

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Direction Générale des Etudes Technologiques

Rapport Final

de la Commission de Réforme des Programmes d'Informatique des Classes Préparatoires

Membres de la commission

Darragi Soumaya	Coordinatrice	IPEST
Ayed Sahloul Nesrine	Membre	IPEIT
Bartaguiz Hichem	Membre	ISEP BG
El Ayeb Faycel	Membre	IPEIM
Fathallah Karim	Membre	IPEIB
Fourati Fethi	Membre	IPEIS
Hamdana Aly	Membre	IPEIB
Hammami Leïla	Membre	IPEIN
Hammami Mohamed	Membre	IPEIEM
Hchaichi Haifa	Membre	ISSAT Gabès
Mabrouk Mohamed	Membre	ISSAT Mahdia
Trabelsi Abdelwaheb	Membre	FSS-Cycle préparatoires

Plan

Préambule.....	3
Critiques de l'existant.....	3
Programme proposé et conditions de mise en œuvre.....	5
Programme des sections MP/PC/T	
1. Programme de première année	7
2. Programme de deuxième année	12
Programme de la section BG	
1. Programme de première année	16
2. Programme de deuxième année	21
Documents de référence et liens utiles	24

Préambule

C'est dans le cadre de la nouvelle réforme des programmes d'enseignement en classes préparatoires, qu'a été établi ce document.

Il s'agit ici de présumer l'ensemble des discussions entre enseignants d'informatique réunis lors d'un atelier national de réflexion sur le programme d'informatique enseigné aux élèves des classes préparatoires. Cet atelier, auquel nous a invité le Ministère de l'Enseignement Supérieur à travers la Direction Générale des Etudes Technologiques, a permis de faire réunir plusieurs enseignants d'informatique afin de consolider les pratiques d'échanges pédagogiques et initier une collaboration mutuelle entre collègues. C'était également une occasion pour mettre à terme la proposition d'un nouveau programme d'enseignement de l'informatique en classes préparatoires.

Critiques de l'existant

Dans ce document nous présentons une réflexion concrète à l'évaluation des progressions pédagogiques actuelles. Cette réflexion, qui est le fruit de plusieurs réunions des membres du comité (PV en annexe), sera à la base d'une proposition d'une réforme et d'une amélioration des méthodes pédagogiques. Ceci est dans le but de contribuer à une formation à l'internationale des futurs ingénieurs tunisiens.

En fait, la démarche pédagogique adoptée actuellement pour l'enseignement de l'informatique en classes préparatoires a montré plusieurs anomalies dont les principales sont :

- Le système d'évaluation trimestriel a montré ses limites vu le nombre important d'évaluations effectuées (1 évaluation toutes les 5-6 semaines). Ceci provoque plusieurs interruptions des cours et une lourde charge d'organisation administrative.
- Le nombre d'heures attribuées à l'enseignement de l'informatique est très faible (2 heures par semaine en première année et 2 heures par quinzaine en deuxième année). En conséquence, l'étudiant assimile très peu la matière et ne lui accorde pas d'intérêt.
- Le coefficient affecté à l'informatique est également faible, voire minuscule, par rapport aux autres matières scientifiques telles que les mathématiques, les sciences physiques et la chimie. Pour cette raison, les étudiants ont montré la nonchalance vis-à-vis de la matière et le désintéressement total bien que c'est important pour un étudiant, futur

ingénieur, d'ancrer en lui les réflexes logiques et optimaux enseignés en informatique.

- A l'ère du numérique, l'outil informatique formel (Maple) enseigné actuellement en classes préparatoires s'est avéré inapproprié. En effet, cet outil a montré ses limites en programmation avancée. Il est donc temps de le remplacer si nous visons l'internationale.
- Les débouchés des étudiants de la filière BG sont différents des autres filières (MP, PC et Technologie). Il est donc nécessaire d'établir un programme spécifique pour cette filière.
- Plusieurs institutions préparatoires souffrent d'un manque de matériels. Dans certain cas, les étudiants n'ont pas eu la chance de faire des travaux pratiques durant une année entière. Dans d'autres institutions les séances de travaux pratiques (TPs) n'ont pas eu lieu dans les meilleures conditions (trois voire quatre étudiants par machine).

