

★ **Buts et contexte de l'épreuve**

1. Préparation de 4h, exposé de 35 minutes et 25 minutes de questions
2. Exposé vivant, synthétique et pédagogique : doit être accessible à quelqu'un qui ne connaît pas le texte (mais le jury aura une version papier, donc pas besoin de rappeler les notations).
3. Dosage entre :
 - a. Introduction motivée du/des modèle(s)
 - b. Preuves mathématiques (complétion d'arguments lacunaires du texte par exemple, avec même précision que dans un développement !)
 - c. Exposition des connaissances
 - d. Retour sur le modèle du texte et prise de recul : étude du comportement qualitatif, dépendance en des paramètres
 - e. Illustrations informatiques (ou essais expliqués) et, si possibles, graphiques. Conséquences des variations de paramètres. **Attention** : ce n'est pas une épreuve de programmation !
 - f. Réponse aux questions du jury
4. Pas de connaissances autres que mathématiques exigées
5. Conserver la rigueur mathématique
6. Savoir utiliser les théorèmes du programme **avec** des hypothèses valables dans le contexte (pas forcément minimales)
7. Structurer son exposé avec un plan (annoncé)
8. Éviter la paraphrase ! (c'est inutile car facilement repéré et lourdement sanctionné)
9. **Ne pas** recycler un développement de leçon : c'est **hors-sujet** direct !
10. Pas seulement introduction à la problématique avant une longue suite de définitions et théorèmes sans lien avec le problème de départ
11. Pas non plus que de simples illustrations informatiques et/ou énumération linéaire des pistes proposées sans contribution substantielle
12. Une des principales difficultés : maîtriser des notions "simples" en situation, pas forcément besoin de virtuosité technique
13. Évaluation sur : rigueur, clarté, gestion du temps, pertinence et cohérence des choix, pédagogie, synthèse (mieux vaut exhiber les étapes cruciales d'un raisonnement que de faire un long calcul pénible et dur à mettre en valeur) ; mais **aussi** : repérer et éventuellement compléter les lacunes (volontaires) du texte

★ **Pendant la préparation**

1. Ne pas écrire sur le texte distribué (il sera réutilisé)
2. Penser au plan et à l'exploitation du tableau
3. Laisser du temps pour préparer les illustrations informatiques
4. Songer à la gestion du temps ; en particulier l'informatique doit être présentée **durant** l'exposé (pas pendant les questions)

★ Pendant l'oral

1. Le jury est censé être bienveillant : il peut aider dans la gestion du temps et donner des indications en cas de blocage. Mais il faut quand-même avoir un peu de répondant.
2. Une certaine "aisance" est attendue (mais le jury est conscient du stress de l'épreuve)
3. Bien gérer son temps ; le jury prévient 10 minutes avant la fin de l'exposé si l'informatique n'a toujours pas été présentée.
4. Pire qu'un exposé trop bref : un exposé qui "joue la montre" en allant délibérément trop lentement.
5. Alternier entre :
 - a. Explications orales
 - b. Rédaction (claire et lisible) au tableau (demander avant d'effacer)
 - c. Illustrations informatiques
6. On peut traiter seulement une partie du texte
7. Comme pour les autres oraux : ne parler que de ce qu'on maîtrise, slogan : "Tout ce que vous direz pourra et sera utilisé contre vous"
8. Pas d'argument d'autorité ("le texte le dit donc c'est vrai")
9. Ne pas se censurer : autocritique appréciée ("je n'ai pas réussi à prouver que...", "les hypothèses du théorème que je veux utiliser ne sont pas valables dans ce cadre", etc)
10. Bien identifier les points importants dans une preuve et les trous ; ne pas se contenter de réécrire la preuve du texte !
11. Faire des dessins et schémas explicatifs !
12. Le retour sur le problème de départ et la critique (justifiée) du modèle mathématique sont appréciés
13. Possible de s'éloigner du découpage du texte et de réarranger les éléments autrement
14. Plans génériques possibles :
 - a. Survol du modèle, de la méthode et des résultats pour détailler des aspects plus mathématiques (toujours avec illustrations)
 - b. Suivre la progression du texte avec détails et illustrations
 - c. Aller loin dans le texte pour prendre du recul et en décrire l'organisation globale, l'aboutissement et les résultats (donc ne pas traiter toutes les preuves)
 - d. Au contraire : se pencher sur un passage restreint de façon très détaillée et en complétant tous les arguments
 - e. **Dans tous les cas**, il faut des illustrations et des preuves (basées sur le texte, par exemple montrer la convergence/stabilité/consistance d'un schéma ; pas de cours !)

★ Sur l'informatique

1. Possibilité de faire des programmes simples utilisant les routines des logiciels employés, mais il vaut mieux les connaître un peu (ex : la commande `ode` de **Scilab** utilise par défaut des méthodes de prédiction-correction d'Adams, sauf pour les problèmes raides où des méthodes implicites sont exploitées ; on peut aussi lui demander d'utiliser RK4...)
2. Analyse sommaire de la complexité des algorithmes (graphiques)
3. Une analyse de l'ordre numérique des méthodes (régression linéaire) peut être demandée
4. Certaines illustrations doivent mener à un retour sur la problématique et le modèle
5. Commenter les résultats obtenus par rapport au modèle du texte
6. Possible de présenter son code, mais insister sur les points-clefs et non tout détailler ligne à ligne (temps)
7. Légender (au-moins à l'oral) les graphiques !
8. Ne pas censurer les programmes qui ne marchent pas (le jury peut aider)