

# Tronc Commun

<b>Chapitre</b>	<b>2. Outils et méthodes d'analyse et de description des systèmes</b>
<b>Objectif général de formation</b>	<ul style="list-style-type: none"><li>• identifier les éléments influents d'un système,</li><li>• décoder son organisation,</li><li>• utiliser un modèle de comportement pour prédire ou valider ses performances.</li></ul>
<b>Paragraphe</b>	2.3 Approche comportementale
<b>Sous paragraphe</b>	2.3.6 Comportements informationnels des systèmes
<b>Connaissances</b>	Modèles algorithmiques : structures algorithmiques élémentaires (boucles, conditions, transitions conditionnelles). Variables
<b>Niveau d'enseignement</b>	Première Terminale
<b>Niveau taxonomique</b>	<b>3.</b> Le contenu est relatif à la <b>maîtrise d'outils d'étude ou d'action</b> : utiliser, manipuler des règles ou des ensembles de règles (algorithme), des principes, des démarches formalisées en vue d'un résultat à atteindre.
<b>Commentaire</b>	<i>Activités pratiques liées à la mise en œuvre d'un produit industriel ou d'un système permettant l'application des différents modèles de description de l'information (en statique et en dynamique) et la caractérisation des entrées-sorties de ses différents constituants. Les modèles de comportement sont étudiés autour d'un point de fonctionnement. Au niveau de l'expression de l'information on se limite aux grandeurs statistiques usuelles (moyenne et écart type)</i>
<b>Liens</b>	

## Remarque préliminaire

Cet item « nécessite une étroite coordination avec la progression pédagogique en mathématiques ». Voir [1] page 7/7.

## Structures algorithmiques

D'après [2].

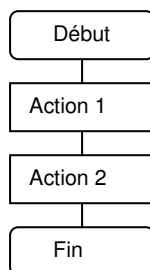
Les codages en langage C sont donnés à titre d'exemple. L'étude de ce langage ne figurant pas dans le tronc commun Sti2D.

La notation algorithmique utilise un **pseudo-code**. Il n'existe pas de normalisation du pseudo-code, mais des **conventions d'usage**.

### Structure linéaire (séquence) .

On exécute successivement une suite d'action dans l'ordre de leur énoncé.

Algorithme



#### Notation algorithmique

**Début**  
*Action 1*  
*Action 2*  
**Fin**

#### Exemple en langage C

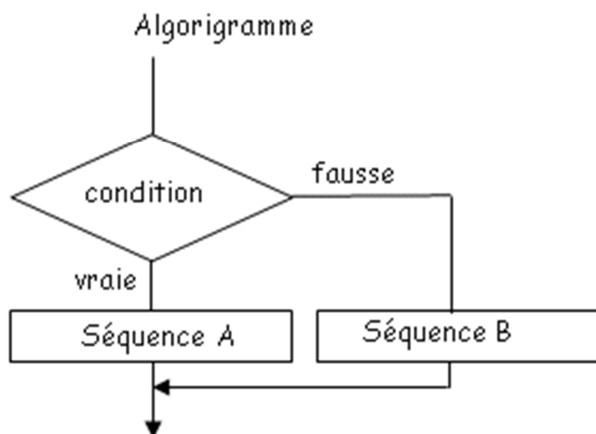
```
{  
    Action 1 ;  
    Action 2 ;  
}
```

## Structures alternatives

### Structure *SI...ALORS...SINON...*

Cette structure offre le choix entre deux séquences s'excluant mutuellement.

# Tronc Commun



## Notation algorithmique

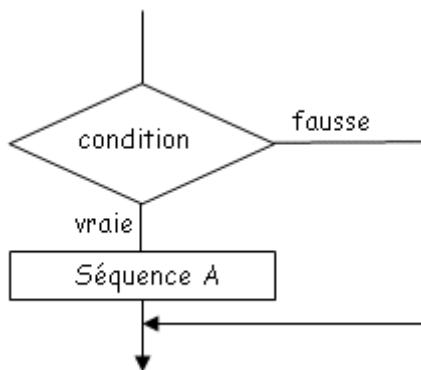
**Si** condition **Alors**  
    Séquence A  
**Sinon**  
    Séquence B  
**Fin Si**

## Exemple en langage C

```
if ( condition )  
{ Séquence A ; }  
else  
{ Séquence B ; }
```

### Remarque :

La structure peut se limiter à **SI...ALORS**, si la condition est vraie on exécute la séquence A si elle est fausse on quitte la structure sans exécuter de séquence.



## Notation algorithmique

**Si** condition **Alors**  
    Séquence A  
**Fin Si**

## Exemple en langage C

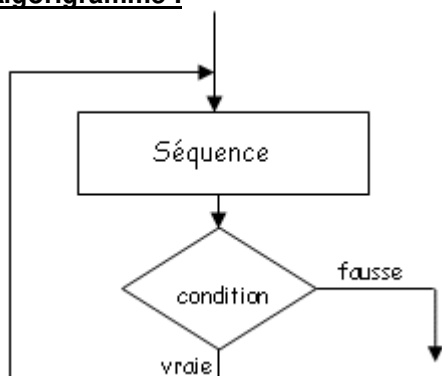
```
if ( condition )  
{ Séquence A ; }
```

## Structures répétitives (ou itératives).

### Structure FAIRE...TANT QUE

La séquence est exécutée au moins une fois, elle est répétée tant qu'elle est vraie.

