

Chapitre 1

Éléments fondamentaux de l'informatique

Monsieur Moore, ingénieur renommé de la société Intel, a énoncé il y a plus de 20 ans une conjecture qui est toujours vraie aujourd'hui : "Les ordinateurs et les logiciels qui les peuplent verront leur capacité doubler tous les 18 mois". La révolution digitale s'est produite sous nos yeux et ses effets sont encore en train de se développer avec des applications qui changent notre quotidien. Les évolutions viennent de partout, bien sûr aux niveaux des capacités des machines, mais aussi au niveau des algorithmes qui s'adaptent pour traiter des quantités d'information de plus en plus grandes.

L'informatique est devenue omniprésente dans la plupart des organisations et des entreprises. Nous avons pu observer l'arrivée des ordinateurs personnels, les fameux *Personal Computer* ou PC sur lesquels les *suites bureautiques* se sont démocratisées. Les *progiciels* ou, en anglais, ERP – *Enterprise Resource Planning* ont ensuite pris rapidement de l'importance pour la gestion centralisée des processus administratifs liés à la comptabilité, aux ressources humaines, à la facturation et à la gestion de la production.

Aujourd'hui, l'informatique s'étend encore et touche au cœur des métiers, tout horizon confondu. Si nous prenons l'exemple d'un hôpital, il ne s'agit plus seulement d'utiliser les systèmes d'information pour gérer les rendez-vous, l'occupation des lits et la réservation des équipements médicaux. L'informatisation touche aussi les données mesurées sur les patients incluant par exemple la numérisation des radiographies, leur visualisation et leur archivage dans les dossiers patients. Récemment, les systèmes auto-apprenants, en anglais *Machine Learning*, révolutionnent le traitement automatique de ces données pour fournir aux médecins des systèmes d'*aides à la décision*.

Les applications où les calculs et les données sont distribués sur différentes machines ont également connus une croissance fulgurante. Ces tendances sont résumées à travers différents termes d'actualité tels que l'informatique dans *le cloud* et le *big data*. La capacité d'interconnexion des ordinateurs s'est aussi rapidement améliorée, bien sûr à travers le réseau Internet, mais aussi via les réseaux sans fils comme la 4G. La hausse des performances a permis la miniaturisation et nous avons vu débarquer les smart phones, véritables mini-ordinateur, au creux de nos mains et quasi en permanence dans nos poches. Le partage de l'information est devenu instantané et omni-présent.

En observant cette révolution numérique, nous sommes convaincus qu'il est aujourd'hui important, voire essentiel pour notre société, de renforcer l'enseignement des

sciences informatiques au niveau des Hautes Ecoles. Un curriculum scientifique se doit d'aller au delà d'une simple utilisation passive des logiciels liés aux différents métiers. Il faut aujourd'hui inclure une compréhension plus avancées des fondements de l'informatique et de la programmation des logiciels. Ceci permettra aux étudiants non seulement de faire une meilleure utilisation des logiciels, mais aussi de les adapter ou d'en créer de nouveaux en les programmant.

Le but de ce premier chapitre est de donner une introduction aux systèmes informatisés et, de façon plus générale, aux sciences de l'informatique. Nous allons définir différents concepts qui sont les fondements nécessaires pour la compréhension du reste de la matière dédiée à la **Programmation Scientifique**. Nous allons premièrement résumer le fonctionnement d'un ordinateur, des systèmes d'exploitation et des logiciels. Ensuite nous allons donner un aperçu de ce que nous appelons la *hiérarchie des données*, essentielle à la compréhension du fonctionnement d'un système d'information. Ensuite, les différents types de logiciels seront présentés avec un focus sur les logiciels scientifiques. Finalement, nous faisons un rappel du fonctionnement des systèmes de fichiers.

1.1 Informatique

Le terme informatique est la contraction des 2 termes **information** et **automatique**. Autrement dit, l'informatique est le traitement automatique de l'information.

Définition 1.1.1

L'**informatique** est la science du traitement *automatique* de l'*information* par des machines. Par extension, l'informatique comprend toutes les techniques qui sont relatives au traitement de l'information, comme par exemple la logique, le calcul numérique, la théorie de l'information, ou encore l'algorithmique.

Les domaines scientifiques inclus dans l'informatique sont nombreux et divers. Une manière de les structurer est de considérer les deux branches suivantes : l'informatique théorique et l'informatique pratique [10]. L'informatique *théorique* concerne la définition de concepts et modèles informatiques comme par exemple la théorie des langages, l'algorithmique, la complexité, la calculabilité, la théorie de l'information ou encore la théorie des graphes. L'informatique *pratique* s'intéresse quand à elle aux techniques concrètes de développement et de mise en œuvre sur le terrain, incluant les aspects pratiques du secteur industriel des *Technologies de l'Information et de la Communication* ou TIC.

Les TIC couvrent à la fois les procédés de l'ingénierie logicielle (écriture de logiciel, mise en place d'architectures de systèmes), l'ingénierie matérielle (intégration électronique, systèmes de télécommunication) et les services liés à l'utilisation de ceux-ci (maintenance et assistance). En Suisse, les TIC constituent le cinquième secteur économique privé du pays, avec seuls le commerce de gros et les services financiers qui lui sont substantiellement supérieurs [1].

→ **Remarque.** Le terme *Information Technology* (IT) est souvent utilisé de façon équivalente au terme *Information and Communication Technology* (ICT). Le terme ICT est néanmoins supposé être plus large car englobant les aspects en lien avec les télécommunications. Nous observons

également que le terme IT est plus présent dans les entreprises alors que le terme ICT semble être plus utilisé dans les milieux académiques et de recherche.

Par extension, l'informatique est aussi associée à des branches plus fondamentales des mathématiques comme le calcul numérique qui lui-même s'appuie sur les mathématiques discrètes. Bien qu'avant tout scientifique, l'informatique est aussi une activité créatrice qui permet de trouver des solutions innovantes à des problèmes concrets. Le développement d'un logiciel adéquat nécessite deux composantes, d'une part une bonne compréhension du problème qui doit être résolu et d'autre part une bonne maîtrise des concepts et outils de base en informatique.

Le terme *systèmes d'information* quand à lui est souvent associé aux logiciels d'entreprise qui traitent les informations centrales de l'entreprise. Ces informations sont liées, par exemple, aux processus de fabrication, de facturation et de ressources humaines. Nous pouvons néanmoins définir un *système d'information* de façon plus large comme tout système qui permet de traiter de l'information. Cette définition plus générale inclut les systèmes de facturation dans les entreprises, les logiciels de traduction automatique de texte, le stockage des images numériques dans les hôpitaux, les réseaux sociaux comme Facebook, etc.

Définition 1.1.2

Un **Système d'Information** est un système, généralement informatisé, dont le but est de traiter l'information. Par traitement, on entend la mémorisation, la restitution, la transmission, la structuration, la transformation et l'exploitation de l'information.

La table 1.1 résume les processus de traitement de l'information en faisant correspondre les avantages d'un traitement automatique.

Traitement de l'information	Traitement automatique
Mémorisation et transmission	Grande quantité de données
Restitution et recherche	Grande vitesse de traitement
Structuration et transformation	Processus répétitif et coût avantageux
Décision automatique	Fiabilité

TABLE 1.1 : Résumé des processus de traitement de l'information et du traitement automatique

1.2 Ordinateur et logiciel

Les ordinateurs peuvent effectuer des calculs et prendre des décisions logiques beaucoup plus rapidement que les humains. Un ordinateur tel que nous pouvons l'acheter aujourd'hui dans les commerces est capable d'exécuter des milliards de calculs dans

une seconde, ce qui représente probablement plus de calculs que ce qu'un humain peut réaliser dans une vie. Les super-ordinateurs quand à eux ont font actuellement l'objet d'une compétition mondiale acharnée. C'est à celui qui réalisera le plus d'instructions par secondes¹. Les meilleurs ont récemment dépassé les 100 petaflops – 100×10^{15} *floating point operations per seconds*, entre autre le Sunway TaihuLight du *Supercomputing Center Wuxi*². Autrement dit, cet ordinateur est capable de réaliser en une seconde plus de 10 millions de calculs pour chaque personne sur notre planète terre.

Le coût des ordinateurs n'a cessé de diminuer, permettant des développements rapide du matériel et des logiciels. Il en va de même pour la taille des composantes électroniques intégrés dans les processeurs. Pour se donner un ordre d'idée, nous avons aujourd'hui dans une montre intelligente de quelques dizaines de gramme une capacité de calcul équivalente à un ordinateur de la fin des années 90 qui pesait quelques dizaines de kilos. Les progrès de la technologie des processeurs à silicium ont permis ce développement et, le silicium étant un des matériaux les plus abondants sur la terre, les ordinateurs sont devenus des biens de consommation courants.

Le schéma en Figure 1.1 donne une vue éclatée d'un ordinateur personnel ou, en anglais, *personal computer*, ou encore, de façon abrégée, PC. . Un ordinateur est composé de divers composantes électroniques définissant le matériels informatique, en anglais le *hardware*. Ces composantes permettent à l'utilisateur d'entrer de l'information (e.g. les *input units* comme le clavier ou la souris) et de recevoir le résultat d'un travail, d'un calcul ou d'un processus (e.g. les *output units* comme les écrans ou les imprimantes).

Définition 1.2.1

Un **ordinateur** se définit comme le matériel informatique comprenant les composantes nécessaires à son fonctionnement autonome. En anglais : *computer*.

