

# Introduction a la programmation

David Wiedemann

## Table des matières

<b>1</b>	<b>Information, Calcul et Communication</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	A quoi ca sert ? . . . . .	2
1.2.1	Calcul scientifique . . . . .	2
1.2.2	La conduite de processus . . . . .	2
1.2.3	La gestion d'information . . . . .	3
<b>2</b>	<b>Plan du cours</b>	<b>3</b>
<b>3</b>	<b>Algorithmes</b>	<b>4</b>
3.1	Formalisation d'algorithmes . . . . .	4
3.1.1	Methodologie . . . . .	4
3.1.2	Qu'est-ce qu'un algorithme ? . . . . .	4

## List of Theorems

1	Definition (Algorithme) . . . . .	4
2	Definition (Algorithme) . . . . .	4

# 1 Information, Calcul et Communication

## 1.1 Introduction

Objectifs :

- Vous convaincre de l'importance de ce cours
  - insister sur le role de l'informatique
- Presenter l'info en tant que discipline scientifique.

Fonde sur 3 grands principes fondamentaux :

- representation discrete du monde
- representation entachee d'erreurs, mais controlee
- variabilite de la difficulte des problemes et des solutions

## 1.2 A quoi ca sert ?

- la simulation/ l'optimisation
- l'automatisation
- Gestions de donnees

### 1.2.1 Calcul scientifique

- Utilisation : simulation de systemes complexes
- Exigences : grande puissance de calcul.

### 1.2.2 La conduite de processus

- Utilisation : tres nombreuses applications : pilotage/surveillance de processus industriels
- Exigences : necessite d'un faible encombrement, consommation reduite, d un cout minimum et d'une grande fiabilite

### 1.2.3 La gestion d'information

- Utilisation : gestion de systemes bancaires ou boursiers, commerce electronique, fichiers de police
- Exigences : importantes de capacite, traitement efficace, controle de processus

## 2 Plan du cours

1. Fondement du calcul
2. Calcul et algorithme
3. Strategies de calcul
4. theorie du calcul
1. Information et communication
2. Echantillonnage
3. Reconstruction
4. Entropie et information
5. Compression des messages/donnees
1. Fondements des systems
2. Architecture des ordinateurs
3. Stockage et reseaux
1. Secureite informatique
2. RSA
3. Problemes sociaux

## 3 Algorithmes

### 3.1 Formalisation d'algorithmes

#### Definition 1 (Algorithme)

*Un algorithme est une description abstraite des etapes conduisant a la solution d'un probleme*

#### Exemple 1

##### **Probleme :**

*Trouver la valeur maximale dans une liste*

*Une liste c'est un element du produit cartésien de  $E^n$ ,  $n$  taille de la liste.*

*On pourrait ordonner la liste et retourner le dernier element.*

- *comparer les elements de la liste entre eux*
- *a chaque fois prendre le plus gd.*
- *au fur et a mesure*

#### 3.1.1 Methodologie

But : Pour un probleme, trouver une sequence d'actions permettant de produire une solution acceptable en un temps raisonnable.

- Bien identifier le probleme
  - Quelle question ?
- Quelles entrees ?
- Quelles sorties ?
- Trouver un algorithme correct ?
  - verifier qu'il est effectivement correct, qu'il se termine dans tous les cas.
- trouver l'algorithme le plus efficace possible

#### 3.1.2 Qu'est-ce qu'un algorithme ?

Moyen pour un humain de représenter la résolution d'un probleme donne

#### Definition 2 (Algorithme)

*Composition d'un ensemble fini d'operations elementaires bien definies operant sur un nombre fini de donnees et effectuant un traitement bien defini :*

— *suite finie de regles a appliquer*

— *dans un ordre determine*

— *a un nombre fini de donnees*

*Un algorithme peut etre*

— *sequentiel : operations s'executent en sequence*

— *parallele : certaines de ses operations s'executent en parallele : simultanement*

— *reparti : certaines de ses operations s'executent sur plusieurs machines.*