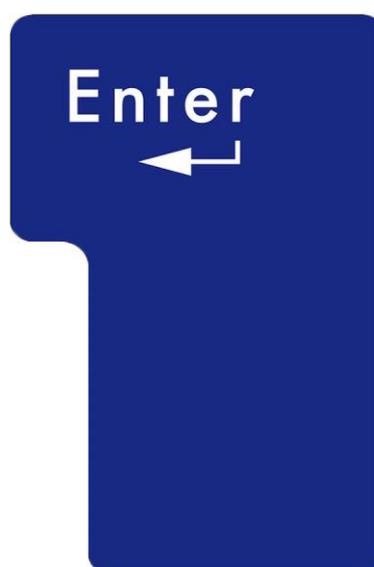




Informatique et sciences du numérique

Enseignants de cycle 3, de cycle 4 et de lycée



Département Éducation et Formation

Palais de la découverte
Avenue Franklin D. Roosevelt
75008 Paris
www.palais-decouverte.fr

2018

ESPACE
INFORMATIQUE
& SCIENCES
DU NUMÉRIQUE

///
Ouverture
le 13 mars 2018



Création graphique : FLOUAS CHARTRON - 04 77 50 10 02



En partenariat avec : **l'AGENCE DU NUMÉRIQUE**

Avec le soutien de :

Sommaire

I L'espace permanent

I.1	Situation et plan	4
I.2	Contenu	5
I.2.1	Les fondamentaux	
I.2.1.1	Les algorithmes	6
I.2.1.2	Les langages	11
I.2.1.3	L'information	15
I.2.1.4	Les machines	19
I.2.2	Le laboratoire	
I.2.2.1	Le <i>big data</i>	22
I.2.2.2	L'apprentissage automatique	27
I.2.2.3	Les réseaux	29
I.2.2.4	Les robots	31
I.2.3	L'arène techno	
I.2.3.1	<i>Fluid</i>	33
I.2.3.2	<i>Xplore Touch</i>	33
I.2.3.3	<i>LiFi</i>	34

II Ressources

II.1	Ateliers et exposés	34
II.2	Bibliographie	37

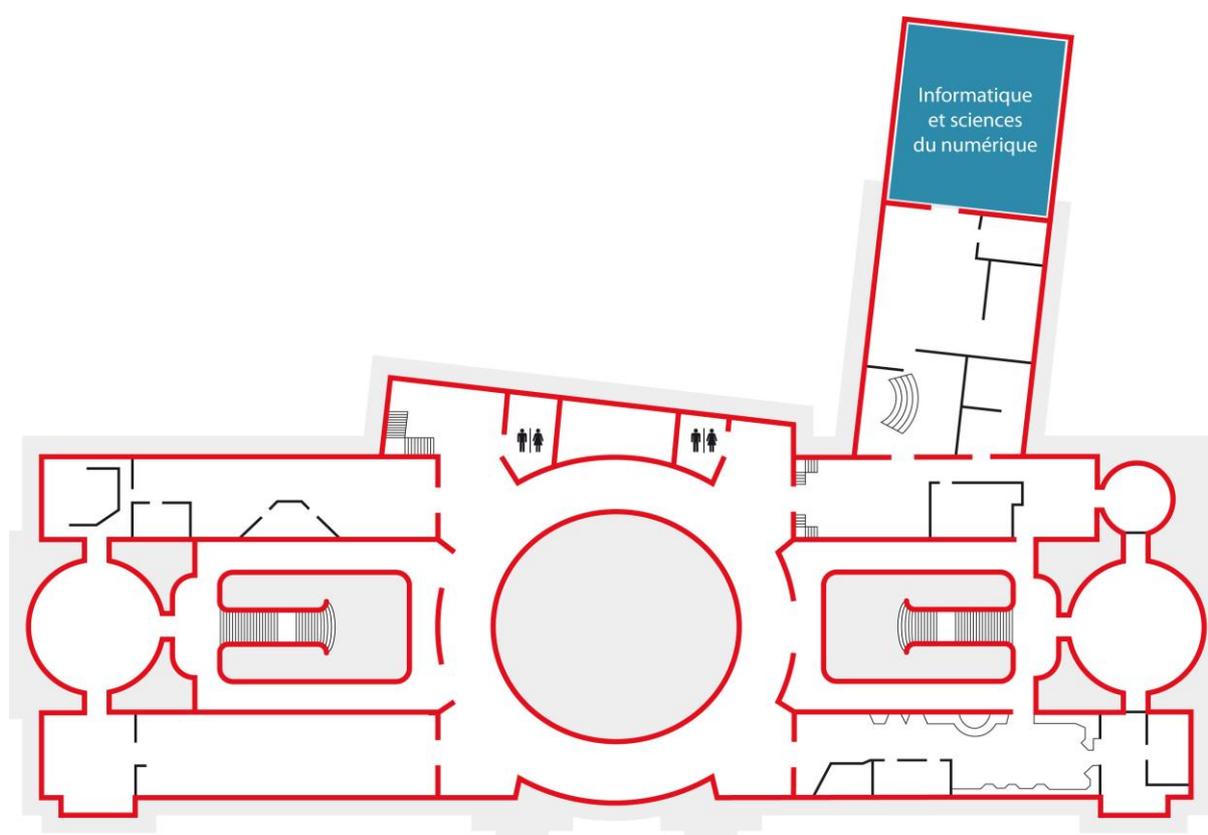
III Informations pratiques 38

IV Annexe 39

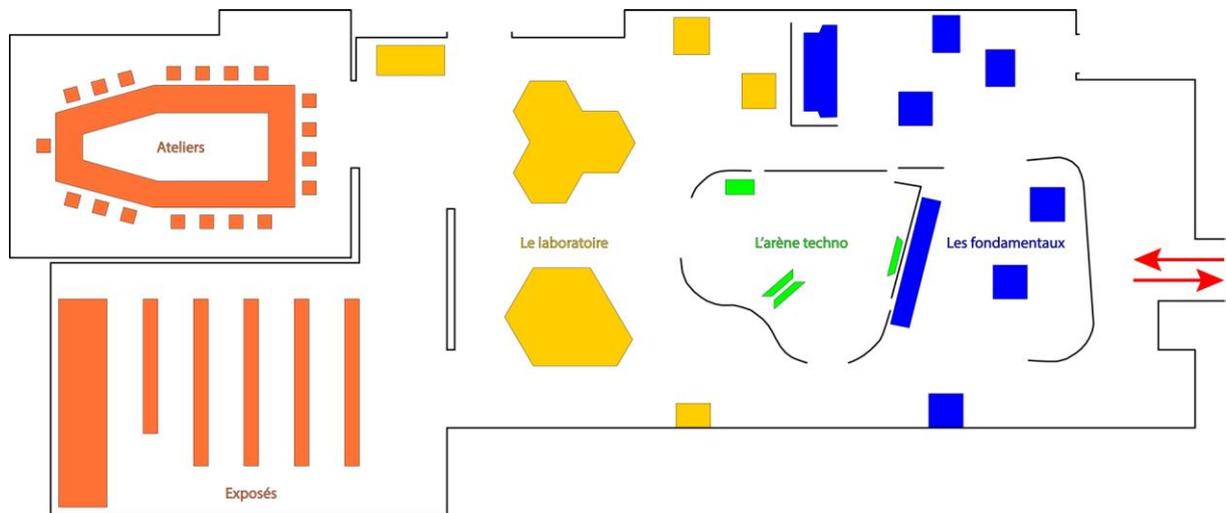
I L'espace permanent

I.1 Situation et plan

L'espace permanent *Informatique et sciences du numérique* est ouvert au public depuis mars 2018. Il occupe une surface totale de 300 m² et prend place au niveau 1 du Palais de la découverte, dont voici le plan schématique.



Le graphique ci-dessous révèle l'organisation schématique de l'exposition permanente *Informatique et sciences du numérique*. Accessible au public handicapé (PMR et malvoyants), elle est présentée en trois langues : français, anglais et italien.



I.2 Contenu

Une question posée à votre smartphone, une nouvelle méthode de diagnostic médical, un véhicule autonome... l'informatique est partout ! Au-delà des outils informatiques devenus omniprésents, une révolution numérique est en train de modifier profondément notre société.

Si *Homo numericus* est familier de l'utilisation de ces outils, la compréhension des sciences et technologies sous-jacentes est moins bien acquise. Elle permettrait pourtant à tout un chacun de devenir un utilisateur actif et critique du numérique. Alors, comment ça marche ?

L'informatique peut être vue comme un grand jeu de construction dont les quatre briques élémentaires – l'information, les machines, les algorithmes et les langages – vous sont expliqués dans la partie « Les fondamentaux ». Vous y découvrirez combien l'informatique est un domaine à la fois créatif, artisanal et ludique.

À mesure que les systèmes issus de l'informatique gagnent en complexité, ils influencent de plus en plus notre quotidien, à l'instar des réseaux, des robots, du *big data* ou de l'apprentissage profond (*deep learning*), que vous expérimenterez dans la partie « Le laboratoire ». En interagissant avec ces systèmes, ils se révéleront plus accessibles que vous ne le pensiez !

Enfin, vous pourrez tester des dispositifs innovants dans « L'arène techno », afin de vous faire une idée des possibles usages de l'informatique dans un futur proche.

I.2.1 Les fondamentaux

Cette partie de l'exposition aborde les quatre éléments fondamentaux de l'informatique : les algorithmes, les langages, l'information et les machines.

I.2.1.1 Les algorithmes

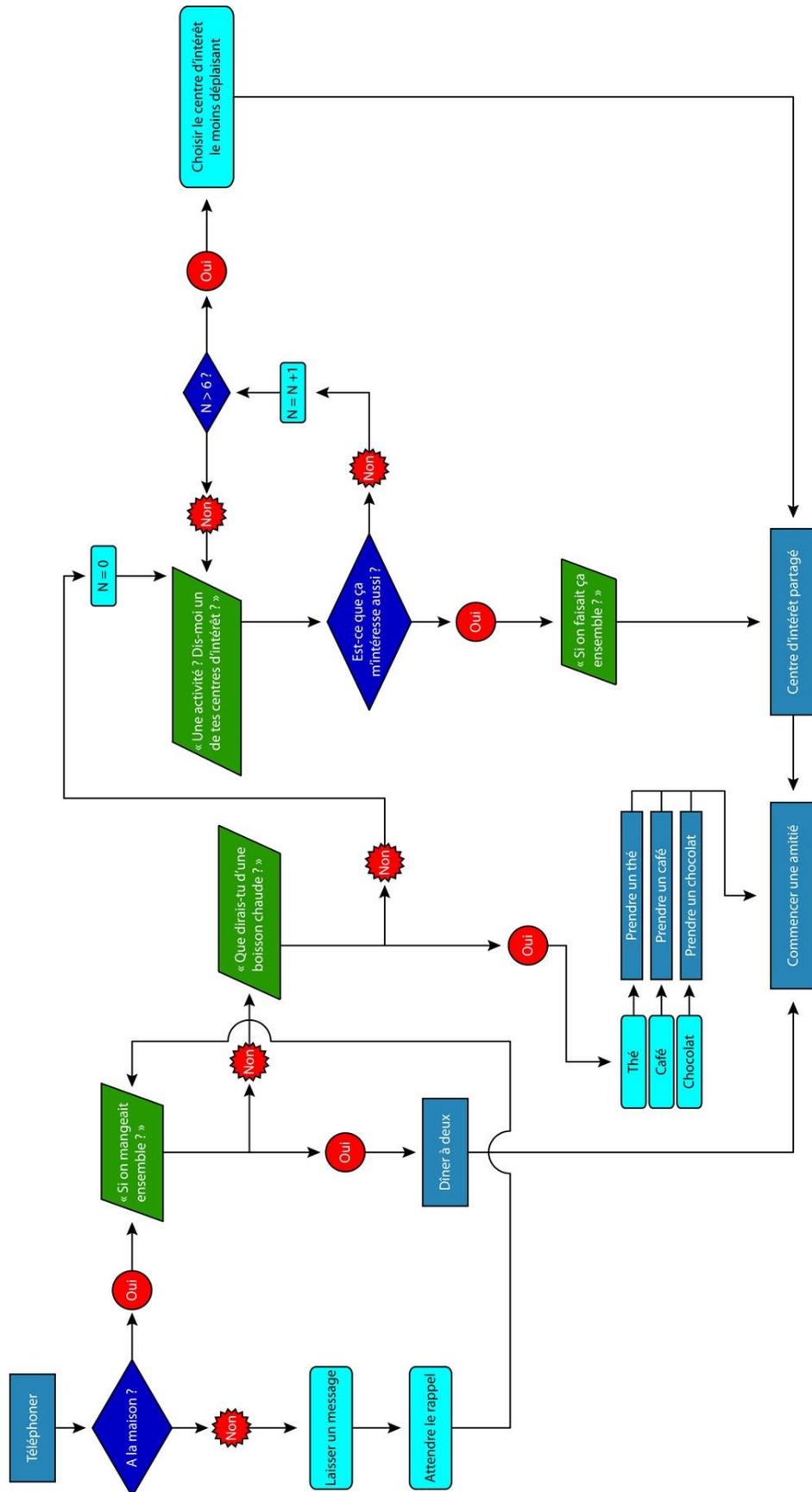
Trier une liste de mots, résoudre un Rubik's Cube, faire une addition, lacer ses chaussures... il y a des algorithmes pour tout. Un algorithme, c'est une méthode systématique pour résoudre un problème par une suite d'opérations élémentaires. Au départ, l'algorithmique n'était d'ailleurs pas dévolue à l'informatique. Il y a 4000 ans, en Mésopotamie, des algorithmes servaient déjà à calculer, entre autres, des racines carrées. Et si on peut imaginer différents algorithmes pour résoudre un problème donné, tous n'ont pas la même efficacité : on parle de *complexité algorithmique*.

Comment trouver un mot dans le dictionnaire ? Voici trois méthodes génériques qui ont inspiré de nombreux autres algorithmes :

- la méthode force brute passe en revue tous les mots depuis la première page jusqu'à trouver le bon ;
- on peut aussi ouvrir le dictionnaire au milieu. Si le mot est avant dans l'ordre alphabétique, on ouvre la moitié précédente au milieu, sinon la moitié suivante, et ainsi de suite jusqu'à isoler le mot. On a divisé le problème en sous-problèmes plus petits, c'est la méthode récursive ;
- enfin, une méthode probabiliste procéderait de même mais en ouvrant une page au hasard plutôt que la page du milieu. Elle est sensiblement aussi efficace.

En page suivante, vous trouverez une représentation humoristique en organigramme d'un algorithme, inspiré de l'épisode 13 de la saison 2 (*The Friendship Algorithm*) de la série télévisée *The Big Bang Theory*. Cet algorithme est censé vous apporter la méthode infaillible pour vous faire des amis...

Enfin, en annexe, vous sont exposés deux algorithmes non présentés dans l'espace permanent *Informatique et sciences du numérique* : l'algorithme d'Euclide et l'algorithme de Nelder-Mead.



Ceux qui portent une cravate la nouent en exécutant quasi machinalement une série de gestes. Combien de nœuds différents peut-on réaliser ? En 1999, Thomas Fink et Yong Mao du laboratoire Cavendish de l'université de Cambridge ont publié le petit article *Designing tie knots by random walks* dans la prestigieuse revue *Nature*, où ils décomposent l'opération en gestes décrits et notés par des signes conventionnels. Cette notation leur permet de générer des algorithmes pour réaliser 85 nœuds différents. Ils retrouvent les quatre nœuds les plus populaires : le nœud simple, le Windsor, le demi-Windsor et le Pratt.

SIGNES CONVENTIONNELS

- G = gauche
- D = droite
- C = centre
- B = glisser l'extrémité large dans la dernière boucle formée
- ⊗ = l'extrémité large passe au-dessus de l'extrémité fine
- ⊙ = l'extrémité large passe au-dessous de l'extrémité fine



exemple : B



exemple : G⊗

Ici, exercez-vous à faire un nœud de cravate en manipulant seulement son extrémité la plus large.

1. Exemple de nœud simple pour s'entraîner.

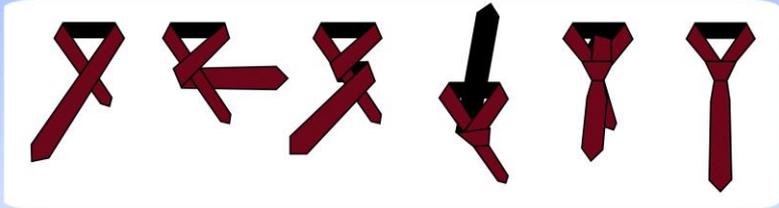
G⊗

D⊙

G⊗

C⊙

B



2. Algorithme pour le nœud demi-Windsor.

G⊗

D⊙

C⊗

D⊙

G⊗

C⊙

B

3. Algorithme pour le nœud Windsor.

G⊗

C⊙

D⊗

G⊙

C⊗

D⊙

G⊗

C⊙

B

Faites votre nœud.

Vos élèves pourront expérimenter la conception d'algorithmes en tentant de sortir du *Labyrinthe algorithmique* (code source téléchargeable et modifiable [ici](#)).



Dans cet élément interactif, on présente aux élèves cinq algorithmes classiques pour résoudre des labyrinthes et on leur demande de les tester puis de les comparer. Pour chaque algorithme, on indique le temps et la mémoire nécessaires pour en sortir.