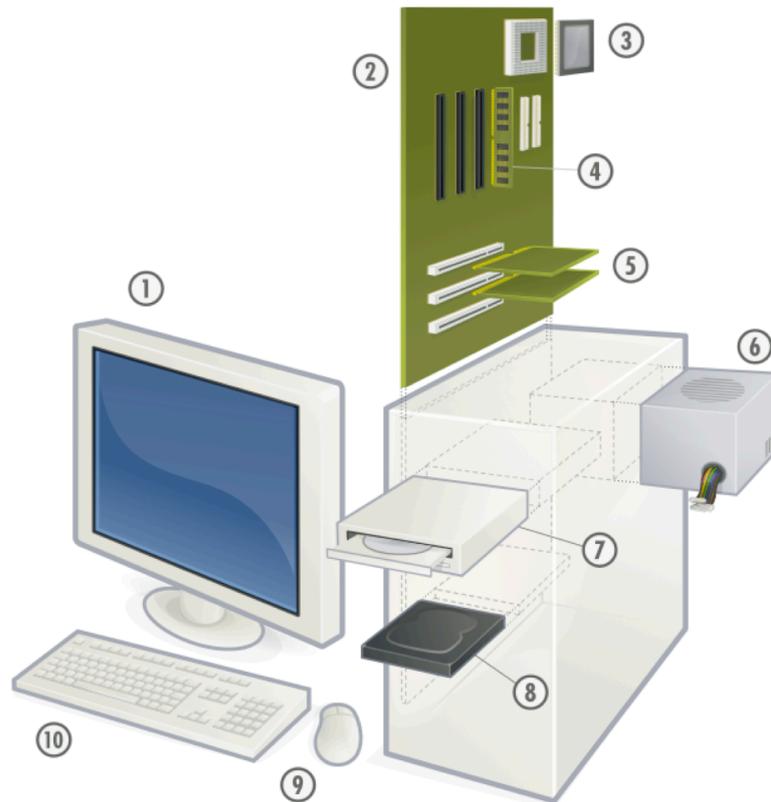
Définition 1.2.2

Le **matériel informatique** comprend les éléments physiques employés pour le traitement de l'information. En anglais : *hardware*.

Les ordinateurs traitent les données sous le contrôle de séquences d'instructions contenues dans les logiciels. Ces "programmes" d'instructions guident l'ordinateur dans les calculs préparés par les développeurs. Les logiciels sont donc les applications qui sont exécutées sur un ordinateur et qui sont utilisées pour rendre différents services. De façon plus détaillée, un logiciel est composé d'instructions qui sont exécutées par l'ordinateur. Ces instructions composent, en quelque sorte, des "partitions de musique" jouées par l'ordinateur. Les instructions respectent une syntaxe et une sémantique propre au langage de programmation utilisé.

¹<http://www.top500.org>

²<http://www.nscwx.cn/wxcyw/>



Source: Wikimedia Commons - Gustavb

FIGURE 1.1 : Structure d'un ordinateur incluant typiquement le matériel informatique suivant : (1) écran, (2) carte mère, (3) unité centrale, (4) mémoire vive (RAM), (5) cartes périphériques, (6) bloc d'alimentation, (7) lecteur de disque optique, (8) disque dur, (9) souris, (10) clavier.

Définition 1.2.3

Un **logiciel** est un programme informatique composé de listes d'instructions exécutables par un ordinateur. Le logiciel se définit également à partir des procédés, des données et de la documentation qui sont relatifs à son fonctionnement. En anglais : *software*.

Les logiciels actuels peuvent contenir des millions d'instructions. Ces instructions sont exécutées par l'ordinateur et plus spécifiquement par le microprocesseur qui est le composant matériel en charge d'interpréter et d'exécuter ces instructions. Les instructions sont généralement exécutées de façon séquentielle, à savoir l'une après l'autre. Certains blocs d'instructions sont, pour certains, exécutés de manière répétitive.

→ **Remarque.** La rapidité d'exécution des instructions permet aux ordinateurs de basculer très rapidement d'un logiciel à un autre, donnant l'impression d'une exécution simultanée des programmes et des tâches au sein d'un même programme. Il est possible par exemple d'écouter de la musique tout en éditant un texte. Cette propriété est appelée multitâche (multitasking en anglais). Nous pouvons également noter que la présence de plusieurs processeurs au sein d'un même ordinateur ou encore l'arrivée de processeurs dits multi-cœur, permet effectivement une exécution en parallèle de plusieurs blocs d'instructions. C'est un des rôles du système d'exploitation (voir description plus loin) de gérer les priorités, i.e. la fréquence et l'ordre des basculements vers les différents logiciels.

1.3 Structure d'un ordinateur

La Figure 1.2 montre un schéma de principe simplifié d'un ordinateur. Sur ce schéma apparaissent différents blocs correspondants aux éléments matériels qui composent un ordinateur. Nous définissons ces éléments ci-dessous et détaillons leur fonctionnement après les définitions.

Définition 1.3.1

Le **Microprocesseur** ou l'**unité centrale** est la partie de l'ordinateur en charge du contrôle, de l'interprétation et de l'exécution des instructions. En anglais : *central processing unit* ou *CPU*.

Définition 1.3.2

La **Mémoire** est la partie de l'ordinateur en charge de l'enregistrement et de la restitution des données. En anglais : *memory unit*.

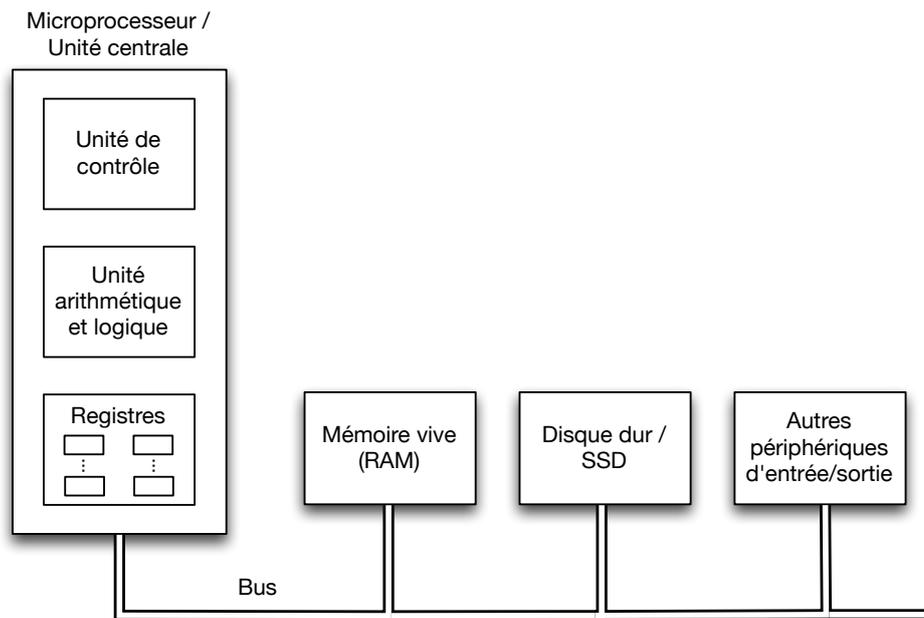


FIGURE 1.2 : Schéma de principe d'un ordinateur avec un microprocesseur, de la mémoire vive, un disque dur et d'autres périphériques d'entrée/sortie. Les différents éléments sont reliés entre eux par un ou plusieurs bus de communication.

Définition 1.3.3

Les **Bus** sont les parties de l'ordinateur en charge de la communication entre les composantes matérielles de l'ordinateur.

Définition 1.3.4

Les **Composantes Périphériques** sont les parties de l'ordinateur en charge des interactions "entrée" et "sorties", typiquement avec l'utilisateur. En anglais : *input and output units*.

→ **Remarque.** Pour illustrer le schéma de la Figure 1.2 et ces différentes définitions, nous pouvons prendre l'exemple d'un programme Python qui est lancé par l'utilisateur. Le but du programme est de tracer la fonction $y = \sin(x)$ pour x allant de 0 à 4π . L'ordinateur va exécuter des opérations selon une séquence (volontairement simplifiée) qui pourrait être la suivante :

1. Après avoir lancé la console Python et tapé le nom du programme, l'utilisateur presse la touche "entrée" du clavier pour lancer son exécution. Le clavier est un périphérique d'entrée et les signaux y relatifs vont transiter sur le bus de données vers le microprocesseur et plus précisément l'unité de contrôle.

2. Les instructions du programme Python sont chargées par le microprocesseur depuis le fichier contenant le programme. Le fichier réside sur une mémoire auxiliaire de type disque dur. Les instructions Python vont donc transiter depuis le disque dur vers la RAM en utilisant le bus de données.
3. Les instructions sont ensuite interprétées et exécutées les unes après les autres par le microprocesseur (et plus précisément l'unité de contrôle) qui les lit au fur et à mesure depuis la RAM.
4. Certaines instructions Python auront pour but de créer une liste de couples de valeurs (x, y) nécessaires pour tracer la fonction. L'unité arithmétique et logique sera sollicitée pour cela, éventuellement en utilisant les registres pour stocker les résultats intermédiaires.
5. L'unité de contrôle ira ensuite stocker les valeurs de la liste finale des couples (x, y) dans la RAM, de nouveau en utilisant le bus.
6. Une fois la liste de valeurs (x, y) complète, les instructions Python suivantes viseront le tracé d'une courbe reliant les points successifs de la liste. Pour cela, le microprocesseur communiquera avec l'écran, un périphérique de sortie, pour donner les instructions de tracé des valeurs successives de la listes (lues depuis la RAM et transitant de nouveau via le bus).

Nous devons souligner ici que la séquence réelle d'opérations exécutées par l'ordinateur est largement plus complexe et pourrait différer passablement de ce qui est décrit ci-dessus. En pratique, le système d'exploitation entre également en jeu. Il s'agit d'un logiciel "intermédiaire" gérant les ressources de l'ordinateur, entre autre l'accès à la mémoire et les interactions avec les périphériques d'entrée et de sortie. Le système d'exploitation va également gérer les autres programmes qui s'exécutent de façon concurrente à notre programme Python. Cette séquence d'opérations va donc différer en fonction de la configuration de l'ordinateur, du système d'exploitation et des autres programmes qui sont potentiellement en cours d'exécution. Nous donnerons plus de détails sur le rôle du système d'exploitation en Section 1.5.

