U2 – MATHÉMATIQUES POUR L'INFORMATIQUE

U21 – MATHÉMATIQUES

Cette unité d'enseignement se décline en six modules spécifiques :

- Arithmétique ;
- Suites numériques ;
- Calcul matriciel 2 ;
- Calcul des propositions et des prédicats, langage ensembliste, calcul booléen ;
- Éléments de la théorie des ensembles ;
- Graphes et ordonnancement.

ARITHMÉTIQUE

Le programme concerne les notions les plus utiles à l'informatique. La numération est indispensable aux langages de bas niveau. L'arithmétique modulaire est utile à la cryptographie, aux corrections d'erreurs et plus généralement à de nombreux algorithmes.

Systèmes de numération

Numération en bases 10, 2 et 16 des entiers et des réels. Conversions entre bases.	
Notion d'arrondi et de précision.	Sans aucune théorie sur les calculs d'incertitude.
Addition, soustraction, multiplication et division des entiers naturels.	En particulier, en base 2 pour les puissances de deux.

Arithmétique modulaire

Division euclidienne : quotient, reste, existence, unicité. Nombres premiers, décomposition en produit de facteurs premiers, entiers premiers entre eux, PGCD de deux entiers.	
Notion de congruence, propriétés élémentaires. Modulo n, les multiples de a sont les multiples de pgcd(a,n).	On évitera tout excès de technicité en s'efforçant d'utiliser des présentations concrètes.

<u>Travaux pratiques</u>

1° Exemples de calculs en bases 2 et 16. Conversions entre bases.	
2° Exemples d'algorithmes de recherche de nombres premiers et de décomposition en facteurs premiers.	Aucune technique n'est censée être connue.
3° Parcours d'une liste circulaire par sauts d'amplitude constante.	On notera en particulier que le parcours n'est exhaustif que quand la longueur du saut et la taille de la liste sont des entiers premiers entre eux.
4° Comparaison entre le calcul binaire et le calcul booléen.	Les booléens seront alors 1 et 0, interprétés comme signifiant « il y en a, ou pas ».

SUITES NUMÉRIQUES

Les suites sont un outil indispensable pour l'étude des "phénomènes discrets", et c'est à ce titre qu'elles font l'objet d'une initiation. Aucune difficulté théorique ne doit être soulevée à leur propos.
Le programme se place dans le cadre des suites définies pour tout entier naturel de l'intervalle d'étude.
On utilisera largement les moyens informatiques (calculatrice, ordinateur), qui permettent notamment de faciliter la compréhension d'un concept ou d'une méthode en l'illustrant graphiquement, numériquement ou dans un contexte lié à la spécialité, sans être limité par d'éventuelles difficultés techniques.

DOMAINE D'ÉTUDE

Les expressions utilisées sont construites à partir :

- des fonctions usuelles :
  - constante,
  - exponentielle $t \mapsto \exp t$ ou $t \mapsto e^t$,
  - logarithme népérien $t \mapsto \ln t$,
  - puissances $t \mapsto t^\alpha$ où $\alpha \in \mathbb{R}$,
- des fonctions qui se déduisent de façon simple des précédentes par opérations algébriques ou par composition.

On consolidera les acquis sur les fonctions usuelles, y compris limites et comparaison des fonctions exponentielle, puissances et logarithme népérien au voisinage de $+\infty$ les représentations graphiques devant jouer un rôle important.

a) Comportement global : suites croissantes, suites décroissantes.	
b) Langage des limites : Introduction du symbole $\lim\limits_{n \to +\infty} u_n$ Suites $u_n = f(n)$.	L'étude des limites par (A, N) et par (ε, N) est hors programme.
Si une fonction f admet une limite l en $+\infty$, alors la suite $u_n = f(n)$ converge vers l. Limite des suites de terme général n, n^2, n^3, $\sqrt{n}$. Limite des suites de terme général $\frac{1}{n}$, $\frac{1}{n^2}$, $\frac{1}{n^3}$, $\frac{1}{\sqrt{n}}$. Limite des suites géométriques (q^n), où q est strictement positif.	L'étude des suites de référence ci-contre et, plus largement, des suites $u_n = f(n)$ est à mener en liaison étroite avec celle des fonctions correspondantes.
Énoncés usuels sur les limites (admis). Comparaison, compatibilité avec l'ordre. Somme, produit, quotient.	Ces énoncés sont calqués sur ceux relatifs aux fonctions. Il n'y a pas lieu de s'attarder sur leur présentation : l'objectif est d'apprendre aux étudiants à les mettre en œuvre sur des exemples simples.
Limite et comportements asymptotiques comparés des suites $(\ln n)$; (a^n), a réel strictement positif ; (n^p), p entier.	