1 – Algorithme aléatoire

À chaque intersection, l'algorithme choisit une direction au hasard, comme le ferait une souris. Il finit toujours par atteindre la sortie, mais cela peut être très long !

2 – Algorithme « longer le mur »

En longeant le mur de gauche (ou celui de droite) depuis le point de départ du labyrinthe, vous êtes assuré de trouver la sortie. Par contre, si le point de départ est contre un mur isolé des autres, comme un îlot... vous tournerez autour sans fin.

3 – Algorithme de Pledge

Il consiste à aller tout droit jusqu'à un mur puis à le longer en additionnant les valeurs des angles chaque fois que l'on tourne. Si l'on se retrouve à nouveau dans la même direction et que la somme des angles vaut 360° , c'est que l'on tourne en rond autour d'un îlot. Dans ce cas, on va tout droit jusqu'à un mur pour le longer.

4 – Algorithme de parcours en profondeur

Lorsqu'un chemin se révèle être une impasse, on applique une marque sur ses cases et on se reporte au début de l'impasse pour poursuivre l'exploration par un chemin non marqué. Les

impasses sont ainsi marquées les unes après les autres et il ne reste plus qu'un chemin non marqué, qui mène du point de départ à la sortie.

5 – Algorithme de parcours en largeur

Cet algorithme trouve le chemin le plus court en comptant les pas. Attribuons le chiffre 0 au point de départ, 1 à ses cases adjacentes, 2 aux suivantes et ainsi de suite. Si plusieurs chemins mènent à la sortie au cours de cet exercice, celle-ci reçoit plusieurs nombres mais ne garde que le plus petit. À la fin, le chemin le plus court part de la sortie et suit les cases dont la valeur est une unité plus petite que la précédente, jusqu'à atteindre le chiffre 0 du départ !

De cinq algorithmes, lequel est le plus efficace pour sortir du labyrinthe ?

Chacun possède ses avantages et ses inconvénients. Voici quelques éléments de comparaison.

Algorithme	Temps d'exécution maximal	Mémoire utilisée	Garantie de trouver au moins une solution ?	Utilisation par un humain dans un labyrinthe ?
1 – Aléatoire	Infini	0	Oui	Oui
2 – Longer le mur	2N	0	Sous condition*	Oui
3 – Pledge	2N	1	Oui	Oui
4 – Parcours en profondeur	N	N	Oui	Non
5 – Parcours en largeur	N	N	Oui	Non

N est le nombre total de cases du labyrinthe. Le temps d'exécution des algorithmes, et parfois l'utilisation de la mémoire, dépendent de **N**.

Ce qu'on appelle la complexité d'un algorithme est la quantité de ressources (temps, mémoire) qu'il utilise. On la mesure généralement par rapport à la taille du problème à résoudre. Prenons un autre exemple que le nombre de cases d'un labyrinthe : trions un jeu de **N** cartes.

Première méthode : parcourir les cartes, poser la plus faible, recommencer avec le reste du jeu et ainsi de suite. Le temps de calcul est ici proportionnel au carré de **N**, ce qui est raisonnable.

Deuxième méthode : tester tous les ordres possibles du jeu de cartes, l'un après l'autre. Le temps d'exécution augmente alors exponentiellement avec **N**. Il faudrait des milliards de milliards d'années à un ordinateur actuel pour trier 32 cartes !

I.2.1.2 Les langages

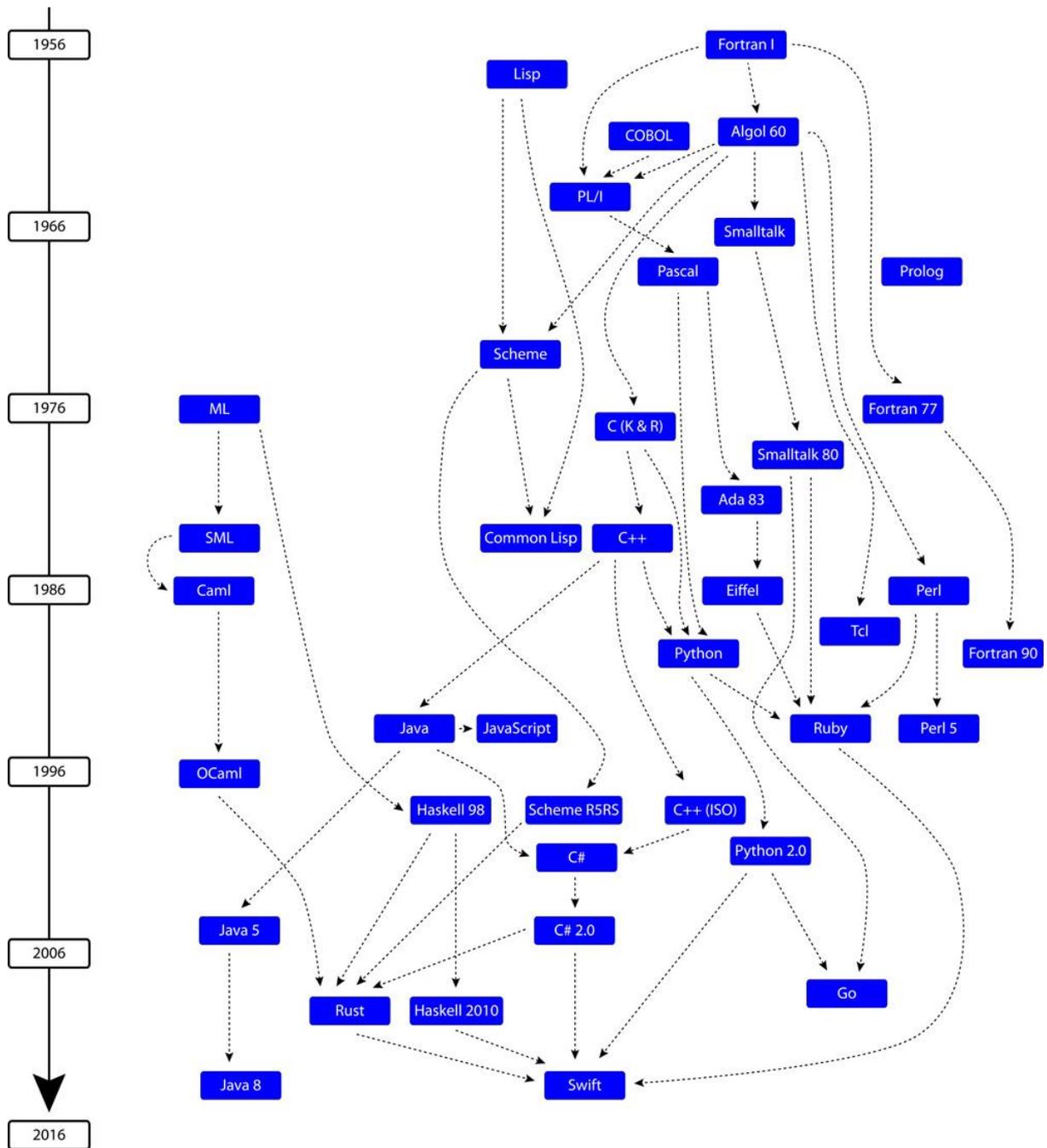
Comment un être humain peut-il commander une machine qui ne fonctionne qu'avec des 0 et des 1 ? Par un langage intermédiaire, dit langage de programmation, qui est ensuite traduit en langage machine fait de 0 et de 1. Tous les langages de programmation servent à expliciter des algorithmes sous forme de programmes (les fameuses *lignes de code*) exécutables par une machine tout en restant lisibles par un humain. Les programmes complexes comme un pilote automatique d'avion ou un jeu vidéo sont une combinaison à grande échelle de programmes élémentaires.

Un **compilateur** est un traducteur automatique de langages de programmation. Typiquement, il sert à traduire un programme écrit dans un langage de programmation en langage machine (des 0 et des 1), se faisant l'interprète entre l'humain et la machine.

Un bug est une erreur de conception dans un programme. La complexité des systèmes informatiques est telle que ces erreurs sont fréquentes et les mises à jour régulières. Mais quand un programme pilote un avion, un robot chirurgical ou gère une centrale nucléaire, la tolérance au bug est minimale. Des programmes d'analyse de programmes sont conçus pour traquer les bugs et augmenter la fiabilité des systèmes.

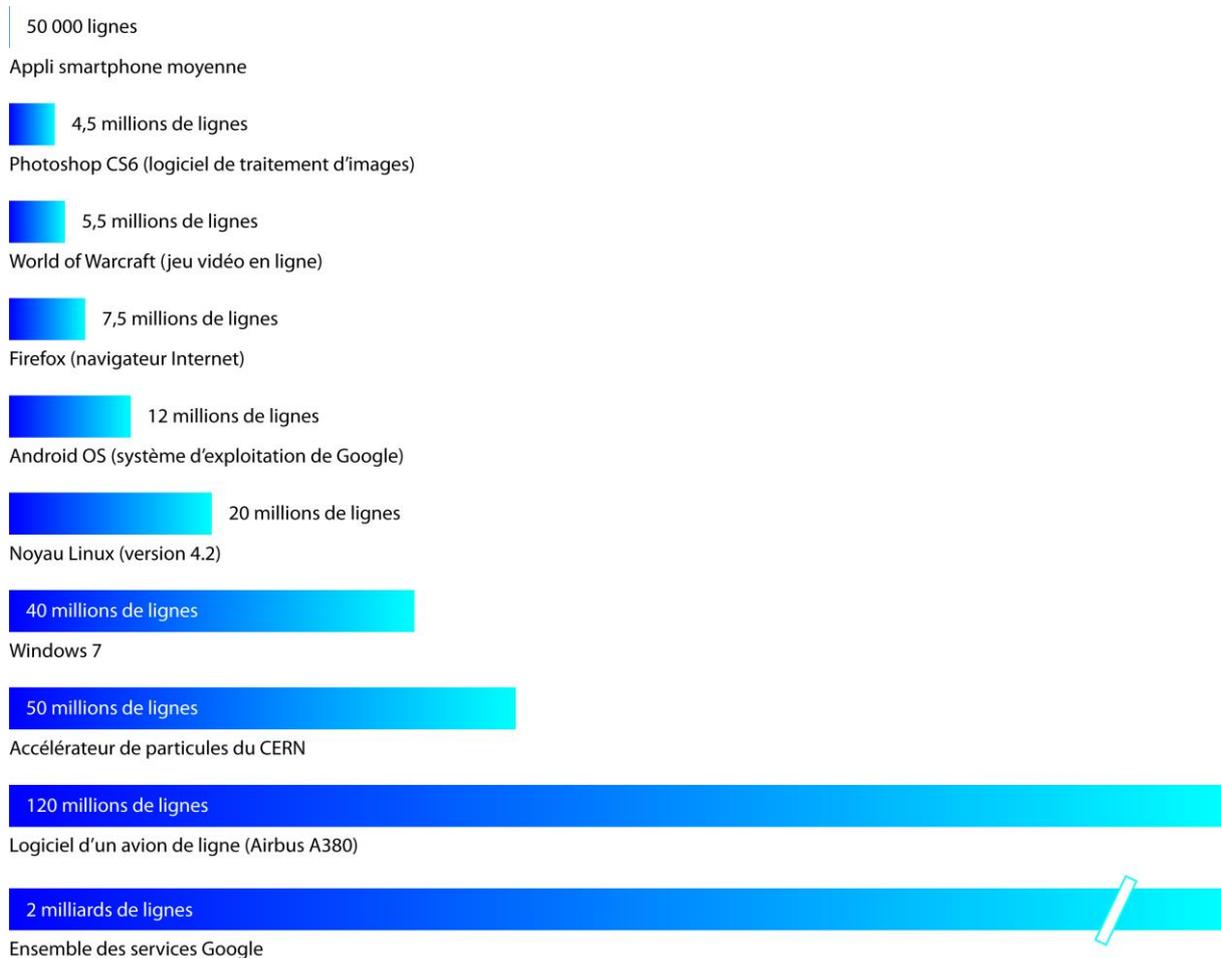


Photographie du premier cas réel de « bug » (*insecte* en anglais), en 1947, dans le journal d'entretien de l'ordinateur Harvard Mark II. Le terme « bug » existait déjà. Trouver un bug réel était donc amusant.
Crédit : Naval Surface Warfare Center, Dahlgren, VA., 1988.



Cet arbre généalogique des principaux langages informatiques ne donne qu'un faible aperçu de leur nombre total, qui s'élèverait à plus de 2500.

Le nombre de lignes de code dans un programme est un indicateur approximatif de sa complexité. Nous utilisons quotidiennement des programmes de 10 à 100 millions de lignes.



Histogramme de tailles de programmes (2013).

Hello world! sont les mots traditionnellement écrits par un programme dont le but est de faire la démonstration rapide d'un langage de programmation ou de tester un compilateur. Dans l'exposition, vous trouverez un panneau présentant les codes pour afficher « Hello world! » dans quinze langages de programmation différents. En voici un aperçu... parfois croustillant.

Ada

```
with Ada.Text_IO;
procedure Hello is
begin
Ada.Text_IO.Put_Line("Hello world!");
end Hello;
```

COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. hello-world.
PROCEDURE DIVISION.
DISPLAY "Hello world!"
.
```

LaTeX

```
\documentclass{minimal}
\begin{document}
Hello world!
\end{document}
```

FORTRAN 90

```
program helloworld
print *, "Hello world!"
end program helloworld
```

PostScript

```
%!PS
/Courier
20 selectfont
72 500 moveto
(Hello world!) show
Showpage
```

C++

```
#include<iostream>
int main()
{
std::cout << "Hello world!\n";
}
```

Go

```
package main
import "fmt"
func main() {
fmt.Printf("Hello world!\n")
}
```

Pascal

```
program HelloWorld(output);
begin
Write('Hello world!')
end.
```

Perl

```
#!/usr/bin/perl
print "Hello world!\n";
```

HTML5

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Hello world!</title>
</head>
<body>
<p>Hello world!</p>
</body>
</html>
```

Malbolge

```
('&#:9]!~}|z2Vxwv-,POqponl$Hjig%eB@@>}=<M:9wv6WsU2T|nm-,jcl(l&%$#"`CB]V?Tx<uVtT`Rpo3NIF.Jh++FdbCBA@?])!~|4XzyTT43Qsqq(Lnmkj"Fhg${z@>
```

Malbolge est un langage de programmation exotique inventé par Ben Olmstead en 1998 et nommé d'après le huitième cercle de l'Enfer dans la *Divine comédie* de Dante, le Malebolge. Particulièrement contre-intuitif, il a été créé pour être (presque) impossible à utiliser...

Vos élèves pourront expérimenter la programmation à l'aide de l'*Encodator*, qui leur permettra de voir comment transformer une tâche, comme classer des chansons, en codes informatiques. L'*Algographe* quant à lui leur offrira l'occasion de manipuler eux-mêmes des codes pour dessiner de superbes dessins projetés sur un mur.

I.2.1.3 L'information

Votre âge, votre chanson préférée, la taille de l'univers et bien d'autres choses sont des informations. Elles peuvent toutes être retranscrites, par exemple, sous forme de texte. Il est possible d'associer à chaque lettre une série de 0 et de 1. Donc, toutes les données de l'univers sans distinction peuvent être consignées sous la forme 0100110101... Pourquoi des 0 et des 1 ? Car cet alphabet à deux caractères, ou *binaire*, est le plus simple. Chaque 0 ou 1 est un bit : une unité d'information qu'une machine peut stocker, répliquer, transmettre, encrypter, analyser, etc.

Bit, byte et octet : quelques définitions et rappels d'informatique

- Un **bit** (pour **binary digit**) est l'élément de base avec lequel travaille un ordinateur. Sa valeur est 0 ou 1, que l'on peut interpréter en oui/non ou vrai/faux. La capacité d'une puce de mémoire s'exprime couramment en bits et les taux de transfert de données, en bit/s.
- **Un octet est un ensemble de 8 bits**. Son symbole est *o*. Un ordinateur ne travaille jamais sur 1 bit à la fois mais sur un ou plusieurs octets, toujours donc sur des multiples de 8 bits. Les premiers ordinateurs personnels étaient 8 bits et ne comptaient que sur un octet à la fois. Les systèmes d'exploitation ont évolué pour passer de 8 bits à 16 bits (la série des Windows 1 à Windows 3.xx), à 32 bits (Windows 95 à Windows 10 32 bits) et enfin à 64 bits (Windows XP 64 bits à Windows 10 64 bits).

- Un **byte** est la plus petite unité adressable d'un ordinateur. Son symbole est *B*. Les bytes de 8 bits ayant très largement supplanté les autres en informatique, on exprime généralement les capacités de mémoire informatique en octet, surtout en français. **Octet et byte sont alors synonymes**. Sachez toutefois que, jusque dans les années 1970, il existait des processeurs avec des bytes de 6, 7 et 9 bits. Aujourd'hui encore, des processeurs utilisant des mémoires adressables par quantité de 4 bits sont utilisés pour programmer des automates ou des équipements industriels simples. De même, les bytes peuvent contenir plus de 8 bits dans le langage C. En bref, en anglais comme en français, on utilise le mot *octet* si l'on veut désigner explicitement une quantité de huit bits ; pour exprimer l'unité d'adressage indépendamment du nombre de bits, on utilise le terme *byte*.