Microprocesseur Le microprocesseur ou unité centrale est un circuit intégré, le coeur de l'ordinateur. Il est composé d'une unité arithmétique et logique (UAL), d'une unité de contrôle (UC) et souvent d'éléments de mémoire rapides situés sur le même circuit intégré comme par exemple les registres et la mémoire cache. L'UAL réalise les opérations arithmétiques (additions, soustraction, ...), logiques (AND, OR, XOR, ...) et relationnelles (comparaison plus petit que, plus grand que, etc). C'est l'UAL qui réalise les calculs dans l'exécution d'un logiciel. L'UC gère le flot d'instruction à exécuter. Une instruction est lue dans la mémoire, selon son type, l'UC ordonne à l'UAL d'effectuer un calcul. Une fois l'instruction exécutée, l'UC passe à l'instruction suivante ou bien va à une autre instruction en fonction des résultats du calcul précédent. L'UC est en quelque sorte le chef d'orchestre qui gère les exécutions des instructions ainsi que la communication entre les différents types de mémoire et l'UAL.

Mémoire Différents types de mémoire coexistent dans un ordinateur (voir Section 1.4). La *mémoire centrale*³ permet un accès rapide aux données typiquement pour effectuer des traitements. En général cette mémoire est volatile, ce qui veut dire que les informations disparaissent lorsque l'ordinateur est éteint. La *mémoire auxiliaire*⁴ est en quelque sorte l'entrepôt long-terme des données, généralement pour un enregistrement non-volatile et ré-inscriptible comme les disques durs, les dvd ou bandes magnétiques. De façon générale, une mémoire peut être schématisée comme une suite de cellules numérotées capables chacune de stocker des informations.

³Ou encore mémoire primaire ou mémoire vive

⁴Ou encore mémoire secondaire

Bus Les différents éléments d'un ordinateur communiquent ensemble à travers ce qu'on appelle des bus. Dans sa forme la plus simple, un bus est un ensemble de fils électriques reliant les éléments de l'ordinateur pour transporter des informations. Chaque fil encode des séquences temporelles de valeur binaire, soit à 1 ou à 0 en fonction de la tension électrique imposée sur le fil. On distingue différents bus pour le transport de données (bus de données), pour la sélection des adresses mémoire (bus d'adresse) et pour les signaux de synchronisation (bus de contrôle). Ces différents bus fonctionnent suivant des standards qui ont évolué au cours du temps. Certains bus sont plutôt dédiés à une communication interne comme PCI, ISA, SCSI. D'autres sont dédiés à une communication avec l'externe comme USB ou Thunderbolt.

Périphériques Les périphériques regroupent de façon générique les différents éléments "extérieurs" au microprocesseur et à la mémoire centrale. On les qualifie souvent de périphériques d'entrées/sorties car ce sont eux qui permettent à l'ordinateur de communiquer avec l'extérieur. Dans les périphériques les plus courants, on retrouve le clavier, l'écran, la carte réseau, la souris, les imprimantes, etc. Les périphériques doivent convertir les informations récupérées de l'extérieur et les rendre compréhensibles par l'ordinateur. Pour cela, on parle de codage/décodage de l'information, souvent pris en charge par un logiciel spécifique appelé le pilote (*driver* en anglais).

1.4 Types de mémoire

La mémoire est un élément clé des ordinateurs actuels. Nous pouvons résumer les caractéristiques principales des mémoires comme suit :

- On parle de mémoire *volatile* lorsque l'information est perdue si l'ordinateur est éteint.
- Une mémoire est dite *permanente* ou *non-volatile* lorsqu'elle permet de stocker de l'information à plus long terme, même lorsque l'ordinateur est hors tension.
- D'autres éléments de mémoire sont dits *immuables* lorsqu'il est impossible d'en changer le contenu.
- Un paramètre important d'une mémoire est le *temps d'accès* qui définit le temps moyen pour accéder à l'information stockée et pour commencer sa lecture. Plus le temps d'accès est petit et plus la mémoire est performante.
- La *capacité* représente la quantité d'information qui peut être stockée sur le support. L'unité de mesure de la capacité est l'*octets*, un regroupement de 8 bits codant une information (en anglais : byte). Les capacités ont grandi au fil des années et l'usage est d'utiliser des multiples de l'octet comme le kilooctet, mégaoctet, gigaoctet, teraoctet, etc.
- Le *débit* mesure la quantité d'information qui peut être lue par unité de temps. Le débit définit en quelque sorte la largeur du tuyau dans lequel passe l'information. Certaines applications multimédias nécessitent un grand débit, comme par exemple la lecture d'un film.
- Le critère de *portabilité* est lié à la possibilité de déconnecter et de déplacer physiquement le support de mémoire d'un ordinateur à un autre.
- Finalement le *prix par unité de stockage* est un critère économique définissant le

ration entre le prix et la quantité d'information qu'il est possible de stocker.

→ **Remarque.** On qualifie une mémoire du terme *auxiliaire* lorsque l'équipement de mémoire est externe (ou "externalisable") par rapport à la carte mère. Les disques durs, les DVD, les memory sticks sont des mémoires auxiliaires non-volatiles qui permettent de stocker de l'information à long terme.

Une mémoire idéale aurait à la fois un petit temps d'accès, un grand débit, une grande capacité et un faible prix. La performance et le prix sont bien sûr corrélés et il faut donc souvent réaliser des compromis ou faire des choix stratégiques car, suivant les applications, la quantité d'information à stocker et son traitement peuvent être très divers. Nous allons donc retrouver différents supports de mémoire qui co-existent dans nos ordinateurs. Nous en illustrons quelques uns sur la Figure 1.3.

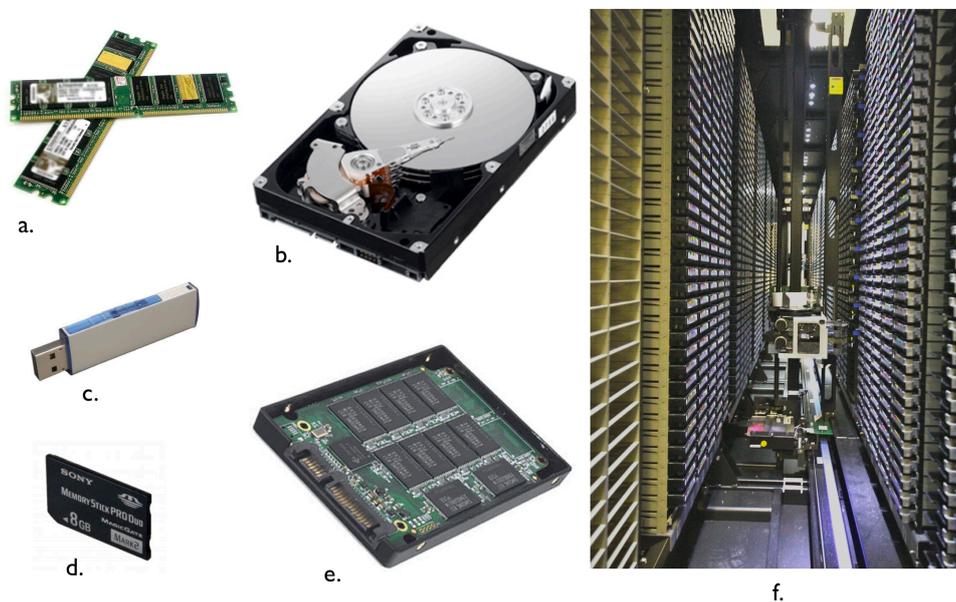


FIGURE 1.3 : Illustration de différents types de mémoire : (a.) Barrettes de RAM, (b.) Disque dur, (c.) Memory stick USB, (d.) Memory stick Flash Card, (e.) Solid State Disk, (f.) Système à bandes magnétiques avec robot d'aide au stockage.

Leurs caractéristiques principales sont illustrées à la Figure 1.4. Sur ce schéma, la base de la pyramide correspond au stockage de grandes quantités d'information, par exemple à des fins de sauvegarde long terme (*backup*). Les backups ne demandent pas un temps d'accès rapide mais demandent par contre une grande capacité ainsi qu'une bonne portabilité pour un stockage dans des endroits sécurisés. Les backups sont souvent réalisés avec des périphériques à bandes magnétiques (*tapes*) ou bien des disques durs externes connectés via le réseau ou USB. Sur le haut de la pyramide, nous retrouvons les types de mémoire volatiles à haute performance de débit et d'accès comme les registres, la cache ou la RAM. Ces mémoires permettent l'exécution rapide des opérations et des calculs demandés par les logiciels.

Nous donnons quelques descriptions supplémentaires sur les types de mémoire ci-dessous.

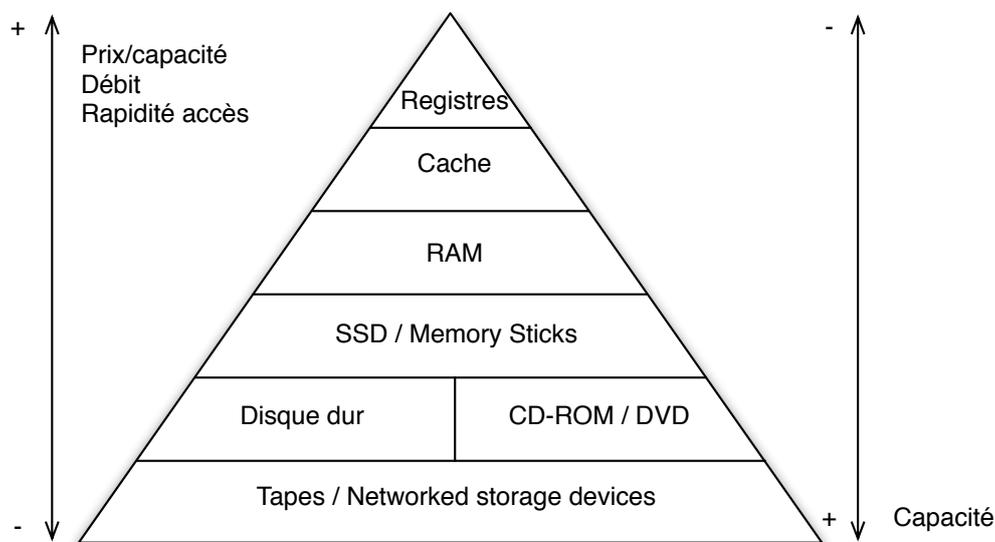


FIGURE 1.4 : Comparaison des types de mémoire selon des critères de prix, de rapidité et de capacité, cette dernière étant représentée par la base de la pyramide.