Travaux pratiques

1° Exemples d'étude de situations relevant de suites arithmétiques ou géométriques.	On privilégiera les situations concrètes, notamment celles issues de la spécialité. Les formules permettant de calculer la somme de termes consécutifs ne seront pas utilisées sans être rappelées. Mis à part le cas des suites arithmétiques ou géométriques, toute étude théorique d'une suite définie par son premier terme et une relation de récurrence $u_{n+1} = f(u_n)$ est hors programme.
2° Exemples d'étude du comportement de suites de la forme $u_n = f(n)$ (encadrement, monotonie, limite).	On se limitera à des cas simples. Il s'agit notamment de pouvoir étudier et comparer, sur certains modèles mathématiques, la tendance à long terme d'un phénomène.

CALCUL MATRICIEL 2

Il s'agit d'une initiation au langage matriciel, s'appuyant sur l'observation de phénomènes issus de la vie courante ou d'exemples concrets. On cherche principalement à introduire un mode de représentation facilitant l'étude de tels phénomènes, sans qu'il soit utile de faire intervenir le concept d'application linéaire. On utilisera largement les moyens électroniques, les calculs à la main étant limités aux cas les plus élémentaires servant à introduire les opérations sur les matrices.

Matrices	Une matrice est introduite comme un tableau de nombres permettant de représenter une situation comportant plusieurs "entrées" et "sorties".
Égalité de deux matrices. Matrices nulles, matrices carrées identité.	
Calcul matriciel élémentaire : addition, multiplication par un nombre, multiplication.	Le choix de la définition de chaque opération portant sur les matrices s'appuie sur l'observation de la signification du tableau de nombres ainsi obtenu. On insistera sur le caractère non commutatif de la multiplication et l'absence de division.
Notion d'inverse. Existence éventuelle d'une matrice inverse. Unicité. Une matrice commute avec son inverse. Savoir reconnaître qu'une matrice est l'inverse d'une autre.	La notion de déterminant n'est pas au programme. Aucune condition d'existence n'est à connaître.

Travaux pratiques

1° Calcul de sommes et de produits de matrices.	Lors des évaluations, le résultat pourra être obtenu à la calculatrice, sans justification.
2° Résolution d'un système de *n* équations à *n* inconnues.	On se placera toujours dans un système de Cramer, sans qu'aucune justification ne soit requise. Ne peut faire l'objet d'une évaluation, sauf à rappeler la méthode.
3° Présentation d'une méthode itérative du calcul de l'inverse, quand il existe.	Aucune justification n'est requise concernant la méthode et elle n'a pas à être apprise. L'évaluation de cette activité relève de l'enseignement d'algorithmique appliquée.

CALCUL DES PROPOSITIONS ET DES PRÉDICATS, LANGAGE ENSEMBLISTE, CALCUL BOOLÉEN

1. CALCUL DES PROPOSITIONS ET DES PRÉDICATS

L'objectif est d'introduire quelques éléments de logique en liaison avec l'enseignement de l'informatique. Il s'agit d'une brève étude destinée à familiariser les élèves à une pratique élémentaire du calcul portant sur des énoncés.

a) Calcul propositionnel Proposition, valeur de vérité Connecteurs logiques : négation (non P, $\neg P$, $\bar{P}$), conjonction (P et Q, $P \wedge Q$), disjonction (P ou Q, $P \vee Q$), implication, équivalence.	On dégagera les propriétés fondamentales des opérations ainsi introduites, de manière à déboucher ensuite sur un exemple d'algèbre de Boole.
b) Calcul des prédicats Variable, constante Quantificateurs $\forall$, $\exists$. Négation de $\forall x, p(x)$; négation de $\exists x, p(x)$.	On se limitera à des cas simples de prédicats portant sur une, deux ou trois variables. On signalera l'importance de l'ordre dans lequel deux quantificateurs interviennent.

