# ANALYSE NUMÉRIQUE

## I

1° On se donne, pour toute la suite, sous le nom de programme :

*a.* Un ensemble fini $\mathbf{V} = \{ V_1, V_2, \ldots, V_u \}$ de symboles $V_i$ $(i = 1, 2, \ldots, u)$.

Ces symboles sont les noms des *mémoires :* une mémoire est une case dans laquelle on admettra pouvoir ranger un nombre rationnel quelconque $x \in Q$;

*b.* Un ensemble fini $\mathbf{S} = \{ S_1, S_2, \ldots, S_r \}$ de symboles $S_j$ $(j = 1, 2, \ldots, r)$. Ces symboles servent à désigner les *instructions* (pas élémentaires du calcul);

*c.* Un ensemble d'instructions :

2° Une *instruction*, désignée par $S_j$, sera de l'un des quatre types suivants :

*Type 1* (instruction arithmétique) :

$$S_j : V_{i_3} := V_{i_1} \tau V_{i_2} ; m$$

où

$\tau$ est l'un des 4 opérateurs arithmétiques :

$+$ addition dans Q
$-$ soustraction dans Q
$\times$ multiplication dans Q
$\div$ division dans Q,

$j$ et $m$ sont des entiers tels que $j + m$ et $j$ soient entre 1 et $r$ ($m$ peut être négatif),
$i_1$ $i_2$ $i_3$ sont des entiers entre 1 et $u$.

L'instruction $S_j$ s'exécute alors, par définition, de la manière suivante :

On considère le contenu (nombre rationnel) des mémoires $V_{i_1}$, $V_{i_2}$; on compose ces contenus selon l'opération $\tau$; on range le résultat dans la mémoire $V_{i_3}$; on passe alors, pour l'exécuter, à l'instruction $S_{j+m}$.

*Type 2* (affectation d'une valeur rationnelle) :

$$S_j : V_{i_1} := x; m \qquad (j, j + m \text{ entiers entre 1 et } r)$$
$$(i_1 \text{ entier entre 1 et } u).$$

On place dans la mémoire de nom $V_{i_1}$ le nombre $x \in Q$ et l'on passe à l'instruction $S_{j+m}$.

*Type 3* (test de signe) :

$$S_j : V_{i_1}; m_1, m_2.$$

On compare à $0 \in Q$ le contenu de $V_{i_1}$; si celui-ci est plus grand que 0, la suite du programme est prise en $S_{j+m_1}$; si celui-ci est inférieur ou égal à zéro, la suite est en $S_{j+m_2}$ ($j + m_1$ et $j + m_2$ entre 1 et $r$).

*Type 4* (Arrêt) :

$$S_j : \text{Arrêt.}$$

Cette instruction signifie que le calcul est terminé : on ne modifie le contenu d'aucune mémoire, et on ne passe pas à une autre instruction.

On définira *un programme* en écrivant une liste finie d'instructions, dans l'ordre croissant des indices des $S_j$ (ce qui ne signifie pas qu'il s'exécutera dans cet ordre). La première instruction sera toujours désignée par $S_1$.

*Q 1. On demande ce qu'exécute, d'après les règles précédentes, le programme suivant, sur le contenu x de* $V_1$ :

$S_1 : V_1 ; 3, 1$
$S_2 : V_2 : = -1 ; 1$
$S_3 : V_1 : = V_1 \times V_2 ; 1$
$S_4$ : *Arrêt.*

*Q 2. Si* $\varepsilon$ *est un rationnel positif donné, supposé placé avant l'exécution dans* $V_1$, *on demande si le programme suivant s'arrête, et que contient* $V_1$ *à l'arrêt.*

$S_1 : V_2 : = 0 ; 1$
$S_2 : V_3 : = 2 ; 1$
$S_3 : V_4 : = V_1 + V_2 ; 1$
$S_4 : V_5 : = V_1 \times V_1 ; 1$
$S_5 : V_6 : = V_5 - V_3 ; 1$
$S_6 : V_5 ; 2, 1$
$S_7 : V_1 : = V_1 + V_4 ; - 3$
$S_8$ : *Arrêt.*

3° Soit P un programme utilisant $\{ V_1 \ldots V_u \}$ comme ensemble de noms de mémoires. Pour $n \leqslant u$, $P(a_1, \ldots, a_n)$ désignera, si P s'arrête, le contenu de $V_1$ à l'arrêt de P, lorsque initialement le contenu de $V_i$ est $a_i$ pour $i = 1, \ldots, n$ et nul pour $i = n + 1, \ldots, u$. Si P ne s'arrête pas, $P(a_1, \ldots, a_n)$ n'est pas défini.