Suite à ces constatations, plusieurs recommandations ont été soulevées:

- Mettre en place un système d'évaluation semestriel au lieu d'un système trimestriel.
- Augmenter le nombre d'heures d'enseignement en deuxième année. Nous recommandons le passage de deux heures par quinzaine à deux heures par semaine en cours intégré (une heure de cours et une heure de TD).
- L'enseignement doit être effectué en laboratoire informatique, afin d'offrir la possibilité à l'étudiant de bien maîtriser l'outil informatique.
- Le coefficient de l'informatique doit passer au moins à 6 au lieu de 3 afin d'obliger l'étudiant à donner plus d'importance à la matière.
- Proposer un programme spécifique pour la filière BG.
- Améliorer les conditions de travail dans plusieurs institutions.
- Prévoir des formations pour les enseignants, sous forme d'école d'été ou des stages de formation.

Comme action à moyen et long terme nous proposons la création d'un comité pédagogique de pilotage mixte et de suivi des statistiques faisant interface entre les niveaux secondaires /préparatoires /écoles d'ingénieurs.

Ce comité aura pour mission d'auditer et d'établir des rapports sur la conduite du système et de transmettre les attentes des acteurs pédagogiques.

Programme proposé et conditions de mise en œuvre

Nous présentons dans ce qui suit, le programme élaboré suite à plusieurs réunions de travail de la commission de réforme du programme de l'enseignement de l'informatique en classes préparatoires.

Ce programme ne peut être concrétisé que si et seulement si :

- Une réponse favorable lui est accordée de la part du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.
- Un système d'évaluation semestriel est mis en place.
- Un enseignement hebdomadaire de deux heures de cours intégré en laboratoire informatique est adopté. Pour assurer une bonne assimilation, cet enseignement doit être effectué par groupe ne dépassant pas une vingtaine d'étudiants.
- Chaque laboratoire informatique est équipé d'un vidéoprojecteur.

Programme des filières

MP/PC/T

Programme de la première Année

Filières MP/ PC/T

Ce programme a été élaboré, sur la base du contenu et non pas sur la répartition des séances.

SEMESTRE I

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE I (6 h) Introduction Générale	Chapitre I : Architecture matérielle et logicielle 1. Architecture d'un ordinateur Mémoires. Microprocesseur. Périphériques. Canal de communication (bus). Démonstration / Démontage d'un PC de bureau (repérage et changements de certains composants, etc.). 2. Les systèmes d'exploitation Notion de système d'exploitation (avec des exemples de systèmes). Le multitâche. Identification des utilisateurs et contrôle d'accès. Système de fichiers.	2h
	Chapitre II : Système de numération 1. Représentation des nombres (naturels, relatifs et réels) dans une base b (binaire, octale, Hexadécimale ...). 2. Transcodage : Passage d'une base à l'autre. 3. Opérations élémentaires sur les nombres binaires. 4. Limites de la représentation (erreurs d'arrondi et de débordement de mémoire). <u>Travaux Dirigés</u> ▪ Représentation des nombres dans une base b. ▪ Passage d'une base à l'autre. ▪ Opérations élémentaires.	4h

Acquis

Au bout de cette partie l'étudiant doit :

- reconnaître les différents composants d'un ordinateur.
- connaître de façon générique les nouvelles architectures physiques, les nouvelles technologies et les différents types de systèmes d'exploitation (Windows, Linux, Android, IOS ...).
- savoir représenter un nombre dans une base **b** et convertir d'une base à l'autre.

PARTIE II (18h) Algorithmique et programmation	Chapitre III: Eléments d'algorithmique 1. Notion d'algorithme. 2. Opérations élémentaires : affectation, entrée/sortie. 3. Structures conditionnelles. 4. Structures itératives. 5. Notion de coût d'un algorithme et classes de complexité. <u>Travaux Dirigés</u> Applications introduisant la notion de coût d'un algorithme. Chapitre IV: Environnement de développement Python 3 1. Historique et raisons du choix du langage. 2. Mode interactif, mode Script, Aide en ligne. 3. Types élémentaires (classes int, str, float, bool, complex). 4. Opérations élémentaires sur les différents types élémentaires (approche classique/ approche orientée objet). 5. Notion de bibliothèque et import des packages prédéfinis (fonctions de bibliothèque). 6. Instructions élémentaires. 7. Structures conditionnelles. 8. Structures itératives. 9. Types composés : les types mutables (listes, dictionnaires, ensembles et array) et non mutables (tuples et chaînes). <u>Travaux Dirigés</u> ▪ Instructions de calculs (opérations arithmétiques, calculs avec import de	8h
		10h

