

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ Abderrahmane Mira - Béjaia  
FACULTÉ DES SCIENCES EXACTES**

Département de Mathématiques

**Mémoire**

**présenté pour l'obtention du diplôme de Master en Mathématiques**

**Option : Analyse et probabilités**

**Par**

**M<sup>me</sup> DELCI née HIDOUSSE Fatima et M<sup>me</sup> MAAFA née AMAR Sabrina**

**Thème**

**Calcul de la fonction de Green de certains problèmes  
aux limites associés aux E.D.O. avec Mathematica**

devant le jury composé de :

|                 |                     |       |                              |               |
|-----------------|---------------------|-------|------------------------------|---------------|
| Mr              | A. BERBOUCHA        | M.C.A | Université A-Mira de Béjaia. | Président.    |
| M <sup>me</sup> | S. ALLILI née ZAHAR | M.C.B | Université A-Mira de Béjaia. | Examinatrice. |
| M <sup>me</sup> | K. KHELOUFI.        | M.C.B | Université A-Mira de Béjaia. | Promotrice.   |

**Juin 2014**

---

# *Remerciements*

*Nous remercions DIEU le tout puissant de nous avoir donné la foi et le courage pour réaliser ce modeste travail et qui a mis dans notre chemin les bonnes personnes. Nous tenons, en premier lieu, à exprimer nos plus vifs remerciements à madame **K.Kheloufi** notre promotrice, pour l'intéressant sujet qu'elle nous a proposé. Ainsi que pour son aide, sa patience, ses conseils, ses encouragements, sa grande disponibilité et son ouverture d'esprit qui nous a aidé à mener à bien ce travail. Nous sommes reconnaissantes pour les connaissances qu'elle nous a prodigué. Il nous est impossible de lui exprimer toute notre gratitude en seulement quelques lignes. Nous adressons notre profonde gratitude et remerciements les plus vifs à monsieur **A. Berboucha**, pour nous avoir fait l'honneur de présider le jury de cette soutenance. Nous tenons à remercier également Madame **S.Allili née Zahar** qui a accepté d'examiner cet humble travail. Nous remercions tous les enseignants qui ont contribué à enrichir nos connaissances tout au long de notre formation. Nous remercions aussi M.Kerai et M. Aissaoui pour leurs aide précieuse durant l'année préparatoire.*

---

# Dédicaces

*Je dédie ce modeste travail à :*

*La mémoire de mes grands-parents.*

*Mes chers parents qui étaient toujours attentifs, affectueux et compréhensifs qui m'ont soutenue durant les laborieuses années de mes études,*

*à qui je témoigne toute ma gratitude.*

*Mes chers frères Rabiâ, Hakim et sa femme Nouria, sans oublier leur petite princesse.*

*Ma très chère sœur Noura et son mari Athmane.*

*A celui avec lequel je vais partager ma vie, mon cher mari Tahar,*

*ainsi que toute ma belle famille chacun par son nom.*

*Tous mes enseignants.*

*Ma copine sabrina et toute sa famille.*

*Ma chère grande famille : mes oncles et mes tantes, mes cousins et mes cousines.*

*Toutes mes amies, surtout Lila, Fahima, Kafia, souad, Chahra, Mazigha, Sabiha.*

*Tous les étudiants de notre département, en particulier notre promotion.*

*FATIMA*

---

# Dédicaces

*Je dédie ce modeste travail avant tout à mes chers parents, mon père qui ne cesse de m'encourager et de me soutenir ; ma mère symbole de sacrifice et qui a tout fait pour moi.*

*A mes frères Abd-Slam et Abd-Elmouméne.*

*A mes sœurs Amel et Widad.*

*A mon mari, qui m'a toujours soutenue et sa famille.*

*Au fiancé de ma sœur, Fouaz.*

*A mes oncles ( Djamel, Smail, Mustapha et Fateh).*

*A ma grande mère.*

*A ma copine fatima et sa famille.*

*A toutes mes amies (surtout Hizia et Walida).*

*Tous les étudiants de notre département, en particulier notre promotion.*

*SABRINA*

---

# Table des matières

|   |           |
|---|-----------|
| <b>Introduction</b>   | <b>1</b>  |
| <b>1 Prise en main de Mathematica</b>   | <b>3</b>  |
| 1.1 Introduction . . . . .  | 3         |
| 1.1.1 Noyau et interface graphique . . . . .  | 5         |
| 1.1.2 L'aide dans Mathematica . . . . .   | 8         |
| 1.2 Calculs avec Mathematica . . . . .  | 8         |
| 1.2.1 Calculs numériques . . . . .  | 8         |
| 1.2.2 Calculs symboliques . . . . .   | 15        |
| 1.2.3 Résolution des équations . . . . .  | 21        |
| 1.3 Fonctions et programmes . . . . .   | 23        |
| 1.3.1 Fonctions prédéfinies par Mathematica . . . . .   | 23        |
| 1.3.2 Fonctions définies par l'utilisateur . . . . .  | 24        |
| 1.3.3 Boucles . . . . .   | 26        |
| 1.3.4 Programation avec des modules . . . . .   | 28        |
| 1.4 Graphes . . . . .   | 29        |
| <b>2 Fonction de Green</b>  | <b>31</b> |
| 2.1 Introduction . . . . .  | 31        |
| 2.2 Définition, existence et unicité de la fonction de Green . . . . .                                | 32        |
| 2.3 Calcul de la fonction de Green associée à des E.D.O. linéaires à coefficients constants . . . . . | 40        |

|          |  |           |
|----------|--|-----------|
| 2.3.1    | Problèmes périodiques . . . . .                        | 47        |
| <b>3</b> | <b>Calcul de la fonction de Green avec Mathematica</b> | <b>50</b> |
| 3.1      | Un module pour calculer la fonction de Green . . . . . | 51        |
| 3.2      | Code de manipulation . . . . .                         | 53        |
| 3.3      | Exemples d'application . . . . .                       | 58        |
|          | <b>Annexes</b>   | <b>62</b> |
|          | <b>Conclusion</b>                                      | <b>67</b> |

---

# Introduction

L'objectif de ce mémoire est de calculer la fonction de Green concernant les problèmes aux limites associés aux équations différentielles ordinaires (E.D.O.) d'ordre  $n$  posés sur des intervalles bornés de  $\mathbb{R}$ , en utilisant le logiciel Mathematica qui permet de faire les calculs symboliques et numériques. Ce mémoire présente d'une manière claire comment déterminer la fonction de Green dans des cas généraux. Le cas des équations différentielles associées à des conditions aux bords non linéaires n'est pas abordé dans ce mémoire.

Les résultats classiques de Peano et Picard assurent, sous des conditions de régularité appropriées sur le terme non linéaire de l'équation, l'existence et l'unicité d'une solution locale, définie dans un voisinage du point de départ, de tout problème de Cauchy.

Cependant, si nous traitons les problèmes aux limites, la solution doit être définie sur l'intervalle entier et l'existence locale n'a aucun sens. L'existence de solutions des problèmes aux limites ne dépend pas uniquement de la régularité des fonctions qui apparaissent dans l'équation. En effet, considérons l'équation  $u'(t) = 1$  associée à la condition aux limites périodique  $u(0) = u(1)$ , il est évident que la solution générale de cette équation est donnée par  $u(t) = t + c$ , avec  $c \in \mathbb{R}$ , et aucune d'elles ne satisfait  $u(0) = u(1)$ .

Donc, pour assurer l'existence de solutions de ce type de problèmes, nous devons tenir compte non seulement de la régularité des fonctions qui apparaissent dans l'équation mais également de la nature des conditions aux bords.

Sous certaines conditions de régularité sur l'opérateur de dérivation linéaire  $L$ , si l'équation  $Lu(t) = \sigma(t)$ ,  $t \in [a, b]$ , associée à des conditions aux bords homogènes appropriées, a seulement la solution triviale  $u \equiv 0$  pour  $\sigma \equiv 0$ , alors l'opérateur linéaire associé  $L$  est inversible et son inverse  $L^{-1}$  est caractérisé par une intégrale du noyau  $G(t, s)$ ,

appelé **fonction de Green**<sup>(1)</sup>. La solution du problème posé est alors donnée par :

$$u(t) = L^{-1}\sigma(t) := \int_a^b G(t, s)\sigma(s)ds, \quad t \in [a, b].$$

L'avantage principal de la fonction de Green est qu'elle est indépendante de la fonction  $\sigma$ . Pour obtenir la solution exacte de l'équation  $Lu(t) = \sigma(t)$ ,  $t \in [a, b]$ , associée à des conditions aux bords homogènes appropriées en changeant le second membre  $\sigma$ , nous devons seulement calculer l'intégrale correspondante.

Ce mémoire comporte trois chapitres organisés comme suit : Dans le premier chapitre, nous avons présenté les notions de base ainsi que quelques techniques avancées du logiciel Mathematica, chaque notion a été illustrée par un exemple d'application. Dans le deuxième chapitre nous avons défini la fonction de Green pour certaines classes de problèmes aux limites, une fonction qui joue un rôle fondamental dans la représentation des solutions de tels problèmes. Plus précisément, la définition de la fonction de Green dans ce chapitre est donnée pour des équations différentielles linéaires d'ordre  $n$  à coefficients constants, associées à des conditions aux limites linéaires. Le troisième chapitre est consacré au calcul de la fonction de Green avec Mathematica, en écrivant un programme qui permet d'illustrer l'étude théorique présentée dans le chapitre précédent.

---

<sup>(1)</sup>George Green (1793-1841) est le premier mathématicien qui a utilisé ce genre de noyau pour résoudre les problèmes aux bords.



---

# Prise en main de Mathematica

## 1.1 Introduction

Mathematica est un puissant logiciel de calcul formel (symbolique) et numérique développé et édité par Wolfram Research depuis 1988, et utilisé dans les milieux scientifiques pour effectuer des calculs algébriques. Il permet essentiellement de faire du calcul formel (manipulation d'expressions mathématiques sous forme symbolique, par exemple : calcul de dérivées et de primitives, simplification d'expressions, etc) et du calcul numérique (évaluation d'expressions mathématiques sous forme numérique; par exemple : calcul des premières décimales du nombre  $\pi$ , évaluation approchée d'intégrales, etc). Mathematica incorpore un langage de programmation sophistiqué et permet aussi de visualiser les résultats en traçant des courbes ou même des surfaces dans l'espace tridimensionnel. Ce logiciel est principalement utilisé dans le domaine des mathématiques mais il trouve aussi une place de choix dans d'autres domaines tels que les sciences physiques, biologiques, sociales, en ingénierie et même en finance.

La version standard de Mathematica contient des milliers de fonctions prédéfinies. Il est possible d'augmenter encore ses capacités en incluant des bibliothèques de fonctions adaptées à des utilisations plus spécifiques dans des domaines spécialisés de physique,

mathématiques, chimie, ingénierie, etc. De ce fait, Mathematica est devenu un outil très utilisé en enseignement, dans la recherche scientifique et dans l'industrie.

Contrairement à la plupart des langages de programmation (tels que C et Fortran), Mathematica peut manier non seulement des quantités numériques (nombres entiers, réels ou complexes), mais aussi des polynômes, des fonctions, des séries, etc. Un autre logiciel de calcul formel et numérique très utilisé est Maple, il a des capacités similaires à celles de Mathematica. Maxima est un logiciel libre et gratuit constituant une alternative aux logiciels commerciaux comme Mathematica ou Maple, avec néanmoins une interface graphique moins conviviale. Matlab est un logiciel commercial très utilisé, édité essentiellement pour effectuer les calculs numériques. Scilab est un logiciel libre alternatif à Matlab.

