

# Une brève histoire de l'informatique

Concepts, techniques, applications

Sacha Krakowiak

Université Grenoble Alpes & Aconit

## Pourquoi s'intéresser à l'histoire de l'informatique ?

### ❖ Une aide pour comprendre le présent

comment on est arrivé à la situation actuelle  
quels sont les défis

Les avancées, mais aussi les échecs...

### ❖ Une vision des idées, non seulement des faits

le contexte, les liens de causalité

### ❖ Un instrument pédagogique

comprendre les concepts et les techniques  
à travers leur émergence et leur évolution

"We would know what they thought when they did it"

R. W. Hamming, 1976

### ❖ Un intérêt propre

les «chefs d'œuvre» du passé  
les acteurs de l'informatique

Des éléments de stabilité dans une discipline qui évolue vite

# C'est quoi, l'informatique ?

## ❖ Quatre facettes

Une science

Science de l'artificiel ...

... mais pas seulement

Une technique et une industrie

Matériel, logiciel, services

Des applications

Dont le champ est croissant

Un impact sociétal

## ❖ Quatre concepts

## ❖ Une méthode

La modélisation et l'abstraction

Un va-et-vient théorie-pratique

Des outils issus de l'informatique

### Information

Un réducteur d'incertitude

Une représentation codée

La base de la communication

### Algorithme

La notion clé !