	fonctions prédéfinis,...) en mode interactif (mode console). ▪ Ecrire et exécuter des programmes (utilisant des instructions élémentaires, des structures conditionnelles et des structures itératives) en mode script. ▪ Manipulation des types mutables et non mutables.	
<u>Acquis</u> – Maîtriser l'environnement Python. – Connaître l'allocation dynamique de la mémoire. – Savoir différencier entre les types mutables et non mutables en important le module copy. – Savoir calculer le coût d'un algorithme et différencier les classes de complexité (linéaire, quadratique, logarithmique, quasi-linéaire, exponentielle...).		
PARTIE III (4h) Programmation modulaire	Chapitre V: Les sous Programmes en algorithmique 1. Fonctions et procédures. 2. Passage de paramètres (Entrée, Sortie, E/S). 3. Variables locales et Variables globales.	4h
Total Semestre I (28H)		
SEMESTRE II		
Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE III Suite (10h) Programmation modulaire	Chapitre VI : Les fonctions en Python 1. Définition de fonctions par : ▪ <i>def</i> ▪ <i>lambda</i> 2. Variables locales et variables globales. 3. Notion de fonction locale. 4. Réutilisation de modules (import de fonctions). 5. Gestion des erreurs (bloc Try... Except). 6. Documentation des fonctions. 7. Coût de fonctions et classes de complexité. <u>Travaux Dirigés (Programmation Python)</u> Exercices d'Arithmétiques, nombres premiers, nombres parfaits, nombres amis, calcul de	6h

	PGCD, PPCM, multiplication égyptienne, recherche de zéro d'une fonction par dichotomie, algorithme de Newton... Chapitre VII : La Récursivité 1. Définition de la récursivité. 2. Avantages et inconvénients de la récursivité (mémoire et temps de calcul). <u>Travaux Dirigés</u> Suites récurrentes, exponentiation rapide, algorithme d'Euclide, fractales...	4h
<u>Acquis</u> – Savoir écrire un programme en python itératif et/ou récursif. – Maîtriser le passage de paramètres. – Connaître la notion de critère d'arrêt et d'appel récursif.		
PARTIE IV (8h) Les itérables et algorithmes de tri	Chapitre VIII : Manipulation des itérables en Python 1. Les méthodes de recherche (liste, chaîne de caractères ...). 2. Les Algorithmes de tri <u>Travaux Dirigés (Programmation Python)</u> Applications avec étude des coûts.	8h
<u>Acquis</u> – Savoir manipuler les conteneurs itérables sous python. – Maîtriser les algorithmes de recherches et de tri (itératifs / récursifs) avec leurs coûts.		
PARTIE VI (10h) Manipulation des fichiers et Simulation Numérique	Chapitre IX : Manipulation des fichiers Python 1. Ouverture et fermeture d'un fichier (texte/binaire). 2. Lecture et écriture depuis/dans un fichier texte et image. Chapitre X : Simulation numérique – Utiliser les classes array et ndarray de la bibliothèque numpy et l'ensemble de fonctions d'algèbre linéaire. – Simulation d'expériences et traçage de courbes (matplotlib). – Résolution d'équations différentielles du	4h 6h

	premier ordre (méthode d'Euler) (scipy). – Méthodes de recherche du zéro d'une fonction (Méthode Newton, Méthode dichotomique, ...). – Méthodes d'intégrations (Méthode des rectangles, Méthode des trapèzes,...) (scipy).	
<u>Acquis :</u> – Savoir manipuler les fichiers avec Python : récupérer et/ou stocker des données depuis des fichiers pour simulation numérique. – Connaître les principales bibliothèques numériques numpy, scipy et matplotlib. – Savoir résoudre des problèmes scientifiques avec Python.		
Total Semestre II (28 H)		