RAM/Cache/Registres La RAM ou mémoire à accès direct ou *Random Access Memory* ou encore mémoire vive, est une mémoire volatile qui permet de stocker de l'information tant que l'ordinateur est allumé et qui se ré-initialise lorsque l'ordinateur est éteint. Cette partie de la mémoire est parfois qualifiée de mémoire centrale. Il existe différentes générations de mémoire RAM selon les technologies semi-conducteurs utilisées. De fonctionnement similaire à la RAM, les registres et la mémoire cache sont des éléments de mémoire intégrés dans le microprocesseur. A la différence de la RAM, les informations stockées en caches et dans les registres ne transitent pas sur un bus et offrent donc un accès très rapide, typiquement à des instructions ou des données qui font l'objet de calculs intensifs.

ROM La ROM ou *Read Only Memory* ou encore mémoire morte, est une mémoire non volatile qui ne s'efface pas lorsque l'appareil qui la contient n'est plus alimenté en électricité. La ROM est utile par exemple pour stocker les informations nécessaires au démarrage d'un ordinateur ou d'un équipement mobile. Ces mémoires ne peuvent pas être accidentellement effacées par l'utilisateur. Il faut noter que certains types de ROM sont quand même effaçables et ré-inscriptibles en plaçant la mémoire dans une machine spéciale..

Mémoire flash Ce type de mémoire est, sur le principe, à mi-chemin entre la RAM et la ROM. Basée également sur des semi-conducteurs, elle est ré-inscriptible comme la RAM et non-volatile comme la ROM. Ses propriétés la rendent très utile pour des applications mobiles comme les appareils photo numériques, les smartphones, etc.

Disque dur Le disque dur est un équipement de mémoire de masse magnétique. Il contient des éléments mécaniques incluant des plateaux rigides en rotation et des bras qui se déplacent sur les plateaux. Les faces des plateaux possèdent une

couche magnétique sur laquelle sont stockées les données. Au bout des bras, des "têtes de lecture/écriture" permettent de lire ou de modifier le champ magnétique local qui encode une information binaire 1 ou 0. Les disques durs sont les équipements de mémoire de masse les plus répandus, typiquement dans les ordinateurs de bureau ou les ordinateurs portables. De nos jours, les capacités atteignent plusieurs téra-octets (*tera-bytes*). Un téra-octet (1 To) représente 10^{12} octets soit 1000000000000 octets.

SSD Le *Solid State Drive* a l'apparence extérieure d'un disque dur classique, a les mêmes dimensions et présente les mêmes types d'interface, mais est constitué de mémoires flash. Le coût reste supérieur à un disque dur mais les temps d'accès sont généralement plus rapides avec une consommation inférieure. Le SSD ne contient pas d'éléments mécaniques sensibles aux chocs et est donc matériellement plus solide qu'un disque dur tout en étant plus silencieux. Certains ordinateurs portables en sont maintenant équipés en remplacement des disques durs.

CD/DVD Le disque compact ou *Compact Disc* (CD) permet de stocker des informations en créant des irrégularités sur la surface réfléchissante d'un disque optique. Un faisceau de lumière cohérente (laser) est projeté sur la surface du disque qui se réfléchit et est ensuite enregistré par un capteur. Lorsqu'une irrégularité est présente, le capteur détecte une interférence et attribue la valeur binaire 1. Lorsque la surface est plane, le capteur attribue la valeur binaire 0. L'inscription des données sur le disque était, à l'origine, réalisable avec des équipements coûteux et non ré-inscriptible. La technologie a évolué vers des CD ré-inscriptible au moyen d'équipements embarqués dans les PC. La technologie *Digital Versatile Disk* ou DVD est similaire sur le principe mais présente une plus grande capacité de stockage d'information.

Bandes magnétiques Les bandes magnétiques ont été utilisées comme mémoires de masse dès les débuts de l'informatique. Le bon rapport prix/densité des bandes magnétiques en font toujours un support de choix pour les sauvegardes informatiques de type backup. Malgré les progrès très importants réalisés en matière de techniques de stockage sur disques magnétiques ou optiques, les bandes magnétiques restent un support intéressant en raison de leur très grande capacité et de leur caractère amovible.

La Table 1.2 résume les différents types de mémoire en fonction des caractéristiques de permanence, d'écriture, de performance, de capacité de stockage, de portabilité et du rapport prix par capacité.

	RAM	ROM	Mém. flash	SSD	Disque dur	CD/DVD	Bandes magn.
Volatile	oui	non	non	non	non	non	non
Réinscriptible	oui	non	oui	oui	oui	oui/non	oui
Temps d'accès	↓↓	↓↓	↓	↓	↔	↑	↑↑
Débit	↑↑	↑↑	↑	↑	↔	↔	↔
Capacité	↓↓	↓↓	↓	↔	↑	↔	↑↑
Portabilité	↔	↔	↑	↔	↔	↑	↑
Prix/capacité	↑↑	↑↑	↑	↑	↔	↓	↓↓

TABLE 1.2 : Comparaison des types de mémoire selon différents critères.

1.5 Système d'exploitation

Un système d'exploitation est un logiciel particulier, ou plutôt un ensemble de logiciels, qui gèrent les capacités matérielles d'un ordinateur, comme l'accès aux capacités de calcul, à la mémoire, aux périphériques, aux fichiers, au réseau, etc. Les capacités matérielles sont "demandées" par des applications tierces, souvent à l'initiative d'un utilisateur qui déclenche une action. Le système d'exploitation réserve alors les ressources demandées pour éviter des interférences avec des demandes provenant d'autres logiciels. En résumé, un système d'exploitation fournit des services qui permettent à une application de s'exécuter en sécurité, en utilisant au mieux les ressources de l'ordinateur, et de façon concurrente avec d'autres applications.

Il existe sur le marché des dizaines de systèmes d'exploitation différents. Ils sont souvent livrés et pré-installés avec l'ordinateur. Parmi les grands éditeurs, nous pouvons citer Windows de Microsoft, Mac OS d'Apple, et Linux qui se partagent le marché des ordinateurs personnels. En 2010, Windows détenait 90% des parts de marché. Les systèmes d'exploitation arrivent également sur les smartphones et les tablettes qui sont en fait devenus des ordinateurs à part entière, avec par exemple Android ou iOS.

Définition 1.5.1

Un **Système d'exploitation** est un logiciel particulier gérant les ressources d'un ordinateur et la coordination des instructions exécutées qui utilisent ces ressources. Ce logiciel est indépendant des logiciels d'application mais indispensable à leur mise en oeuvre. En anglais : *Operating System* ou *OS*.

Les fonctions principales des systèmes d'exploitation sont :

- gestion de la mémoire
- gestion des périphériques
- séquence et mise en priorité des opérations
- gestion du système de fichiers
- gestion des utilisateurs, des groupes et de leurs prérogatives (permissions)

Le système d'exploitation est quasiment le premier programme exécuté lors de la mise en marche de l'ordinateur. Il offre ensuite l'accès à des commandes de base qui facilitent la création de logiciels tiers, les "applications". C'est donc un intermédiaire entre ces logiciels et le matériel sous-jacent. La Figure 1.5 illustre ce principe. Un système d'exploitation est particulièrement utile pour garantir la pérennité des applications tierces en cas d'évolution du matériel. Par exemple, une commande d'ouverture de fichier appelée par une application restera identique pour accéder à un fichier stocké sur un disque SSD ou sur un disque dur. On parle de *couche d'abstraction* entre les applications et le matériel.

→ **Remarque.** Avant le démarrage du système d'exploitation, il y a le BIOS qui est une séquence d'instructions exécutée au démarrage de l'ordinateur et dont l'objectif est l'initialisation de certains périphériques de l'ordinateur et le démarrage du système d'exploitation.

Les fonctionnalités offertes par le système d'exploitation diffèrent d'un modèle à l'autre en fonction du matériel sous-jacent, mais nous allons retrouver de façon générique des commandes de base pour gérer la mémoire, l'accès aux fichiers, l'allocation de

ressources, etc. Pour désigner ces commandes “coeur”, on parle du *noyau*⁵ du système d'exploitation.

Autour du noyau, des services de plus en plus évolués se sont greffés au fil du temps, entre autre les gestionnaires d'*interfaces graphiques utilisateurs*⁶ basés sur des fenêtres dans lesquelles les applications sont exécutées. Il est de nos jours difficile de définir où s'arrête un système d'exploitation par rapport aux applications tierces. D'une part les éditeurs entretiennent le flou et considèrent comme faisant partie du système d'exploitation des applications “haut-niveau” qui l'accompagnent. D'autre part, les fonctionnalités et commandes de base offertes par les systèmes d'exploitation sont de plus en plus évoluées. Parmi les fonctions évoluées les plus récentes, nous pouvons citer la synthèse de la parole permettant de lire le contenu textuel d'une application, par exemple une page web, ou encore les services biométriques permettant d'identifier un utilisateur via son empreinte digitale ou la reconnaissance faciale.

→ **Remarque.** Les commandes de base du système d'exploitation sont accessibles à travers différents canaux : via l'interface graphique, via des commandes en ligne, via des bibliothèques de fonctions utilisées par les applications.