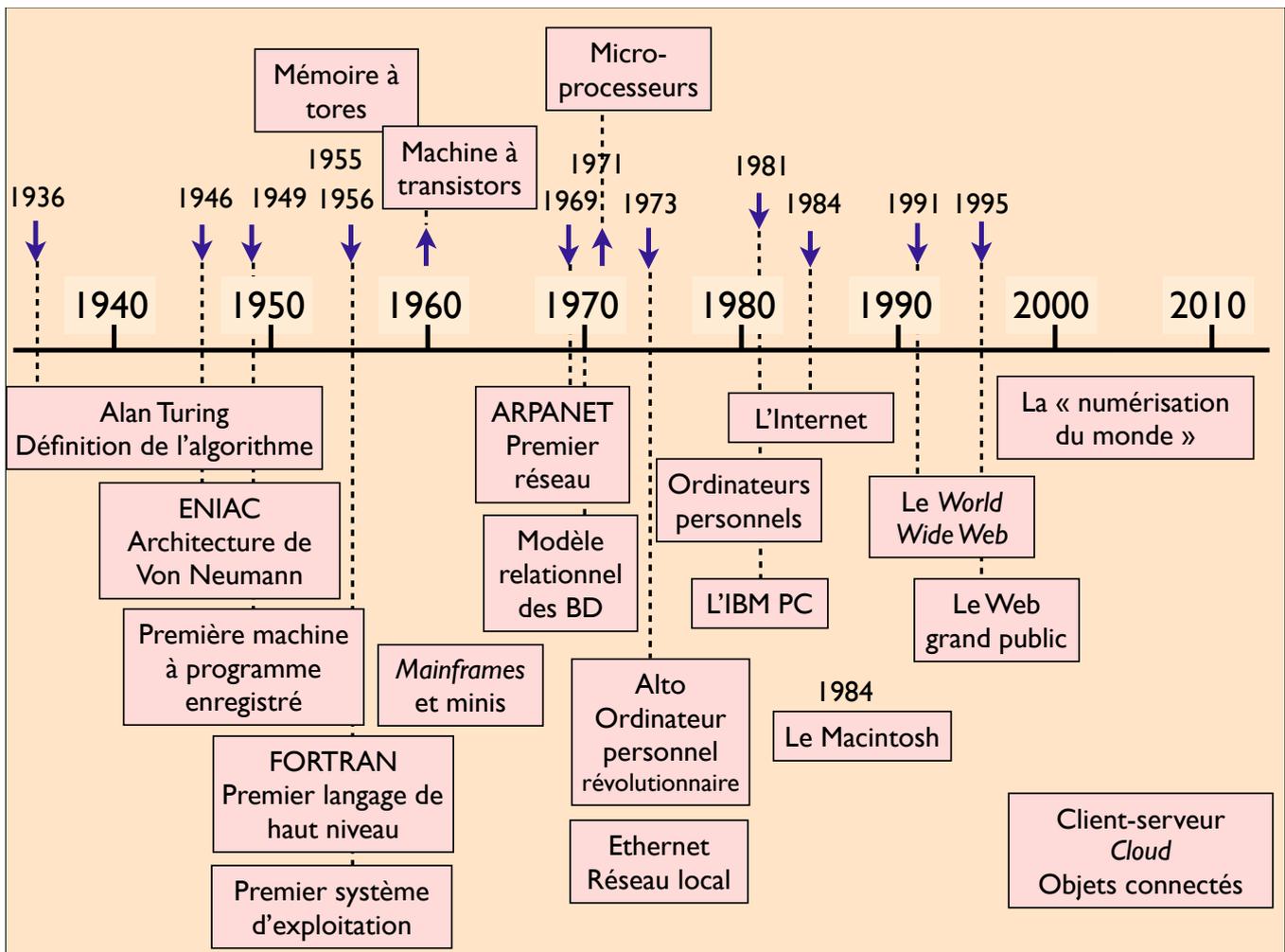
d'Euclide à Turing

### Machine

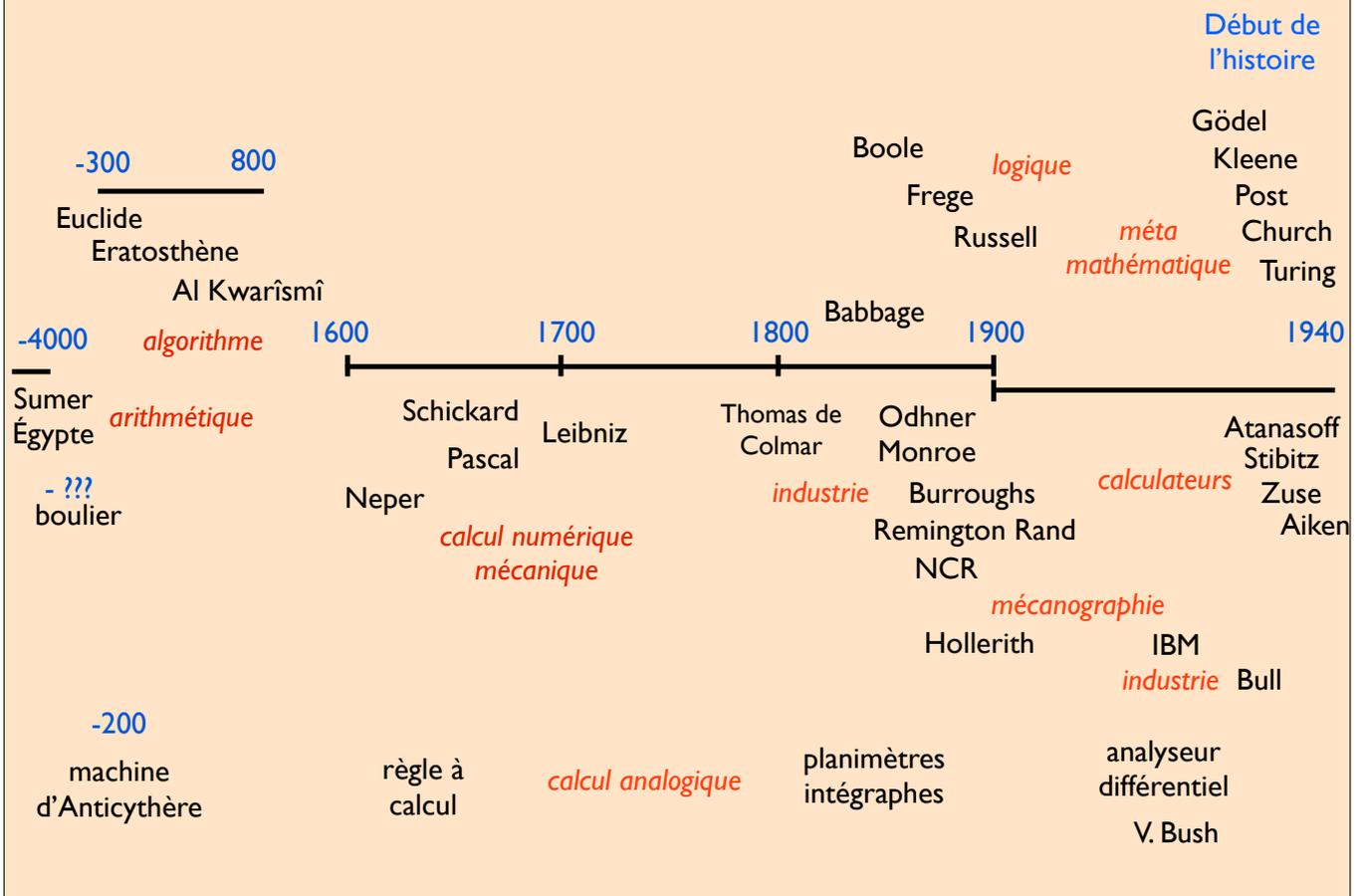
Réelle ou virtuelle

### Langage

Exprime un algorithme  
pour une machine



# La préhistoire de l'informatique



## Quelques précurseurs

### ❖ Euclide (-300)

l'un des premiers algorithmes connus (-300)



### ❖ Al Kwarîsmî

traité des algorithmes (800)



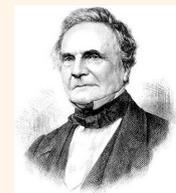
### ❖ Schickard, Pascal, Leibniz

les premiers calculateurs mécaniques (1620-1670)