### Algorithme :



## Notation algorithmique

**Faire**  
    Séquence  
**Tant que** condition vraie

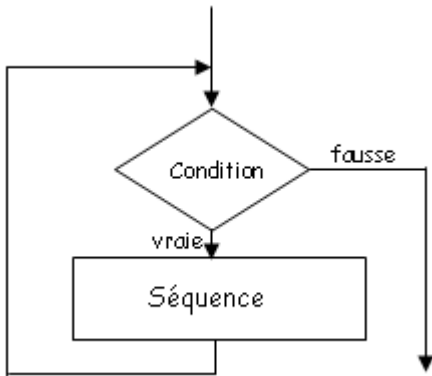
## Exemple en langage C

```
do  
{ Séquence ; }  
while (condition vraie) ;
```

### Structure TANT QUE...FAIRE

On teste d'abord la condition. La séquence est exécutée tant que la condition est vraie.

# Tronc Commun



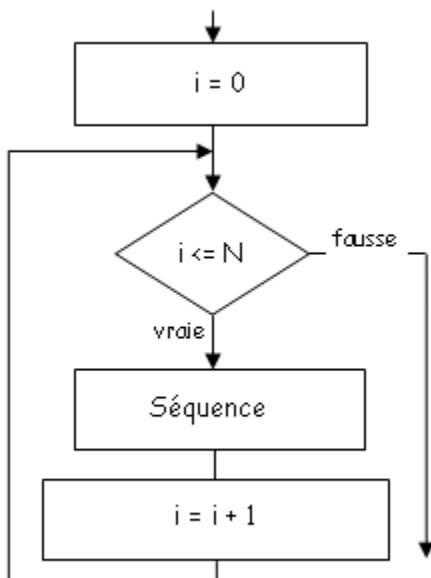
**Notation algorithmique**  
**Tant que** *condition vraie*  
*Séquence*  
**Fin tant que**

**Exemple en langage C**  
**while** (*condition*)  
{ *Séquence* ; }

## Structure POUR...FAIRE

On effectue un nombre d'itérations donné

### Algorithme :



### Notation algorithmique

**Pour** *i = 0 à N*  
**Faire** *Séquence*  
**Fin Pour**

### Exemple en langage C

```
for ( i=0; i<=N; i++ )  
{ Séquence ; }
```

## Variables

En informatique, les variables associent un nom (le symbole) à une valeur ou un objet [4].

Les variables peuvent être **typées**, mais ceci est très dépendant de l'environnement de programmation utilisé.

En **programmation orientée objet**, le type d'une variable est une **classe**, et la variable désigne un **objet** qui est une **instance** de cette **classe**.

## Etudier l'algorithmique sans programmer

Exécuter des algorithmes sans langage de programmation peut sembler une gageure.

C'est pourtant ce que permet Algobox [3].

En réalité, la programmation s'effectue à l'aide d'un pseudo-code dont les instructions sont entrées en cliquant dans l'interface graphique.

# Tronc Commun

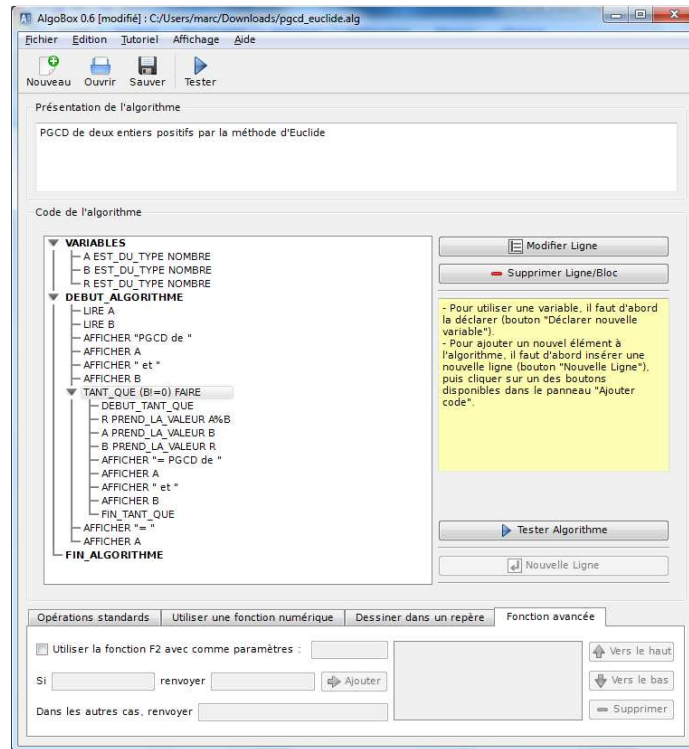


Figure 1 : algorithme du PGCD dans AlgoBox

## Références

- [1] Extrait du B.O. spécial n°3 du 17 mars 2011 : Mathématiques - classe de 1ère des séries STI2D et STL : [http://euler.ac-versailles.fr/webMathematica/textes\\_officiels/officiel\\_2012/maths\\_STI2D\\_STL\\_171037.pdf](http://euler.ac-versailles.fr/webMathematica/textes_officiels/officiel_2012/maths_STI2D_STL_171037.pdf)
- [2] Cours d'algorithmique S-SI – Hoareau - Lycée Louis Payen <http://louispayen.apinc.org/cours/algorithmiques1.doc>
- [3] AlgoBox : Logiciel pédagogique d'aide à la création et à l'exécution d'algorithmes <http://www.xm1math.net/algoBox/>
- [4] Wikipedia – variables (informatique) : [http://fr.wikipedia.org/wiki/Variable\\_%28informatique%29](http://fr.wikipedia.org/wiki/Variable_%28informatique%29)