Les Figures 1.6, 1.7, 1.9 illustrent le *look-n-feel* ainsi que les outils typiques fournis par les systèmes d'exploitations les plus répandus.



FIGURE 1.5 : Les applications tierces reposent sur le système d'exploitation. Le système d'exploitation offre une couche d'abstraction par rapport au matériel informatique sous-jacent.

1.6 Evolution des performances

"Le nombre de transistors va doubler tous les ans.". Cette hypothèse a été énoncée en 1965 par l'ingénieur Gordon Moore, un des fondateurs d'Intel. Elle s'est vérifiée d'années

⁵En anglais, *kernel*

⁶En anglais, *Graphical User Interface*

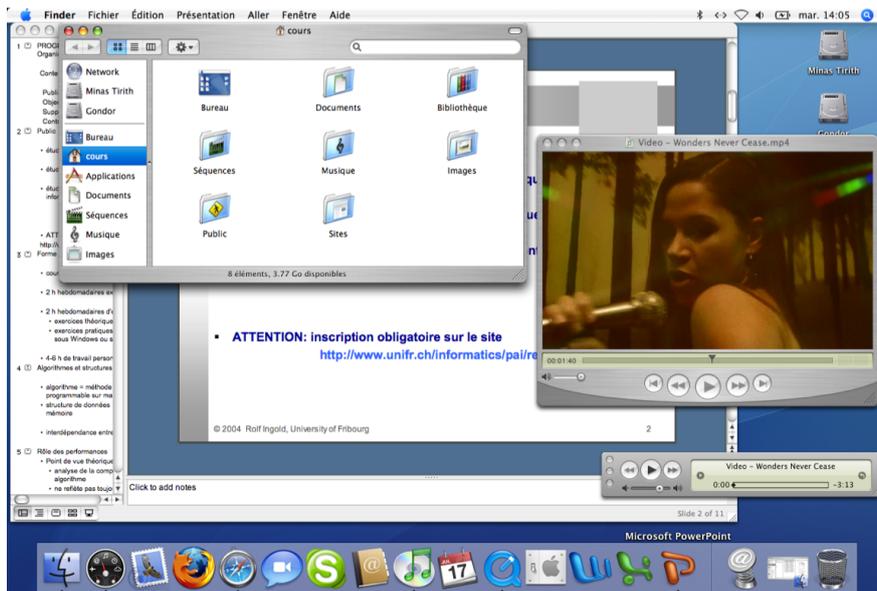


FIGURE 1.6 : Illustration du système d'exploitation Mac OS d'Apple.

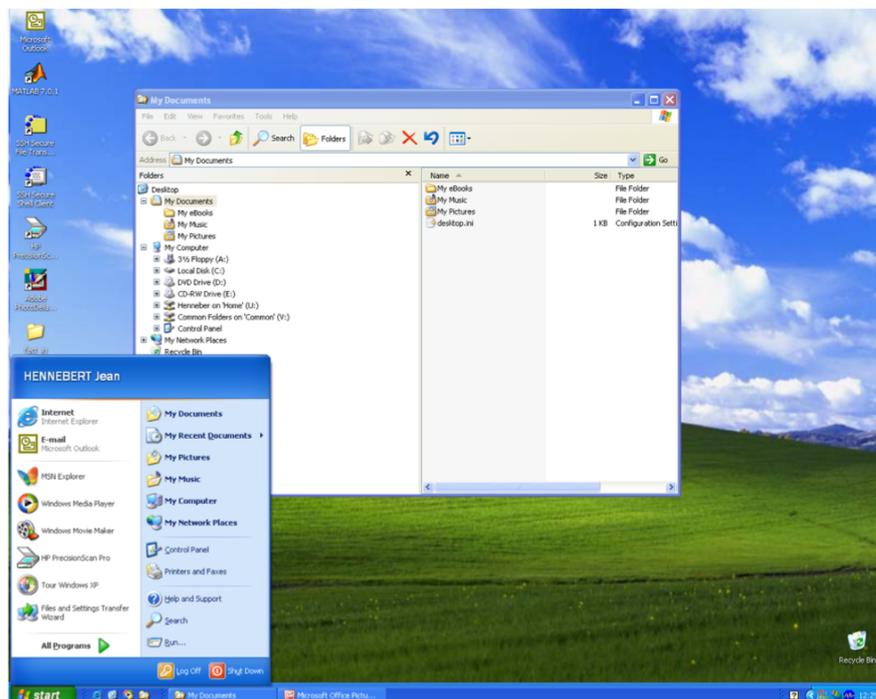


FIGURE 1.7 : Illustration du système d'exploitation Windows XP de Microsoft.

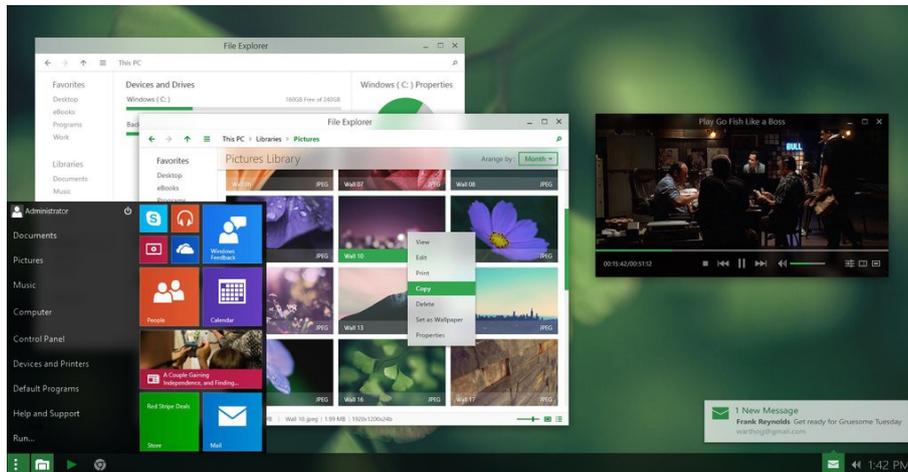


FIGURE 1.8 : Illustration du système d'exploitation Windows 10 de Microsoft.

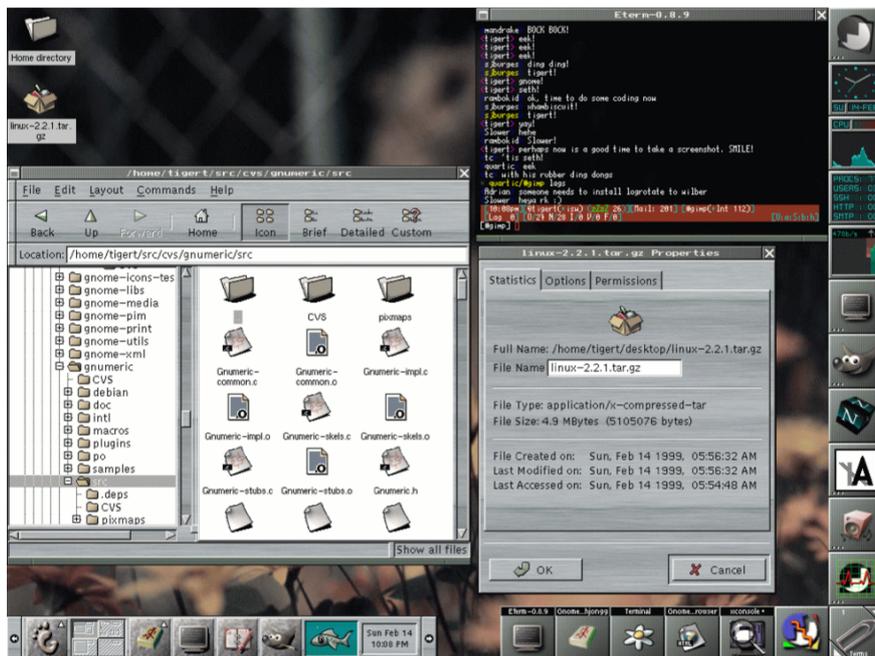


FIGURE 1.9 : Illustration du système d'exploitation Linux et fenêtrage Gnome.

les grandes surfaces des mémoires au même format qui ont des capacités 100 fois plus grandes.

Les autres paramètres importants des ordinateurs ont également suivi une évolution exponentielle. On parle de loi de Moore généralisée en rapport aux capacités de calcul, aux capacités des mémoires RAM et mémoires disques, aux bandes passantes des réseaux informatiques. Les conséquences de ces évolutions sont la miniaturisation des équipements, l'apparition de nouvelles applications et de nouveaux comportements des utilisateurs.

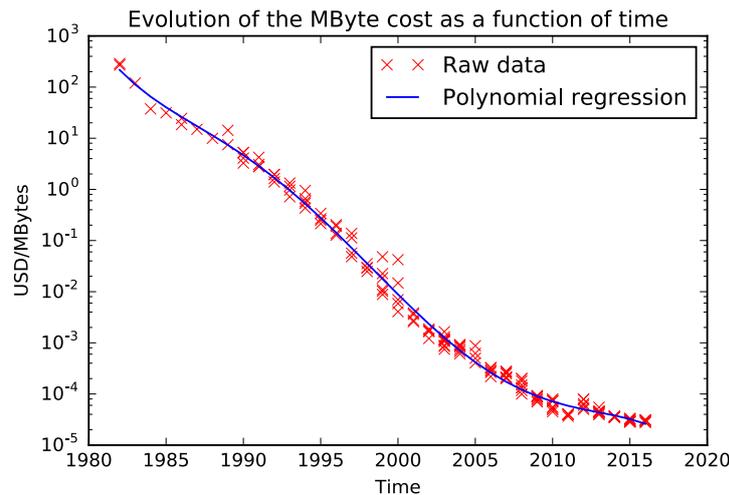


FIGURE 1.11 : Evolution du prix par MBytes pour les disques durs. L'échelle selon l'axe y est logarithmique, traduisant une diminution exponentielle du prix au fil des années.