2. LANGAGE ENSEMBLISTE

Sans développer une théorie générale des ensembles, l'objectif est de consolider et de prolonger les acquis des élèves sur les ensembles en liaison avec l'enseignement de l'informatique.

Ensemble, appartenance, inclusion. Ensemble *P* (E) des parties d'un ensemble E. Complémentaire d'une partie, intersection et réunion de deux parties. Les éléments *x* d'un ensemble E satisfaisant à une relation *p(x)* constituent une partie de E.	On dégagera les propriétés fondamentales des opérations ainsi introduites, de manière à déboucher ensuite sur un exemple d'algèbre de Boole. Cela permet d'interpréter en termes ensemblistes l'implication, la conjonction et la disjonction de deux relations, ainsi que la négation d'une relation.

3. CALCUL BOOLÉEN

Cette brève étude est à mener en coordination étroite avec l'enseignement de l'informatique. Il convient d'introduire la notion d'algèbre de Boole à partir des deux exemples précédents. Il s'agit essentiellement d'effectuer des calculs permettant de simplifier des expressions booléennes.

Définition d'une algèbre de Boole. Propriétés des opérations, lois de Morgan.	On adoptera les notations usuelles $\bar{a}$, a + b, ab.

<u>Travaux pratiques</u>

1° Exemples simples de calculs portant sur des énoncés.	On se limitera à des cas simples où l'utilisation des tables de vérité ou de propriétés élémentaires permet de conclure sans excès de technicité.
2° Traduire une instruction de boucle à l'aide de connecteurs logiques.	L'évaluation de cette activité relève de l'enseignement d'algorithmique appliquée.
3° Exemples simples de calculs portant sur des variables booléennes.	On se limitera à des cas simples, comportant au plus trois variables booléennes, où l'utilisation de tableaux de Karnaugh ou de propriétés algébriques élémentaires permet de conclure sans excès de technicité. On signalera l'intérêt des connecteurs non-ou (nor), non-et (nand).

ÉLÉMENTS DE LA THÉORIE DES ENSEMBLES

Ce module vient compléter, concernant les ensembles, celui relatif aux algèbres de Boole. Il développe les notions de produit cartésien, de relation et d'application en liaison avec les nombreuses utilisations qui en sont faites en informatique.

1° Produit cartésien de deux ensembles Cardinal de E×F dans le cas où E et F sont finis.	On généralisera au cas du produit cartésien de n ensembles finis.
2° Relations binaires Définition, propriétés. Relations d'équivalence, relations d'ordre.	On évitera un trop grand formalisme. On ne s'intéresse qu'aux utilisations en informatique.
3° Application f d'un ensemble E dans un ensemble F Image d'une partie A de E, image réciproque d'une partie B de F.	Les exemples illustrant ce paragraphe seront choisis en liaison avec l'enseignement de l'informatique. On soulignera l'importance de la notion d'injection pour coder des informations et de la notion d'image réciproque pour effectuer des tris.
Injection, surjection, bijection.	On attachera plus d'importance à une caractérisation textuelle qu'à l'énoncé de prédicats.
Composition d'applications.	

<u>Travaux pratiques</u>

1° Exemples de situations où la modélisation ou des contraintes de fonctionnement requièrent des exigences se traduisant : en termes de relation d'ordre ou d'équivalence, en termes d'injection, de surjection ou de bijection. 2° Exemples de composition d'applications toutes deux soit injectives, soit surjectives, soit bijectives.	On trouvera de nombreux exemples en informatique (codage, tri, compression…)

GRAPHES ET ORDONNANCEMENT

<u>Graphes</u>

L'objectif est d'introduire et de mettre en œuvre, dans des situations concrètes très élémentaires et sans théorie générale, des algorithmes permettant de résoudre les problèmes figurant dans la rubrique de travaux pratiques.

Modes de représentation d'un graphe fini simple orienté : représentation géométrique, tableau des successeurs ou des prédécesseurs, matrice d'adjacence booléenne.	La définition d'un graphe fini simple orienté n'est pas au programme.
Interprétation des puissances entières et booléennes de la matrice d'adjacence. Chemin, circuit, boucle, chemin hamiltonien. Fermeture transitive.	Uniquement pour un graphe non valué.
Pour un graphe sans circuit : niveau d'un sommet, niveaux du graphe.	Il conviendra de savoir déterminer les niveaux, sans qu'aucune méthode ne soit imposée.

