Ecole d'ingénieurs Sup Galilée
Energétique - Informatique - Instrumentation
Mathématiques Appliquées et Calcul Scientifique
Télécommunications et Réseaux

Université Sorbonne Paris Nord

Ingénieurs ENER 1 - Méthodes numériques (S6)

François Cuvelier

Laboratoire d'Analyse Géométrie et Applications
Institut Galilée
Université Paris XIII.

2023/01/24

Divers

- Volume horaire : $8 \times$ 3h cours/TDs, $4 \times 4h$ TPs.
- Prérequis du cours :
  - Cours commun *Mathématiques pour l'ingénieur*, ...
  - *Algorithmique*
  - Langage de programmation (TP) : *Matlab/Octave*
- Note finale : $(2P + CC + TP)/4$.
  - P : partiel du 4 avril 2023 (2h30) ?,
  - TP : travaux pratiques en mai et juin,
  - CC : contrôle continu (2 interrogations écrites de 30mn, séances 3 (21/02) et 6 (21/03) ?).

Objectifs

- Algorithmique numérique.
- Poursuite de l'apprentissage de Matlab/Octave.
- Résolution numériques d'équations différentielles ordinaires (E.D.O.[fr] ou O.D.E.[en]).
- Résolution numériques d'équations aux dérivées partielles (E.D.P.[fr] ou P.D.E.[en]) par des méthodes de différences finies.

Plan du cours

Chapitre I	:	Algorithmique numérique
Chapitre II	:	Dérivation numérique
Chapitre III	:	Résolution numérique des E.D.O.
Chapitre IV	:	Résolution numérique des E.D.P.

Première partie I

Algorithmique numérique

Plan

Définition

Définition 1.1 (Petit Robert 97)

Algorithmique : Enchaînement d'actions nécessaires à l'accomplissement d'une tâche.

Exemple 1 : permutation

Nous voulons permutter deux voitures sur un parking de trois places numérotées de 1 à 3 et ceci sans géner la circulation.
La première voiture, une Renault Zoé, est sur l'emplacement 2, la seconde, une Citroën C1, est sur l'emplacement 3.
Donner un algorithme permettant de résoudre cette tâche.

Exemple 2 :

Donner un algorithme permettant de résoudre

$$ax = b$$

Caractéristiques d'un *bon* algorithme

- Il ne souffre d'aucune ambiguité $\Rightarrow$ très clair.
- Combinaison d'opérations (actions) élémentaires.
- Pour toutes les données d'entrée, l'algorithme doit fournir un résultat en un nombre fini d'opérations.

Première approche méthodologique

Etape 1 : Définir clairement le problème.

Première approche méthodologique

Etape 1 : Définir clairement le problème.

Etape 2 : Rechercher une méthode de résolution (formules, ...)

Première approche méthodologique

Etape 1 : Définir clairement le problème.
Etape 2 : Rechercher une méthode de résolution (formules, ...)
Etape 3 : Ecrire l'algorithme (par raffinement successif pour des algorithmes *compliqués*).

Plan

Vocabulaire de base

- constantes,variables,
- opérateurs (arithmétiques, relationnels, logiques),
- expressions,
- instructions (simples et composées),
- fonctions.

Données et constantes

- Donnée $\Rightarrow$ introduite par l'utilisateur
- Constante $\Rightarrow$ symbole, identificateur non modifiable

Variables

Définition 2.1
Une variable est un objet dont la valeur est modifiable, qui possède un nom et un type (entier, caractère, réel, complexe, tableau, matrice, vecteur...).