1.7 Réseaux informatiques

Définition 1.7.1 (Réseaux informatiques)

Les **réseaux informatiques** se définissent par les moyens matériels et logiciels mis en oeuvre pour assurer la communication entre les ordinateurs. En anglais : *computer networks*.

Les télécommunications entre les ordinateurs ont pris une importance grandissante, principalement avec le développement d'Internet. Les liaisons entre ordinateurs prennent le terme générique de *réseaux informatiques*. La bande passante est le critère principal de leur performance.

Définition 1.7.2 (Bande passante)

La **bande passante** définit la quantité d'information que peut véhiculer un canal de communication. La bande passante se mesure en bits par seconde (bps) ou en octets (bytes) par seconde (B/s). En anglais : *bandwidth*).

En pratique, les réseaux reposent sur des équipements et des logiciels spécifiques. Au niveau des équipements, ceux-ci sont dédiés au *transfert* des données à distance via des câbles (par exemple fibres optiques) ou des liaisons sans fils (par exemple antennes Wi-Fi), au *routage* des données via des équipements de type routeurs et encore à la *connexion* d'un PC au réseau via des cartes et interfaces réseaux. Au niveau des logiciels, ceux-ci sont dédiés à l'encodage et au décodage des données à transférer, typiquement en respectant des protocoles qui sont standardisés afin de rendre la communication indépendante des ordinateurs et des équipements. Ces protocoles sont souvent combinés, "empilés" les uns aux autres pour réaliser une communication complète. Par exemple, l'ensemble des protocoles utilisés pour le transfert des données sur Internet s'appelle la suite TCP/IP.. En haut de la hiérarchie des protocoles, l'utilisateur n'a souvent conscience que du protocole "applicatif" qui, comme son nom l'indique, est lié à l'application de communication, comme par exemple FTP⁷ pour le transfert de fichier, HTTP⁸ pour le transfert de documents web ou encore SMTP⁹ pour le transfert d'emails. Les protocoles de communication offrent des capacités d'échanges d'information synchrones, comme par exemple la conférence audio, ou asynchrones comme par exemple l'email.

Observation de l'ingénieur informaticien. Au niveau de l'empilement des protocoles de communication, le modèle OSI¹⁰ définit les différents protocoles selon 7 couches [11]. Les 3 premières sont des couches proches du matériel, définissant typiquement comment l'information est transmise sous forme de bits et d'ensembles de bits (trames, paquets, datagrammes), d'un équipement à l'autre avec potentiellement un routage. Les 4 dernières sont des couches dites "hautes", plus proches des applications finales (visibles des utilisateurs). Ces couches définissent les notions de communication de bout en bout, de synchronisation des échanges, d'ouverture des sessions, du codage des données applicatives et finalement des protocoles liés à l'application finale.

1.8 Evolution des logiciels

Les logiciels suivent l'évolution du matériel. Les applications peuvent compter sur une puissance de calcul et sur des capacités de stockage de données de plus en plus grandes. La bande passante des réseaux grandit également et permet de nouveaux types d'applications, comme par exemple les applications client-serveur.

⁷File Transfer Protocol

⁸Hyper Text Transfer Protocol

⁹Simple Mail Transfer Protocol

¹⁰De l'anglais Open Systems Interconnection.

Définition 1.8.1 (Application client-serveur)

Une application **client-serveur** est une application composée d'une partie "client" qui envoie des requêtes, typiquement à travers Internet, et l'autre qualifiée de "serveur", qui répond aux requêtes. En anglais : *client-server application*.

L'application *Mathematica*, par exemple, a repris le mode de fonctionnement client-serveur à travers son concept de *front-end* (client) et de *kernel* (serveur). Son principe est de dissocier la partie présentation des expressions de la partie calcul. Les calculs peuvent donc s'exécuter, si besoin, au moyen d'un kernel s'exécutant sur un ordinateur tiers, généralement plus puissant que l'ordinateur en face duquel se trouve l'utilisateur.

Observation de l'ingénieur informaticien. Par extension, le terme "client" désigne l'ordinateur sur lequel est exécuté la partie cliente de l'application, et le terme "serveur" désigne l'ordinateur sur lequel réside la partie serveur de l'application. En général, les serveurs sont plus puissants que les clients qui, de nos jours, sont souvent des équipements mobiles comme les ordinateurs portables, les tablettes ou les smartphones. Parfois, le serveur est lui-même composé de deux parties, l'une couvrant la partie purement applicative et l'autre l'accès aux données. Dans ce cas, l'architecture est qualifiée de *trois tiers* avec le client, le serveur d'application et le serveur de données. Lorsque le client est un navigateur web, on parle de *client léger*.

De façon non exhaustive, nous pouvons souligner les évolutions suivantes :

- Les applications de communication ont également connu un essor important avec les outils de messagerie email, de messagerie instantanée, et encore de voix sur IP. On parle de convergence des outils de communication avec les téléphones mobiles qui deviennent de vrais ordinateurs et les ordinateurs qui offrent des capacités de communication équivalente aux téléphones.
- L'ingénierie logicielle, aussi dénommée *génie logiciel*, évolue également avec l'arrivée de langages de programmation de plus en plus évolués et la mise à disposition de bibliothèques de fonctionnalités de plus en plus grandes. Les applications se développent plus rapidement et, corollaire à cela, deviennent de plus en plus complexes. L'organisation des équipes de développeurs s'adapte également aux contraintes du marché avec l'apparition de méthodologies de gestion de projet dédiées aux développements logiciels, comme par exemple les méthodologies agiles de type *Extreme Programming* ou *Scrum*.
- Les interfaces sont de plus en plus intuitives intégrant des capacités de reconnaissance de signaux tels que l'écriture, la voix et le geste. De façon générale, les capacités humaines sont copiées, incluant également la biométrie, la traduction automatique de texte, la composition de résumés, etc.
- L'ordinateur peut se toucher via les écrans tactiles et, comme il s'invite dans nos poches via les smart-phones, adapte ses capacités à l'environnement ambiant (*context-awareness*). Le terme *pervasive computing* est apparu et désigne le fait que les logiciels, comme le matériel, sont accessibles de partout.

- Etant donné la puissance de calcul grandissante, les développeurs de logiciels et les scientifiques exploitent de nouveaux types d'algorithmes, comme par exemple ceux basés sur la force brute qui n'étaient simplement pas viables avant. Les styles de programmation changent également. On privilégie la portabilité du code, la facilité de son écriture, sa réutilisation et on met moins l'accent sur l'optimisation, la rapidité des algorithmes vu que de toute façon, les processeurs calculent tellement rapidement.

1.9 Familles de logiciels

Les logiciels peuvent se décliner en 2 grands groupes. Les **logiciels propriétaires** sont généralement payants et donc bien supportés, souvent évolutifs, documentés et disposent, pour les grands titres, d'une large base d'utilisateurs. Les **logiciels libres** sont gratuits, permettent un accès au code source ainsi que la possibilité de le redistribuer après modification, et cela en fonction des licences associées. Les logiciels libres manquent souvent de support, d'évolution, de documentation sauf pour certains logiciels de référence, comme par exemple ceux qui sont distribués par Apache¹¹.

Observation de l'ingénieur informaticien. Les logiciels libres ne doivent pas être confondus avec les logiciels *open source* qui sont des logiciels pour lesquels l'utilisateur a accès aux sources, peut en vérifier la qualité, mais ne sont pas forcément gratuits ni libres d'évolution. Il ne faut pas confondre non plus logiciels libres et logiciels *freewares* qui sont gratuits mais dont le code n'est pas systématiquement accessible.

Le choix d'un logiciel se fait généralement selon des critères commerciaux et non-commerciaux.

Critères commerciaux Coût du logiciel, base installée (acceptation du logiciel par ma communauté), santé financière de l'éditeur du logiciel, mise à jour du logiciel (évolution), règlement interne à l'entreprise¹²

Critères non-commerciaux Nombre de fonctionnalités, lisibilité et portabilité de mes données (importation/exportation vers d'autres logiciels), capacité de contrôle de qualité du logiciel (open-source), "utilisabilité" du logiciel¹³, existence de support, de documentations, de livres de référence.

De façon **non exhaustive**, nous pouvons classer les logiciels dans les familles suivantes :

1. BUREAUTIQUE.

- Traitement de texte : Word / Open Office / Page
- Editeurs de fichiers : Ultraedit / Emacs
- Traitement de texte scientifiques : LaTeX
- Tableurs : Excel / Numbers
- Présentation : PowerPoint / Impress / KeyNote

¹¹www.apache.org

¹²Par exemple, certaines organisations interdisent l'achat et l'utilisation de logiciels dont l'éditeur n'est pas coté en bourse.

¹³En anglais : *usability*

- Brainstorming : Visual-mind
 - Messaging, calendaring, contacts : Outlook / Eudora / Netscape
 - Gestion de projet : MS Project / Omniplan
2. LOGICIELS - ENTERPRISE RESOURCE PLANNING
 - SAP : human resources, finances, business process, ...
 3. GESTIONNAIRES DE BASES DE DONNÉES (SGBD)
 - Basés SQL : Oracle, MySQL, Postgres
 - Autres : Access
 4. GRAPHISMES
 - Vectoriels : Viso, Corel Draw, Omnigraffle
 - Photos : Photoshop, iPhoto, Aperture
 - Internet : Fireworks
 5. CALCULS SCIENTIFIQUES
 - Calcul symbolique : Mathematica
 - Calcul matriciel : Matlab, Octave
 6. ENVIRONNEMENTS DE PROGRAMMATION
 - Eclipse
 - Visual Studio
 7. GESTION DES FICHIERS
 - Archivage/Compression : Zip, RAR
 - Encryption : PGP
 - Backup : Legato, Time Machine
 8. INTERNET
 - Navigateurs : Explorer, Firefox, Safari, Chrome, Opera
 - Editeurs web : Dreamweaver, Frontpage
 - Serveurs web : Apache, Tomcat
 9. PROTECTION
 - Anti-virus : Norton
 - Anti-spam : F-Secure
 10. AUTRES LOGICIELS SPÉCIFIQUES À UN DOMAINE D'ACTIVITÉ
 - Architecture, DAO : autocad
 - Chimie : Compuchem

1.10 Systèmes de fichiers

Définition 1.10.1 (Fichier)

Un **fichier** est une suite de valeurs représentant des données et/ou des commandes informatiques qui sont enregistrées sur un support de mémoire et identifié par un nom. En anglais : *File*.