On dit qu'une application $f$ d'une partie $\Omega$ de $Q^n$ dans Q *est programmable dans* $\Omega$ si :

1. Il existe un programme P pour lequel $P(a_1, \ldots, a_n)$ est défini si et seulement si $(a_1, \ldots, a_n) \in \Omega$;

2. $P(a_1, \ldots, a_n) = f(a_1, \ldots, a_n)$.

*Q 3. L'application* $f : x \in Q \longmapsto | x | \in Q$ *est-elle programmable?*

*Q 4. Les applications* $f_i : (a, b) \in Q^2 \to a \; \tau_i \; b \in Q$, *pour* $\tau_1 = +$, $\tau_2 = -$, $\tau_3 = \times$, *sont-elles programmables ainsi que*

$$f_4 : (a, b) \in Q \times Q^* \to a \div b ? \qquad (Q^* = Q - \{0\}).$$

4° Prouver le théorème suivant :

*Q 5. Soient* $f(a_1, a_2, \ldots, a_n)$ *et* $g_i(b_1, b_2, \ldots b_m)$ $(i = 1, \ldots, n)$ *des applications programmables; l'application composée* $h(b_1, \ldots, b_m) = f[g_1(b_1, \ldots, b_m), g_2(b_1, \ldots, b_m), \ldots, g_n(b_1, \ldots, b_m)]$ *est programmable dans la partie de* $Q^m$ *où elle est définie.*

*Comme corollaire, en déduire que, si* $f(a_1, \ldots, a_n)$ *et* $g(a_1, \ldots, a_n)$ *sont programmables, alors :*

$$h(a_1, \ldots, a_n) = f(a_1, \ldots, a_n) \; \tau \; g(a_1, \ldots, a_n) \qquad (\tau = +, -, \times, \div)$$

*est programmable, h étant définie dans la partie de* $Q^n$ *où f et g sont simultanément définies (et, pour* $\tau = \div$, *telles que g ne prenne pas la valeur nulle).*

*Par exemple, f définie sur* $Q^2$ *par* $(a, b) \in Q^2 \longmapsto Max\,(a, b) \in Q$ *est-elle programmable?*

## II

$Q^+$ désigne l'ensemble des rationnels strictement positifs ($> 0$).

1° Une application $\alpha$ de $Q^+$ dans Q est appelée *processus calculable* si :

*a.* elle est programmable;

*b.* $\forall\, \varepsilon_1, \varepsilon_2 \in Q^+ \quad |\alpha(\varepsilon_1) - \alpha(\varepsilon_2)| \leqslant \varepsilon_1 + \varepsilon_2$.

*Q 6. En utilisant l'exemple de la question Q 2, montrer qu'il existe un processus calculable pour obtenir une approximation rationnelle de $\sqrt{2}$ avec une erreur inférieure ou égale à $\varepsilon$.*

On dira qu'un processus calculable $\alpha$ est un *processus calculable nul* si

$$\forall\, \varepsilon \in Q^+ \quad |\alpha(\varepsilon)| \leqslant \varepsilon\,.$$

Les processus définis par $\alpha(\varepsilon) = \varepsilon$ ou $\alpha(\varepsilon) = 0$ sont évidemment des processus calculables nuls.

*Q 7. Soit $\alpha$ un processus calculable vérifiant la propriété suivante :*

*Il existe $\varepsilon_0 \in Q^+$ et $r$ rationnel, positif ou nul tels que $\varepsilon < \varepsilon_0$ entraîne $|\alpha(\varepsilon)| \leqslant r\,\varepsilon$.*

*Démontrer que $\alpha$ est un processus calculable nul.*

2° Soient $\alpha$ et $\beta$ deux processus calculables.
On pose, pour $\varepsilon \in Q^+$, $(\alpha \pm \beta)(\varepsilon) = \alpha(\varepsilon/2) \pm \beta(\varepsilon/2)$.