**Caractéristiques de Mathematica :** Mathematica compte :

- une bibliothèque de fonctions élémentaires,
- une bibliothèque de fonctions spéciales,
- outils pour la manipulation des matrices,
- outils pour manipuler les nombres complexes,
- outils pour tracer de graphes en 2D et 3D mais aussi la création d'animations,
- outils pour la visualisation et l'analyse des graphes,
- outils pour résoudre des problèmes combinatoires, une bibliothèque de fonctions de la théorie des nombres.

Mathematica offre un langage de programmation de type essentiellement fonctionnel et il permet :

- de résoudre différents types d'équations : équations algébriques, équations différentielles, équations aux dérivées partielles,




- de faire des calculs statistiques, de tester des hypothèses, de faire des calculs probabilistes,
- la manipulation des expressions régulières.

Pour réaliser ce chapitre on s'est basé sur les références [4], [5], [6], [7].

### 1.1.1 Noyau et interface graphique

Mathematica est composé de deux parties : le noyau ("**kernel**") et l'interface graphique ("**front end**"). Le noyau constitue le coeur du logiciel; il interprète les instructions d'entrée (écrites en langage Mathematica), puis calcule et retourne le résultat. L'interface graphique s'occupe de l'interaction avec l'utilisateur. Elle gère le fichier de travail (souvent appelé "**feuille Mathematica**" ou "**notebook**"), permet de taper les instructions et de visualiser les résultats. Le logiciel dispose aussi d'un traitement de texte, permettant ainsi d'inclure du texte parmi les calculs effectués. Plusieurs palettes d'outils sont disponibles pour aider à l'édition aussi bien de textes que d'expressions mathématiques (voir menu "**Palettes**").

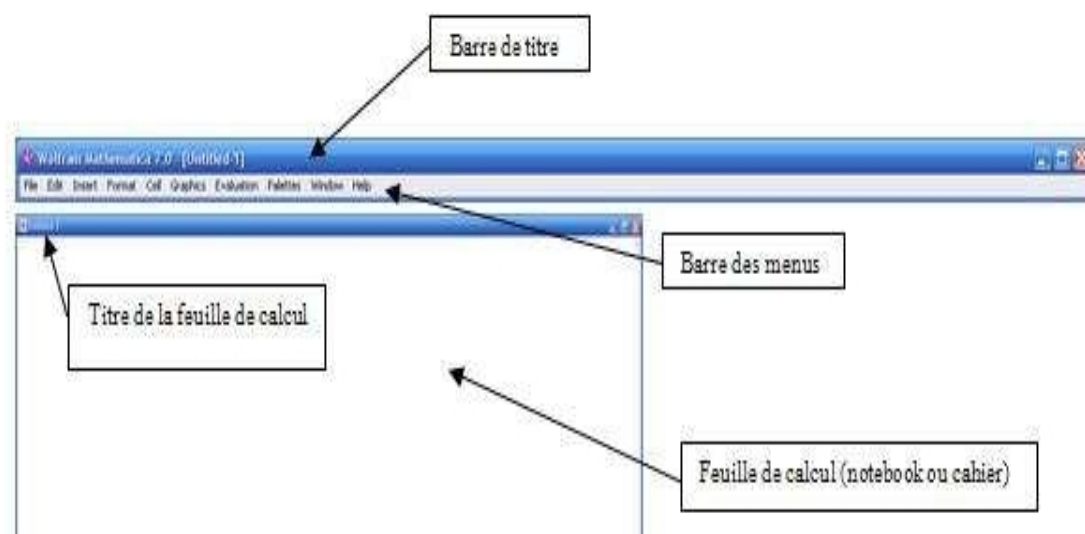
#### Fichier Mathematica

Un fichier Mathematica, aussi appelé "**notebook**", a une extension ".nb". Il est structuré en cellules ("**cells**"). Une cellule est constituée d'une ou de plusieurs lignes et est repérée par un crochet à droite du fichier. On peut avoir des cellules contenant des instructions Mathematica (cellule de type "**In**"), des résultats de calculs (cellule de type "**Out**"), du texte (cellule du type "**Text**"), un titre de paragraphe (cellule de type "**Section**"), etc. On peut sélectionner la totalité d'une cellule en cliquant sur son crochet à droite. On peut ensuite effacer ( $\Leftarrow$ ), copier (+c), couper (+x), coller (+v) la cellule, comme dans un logiciel de traitement de texte normal. Par défaut, les nouvelles cellules créées sont des cellules d'instructions Mathematica. On peut changer le type d'une cellule, après l'avoir sélectionnée, par le menu "**Format > Style**". Une cellule peut contenir plusieurs sous-cellules, formant ainsi des groupes structurés de cellules. Pour

faciliter la lecture du fichier, on peut ouvrir ou fermer des groupes de cellules en double-cliquant sur le crochet correspondant (ou en cliquant sur le triangle à gauche du titre). Il faut penser souvent à sauvegarder le fichier Mathematica (menu **"File"**, puis **"Save"** ou **"Save As"**), les mauvaises manipulations étant fréquentes.

## L'interface de Mathematica

Lorsque vous démarrez le logiciel, voici ce que vous y retrouverez :



Il est possible d'ouvrir plusieurs feuilles de calcul à la fois. Celle qui est en cours d'écriture est affichée dans la barre de titre.

## Utilisation de palettes

Certains utilisateurs préfèrent utiliser des raccourcis vers des commandes. Ces raccourcis sont disponibles dans le menu Palettes de la barre titre. Vous y trouverez également des symboles mathématiques.



**Exemple 1.1.1** Pour calculer la somme de  $(2k-1)$  avec  $k$  variant de 1 jusqu'à 12, on peut utiliser la fonction prédéfinie "**Sum**", comme on peut calculer cette somme d'une façon directe en utilisant les palettes par un clic sur  $\sum$  :

$$In[1] := \sum_{k=1}^{12} (2k - 1)$$

$$Out[1] := 144$$

### Quelques raccourcis clavier

Pour taper plus vite, il est également utile de connaître quelques raccourcis clavier :

|                                     |              |   |   |
|-------------------------------------|--------------|---|---|
| <b>Ctrl</b>                         | +6           | pour faire les puissances (exemple : $x^2$ )      |   |
| <b>Ctrl</b>                         | + /          | pour faire les fractions (exemple $\frac{1}{2}$ ) |   |
| <b>Échap</b>                        | <i>ee</i>    | <b>Échap</b>                                      | pour écrire l'exponentielle à (équivalent à <i>Exp</i> )      |
| <b>Échap</b>                        | <i>P</i>     | <b>Échap</b>                                      | pour écrire le nombre $\pi$ (équivalent à <i>Pi</i> )         |
| <b>Échap</b>                        | <i>ii</i>    | <b>Échap</b>                                      | pour écrire le nombre imaginaire $i$ (équivalent à <i>I</i> ) |
| <b>Échap</b>                        | <i>Inf</i>   | <b>Échap</b>                                      | pour écrire $\infty$ (équivalent à <i>Infinity</i> )          |
| <b>Échap</b>                        | <i>inter</i> | <b>Échap</b>                                      | pour faire l'union « $\cup$ »                                 |
| <b>Échap</b>                        | <i>un</i>    | <b>Échap</b>                                      | pour faire l'intersection « $\cap$ »                          |
| <b>Le symbole <math>\leq</math></b> |              |   | est obtenue en tapant le signe « $<$ » suivit de « $=$ ».     |

### 1.1.2 L'aide dans Mathematica

Mathematica inclut une documentation exhaustive (voir le menu "**Help**", puis "Documentation Center" ou "**Virtual Book**"), incluant de nombreux exemples directement exécutables dans les pages d'aide. La touche "**F1**" est très pratique : dans un fichier Mathematica, après avoir placé le curseur sur une instruction Mathematica, appuyez sur F1 pour afficher la page d'aide correspondante à cette instruction. Pour des informations encore plus complètes, on peut utiliser un moteur de recherche comme Google en ajoutant le mot Mathematica à la recherche. On peut aussi taper "**?commande**" pour accéder à l'aide. La variante ?? donne plus d'informations, si on ne se souvient pas de l'écriture complète de la commande, on peut remplacer les lettres inconnues par "\*" :

?Factor

?? FactorInteger

?Si\*

## 1.2 Calculs avec Mathematica

### 1.2.1 Calculs numériques

#### Un simple calcul arithmétique

Avec Mathematica nous pouvons faire des calculs très simples en utilisant (+) pour l'addition, (-) pour la soustraction, (\*) pour la multiplication et (/) pour la division. Nous tapons une instruction (par exemple "1+5-8"), puis et en appuyant sur la touche "**shift**"( $\uparrow$ ) et "**Entrée**"( $\leftarrow$ ) à la fois, le résultat ("- 2") s'affichant sur une nouvelle ligne

```
In[2]:= 1 + 5 - 8
```

```
Out[2]= -2
```

La ligne des données est désignée par "**In**" suivi du numéro de l'instruction, la ligne de résultat est désignée par "**Out**" suivi du numéro de l'instruction correspondante. Voici un exemple d'un calcul avec une multiplication (\*) et une division (/) des nombres entiers avec une puissance (^) :

```
In[3]:= (2 * 5)^2/9
```

```
Out[3]=  $\frac{100}{9}$ 
```

Dans l'exemple précédent le résultat obtenu est un nombre rationnel, pour avoir une valeur décimale approchée d'une expression, on peut utiliser la fonction "N" en donnant l'argument entre crochets, comme on peut utiliser la forme suivante "expression//N ":

```
In[4]:=N[(2 * 5)^2/9]
```

```
Out[4]= 11.1111
```

```
In[5]:= [(2 * 5)^2/9]//N
```

```
Out[5]= 11.1111
```

Comme on peut faire de simples calculs arithmétiques sur des nombres décimaux :

```
In[6]:= 2.34 + 9.45
```

```
Out[6]= 11.79
```

**Remarque 1.2.1** *Par défaut, Mathematica affiche les calculs avec 6 chiffres significatifs. On peut spécifier une précision plus élevée en donnant le nombre de chiffres significatifs en second argument. Il faut noter que les entiers sont considérés comme des expressions exactes, tandis qu'une expression avec un point est une approximation numérique (par défaut correcte à 6 chiffres). Si l'on mélange plusieurs approximations numériques, Mathematica ajuste la précision à celle du membre le moins précis, comme le montre cet exemple :*

**Exemple 1.2.1** `In[7]:=1. + N[Pi, 50]`

```
Out[7]= 4.14159
```

### Quelques constantes mathématiques

Mathematica dispose de quelques variables déjà définies (constantes) et prêtes à être utilisées, par exemple :

#### - Le nombre $\pi$

```
In[8]:= Pi
```

Out[8]=  $\pi$

In[9]:=  $Pi//N$

Out[9]= 3.14159

**- Le nombre  $e$**

In[10]:=  $E$

Out[10]=  $e$

In[11]:=  $E//N$

Out[11]= 2.71828

**- Le nombre imaginaire  $i$**

In[12]:=  $I$

Out[12]=  $i$

**- L'infini  $\infty$**

In[13]:=  $Infinity$

Out[13]=  $\infty$

**Remarque 1.2.2** Dans Mathematica, il est possible d'affecter une valeur à une variable. Le signe " = " réalise ce que l'on appelle une "affectation" (et on dit qu'on affecte à "x" la valeur 2). Dans tous les calculs suivants, Mathematica remplacera la variable "x" par sa valeur :

In[14]:=  $x = 2$

Out[14]= 2

In[15]:=  $x^3 + 1$

Out[15]= 9

Vous pouvez toujours demander la valeur d'une variable en écrivant le nom de la variable:

In[16]:=  $x$

Out[16]= 2

Pour dégager la valeur de  $x$ , on écrit " $x = .$ " :



`In[17]:= x = .`

`In[18]:= x`

`Out[18]= x`

## Nombres complexes

Il est possible de faire des calculs avec des nombres complexes : Si  $z = x + iy$ , alors le conjugué  $\bar{z} = x - iy$  est obtenu par **Conjugate**[ $z$ ], la partie réelle est obtenue par **Re**[ $z$ ], la partie imaginaire est obtenue par **Im**[ $z$ ]. Dans l'écriture  $z = |z|e^{i\theta}$ , la norme  $|z|$  est donnée par **Abs**[ $z$ ] et l'argument  $\theta$  par **Arg**[ $z$ ]. Dans les manipulations courantes, Mathematica suppose que tous les nombres peuvent être complexes. Il est cependant possible de développer une expression complexe en utilisant "**Expand**" et "**ComplexExpand**".

