtant beaucoup de candidats se réfugient par un réflexe qui semble incontournable, dans l'usage immodéré de procédures en cascade ce qui conduit, force est de le constater, à un château branlant qui ne produit aucun résultat au final (exemple extrême mais rencontré : un candidat a écrit une procédure pour créer la fonction factorielle). L'écriture d'une simple boucle, d'une fonction avec des commandes données directement par le logiciel conduit pourtant immédiatement au résultat. De la même façon que le candidat a perdu trop de temps à écrire ses procédures pendant la préparation, on perd de nouveau trop de temps lors de la partie orale de l'épreuve à rechercher les erreurs de programmation et on doit souvent passer à la suite des questions proposées. Les candidats qui souhaitent écrire des procédures récursives, souvent très bien adaptées en effet, doivent savoir maitriser leur syntaxe. En résumé, cette épreuve n'est pas une épreuve d'algorithmique et de programmation, mais d'utilisation d'un logiciel de calcul formel.

Fonctions et expressions

Il est essentiel de savoir distinguer fonctions et expressions, savoir créer et manipuler les séquences, listes et ensembles. Le recours à ces objets est constant, et ne pas savoir les construire ou les reconnaitre est un handicap difficilement surmontable. Un candidat qui, dès le départ, ne sait pas fabriquer une fonction ou une séquence (situation hélas encore trop vue cette année), aura beau se réfugier derrière le camouflage de l'écriture d'une procédure (cf ci-dessus), il échouera à obtenir le moindre résultat. Il est souvent demandé d'afficher des suites de résultats numériques (les 20 premiers termes d'une suite récurrente, la valeur du déterminant d'une matrice A_n pour n de 2 à 10, le tracé des 10 premières fonctions d'une suite $(f_n)_{n\in\mathbb{N}}$). L'usage des commandes **seq** (Maple) / **Table** (Mathematica) est pratique et doit être un réflexe. Avec une fonction ou une expression, il est attendu que le candidat sache évaluer cet objet. Il faut trop souvent proposer de faire une substitution pour évaluer une expression ! Nous conseillons par exemple de s'entrainer à écrire le code pour calculer les 10 premiers termes d'une suite de fonctions construite par récurrence : par exemple à partir d'une fonction initiale f et d'une fonction de deux variables K librement choisies

$$f_0 = f \text{ et } \forall n \in \mathbb{N},\ f_{n+1}(x) = \int_0^1 f_n(t)K(x,t)dt$$

ce qui n'est pas sans quelques pièges de programmation.

Matrices et changement de bases

Lorsqu'un énoncé demande de travailler de taille inférieure ou égale à 6, il n'est pas rare de voir des candidats remplir les 36 coefficients les uns après les autres. Il faut absolument maitriser les techniques qui permettent de s'épargner une tâche aussi fastidieuse. Notamment, il est suggéré de connaitre la syntaxe permettant de définir une matrice A à partir de l'application $(i,j) \to A_{i,j}$. De la même manière, il est important de savoir construire une matrice de passage à partir d'une famille de vecteurs pour s'épargner une recopie de coefficients.

Résolution d'équations numériques, d'équations différentielles

Il faut savoir que les commandes de résolution s'appliquent à des équations (différentielles) scalaires ou des systèmes d'équations (différentielles) scalaires. Ainsi en algèbre linéaire, si on apprécie que beaucoup de candidats savent désormais entrer une matrice avec une fonction définissante des coefficients, trop espèrent par exemple qu'en demandant : **solve**$(AX - XA = 0, X)$, le logiciel permettra de déterminer toutes les matrices qui commutent avec la matrice A. Une autre nécessité après l'usage d'une commande **solve / NSolve**, ou **dsolve / DSolve** est de savoir affecter les résultats à l'objet inconnu (un polynôme, une matrice ... dont le logiciel aura justement obtenu les coefficients, ou la solution d'une équation différentielle) car on veut pouvoir manipuler cette solution pour la suite de l'exercice. Il faut aussi bien lire l'énoncé et ne pas demander un résultat exact quand on demande des valeurs approchées **(fsolve / NSolve)**, savoir préciser un intervalle de résolution, être vigilant si le logiciel renvoie une seule valeur pour ne pas affirmer catégoriquement que l'équation proposée n'admet qu'une seule solution. Pour un exercice étudiant une équation différentielle, il est rare d'espérer obtenir (dans les exercices proposés) la solution demandée avec les fonctions usuelles du programme ; l'usage des commandes **dsolve(. . ., numeric) / NDSolve** permet d'obtenir des valeurs numériques de la solution recherchée, ou de faire un tracé (cf ci-dessous).