Bit, byte et octet : leurs multiples

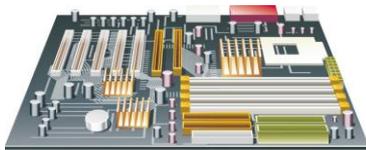
Comme nous venons de le voir, un bit ne peut avoir que deux valeurs, 0 ou 1. Un bit correspond donc à $2^1 = 2$ possibilités, deux bits à $2^2 = 4$ possibilités, trois bits à $2^3 = 8$ possibilités et dix bits à $2^{10} = 1\,024$ possibilités. Ainsi, les puissances de deux interviennent naturellement dans la mesure de la capacité des mémoires d'ordinateurs et des périphériques de stockage. Or, par une coïncidence arithmétique, il se trouve que 1 000, qui correspond au préfixe *kilo*, est une bonne approximation de $2^{10} = 1\,024$. De même, 10^6 (préfixe *méga*) est assez proche de $2^{20} (= 1\,048\,576)$ et 10^9 (préfixe *giga*), de $2^{30} (= 1\,073\,741\,824)$. Les premiers informaticiens ont profité de ces coïncidences pour appliquer les préfixes du Système international d'unités *kilo*, *méga*, *giga*, etc. à des unités qui nécessiteraient des préfixes différents. Si l'erreur ainsi commise était faible pour les premières capacités mémoires qui s'exprimaient en kilooctets (2,4 %), elle est devenue difficilement tolérable avec les capacités actuelles qui se montent en gigaoctets (7,4 %) voire en téraoctets (10,0 %). À la fin des années 1990, la Commission électrotechnique internationale a donc publié une norme qui, d'une part, stipule que les préfixes du Système international d'unités ont toujours leurs valeurs de puissances de dix et ne doivent jamais être utilisés comme puissance de deux, et, d'autre part, introduit les préfixes binaires suivants pour représenter les puissances de deux : *kibi* (pour **kilo binaire**), *mébi* (pour **méga binaire**), *gibi* (pour **giga binaire**), *tébi* (pour **téra binaire**), etc.

Pour les multiples de l'octet, cela donne :

1 kibiocet	(kio)	= 2^{10} octets = 1 024 octets	
1 mébiocet	(Mio)	= 2^{20} octets = 1 024 kio	= 1 048 576 octets
1 gibiocet	(Gio)	= 2^{30} octets = 1 024 Mio	= 1 073 741 824 octets
1 tébiocet	(Tio)	= 2^{40} octets = 1 024 Gio	= 1 099 511 627 776 octets
1 pébiocet	(Pio)	= 2^{50} octets = 1 024 Tio	= 1 125 899 906 842 624 octets
1 exbiocet	(Eio)	= 2^{60} octets = 1 024 Pio	= 1 152 921 504 606 846 976 octets
1 zébiocet	(Zio)	= 2^{70} octets = 1 024 Eio	= 1 180 591 620 717 411 303 424 octets
1 yobiocet	(Yio)	= 2^{80} octets = 1 024 Zio	= 1 208 925 819 614 629 174 706 176 octets

Les préfixes du Système international d'unités correspondent aux mêmes multiplicateurs que dans les autres domaines. Ainsi :

1 kilooctet	(ko)	= 10^3 octets = 1 000 octets	
1 mégaoctet	(Mo)	= 10^6 octets = 1 000 ko	= 1 000 000 octets
1 gigaoctet	(Go)	= 10^9 octets = 1 000 Mo	= 1 000 000 000 octets
1 téraoctet	(To)	= 10^{12} octets = 1 000 Go	= 1 000 000 000 000 octets
1 pétaoctet	(Po)	= 10^{15} octets = 1 000 To	= 1 000 000 000 000 000 octets
1 exaoctet	(Eo)	= 10^{18} octets = 1 000 Po	= 1 000 000 000 000 000 000 octets
1 zettaoctet	(Zo)	= 10^{21} octets = 1 000 Eo	= 1 000 000 000 000 000 000 000 octets
1 yottaoctet	(Yo)	= 10^{24} octets = 1 000 Zo	= 1 000 000 000 000 000 000 000 000 octets



L'usage traditionnel et erroné reste largement en vigueur chez les professionnels comme le grand public (le document que vous lisez ne fait pas exception), même si c'est en contradiction avec les recommandations qui définissent clairement d'autres préfixes. L'usage des préfixes binaires reste très confidentiel et ne se répand presque pas dans le langage courant, alors que les valeurs représentées par ces unités en puissance de deux sont très utilisées dans les applications, notamment les systèmes d'exploitation. Cependant, leur utilisation commence à se répandre.

Cette distinction entre préfixes binaires et décimaux est nécessaire, car la confusion est utilisée depuis longtemps par les fabricants de disques durs. Le fait que l'usage (pour une même capacité) de préfixes en puissances de dix permette d'afficher commercialement des valeurs supérieures à celles données par les puissances de deux peut introduire une erreur d'appréciation de la part d'utilisateurs non avertis. Ainsi, un disque dur de 2 To ($2 \cdot 10^{12}$ octets) contient le même nombre d'octets qu'un disque de 1,819 Tio ($1,819 \cdot 2^{40}$ octets).

La théorie de l'information

Élaborée par Claude Shannon en 1948, elle posa les fondements de la télécommunication. Le problème que Shannon cherchait à résoudre était de transmettre des messages de la manière la plus efficace et la plus fiable possible. Pour cela, il supposa que la quantité d'informations que contient un message peut être mesurée en nombre de bits. Le message « noir ou blanc », par exemple, contient un seul bit d'information, qui vaut soit 0 pour « noir » soit 1 pour « blanc ».

Les machines numériques ne traitent que les 0 et les 1, nous devons leur fournir toutes les données – texte, image, sont, etc. – sous cette unique forme : c’est ce qu’on appelle numériser l’information.

Numériser l’image

L’image est d’abord découpée en *pixels* (pour *picture elements* ou éléments d’image). Dans ce portrait, il y en a 1 600. Ensuite, le niveau de bleu de chacun d’eux est encodé. Ici, il n’y a que quatre niveaux de bleu, une paire de bits suffit donc à encoder chaque pixel : 00 = blanc, 01 = clair, 10 = moyen et 11 = sombre. Pour encoder une photographie en couleurs qui comprend bien plus de nuances, il faudrait évidemment beaucoup plus de bits par pixel.

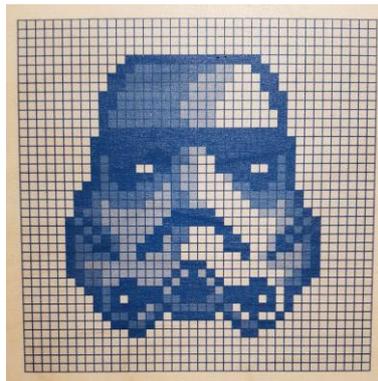
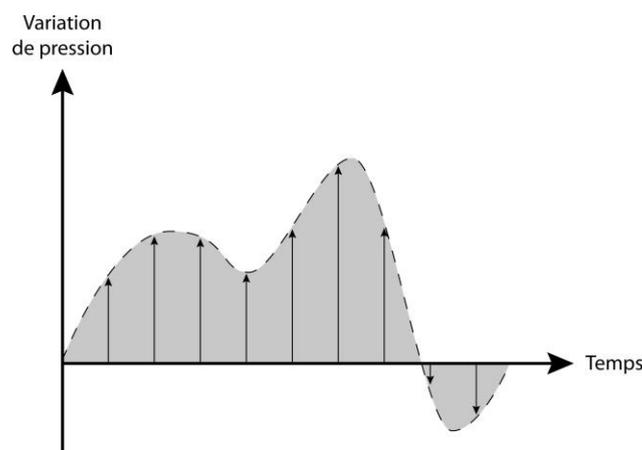


Image de la taille d’une icône.

Numériser le son

Le son résulte de variations continues de pression qui se propagent dans l’air. Pour le numériser, on le découpe en une suite de mesures de pression à une fréquence bien choisie. Ce sont les valeurs de ces mesures qui sont ensuite encodées en bits.



Échantillonnage d’une onde sonore.

Numériser le texte

On peut associer à chaque caractère de texte un octet (une séquence de huit 0 et 1), comme la norme ASCII. Cela permet d'encoder 256 caractères. Mais pour inclure les alphabets chinois, arabe, les émoticônes, etc., il a fallu développer l'UTF-8, une norme d'encodage sur un à quatre octets, qui permet de représenter plus d'un million de caractères.

0	00110000
1	00110001
2	00110010
3	00110011
4	00110100
5	00110101
6	00110110
7	00110111
8	00111000
9	00111001
A	01000001
B	01000010

C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000
I	01001001
J	01001010
K	01001011
L	01001100
M	01001101
N	01001110

O	01001111
P	01010000
Q	01010001
R	01010010
S	01010011
T	01010100
U	01010101
V	01010110
W	01010111
X	01011000
Y	01011001
Z	01011010

Les dix chiffres et l'alphabet en code ASCII.

Vos élèves et vous expérimenterez la numérisation de votre portrait et la manipulation de ses bits avec le multimédia *Pixelomaton*.

I.2.1.4 Les machines

Les machines peuvent-elles traiter l'information à notre place ? Cette ambition est à l'origine de l'informatique. Les premières machines étaient mécaniques, programmées par des instructions sur des cartes perforées. L'invention du tube à vide a permis la réalisation des premières machines électroniques, les ordinateurs. Machines universelles, capables d'exécuter n'importe quel algorithme, leur fonctionnement a été perfectionné, mais il a peu changé fondamentalement : un processeur exécute les opérations élémentaires d'un programme sur des données binaires (les bits) qu'il lit et écrit à partir d'une mémoire ou d'un périphérique.

Quelles que soient les machines utilisées, l'ensemble des problèmes qu'elles peuvent résoudre est le même. Selon la thèse Church – Turing élaborée vers 1940, toutes les manières de calculer que l'on a pu imaginer sont équivalentes, que l'on compte avec un supercalculateur ou simplement en ajoutant des cailloux (*calculus* en latin) ; seule la vitesse diffère. Ce constat est l'un des fondements de l'informatique en tant que science : c'est la science de ce qui peut être calculé.

Le mathématicien britannique George Boole (1815 – 1864) a établi vers 1850 un système de raisonnement logique que les informaticiens utiliseront pour programmer les ordinateurs.

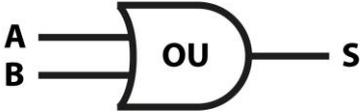
Ce système logique est basé sur le vrai et le faux, complété d'opérations logiques comme « ET », « OU » et « NON ». Par exemple, pour pouvoir sortir, une élève doit analyser la situation suivante :

- A = elle n'a pas de devoir à faire ;
- B = il pleut ;
- C = elle a un parapluie.

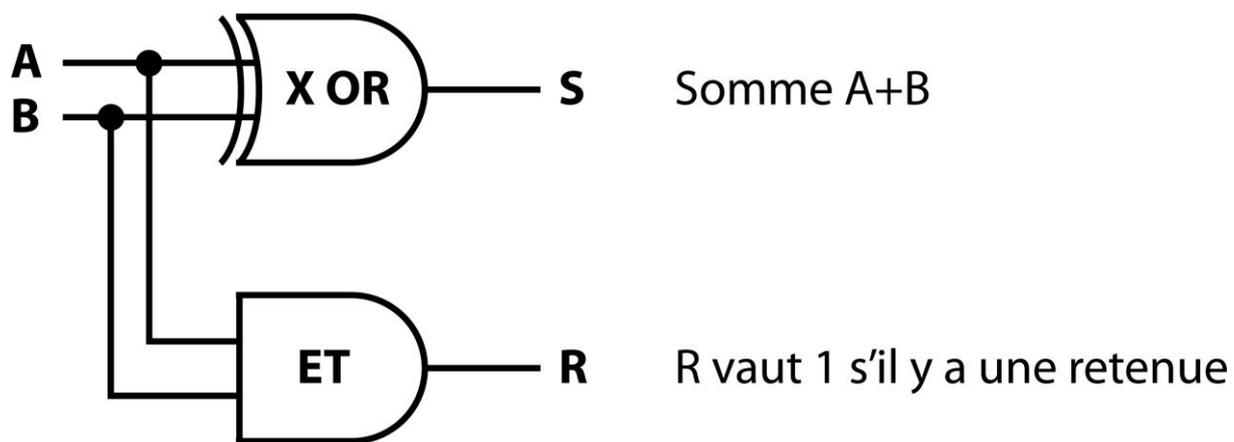
Elle pourra sortir si A est vrai et B est faux, OU, si B est vrai, il faudra que C soit vrai. Ce qu'on peut écrire sous la forme (A ET NON B) OU (A ET B ET C).

Ce système peut être transcrit en signaux électriques pour les machines. L'état 0 pour « faux » quand le courant ne passe pas, et l'état 1 pour « vrai » quand le courant passe.

Pour fabriquer un état à deux niveaux 0 ou 1 de Boole, il faut un interrupteur (on – off). Le transistor, petit, rapide et économe en énergie, remplit parfaitement ce rôle. Un assemblage astucieux de transistors, diodes et résistances, permet de créer des opérateurs logiques, qui sont ensuite combinés pour former des circuits intégrés. Un microprocesseur de smartphone est un circuit intégré de plusieurs milliards de transistors, chacun enchaînant des opérations logiques booléennes pour réaliser toutes ces tâches qui vous servent au quotidien.

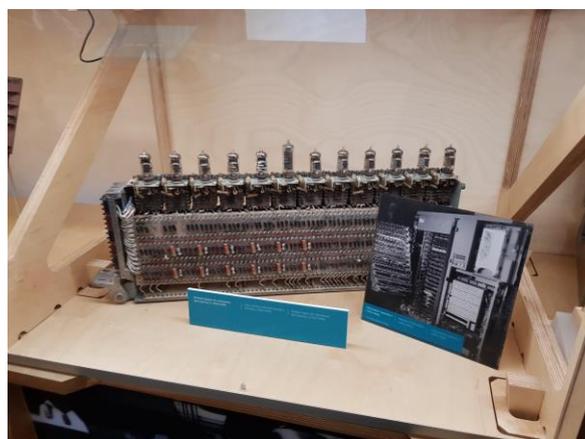
Opérateur logique	Règle
	<p>Si A = 1 ET B = 1 alors S = 1 sinon S = 0</p>
	<p>Si A = 1 OU B = 1 alors S = 1 sinon S = 0</p>
	<p>Si (A = 1 ET B = 0) OU (A=0 ET B=1) alors S = 1 sinon S = 0</p>

Exemples d'opérateurs logiques (ou portes logiques).



Circuit logique de l'addition binaire. La retenue R peut être connectée à l'entrée d'un autre circuit d'addition. Il faut en combiner près d'un millier pour réaliser l'addition des processeurs actuels.

Dans cet espace, vos élèves pourront découvrir une galerie de l'évolution numérique dont voici quelques exemples.



I.2.2 Le laboratoire

Cette partie de l'exposition aborde des thèmes de recherche actuels : le big data, l'apprentissage automatique (liée à l'intelligence artificielle), les réseaux, les robots.

I.2.2.1 Le big data

Le Big Data se définit par d'énormes volumes de données en constante évolution et aux formats variés. Par exemple, les données de Facebook ou de Google et leur exploitation par ces opérateurs constituent du Big data. L'élément d'exposition *Big Data* vous permettra de visualiser et de consulter des données urbaines. Vous pourrez aussi tester une application de prédiction des flux de Vélib', des accidents de la route, du confort de la gare ferroviaire d'Aix en Provence ou de la Place de la Nation à Paris. Une fresque graphique permet d'apprendre les ordres de grandeur du Big Data et de découvrir les merveilles de la datavisualisation.

Les données sont la matière première de la révolution numérique qui fait émerger de nouveaux rapports entre les citoyens, les États et les entreprises. La croissance exponentielle et vertigineuse de leur production justifie leur place centrale dans le discours et les représentations de l'exposition. La profusion des données a donc mené à la naissance d'un nouveau domaine technologique : le *Big Data* ou *mégadonnées*, terme d'ailleurs recommandé par la Délégation générale à la langue française et aux langues de France, un service rattaché au ministère de la Culture. Il est important de comprendre dès maintenant que l'objet du Big Data n'est pas l'information mais bien la donnée elle-même. Même s'il manque d'une définition sérieuse, le Big Data regroupe une famille d'outils qui permettent de stocker, traiter et analyser des déluges de données hétérogènes afin d'en faire ressortir de la valeur et, dans le cas des entreprises, de la création de richesse. Ces outils répondent à une problématique triple : variété, volume et vitesse. C'est la règle des 3V... qui à vrai dire, est plus un slogan susceptible d'évoluer en fonction de stratégies commerciales qu'une véritable règle. Les premiers outils du Big Data ont été créés par les entreprises chefs de file du web comme Google, Amazon ou Yahoo.