## Listes

Dans Mathematica, on utilise souvent les listes qui sont définies de la manière suivante :

1. Nom de la liste = {éléments de la liste}.
2. Les éléments de la liste doivent être séparés par des virgules.
3. On extrait un élément d'une liste à l'aide du double crochet `[[n]]` qui repère la position du  $n^{\text{ème}}$  élément de la liste considérée.
4. Pour extraire plusieurs éléments, il faut rajouter des accolades (puisque c'est une sous-liste) ; par exemple `L[[{2, 5, 3}]]`.
5. On peut créer facilement des listes par l'ordre **Range**[ $n$ ] qui crée la liste des  $n$  premiers entiers.

## Opérations sur les liste

Plusieurs opérations sont possibles sur les listes. On peut additionner (+), soustraire (-), multiplier (\*) et diviser (/) les éléments de deux listes en autant qu'elles aient le même format. On peut aussi affecter une liste d'un exposant. Des opérations d'ensemble sont

aussi possibles. L'intersection de deux listes est déterminée en utilisant «  $\cap$  » et encore en faisant l'union par «  $\cup$  ». Lorsqu'une liste est définie, l'utilisation des crochets doubles permet d'en choisir un élément.

**Exemple 1.2.2**  $In[19] := L = \{1, 0, \{2, -1, 5\}, \{3, 4\}\}$

$Out[19] = \{1, 0, \{2, -1, 5\}, \{3, 4\}\}$

$In[20] := L[[2]]$

$Out[20] = 0$

*Cette commande retourne le 2<sup>ème</sup> élément de L.*

$In[21] := L[[4, 1]]$

$Out[21] = 3$

*Cette commande retourne le 1<sup>er</sup> élément du quatrième élément de L.*

## Calcul vectoriel

En Mathematica, les composantes d'un vecteur s'écrivent avec une liste entre accolades et séparées par des virgules. Pour faire appel à la i<sup>ème</sup> composante d'un vecteur, il faut taper seulement "**nomvecteur**[[i]]".

$In[22] := v = \{1, 3\}$

$Out[22] = \{1, 3\}$

$In[23] := v[[2]]$

$Out[23] = 3$

On peut faire plusieurs opérations sur ce vecteur, voici quelques exemples :

### - Multiplication par un scalaire :

$In[24] := 2v$

$Out[24] = \{2, 6\}$

### - Calcul de la norme : Norm[expr, p]

$In[25] := Norm[v, 1]$

$Out[25] = 4$

```
In[26]:= Norm[v, 2]
```

```
Out[26]= $\sqrt{10}$ 
```

```
In[27]:= Norm[v, Infinity]
```

```
Out[27]= 3
```

#### - Produit vectoriel :

```
In[28]:= u = {0, 2}
```

```
In[29]:= v * u
```

```
Out[29]={0, 6}
```

Cette commande retourne la multiplication de  $u$  par  $v$  élément par élément.

#### - Produit scalaire :

```
In[30]:= u = {0, 2}
```

```
In[31]:= v.u
```

```
Out[31]=6
```

#### Calcul matriciel

En Mathematica, une matrice est définie comme une liste de vecteurs lignes. Par exemple, on définit la matrice  $A$  de dimension  $2 \times 2$

```
In[32]:= A = {{0, 2}, {-1, 3}}
```

```
Out[32]= {{0, 2}, {-1, 3}}
```

On peut utiliser la fonction "**MatrixForm**" pour afficher la matrice sous forme plus claire

```
In[33]:= A//MatrixForm
```

```
Out[33]//MatrixForm = 
$$\begin{pmatrix} 0 & 2 \\ -1 & 3 \end{pmatrix}$$

```

#### - Multiplication par un scalaire :

```
In[34]:= 3A
```

```
Out[34]= {{0, 6}, {-3, 9}}
```

**- Calcul de la trace :  $\text{Tr}[ ]$** 

In[35]:=  $\text{Tr}[A]$

Out[35]= 3

**- Calcul du déterminant :  $\text{Det}[ ]$** 

In[36]:=  $\text{Det}[A]$

Out[36]= 2

**- Calcul de la matrice inverse :  $\text{Inverse}[ ]$** 

In[37]:=  $\text{Inverse}[A]$

Out[37]=  $\{\{\frac{3}{2}, -1\}, \{\frac{1}{2}, 0\}\}$

**- Calcul de la transposée :  $\text{Transpose}[ ]$** 

In[38]:=  $\text{Transpose}[A]$

Out[38]=  $\{0, -1\}, \{2, 3\}$

**- Calcul des valeurs propres et des vecteurs propres :**

In[39]:=  $\text{Eigenvalues}[A]$

Out[39]=  $\{2, 1\}$ .

In[40]:=  $\text{Eigenvectors}[A]$

Out[40]=  $\{\{1, 1\}, \{2, 1\}\}$

**- Le produit matriciel se calcule avec l'opération  $\cdot$  :**

In[41]:=  $B = \{\{1, -1\}, \{2, -2\}\}$

Out[41]=  $\{\{1, -1\}, \{2, -2\}\}$

In[42]:=  $A \cdot B$

Out[42]=  $\{\{4, -4\}, \{5, -5\}\}$

**- Calcul de la norme : Norm[expr, p]**

```
In[43]:= Norm[B, 1]
```

```
Out[43]= 3
```

```
In[44]:= Norm[B, 2]
```

```
Out[44]=  $\sqrt{10}$ 
```

```
In[45]:= Norm[B, Infinity]
```

```
Out[45]= 4
```

**- Rang d'une matrice : MatrixRank [ ]**

```
In[46]:= MatrixRank[{ {1, -1}, {2, -2} }]
```

```
Out[46]= 1
```

## 1.2.2 Calculs symboliques

**- Différentiation**

Mathematica peut dériver une fonction par rapport à une de ses variables (dérivée partielle)

$$\mathbf{D}[\text{Expr}, x].$$

Il donne aussi la différentielle totale :

$$\mathbf{Dt}[\text{Expr}]$$

et on note la différentielle  $dx$  par  $Dt[x]$ .

**Exemple 1.2.3** *Pour avoir la dérivée de  $x^2$  il suffit d'écrire sous la forme suivante :*

```
In[47]:= D[x^2, x]
```

```
Out[47]= 2x
```

**- Intégration**

i) Les intégrales indéfinies (primitives) se calculent en un tour de main :

$$\mathbf{Integrate}[\text{Expr}, x]$$

ii) Il en est de même pour les intégrales définies (propres ou impropres) :

**Integrate**[*Expr*, {**x**, *a*, *b*}] et **Integrate**[*Expr*, {*x*, 0, *Infinity*}]

iii) La même chose pour les intégrales multiples :

**Integrate**[ $x^2 + y^2$ , {*x*, 0, 1}, {*y*, 0, *x*}] . L'intégration se fait d'abord sur la variable *y*, puis sur la variable *x*.

vi) Calcul d'une valeur approchée d'une intégrale par **NIntegrate**.

**Exemple 1.2.4** i) *Calcul d'une primitive de sin(x) :*

*In*[48]:=Integrate[Sin[x], x]

*Out*[48]= -Cos[x]

ii) *Calcul de l'intégrale  $\int_0^\pi \sin(x) dx$*

*In*[49]:=Integrate[Sin[x], {x, 0,  $\pi$ }]

*Out*[49]:=2

iii) *Calcul de l'intégrale  $\int_0^\infty e^{-x} dx$  :*

*In*[50]:=Integrate[Exp[-x], {x, 0,  $\infty$ }]

*Out*[50]=  $e^{-0}$

vi) *Calcul d'une valeur approchée de  $\int_0^1 \log(\tan(x)) dx$  :*

*In*[51]:=NIntegrate[Log[Tan[x]], {x, 0, 1}]

*Out*[51]= -0.869182

## - Simplification

Pour simplifier une expression mathématique, il suffit d'écrire

**Simplify**[*Expr*] ou *Expr*//**Simplify**

*In*[52]:=Simplify[Exp[-0]]

*Out*[52]= 1

**- Développement des expressions**

In[53]:=Expand[(x + 1)(x + 3)]

Out[53]= 3 + 4x + x^2

**- Factorisation des expressions**

In[54]:=Factor[3 + 4x + x^2]

Out[54]= (x + 1)(x + 3)

**- Limites**

Le calcul de la limite se fait comme suit :

**t** = expression;

**t/.x->x<sub>0</sub>** ou bien **Limit**[t, x->x<sub>0</sub>].

**Exemple 1.2.5** Calcul de la limite de  $(x^2 + 1)$  quand  $x \rightarrow 1$

In[55]:=Limit[x^2 + 1, x->1]

Out[55]= 2

**- Séries de Taylor**

La série de Taylor d'une fonction  $f(x)$  autour de  $x_0$  "à l'ordre  $n$ " s'obtient en général par

**Series**[f[x], {x, x<sub>0</sub>, n}]

**Exemple 1.2.6** Calcul du développement limité de  $\sin(x)$  autour de  $x = 0$  à l'ordre 3

In[56]:=Series[Sin[x], {x, 0, 3}]

Out[56]= x -  $\frac{x^3}{6}$  + 0[x]^4

**Minimisation et maximisation**

On cherche souvent le minimum ou le maximum global d'une fonction  $f = f(x)$  par rapport à  $x$ . Les fonctions "**Minimize**" et "**Maximize**" permettent d'effectuer cette tâche :

```
In[57]:= f[x_] := 2x^2+5
In[58]:=Minimize[f[x], x]
Out[58]:={5, {x -> 0}}
In[59]:= f[x_] := Exp[-Abs[x + 2]]
In[60]:=Maximize[f[x], x]
Out[60]:={1, {x -> -2}}
```