Programme de la deuxième Année

Filières MP/ PC/PT

SEMESTRE I

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE I (8h) Rappel et Structures de Données Avancées	Chapitre 0 : Rappel général – Programmation Python. – Syntaxe et types de bases. <u>Travaux Dirigés</u> Exercices extraits de concours.	4h
	Chapitre I : Structures de données avancées 1. Notion de Pile. 2. Notion de File. <u>Travaux Dirigés</u> ▪ Modélisation avec les listes sous Python. ▪ Notation post fixée, expression bien parenthésée.	4h
<u>Acquis</u> Savoir modéliser les piles et les files sous python au moyen de listes.		
PARTIE II (8h) Programmation Orientée Objet	Chapitre II: Approche orientée objet et Programmation avec Python 1. Objet 2. Classe (Attributs, méthodes, constructeur/ destructeur ...). 3. Encapsulation. 4. Héritage. <u>Travaux dirigés:</u> ▪ Représentation des formes géométriques (droite, carré, triangle, etc.). ▪ Jeu de carte. ▪ Etc.	8h

<u>Acquis</u> - Savoir créer et instancier des classes. - Savoir programmer en utilisant la notion d'objet (attributs et méthodes). - Connaître la notion d'héritage.		
PARTIE II (12h) Les bases de données	Chapitre III: Introduction aux bases de données 1. Limite des fichiers 2. Notions de bases de données (définition) 3. Les systèmes de gestion de bases de données - Définition. - Objectifs. 4. Architecture des SGBD - Architecture client/ serveur - Architecture trois-tiers 5. Modèle de données - Le modèle relationnel: l'algèbre relationnelle - La relation : ✓ Schéma de la relation (Attributs et contraintes d'intégrité). ✓ Opérations unaires: projection, sélection. ✓ Opérations binaires : intersection; union; jointure; différence.	6h
	Chapitre IV: Passage à la pratique (module SQLite3 de Python) 1. Création d'une base. 2. Création de tables (schéma de la table, alimentation). 3. Langage SQL: manipulation des données - Projection - Sélection - Opérateurs SQL ✓ Clauses(fonctions d'agrégations, having, order by, group by, etc.) ✓ Union, intersection, différence, jointure <u>Travaux Dirigés</u> Requêtes simples et imbriquées.	6h

Acquis

- Assimiler les notions de base de données et les systèmes de gestion de base de données.
- Savoir créer et alimenter une base de données avec Python.
- Savoir écrire des requêtes en langage algébrique et langage SQL.

Total Semestre I**(28 H)****SEMESTRE II**

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE III (22h) Simulation numérique	Thèmes à aborder: 1. Algèbre linéaire: Calcul matriciel, méthode de pivot de Gauss, Décomposition LU 2. Cryptographie: Méthode Hill, méthode RSA, codage de Hamming ... 3. Traitement d'image: Compression Dilatation 4. Problèmes d'interpolations: Interpolation de Lagrange 5. Programmation Orientée Objet et interfaces graphiques	22h

Acquis

L'étudiant doit être capable de résoudre des problèmes liés aux mathématiques, aux sciences physiques, ou à n'importe quel autre domaine cité ci-haut.

Programme de la filière BG

Programme de la première Année

Filière BG

Ce programme a été élaboré, sur la base du contenu et non pas sur la répartition des séances.

SEMESTRE I

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE I (6 h) Introduction Générale	Chapitre I : Architecture matérielle et logicielle 1. Architecture d'un ordinateur Mémoires. Microprocesseur. Périphériques. Canal de communication (bus). Démonstration / Démontage d'un PC de bureau (repérage et changements de certains composants, etc.). 2. Les systèmes d'exploitation Notion de système d'exploitation (avec des exemples de systèmes). Le multi tâche. Identification des utilisateurs et contrôle d'accès. Système de fichiers.	2h
	Chapitre II : Système de numération 1. Représentation des nombres (naturels, relatifs et réels) dans une base b (binaire, octale, Hexadécimale ...). 2. Transcodage : Passage d'une base à l'autre. 3. Opérations élémentaires sur les nombres binaires. 4. Limites de la représentation (erreurs d'arrondi et de débordement de mémoire). <u>Travaux Dirigés:</u> ▪ Représentation des nombres dans une base b. ▪ Passage d'une base à l'autre.	4h