*Q 8. Prouver que $\alpha \pm \beta$ sont des processus calculables (on utilisera les résultats de Q 5). On désigne par $\Gamma$ l'ensemble des processus calculables, et dans $\Gamma$ on définit la relation $\sim$ par :*

$$\alpha \sim \beta \Leftrightarrow [\alpha - \beta \text{ est un processus calculable nul}].$$

*Q 9. Prouver que la relation $\sim$ est d'équivalence (on peut utiliser les résultats de Q 7 pour démontrer la transitivité).*

*Q 10. Si $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \Gamma$ avec $\alpha_1 \sim \alpha_2$ et $\beta_1 \sim \beta_2$, montrer que $\alpha_1 \pm \beta_1 \sim \alpha_2 \pm \beta_2$.*

*Définition :* on appelle *nombre calculable* toute classe de processus calculables modulo la relation d'équivalence $\sim$. On utilisera des minuscules latines *a, b, ...* pour les désigner, et les minuscules grecques, $\alpha, \beta, \ldots$ correspondantes pour les représentants des classes ; $a = \{\alpha\}$ signifiera que $a$ est la classe du processus $\alpha$. L'ensemble de ces classes est noté C.

*Q 11. Si $a = \{\alpha\}$ et $b = \{\beta\}$, prouver que sur C la loi d'addition : $a + b = \{\alpha + \beta\}$ est bien définie et est une loi de groupe commutatif.*

3° Soient $\alpha, \beta \in \Gamma$; on pose, pour tout $\varepsilon \in Q^+$ et tout entier N strictement positif

$$\varphi(\varepsilon, N) = \big(|\alpha(\varepsilon/N)| + |\beta(\varepsilon/N)| + 2\,\varepsilon/N\big)\, 2\varepsilon/N\,.$$

Alors, pour tout $\varepsilon \in Q^+$, on a : $\lim\limits_{N\to\infty} \varphi(\varepsilon, N) = 0$.

Désignant par $N(\varepsilon)$ le plus petit entier tel que $\varphi(\varepsilon, N) \leqslant \varepsilon$, on pose :

$$(\alpha.\beta)(\varepsilon) = \alpha[\varepsilon/N(\varepsilon)]\,.\beta[\varepsilon/N(\varepsilon)].$$

*Q 12. Justifier l'existence de $N(\varepsilon)$, et montrer que $\alpha.\beta$ est un processus calculable.*

*Q 13. Démontrer que, si $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \Gamma$ avec $\alpha_1 \sim \alpha_2$ et $\beta_1 \sim \beta_2$, alors $\alpha_1.\beta_1 \sim \alpha_2.\beta_2$.*

*Q 14. Si* $a = \{\alpha\}$, $b = \{\beta\}$, *on pose :* $a.b = \{\alpha.\beta\}$.

*Démontrer qu'on définit ainsi sur C une loi de composition commutative, associative, et distributive par rapport à la loi d'addition précédente.*

4° Soient $\alpha$, $\beta$ deux éléments de $\Gamma$, $\beta$ n'étant pas un processus calculable nul; on pose :

$$(\alpha/\beta)(\varepsilon) = \alpha[\varepsilon/M(\varepsilon)]/\beta[\varepsilon/M(\varepsilon)]$$

où $M(\varepsilon)$ est le plus petit des entiers M tels que :

*i.* $|\beta(\varepsilon/M)| > 2\,\varepsilon/M$;

*ii.* $(2\varepsilon/M)\big(|\alpha(\varepsilon/M)| + |\beta(\varepsilon/M)|\big) \leqslant \varepsilon\big(|\beta(\varepsilon/M)| - 2\varepsilon/M\big).|\beta(\varepsilon/M)|$.

*Q 15. Justifier l'existence de* $M(\varepsilon)$ *et prouver que* $\alpha/\beta$ *est un processus calculable.*

*Q 16. Si* $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \Gamma$ *et* $\alpha_1 \sim \alpha_2$ *et* $\beta_1 \sim \beta_2$ *(sans être équivalent aux processus nuls), alors* $\alpha_1/\beta_1 \sim \alpha_2/\beta_2$.

*Q 17. Si b est un nombre calculable différent de la classe des processus nuls, on posera* $a/b = \{\alpha/\beta\}$. *Démontrer que cette loi est bien définie.*

*Q 18. Prouver que les lois précédentes* $(\pm, ., /)$ *donnent à C une structure de corps commutatif : c'est le corps des nombres calculables. Préciser les relations avec Q : peut-on considérer Q comme partie de C?*