Ces fonctions retournent la valeur de l'extrémum et sa localisation en  $x$ . Il existe également des variantes numériques de ces fonctions dans le cas où l'extrémum n'est pas calculable exactement :

```
In[61]:= f[x_] := x^4 + x^3 + x^2 + x + 1
In[62]:=NMinimize[f[x], x]
Out[62]:={0.673553, {x -> -0.60583}}
In[63]:= f[x_] := -x^4 + x^3 + x^2 + x + 1
In[64]:=NMaximize[f[x], x]
Out[64]:={3.33157, {x -> 1.28858}}
```

Les fonctions "**FindMinimum**" et "**FindMaximum**" permettent de rechercher des minimums et maximums locaux en donnant un point de départ. Par exemple, la recherche d'un minimum local de la fonction  $\text{Sin}(x)\text{Exp}(-x^2)$  en partant de  $x = -1.5$  donne

```
In[65]:=FindMinimum[Sin[x]Exp[-x^2], {x, -1.5}]
Out[65]:={-0.396653, {x -> -0.653271}}
```

En partant de  $x = 1.2$ , on trouve un autre minimum local

```
In[66]:=FindMinimum[Sin[x]Exp[-x^2], {x, 1.2}]
Out[66]:={-2.94488 × 10-6, {x -> 3.29231}}
```

De la même manière, voici un exemple de recherche d'un maximum local en partant de  $x = 2$

```
In[67]:=FindMaximum[Sin[x]Exp[-x^2], {x, 2}]
Out[67]:={0.396653, {x -> 0.653271}}
```



## Substitution

L'opération `"/."` (correspondante à une fonction appelée **"ReplaceAll"** en Mathematica) permet de substituer dans une expression une variable par une valeur ou par une autre variable. Par exemple, pour remplacer  $x$  par 4 dans l'expression  $(x + 1)$ , on peut écrire

```
In[68]:= x + 1 /. x -> 4
```

```
Out[68]= 5
```

`"x -> 4"` s'appelle une règle (**"rule"** en anglais) de substitution. L'avantage est que `"4"` n'est pas affecté à la variable  $x$  globalement dans tout le notebook, mais la substitution ne se fait que localement dans l'expression ci-dessus, et on peut continuer à utiliser  $x$  par la suite comme une variable non définie :

```
In[69]:= x
```

```
Out[69]= x
```

On peut faire plusieurs substitutions avec la syntaxe suivante

```
In[70]:= (x + abc)^3 /. {x -> 1, abc -> y}
```

```
Out[70]= (1 + y)^3
```

Souvent les fonctions Mathematica retournent les résultats sous forme d'une liste de règles, par exemple :

```
In[71]:= resultat = Solve[x^2 - x - 2 == 0, x] (voir la page 21)
```

```
Out[71]= {{x -> -1}, {x -> 2}}
```

où nous avons stocké le résultat dans la variable **"resultat"**. La première solution  $x = -1$  correspond au premier élément de la liste **"resultat"** :

```
In[72]:= resultat[[1]]
```

```
Out[72]= {x -> -1}
```

Puisque `resultat[[1]]` est sous la forme d'une règle, il suffit d'appliquer une substitution sur la variable `"x"` pour extraire la valeur `"-1"`:

```
In[73]:= x /. resultat[[1]]
```

```
Out[73]= -1
```

## Pipeline

L'opération `//` (appelée "**pipeline**", ou tube en français) permet d'appliquer une fonction au dernier résultat avec une syntaxe inversée. Ainsi, l'instruction suivante :

```
In[74]:=x//f
Out[74]= f[x]
est équivalente à
In[75]:=f[x]
Out[75]= f[x]
```

L'exemple ci-dessus ne présente pas vraiment d'intérêt et rend l'écriture beaucoup moins claire. En revanche, le pipeline est très pratique pour appliquer une fonction Mathematica à une expression déjà tapée sans avoir à revenir au début de la ligne, par exemple:

```
In[76]:= (cos[x] + sin[x])^2 - 2cos[x]sin[x]// Expand
Out[76]= cos[x]^2 + sin[x]^2
```

On peut empiler plusieurs fonctions avec les pipelines, par exemple appliquons la fonction "**Simplify**" au résultat précédent

```
In[77]:= (cos[x] + sin[x])^2 - 2cos[x]sin[x]// Expand // Simplify
Out[77]= 1
```

Autre exemple, appliquons la fonction "**N**" au nombre  $\pi$  avec un pipeline

```
In[78]:=  $\pi$  // N
Out[78]= 3.14159
```

Il est également possible d'utiliser le pipeline pour appliquer une fonction avec plus d'un argument, on utilise alors la syntaxe suivante :

```
In[79]:=  $\pi$  // N[#, 10]&
Out[79]= 3.1415926535
```

qui est équivalent à

```
In[80]:= N[ $\pi$ , 10]
Out[80]= 3.1415926535
```

Le symbole "#" repère l'endroit où l'expression qui précède (ici  $\pi$ ) sera insérée et "&" marque la fin de la fonction. La simplicité de la syntaxe est néanmoins un peu perdue.

### 1.2.3 Résolution des équations

#### Equations algébriques

Pour définir une équation nous devons écrire « == ». On aura donc

$$\text{expression1} == \text{expression2}.$$

La commande **Solve**[équation, variable] permet de résoudre une équation.

**Remarque 1.2.3** *Bien noter la différence entre les écritures "=" et "==": la première est une attribution et la seconde est une comparaison.*

**Exemple 1.2.7** *Résolution de l'équation  $x^2 - 1 = 0$  d'inconnue  $x$  :*

`In[81]:=Solve[x^2 - 1 == 0, x]`

`Out[81]= {{x -> -1}, {x -> 1}}`

La fonction "Solve" ne permet pas uniquement de résoudre une seule équation à une seule inconnue. Elle peut traiter n'importe quel nombre d'équations et d'inconnues. Par exemple, pour résoudre le système d'équations linéaires à deux inconnues

$$\begin{cases} 2x + y = 2 \\ 4x - 3y = 0, \end{cases}$$

on utilise "Solve" de la façon suivante (après éventuellement avoir effacé avec la fonction "Clear" toutes valeurs pouvant être déjà contenues dans  $x, y$ )

`Clear[x, y]`

`In[82]:=Solve[{2x + y == 2, 4x - 3y == 0}, {x, y}]`

`Out[82]= {{x -> 3/5, y -> 4/5}}`

On peut aussi résoudre des équations non linéaires, si de plus les calculs ne sont pas réalisables alors il faut utiliser une résolution numérique approchée avec "NSolve".

**Exemple 1.2.8** Calcul des solutions approchées de l'équation :  $x^3 + x + 5 = 0$  :

$In[83]:=NSolve[x^3 + x + 5 == 0, x]$

$Out[83]=\{\{x \rightarrow -1.51598\}, \{x \rightarrow 0.75799 - 1.65035i\}, \{x \rightarrow 0.75799 + 1.65035i\}\}$

## Equations différentielles

La fonction "**DSolve**" permet de résoudre des équations différentielles. Définissons d'abord une équation différentielle et sa condition initiale :

$equadif = y'[x] == ay[x];$

$condinit = y[0] == 1$

La solution générale et celle qui satisfait la condition initiale s'obtiennent alors par

**DSolve**[equadif, y[x], x] et **DSolve**{equadif, condinit}, y[x], x].

Notons que le deuxième argument est la fonction pour laquelle on veut résoudre l'équation différentielle, tandis que le troisième argument est la variable indépendante. Il est également possible de résoudre un système d'équations différentielles :

$DSolve[\{x'[t] == y[t], y'[t] == x[t]\}, \{x[t], y[t]\}, t]$

**Exemple 1.2.9** Résolution d'une équation différentielle du second ordre non homogène

$In[83]:=DSolve[y''[x] + y[x] == Sin[x], y[x], x]$

$Out[83]=\{\{y[x] \rightarrow C[1]Cos[x] + C[2]Sin[x] + \frac{1}{4}(-2xCos[x] - 2Cos[x]^2Sin[x] + Cos[x]Sin[2x])\}\}$

$C[1], C[2]$  sont des constantes d'intégration. On peut ajouter des conditions initiales (ou aux limites) pour déterminer les constantes  $C[1]$  et  $C[2]$ .

$In[84]:=DSolve[\{y''[x] + y[x] == Sin[x], y[0] == 0, y[1] == 0\}, y[x], x]$

$Out[84]=\{\{y[x] \rightarrow \frac{1}{4}(-2xCos[x] + 2Cos[1]^2Sin[x] - 2Cos[x]^2Sin[x] + 2Cot[1]Sin[x] - Cot[1]Sin[2]Sin[x] + Cos[x]Sin[2x])\}\}$

Si l'équation différentielle n'est pas résoluble exactement, on peut trouver une solution numérique avec la fonction "**NDSolve**".

**Remarque 1.2.4** *Il est possible de combiner des résultats obtenus précédemment pour faire des calculs plus complexes. On peut utiliser % pour référer au résultat précédent, %% à celui d'avant, et ainsi de suite. Comme on peut utiliser l'expression %n qui réfère au résultat Out[n].*

**Exemple 1.2.10**  $In[85]:= 34^2$

$Out[85]= 1156$

$In[86]:= 12^3$

$Out[86]= 1728$

$In[87]:= \% - \%$

$Out[87]= -572$

## 1.3 Fonctions et programmes

### 1.3.1 Fonctions prédéfinies par Mathematica

Mathematica possède une large collection de fonctions prédéfinies. Par convention :

- 1- ces fonctions commencent par une majuscule et prennent leur argument entre des crochets. Si une fonction est composée de plus d'un mot, chacun de ceux-ci doit débiter par une lettre majuscule;
- 2- les arguments doivent être définis à l'intérieur de crochets;
- 3- les arguments doivent être séparés par des virgules.

## Quelques fonctions mathématiques

|  |  |
|--|--|
| <b>N</b> [ $x, m$ ] ou <b>N</b> [ $x$ ]  | $m$ chiffres significatifs, si $m$ est absent,<br>l'approximation se fera à 6 chiffres   |
| <b>Sin</b> , <b>Cos</b> , <b>Tan</b> , <b>Cot</b><br><b>Sec</b> ( $\frac{1}{\text{Cos}}$ ), <b>Csc</b> ( $\frac{1}{\text{Sin}}$ )<br><b>ArcSin</b> , <b>ArcCos</b> , <b>ArcTan</b> | Les fonctions trigonometriques.<br>Les calculs se font avec des angles en radians.<br>Si par contre vous désirez utiliser des angles en degrés,<br>il faut utiliser le symbole «°» |
| <b>Exp</b> [ $x$ ]   | Retourne le nombre $e$ affecté de l'exposant $x$   |
| <b>Log</b> [ $x$ ] et <b>Log</b> [ $b, x$ ]  | Logarithme népérien et logarithme en base $b$ , le plus<br>souvent $b=10$  |
| <b>Factor</b> [ <i>polynôme</i> ]  | Factorise un polynôme  |
| <b>FactorInteger</b> [ $n$ ]   | Factorise un nombre $n$ en produit de nombres premiers   |
| <b>Expand</b> [ <i>expression</i> ]  | Développe une expression algébrique  |
| <b>Numerator</b> [ <i>fraction</i> ]<br><b>Denominator</b> [ <i>fraction</i> ]   | Le numérateur et le dénominateur d'une fraction algébrique   |
| <b>Apart</b> [ <i>fraction</i> ]   | Décomposer une fraction en somme de fractions partielles   |
| <b>GCD</b> [ $a, b$ ]  | Le plus grand diviseur commun entre $a$ et $b$   |
| <b>LCM</b> [ $a, b$ ]  | Le plus petit multiple commun entre $a$ et $b$   |
| <b>Sqrt</b> [ $z$ ] ou $z^{(1/2)}$   | Racine carrée du nombre $z$  |
| <b>Max</b> [ $\{x, y, \dots\}$ ], <b>Min</b> [ $\{x, y, \dots\}$ ]   | Le maximum et minimum d'une liste  |
| <b>Factorial</b> [ $n$ ] ou $n!$   | Factorielle  |
| <b>Simplify</b> [ <i>expression</i> ]  | Simplifier une expression algébrique   |
| <b>Abs</b> [ $z$ ]   | valeur absolue du nombre $z$   |

## 1.3.2 Fonctions définies par l'utilisateur

Voici quelques règles à suivre pour la définition de vos propres fonctions :

1- L'argument de la fonction doit être suivi du caractère « `_` » afin d'identifier l'argument comme étant une variable. Cette variable pourra ainsi être remplacée lors de l'utilisation de la fonction.