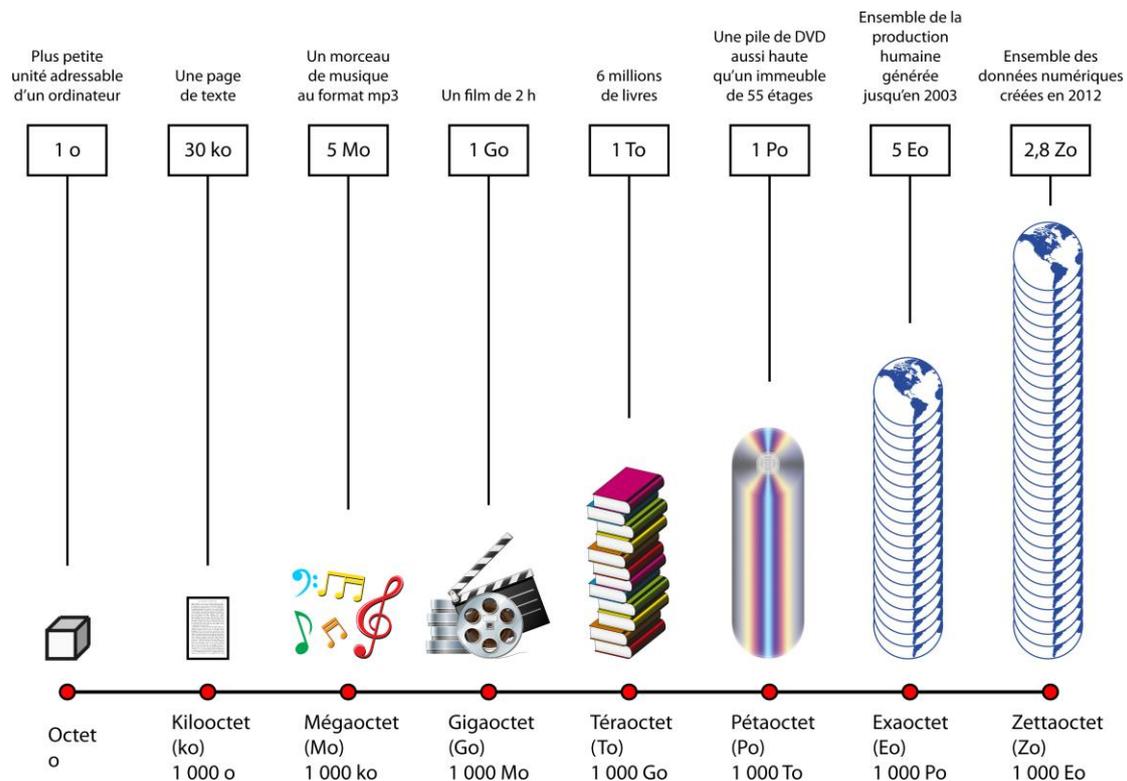
Revenons à la « règle » des 3V qui caractérisent le Big Data.

V comme... variété des données

Une donnée est la représentation d'une information sous une forme qui permettra de la stocker, de la transmettre, de la transformer et de l'analyser. Dans le domaine informatique, on parle de données numériques, c'est-à-dire des données compréhensibles par un ordinateur. Qu'elles soient structurées dans des tableaux répertoriant, par exemple, des personnes avec noms, dates de naissance, numéros de téléphone etc., ou non-structurées comme les images, les vidéos ou les textes qui circulent par les réseaux, la variété de ces données s'enrichit à mesure que se développent de nouveaux outils de collecte. Web, smartphones, objets connectés, les capteurs se multiplient dans notre quotidien et entreprennent de numériser le monde réel en y prélevant des données de tous types.

V comme... volume des données

La quantité de données générée dans le monde est en pleine expansion et suit une loi quasi exponentielle. Les chiffres parlent d'eux-mêmes. Nous générons aujourd'hui en une journée plus de données qu'il n'en a été produit entre les débuts de l'humanité et l'an 2000. Chaque seconde, plus d'une heure de vidéos est téléchargée vers le site web d'hébergement YouTube et plus de 1,5 million de courriers électroniques sont envoyés à travers le monde.



Les scientifiques ne sont pas en reste. En seulement huit ans (de 2000 à 2008), le *Sloan Digital Sky Survey* (SDSS), un programme de relevés des objets célestes utilisant un télescope dédié de 2,5 mètres de diamètre au Nouveau-Mexique, a enregistré 140 téraoctets d'images, soit 140 000 gigaoctets. Son successeur, le *Large Synoptic Survey Telescope* (LSST) collectera la même quantité de données tous les cinq jours ! Ce télescope, dont le miroir primaire possédera un diamètre de 8,4 mètres, devrait voir sa première lumière dans les montagnes chiliennes en 2019 et sera muni d'une caméra CCD de 3,2 gigapixels. Le plus puissant accélérateur de particules construit à ce jour, le *Large Hadron Collider* (LHC), produit, après filtrage, 25 pétaoctets de données (25 millions de gigaoctets) chaque année soit l'équivalent de plus de trois millions de DVD. Enfin, le projet le plus ambitieux est le *Square Kilometer Array* (SKA), un réseau de plusieurs dizaines de milliers d'antennes réparties entre l'Australie et l'Afrique du Sud, qui sera équivalent à un seul radiotélescope doté d'une surface collectrice de 1 km². SKA devrait être construit en deux phases. À la fin de la première phase en 2023, alors que le radiotélescope ne sera exploité qu'à 10 % de ses capacités, ce ne seront pas moins de 157 téraoctets de données brutes – soit l'équivalent de plus de 30 000 DVD – que les superordinateurs devront traiter... chaque seconde !



Vue d'artiste du Large Synoptic Survey Telescope. Crédit : LSST.

Tous ces nombres sont impressionnants et donnent le vertige. Est-ce toujours le cas lorsqu'on les compare à la quantité d'informations générée par le vivant ? L'information contenue dans l'ADN des êtres vivants possède la plus forte concentration d'informations connue à ce jour. Les calculs montrent qu'en théorie, un seul gramme d'ADN pourrait stocker environ 400 exaoctets de données, soit 400 milliards de gigaoctets !

V comme... vitesse des données

Accumuler beaucoup d'informations est une chose mais encore faut-il savoir les traiter à temps pour qu'elles restent pertinentes. La vitesse est donc l'un des facteurs d'émergence de ce nouveau rapport aux données numériques et se caractérise par l'utilisation de technologies de pointe pour obtenir des performances inédites. Produites de plus en plus rapidement, transmises à la vitesse de la lumière dans des fibres optiques et traitées par des ordinateurs de plus en plus puissants, les données sont ainsi transformées en informations exploitables à une vitesse proche du temps réel. Le risque pour l'Homme est de perdre une grande partie de la maîtrise du système quand, dans le cas des transactions boursières, les opérateurs sont des algorithmes informatiques capables de lancer des ordres d'achat ou de vente en quelques microsecondes, sans disposer de tous les critères pertinents d'analyse pour le moyen et long terme.

On emploie diverses unités de mesure pour quantifier la puissance de calcul d'une machine et déterminer ses performances de manière universelle. Parmi ces unités : les FLOPS, ou « opérations en virgule flottante par seconde » (en anglais : *F*loating-*p*oint *O*perations *P*er *S*econd). Un téléphone d'un gigaFLOPS peut réaliser un milliard d'opérations sur des valeurs non entières en une seconde.

La puissance de calcul des ordinateurs a tellement évolué que les machines personnelles rivalisent aisément avec les machines les plus imposantes d'il y a quelques années. Le système de guidage de la fusée Apollo 11, en 1969, n'était que deux fois plus puissant que la console grand public Nintendo Entertainment System (NES) commercialisée 16 ans plus tard.

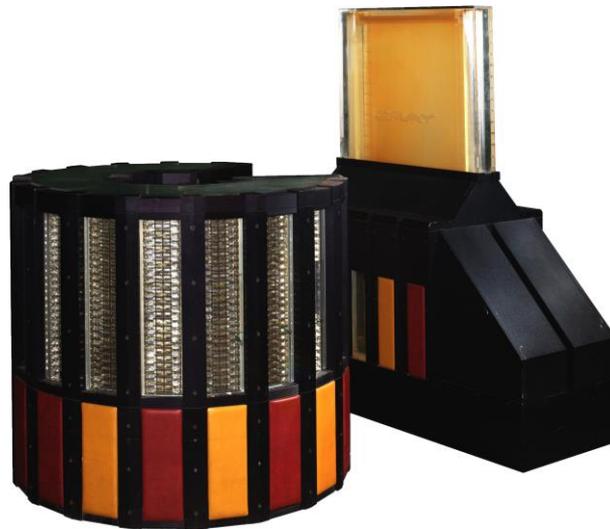
Voici quelques comparaisons particulièrement révélatrices.

1 superordinateur Cray-2 de 1985 (1,9 gigaFLOPS) \approx **1** Apple iPhone 4 de 2010 (1,6 gigaFLOPS)

1 Samsung Galaxy S6 de 2015 (34,8 gigaFLOPS) \approx **5** PlayStation 2 de 2000 (6,2 gigaFLOPS chacune)

1 ordinateur de bureau typique de 2017 (90 gigaFLOPS) \approx **8** superordinateurs Deeper Blue de 1997 (11,4 gigaFLOPS). Cet ordinateur, spécialisé dans le jeu d'échecs, avait battu le champion du monde de l'époque Garry Kasparov.

1 superordinateur Sunway TaihuLight de 2016 (93,0 pétaFLOPS) \approx **22 100** PlayStation 4 Pro de 2016 (4,2 téraFLOPS chacune). Ce supercalculateur chinois est le plus puissant au monde en mars 2017.



Unité centrale du supercalculateur Cray-2 de 1985 (à gauche) et son système de refroidissement (à droite) exposés à l'École polytechnique fédérale de Lausanne (EPFL). Crédit : Rama / English Wikipedia.

V comme... **v**éracité, **v**aleur, **v**ert

Que ce soit dans la collecte des données, dans la manière de les recouper, de les croiser, il devient de plus en plus crucial de veiller à la véracité des données, leur précision, leur pertinence par rapport au domaine dans lequel elles sont étudiées. Mais une donnée isolée n'a aucune valeur.

C'est dans l'accumulation et le recoupement que réside la valeur des données. Aujourd'hui, c'est la création de valeur (économique, scientifique, culturelle) qui est le principal moteur du développement d'outils. De plus, la dimension écologique des données ne doit pas être négligée. En effet, les premiers ordinateurs, tel l'ENIAC, consommaient autant d'électricité qu'une petite ville. Aujourd'hui, un téléphone est plus puissant qu'eux, mais il ne consomme presque rien. Toutefois, les ordinateurs sont nombreux. Leur consommation globale est donc importante. Ils sont responsables de 2 % des émissions de dioxyde de carbone, ce qui est comparable aux émissions du transport aérien. S'ils participent au problème, ils aident aussi à chercher des solutions : les simulations numériques sont essentielles pour comprendre les dérèglements climatiques.

Quelques chiffres : en 2010, Google utilisait 900 000 serveurs répartis dans différents centres de données à travers le monde. Leur consommation globale s'élevait à 260 millions de watts, comme une ville française de 430 000 habitants.

Finalement, en quoi les techniques du Big Data diffèrent-elles des techniques traditionnelles d'analyse des bases de données ? La différence est de nature conceptuelle. La fouille de données classique s'appuie sur un modèle, ce qui revient à adopter un raisonnement déductif. Au contraire, le Big Data consiste à chercher, par induction, des modèles prédictifs dans de grands volumes de données à faible densité en information.



Le Centre de données de l'Utah (Utah Data Center) est un centre de stockage et de traitement de données géré par la NSA (National Security Agency) pour le compte de la Communauté du renseignement des États-Unis. La capacité de stockage de ce centre est sujette à débat, l'information étant, bien évidemment, confidentielle. Des estimations font état de quelques exaoctets... à un millier de zettaoctets (ce que l'on appelle un yottaoctet). En l'état actuel de nos connaissances, cette dernière valeur est peu crédible des points de vue technologique, énergétique et financier.

I.2.2.2 L'apprentissage automatique

Il n'y a rien d'intelligent dans l'intelligence artificielle. L'un des principaux ingrédients de ce domaine est l'apprentissage automatique, c'est-à-dire la capacité de concevoir des algorithmes, des codes, capables d'apprendre à partir de données massives. C'est ainsi que vous êtes parfois étonné de voir une machine capable de reconnaître un dessin. Vos élèves joueront au professeur avec *l'Apprenti Illustrateur*. Une fresque graphique leur expliquera également ce qu'est un neurone artificiel, leur proposera de découvrir l'histoire du Perceptron et des premiers réseaux de neurones, ainsi que la diversité des méthodes d'apprentissage.

Les algorithmes d'apprentissage automatique

S'il existe de nombreux algorithmes pour résoudre divers problèmes (comme effectuer un tri rapide) ceux-ci fournissent toujours un résultat identique pour de mêmes entrées. Quand le problème est trop compliqué, on a recours à des méthodes qui permettent aux machines d'apprendre à partir des données qui leur sont soumises.

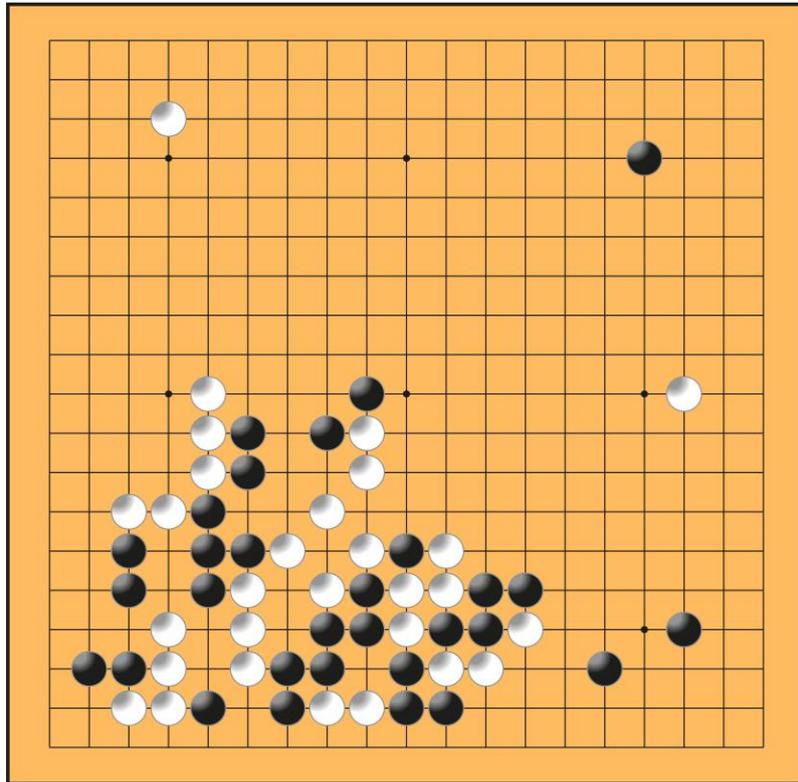
Un humain n'a aucun mal à reconnaître un chien. Mais un ordinateur ? Formaliser toutes les variantes possibles du « chien » est bien trop complexe pour écrire un algorithme classique capable, par exemple, de repérer cet animal sur une image. Un algorithme d'apprentissage automatique va s'appuyer sur de nombreuses images où des personnes ont marqué la présence d'un chien. Il va apprendre à reconnaître l'animal sans passer par une définition exhaustive de « chien », un peu comme un enfant apprend à parler par imitation, sans connaître la définition exacte des mots et la grammaire. Des exemples familiers ? Votre filtre anti-spams, qui s'appuie sur vos choix pour écarter des mails indésirables, et les articles « que vous devriez aimer » sur les sites de vente en ligne.

C'est en 1997 que le superordinateur Deep Blue, et plus précisément sa version améliorée Deeper Blue, a remporté un tournoi d'échecs contre le champion du monde Garry Kasparov. Mais comment ? L'ordinateur s'est appuyé sur des parties perdues par Kasparov contre des humains. À chaque mouvement de Kasparov, Deep Blue puisait dans ce répertoire. Puis, lorsque suffisamment de pièces furent éliminées, le nombre de coups possibles a diminué. Il est devenu plus facile à Deep Blue d'envisager tous les cas possibles et de dominer la partie.



En mai 1997, le champion du monde d'échecs Garry Kasparov s'incline face à Deeper Blue, capable de calculer de 100 à 300 millions de positions par seconde. Deeper Blue défait Kasparov 3,5 à 2,5 dans un match à 6 parties.

Le jeu de go est un jeu de plateau très populaire en Asie de l'Est. Pour un ordinateur, il est plus complexe à maîtriser que le jeu d'échecs. Pourquoi ? Ceci est principalement dû au fait que le plateau de go est bien plus étendu, que la plupart des coups sont légaux et souvent plausibles, et également au fait que la capture des pions rend possible de rejouer dans les espaces ainsi libérés. Pourtant, l'algorithme développé par la société DeepMind, AlphaGo, a réussi à battre en mars 2016 le champion coréen Lee Sedol. Pour gagner, AlphaGo s'est appuyé sur l'apprentissage automatique. D'abord, il a analysé un très grand nombre de situations pour apprendre comment des experts avaient joué avec succès. Ensuite, il a joué contre lui-même des millions de parties pour explorer des situations inédites. C'est en combinant des puissances de calcul considérables et des algorithmes d'apprentissage nouveaux que le programme a vaincu ce joueur professionnel, l'un des meilleurs au monde.



Les 59 premiers coups d'une partie de go.

I.2.2.3 Les réseaux

L'information circule à travers des infrastructures matérielles, des réseaux, Internet étant le réseau des réseaux. Venez découvrir à travers le jeu collectif *Réseau Dérouté* comment s'échangent les données à travers un réseau ou encore toucher un câble transatlantique. Une fresque graphique vous expliquera alors ce que sont les serveurs, routeurs et autres data centers.