Arborescence.	La notion de connexité étant hors programme, on se limitera à la présentation d'exemples simples d'arborescences à partir de leur représentation géométrique, sans recherche d'une caractérisation générale.
Longueur d'un chemin, chemin optimal en longueur. Graphes valués (pondérés). Chemin optimal en valeur.	On observera l'importance du résultat : tout sous chemin d'un chemin optimal est optimal. Simple présentation, sans théorie particulière.

Ordonnancement

L'objectif est double : sensibiliser l'étudiant aux problèmes d'ordonnancement et traiter manuellement un algorithme. Aucune justification théorique des algorithmes utilisés n'est au programme. On abordera MPM ou PERT. On s'attachera surtout à la compréhension des mécanismes. Et, les cas traités resteront suffisamment modestes pour que la rapidité ne soit pas un critère d'évaluation fondamental.

Méthode M.P.M ou méthode P.E.R.T., principe de représentation. Dates au plus tôt, au plus tard. Tâches et chemin critiques. Marge totale, libre, certaine.	L'étudiant doit savoir mettre en œuvre l'algorithme utilisé et les interprétations des notions abordées doivent être connues. Aucune autre compétence théorique n'est requise.

Travaux pratiques

1° Exemples de mise en œuvre d'algorithmes permettant d'obtenir pour un graphe : • les chemins de longueur p, • la fermeture transitive, • les niveaux, dans le cas d'un graphe sans circuit, • une optimisation.	À partir d'exemples très élémentaires et sans introduire une théorie générale, on montrera l'intérêt des méthodes matricielles mettant en œuvre l'addition et la multiplication booléennes des matrices d'adjacence. Dans une évaluation, tout énoncé relatif à ces algorithmes doit comporter des indications sur la méthode à suivre.
2° Exemples de résolution de problèmes d'ordonnancement par la méthode des potentiels métra (M.P.M.) ou la méthode P.E.R.T.	On présentera quelques cas concrets simplifiés et on les interprétera.

U22 – ALGORITHMIQUE APPLIQUÉE

Les thématiques abordées lors de l'étude de ce module sont très ouvertes, mais l'objectif visé, à l'intérieur du processus de conception, est fortement ciblé. On veillera à ce que les situations proposées soient mathématiquement achevées. Alors qu'une solution, voire un pré-algorithme, peuvent être décrits de manière très libre, textuelle ou graphique, par formules ou symboles, par l'exemple ou de manière inachevée, on s'attache ici à l'exprimer en utilisant les outils algorithmiques usuels, pour la rendre directement convertible et exécutable sur machine.

Afin de faciliter la compréhension des mécanismes et la détection d'éventuelles erreurs, il est impératif de les concrétiser par l'emploi d'un langage de programmation et de conduire l'étudiant à réaliser des tests. La tâche inverse, consistant à comprendre un algorithme, présente également un grand intérêt pour l'assimilation des mécanismes et lors d'opérations de maintenance.

Les compétences et savoir-faire à acquérir concernent la compréhension des solutions proposées, l'interprétation des algorithmes (découverte de leurs rôles), leur construction conforme aux prescriptions et convenablement documentée, leur transcription dans un langage informatique, leur mise au point et leur validation.

Les contrôles d'exécution constituent le cœur des mécanismes algorithmiques de base. À ce titre, on attachera un soin tout particulier à leur étude progressive mais détaillée. On ne se limitera, en aucun cas, à en définir les fonctionnements. Leur maîtrise pratique est essentielle et les évaluations doivent être centrées sur eux.

Pour l'écriture des algorithmes, on évitera l'utilisation de symboles graphiques contraignants. Une représentation textuelle convenablement indentée avec des indicateurs de début et de fin explicites conviendra. Pour faciliter la compréhension, on exigera également un en-tête composé d'un nom, d'un rôle, de l'indication des données d'entrée et de sortie et de la déclaration typée des données locales. Des commentaires seront ajoutés, si utiles, notamment pour préciser le rôle des variables et d'éventuelles indications méthodologiques.

Pour la programmation, on peut notamment utiliser un tableur, un langage de calcul formel ou un langage de haut niveau, éventuellement exécutable dans un navigateur. Aucun langage ni aucun logiciel n'est imposé, mais il convient de s'assurer qu'il est accepté par le centre d'examen.