- 2- Utiliser des crochets pour définir les arguments.
- 3- Utiliser `:=` pour séparer le nom de la fonction (à gauche) et sa définition (à droite).

**Exemple 1.3.1** `In[88]:= f[x_] := x^3 + 1`

`In[89]:= f[2]`

`Out[89]= 9`

`In[90]:= f[2 + 3I]`

`Out[90]= -45 + 9i`

On peut définir des fonctions à plusieurs variables

`In[100]:= g[x_, y_] := x^2 + y/2`

**Remarque 1.3.1** 1. Si l'on veut réutiliser le même nom d'une variable ou d'une fonction dans un autre contexte plus tard, on peut le libérer avec

`Clear[nom de variable ou fonction à libérer]`

2. Pour supprimer définitivement une variable ou une fonction ainsi que sa définition, on peut utiliser

`Remove[nom de variable ou fonction à supprimer]`

3. Il est bien entendu interdit de libérer les variables et les fonctions prédéfinies.

### Fonction définie par parties

La fonction `Piecewise`[`{{valeur1, condition1}, {valeur2, condition2}, ...}`] permet de définir une fonction par parties.

**Exemple 1.3.2**

`In[101] := g[x_] := Piecewise[{{x^2, x < 0}, {1, x == 0}, {x + 1, x > 0}}]`

La fonction définie est  $g(x) = \begin{cases} x^2 & \text{si } x < 0 \\ 1 & \text{si } x = 0 \\ x + 1 & \text{si } x > 0 \end{cases}$

Le symbole  $\leq$  est obtenue en tapant le signe « < » suivi de « = ».

## Lettres protégées

Les lettres **C**, **D**, **E**, **I**, **K**, **N** et **O** sont protégées par Mathematica, ce qui rend leur utilisation impossible pour la définition de fonctions.

### 1.3.3 Boucles

#### Boucle Do

Une boucle **Do** permet d'exécuter plusieurs fois une instruction. La commande

**Do**[instruction, {i, debut, fin}]

fait varier l'entier "i" (appelé "itérateur") de l'entier "debut" à l'entier "fin" et exécute à chaque fois "instruction". Par exemple pour afficher "Coucou" trois fois, on utilise une boucle **Do** exécutant l'instruction **Print**["Coucou"] lorsque i varie de 1 jusqu'à 3 :

```
In[102]:=Do[Print["Coucou"], {i, 1, 3}]
```

```
Out[102]=Coucou
```

```
Coucou
```

```
Coucou
```

La boucle **Do** est plus intéressante si on a besoin d'utiliser l'entier i dans l'instruction.

Par exemple, affichons les 3 premiers entiers

```
In[103]:=Do[Print[i], {i, 1, 3}]
```

```
Out[103]=1
```

```
2
```

```
3
```

Utilisons une boucle **Do** pour calculer la somme des 30 premiers entiers  $s = \sum_1^{30} i$ .

On commence par initialiser la variable s à 0

```
In[104]:= s = 0
```

```
Out[104]= 0
```

puis on fait varier i de 1 jusqu'à 30 et à chaque itération on calcule  $s + i$  et on affecte le résultat à la variable s

```
In[105]:= Do[s = s + i, {i, 1, 30}]
```



A la fin des itérations, la variable `s` contient la somme recherchée

```
In[106]:= s
```

```
Out[106]= 525
```

Il est souvent plus clair (et plus sûr) de regrouper ces trois instructions en une seule instruction dans une seule cellule en les séparant par des ";" (*qui permettent de supprimer les affichages intermédiaires*)

```
In[107]:= s = 0;
```

```
Do[s = s + i, {i, 1, 30}];
```

```
s
```

```
Out[107]= 525
```

Calculons la somme `s` et le produit `p`, des 7 premiers entiers,  $s = \sum_1^7 i$ ,  $p = \prod_{i=1}^7 i$  avec une seule boucle `Do` :

```
In[108]:= s = 0;
```

```
p = 1;
```

```
Do[s = s + i;
```

```
p = p * i, {i, 1, 7}];
```

```
s
```

```
p
```

```
Out[108]=28
```

```
Out[108]=5040
```

**Remarque 1.3.2** *Les retours à la ligne ne sont pas obligatoires mais facilitent la lecture.*

Remarquons que Mathematica dispose d'une fonction "**Sum**" permettant de faire cette somme plus directement

```
In[109]:= Sum[i, {i, 1, 30}]
```

```
Out[109]=525
```

Néanmoins, les boucles `Do` permettent de faire beaucoup plus de choses. Par défaut, l'itérateur varie par incrément de 1, mais on peut aussi utiliser un autre incrément. Par exemple, dans la boucle **Do** suivante l'entier `j` varie de 3 jusqu'à 8 par incrément de 2

```
In[110]:= Do[Print[j], { j, 3, 8, 2}]  
3  
5  
7
```

### Structure conditionnelle If

La structure conditionnelle "**If**" permet d'exécuter une instruction ou une autre suivant qu'une condition (appelée expression logique en informatique) est vraie ou fausse. La syntaxe est la suivante :

**If** [expression logique, instruction1, instruction2].

Si "expression logique est vraie ("True" en Mathematica) alors uniquement "instruction1" sera exécutée; si au contraire "expression logique" est fausse ("False" en Mathematica) alors uniquement "instruction2" sera exécutée.

**Exemple 1.3.3** *In[111]:=If [7 < 8, x, y]*

*Out[111]= x*

### 1.3.4 Programation avec des modules

En Mathematica, un "Module" est l'équivalent de ce que l'on appelle souvent une "routine" dans d'autres langages de programmation. Il s'agit d'un ensemble d'instructions regroupées dans une même entité indépendante. Reprenons l'exemple du calcul du produit scalaire de deux vecteurs par une boucle **Do**. On peut utiliser un module pour définir une fonction "**produitscalaire**" prenant comme arguments deux vecteurs *u* et *v*, et retournant la valeur de leur produit scalaire :

```
In[112]:= produitscalaire[u_, v_] := Module[{produit, i},  
produit = 0;  
Do[produit = produit + u[[i]]v[[i]], {i, 1, 3}];  
produit]
```

Dans cet exemple, "*produit*, *i*" sur la première ligne indique que "*produit*" et "*i*" sont des variables locales, c'est-à-dire qu'elles seront modifiées uniquement à l'intérieur du Module, et pas globalement pour tout le notebook. Utiliser des variables locales limite les risques d'erreurs en évitant les conflits avec d'autres parties du notebook où ces mêmes variables pourraient être aussi utilisées. L'autre avantage de créer des modules est de pouvoir réutiliser facilement autant que l'on veut une même partie de code. Par exemple, on peut calculer le produit scalaire des vecteurs  $\{2, 0, 1\}$  et  $\{1, -1, 3\}$

```
In[113]:= produitscalaire[{2, 0, 1}, {1, -1, 3}]
```

```
Out[113]= 5
```

puis le produit scalaire des vecteurs  $\{4, 1, 6\}$  et  $\{2, 3, 0\}$

```
In[114]:= produitscalaire [{4, 1, 6}, {2, 3, 0}]
```

```
Out[114]= 11
```

Découper un long code en petits modules indépendants permet de diminuer la complexité du code. C'est une bonne pratique de programmation.

## Commentaires

Pour insérer des commentaires dans un programme on l'encadre par :

```
(*... commentaire ...*).
```

## 1.4 Graphes

Tracer le graphe d'une fonction ou d'une collection de fonctions est aisé avec Mathematica, voici un exemple de traçage :

```
Plot[expr, {x, a, b}], Plot[{expr1, expr2}, {x, a, b}]
```

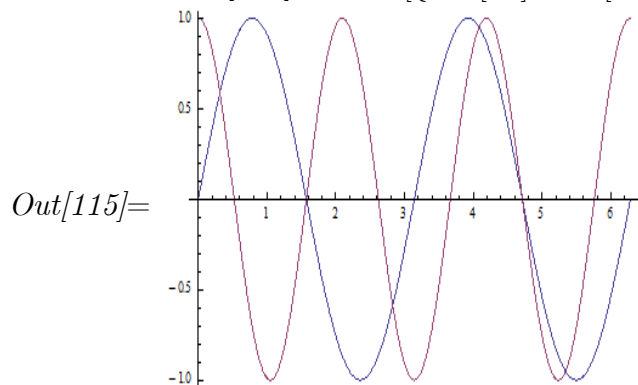
(Tracer l'expression quand  $x$  prend des valeurs de  $a$  à  $b$ ). Les possibilités pour personnaliser le graphe sont nombreuses, par exemple :

```
Plot[expr, {x, a, b}, Frame → True, GridLines → Automatic]
```

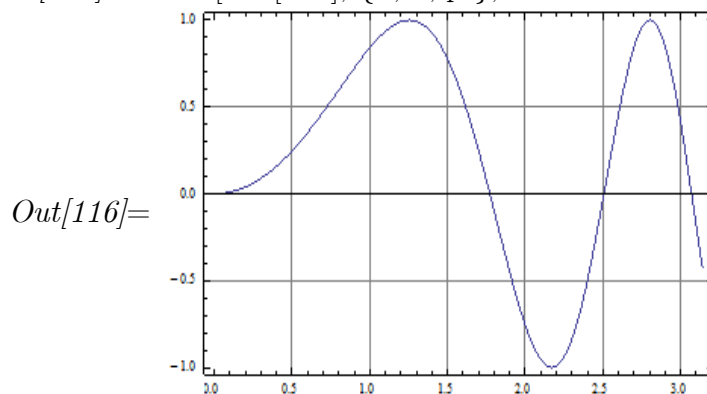
On peut également faire des graphes paramétriques en trois dimensions :

**ParametricPlot3D**[{*exper1*, *exper2*, *exper3*}, {*t*, *a*, *b*}, {*u*, *c*, *d*}]

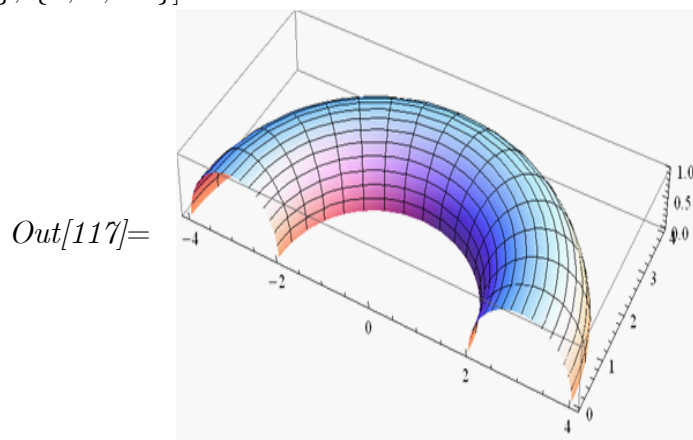
**Exemple 1.4.1** *In*[115]:= **Plot**[{*Sin*[2*x*], *Cos*[3*x*]}, {*x*, 0, 2*Pi*}]