Un réseau est un ensemble de machines connectées pour échanger de l'information. Votre réseau domestique (ordinateurs, tablettes, TV, etc.) est relié par la *box* au réseau de votre fournisseur d'accès, qui vous connecte au réseau Internet... qui est la plus vaste interconnexion de réseaux au monde : 668 000 réseaux en 2017. À chaque nœud de réseau, un routeur (chez vous, la *box*) assure la bonne distribution de l'information. Internet est donc un réseau décentralisé.

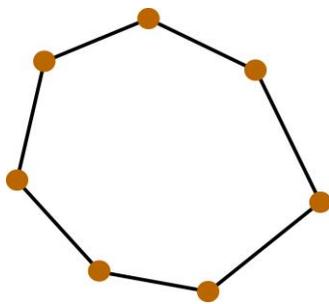
Internet ou web ?

Internet est un réseau de machines. Le Web est l'ensemble des pages Web qui y sont stockées et consultables.

La tendance est inéluctable : des objets connectés par milliards captent des données sur l'environnement, les villes, les maisons, notre santé ou encore les systèmes de production. Montres, pacemakers, alarmes, compteurs électriques connectés existent déjà. Ce sont autant de nouveaux hôtes sur les réseaux, nécessitant chacun leur propre adresse réseau. La dernière version du protocole Internet (IPv6), codant les adresses sur 128 bits, offre de quoi accueillir ces nouveaux venus toujours plus nombreux.

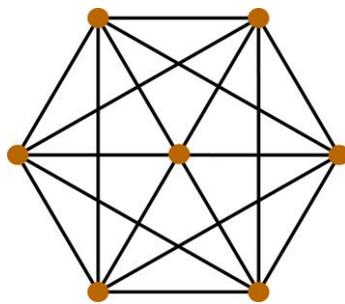
Les topologies de réseau

Qu'il soit planifié ou non, un réseau peut prendre plusieurs formes, appelées topologies. En connectant des centaines de milliers de topologies variées les unes aux autres, on obtient un réseau comme Internet.



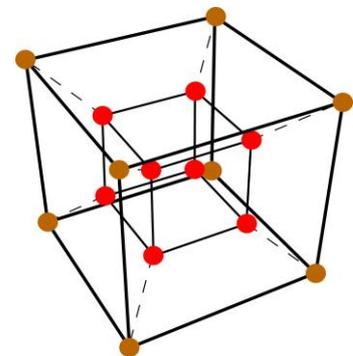
Cycle

Topologie la plus simple résistant à une panne de routeur (les données pouvant faire le tour). C'est souvent le motif de base de réseaux ADSL et fibre.



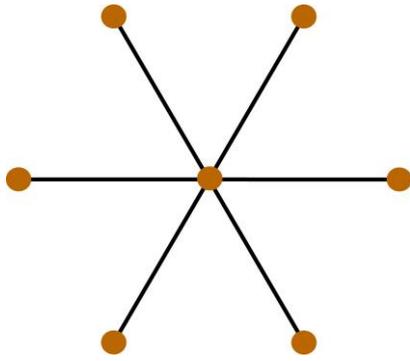
Clique

Topologie idéale, chacun est connecté à tous les autres, mais irréalisable dès qu'on est nombreux.



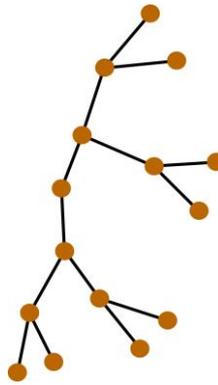
Hypercube

Topologie typique de *data center*, car elle connecte un grand nombre de machines sans surcharge de liens ni de routeurs et offre des chemins courts.



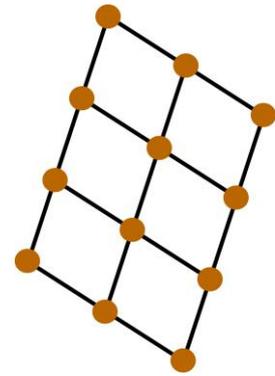
Étoile

Topologie la plus simple. Un routeur centralise toutes les connexions mais ne peut en gérer qu'un nombre limité.



Arbre

Topologie minimale pour connecter un ensemble de routeurs. Le routage est simple car un seul chemin relie deux routeurs.



Grille

Réseau maillé obtenu, par exemple, quand chaque routeur se connecte aux plus proches voisins.

I.2.2.4 Les robots

Un robot est une machine qui interagit physiquement avec son environnement, à l'aide de capteurs pour détecter et d'actionneurs pour agir, selon un programme qui définit son comportement. Cette capacité à percevoir une situation, à l'analyser puis à décider d'une action est ce qui différencie un robot d'un simple automate. La robotique s'inspire largement du vivant pour imaginer des organes sensoriels (capteurs de pression, de mouvement, de température...), des organes moteurs (mobilité, manipulation, articulations...) et des fonctions de décision (algorithmes d'apprentissage automatique...).

Robotique et intelligence artificielle

Marcher, saisir un objet, taper dans un ballon, explorer un environnement inconnu... ces tâches simples pour nous peuvent s'avérer très complexes pour une machine. Prenons un robot bipède : parmi toutes les combinaisons de mouvements possibles, lesquelles permettent de marcher droit et sans tomber ? Grâce aux algorithmes d'apprentissage par renforcement, il est possible d'explorer ces combinaisons par un jeu d'essais et d'erreurs afin de trouver des mouvements qui fonctionnent.

Voici trois axes de recherche actuels en robotique.

→ La robotique molle. Elle désigne des robots flexibles, capables, par exemple, de se faufiler dans diverses ouvertures. Par ailleurs, un contact « doux » est préférable en médecine (robot chirurgical) et de manière générale à proximité des humains.

→ La robotique développementale. Un bébé qui découvre son corps par interactions avec l'environnement et avec ses congénères acquiert un savoir-faire. De façon similaire, la robotique développementale étudie les mécanismes qui permettraient à un robot de se programmer tout seul par interactions. Elle s'appuie sur les algorithmes d'apprentissage par renforcement.

→ Les essaims de robots. La robotique en essaim permet de faire collaborer de nombreux robots entre eux pour qu'ils réalisent une tâche. La collaboration entre robots individuels, mais aussi avec des systèmes robotiques d'objets connectés entre eux (robots, capteurs, actionneurs), ouvre de nouvelles perspectives.

Dans cet espace, vos élèves auront l'occasion de piloter *Abrarobotix*, un bras robotique semi-autonome qui interprétera leurs mouvements.



Le robot *Abrarobotix*, prêt à saisir sa proie.

I.2.3 L'arène techno

Cette partie de l'exposition montre des produits finis de haute technologie qui sont en train ou vont rentrer dans nos vies.

I.2.3.1 *Fluid*

Au bloc opératoire, les chirurgiens évitent tout contact avec leur environnement afin de respecter des conditions strictes de stérilité. Comment, alors, accéder aux dossiers médicaux des patients en cours d'opération ?

La technologie *Fluid*, développée par la société Therapixel, est une interface homme – machine permettant d'interagir sans contact avec un écran, par commandes gestuelles. Un capteur de mouvements par télémétrie infrarouge (élément conçu au départ pour les jeux vidéo), mesure la position des mains 30 fois par seconde.

Une première phase d'apprentissage supervisé a servi à identifier les mains et leurs positions. Une seconde phase d'apprentissage par renforcement a permis d'associer certains gestes à des commandes simples : sélectionner, déplacer, zoomer, faire défiler... Le fonctionnement doit être parfaitement fiable, fluide et d'une prise en main rapide, afin que la technologie s'efface et que les chirurgiens concentrent toute leur attention sur les patients.

Vos élèves pourront ici se familiariser avec l'utilisation de *Fluid*.

I.2.3.2 *Xplore Touch*

Les écrans dits tactiles sont sensibles au contact mais ne peuvent reproduire le sens du toucher. L'écran présenté ici (*Xplore Touch* de la société Hap2U) propose d'y remédier : il restitue une sensation tactile (fonction haptique) et réagit à la pression (fonction kinesthésique).

Lorsque l'on pose son doigt sur l'écran, la position de celui-ci est détectée et les informations tactiles sont générées par des actionneurs piézoélectriques situés autour de l'écran. Ceux-ci se déforment rapidement sous l'effet d'un courant électrique, provoquant des ondes de fréquence ultrasonique sur la dalle qui recouvre l'écran. Ces micro-vibrations générées dans la dalle permettent de modifier en temps réel le frottement du doigt sur l'écran.

En synchronisant alors les vibrations avec la position du doigt, l'utilisateur peut ressentir l'illusion de textures, comme celle des écailles d'un poisson, ou avoir l'impression d'utiliser la force de son doigt lorsqu'il déplace un curseur virtuel. Des usages sont imaginés pour les smartphones et tablettes, les commandes de tableau de bord, les jeux vidéo ou encore pour le commerce en ligne.



Passez doucement votre doigt sur l'écran et sentez...

I.2.3.3 LiFi

LiFi (pour *light fidelity*) est une technologie développée par la société Lucibel. Il s'agit d'un moyen de communication sans fil qui transmet des informations via la lumière d'un spot lumineux LED. À la différence des protocoles Wi-Fi et Bluetooth, qui reposent sur les ondes radio, c'est ici la lumière visible qui porte l'information descendante (l'information ascendante est émise en lumière infrarouge).

L'intensité de la lumière émise par la LED varie avec une fréquence suffisamment rapide pour rester imperceptible à l'œil humain. Mais un photorécepteur électronique peut détecter ce scintillement et le transmettre comme une suite de 0 et de 1, un message binaire.

La connexion n'est possible qu'au sein d'un cône de lumière, créant ainsi une zone de confidentialité de l'information. De plus, cette technologie offre une alternative aux ondes radio (Wi-Fi) de plus en plus souvent proscrites (avions, hôpitaux, etc.).

Dans cet élément d'exposition, vos élèves pourront tester une visioconférence en LiFi à deux.

II Ressources

II.1 Ateliers et exposés

L'espace permanent *Informatique et sciences du numérique* contient un atelier qui peut accueillir de petits groupes et un amphithéâtre où sont donnés des exposés. Des exposés peuvent également prendre place en salle *Turing (Bbis)*, qui se trouve au niveau 0 du Palais de la découverte, entre la salle *Électrostatique (B)* et la salle *Sons et vibrations (A)*. Vous trouverez un plan détaillé des niveaux 0 et 1 [ici](#).

Un atelier a une durée comprise entre 1 h et 3 h. Il peut accueillir 16 élèves au maximum. Les élèves sont invités à expérimenter et manipuler par eux-mêmes avec du matériel et de la documentation. Un exposé dure 50 minutes et peut s'adresser à une classe entière. Un médiateur scientifique effectue plusieurs expériences et invite les élèves à participer.

Activités pour le primaire

Ateliers

L'informatique sans ordinateur

Pour le CM2

Comment un DVD peut-il contenir un film ? Cette question permettra d'aborder, de manière ludique, les notions d'algorithmes, de données et de leurs représentations, de détections des erreurs... par l'expérience et sans ordinateur !

Robot Thymio

Pour le CM2

Comment programmer un robot pour lui faire suivre une main, éviter un objet ou quitter une pièce ? C'est ce que vos élèves découvriront grâce au robot éducatif Thymio (un robot pour deux élèves) et son langage graphique décrivant ses capteurs et les actions qu'il peut entreprendre. Autant d'applications rencontrées dans des usages quotidiens, dans l'espace public ou dans l'industrie.

Activités pour le collège

Ateliers

L'informatique sans ordinateur

De la 6^e à la 5^e

Comment un DVD peut-il contenir un film ? Cette question permettra d'aborder, de manière ludique, les notions d'algorithmes, de données et de leurs représentations, de détections des erreurs... par l'expérience et sans ordinateur !

Robot Thymio

De la 6^e à la 3^e

Comment programmer un robot pour lui faire suivre une main, éviter un objet ou quitter une pièce ? C'est ce que vos élèves découvriront grâce au robot éducatif Thymio (un robot pour deux élèves) et son langage graphique décrivant ses capteurs et les actions qu'il peut entreprendre. Autant d'applications rencontrées dans des usages quotidiens, dans l'espace public ou dans l'industrie.

Exposés

Salle Turing

Qu'est-ce que l'informatique ?

De la 4^e à la 3^e

L'informatique se compose d'algorithmes, de programmes, d'ordinateurs et de données. Explications sur la nature de ces quatre composantes et leurs interactions, depuis la machine à additionner de Blaise Pascal jusqu'aux applications les plus modernes comme les réseaux sociaux.

Histoire et fonctionnement d'Internet

De la 4^e à la 3^e

Paquets de données, routeurs, IP... Comment fonctionne Internet ? C'est ce que nous découvrirons à travers son histoire, du premier mail aux objets connectés en passant par la création du web.

Algorithmique : combien ça coûte ?

De la 4^e à la 3^e

Au cœur du fonctionnement de nos ordinateurs, les algorithmes se résument à des séquences d'instructions élémentaires. Mais tous ne se valent pas et leur coût varie en termes de mémoire et de temps d'exécution, laissant apparaître un paysage vertigineux.

Un robot, comment ça robotte ?

Pour la 3^e

Découverte de l'histoire des robots, de leur programmation pour effectuer une tâche en apparence simple comme se déplacer, jusqu'aux dernières innovations comme les robots mous ou les robots en essaim...

Textes, images et sons de vos ordinateurs

Pour la 3^e

Numérisées, compressées et enregistrées, comment toutes ces données sont-elles transformées puis stockées (disque dur, DVD, clé USB...) ?

Activités pour le lycée

Exposés

Salle Turing

Qu'est-ce que l'informatique ?

De la 2^{de} à la terminale

L'informatique se compose d'algorithmes, de programmes, d'ordinateurs et de données. Explications sur la nature de ces quatre composantes et leurs interactions, depuis la machine à additionner de Blaise Pascal jusqu'aux applications les plus modernes comme les réseaux sociaux.

Histoire et fonctionnement d'Internet

De la 2^{de} à la terminale

Paquets de données, routeurs, IP... Comment fonctionne Internet ? C'est ce que nous découvrirons à travers son histoire, du premier mail aux objets connectés en passant par la création du web.

Algorithmique : combien ça coûte ?

De la 2^{de} à la terminale

Au cœur du fonctionnement de nos ordinateurs, les algorithmes se résument à des séquences d'instructions élémentaires. Mais tous ne se valent pas et leur coût varie en termes de mémoire et de temps d'exécution, laissant apparaître un paysage vertigineux.

Un robot, comment ça robotte ?

De la 2^{de} à la terminale

Découverte de l'histoire des robots, de leur programmation pour effectuer une tâche en apparence simple comme se déplacer, jusqu'aux dernières innovations comme les robots mous ou les robots en essaim...

Textes, images et sons de vos ordinateurs

De la 2^{de} à la terminale

Numérisées, compressées et enregistrées, comment toutes ces données sont-elles transformées puis stockées (disque dur, DVD, clé USB...) ?

L'informatique et les autres sciences

De la 2^{de} à la terminale

Grâce à l'informatique, les simulations numériques ont révolutionné les sciences comme l'écologie, la physique, les mathématiques ou l'astronomie. Mais comment fonctionnent ces modélisations et quelles en sont les limites ?



La salle d'exposés dans l'espace permanent.



La salle d'ateliers.

II.2 Bibliographie

- Serge Abiteboul et Gilles Dowek, *Le temps des algorithmes*, éd. Le Pommier, coll. Essais et Documents, 2017.
- Serge Abiteboul et Valérie Peugeot, *Terra Data. Qu'allons-nous faire des données numériques*, éd. Le Pommier, coll. Le Collège, 2017.
- Gérard Berry, *L'Hyperpuissance de l'informatique. Algorithmes, données, machines, réseaux*, éd. Odile Jacob, 2017.
- Stéphan Julienne, *La vie connectée. Comprendre comment ça marche...*, éd. Nathan / Cité des sciences et de l'industrie, coll. Dokéo, 2014.
- Emmanuel Lazard et Pierre Mounier-Kuhn, *Histoire illustrée de l'informatique*, éd. EDP Sciences, 2016.



III Informations pratiques

Adresse

Palais de la découverte
Avenue Franklin D. Roosevelt
75008 Paris
Tél. : 01 56 43 20 20
www.palais-decouverte.fr

Accès

Métro : Champs-Élysées Clémenceau (ligne 1 et ligne 13) ou Franklin Roosevelt (ligne 9)
Bus : 28, 42, 52, 63, 72, 73, 80, 83, 93
R.E.R. : Invalides (ligne C)

Horaires d'ouverture

Du mardi au samedi de 9 h 30 à 18 h, le dimanche et jours fériés de 10 h à 19 h.
Fermeture les lundis, le 1^{er} janvier, le 1^{er} mai et le 25 décembre.

Tarifs scolaires (valables au 1^{er} septembre 2017 au 31 août 2018)

Tarif par élève : 4,50 €
Tarif « éducation prioritaire » : 2,50 €
Supplément planétarium : 2,50 €

- 1 gratuité pour 5 entrées pour la maternelle
- 1 gratuité pour 12 entrées payantes pour l'élémentaire
- 1 gratuité pour 15 entrées payantes pour le secondaire

Le billet donne accès à toutes les expositions, aux ateliers scientifiques et aux exposés du Palais de la découverte (sur réservation et dans la limite des places disponibles).