	▪ Opérations élémentaires.	
<u>Acquis</u> Au bout de cette partie, l'étudiant doit : – reconnaître les différents composants d'un ordinateur. – connaître de façon générique les nouvelles architectures physiques, les nouvelles technologies et les différents types de systèmes d'exploitation (Windows, Linux, Android, IOS ...). – savoir représenter un nombre dans une base <i>b</i> et convertir d'une base à l'autre.		
PARTIE II (18h) Algorithmique et programmation	Chapitre III: Eléments d'algorithmique 1. Notion d'algorithme. 2. Opérations élémentaires : affectation, entrée/sortie. 3. Structures conditionnelles. 4. Structures itératives. 5. Notion de coût d'un algorithme et classes de complexité. <u>Travaux Dirigés</u> Applications introduisant la notion de coût d'un algorithme.	8h
	Chapitre IV: Environnement de développement Python3 1. Historique et raisons du choix du langage. 2. Mode interactif, mode Script, Aide en ligne. 3. Types élémentaires (classes int, str, float, bool, complex). 4. Opérations élémentaires sur les différents types élémentaires (approche classique/ approche orientée objet). 5. Notion de bibliothèque et import des packages prédéfinis (fonctions de bibliothèque). 6. Instructions élémentaires. 7. Structures conditionnelles. 8. Structures itératives 9. Types composés : les types mutables (listes, dictionnaires, ensembles et array) et non mutables (tuples et chaînes).	10h

	<u>Travaux Dirigés</u> ▪ Instructions de calculs (opérations arithmétiques, calculs avec import de fonctions prédéfinies,...) en mode interactif (mode console). ▪ Écrire et exécuter des programmes (utilisant des instructions élémentaires, des structures conditionnelles et des structures itératives) en mode script. ▪ Manipulation des types mutables et non mutables.	
<u>Acquis</u> – Maîtriser l'environnement Python. – Connaître l'allocation dynamique de la mémoire. – Savoir différencier entre les types mutables et non mutables en important le module copy. – Savoir calculer le coût d'un algorithme et différencier les classes de complexité (linéaire, quadratique, logarithmique, quasi-linéaire, exponentielle...).		
PARTIE III (4h) Programmation modulaire	Chapitre V: Les sous Programmes en algorithmique 1. Fonctions et procédures. 2. Passage de paramètres (Entrée, Sortie, E/S). 3. Variables locales et Variables globales.	4h
Total Semestre I (28 H)		
SEMESTRE II		
Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE III Suite (8 h) Programmation modulaire	Chapitre VI : Les fonctions en Python 1. Définition de fonctions par : ✓ <i>def</i> ✓ <i>lambda</i> 2. Variables locales et variables globales. 3. Notion de fonction locale. 4. Réutilisation de modules (import de fonctions). 5. Gestion des erreurs (bloc Try... Except). 6. Documentation des fonctions. 7. Coût de fonctions et classes de complexité.	8h

	<u>Travaux Dirigés (Python)</u> Exercices d'Arithmétique, nombres premiers, nombres parfaits, nombres amis, calcul de PGCD, PPCM, multiplication égyptienne, recherche du zéro d'une fonction par dichotomie, algorithme de Newton...	
<u>Acquis</u> - Savoir écrire un programme en python. - Maîtriser le passage des paramètres.		
PARTIE IV (10h) Les itérables et algorithmes de tri	Chapitre VII : Les itérables 1. Les méthodes de recherche (liste, array, chaîne de caractères ...). 2. Les Algorithmes de tri (sélection, insertion, à bulles...). <u>Travaux Dirigés (Programmation Python) :</u> Applications avec étude des coûts.	10h
<u>Acquis</u> - Savoir manipuler les tableaux en algorithmique et en python. - Connaître les algorithmes de recherches et de tri avec leurs coûts.		
PARTIE VI (10h) Manipulation des fichiers et Simulation Numérique	Chapitre VIII : Manipulation des fichiers Python 1. Ouverture et fermeture d'un fichier. 2. Lecture et écriture dans un fichier texte et image. Chapitre IX : Simulation numérique – Simulation d'expériences et traçage de courbes (matplotlib). – Biologie : Algorithme qui permet de déterminer la séquence d'une protéine à partir d'une séquence d'ADN en passant par l'ARN messenger en utilisant le code génétique et déterminer la structure tridimensionnelle de la protéine. – Géologie: Numériser les courbes de niveau à partir d'une carte topographique pour générer un modèle numérique de terrain (MNT) Transformer une surface plane (image) en une structure tridimensionnelle (relief).	4h 6h

Acquis

- Savoir manipuler les fichiers avec Python.
- Connaître les principales bibliothèques numériques numpy, scipy et matplotlib.
- Savoir résoudre des problèmes scientifiques avec Python.