*In*[116]:= **Plot**[*Sin*[*x*<sup>2</sup>], {*x*, 0, *pi*}, *Frame* -> *True*, *GridLines* -> *Automatic*]



*In*[117]:= **ParametricPlot3D**[{*Cos*[*t*](3 + *Cos*[*u*]), *Sin*[*t*](3 + *Cos*[*u*]), *Sin*[*u*]}, {*t*, 0, *Pi*}, {*u*, 0, *Pi*}]



## 2.1 Introduction

Lorsqu'on considère une équation différentielle ordinaire linéaire homogène d'ordre  $n$ ,  $Lu(t) = 0$ ,  $t \in [a, b]$ , où  $L$  est un opérateur de dérivation linéaire, on sait très bien qu'il y a  $n$  fonctions linéairement indépendantes, qui génèrent la solution générale de cette équation, c'est-à-dire n'importe quelle solution de l'équation considérée est une combinaison linéaire de telles fonctions. Quand une équation non homogène  $Lu(t) = \sigma(t)$ ,  $t \in [a, b]$  est considérée, nous devons trouver une solution particulière, pour chaque fonction  $\sigma$  donnée. En l'ajoutant à l'expression générale de la solution de l'équation homogène, nous obtenons la solution générale de l'équation non homogène.

Si nous fixons les valeurs de  $u$  et de ses  $(n - 1)$  premières dérivées au point de départ  $a$ , sous des conditions de régularité appropriées sur les données, nous savons qu'un tel problème de Cauchy est uniquement soluble. De plus, nous aurons l'expression de sa solution unique en obtenant les valeurs uniques des coefficients dans l'expression donnée pour le cas général. Quand les coefficients de l'opérateur linéaire  $L$  sont des constantes, de telles valeurs sont calculées en résolvant un système algébrique linéaire d'ordre  $n$ .

Cependant, lorsque nous traitons le problème dans lequel la fonction  $u$ , et/ou certaines de ses dérivées jusqu'à l'ordre  $n - 1$ , atteignent leurs valeurs en deux points extrêmes  $a$  et  $b$ , nous savons que l'existence de solution d'un tel problème aux limites a deux points, en général, n'est pas assurée. Pour cette raison il est très important de développer des

outils qui nous permettent d'assurer l'existence et l'unicité de la solution de ce type de problèmes, et encore de calculer son expression exacte.

Parmi les méthodes utilisées pour résoudre ce type de problèmes, on trouve le calcul de la fonction de Green : en général, si l'équation  $Lu(t) = \sigma(t)$  associée à des conditions aux bords homogènes, a seulement la solution triviale  $u \equiv 0$  pour  $\sigma \equiv 0$ , alors l'opérateur linéaire associé  $L$  est inversible et son inverse  $L^{-1}$  est caractérisé par une intégrale du noyau  $G(t, s)$ , appelé **fonction de Green**. La solution du problème posé est alors donnée par :

$$u(t) = L^{-1}\sigma(t) := \int_a^b G(t, s)\sigma(s)ds, \quad t \in [a, b].$$

## 2.2 Définition, existence et unicité de la fonction de Green

Considérons le problème aux limites à deux points d'ordre  $n$  de la forme suivante

$$\begin{cases} L_n y(t) = \sigma(t), \quad t \in I \equiv [a, b], \\ U_i(y) = \gamma_i, \quad i = 1, \dots, m, \end{cases} \quad (2.2.1)$$

où

$$L_n y(t) \equiv a_0(t)y^{(n)}(t) + a_1(t)y^{(n-1)}(t) + \dots + a_{n-1}(t)y'(t) + a_n(t)y(t), \quad t \in I,$$

et

$$U_i(y) = \sum_{j=0}^{n-1} (\alpha_j^i y^{(j)}(a) + \beta_j^i y^{(j)}(b)), \quad i = 1, \dots, m, \quad m \leq n, \quad (2.2.2)$$

où  $\alpha_j^i, \beta_j^i$  sont des constantes réelles pour  $i = 1, \dots, m, j = 0, \dots, n-1$ ,  $\sigma$  et  $a_k$  sont des fonctions continues sur  $I$  pour  $k = 0, \dots, n$ , et  $a_0(t) \neq 0$  pour  $t \in I$ .

**Remarques 2.2.1** 1. L'ensemble des solutions de l'équation linéaire  $L_n y(t) = 0$  est un sous-espace vectoriel de dimension  $n$  de  $C^n(I)$ , n'importe quelle base d'un tel sous-espace s'appellera un ensemble fondamental de solutions de l'équation  $L_n y(t) = 0$ .

2. Le problème homogène

$$\begin{cases} L_n y(t) = 0, \quad t \in I, \\ U_i(y) = 0, \quad i = 1, \dots, m. \end{cases} \quad (2.2.3)$$

est dit  $k$ -compatible,  $0 \leq k \leq n$  si son ensemble de solutions est de dimension  $k$ .

**Proposition 2.2.1** [1], [3] Soit  $\{y_1, \dots, y_n\}$  l'ensemble fondamental de solutions de  $L_n y(t) =$

0. Le problème (2.2.3) est  $k$ -compatible si et seulement si

$$\text{rang} \begin{pmatrix} U_1(y_1) & \cdots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_m(y_1) & \cdots & U_m(y_n) \end{pmatrix} = n - k.$$

**Corollaire 2.2.1** Le problème homogène (2.2.3) admet seulement la solution triviale si et seulement si,

$$\text{rang} \begin{pmatrix} U_1(y_1) & \cdots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_m(y_1) & \cdots & U_m(y_n) \end{pmatrix} = n.$$

En particulier le nombre des conditions aux bords  $m$  doit être égal à  $n$  (l'ordre de l'équation).

**Exemple 2.2.1** Soit le problème aux limites suivant :

$$(F) \begin{cases} y'' + y = 0, & 0 < t < 1, \\ y(0) = y(1) = 0. \end{cases}$$

Ici  $n = m = 2$ . L'équation du problème (F) a pour solution générale

$$y(t) = k_1 \cos(t) + k_2 \sin(t), \quad k_1, k_2 \in \mathbb{R}.$$

Donc, d'une part, on a  $\begin{cases} y(0) = 0 \\ y(1) = 0 \end{cases} \Rightarrow k_1 = k_2 = 0$ , d'où le problème (F) admet seulement la solution triviale  $y = 0$ .

$$D'autre part, on a \begin{cases} y_1(t) = \cos(t) \\ y_2(t) = \sin(t) \end{cases} \quad \text{et} \quad \begin{cases} U_1(y) = y(0) \\ U_2(y) = y(1). \end{cases} \quad \text{Alors} \quad \begin{cases} U_1(y_1) = 1 \\ U_1(y_2) = 0 \\ U_2(y_1) = \cos(1) \\ U_2(y_2) = \sin(1) \end{cases}$$

et donc

$$\text{rang} \begin{pmatrix} U_1(y_1) & U_1(y_2) \\ U_2(y_1) & U_2(y_2) \end{pmatrix} = \text{rang} \begin{pmatrix} 1 & 0 \\ \cos(1) & \sin(1) \end{pmatrix} = 2 = n.$$

**Proposition 2.2.2** [1] Soit  $\{y_1, \dots, y_n\}$  l'ensemble fondamental de solutions de  $L_n y(t) = 0$  et soit  $y_p$  une solution arbitraire de l'équation non homogène  $L_n y(t) = \sigma(t)$ . Alors, toute translation de l'ensemble de solutions du problème (2.2.1) à l'origine est de dimension égale à  $k$  si et seulement si

$$\text{rang} \begin{pmatrix} U_1(y_1) & \cdots & U_1(y_n) \\ \vdots & \ddots & \vdots \\ U_m(y_1) & \cdots & U_m(y_n) \end{pmatrix} = \text{rang} \left( \begin{array}{ccc|c} U_1(y_1) & \cdots & U_1(y_n) & \gamma_1 - U_1(y_p) \\ & & \ddots & \vdots \\ U_m(y_1) & \cdots & U_m(y_n) & \gamma_n - U_m(y_p) \end{array} \right) = n - k.$$

Comme conséquence de la proposition 2.2.2 et du corollaire 2.2.1 nous obtenons le célèbre théorème de Fredholm.

**Théorème 2.2.1 (Alternative de Fredholm)** Le problème non homogène

$$L_n y(t) = \sigma(t), \quad t \in I, \quad U_i(y) = \gamma_i, \quad i = 1, \dots, n, \quad (2.2.4)$$

où le nombre de conditions aux bords est égal à l'ordre de l'équation, admet une solution unique si et seulement si, le problème homogène

$$L_n y(t) = 0, \quad t \in I, \quad U_i(y) = 0, \quad i = 1, \dots, n \quad (2.2.5)$$

admet comme seule solution la solution triviale ( $y \equiv 0$ ).

**Preuve.** L'ensemble de solutions du problème non homogène est de dimension  $k$  si et seulement si, le rang des deux matrices données dans la proposition 2.2.2 est  $n - k$ . En particulier il existe une seule solution si le rang de ces deux matrices est égal à  $n$ , ce qui est équivalent, d'après le corollaire 2.2.1, à l'unicité de la solution triviale du problème homogène. ■

Maintenant nous présentons la définition axiomatique de la fonction de Green associée au problème homogène (2.2.3).

**Définition 2.2.1** Une fonction  $G$  est appelée fonction de Green du problème (2.2.3) si elle satisfait les six propriétés suivantes :

1.  $G$  est définie sur le carré  $I \times I$ .



2. Pour  $k = 0, 1, \dots, n-2$ ,  $G$  est continue et possède des dérivées partielles  $\frac{\partial^k G}{\partial t^k}$  continues sur le carré  $I \times I$ .
3.  $\frac{\partial^{n-1} G}{\partial t^{n-1}}$  et  $\frac{\partial^n G}{\partial t^n}$  existent et elles sont continues sur les triangles  $a \leq s < t \leq b$  et  $a \leq t < s \leq b$ .
4. Pour chaque  $t \in (a, b)$  les limites  $\frac{\partial^{n-1} G}{\partial t^{n-1}}(t^+, t)$  et  $\frac{\partial^{n-1} G}{\partial t^{n-1}}(t^-, t)$  existent (i.e., les limites de  $\frac{\partial^{n-1} G}{\partial t^{n-1}}(t, s)$  quand  $(t, s) \mapsto (t, t)$  avec  $s > t$  ou  $s < t$ ) et de plus

$$\frac{\partial^{n-1} G}{\partial t^{n-1}}(t, t^+) - \frac{\partial^{n-1} G}{\partial t^{n-1}}(t, t^-) = -\frac{1}{a_0(t)}.$$

5. Pour chaque  $s \in (a, b)$ , la fonction  $t \mapsto G(t, s)$  est solution de l'équation différentielle  $L_n y = 0$ , pour  $t \in [a, s)$  et  $t \in (s, b]$ , c'est à dire

$$a_0(t) \frac{\partial^n G}{\partial t^n}(t, s) + a_1(t) \frac{\partial^{n-1} G}{\partial t^{n-1}}(t, s) + \dots + a_{n-1}(t) \frac{\partial G}{\partial t}(t, s) + a_n(t) G(t, s) = 0,$$

sur les deux intervalles.