Opérateurs arithmétiques

Nom	Symbole	Exemple
addition	$+$	$a+b$
soustraction	$-$	$a-b$
opposé	$-$	$-a$
produit	$*$	$a*b$
division	$/$	a/b

Opérateurs relationnels

Nom	Symbole	Exemple
identique	$==$	$a == b$
différent	$\sim=$	$a \sim= b$
inférieur	$<$	$a < b$
supérieur	$>$	$a > b$
inférieur ou égal	$<=$	$a <= b$
supérieur ou égal	$>=$	$a >= b$

Opérateurs logiques

Nom	Symbole	Exemple
négation	$\sim$	$\sim a$
ou	$\vert$	$a \vert b$
et	$\&$	$a\&b$

Opérateur d'affectation

Nom	Symbole	Exemple
affectation	$\leftarrow$	$a \leftarrow b$

Expressions

Définition 2.2

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Expressions

Définition 2.2

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression numérique

$$(b * b - 4 * a * c)/(2 * a)$$

Expressions

Définition 2.2

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression numérique

$$(b * b - 4 * a * c)/(2 * a)$$

Opérandes $\Rightarrow$ identifiants a, b, c,
constantes 4 et 2.

Expressions

Définition 2.2

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression numérique

$$(b * b - 4 * a * c)/(2 * a)$$

Opérandes $\Rightarrow$ identifiants a, b, c, constantes 4 et 2.
Opérateurs $\Rightarrow$ symboles $*$, $-$ et $/$

Expressions

Définition 2.3

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression booléenne

$$(x < 3.14)$$

Expressions

Définition 2.3

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression booléenne

$$(x < 3.14)$$

Opérandes $\Rightarrow$ identifiants x et contantes 3.14

Expressions

Définition 2.3

Une expression est un groupe d'opérandes (i.e. nombres, constantes, variables, ...) liées par certains opérateurs pour former un terme algébrique qui représente une valeur (i.e. un élément de donnée simple)

Exemple d'expression booléenne

$$(x < 3.14)$$

Opérandes $\Rightarrow$ identifiants x et contantes 3.14
Opérateurs $\Rightarrow$ symboles $<$

Instructions

Définition 2.4

Une **instruction** est un ordre ou un groupe d'ordres qui déclenche l'exécution de certaines actions par l'ordinateur. Il y a deux types d'instructions : simple et structuré.

Instructions simples

- affectation d'une valeur a une variable.
- appel d'une fonction (procedure, subroutine, ... suivant les langages).

Instructions structurées

1. les instructions composées, groupe de pulsieurs instructions simples,
2. les instructions répétitives, permettant l'exécution répétée d'instructions simples, (i.e. boucles «pour», «tant que»)
3. les instructions conditionnelles, lesquels ne sont exécutées que si une certaine condition est respectée (i.e. «si»)

Exemple : boucle «pour»

Algorithm Exemple boucle «pour»

Données : n, un entier positif

1: $S \leftarrow 0$
2: **Pour** $i \leftarrow 1$ **à** n **faire**
3: $S \leftarrow S + \cos(i\hat{\ }2)$
4: **Fin Pour**

Listing – (Matlab) Exemple boucle `for`

```
n=input('n=');
assert(n>=0)
S=0;
for i=1:n
  S=S+cos(i^2);
end
```

Mais que fait-il ?

$S = ?$

Exemple : boucle «tant que»

Algorithm Exemple boucle «tant que»

1: $i \leftarrow 0, \ x \leftarrow 1$
2: **Tantque** $i < 1000$ **faire**
3: $x \leftarrow x + i * i$
4: $i \leftarrow i + 1$
5: **Fin Tantque**

Listing – (Matlab) Exemple boucle `while`

```
i=0;x=1;
while (i<1000)
  x=x+i*i;
  i=i+1;
end
```

Mais que fait-il ?

$i = ?$, $x = ?$

Exemple : instructions conditionnelles «si»

Algorithm Exemple instruction «si»

Données : *age*, un réel.