Tracés

Il faut savoir tracer des objets (courbes ou surfaces) paramétrés ou définis implicitement ; savoir superposer plusieurs graphes de même nature ou de natures différentes. Pour les équations différentielles, il est indispensable de savoir tracer une solution avec conditions initiales (par exemple avec Maple, en choisissant entre les commandes **DEplot** (de **DEtools**), ou **odeplot** si on a utilisé un **dsolve(. . ., numeric)**). Notons en géométrie que l'idée de pouvoir mener des calculs techniques, bien pratiques avec le logiciel, par exemple pour déterminer l'intersection de deux droites, ou d'une droite et d'une conique, rebute la plupart des candidats qui préfèrent mener les calculs à la main.

Nous reproduisons comme les années passées, et pour faciliter la bonne préparation à cette épreuve, une liste placée en annexe, des savoir-faire qui sont régulièrement utilisés, et dont on attend que les commandes usuelles nécessaires pour leur mise en œuvre soient connues.

À propos des connaissances mathématiques

Algèbre et géométrie

Plusieurs exercices sont proposés chaque année en arithmétique sur $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$. Ces exercices ont assez rarement donné lieu à de bonnes prestations, il convient donc de se préparer davantage sur ce chapitre. La notion de groupe cyclique, la détermination des inversibles de l'anneau $\mathbb{Z}/n\mathbb{Z}$ ne sont pas toujours bien connues. L'algèbre linéaire est naturellement abordée par le point de vue « matriciel » avec le logiciel de calcul formel. Par contre il est en général nécessaire de passer au point de vue « vectoriel » avec les applications linéaires pour aborder les questions théoriques. Cela éviterait d'entendre trop souvent la formulation : « dans la nouvelle base, la matrice A s'écrit . . . ».

à une demande de précision sur les formules de changement de bases, on reçoit souvent une réponse erronée. Dans la « formule de la comatrice », beaucoup ne voient qu'un outil pour expliciter une matrice inverse, mais en oublient la validité générale, ainsi que le caractère permutable d'une matrice carrée et de la transposée de sa comatrice. Si le « théorème de décomposition des noyaux » est désormais assez bien repéré, l'usage des restrictions à des sous-espaces stables n'est pas assez naturel ; et les justifications de stabilité sont souvent laborieuses ; il n'est pas inutile de disposer dans sa tête d'une liste de résultats élémentaires du cours concernant les sous-espaces stables. Les prestations en algèbre euclidienne sont décevantes. Le passage du « théorique » au « pratique » avec le logiciel s'avère souvent difficile. Comme tous les ans, il faut supplier pour obtenir un énoncé complet sur la réduction des endomorphismes autoadjoints ; on oublie de préciser qu'il existe une base orthonormée de vecteurs propres et il n'est jamais répondu que l'espace est somme directe orthogonale des sous-espaces propres. L'usage de la stabilité par l'adjoint, de l'orthogonal d'un sous-espace stable, est rarement vu (et parfois le théorème est mal connu). De même pour une situation exploitant le théorème de distance à un sous-espace de dimension finie. En géométrie, l'utilisation de droites dans l'espace (définies par un paramétrage ou un couple d'équations cartésiennes), et parfois même dans le plan, des questions élémentaires sur des coniques et des quadriques restent un point d'écueil anormal. Effectuer un changement de repère orthonormé dans R3 , avec ou sans l'aide du logiciel, semble à certains bien redoutable ! Pour des courbes usuelles, les candidats n'ont parfois pas le réflexe d'introduire eux-même un paramétrage judicieux. Par la suite, l'écriture d'une équation de tangente ou de normale à une courbe plane est une question qui pose d'énormes difficultés. Pour une surface définie par une équation cartésienne, la notion de point singulier est parfois méconnue et donc, le rôle du vecteur gradient.