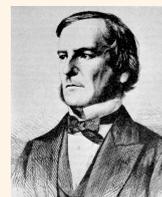
### ❖ Babbage

la machine analytique, précurseur de l'ordinateur (1840)



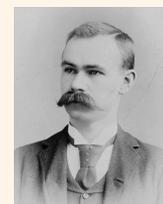
### ❖ Boole

logique et algèbre booléennes (1850)



### ❖ Hollerith

la mécanographie (1890-1960)



## Les débuts de l'informatique

### ❖ Le début du XX-ème siècle : une époque de remise en question dans les sciences

en physique : la relativité, les quanta, ...

en mathématiques : la «crise des fondements»

en logique : Gödel



Kurt Gödel  
(1906-1977)

### ❖ Vers la fin des années 1930, deux courants qui s'ignorent mutuellement

la «calculabilité» et la formalisation des algorithmes (Church, Gödel, Kleene, Turing, ...)

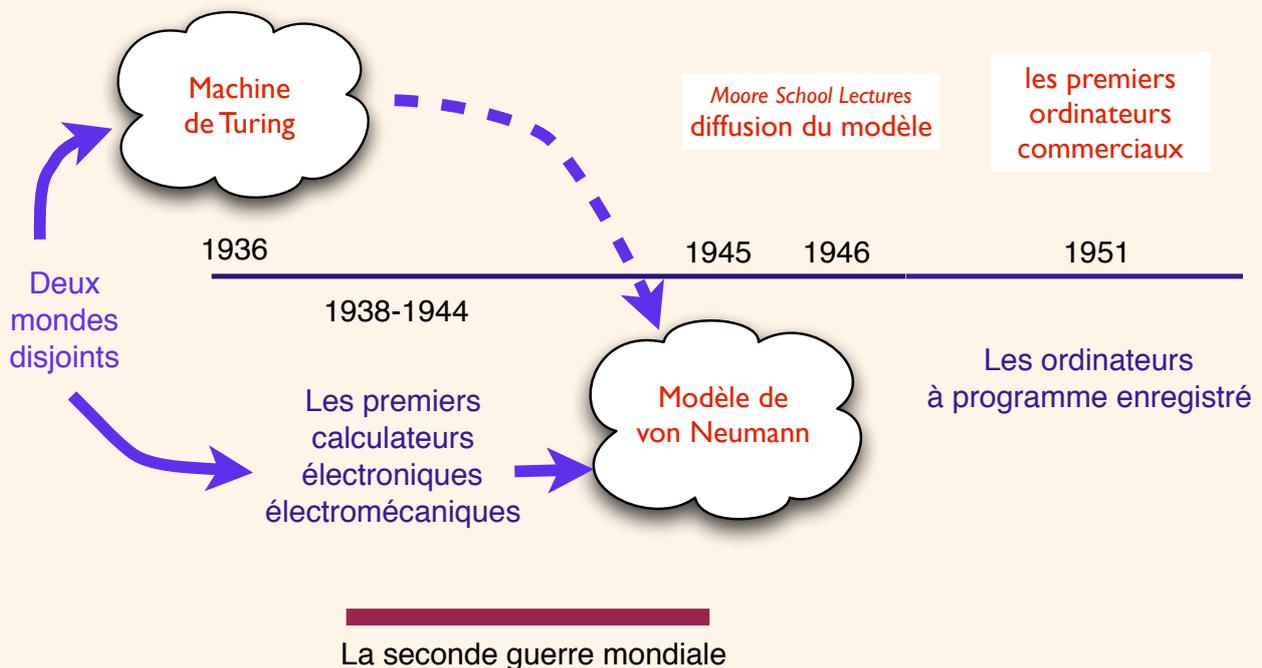
les prototypes des «calculateurs électroniques» (Atanasoff, Zuse, Eckert & Mauchly, ...)