Réservation groupes (à partir de 10 personnes)



groupe.palais@universcience.fr



01 56 43 20 25



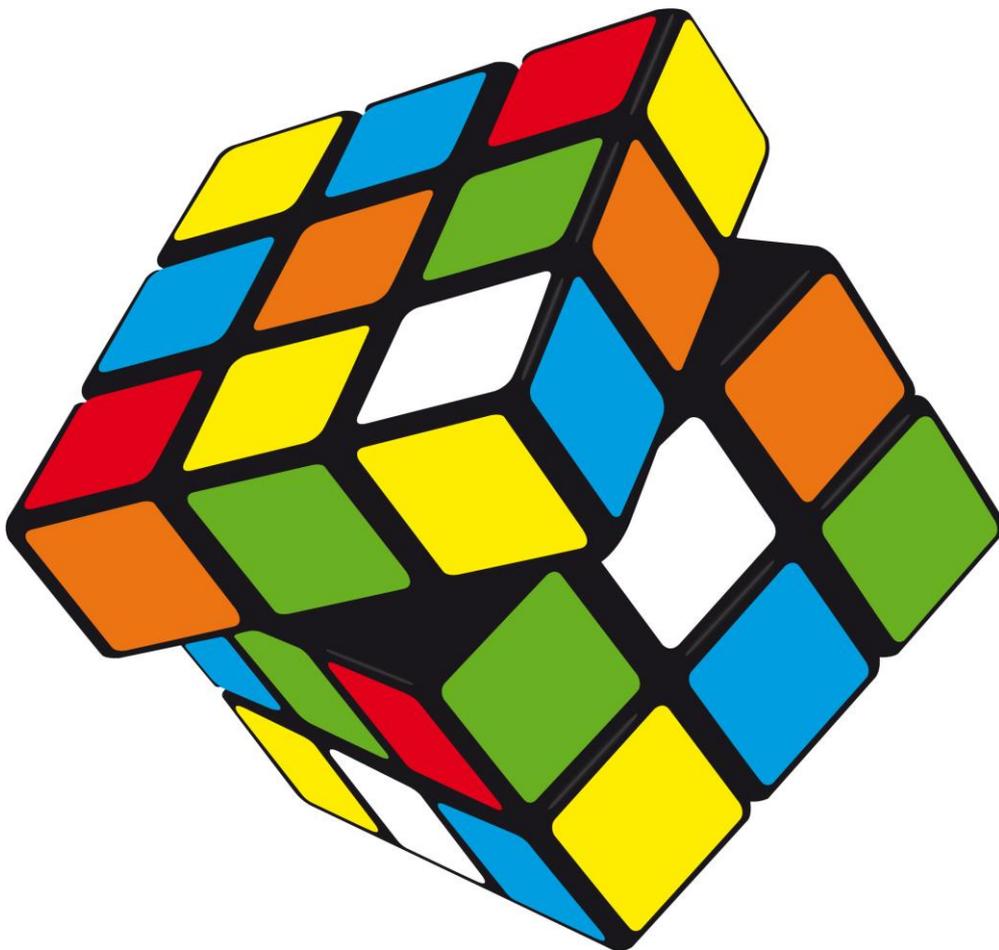
01 56 43 20 29



Palais de la découverte
Bureau des groupes
Avenue Franklin Roosevelt
75008 Paris

IV Annexe : description de deux algorithmes

Les lignes qui suivent décrivent deux algorithmes non présentés dans l'espace permanent *Informatique et sciences du numériques*. Il s'agit de l'**algorithme d'Euclide** et l'**algorithme de Nelder-Mead**. Le premier permet de déterminer le plus grand commun diviseur (PGCD) de deux entiers sans connaître leur factorisation ; le second, la résolution des problèmes d'optimisation de fonctions non linéaires. Son application à la détermination de la plus petite distance entre deux orbites planétaires dans un espace tridimensionnel est largement inspirée du chapitre 69 *The Simplex method and the least distance between two planetary orbits* de l'ouvrage *More Mathematical Astronomy Morsels* de Jean Meeus, paru en 2002 aux éditions Willmann-Bell.



Avez-vous joué avec le Rubik's cube ?
Il existe plusieurs méthodes systématiques ou algorithmes pour en venir à bout.

L'algorithme d'Euclide

L'algorithme d'Euclide, formulé géométriquement par le grand mathématicien il y a 2 300 ans, utilise le fait que le PGCD de deux entiers ne change pas lorsqu'on remplace l'entier le plus grand par la différence entre lui et l'entier le plus petit. Ainsi, le PGCD de 252 et de 105 est également le PGCD de 105 et de $252 - 105 = 147$. Puisque ce remplacement réduit le plus grand des deux nombres, la répétition du processus donne des paires de plus en plus petites, jusqu'à ce que les deux nombres soient égaux. Cette valeur commune est alors le PGCD, ici 21. L'algorithme d'Euclide est le suivant : Soit deux entiers a et b avec $a > b$. On commence par calculer le reste de la division de a par b , qu'on note r . On remplace ensuite a par b puis b par r et on réapplique le procédé depuis le début. On obtient ainsi une suite qui vaut 0 à un certain rang. Le PGCD cherché est le terme précédent de la suite. Exerçons-nous sur deux exemples et cherchons tout d'abord le PGCD de 782 et 221.

$$782 = 3 \times 221 + 119$$

$$221 = 1 \times 119 + 102$$

$$119 = 1 \times 102 + 17$$

$$102 = 6 \times 17 + 0. \text{ Le PGCD de 782 et 221 est 17.}$$

Cherchons ensuite le PGCD de 1 327 et 753.

$$1\ 327 = 1 \times 753 + 574$$

$$753 = 1 \times 574 + 179$$

$$574 = 3 \times 179 + 37$$

$$179 = 4 \times 37 + 31$$

$$37 = 1 \times 31 + 6$$

$$31 = 5 \times 6 + 1$$

$$6 = 6 \times 1 + 0. \text{ Le PGCD de 1 327 et 221 est 1. Ces deux entiers sont donc premiers entre eux.}$$



Un des plus anciennes versions connues des *Éléments*, découverte en 1896-97 sur le site d'Oxyrhynque en Égypte par Bernard Grenfell et Arthur Hunt. Ce fragment de papyrus daterait de la fin du 1^{er} siècle de notre ère.

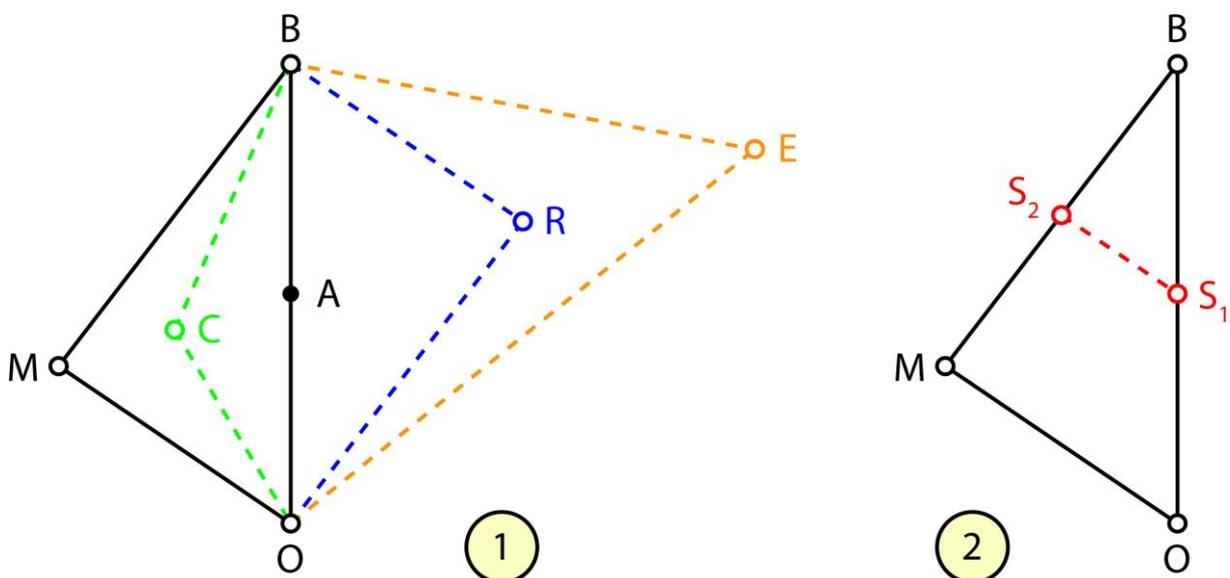
L'algorithme de Nelder-Mead

L'algorithme de Nelder-Mead a été décrit par John Nelder et Roger Mead dans l'article *A Simplex Method for Function Minimization* paru en 1965 dans la revue à comité de lecture *The Computer Journal*. Il fut popularisé par le numéro de mai 1984 du magazine informatique *Byte* grâce à l'article *Fitting Curves to Data. The Simplex algorithm is the answer*. En voici le principe.

Soit f une fonction de deux variables indépendantes x et y . **Pour quelles valeurs de x et y la fonction passe-t-elle par un minimum ?**

On commence par calculer la valeur de la fonction f pour trois points (x_1, y_1) , (x_2, y_2) et (x_3, y_3) dont on pense qu'ils pourraient être assez proches du couple recherché. Une estimation grossière est suffisante puisque l'algorithme de Nelder-Mead convergera, de toute façon, vers un minimum. **Les trois points (x_1, y_1) , (x_2, y_2) et (x_3, y_3) définissent un triangle dans le plan x - y .** Pour atteindre le minimum de la fonction f , l'algorithme déplace le triangle dans le « sens de la descente », l'accélérant, le ralentissant ou le déformant si besoin est.

La première étape de l'algorithme consiste à repérer, parmi les trois couples (x_i, y_i) , celui qui donne la valeur la plus faible à la fonction f (le point B de la figure ci-dessous) et celui qui lui donne la valeur la plus élevée (le point M). Comme on cherche le minimum de f , le couple donnant la valeur la plus élevée – donc le point M – doit être rejeté. On lui substitue un nouveau couple qui correspond à l'application de l'une des quatre transformations suivantes au sommet M : symétrie centrale, extension, contraction ou rétrécissement.



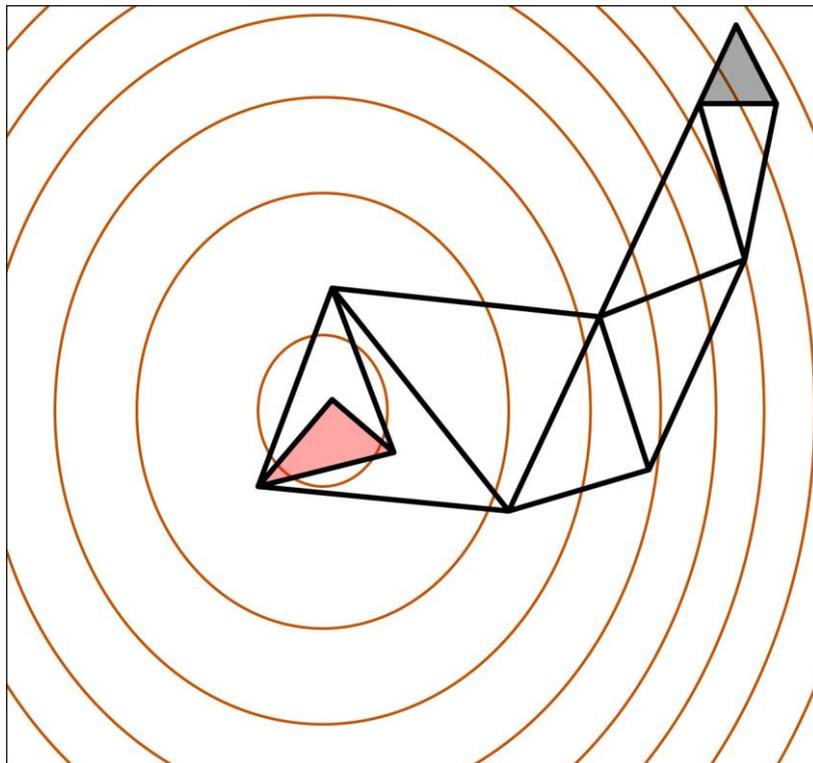
BMO est notre triangle de départ. B est le meilleur sommet. Parmi les trois couples (x_i, y_i) , il représente celui qui donne la valeur la plus faible à la fonction f . M est le plus mauvais sommet et c'est lui qui est rejeté. Le schéma illustre les quatre transformations de l'algorithme de Nelder-Mead : ① R est le symétrique de M par rapport à A (milieu du segment $[OB]$), E le sommet étendu et C le sommet contracté ② S_1 et S_2 , les sommets rétrécis.

On construit le point R , symétrique du point M par rapport à A , le milieu du segment $[OB]$. On teste alors le couple (x_R, y_R) et s'il donne une réponse qui n'est ni meilleure que le couple (x_B, y_B) , ni plus mauvaise que le couple (x_M, y_M) , il est conservé.

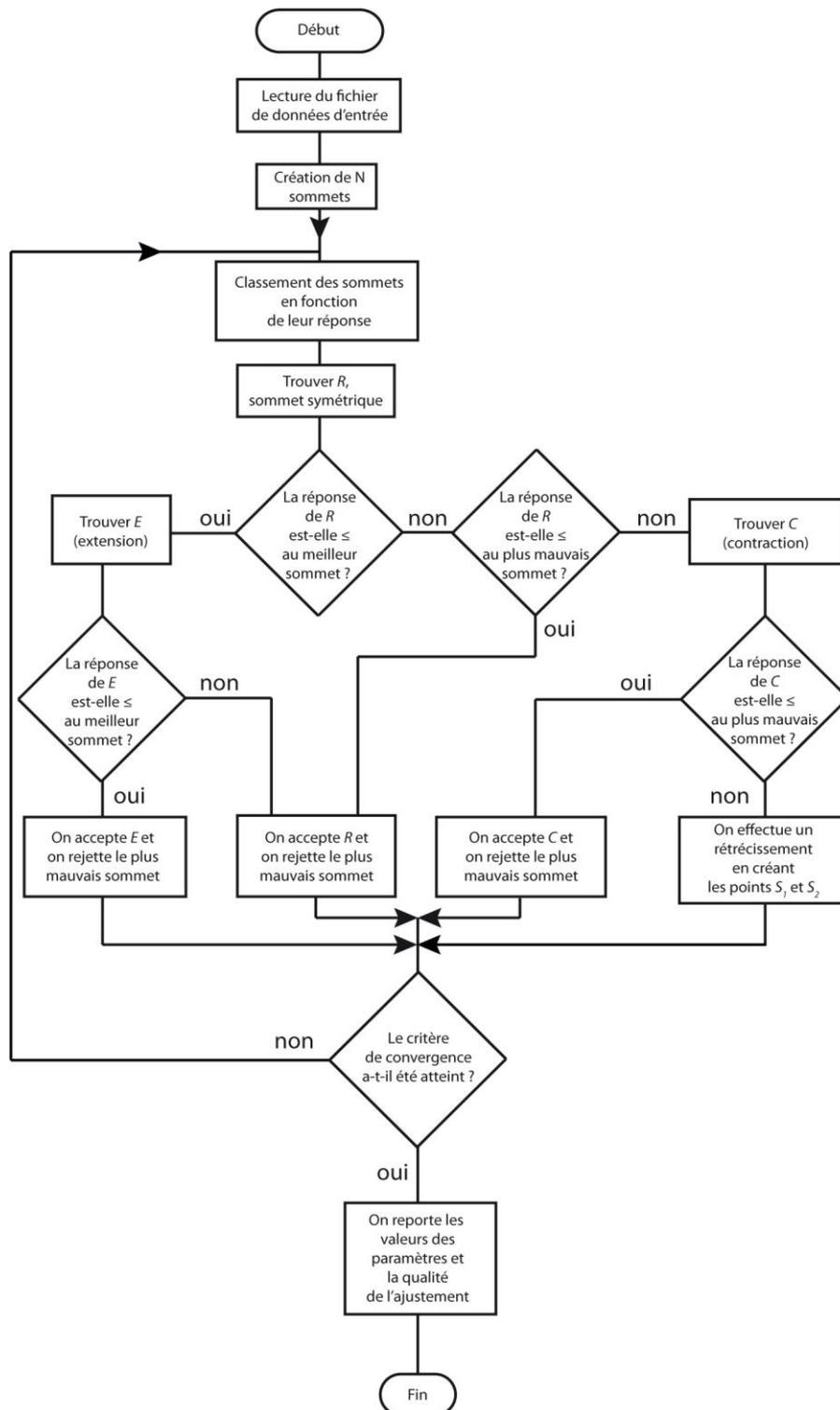
S'il donne une réponse meilleure (donc une valeur plus faible) que le couple (x_B, y_B) , on considère que la direction trouvée est la bonne et l'on construit le point E , symétrique de A par rapport à R . On parle d'extension. Le point E est conservé s'il donne une réponse meilleure que le point rejeté M ; sinon, on garde R .

Si le point R donne une réponse plus mauvaise que le point M , on teste une contraction en construisant le point C , milieu du segment $[MA]$. Là aussi, C est conservé s'il donne une réponse meilleure que le point rejeté M ; sinon, on effectue un rétrécissement en substituant aux points O et M les milieux respectifs S_1 et S_2 des segments $[OB]$ et $[MB]$.