Analyse et géométrie différentielle

Le cas des suites récurrentes reste de façon surprenante et comme les années précédentes, délicat, même pour une récurrence usuelle $u_{n+1} = f(u_n)$ dans $\mathbb{R}$. Outre qu'une visualisation schématique, au tableau, du comportement dans ce cas usuel n'est pas naturelle, la lecture de résultats suggérés par le logiciel n'est pas bien exploitée : monotonie, deviner une majoration facile de u_n. La recherche et la manipulation d'équivalents ou de développements asymptotiques est laborieuse. On a vu apparaitre l'idée inquiétante qu'un résultat asymptotique se traduit par une égalité à partir d'un certain rang, par exemple :

$$u_n \sim \frac{1}{n} \text{ implique } u_n = \frac{1}{n} \text{ « pour } n \text{ assez grand »}$$

Concernant l'intégration (intégrabilité et intégrales impropres, intégrales à paramètre, intégration des suites et des séries de fonctions), ce n'est pas forcément la question de la domination qui est la moins bien traitée, mais, dès le départ, celle de l'existence des intégrales qui est mal abordée ; la suite de l'exercice s'en trouve rapidement faussée (faute dans le domaine de définition, ne pas repérer une intégrale convergente d'une fonction non intégrable). Pour l'intégration des séries de fonctions, il y a trop d'imprécision dans le choix du théorème utilisé (qui se résume souvent par l'incantation « par convergence dominée »). La formule de Taylor avec reste intégral qui a été nécessaire pour certains exercices nécessite presque systématiquement deux à trois corrections. L'importance de la convergence normale pour assurer une convergence uniforme est mal mise en valeur, et les majorations sont imprécises, la faute classique étant d'obtenir un équivalent du terme général $u_n(x)$ qui par bonheur fait disparaitre le paramètre x et permet, croit-on, de conclure ! Avec les séries de Fourier, les hypothèses précises des théorèmes sont trop mal acquises, les dits-théorèmes étant souvent inversés (convergence normale et Dirichlet). L'usage du logiciel pour un calcul de coefficients de Fourier est classique et pourtant rarement satisfaisant ; par exemple, trop de candidats ne cherchent pas à simplifier les $\cos(n\pi)$ ou $\sin(n\pi)$ renvoyés et n'hésitent pas à les conserver au tableau et ne savent d'ailleurs pas comment simplifier cela avec le logiciel ! Pour les séries entières, l'énoncé demande souvent l'aide du logiciel pour avoir une idée de l'ordre de grandeur des coefficients a_n, ou a_n/n !, mais rares sont pourtant ceux qui tirent une conclusion satisfaisante de l'observation du calcul approché de termes de ces suites ! L'affirmation qu'une série entière converge uniformément sur le disque ouvert de convergence résiste au fil des années.

Il semble pour les candidats, que le cas des équations différentielles linéaires soit trop simple pour mériter aucune justification ; même si l'énoncé ne pose pas de question explicite, on attend que le choix de l'intervalle de résolution, la dimension de l'espace des solutions soient indiqués directement en abordant ces questions. Concernant les équations différentielles non linéaires, l'énoncé du théorème de Cauchy-Lispchitz est rarement convenable, les justifications se limitant dans le meilleur des cas au caractère C^1 de la première fonction qui passe. L'ordre 2 notamment nécessite une application soignée qui n'est jamais faite. La détermination de l'intervalle de définition d'une solution maximale est certes souvent délicate, mais le jury attend quand même que des raisonnements classiques soient mis en œuvre. Il faut notamment savoir justifier l'existence d'une limite en tout point réel d'une fonction C^1 à dérivée bornée. Lorsque ce résultat est su, les candidats ont alors trop souvent tendance à se contenter de prolonger la fonction, sans conclure proprement.

À propos de l'attitude générale des candidats

À connaissances équivalentes, il va de soi que la préférence du jury ira vers un candidat dynamique et réactif plutôt que vers un candidat taciturne qui ne daigne pas lever les yeux de son tableau et ne suit pas les indications. Nous dégageons ici quelques idées moins évidentes à retenir.