6. Pour chaque  $s \in (a, b)$ , la fonction  $t \mapsto G(t, s)$  vérifie les conditions aux bords  $U_i(G(\cdot, s)) = 0$ ,  $i = 1, \dots, m$ , c'est à dire

$$\sum_{j=0}^{n-1} (\alpha_j^i \frac{\partial^j G}{\partial t^j}(a, s) + \beta_j^i \frac{\partial^j G}{\partial t^j}(b, s)), \quad i = 1, \dots, m.$$

L'importance principale de la définition précédente est que l'opérateur intégral, dont le noyau est la fonction de Green, nous donne la solution du semi-problème homogène

$$\begin{cases} L_n y(t) = \sigma(t), & t \in I, \\ U_i(y) = 0, & i = 1, \dots, m. \end{cases} \quad (2.2.6)$$

**Théorème 2.2.2** [1] Soit  $G$  la fonction de Green du problème homogène (2.2.3). Alors, pour toute fonction continue  $\sigma$ ,

$$y(t) = \int_a^b G(t, s) \sigma(s) ds, \quad t \in I \quad (2.2.7)$$

est une solution du problème semi-homogène (2.2.6).

**Preuve.** Démontrons le théorème pour  $n = 2$  et  $m = 1$ . Soit la fonction  $y$  définie par:

$$y(t) = \int_a^b G(t, s)\sigma(s)ds = \int_a^t G(t, s)\sigma(s)ds + \int_t^b G(t, s)\sigma(s)ds.$$

1.  $y$  est solution de l'équation  $L_2y(t) = \sigma(t)$ ,  $t \in [a, b]$ . En effet, grâce à la dérivabilité de  $G$  par rapport à  $t$  dans chacun des intervalles et de la relation suivante :

$$\frac{\partial}{\partial t} \left( \int_{u(t)}^{v(t)} f(t, s)ds \right) = v'(t)f(t, t) - u'(t)f(t, t) + \int_{u(t)}^{v(t)} \frac{\partial f(t, s)}{\partial t} ds,$$

on aura

$$\begin{aligned} y'(t) &= G(t, t)\sigma(t) + \int_a^t \frac{\partial G}{\partial t}(t, s)\sigma(s)ds - G(t, t)\sigma(t) + \int_t^b \frac{\partial G}{\partial t}(t, s)\sigma(s)ds \\ &= \int_a^b \frac{\partial G}{\partial t}(t, s)\sigma(s)ds. \end{aligned}$$

Calculons  $y''$

$$\begin{aligned} y''(t) &= \int_a^t \frac{\partial^2 G}{\partial t^2}(t, s)\sigma(s)ds + \sigma(t)\frac{\partial G}{\partial t}(t, t^-) + \int_t^b \frac{\partial^2 G}{\partial t^2}(t, s)\sigma(s)ds - \sigma(t)\frac{\partial G}{\partial t}(t, t^+) \\ &= \int_a^b \frac{\partial^2 G}{\partial t^2}(t, s)\sigma(s)ds - \sigma(t)\left[ \frac{\partial G}{\partial t}(t, t^+) - \frac{\partial G}{\partial t}(t, t^-) \right]. \end{aligned}$$

Or  $\frac{\partial G}{\partial t}(t^+, t) - \frac{\partial G}{\partial t}(t^-, t) = -\frac{1}{a_0(t)}$ , (propriété 4 de  $G$ ), on en déduit l'expression

$$y''(t) = \int_a^b \frac{\partial^2 G}{\partial t^2}(t, s)\sigma(s)ds + \frac{\sigma(t)}{a_0(t)},$$

par suite,

$$\begin{aligned} &a_0(t)y''(t) + a_1(t)y'(t) + a_2(t)y(t) \\ &= \int_a^b \left[ a_0(t)\frac{\partial^2 G}{\partial t^2}(t, s) + a_1(t)\frac{\partial G}{\partial t}(t, s) + a_2(t)G(t, s) \right] \sigma(s)ds + \sigma(t) \\ &= \sigma(t) \quad (\text{car la fonction } t \mapsto G(t, s) \text{ est solution de l'équation } L_2y(t) = 0 \text{ pour tout } t \neq s). \end{aligned}$$

2. La fonction  $y$  vérifie les conditions aux bords homogène. En effet,

$$\begin{aligned} U_1(y) &= \alpha_0^1 y(a) + \beta_0^1 y(b) + \alpha_1^1 y'(a) + \beta_1^1 y'(b) \\ &= \int_a^b [\alpha_0^1 G(a, s) + \beta_0^1 G(b, s) + \alpha_1^1 \frac{\partial G}{\partial t}(a, s) + \beta_1^1 \frac{\partial G}{\partial t}(b, s)] \sigma(s)ds \\ &= \int_a^b U_1(G(\cdot, s)) \sigma(s)ds \\ &= 0 \quad (\text{car } U_1(G(\cdot, s)) = 0). \end{aligned}$$

■

**Théorème 2.2.3** [1] (**Existence et unicité de la fonction de Green**) *Supposons que le problème homogène (2.2.5) a seulement la solution triviale. Alors il existe une unique fonction de Green  $G$ , associée à (2.2.5). De plus, pour toute fonction continue  $\sigma$ , la solution unique du problème semi-homogène*

$$\begin{cases} L_n y(t) = \sigma(t), & t \in I, \\ U_i(y) = 0, & i = 1, \dots, n \end{cases} \quad (2.2.8)$$

est donnée par l'expression (2.2.7).

**Preuve.** D'abord nous montrerons qu'une fonction de Green associée au problème de Cauchy

$$\begin{cases} L_n y(t) = 0, & t \in I, \\ y^{(i)}(a) = 0, & i = 0, \dots, n-1 \end{cases} \quad (2.2.9)$$

est donnée par

$$\tilde{K}(t, s) = \begin{cases} K(t, s), & \text{si } a \leq s \leq t, \\ 0, & \text{si } t < s \leq b, \end{cases}$$

avec

$$K(t, s) := \frac{\begin{vmatrix} y_1(s) & \dots & y_n(s) \\ y_1'(s) & \dots & y_n'(s) \\ \vdots & \ddots & \vdots \\ y_1^{(n-2)}(s) & \dots & y_n^{(n-2)}(s) \\ y_1(t) & \dots & y_n(t) \end{vmatrix}}{a_0(s)W(y_1, \dots, y_n)(s)}.$$

Ici  $\{y_1, \dots, y_n\}$  est l'ensemble fondamental de solutions de l'équation  $L_n y(t) = 0$ , et

$$W(y_1, \dots, y_n)(s) = \begin{vmatrix} y_1(s) & \dots & y_n(s) \\ y_1'(s) & \dots & y_n'(s) \\ \vdots & \ddots & \vdots \\ y_1^{(n-1)}(s) & \dots & y_n^{(n-1)}(s) \end{vmatrix}$$

est son Wronskien correspondant.

La fonction  $\tilde{K}$  satisfait les conditions **(1)-(6)**, de la définition (2.2.1), pour le problème (2.2.9). En effet; pour  $k = 0, 1, \dots, n-2$ , les dérivées partielles  $\frac{\partial^k \tilde{K}}{\partial t^k}$  existent et elles sont

données par :

$$\frac{\partial^k \tilde{K}}{\partial t^k}(t, s) = \begin{cases} \frac{\begin{vmatrix} y_1(s) & \dots & y_n(s) \\ y_1'(s) & \dots & y_n'(s) \\ \vdots & \ddots & \vdots \\ y_1^{(n-2)}(s) & \dots & y_n^{(n-2)}(s) \\ y_1^{(k)}(t) & \dots & y_n^{(k)}(t) \end{vmatrix}}{a_0(s)W(y_1, \dots, y_n)(s)}, & \text{si } a \leq s \leq t, \\ 0, & \text{si } t < s \leq b, \end{cases}$$

ainsi, elles sont continues sur  $I \times I$  pour tout  $n \geq 2$  (notons que si  $s = t$  alors deux lignes sont égales dans le déterminant). Par conséquent, la condition **(2)** de la définition (2.2.1) est satisfaite.

La même formule est également valide sur les deux triangles pour les dérivées partielles

$$\frac{\partial^{n-1} \tilde{K}}{\partial t^{n-1}}(t, s) \text{ et } \frac{\partial^n \tilde{K}}{\partial t^n}(t, s),$$

donc la condition **(3)** est également satisfaite. De plus on a

$$\frac{\partial^{n-1} \tilde{K}}{\partial t^{n-1}}(t, t^+) = 0$$

et

$$\frac{\partial^{n-1} \tilde{K}}{\partial t^{n-1}}(t, t^-) = \frac{\partial^{n-1} K}{\partial t^{n-1}}(t, t^-) = \frac{\begin{vmatrix} y_1(t) & \dots & y_n(t) \\ y_1'(t) & \dots & y_n'(t) \\ \vdots & \ddots & \vdots \\ y_1^{(n-2)}(t) & \dots & y_n^{(n-2)}(t) \\ y_1^{(n-1)}(t) & \dots & y_n^{(n-1)}(t) \end{vmatrix}}{a_0(t)W(y_1, \dots, y_n)(t)} = \frac{1}{a_0(t)}.$$

Donc la condition **(4)** est vérifiée.

Sur le premier triangle,  $a \leq s < t$  pour  $s$  fixé,  $\tilde{K}(t, s)$  est une combinaison linéaire de  $y_1, \dots, y_n$ , ce qui entraîne que la condition **(5)** est satisfaite.

Constatons que, pour tout  $s \in (a, b)$ , on a

$$\frac{\partial^k \tilde{K}}{\partial t^k}(a, s) = 0 \text{ pour tout } k = 0, \dots, n-1.$$

Donc la condition **(6)** est vérifiée.