La figure suivante montre l'exemple d'un triangle se déplaçant sur les lignes de niveau d'une fonction. Le triangle initial est grisé. Après huit transformations, on le retrouve déformé sous la forme du triangle rose près du minimum.



Un inconvénient de la méthode classique de Nelder-Mead est qu'elle peut aboutir à un minimum local. Pour des fonctions ayant plusieurs minima, le point vers lequel l'algorithme converge dépend des points initiaux et de la taille initiale du triangle.

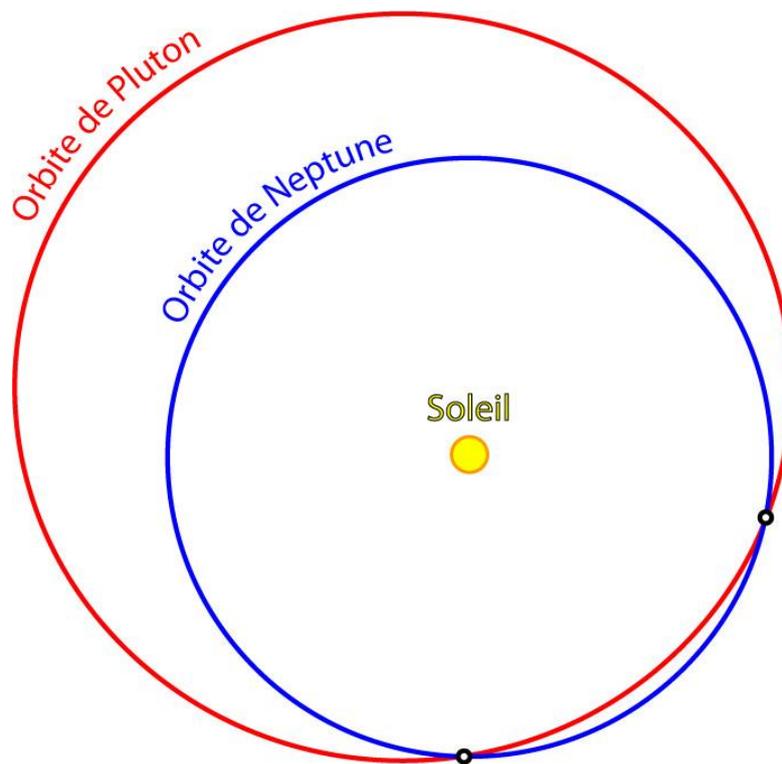


Organigramme de programmation de l'algorithme de Nelder-Mead.

Il y a différentes façons d'exprimer le critère de convergence permettant de mettre un terme à l'algorithme.

Dès le départ, on peut se donner un nombre maximal d'itérations, i . On peut également se donner une aire prédéfinie j ; l'algorithme s'arrête une fois que le triangle possède une surface qui lui est inférieure. On peut enfin se dire que la convergence est satisfaisante lorsque k évaluations successives de la fonction diffèrent de moins d'un seuil prédéfini l .

Dans son ouvrage *More Mathematical Astronomy Morsels*, Jean Meeus applique l'algorithme de Nelder-Mead au calcul de la plus petite distance entre l'orbite de la Terre et celle de l'astéroïde (3838) Épona, petit corps de quelques kilomètres à l'orbite très excentrique découvert en 1986. Nous l'appliquerons au calcul de la plus petite distance entre les orbites de la planète Neptune et celle de la planète naine Pluton. **Attention, nous parlons bien de la distance minimale entre les deux orbites et pas de la distance minimale entre les deux corps !**



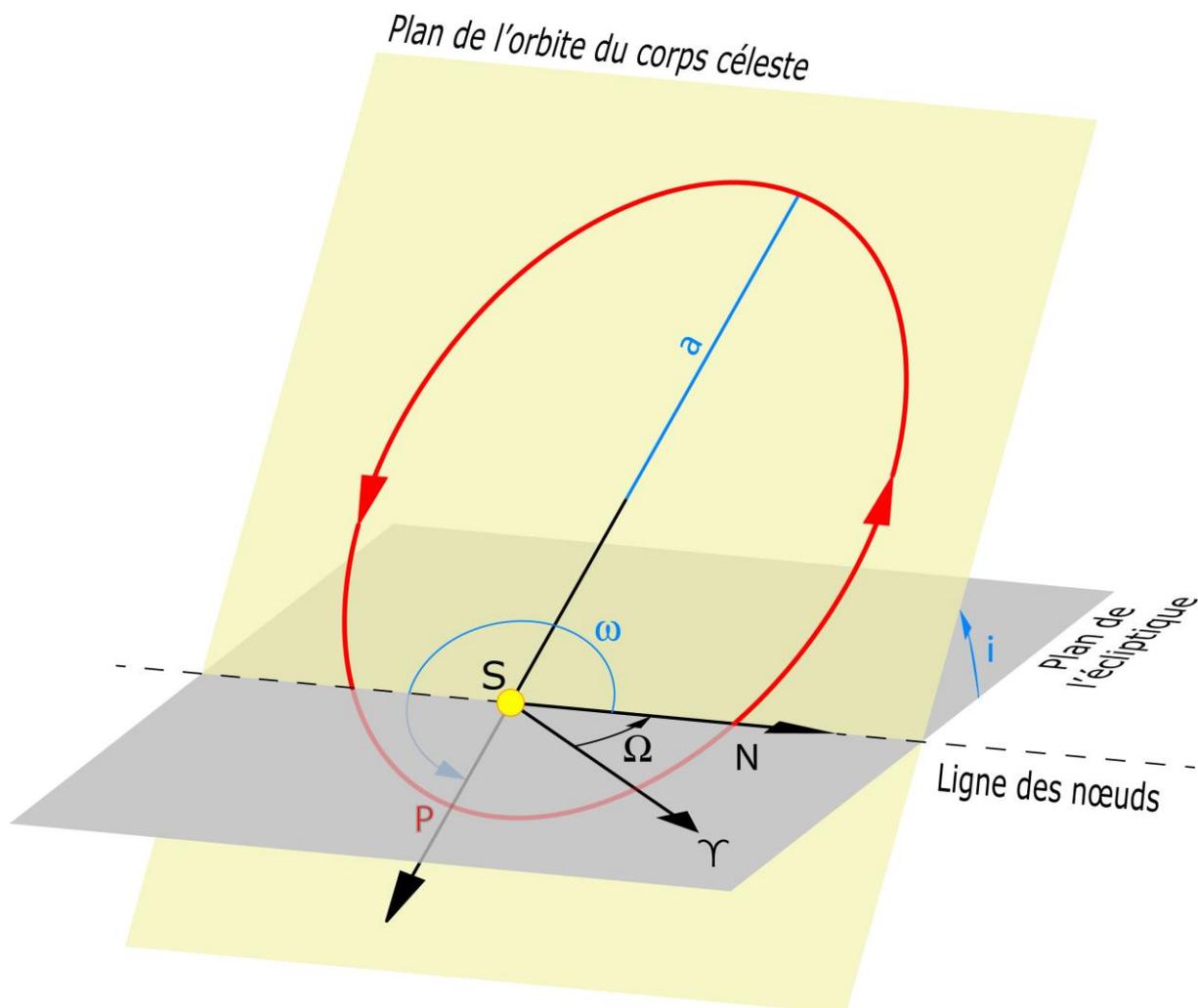
Projection des orbites de Neptune et de Pluton autour du Soleil sur le plan de l'écliptique.

Comme le montre le schéma ci-dessus, les orbites de Neptune et de Pluton autour du Soleil se croisent en deux points lorsqu'on les projette sur le plan de l'écliptique – le plan dans lequel la Terre tourne autour du Soleil. La collision entre les deux corps célestes est-elle seulement possible ?

Comment s'attaquer à ce problème ?

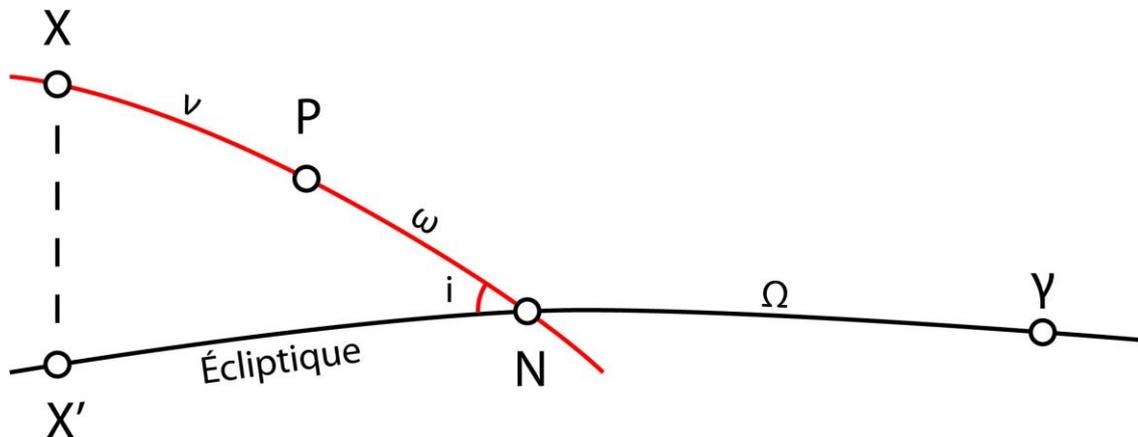
Il convient tout d'abord de caractériser correctement les orbites elliptiques des deux corps autour du Soleil, c'est-à-dire de préciser leur forme et leur orientation dans l'espace. On se donne donc un plan de référence, le plan de l'écliptique déjà présenté quelques lignes plus haut. On cherche ensuite dans la littérature spécialisée une série d'éléments appelés *éléments orbitaux*, qui permettent de préciser, à une date donnée, l'orbite elliptique des corps considérés par rapport à ce plan de référence. Ils sont au nombre de cinq :

- le **demi-grand axe** a , qui n'est autre que la moitié du plus long diamètre de l'ellipse. C'est une mesure de la taille absolue de l'ellipse et dans le système solaire, on l'exprime volontiers en *unité astronomique* (ua), une unité de distance s'élevant à 149 597 870,7 kilomètres ;
- l'**excentricité** e , qui est une mesure de l'aplatissement de l'ellipse ($0 \leq e < 1$) ;
- l'**inclinaison** i du plan considéré sur le plan de l'écliptique. Si i est compris entre 0° et 90° , le mouvement est dit *direct*. S'il est compris entre 90° et 180° , il est dit *rétrograde* ;
- la **longitude du nœud ascendant** Ω , qui est l'angle entre la direction du point vernal γ (position occupée par le Soleil à l'équinoxe de mars) et la ligne des nœuds (droite d'intersection du plan orbital avec le plan de l'écliptique) ;
- l'**argument du périhélie** ω , c'est-à-dire l'angle entre le nœud ascendant et le périhélie, mesuré dans le plan orbital et dans la direction du mouvement.

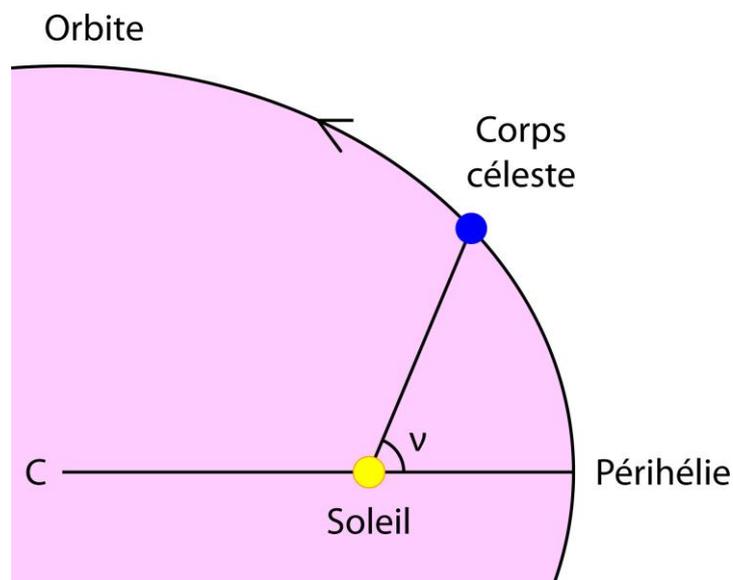


S est le Soleil. Le plan de l'écliptique, notre plan de référence, est en gris. Il porte la direction du point vernal γ . Le plan orbital du corps céleste est en jaune. Il est incliné d'un angle i par rapport au plan de l'écliptique. Les deux plans se coupent selon une droite, la ligne des nœuds. Dans le plan orbital est représentée, en rouge, l'ellipse décrite par le corps. On note a son demi-grand axe, P le périhélie et ω l'argument du périhélie.

Dans la figure suivante, l'arc $\gamma NX'$ représente une partie de l'écliptique vu depuis le Soleil. Il s'agit de l'intersection du plan de l'écliptique avec la sphère céleste. NPX est une partie de l'orbite du corps céleste, qui est l'intersection de son plan orbital avec la sphère céleste. γ est le point vernal (l'origine des longitudes), N le nœud ascendant de l'orbite, P le périhélie et X la position du corps céleste à un instant donné. L'arc XX' est une partie du grand cercle perpendiculaire à l'écliptique et passant par X .



L'arc $\gamma NX'$ est donc la *longitude écliptique* λ du corps céleste ; toutefois, ce n'est pas d'elle dont nous avons besoin mais de la *longitude vraie* l , qui est la somme des arcs $\gamma N = \Omega$ et $NPX = \omega + \nu$ où ν est l'anomalie vraie, à savoir l'angle entre la direction du périhélie et la position de l'objet sur son orbite, mesuré au foyer de l'ellipse (voir schéma ci-dessous). Notez que ces arcs sont mesurés dans deux plans différents.



Le Soleil occupe l'un des deux foyers de l'ellipse décrite par le corps céleste. L'anomalie vraie ν est l'angle entre la direction du périhélie et celle du corps, mesuré depuis le Soleil.

Pour toutes les valeurs de la longitude vraie l , nous pouvons calculer les deux angles :

$$u = l - \Omega = \text{arc } NX \quad (\text{argument de latitude})$$

$$v = l - \Omega - \omega = \text{arc } PX \quad (\text{anomalie vraie})$$

Nous avons désormais tous les outils en main pour mener à bien la mission que nous nous sommes assignée. En effet, on peut maintenant déterminer :

- le rayon vecteur r du corps céleste (sa distance au Soleil) grâce à la formule

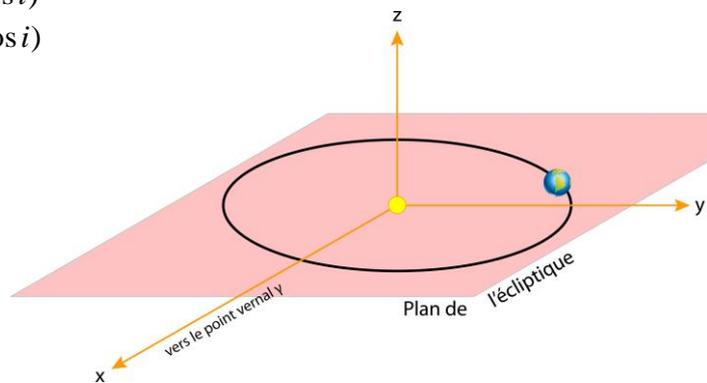
$$r = \frac{a(1-e^2)}{1+e \cos v} ;$$

- les coordonnées rectangulaires de ce corps dans un référentiel héliocentrique

$$x = r(\cos \Omega \cos u - \sin \Omega \sin u \cos i)$$

$$y = r(\sin \Omega \cos u + \cos \Omega \sin u \cos i)$$

$$z = r \sin i \sin u.$$



De la même manière, si a' , e' , i' , Ω' et ω' sont les éléments orbitaux de l'orbite du second corps, on peut calculer u' , v' , r' et les coordonnées rectangulaires x' , y' et z' pour toute valeur de la longitude vraie l' sur cette orbite. Il convient, bien évidemment, d'exprimer les éléments des deux orbites au même instant.

Trouver la distance minimale entre les deux orbites revient à déterminer le couple de longitudes vraies l et l' pour lequel le carré de la distance entre les deux orbites $d^2 = (x - x')^2 + (y - y')^2 + (z - z')^2$ soit minimal. Voici donc la sympathique fonction sur laquelle nous allons travailler :

$$d^2(l, l') = \left(\frac{a(1-e^2)}{1+e \cos(l-\Omega-\omega)} (\cos \Omega \cos(l-\Omega) - \sin \Omega \sin(l-\Omega) \cos i) - \frac{a'(1-e'^2)}{1+e' \cos(l'-\Omega'-\omega')} (\cos \Omega' \cos(l'-\Omega') - \sin \Omega' \sin(l'-\Omega') \cos i') \right)^2$$

$$+ \left(\frac{a(1-e^2)}{1+e \cos(l-\Omega-\omega)} (\sin \Omega \cos(l-\Omega) + \cos \Omega \sin(l-\Omega) \cos i) - \frac{a'(1-e'^2)}{1+e' \cos(l'-\Omega'-\omega')} (\sin \Omega' \cos(l'-\Omega') + \cos \Omega' \sin(l'-\Omega') \cos i') \right)^2$$

$$+ \left(\frac{a(1-e^2)}{1+e \cos(l-\Omega-\omega)} (\sin i \sin(l-\Omega)) - \frac{a'(1-e'^2)}{1+e' \cos(l'-\Omega'-\omega')} (\sin i' \sin(l'-\Omega')) \right)^2.$$

Puisqu'il faut bien commencer quelque part, nous considérerons d'abord des longitudes vraies égales ($l = l'$) sur les deux orbites.