- L'initiative et l'autonomie sont des vertus essentielles de l'oral. Il ne faut donc pas attendre l'approbation de l'examinateur pour explorer une piste (celui-ci interviendra de son propre chef si la piste n'est pas bonne, ou s'il attend des précisions). Dans le même registre, les expressions de la forme « il n'y a qu'à » ou « il suffit de » sont à proscrire de l'oral ! Il faut donner **par défaut** les précisions (notamment les hypothèses des théorèmes, et toutes leurs vérifications, les calculs, etc ...).
- Il arrive parfois qu'un candidat propose de sauter une question, le plus souvent en fin d'oral lorsqu'il se retrouve bloqué sur une difficulté et souhaite montrer et discuter de ce qu'il a traité dans la suite. Rappelons que c'est l'examinateur qui décide seul du déroulement de l'oral et notamment qui prend la liberté d'insister sur un blocage (pour évaluer les lacunes du candidat, ou sa réactivité aux indications) ou au contraire de passer à la suite. Les tentatives pour détourner les difficultés ou masquer une méconnaissance du logiciel ne peuvent que desservir les candidats.
- Le tableau est un outil essentiel de l'oral. Il ne doit pas s'agir d'un brouillon (nombre de candidats écrivent dans tous les sens possibles !). Il ne doit pas s'agir non plus d'une copie. Il est en revanche apprécié que les éléments essentiels de logique s'y retrouvent (introduction des variables, symboles d'implications ou d'équivalence). Par ailleurs, il serait bienvenu de penser à ne pas se tenir entre son texte et l'examinateur (qui n'a pas la faculté de lire à travers les candidats).

Conclusions

Les remarques précédentes ne doivent pas occulter le fait qu'une majeure partie des candidats a bien compris la nécessité d'une préparation régulière de l'épreuve et que de nombreuses prestations ont donné lieu à d'excellentes notes. Nous nous permettons néanmoins d'insister sur le fait que l'épreuve reste une épreuve de mathématique, pas de programmation. À ce titre, la préparation doit se focaliser sur les outils fournis par le logiciel, dont le maniement doit être travaillé tout au long de l'année. L'acquisition d'une réelle technicité est un atout essentiel, notamment pour éviter de perdre du temps pendant la préparation.

Annexe : listes des compétences attendues pour l'épreuve de calcul formel

Les compétences ci-dessous forment une liste relativement exhaustive des connaissances du candidat idéal. On remarquera qu'il n'est nulle part mentionné la notion de procédure car leur usage n'est pas imposé.

Calcul algébrique (entiers, polynômes, équations) **:**

- savoir calculer le quotient, le reste dans une division euclidienne dans $\mathbb{Z}$, dans $\mathbb{Q}[X]$;
- savoir travailler « modulo n » ;
- savoir factoriser (dans $\mathbb{Q}[X]$ et éventuellement dans une extension simple suggérée par l'énoncé), développer, ordonner un polynôme ;
- savoir obtenir tous les coefficients, ou des coefficients précis d'un polynôme ;
- savoir calculer le pgcd de deux entiers, de deux polynômes ;
- savoir obtenir un couple donnant la relation de Bézout ;
- savoir déterminer les racines d'une équation (algébrique ou non) de façon exacte, de façon approchée ;
- savoir déterminer une valeur approchée d'une racine localisée dans un intervalle ;
- savoir décomposer une fraction rationnelle en éléments simples dans $\mathbb{Q}(X)$ (éventuellement dans une extension simple de $\mathbb{Q}$ suggérée par l'énoncé).

Calcul matriciel :

- savoir construire une matrice dont les coefficients sont donnés par une formule fonction du couple (i, j), et dont la taille peut être variable (il ne peut être question de se limiter à savoir entrer une matrice de taille 3 par ses neuf coefficients) ;
- savoir calculer des produits matriciels, créer une matrice diagonale et *a fortiori* la matrice identité, former la transposée ;
- savoir calculer le rang, le noyau ou l'image d'une matrice (en obtenant une base de ces sous-espaces), savoir effectuer un changement de bases (en évitant de recopier à la main la matrice de passage) ;
- savoir calculer le déterminant, éventuellement l'inverse, la comatrice (ou sa transposée) d'une matrice carrée ;
- savoir calculer le polynôme caractéristique d'une matrice carrée, ses valeurs propres, ses vecteurs propres (de manière exacte ou approchée) ;

- savoir résoudre une équation d'inconnue matricielle (après l'avoir transformée en un ensemble d'équations scalaires d'inconnues les coefficients) ;
- savoir calculer le produit scalaire, le produit vectoriel de deux vecteurs de $\mathbb{R}^3$.

Fonctions d'une ou plusieurs variables réelles, calcul différentiel, calcul intégral :

- comprendre la différence fondamentale entre fonctions et expressions ;
- savoir composer des fonctions (ou des opérateurs), calculer des dérivées d'ordre supérieur à un ;
- savoir calculer un développement limité, savoir extraire la partie régulière d'un tel développement ;
- savoir calculer, sous la forme d'une fonction ou d'une expression un terme arbitraire d'une suite de fonctions définies par une relation de récurrence ou une formule.