### ❖ Ces courants se rejoindront en 1945

von Neumann

## De Turing à von Neumann

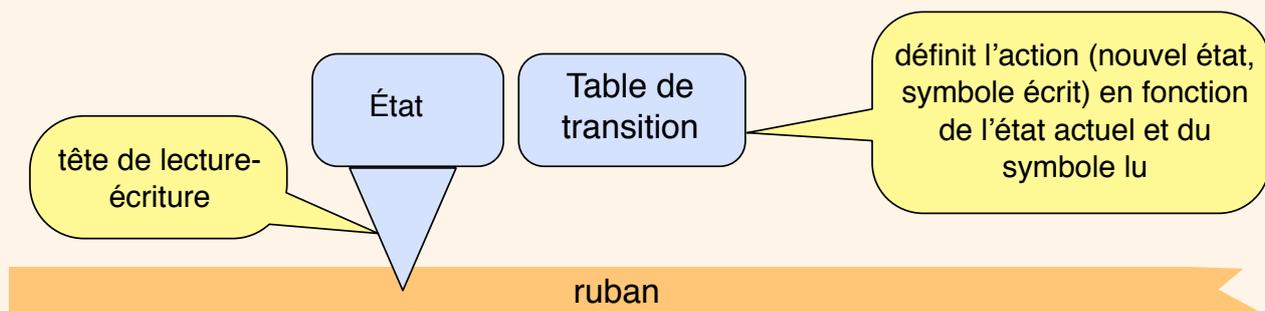
### ❖ 1936-1946 : la décennie décisive



## Retour sur les algorithmes

- ❖ Que peut-on «effectivement» calculer ?
- ❖ De nombreuses propositions dans les années 1930
  - les fonctions récursives (Kleene)
  - le lambda-calcul (Church)
  - la machine de Turing
  - ...
- ❖ Tous ces schémas, définis indépendamment, sont équivalents !
  - D'où la **thèse de Church-Turing** (indémontrable, mais jusqu'ici non contredite) :
  - Un algorithme, c'est ce qui est réalisable par une machine de Turing (ou tout schéma équivalent)

## La machine de Turing (1936)



Alan Turing  
(1912-1954)



National Portrait Gallery

C'est une machine *abstraite* (elle serait monstrueusement inefficace !)

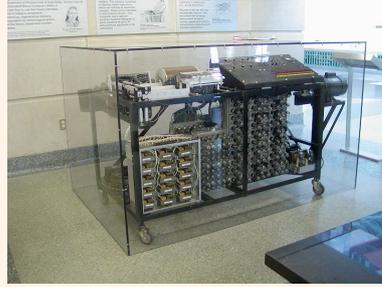
Il existe une machine *universelle* (capable de simuler n'importe quelle machine)

Il n'y a pas d'algorithme général pour dire si une machine va s'arrêter ou tourner indéfiniment

# Les premiers calculateurs

Atanasoff-Berry  
Computer (ABC)  
Iowa, 1939-42

Spécialisée (systèmes  
linéaires). Électronique  
(tubes), arith. binaire.  
non Turing-complète

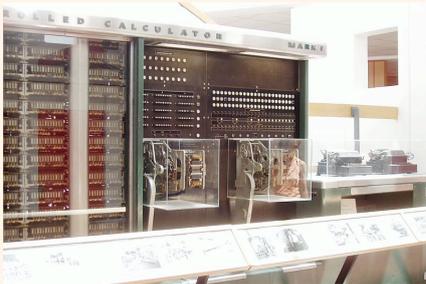


Zuse Z3 (1941)



Électromécanique (relais), binaire,  
programme sur bande perforée,  
Turing-complète

Harvard Mark 1 - IBM ASCC  
(H. Aiken, 1944)



Électromécanique, décimale, programme sur  
bande perforée, non Turing-complète



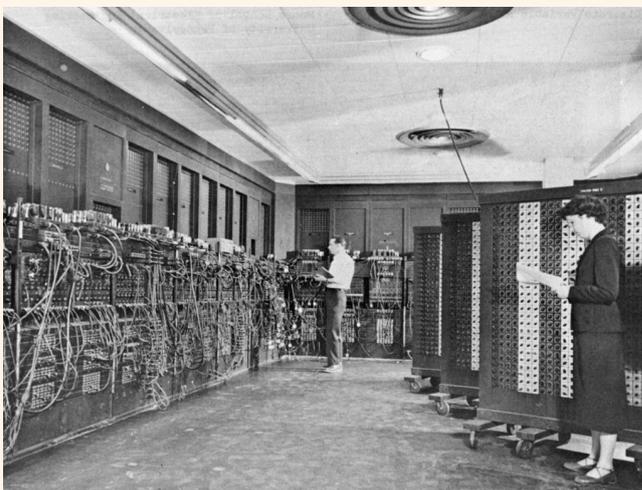
Colossus  
(T. Flowers,  
Bletchley Park,  
1943-44)

Électronique, binaire, spécialisée dans décryptage  
(cassage code Lorenz), non Turing-complète

# L'ENIAC

❖ Le premier ordinateur entièrement électronique, Univ.  
de Pennsylvanie (Eckert-Mauchly, 1946)

Arithmétique décimale, programmation par câblage  
18 000 tubes, 27 tonnes, 150 kw. Turing-complète



J. Presper Eckert  
1919-1995



John W. Mauchly  
1907-1980

# L'ENIAC

## ❖ Caractéristiques

27 tonnes, salle de 18m X 9m  
42 modules disposés en U  
18 000 tubes, 1 500 relais, 7 500 diodes  
70 000 résistances, 10 000 capacités  
150 KW, refroidi par air

## ❖ Performances

Par seconde :

5 000 additions, 385 multiplications,  
40 divisions, 3 racines carrées

Nombres décimaux, 10 chiffres + signe

Possibilité de calcul en double précision

Possibilité de parallélisme (géré à la main)