Un fichier est généré par un logiciel à l'initiative de l'utilisateur ou automatiquement par un programme. Un fichier contient un programme ou des données ou les deux. Le fichier est identifié par un nom qui contient généralement 3 parties : le chemin, le nom

de fichier, l'extension. La Table 1.3 résume les extensions courantes utilisées. Les fichiers sont enregistrés sur un support mémoire comme le disque dur, le memory stick, ou encore sur un support mémoire distant à travers le réseau Internet. Le fichier est parfois **chiffré/crypté** pour ne permettre la lecture qu'à des personnes autorisées (qui ont la clé de décryptage), **compressé** pour réduire sa taille sur le support, **archivé** pour permettre l'inclusion de plusieurs fichiers dans un seul appelé l'archive. Les fichiers sont également caractérisés par d'autres paramètres qui sont gérés par le système d'exploitation comme la date de création, la date de modification, l'auteur, la taille, etc.

Observation de l'ingénieur informaticien. La dernière partie, l'extension, définit le type de fichier et est utilisé par le système d'exploitation pour faire l'association au programme qui est "capable" de traiter ce fichier. Par exemple, un fichier dont l'extension est .doc est probablement généré par le logiciel Word de Microsoft. Lorsqu'on double clique sur l'icône d'un fichier .doc dans l'explorateur de fichier, le système d'exploitation démarre le programme associé en lui donnant la référence au fichier qu'il s'agit d'ouvrir. L'extension du fichier est souvent composée de 3 ou 4 caractères.

Extensions courantes	
.exe	fichier executable
.txt	document texte simple
.pdf	documents pour visualisation et impression
.gif, .jpg	fichiers contenant des images compressées
.doc, .docx	document word
.xls	document excel
.c, .java	code source en pascal, c ou java
.nb, .m	document Notebook Mathematica
.m	fichier de script ou fonctions Matlab
.zip, .rar	archive contenant un ou plusieurs fichiers compressés

TABLE 1.3 : Liste des extensions courantes pour les fichiers.

→ **Remarque.** Pour les fichiers texte simples, les correspondances entre les séquences de bits et les caractères sont définies par des tables, par exemple Unicode. Le terme pdf signifie portable document format, qui est un standard ouvert défini par Adobe pour l'échange, la visualisation et l'impression de documents textes.

Définition 1.10.2 (Répertoire)

Un **répertoire**, ou encore *dossier*, est un conteneur de fichiers. C'est en fait un fichier particulier qui contient d'autres fichiers et éventuellement d'autres répertoires. La notion de répertoire permet d'organiser les fichiers de façon hiérarchique. En anglais : *folder* ou *directory*.

Un répertoire peut donc contenir des fichiers et des sous-répertoires, permettant à un utilisateur de structurer de façon logique ses fichiers. On utilise parfois une représentation sous forme d'arbre pour représenter cette structure, comme illustré sur la Figure 1.12. La *racine* désigne le support mémoire qui est utilisé. Sous Windows, une lettre désigne le support mémoire, comme par exemple C : est souvent associé au disque dur principal. On utilise également le terme *volume* ou *disque* pour désigner le support physique sur lequel se trouve le fichier.

Il est important de bien nommer les fichiers et de les organiser de façon compréhensible dans des répertoires. En effet, nous avons de plus en plus de fichiers à traiter et nous devons parfois les communiquer à des collègues. Une bonne organisation et dénomination des fichiers permet de gagner du temps en facilitant leur recherche. Il faut également noter que la dénomination des fichiers et leur organisation est parfois imposée pour être conforme à un processus d'entreprise.

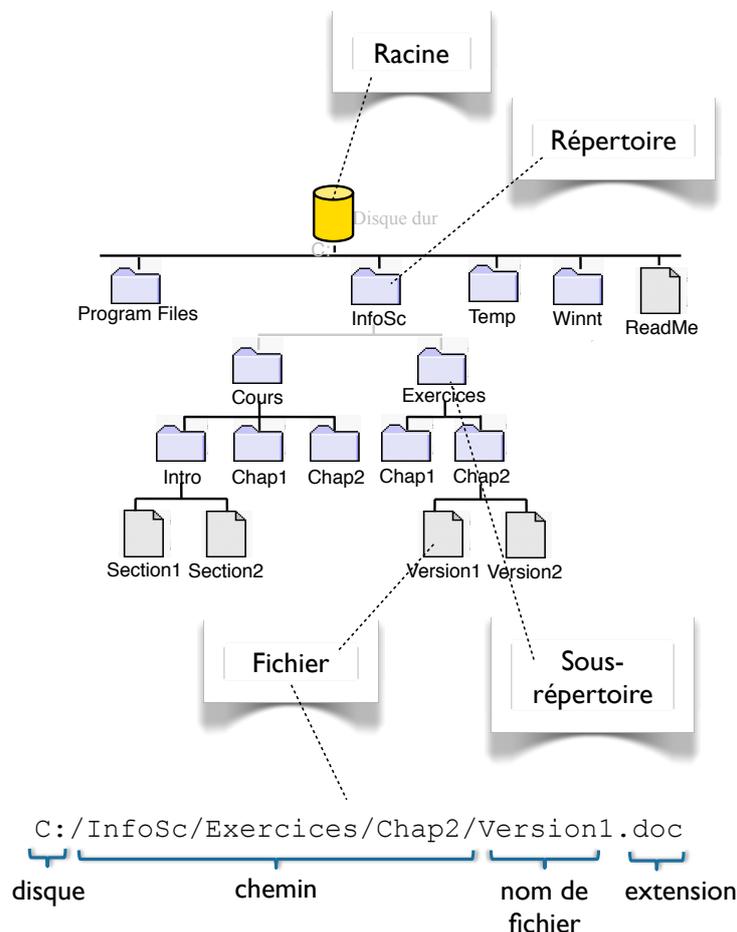


FIGURE 1.12 : Illustration d'un système de fichiers. Représentation en arbre.

Observation de l'ingénieur informaticien. Il existe certaines règles de dénomination des fichiers : dans un même répertoire, tous les noms doivent être différents, interdiction d'utiliser certains caractères spéciaux comme par exemple { } [] * ; / \.

Certains fichiers ou répertoires sont cachés à l'utilisateur par le système d'exploitation. Certains fichiers ou répertoires sont partagés, c'est-à-dire visibles par d'autres utilisateurs potentiellement à travers Internet sur d'autres machines. De façon plus formelle, des *droits* sont associés aux fichiers, permettant ou pas la lecture, l'écriture, l'exécution. Les utilisateurs n'ayant pas assez de privilèges peuvent se voir refuser certains droits. La gestion de ces droits et privilèges dépend des systèmes d'exploitations et des types de systèmes de fichier qui sont utilisés.

1.11 Environnement de développement intégré et Python

Il est maintenant temps de nous plonger dans le coeur du sujet, à savoir la programmation des ordinateurs. Nous avons choisi d'utiliser Python comme langage de programmation [6]. Gratuit, libre de droit, avec une base d'utilisateur grandissante et bien adapté aux besoins de la programmation scientifique, nous sommes convaincus que Python est un langage adéquat pour des débuts en programmation. Python est souvent qualifié de "couteau suisse" de la programmation car il inclut une grande quantité de fonctionnalités rapidement utilisables, entre autre pour l'exploration et la visualisation de données scientifiques. La syntaxe de Python est proche d'autres langages comme C ou Java, mais moins verbeux et donc probablement plus "lisible". Python est développé par la *Python Software Foundation*¹⁴ qui coordonne les différentes évolutions du langage et qui anime la communauté d'utilisateurs de Python. Le logo du langage Python est illustré à la Figure 1.13.



FIGURE 1.13 : Logo du langage Python

Observation de l'ingénieur informaticien. Python a été créé en 1990 par Guido van Rossum et a évolué au fil des années. En 2010, la version 2.7 du langage a été mise à disposition et est, depuis lors, considérée comme la version la plus stable du langage. De 2008 à 2016, la fondation Python a fait évoluer le langage de la version 3.0 à 3.6 mais comme beaucoup de bibliothèques externes ne sont pas compatibles avec les versions 3.x, beaucoup de développements se font encore avec la version 2.7. Pour cette raison, nous avons opté de baser les exemples donnés dans ce livre sur la version 2.7 du langage. Leur évolution vers la version 3.0 ne devrait néanmoins pas causer de grandes difficultés.

Python est donc un langage de programmation qui permet aux utilisateurs d'exprimer des commandes que l'ordinateur va ensuite interpréter et exécuter de façon successive. Le terme "Python" désigne à la fois le langage de programmation avec ses règles syntaxiques et sémantiques, mais aussi le logiciel qui va interpréter et exécuter les commandes que l'utilisateur aura écrites. Dit autrement, Python est un logiciel qui permet d'exécuter d'autres logiciels écrits dans le langage Python.

A ce stade de la lecture, nous vous suggérons de réaliser la première série de travaux pratiques "**S01 - Installation et prise en main de Python**" qui va vous guider dans l'installation de Python sur votre ordinateur. Comme indiqué dans ces travaux pratiques,

¹⁴<https://www.python.org>

nous vous recommandons d'installer un environnement intégré de programmation pour l'écriture du code Python.

Définition 1.11.1 (Environnement de développement intégré)

Un **Environnement de développement intégré** est un logiciel qui permet l'écriture, l'analyse, le débogage et l'exécution de codes de programmation. En anglais : *Integrated Development Environment* ou *IDE*.