Total Semestre II**(28 H)**

Programme de la deuxième Année

Filière BG

SEMESTRE I

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE I (8h) Rappel et Structures de données avancées	Chapitre 0 : Rappel général – Programmation Python. – Syntaxe et types de base. – Les tableaux. <u>Travaux Dirigés</u> Exercices extraits de concours.	8h
PARTIE II (20h) Les bases de données	Chapitre I: Introduction aux bases de données 1. Limite des fichiers 2. Notions de bases de données (définition) 3. Les systèmes de gestion de base de données – Définition. – Objectifs. 4. Architectures des SGBD : – Architecture client/ serveur – Architecture trois-tiers 5. Modèle de données – Le modèle relationnel : l'algèbre relationnelle. – La relation : ✓ Schéma de la relation (Attributs et contraintes d'intégrité). ✓ Operations unaires: projection, sélection. ✓ Opérations binaires : intersection; union; jointure; différence. Chapitre II: Passage à la pratique (module SQLite de Python) – Création d'une base. – Création de tables (schéma de la table,	10h

	alimentation). – Langage SQL : manipulation des données ▪ Projection ▪ Sélection ▪ Opérateurs SQL ✓ Clauses (<i>fonctions d'agrégations, having, order by, group by, etc.</i>) ✓ Union, intersection, différence, jointure <u>Travaux Dirigés</u> Requêtes simples et imbriquées.	10h
--	--	-----

Acquis

- Assimiler la notion de base de données et les systèmes de gestion de base de données.
- Savoir créer et alimenter une base de données avec Python.
- Savoir écrire des requêtes en langage algébrique et langage SQL.

Total Semestre I (28 H)

SEMESTRE II

Titre des chapitres & contenu		Charge Horaire Approximative
PARTIE III (22h) Simulation Numérique	Thèmes à aborder: – Traitement d'image: ▪ Compression ▪ Dilatation – Biologie : ▪ Applications liées à la bioinformatique. ▪ La transpiration : ✓ Estimer la surface d'une forme irrégulière (feuille). ✓ Calculer le déficit hydrique par la feuille et la vitesse de transpiration. – Géologie : ▪ Simulation numérique de la déformation des matériaux en utilisant un modèle (ANSYS).	22h

Acquis

Savoir résoudre des problèmes biologique et géologique à l'aide de l'outil informatique

Documents de référence et liens utiles

- [1] Programmes des classes préparatoires aux Grandes Écoles Filières scientifique. www.enseignementsup-recherche.gouv.fr. Réforme 2013
- [2] [Amar Oussalah](#), [Thierry Audibert](#). Informatique en classes préparatoires scientifiques 1re et 2e années : Programmation et calcul scientifique en Python et Scilab
- [3] [Benjamin Wack](#) , [Sylvain Conchon](#) , [Judicaël Courant](#) , [Marc de Falco](#) , [Gilles Dowek](#) , [Jean-Christophe Filliâtre](#) , [Stéphane Gonnord](#) . Informatique pour tous en classes préparatoires aux grandes écoles : Manuel d'algorithmique et programmation structurée avec Python, Nouveaux programmes 2013, Voies MP, PC, PSI, PT, TPC et TS
- [4] Gérard Swinnen, Apprendre à programmer avec Python 3.
- [5] Eric Le Nagard. Informatique : Initiation à l'algorithmique en Scilab et Python.
- [6] Alexandre Casamayou-Boucau, Guillaume Connan, Pascal Chauvin. Programmation en Python pour les mathématiques.
- [7] Site officiel de python. www.python.org. Dernier visite Aout 2015.
- [8] Site officiel de scipy. www.scipy.org/. Dernier visite Aout 2015.
- [9] Site officiel de Numpy. www.numpy.org/. Dernier visite Aout 2015.
- [10] Site officiel de Sqlite. www.sqlite.org. Dernier visite Aout 2015.