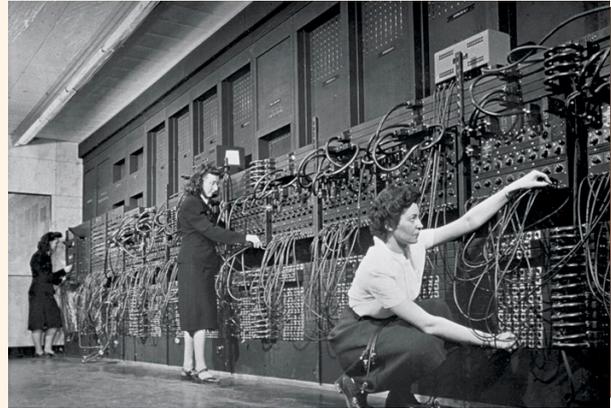
## ❖ Fiabilité

Au début : temps moyen entre pannes  
quelques heures

Après 1948 : deux jours

## Les programmeuses de l'ENIAC

Kathleen McNulty Mauchly Antonelli,  
Jean Jennings Bartik,  
Frances Snyder Holberton,  
Marlyn Wescoff Meltzer,  
Frances Bilas Spence,  
Ruth Lichterman Teitelbaum



Elles ont attendu 1997 pour être officiellement honorées

[eniacprogrammers.org](http://eniacprogrammers.org)

## Le modèle «de von Neumann»

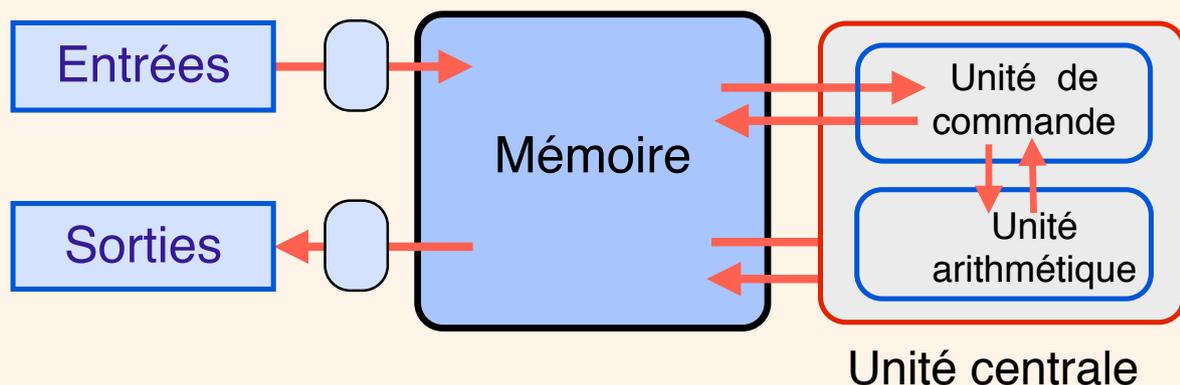
### ❖ von Neumann (avec Eckert et Mauchly, 1945)

### ❖ Traits essentiels

Séparation mémoire-unité centrale

Séparation calcul-séquencement

La mémoire contient les données et les instructions



# La machine à programme enregistré : développement

## ❖ Les efforts initiaux

Les «Moore School lectures» (1946), diffusion des idées

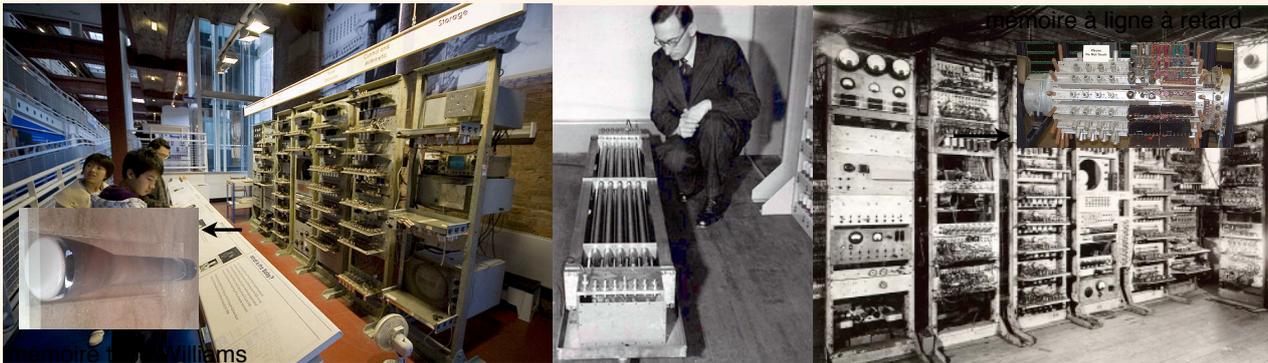
À la suite d'ENIAC : EDVAC (1949-1951)

La machine de Princeton IAS (von Neumann, Bigelow)