*Q 19. Démontrer qu'on peut étendre à C la relation d'ordre* $\leqslant$ *de Q.*

## III

1° *Codification des programmes.*

Soit un programme P, défini par la donnée de $r$ instructions désignées par $S_j$ $(j = 1, \ldots, r)$. A l'instruction $S_j$ on fait correspondre 7 entiers non négatifs $S_j^1, S_j^2, S_j^3, S_j^4, S_j^5, S_j^6, S_j^7$ selon la règle exprimée par la table ci- dessous, commentée ensuite :

| | $S_j$ | $S_j^1$ | $S_j^2$ | $S_j^3$ | $S_j^4$ | $S_j^5$ | $S_j^6$ | $S_j^7$ |
|---|---|---|---|---|---|---|---|---|
| Type 1 | $V_{t_3} := V_{t_1} \tau V_{t_2}$ ; $m$ | 1 | $\sigma(\tau)$ | $i_1$ | $i_2$ | $i_3$ | $\lvert m \rvert$ | $t(m)$ |
| Type 2 | $V_{t_1} := x$ ; $m$ | 2 | $t(p)$ | $i_1$ | $\lvert p \rvert$ | $q$ | $\lvert m \rvert$ | $t(m)$ |
| Type 3 | $V_{t_1}$ ; $m_1$, $m_2$ | 3 | 0 | $i_1$ | $\lvert m_1 \rvert$ | $t(m_1)$ | $\lvert m_2 \rvert$ | $t(m_2)$ |
| Type 4 | Arrêt | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

où :

$t(y)$ est définie pour $y \in Z$, et vaut $t(y) = 0$ si $y > 0$, $t(y) = 1$ pour $y \leqslant 0$;

$| y |$ est valeur absolue de $y$;

$\sigma(\tau) = 0, 1, 2, 3$ selon que $\tau$ est $+, —, \times, \div$ respectivement;
enfin, si $S_j$ est de type 2 c'est-à-dire de la forme $V_{t_1} := x; m$, on admet que le rationnel $x$ est donné sous la forme $p/q$ où $p$ et $q$ sont deux entiers relatifs tels que :

*i.* $p$ et $q$ sont premiers entre eux;

*ii.* $q$ est $> 0$;

*iii.* si $x = 0$, alors $p = 0$ et $q = 1$.

*Q 20. Démontrer que le 7-uple $(S_j^1, \ldots S_j^7)$ permet de reconstituer l'expression de l'instruction $S_j$.*

*Soit alors $\{ n_t \}$ $(t = 1, \ldots, m)$ une suite finie d'entiers positifs ou nuls.*

*On note :*

$$n_t = \sum_{i=0}^{i=r(t)} D_t^i 2^i$$

*l'expression binaire de $n_t$ où les $D_t^i$ valent 0 ou 1, $D_t^{r(t)}$ valant toujours 1, sauf pour $n_t = 0$ où $r(t)$ vaut 0 et $D_t^o = 0$.*

*On définit alors, pour $t = 1, \ldots, m$*

$$\omega(t) = \sum_{i=1}^{i=t-1} [r(i) + 2] \quad \textit{pour} \quad t > 1; \quad \omega(1) = 0$$

*et l'on pose :*

$$< n_1, n_2, \ldots, n_m > = \sum_{t=1}^{t=m} 3^{\omega(t)} [2.3^{r(t)+1} + \sum_{i=0}^{i=r(t)} D_t^i 3^i] .$$

Soit P un programme, défini par les instructions $S_1, S_2, \ldots, S_r$; on définit alors *l'entier descriptif* $N_p$ de P par :

$$N_p = < S_1^1, S_1^2, \ldots, S_1^7, S_2^1, S_2^2, \ldots S_2^7, \ldots S_r^1, S_r^2, \ldots, S_r^7 > .$$

*Q 21. On expliquera la formation de $N_p$ pour le programme P de la question Q 1.*

*Q 22. Démontrer que $N_p$ est caractéristique d'un programme et que l'ensemble de tous les programmes est dénombrable.*

2° Pour tout processus calculable $\alpha$, il existe (cf. II.1) un programme $P_\alpha$, dont l'*entier descriptif* est $N_\alpha$, permettant de calculer, pour tout $\varepsilon$ de $Q^+$, la valeur $\alpha(\varepsilon)$.
Soit alors $F(n, \varepsilon)$ une application programmable de l'entier positif $n$ et du rationnel positif $\varepsilon$, ayant les propriétés suivantes :

*a.* Pour tout processus calculable $\alpha$, le processus :

$$\Phi_\alpha : \varepsilon \in Q^+ \longmapsto F(N_\alpha, \varepsilon) \text{ est calculable;}$$

*b.* Si $\alpha \sim \beta$, alors $\Phi_\alpha \sim \Phi_\beta$.