Les sujets empruntés à la vie courante pourront être utilisés à chaque fois qu'ils permettent d'illustrer un mécanisme simple avec pertinence. Sinon, on préférera utiliser des sujets dérivés directement des modules mathématiques de l'U21 et, avec un certain équilibre, des sujets associés à des thématiques informatiques (par exemple : codage, cryptage et décryptage, redondance de sécurité, tri itératif et tri récursif, parcours de graphes). Ces sujets seront abordés à fin d'illustration des concepts fondamentaux d'algorithmique et de familiarisation avec les notions, les entités et les méthodes manipulés par ces thèmes, sans qu'aucune connaissance spécifique ne soit exigible de l'étudiant dans ces derniers domaines.

Généralités

Les concepts fondamentaux (algorithme, finitude, modularité, identifiant, constante, variable, fonction, procédure, expression numérique, expression conditionnelle et plus généralement booléenne…) seront acquis par l'usage, sans faire appel à des définitions formelles préalables.	Le formalisme a posteriori, utile à une compréhension fine, n'est pas exclu, mais ne peut faire l'objet d'évaluation.

Données manipulées

Types simples : entier naturel, entier relatif, réel,	
booléen,	Peu importe la façon dont les valeurs de vérité sont représentées.
chaîne de caractères.	On évitera de considérer une chaîne comme un tableau de caractères.
Tableaux à une ou deux dimensions de type homogène, tableaux à deux dimensions constitués de tableaux à une dimension dont les types ne sont pas homogènes.	On adaptera la construction et l'exploitation de ces tableaux aux possibilités de l'outil informatique utilisé. Les structures de données, les objets, les fichiers ne sont pas au programme de mathématiques : ils figurent dans les enseignements professionnels.
Paramètres d'entrée, valeur(s) retournée(s) par une fonction, variables globales ou locales.	Sans aborder la programmation objet, les concepts de modularité, d'interface et de portée des variables devront être assimilés.

Instructions de base et opérateurs utilisés

Lecture, écriture. Affectation, affectation récursive (la variable affectée participe à l'expression évaluée). Opérateurs numériques : addition, soustraction, multiplication, division, exponentiation, quotient et reste de la division entière, signes. Fonctions mathématiques usuelles.	
Opérateurs de comparaison : =, <> ou !=, <, <=, >, >=	Entre numériques et entre chaînes de caractères.
Opérateurs booléens : non, et, ou, oux	
Opérateurs booléens bit à bit	On interprétera notamment en termes de masque, de mise à un, de mise à zéro, de changement d'état.
Opérateurs de chaînes : concaténation. Fonctions permettant l'extraction en début, milieu ou fin ; la recherche d'un motif.	L'usage d'expressions régulières simples est possible, mais l'étude des expressions régulières est hors programme.
Transtypage	
Toute autre instruction, fonction ou procédure utile aux algorithmes traités.	Les descriptions sémantique et syntaxique précises seront alors mises à disposition de l'étudiant.

Contrôle d'exécution

Par défaut : l'exécution séquentielle.	
Exécution à structure conditionnelle (si-alors-sinon),	Afin d'en maîtriser le fonctionnement, les structures d'exécution seront elles-mêmes présentées sous forme d'algorithmes.
Exécution à structure itérative déterministe (pour) et indéterministe (tant que / répéter jusqu'à ce que). Méthodologie de construction des structures itératives : raisonnement par récurrence, initialisation, mise à jour itérative et neutralisation des cas indésirables, calcul itératif (souvent récursif), mise en forme finale.	Le raisonnement par récurrence n'a pas à être évalué pour des démonstrations. Il est introduit pour servir de base à une méthodologie non empirique de construction des itérations.

Symboles $\sum_{i=\dots}^{\dots}$ et $\prod_{i=\dots}^{\dots}$, traduction algorithmique.	
Structures imbriquées, y compris quand les éléments de contrôle des structures internes dépendent de ceux des structures externes. Le nombre d'imbrications n'est pas limité, sauf pour les itérations en dépendance, où on se limitera à deux.	On évitera les excès de complexité.
Récursivité. Nécessité d'un test. Nécessité de cas particuliers résolus sans appel à la récursivité, finitude.	Pas de récursivité mutuelle de plusieurs procédures. Aucune connaissance théorique sur la complexité ou la conversion en itérations.