## ❖ Les premières réalisations

Mark-1, Université de Manchester (Williams, Kilburn, 1949) - tube Williams

EDSAC, Université de Cambridge (Wilkes, 1949) - ligne à retard



# Où en est-on au début des années 50 ?

## ❖ Les avancées

Turing (et les autres) : définition et limites de l'algorithme, la machine universelle

von Neumann (et les autres): le modèle de base, toujours valable  
les premiers ordinateurs commerciaux

## ❖ Les limites

une «technologie» déficiente, les tubes (peu fiables) ; les mémoires à lignes à retard (complexes, réglage délicat)

une programmation à très bas niveau, sans outils conceptuels,  
fastidieuse et sujette aux erreurs

un champ restreint d'applications

l'informatique est encore (pour longtemps) une affaire de professionnels

# Du binaire à l'assembleur

```
mov    di,[wRootEntries]    ;di=number of entries in the root dir.
mov    cl,4                 ;shift four bits right (/16)
shr    di,cl                ;di=di*32/512 (assumed sector size)
                                ;so, di=number of root dir. sectors
add    [wResSecs],di        ;fix up the data sector variable
                                ;now it points to the first cluster

read_root:
  push  di                  ;preserve the counter

  push  es                  ;preserve es
  mov   cl,1                ;one sector
  call  read_linear         ;read sector (ax gets increased)
  pop   es                  ;restore es

  mov   di,bx               ;es:di->RootDir
  mov   cl,512/32           ;16 entries, sector size of 512 assumed
                                ;(32=size of one entry)

read_root_nextfile:
  push  cx                  ;preserve counter

  mov   cl,11               ;file name length
  mov   si,sFileName        ;ds:di->OS file name
  repe cmpsb                ;compare
  je    short found_system  ;jump if equal (file found)
  add   di,cx                ;increase si by the rest of cx
  add   di,21                ;and fix up by 21 bytes - move to the
                                ;next entry
  pop   cx                  ;restore counter
  loop  read_root_nextfile  ;repeat for next entry
```

un morceau de  
programme en assembleur  
x86

## Qu'est ce qu'un langage de programmation ?

### ❖ Motivation

Un programme en assembleur décrit un algorithme en termes de *ce que sait faire la machine*

On souhaite une expression en termes de *ce que veut l'utilisateur*

### ❖ Qu'attend-on d'un langage de programmation ?

«Haut niveau» d'expression

proche du domaine d'application

indépendant de la machine d'exécution

Efficacité

Puissance d'expression

capable de décrire tous les traitements envisagés

Facilité d'apprentissage et commodité d'utilisation

Rigueur sémantique

Sûreté réduit les risques d'erreur

si  $Z > 0$   
{ $X[3] = N+2$ }

trier(Tableau)

dans les années 1950

aujourd'hui

## Le premier langage de programmation

### ❖ Fortran (*Formula Translation*)

Développé chez IBM par l'équipe de John Backus (1954-57)

Langage orienté vers le calcul scientifique

Évaluation de formules

Définition de fonctions réutilisables

Toujours utilisé aujourd'hui (mais a beaucoup évolué)

### ❖ Une révolution dans la pratique de la programmation

Ouvre la programmation aux utilisateurs ...

... sans sacrifier l'efficacité

### ❖ Les limites du premier Fortran

Manque de rigueur dans la définition

Manque de sûreté : mauvaise détection des erreurs

## Les langages influents des années 60

### ❖ Fortran (1954) - John Backus

Le premier langage de haut niveau, toujours utilisé

### ❖ Algol 60 (1958-61) - comité de scientifiques (Peter Naur, rapporteur)

Définition d'un langage sur une base rigoureuse

objectif atteint, malgré définition par un comité ...

Peu utilisé mais très forte influence  
sur les langages futurs

### ❖ Lisp (1958) - John Mc Carthy, MIT

Premier langage «fonctionnel»

Structure de base, pour  
programmes et données : la *liste*

Le langage favori pour l'intelligence  
artificielle

### ❖ Cobol (1959) - comité d'industriels

Langage pour applications  
de «gestion»

Toujours utilisé, malgré  
son manque d'élégance

### ❖ Basic (1963) - John Kemeny

Langage simple (dérivé de  
Fortran, interactif)

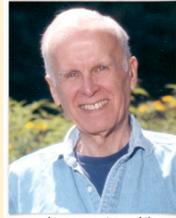
Adapté au temps partagé

```

1 DIMENSION A(11)
2 READ A
3 DO 3,8,11 J=1,11
4 3 I=11-J
5 Y=SQRT(ABS(A(I+1)))+5*A(I+1)**3
6 IF (400. >=Y) 8,4
7 4 PRINT I,999.
8 GO TO 2
9 8 PRINT I,Y
10 11 STOP

```

## Fortran



John Backus  
1924-2007

Image courtesy of the  
Computer History Museum

```

: SET SOURCEFORMAT="FREE"
IDENTIFICATION DIVISION.
PROGRAM ID. Iteration-If.
AUTHOR. Michael Coghlan.