Si I est une partie non vide de C et s'il existe une telle application programmable $F(n, \varepsilon)$, on définit une *fonction calculable f*, de domaine I et à valeur dans C, en posant $f(x) = \{ \Phi_\alpha \}$ pour tout $x = \{\alpha\} \in I$.

*Q 23. Démontrer que $x \longmapsto x$ (l'identité sur C), $x \longmapsto |x|$ (la valeur absolue sur C) sont des fonctions calculables.*

*Q 24. Si f, g sont des fonctions calculables sur I, démontrer que :*

$x \longmapsto f(x) \pm g(x)$ *et* $x \to f(x).\ g(x)$ *sont des fonctions calculables sur I;*

$x \longmapsto f(x)/g(x)$ *est une fonction calculable sur I, si* $g(x) \neq 0$ *pour* $x \in I$.

3° Soit $\alpha_n(\varepsilon)$ une application programmable de $n$ (entier positif) et de $\varepsilon$ (rationnel positif) telle que, pour $n$ fixé, $\alpha_n : \varepsilon \longmapsto \alpha_n(\varepsilon)$ soit un processus calculable; en posant $a_n = \{\alpha_n\} \in C$, on définit une *suite calculable de nombres calculables.*

On dit qu'une *telle suite* $a_n$ converge vers $l \in C$, et on écrit $\lim_{n\to\infty} a_n = l$, s'il existe une fonction calculable $f$ définie pour tout $e \in C^+$ (où $C^+$ est l'ensemble des nombres calculables positifs), telle que $|\ a_n - l\ | \leqslant e$ pour tout $n \leqslant f(e)$. Une *telle suite* est dite « de Cauchy », s'il existe une fonction calculable $g$ définie sur $C^+$ telle que $|a_n - a_m| \leqslant e$ pour tout $n \geqslant g(e)$ et $m \geqslant g(e)$.

*Q 25. Démontrer qu'une condition nécessaire et suffisante, pour qu'une suite calculable de nombres calculables* $a_n$ *converge vers une limite calculable, est qu'elle soit « de Cauchy ».*

## ANALYSE NUMERIQUE

**Compte rendu de l'épreuve** (361 candidats) (225 candidates)

Le problème consistait dans l'étude progressive de résultats de base concernant les applications programmables, les processus, nombres et fonctions calculables, et enfin (dernière question) les suites calculables de nombres calculables. Il s'agit là d'analyse dite « constructive » dans le corps des « nombres calculables » (accessibles à un ordinateur qui calculerait exactement dans Q).

Cette motivation « concrète » semble avoir à la fois intéressé et désorienté les candidats. Le travail proposé allait de questions simples à des questions difficiles. Cette nécessité d'abstraire dans un contexte original semble avoir souvent impressionné les candidats et l'ensemble du problème a prouvé le peu d'habitude à raisonner sur des formalismes.

A) Les 6 premières questions correspondent à l'apprentissage du langage de programmation élémentaire (mais parfaitement suffisant) défini dans l'introduction. Elles sont à peu près comprises en général. Néanmoins, trop peu de candidats ont le souci d'écrire des instructions formellement correctes (on écrit souvent : $S_1 : V_1 := V_2 ; 1$ ou encore $S_1 : V_1 = - V_2 ; 1$, voire même $S_1 : V_1 := g_1(b_1, \ldots, b_m) ; 1$). D'autre part, rares sont les copies qui en $Q_5$ pensent à gérer correctement les mémoires, de façon à ne pas « écraser » les résultats intermédiaires du calcul. La question $Q_6$ est rarement comprise ; cette première partie montre qu'assez peu de candidats ont le sens de la programmation, c'est-à-dire de l'enchaînement d'instructions correctes dans un formalisme rigoureux.

B) On étudie alors, de $Q_7$ à $Q_{11}$, une loi d'addition sur l'ensemble $\Gamma$ des processus calculables, définie par $(\alpha+\beta)(\varepsilon) = \alpha(\varepsilon/2) + \beta(\varepsilon/2)$. Son introduction est due au fait que, lorsque deux quantités sont connues à $\varepsilon/2$ près, leur somme l'est à $\varepsilon$ près. Cette loi « bizarre » fait trébucher beaucoup de candidats. $Q_7$, exercice élémentaire d'analyse classique, est rarement traité correctement (50 % des copies seulement) par contre $Q_8$ l'est généralement bien.