Travaux pratiques

1° Exemple(s) de récursivité terminale, conversion en algorithme non récursif. Exemple de récursivité non terminale.	Aucune connaissance n'est exigible.
2° Variantes où les notions de complexité temporelle et de complexité spatiale peuvent être abordées.	L'étude de la complexité des algorithmes se limitera au calcul d'un nombre minimum ou maximum d'opérations significatives ou de taille des données.
3° Exemples d'effets indésirables (effets de bord, évaluation partielle lors de calcul d'expressions booléennes, débordements ou approximations numériques, transtypage, utilisation d'indices hors domaine,…).	Afin de mieux sensibiliser l'étudiant à certains risques, on s'efforcera de présenter des cas aux conséquences spectaculaires. Aucune théorie n'est au programme.
4° Algorithme délibérément erroné dont les défauts seront repérés par débogage.	On procédera essentiellement par suivi de variables.
5° Interprétation d'algorithmes.	L'ajout d'une ou plusieurs procédures ou fonctions à un ensemble interdépendant peut constituer une excellente base.

UF1 – LANGUE VIVANTE ÉTRANGÈRE 2

Le niveau à atteindre est celui fixé dans les programmes pour le cycle terminal (BO hors série n°7 28 août 2003) en référence au Cadre européen commun de référence pour les langues[6]: le niveau B2 pour la langue anglaise et le niveau B1 pour la seconde langue vivante étudiée, ici à titre facultatif.

Les tâches à mettre en œuvre sont à tirer de la description de l'épreuve U12 – Expression et communication en langue anglaise.

[6] Cadre européen commun de référence pour les langues ; apprendre, enseigner, évaluer ; Conseil de l'Europe 2001

UF2 – **MATHÉMATIQUES APPROFONDIES**

Cette unité d'enseignement facultative se décline en cinq modules standard :

- Fonction d'une variable réelle ;
- Calcul différentiel et intégral 1 ;
- Statistique descriptive ;
- Calcul des probabilités 2 ;
- Fiabilité.

1. FONCTION D'UNE VARIABLE RÉELLE

Sauf b) et c).

On se limitera aux fonctions à valeurs réelles et il n'y a pas lieu de reprendre ce qui est traité dans l'unité U21. Par contre, on complètera en fonction des besoins des autres modules de l'option.

Les calculs à la main nécessaires à la compréhension resteront élémentaires. On s'appuiera sur les calculatrices formelles pour le reste.

La notion de domaine de définition et une présentation très simplifiée de la continuité sont souhaitables pour aider à la compréhension. Cependant, la recherche des domaines de définition et l'étude de la continuité sont hors programme.

2. CALCUL DIFFÉRENTIEL ET INTÉGRAL 1

Ne sont pas au programme :

- *l'interprétation cinématique de la dérivée en un point,*
- *le calcul de primitive autrement que par dérivation,*
- *les fonctions périodiques ne sont pas au programme.*

La compréhension d'une modélisation et son interprétation seront jugées plus importantes qu'une agilité calculatoire, que toute bonne calculatrice supplantera avec aisance.

Concernant les calculs de dérivée, de limite, de primitive et d'intégrale, une certaine pratique manuelle peut aider à la compréhension. Elle sera limitée à des cas simples. La lecture inverse d'un tableau de dérivées sera proscrite, mais on pourra faire justifier une primitive par dérivation. Les autres cas seront résolus par le calcul formel.

L'étudiant doit savoir construire un tableau de variations, déterminer les extremums, interpréter les limites et les asymptotes, calculer des valeurs, construire et interpréter les graphiques, identifier ou calculer des solutions particulières et avoir un sens critique pour la validation de celles-ci. À titre d'exemple, rejeter éventuellement une solution négative, non entière ou plus généralement hors domaine, constitue un aspect très important de l'interprétation.

La calculatrice formelle permet d'étendre l'étude à des domaines où la technicité ne serait pas accessible à l'étudiant. Cependant, il doit rester le maître d'œuvre, justifier ses choix et commenter ses interprétations.

3. STATISTIQUE DESCRIPTIVE

Module standard

4. CALCUL DES PROBABILITÉS 2

Sauf d) le théorème de la limite centrée et des distributions d'échantillonnage

5. FIABILITÉ

A l'exception du paragraphe c) et du TP 2