1: **Si** $age >= 18$ **alors**
2: affiche('majeur')
3: **Sinon Si** $age >= 0$ **alors**
4: affiche('mineur')
5: **Sinon**
6: affiche('en devenir')
7: **Fin Si**

Listing – (Matlab) Exemple instruction `if`

```
age=input('age=');
if age>=18
  disp('majeur')
elseif age>=0
  disp('mineur')
else
  disp('en devenir')
end
```


Plan

Description du problème

- Spécification d'un ensemble de données
  Origine : énoncé, hypothèses, sources externes, ...
- Spécification d'un ensemble de buts à atteindre
  Origine : résultats, opérations à effectuer, ...
- Spécification des contraintes

Recherche d'une méthode de résolution

- Clarifier l'énoncé.
- Simplifier le problème.
- Ne pas chercher à le traiter directement dans sa globalité.
- S'assurer que le problème est soluble (sinon problème d'indécidabilité !)
- Recherche d'une stratégie de construction de l'algorithme
- Décomposer le problème en sous problèmes partiels plus simples.
- Effectuer des raffinements successifs de chaque sous problème. Le niveau de raffinement le plus élémentaire étant celui des instructions.

Réalisation d'un algorithme

- Les types des données et des résultats doivent être précisés.
- L'algorithme doit fournir au moins un résultat (qui peut être graphique).
- L'algorithme doit être exécuté en un nombre fini d'opérations.
- L'algorithme doit être spécifié clairement, sans la moindre ambiguïté.

Exercices

Les deux exercices qui suivent sont intentionnellement mal rédigés ! ! !

Exercice 1

Ecrire un algorithme permettant de calculer

$$S(x) = \sum_{k=1}^{n} k \sin(2 * k * x)$$

Exercices

Exercice 2

Ecrire un algorithme permettant de calculer

$$P(z) = \prod_{n=1}^{k} sin(2 * k * z/n)^k$$

Exercices

Exercice 3

Reprendre les deux exercices précédents en utilisant les boucles «tant que».

Plan

Les fonctions

Les fonctions permettent

- d'automatiser certaines tâches répétitives au sein d'un même algorithme,
- d'ajouter à la clarté de la l'algorithme,
- l'utilisation de portion de code dans un autre algorithme,
- ...

Les fonctions prédéfinies

- les fonctions d'affichage et de lecture : **Affiche**, **Lit**
- les fonctions mathématiques :

$$\sin,\ \cos,\ \exp,\ \cdots$$

- les fonctions de gestion de fichiers
- ...

Ecrire ses propres fonctions

Avant d'écrire une fonction, voici les questions à se poser :

1. Que doit-elle calculer/réaliser précisement (but) ?
2. Quelles sont ses données (avec leurs limitations) ?

Syntaxe

Fonction $[args_1, \ldots, args_n] \leftarrow$ NomFonction($arge_1, \ldots, arge_m$)
  instructions
Fin Fonction

Fonction $args \leftarrow$ NomFonction($arge_1, \ldots, arge_m$)
  instructions
Fin Fonction

Résoudre $ax = b$

Ecrire une fonction permettant de résoudre

$$ax = b$$

- **But** :
- **Données** :
- **Résultats** :

Résoudre $ax = b$

Ecrire une fonction permettant de résoudre

$$ax = b$$

- **But** :
  Enoncé insuffisament précis ! On choisi ici le problème :
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$ donnés, trouver $x \in \mathbb{R}$ solution de $ax = b$
  On aurait pu prendre a une matrice réelle d'ordre n et $b \in \mathbb{R}^n$*...*
- **Données** :
  $a \in \mathbb{R}^*$ et $b \in \mathbb{R}$.
- **Résultats** :
  $x \in \mathbb{R}$.

Résoudre $ax = b$

Algorithm Exemple de fonction : Résolution de l'équation du premier degré $ax = b$.

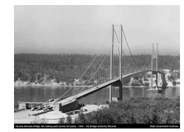
Données : a : nombre réel différent de 0
b : nombre réel.
Résultat : x : un réel.

Fonction $x \leftarrow$ REPD(a, b)
  $x \leftarrow b/a$
Fin Fonction

Plan

Mais avant de pousuivre ...

(a) Pont de la Basse-Chaîne, *Angers* (1850)



(b) Takoma Narrows Bridge, *Washington* (1940)

(c) Millenium Bridge, *London* (2000)

Figure – Une histoire de ponts