Un premier clivage se produit alors : les copies médiocres ne parviennent pas à « admettre » la définition de l'addition des processus calculables, l'appliquent de façon incorrecte (en revenant irrésistiblement à l'addition usuelle des applications) et la déclarent même « évidemment associative puisque l'addition des applications l'est ».

Néanmoins, entre $Q_7$ et $Q_{11}$, un lot important de copies prouve que l'essentiel est souvent compris : à savoir que ($\Gamma$+ ) n'étant pas associatif, on peut définir sur $\Gamma$ une relation d'équivalence compatible avec l'addition dans $\Gamma$. Cette relation revient à identifier deux processus calculables dont la différence est «nulle à $\varepsilon$ près» (c'est-à-dire : est un processus calculable nul). La loi induite sur le quotient - noté C - est alors une loi de groupe abélien, résultat important, qui semble avoir intéressé les candidats l'ayant compris.

C) De $Q_{12}$ à $Q_{19}$, il s'agit de montrer que C est non seulement un groupe abélien, mais peut être muni d'une structure de corps commutatif qui étend celle de Q (corps des nombres calculables). L'injection de Q dans C est alors la suivante : si $q \in Q$, on note encore $q$ la classe des processus calculables équivalents au processus $\alpha$ tel que $\forall \varepsilon \in Q^+$, $\alpha(\varepsilon) = q$.

Aucune copie ne traite de façon complète cette partie, qui comporte à la fois une démarche abstraite et des calculs. Un bon nombre de copies s'arrêtent d'ailleurs en $Q_{12}$, pour reprendre parfois en $Q_{20}$, $Q_{21}$, $Q_{22}$.

Un deuxième clivage se produit alors : sans parvenir à faire les calculs correspondants, un lot de copies arrive néanmoins à effectuer la démarche correcte : il leur est attribué, si le calcul est correctement posé, la quasi totalité du barême. A ce propos, bien peu de candidats prouvent qu'ils ont le sens de trouver le point essentiel des démonstrations demandées.

L'extension à C de la relation d'ordre habituelle sur Q ($Q_{19}$) n'est pratiquement jamais faite correctement : pour $a \in C$, $a = \{\alpha\}$ ; on pose $a \geqslant 0$ s'il existe $\varepsilon > 0$ tel que l'on ait $\alpha \geqslant \varepsilon$ (dans Q). Il reste à montrer que cette définition est cohérente, c'est-à-dire qu'elle ne dépend pas du représentant $\alpha$ choisi.

D) Les questions $Q_{20}$ à $Q_{22}$ reviennent sur des problèmes plus «concrets», précisément sur le codage des programmes (de façon assez fidèle à ce qui se fait réellement dans un ordinateur). Elles connaissent un regain d'intérêt. La plupart des candidats parviennent à montrer, par l'intermédiaire de l'entier descriptif d'un programme, que l'ensemble des programmes est dénombrable. Très peu comprennent que ce codage peut servir à écrire un programme générant puis exécutant un programme d'entier descriptif donné.

E) Enfin, la dernière partie (questions $Q_{23}$ à $Q_{25}$), portant sur les fonctions et les suites calculables, n'est abordée de façon sérieuse que par quelques copies. La partie difficile de $Q_{25}$, montrant que C est «complet», n'est jamais abordée.

**CONCLUSIONS** (sur 225 copies féminines et 361 copies masculines)

1) pas de copie de très bonne qualité ;

2) un petit lot de bonnes copies, ayant compris l'essentiel du problème, et faisant preuve d'aisance et de solidité, ce qui est agréable à constater.

3) Un lot important de copies «moyennes», beaucoup plus timorées, préférant (ce qui est effectivement préférable) en faire peu (jusqu'à $Q_{11}$ généralement) plutôt que d'écrire des choses fausses.

4) Une grande masse de copies très médiocres (premier clivage signalé) contenant l'erreur de base : l'impression est pénible de constater qu'il s'agit là de candidats ne maîtrisant même pas des notions simples.

Répartition des notes (barème commun sur 40)

| | Féminines | Masculines |
|---|---|---|
| entre 0 et 5 | 27 copies | 26 copies |
| entre 6 et 10 | 70 copies | 37 copies |
| entre 11 et 15 | 77 copies | 153 copies |
| entre 16 et 19 | 38 copies | 106 copies |
| entre 20 et 40 | 13 copies | 39 copies |
| | 225 copies | 361 copies |

F. ROBERT

N. GASTINEL