Les éléments orbitaux de Neptune et de Pluton le 1^{er} janvier 2000 sont donnés par E. M. Standish, X. X. Newhall, J. G. Williams et D. K. Yeomans dans le chapitre 5 *Orbital Ephemerides of the Sun, Moon and Planets* de l'ouvrage *Explanatory Supplements to the Astronomical Almanac*, édité par P. K. Seidelmann de l'U.S. Naval Observatory (University Science Books, 1992).

Neptune

Demi-grand axe a	30,068 963 48 ua
Excentricité e	0,008 585 87
Inclinaison i	1,769 17°
Longitude du nœud ascendant Ω	131,721 69°
Argument du périhélie ω	273,249 66°



Pluton

Demi-grand axe a'	39,481 686 77 ua
Excentricité e'	0,248 807 66
Inclinaison i'	17,141 75°
Longitude du nœud ascendant Ω'	110,303 47°
Argument du périhélie ω'	113,763 29°



Le tableau ci-dessous présente les valeurs calculées de d (en unité astronomique) correspondant à des valeurs de $l = l'$ croissant par sauts de 10°.

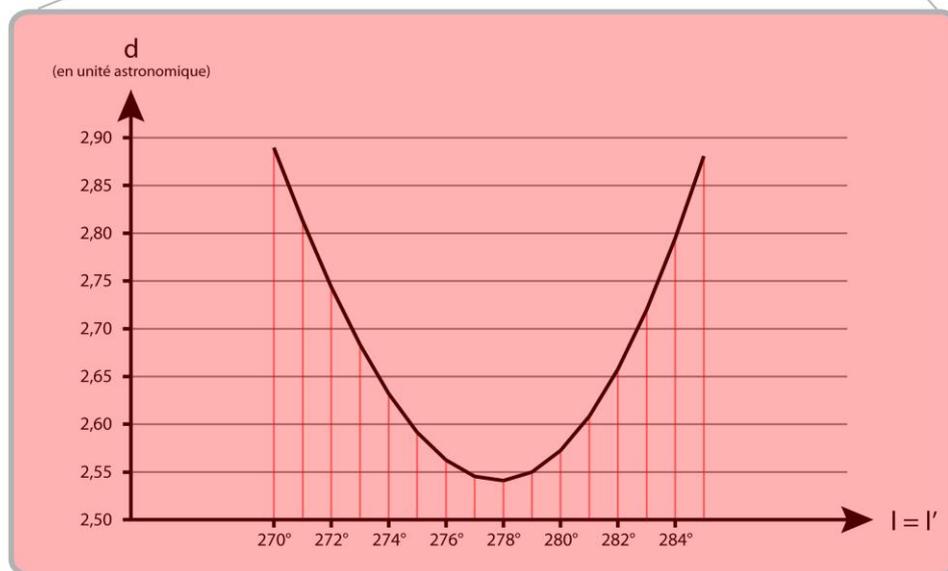
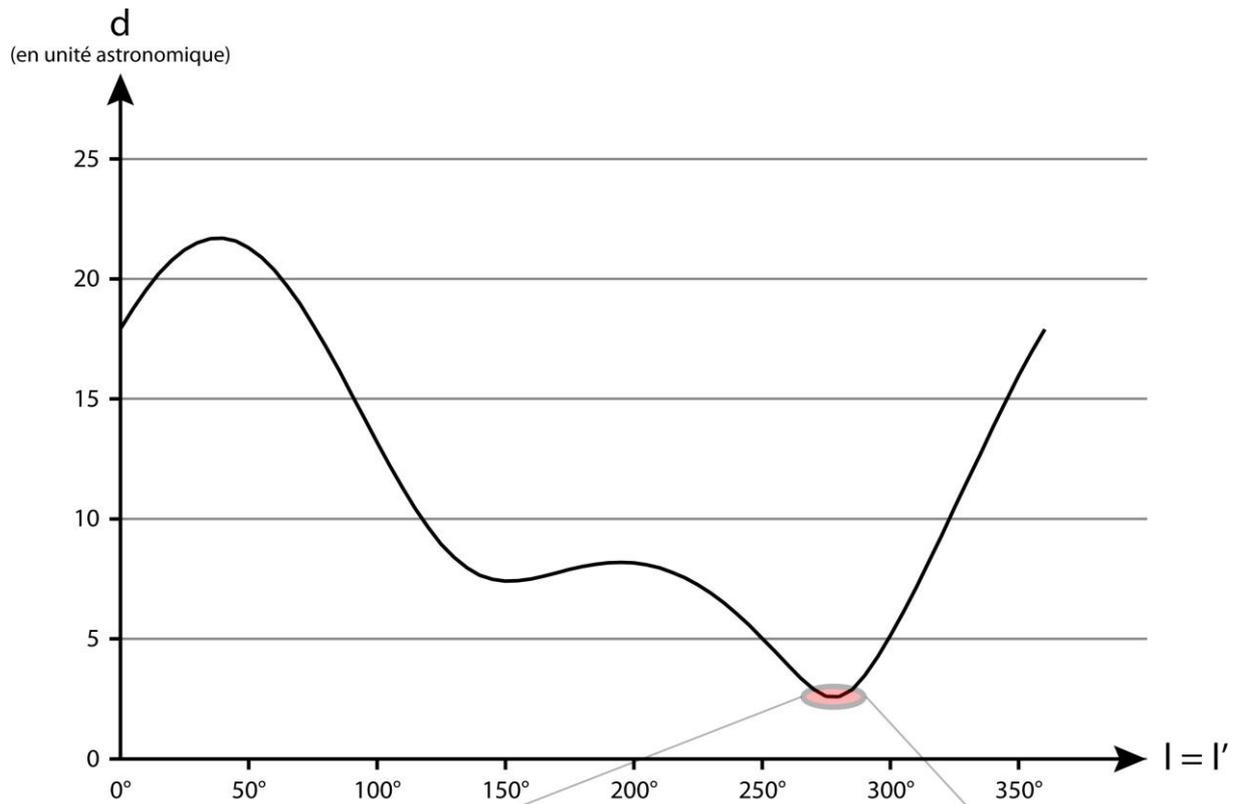
$l = l'$	0°	10°	20°	30°	40°	50°	60°	70°	80°
d (ua)	17,90	19,53	20,77	21,50	21,68	21,29	20,35	18,94	17,18

$l = l'$	90°	100°	110°	120°	130°	140°	150°	160°	170°
d (ua)	15,21	13,18	11,27	9,619	8,384	7,651	7,397	7,488	7,741

$l = l'$	180°	190°	200°	210°	220°	230°	240°	250°	260°
d (ua)	8,001	8,161	8,156	7,951	7,530	6,891	6,046	5,024	3,901

$l = l'$	270°	280°	290°	300°	310°	320°	330°	340°	350°
d (ua)	2,890	2,572	3,464	5,130	7,143	9,326	11,58	13,82	15,96

Enfin, les graphiques en page suivante montrent de façon plus détaillée l'évolution de d en fonction de $l = l'$.



D'après le graphique du haut, le minimum de distance est atteint pour $l = l'$ compris entre 250° et 300° . Une étude plus fine (graphique du bas) place ce minimum vers 278° .

L'application de l'algorithme de Nelder-Mead nécessite le calcul de trois valeurs de la fonction $d = f(l, l')$ pour définir le triangle dans le plan $l - l'$. Soit donc les trois points A, B et C, choisis arbitrairement tels que leurs coordonnées l et l' soient proches de 278° . Les trois valeurs de l ne doivent pas être les mêmes, tout comme les trois valeurs de l' . Pour chacun des points, nous calculons la distance d correspondante entre les deux orbites.

$$A \quad l = 278^\circ \quad l' = 278^\circ \quad \rightarrow \quad d_A = 2,540\,969\,652$$

$$B \quad l = 277^\circ \quad l' = 277^\circ \quad \rightarrow \quad d_B = 2,545\,238\,895$$

$$C \quad l = 278^\circ \quad l' = 277^\circ \quad \rightarrow \quad d_C = 2,549\,788\,763$$

Le triangle ABC est caractérisé par une amplitude en l de 1° , en l' de 1° et en d de 0,008 819 112 unité astronomique, soit un peu plus de 1,3 million de kilomètres.

Dans le triangle ABC , A est le meilleur sommet et C le plus mauvais. On construit d'abord R, symétrique de C par rapport à O, le milieu du segment $[AB]$. Ses coordonnées sont $l_R = 277^\circ$ et $l'_R = 278^\circ$.

$$R \quad l = 277^\circ \quad l' = 278^\circ \quad \rightarrow \quad d_R = 2,646\,979\,002 > d_C$$

Visiblement, R donne une réponse plus mauvaise que le point C. On teste donc une contraction en construisant le point D, milieu du segment $[CO]$. Ses coordonnées sont $l_D = 277,75^\circ$ et $l'_D = 277,25^\circ$.

$$D \quad l = 277,75^\circ \quad l' = 277,25^\circ \quad \rightarrow \quad d_D = 2,531\,149\,559 < d_C$$

Le point contracté D est donc accepté.

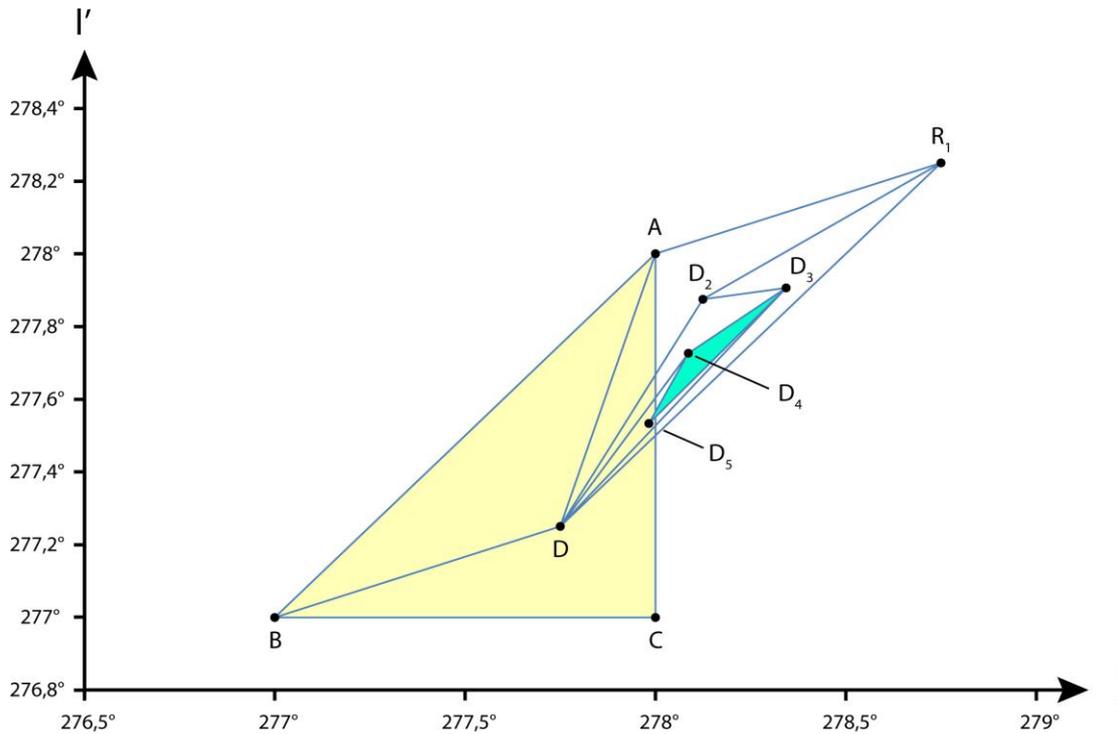
Nous nous retrouvons avec un nouveau triangle, ABD :

$$A \quad l = 278^\circ \quad l' = 278^\circ \quad \rightarrow \quad d_A = 2,540\,969\,652$$

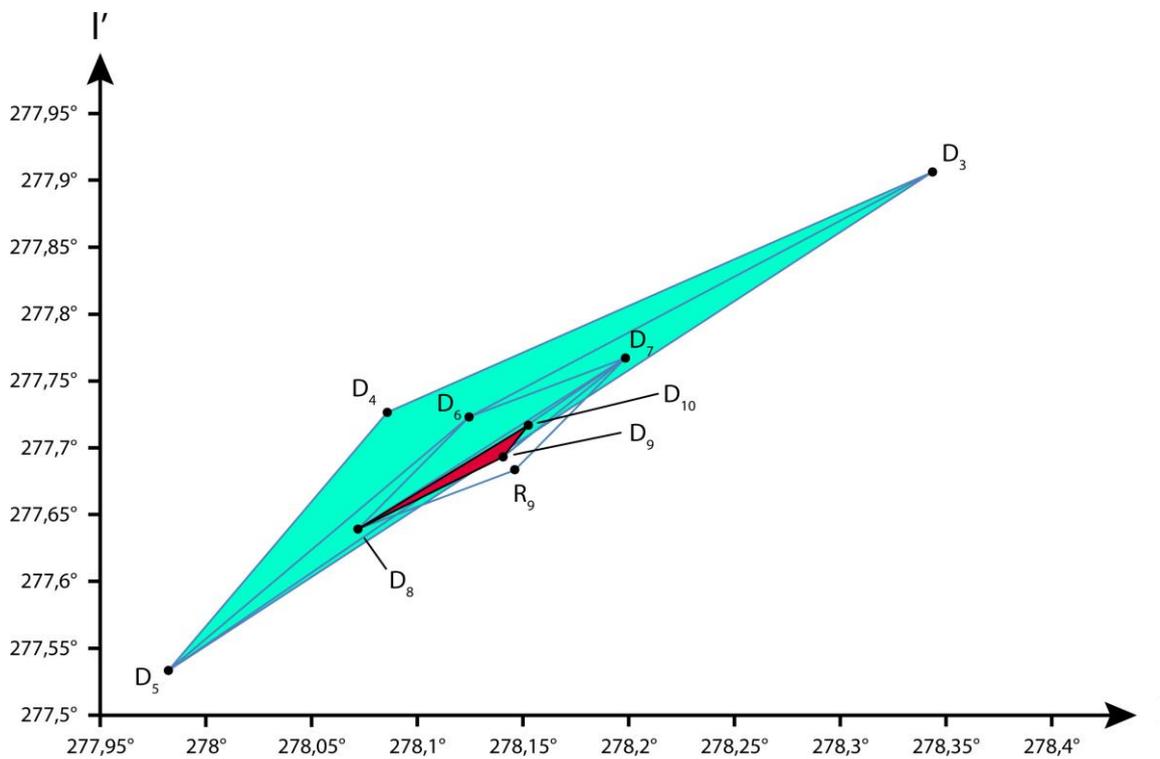
$$B \quad l = 277^\circ \quad l' = 277^\circ \quad \rightarrow \quad d_B = 2,545\,238\,895$$

$$D \quad l = 277,75^\circ \quad l' = 277,25^\circ \quad \rightarrow \quad d_D = 2,531\,149\,559$$

Les calculs, dont nous vous faisons grâce, ont ensuite été menés jusqu'à la 12^e étape. Les deux graphiques suivants présentent les triangles associés.



Le triangle de départ ABC est en jaune. Six transformations plus tard, on obtient le triangle bleu $D_3D_4D_5$.



Un zoom a été effectué sur le triangle bleu $D_3D_4D_5$. Après six transformations, on obtient le triangle rouge $D_8D_9D_{10}$.

Le triangle final de cette étude est $D_8D_9D_{10}$.

D_8	$l = 278,071\ 990\ 967^\circ$	$l' = 277,639\ 190\ 674^\circ$	$d_{D8} = 2,529\ 768\ 917\ \text{ua}$
D_9	$l = 278,140\ 693\ 665^\circ$	$l' = 277,693\ 229\ 675^\circ$	$d_{D9} = 2,529\ 757\ 431\ \text{ua}$
D_{10}	$l = 278,152\ 475\ 357^\circ$	$l' = 277,716\ 711\ 044^\circ$	$d_{D10} = 2,529\ 754\ 283\ \text{ua}$

Il est caractérisé par les amplitudes :

$$\Delta l = 0,080\ 484\ 390^\circ \approx 4' 50'' \quad \Delta l' = 0,077\ 520\ 370^\circ \approx 4' 40''$$

$$\Delta d = 0,000\ 014\ 633\ \text{ua} \approx 2189\ \text{km}.$$

Conclusion : la distance minimale entre les orbites de Neptune et de Pluton est, selon toute vraisemblance, un peu inférieure à 2,53 unités astronomiques soit environ 380 millions de kilomètres. Sur l'orbite de Neptune, elle correspond à une longitude vraie proche de $278,1^\circ$ et sur celle de Pluton, proche de $277,7^\circ$. Pas de collision possible !

En fait, pour des raisons de résonance entre les deux corps, Neptune et Pluton ne s'approchent jamais l'un de l'autre à moins de 17 unités astronomiques.

Nous venons de passer en revue les algorithmes d'Euclide et de Nelder-Mead. Il en existe, bien sûr, d'autres en très grand nombre.