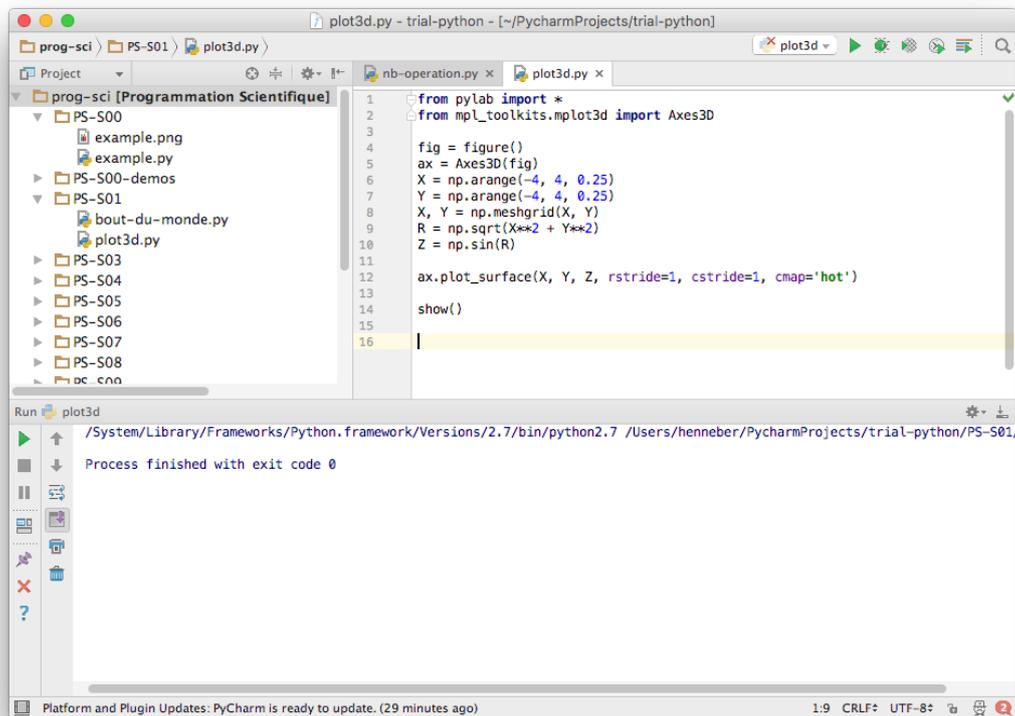


FIGURE 1.14 : Illustration de l'environnement de développement intégré Pycharm. De base, l'environnement Pycharm permet de naviguer dans les fichiers de programmation (partie en haut à gauche), d'éditer un programme (partie en haut à droite) et d'observer les résultats de son exécution (partie en bas).

Nous illustrons à la Figure 1.14 l'environnement de développement intégré Pycharm dédié à Python et développé par la société JetBrains¹⁵. Ces environnements permettent de gérer un ensemble de fichiers de programmation, d'éditer le code Python et d'exécuter les programmes. Au niveau de l'écriture du code, la plupart de ces environnements offrent des fonctionnalités d'exécution pas à pas qui permettent l'analyse et le débogage. Nous

¹⁵L'édition *Community* de Pycharm est disponible gratuitement sous <https://www.jetbrains.com/pycharm/> et permet de réaliser l'ensemble des exercices de ce livre.

reviendront plus tard sur ces fonctionnalités. L'exécution du code `plot3d.py` de la Figure 1.14 est un graphique 3D qui est illustré en Figure 1.15.

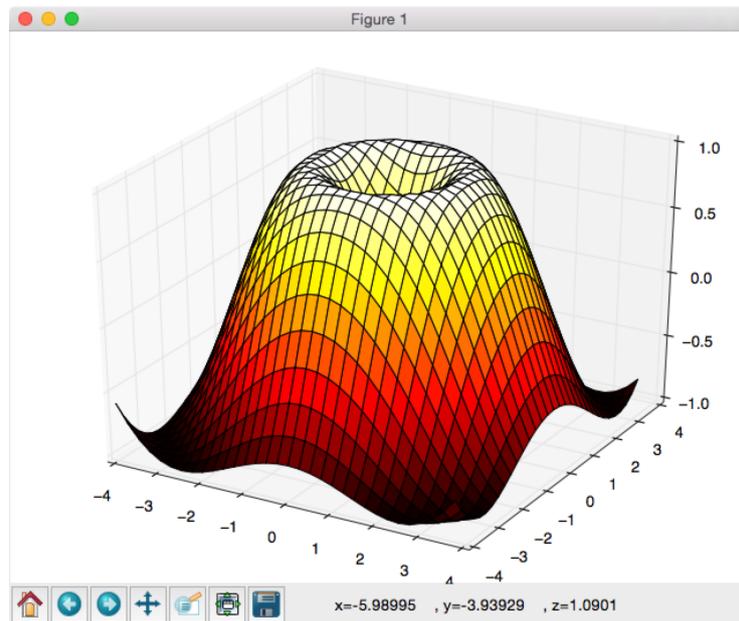


FIGURE 1.15 : Résultat de l'exécution du code Python donné en exemple dans la Figure 1.14. Vous pouvez interagir avec la figure 3D avec la souris qui permet de changer le point de vue.

Vous noterez que les environnements de développement tels que Pycharm utilisent certaines conventions pour les couleurs des différents éléments qui composent les commandes Python que vous entrez dans l'éditeur ou dans la console. Comme illustré à la Figure 1.14, la configuration de base de PyCharm utilise le vert pour les chaînes de caractère (*string*), le bleu est utilisé pour les mots réservés du langage Python (*keywords*), le bleu léger pour les valeurs numériques, etc. Nous allons utiliser ces conventions dans ce livre et dans les séries d'exercices.

Si vous avez réalisé la première série d'exercice "**S01 - Installation et prise en main de Python**", vous devriez maintenant être capable d'entrer des commandes Python dans la console ou dans un script de l'environnement de programmation. Vous devriez également être capable d'exécuter un script Python et d'en observer le résultat. Veuillez noter qu'à ce stade, le but n'est pas de comprendre les commandes Python mais de s'habituer avec l'environnement de programmation et ses fonctionnalités.

Avant de passer au chapitre suivant, nous vous proposons de lire le Listing 1.1. Il s'agit d'un programme Python qui permet de calculer le nombre d'opérations que votre ordinateur peut réaliser par secondes.

```
1 import time
2
3 maxTime = 5
4 operationCounter = 0
5
6 timeStart = time.time()
7
8 while time.time() - timeStart < maxTime:
9     operationCounter = operationCounter + 1
10
11 print('processing time      : ', maxTime, 'seconds')
12 print('operations          : ', operationCounter)
13 print('time per operation   : ', maxTime * 1.0 / operationCounter, '
    seconds')
14 print('operations per second: ', operationCounter / maxTime)
```

Listing 1.1 : nb-operation.py

Dans le Listing 1.1, la ligne 1 permet d'utiliser une librairie Python pour l'observation de l'heure et du temps écoulé. Les lignes 3 et 4 initialisent les variables dont le programme a besoin, entre autre le temps de mesure qui est ici défini égal à 5 secondes. La ligne 6 permet de récupérer l'heure actuelle. Les lignes 8 et 9 définissent un bloc d'instruction itératif (ou plus communément une *boucle*) : tant que le temps écoulé est inférieur à `maxTime` (5 secondes dans notre cas), un calcul est effectué en ligne 9. Ce calcul est une simple incrémentation de la variable `operationCounter` qui permet de compter le nombre d'opérations effectuées. Les lignes 11 à 14 permettent d'imprimer (`print`) sur la console différentes informations. Le résultat de l'exécution sur un ordinateur¹⁶ est le suivant :

```
1 /System/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6
  /Users/henneber/PycharmProjects/prog-sci/PS-S00-demos/nb-operation.
  py
2 processing time      : 5 seconds
3 operations          : 26015238
4 time per operation   : 1.92195051223e-07 seconds
5 operations per second: 5203047
6
7 Process finished with exit code 0
```

Nous observons que l'ordinateur a exécuté 26 millions d'opération en 5 secondes, soit plus de 5 millions de fois par seconde. Vous pouvez maintenant comparer avec votre propre ordinateur en exécutant ce code. Il est fort probable que si vous avez un ordinateur plus récent avec un processeur plus performant, le nombre d'opérations que vous observerez sera plus élevé, ou à l'inverse moins élevé dans le cas d'un processeur moins performant.

¹⁶MacBook Pro 2015 avec processeur Intel Core I7 2.6 GHz.

Observation de l'ingénieur informaticien. Quelques observations complémentaires peuvent être faites sur ce résultat d'exécution. Bien que le nombre mesuré d'opérations reste impressionnant, en Section 1.2, nous vous annonçons des ordinateurs capables d'exécuter des milliards d'opérations par seconde. Comment se fait-il que nous n'observions "que" 5 millions d'opérations par secondes ? Premièrement il faut considérer la structure des processeurs actuels qui est organisée en "coeurs" fonctionnant en parallèle. Dans le code envisagé ici, l'exécution des instructions est purement séquentielle sur un seul coeur. Le code devrait être écrit autrement pour envisager une "distribution" du calcul sur les coeurs (hors périmètre de ce livre). Ensuite Python est un langage de haut niveau qui est interprété. Ceci veut dire que les instructions Python sont d'abord traduites avant de pouvoir être exécutées par le processeur et cette traduction prend un temps non négligeable. Ensuite, il faut noter que les lignes 8 et 9 demandent beaucoup plus qu'une seule opération mathématique. La ligne 8 par exemple réalise une différence et une comparaison tandis que la ligne 9 réalise une addition et une affectation (voir chapitre suivant), soit au minimum 4 opérations mathématiques plus la gestion de la boucle. Finalement, il faut également tenir compte du système d'exploitation qui, comme expliqué en Section 1.5, s'occupe de beaucoup d'autres choses en même temps que l'exécution de notre code Python.