```

## Cobol

```

DATA DIVISION.
WORKING STORAGE SECTION.
01 Num1 PIC 9 VALUE ZEROS.
01 Num2 PIC 9 VALUE ZEROS.
01 Result PIC 99 VALUE ZEROS.
01 Operator PIC X VALUE SPACE

PROCEDURE DIVISION.
Calculator.
PERFORM 3 TIMES
DISPLAY "Enter First Number : " WITH NO ADVANCING
ACCEPT Num1
DISPLAY "Enter Second Number : " WITH NO ADVANCING
ACCEPT Num2
DISPLAY "Enter Operator (+ or *) : " WITH NO ADVANCING
ACCEPT Operator
IF OPERATOR = "+" THEN
ADD Num1, Num2 GIVING Result
ENDIF
IF OPERATOR = "*" THEN
MULTIPLY Num1 BY Num2 GIVING Result
ENDIF
DISPLAY "Result is = ", Result
END PERFORM
STOP RUN

```



Wikimedia Commons  
CC-BY-SA-2.5 by Smithsonian Inst.

Grace Hopper  
1906-1992

## Algol 60



Peter Naur  
1928 - 2016

Wikimedia Commons  
CC-BY-SA-2.5 by ericij

```

procedure hanoi(N,x,y,z); value N,x,y,z;
integer N, x, y, z;
comment déplace une tour de hauteur N
du plot x vers le plot z ;
if N>0 then
begin
hanoi(N-1, x, z, y); (a)
deplacer(x,z); (b)
hanoi(N-1, y, x, z); (c)
end ;

```

## Lisp

John McCarthy  
1927-2011.



Wikimedia Commons  
CC-BY-SA-2.0 by «null0»

```

(DEFUN DEPLACER (d X Y)
(LIST (LIST 'DÉPLACER 'DISQUE d 'DE X 'VERS Y)))

(DEFUN HANOI (N X Y Z)
; Déplace une tour de hauteur N de X vers Z
(COND ((> N 0)
(APPEND (HANOI (- N 1) X Z Y)
(DEPLACER n X Z)
(HANOI (- N 1) Y X Z))))

```

## Avancées technologiques majeures

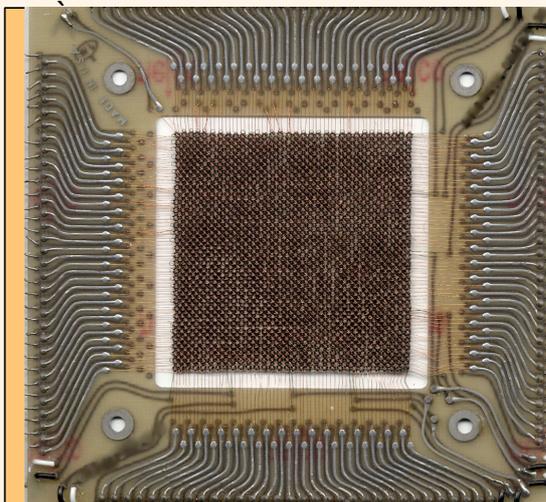
### ❖ Des tubes aux transistors et aux circuits intégrés

Inventions : 1947, transistor ; fin années 50 : circuits intégrés

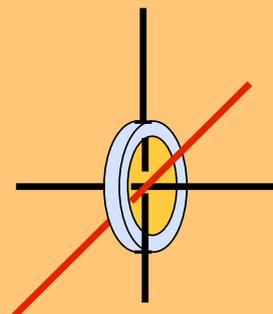
À partir de 1955

circuits : les transistors remplacent les tubes

mémoires : les tores de ferrite remplacent les tambours magnétiques



Bloc de mémoire d'un CDC 6600  
Source : Thierry46, Wikimedia Commons



## Ordinateurs commerciaux 1955-60



IBM 704  
(NACA)

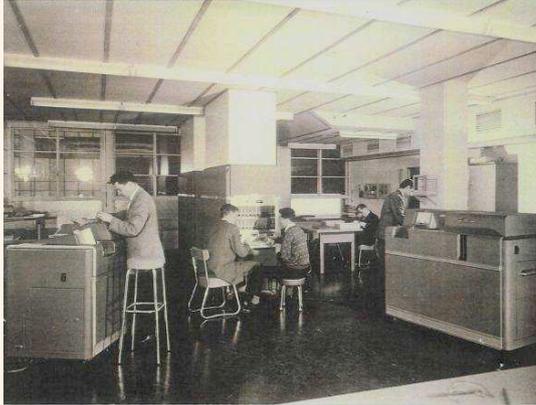
Wikimedia Commons  
domaine public



Univac 1103  
Lockheed

Wikimedia Commons  
domaine public

Wikimedia Commons  
domaine public



Source photo  
Fédération  
des équipes Bull

IBM 650 (Sogreah)



bi-IBM 7090 (NASA)

CC-BY-NC-SA 3.0 FR - S. Krakowiak, 2016/17

Histoire de l'informatique

4 - 23

## Ordinateurs commerciaux des années 1960

### ❖ La première «famille» : la série IBM 360 (1964)

Une ligne d'ordinateurs compatibles

grâce à la microprogrammation

Des avancées techniques

Le cache (IBM 360/85, 1968)

Les machines virtuelles (IBM 360/67, 1966)

Un système d'exploitation gros et complexe (OS/360)

Traitement par lots, premier système utilisant les disques

Un succès commercial



IBM 360/40

Acomit

### ❖ Le premier «super-calculateur» : CDC 6600

Seymour Cray (1964)

Processeurs spécialisés, traitement parallèle



CDC 6600

### ❖ Le premier mini-ordinateur : DEC PDP-8 (1965)



DEC PDP 8

## Les premières applications (1)

### ❖ Le calcul scientifique

L'objectif : trouver des solutions numériques à des problèmes de nature mathématique

Les bases : calcul numérique, puis analyse numérique

Une discipline en très fort développement

Les domaines d'application

Recherche : physique, chimie, mécanique, astrophysique, ...

Médecine et biologie peu impliquées au début

Sciences de l'ingénieur : électronique, électrotechnique, génie civil, nucléaire, spatial ...

Premier outil graphique : Sketchpad (Ivan Sutherland, 1961)

### ❖ La commande de procédés industriels

L'objectif : piloter un processus évolutif pour atteindre un but fixé

Les bases : l'automatique

Les applications : processus de fabrication, chimie, énergie, transports

## Les premières applications (2)

### ❖ Les applications de gestion

Objectif : automatiser les opérations de comptabilité, gestion des stocks, gestion de personnel, ...

Bases :

analyse fonctionnelle (le «quoi ?» : besoins et contraintes)

analyse organique (le «comment ?» : modèle de réalisation)

Limites

méthodes encore peu évoluées, difficulté sous-estimée

pas de notion globale de «système d'information»

### ❖ La recherche opérationnelle

Objectif : aide à la décision, notamment en environnement incertain

Bases : mathématiques «discrètes» (combinatoire, ...)

Applications : initialement militaires, puis approvisionnement, implantation, logistique, ...

### ❖ Une tentative prématurée : la traduction automatique

## Où en est-on vers la fin des années 60 ?

### ❖ Les avancées

Les langages de programmation de haut niveau

Les systèmes d'exploitation

Les nouvelles générations d'ordinateurs

Le développement d'une industrie informatique

Constructeurs de machines

Sociétés de service

### ❖ Les limites

Le défi de la production de logiciel

Programmes incorrects, délais non respectés, budgets dépassés ...

Des domaines d'application encore restreints

Encore peu d'impact sociétal

Une organisation centralisée

## Pour aller plus loin

❖ Une histoire plus complète : <http://lig-membres.imag.fr/krakowia>

❖ Le site de l'association Aconit

<http://aconit.fr>

❖ Un site de diffusion de la culture informatique

Interstices (INRIA, CNRS, universités) : <http://interstices.info/>

❖ Le site de Gérard Berry, Collège de France

<https://www.college-de-france.fr/site/gerard-berry/>

❖ Le site de Gilles Dowek

<http://www.lsv.fr/~dowek/>

❖ Des livres

Andrew Hodges. *Alan Turing, ou l'énigme de l'intelligence*, Payot, 1988

Pierre-Éric Mounier-Kuhn. *L'informatique en France de la seconde guerre mondiale au Plan Calcul*, Presses de l'université Paris-Sorbonne, 2010

## Pour aller plus loin (suite)

### ❖ Sur Alan Turing

Plusieurs articles dans Interstices, C'était hier : <https://interstices.info>

Le film *The Imitation Game* (2014), version romancée de la période Bletchley Park

### ❖ Sur le calcul mécanique

Le site d'André Delvaux : [calmeqa.free.fr](http://calmeqa.free.fr)

### ❖ Sur les débuts de l'informatique en France

site de la fédération des équipes Bull : <http://www.feb-patrimoine.com/>

voir en particulier texte de Bruno Leclerc sur Gamma 3 et Gamma ET

G. Ramunni, La non-construction du premier calculateur électronique du CNRS : [www.histcnrs.fr/pdf/cahiers-cnrs/ramunni.pdf](http://www.histcnrs.fr/pdf/cahiers-cnrs/ramunni.pdf)

F. H. Raymond, Une aventure qui se termine mal : la SEA  
[jacques-andre.fr/chi/chi88/raymond-sea.html](http://jacques-andre.fr/chi/chi88/raymond-sea.html)

## La suite de l'histoire...

### ❖ Le matériel

*Mainframes* et minis, ordinateurs personnels, stations de travail, calcul parallèle

### ❖ Le logiciel

Langages et génie logiciel, systèmes d'exploitation, bases de données

### ❖ L'Internet et le World Wide Web

De l'ARPANET à l'Internet, le Web et la conquête du grand public

### ❖ L'intelligence artificielle

Espoirs et déceptions, mythes et réalité