Maintenant, afin de construire une fonction de Green  $G$  pour le problème (2.2.5), considérons  $n$  fonctions continues sur  $I$ , notées  $c_1, \dots, c_n$ , et cherchons une fonction  $G$  sous la forme

$$G(t, s) = \tilde{K}(t, s) + c_1(s)y_1(t) + \dots + c_n(s)y_n(t).$$

Il est facile de vérifier que la fonction  $G$  satisfait les conditions **(1)-(5)**. Il reste seulement à prouver qu'il est possible de choisir  $c_1, \dots, c_n$ , telles que la condition **(6)** soit satisfaite par  $G$  pour le problème (2.2.5), c'est à dire, pour chaque  $s \in (a, b)$  nous devons vérifier que

$$U_i(G(., s)) = 0, \quad i = 1, \dots, n, \quad \forall s \in I.$$

Par linéarité, nous avons

$$U_i(G(., s)) = U_i(\tilde{K}(., s)) + \sum_{j=1}^n c_j(s) U_i(y_j), \quad i = 1, \dots, n,$$

ce qui entraîne que  $(c_1(s), \dots, c_n(s))$  est une solution du système linéaire

$$\begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ & \ddots & \\ U_n(y_1) & \dots & U_n(y_n) \end{pmatrix} \begin{pmatrix} c_1(s) \\ \vdots \\ c_n(s) \end{pmatrix} = - \begin{pmatrix} U_1(\tilde{K}(., s)) \\ \vdots \\ U_n(\tilde{K}(., s)) \end{pmatrix}.$$

Puisque le problème homogène (2.2.5) a seulement la solution triviale, le corollaire (2.2.1) assure que le rang de la matrice précédente est égal à  $n$ . Donc, la seule solution de ce système est donnée par

$$\begin{pmatrix} c_1(s) \\ \vdots \\ c_n(s) \end{pmatrix} = - \begin{pmatrix} U_1(y_1) & \dots & U_1(y_n) \\ & \ddots & \\ U_n(y_1) & \dots & U_n(y_n) \end{pmatrix}^{-1} \begin{pmatrix} U_1(\tilde{K}(., s)) \\ \vdots \\ U_n(\tilde{K}(., s)) \end{pmatrix}.$$

De plus, de cette expression, nous savons que les fonctions  $c_1, \dots, c_n$  sont continues et par conséquent,  $G$  est une fonction de Green pour le problème (2.2.5). Du théorème (2.2.2) nous avons

$$y(t) = \int_a^b G(t, s)\sigma(s)ds, \quad t \in I$$

### 2.3. Calcul de la fonction de Green associée à des E.D.O. linéaires à coefficients constants

est une solution du problème (2.2.8) et de plus, d'après l'alternative de Fredholm (voir le théorème (2.2.1)), cette solution est unique.

Maintenant, pour prouver l'unicité de la fonction de Green, nous supposons que  $H$  est une autre fonction vérifiant les conditions **(1)-(6)** pour le problème (2.2.5), puis pour tout  $t \in I$  et toute fonction continue  $\sigma$ , on aura

$$\int_a^b G(t, s)\sigma(s)ds = \int_a^b H(t, s)\sigma(s)ds, \quad t \in I$$

alors  $\int_a^b [G(t, s) - H(t, s)]\sigma(s)ds = 0, \quad t \in I$ , ce qui entraîne que

$$G(t, s) = H(t, s)$$

pour tout  $t, s \in I$ , si  $n \geq 2$ , et pour tout  $t \neq s$  si  $n = 1$ , parce que dans ce dernier cas la fonction de Green est discontinue sur la diagonale. ■

## 2.3 Calcul de la fonction de Green associée à des E.D.O. linéaires à coefficients constants

Dans cette section nous traitons le problème aux limites à deux-points

$$\begin{cases} \mathcal{L}_n y(t) = \sigma(t), & t \in I, \\ U_i(y) = 0, & i = 1, \dots, n, \end{cases} \quad (2.3.1)$$

où

$$\mathcal{L}_n y(t) = y^{(n)}(t) + a_1 y^{(n-1)}(t) + \dots + a_{n-1} y'(t) + a_n y(t), \quad t \in I,$$

avec  $a_1, \dots, a_n \in \mathbb{R}$ . De plus nous supposons que les fonctions linéaires  $U_i$  définies par (2.2.2), avec  $a_i, \alpha_j^i, \beta_j^i \in \mathbb{R}$  pour  $i = 1, \dots, n$  et  $j = 1, \dots, n-1$ , sont telles que le problème homogène associé :

$$\begin{cases} \mathcal{L}_n y(t) = 0, & t \in I, \\ U_i(y) = 0, & i = 1, \dots, n \end{cases} \quad (2.3.2)$$

a seulement la solution triviale.

Remarquons que ces deux problèmes sont des cas particuliers de (2.2.8) et de (2.2.5) respectivement. Nous présentons une méthode pour calculer explicitement la fonction de Green, une telle méthode nous permettra d'obtenir son expression en résolvant un système algébrique linéaire de  $n$  équations à  $n$  inconnues.

**Théorème 2.3.1** [1] *Soit  $r$  l'unique solution du problème de Cauchy*

$$\begin{cases} u^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} u^{(i)}(t) = 0, & t \in \mathbb{R}_+ \\ u^{(i)}(0) = 0, & i = 0, \dots, n-2, \\ u^{(n-1)}(0) = 1. \end{cases} \quad (2.3.3)$$

*Alors, l'unique solution du problème de Cauchy non homogène*

$$\begin{cases} y^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} y^{(i)}(t) = \sigma(t), & t \in I \\ y^{(i)}(a) = \lambda_i, & i = 0, \dots, n-1, \end{cases} \quad (2.3.4)$$

avec  $\sigma \in C(I)$  et  $\lambda_i \in \mathbb{R}, i = 0, \dots, n-1$ , est donnée par :

$$y(t) = \int_a^t r(t-s) \sigma(s) ds + \sum_{k=0}^{n-1} \lambda_k y_k(t) \quad (2.3.5)$$

où

$$y_k(t) = r^{(n-k-1)}(t-a) + \sum_{j=k+1}^{n-1} a_{n-j} r^{(j-k-1)}(t-a), \quad t \in \mathbb{R}, \quad k = 0, \dots, n-1. \quad (2.3.6)$$

**Preuve.** D'une part, soit la fonction  $v$  définie par :

$$v(t) = \int_a^t r(t-s) \sigma(s) ds, \quad t \in I.$$

D'après la règle du Leibniz de différentiation <sup>(1)</sup>, nous obtenons

$$v^{(i)}(t) = \int_a^t r^{(i)}(t-s) \sigma(s) ds, \quad t \in I, \quad i = 0, \dots, n-1,$$

et

$$v^{(n)}(t) = \int_a^t r^{(n)}(t-s) \sigma(s) ds + \sigma(t), \quad t \in I.$$

---

<sup>(1)</sup>  $(fg)^{(n)} = \sum_{k=0}^n C_n^k f^{(k)} g^{(n-k)}$

Par conséquent,

$$v^{(i)}(a) = 0, \quad \forall i = 0, \dots, n-1$$

et, pour tout  $t \in I$ ,

$$v^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} v^{(i)}(t) = \int_a^t \left[ r^{(n)}(t-s) + \sum_{i=0}^{n-1} a_{n-i} r^{(i)}(t-s) \right] \sigma(s) ds + \sigma(t) = \sigma(t).$$

C'est à dire  $v$  est l'unique solution du problème de Cauchy semi-homogène

$$\begin{cases} v^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} v^{(i)}(t) = \sigma(t), & t \in I, \\ v^{(i)}(a) = 0, & i = 0, \dots, n-1. \end{cases}$$

D'autre part, on a

$$\begin{cases} r^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} r^{(i)}(t) = 0, & t \in \mathbb{R}_+ \\ r(0) = r'(0) = \dots = r^{(n-2)}(0) = 0 \\ r^{(n-1)}(0) = 1 \end{cases} \implies \begin{cases} (r')^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} (r')^{(i)}(t) = 0, & t \in \mathbb{R}_+ \\ r'(0) = \dots = r^{(n-2)}(0) = 0 \\ r^{(n-1)}(0) = 1 \\ r^{(n)}(0) = -a_1. \end{cases}$$

Donc la fonction

$$t \longmapsto y_{n-2}(t) = r'(t-a) + a_1 r(t-a)$$

est l'unique solution du problème de Cauchy posé sur l'intervalle  $I$

$$\begin{cases} y^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} y^{(i)}(t) = 0, & t \in I, \\ y^{(i)}(a) = 0, & i = 0, \dots, n-1, i \neq n-2, \\ y^{(n-2)}(a) = 1. \end{cases}$$

De façon analogue, nous pouvons prouver que pour tout  $j \in \{0, 1, \dots, n-1\}$ , la fonction  $y_{n-j-1}$  définie par  $y_{n-j-1}(t) = r^{(j)}(t-a) + \sum_{k=n-j}^{n-1} a_{n-k} y^{(k)}(t)$  est l'unique solution du problème

$$\begin{cases} y^{(n)}(t) + \sum_{i=0}^{n-1} a_{n-i} y^{(i)}(t) = 0, & t \in I, \\ y^{(i)}(a) = 0, & i = 0, \dots, n-1, i \neq n-j-1 \\ y^{(n-j-1)}(a) = 1. \end{cases} \quad (2.3.7)$$

Par conséquent,

$$u(t) = v(t) + \sum_{i=0}^{n-1} \lambda_i y_i(t), \quad t \in I$$



est l'unique solution du problème de Cauchy non homogène (2.3.4). ■

Du résultat précédent, nous savons que l'unique solution du problème de Cauchy (2.3.4), posé sur l'intervalle  $[a, b]$ , est donnée par l'expression (2.3.5), où  $r$  est la solution unique du problème de Cauchy (2.3.3), les fonctions  $y_k$  vérifiant (2.3.6) et  $\lambda_k$  sont des paramètres réels donnés.

D'une façon analogue à la preuve du théorème (2.2.3), si nous considérons le problème aux bords (2.3.1), nous chercherons une fonction de Green de la forme

$$G(t, s) = \begin{cases} r(t - s) + \sum_{k=0}^{n-1} y_k(t) d_k(s), & \text{si } a \leq s \leq t \leq b \\ \sum_{k=0}^{n-1} y_k(t) d_k(s), & \text{si } a \leq t < s \leq b \end{cases} \quad (2.3.8)$$

où les fonctions réelles continues  $d_k$  sont les inconnues. De (2.2.7) et (2.3.8), nous obtenons

$$\begin{aligned} 0 &= \sum_{j=0}^{n-1} (\alpha_j^i y^{(j)}(a) + \beta_j^i y^{(j)}(b)) \\ &= \sum_{j=0}^{n-1} \left[ \alpha_j^i \int_a^b \sum_{k=0}^{n-1} y_k^{(j)}(a) d_k(s) \sigma(t) d(s) + \beta_j^i \int_a^b r^{(j)}(b - s) \sigma(s) ds + \beta_j^i \int_a^b \sum_{k=0}^{n-1} y_k^{(j)}(a) d_k(s) \sigma(t) ds \right] \\ &= \sum_{j=0}^{n-1} \left[ \beta_j^i \int_a^b r^{(j)}(b - s) \sigma(s) ds \right] + \sum_{j=0}^{n-1} \int_a^b d_k(s) \left[ \alpha_j^i \sum_{k=0}^{n-1} y_k^{(j)}(a) + \beta_j^i \sum_{k=0}^{n-1} y_k^{(j)}(b) \right] \sigma(t) ds \\ &= \int_a^b \left[ \sum_{j=0}^{n-1} \beta_j^i r^{(j)}(b - s) + \sum_{k=0}^{n-1} d_k(s) U_i(y_k) \right] \sigma(t) ds. \end{aligned}$$

Puisque  $r, y_k$  et  $U_i, i, k = 1, \dots, n$  sont obtenues précédemment, en résolvant le système linéaire

$$\sum_{j=0}^{n-1} d_k(s) U_i(y_k) = - \sum_{j=0}^{n-1} \beta_j^i r^{(j)}(b - s), \quad i = 1, \dots, n, \quad (2.3.9)$$

nous obtenons l'expression de  $d_k(s)$  et donc la formule de  $G(t, s)$ .

Notons que le système (2.3.9) est équivalent à

$$\begin{pmatrix} U_1(y_0) & \cdots & U_1(y_{n-1}) \\ \vdots & \ddots & \vdots \\ U_n(y_0) & \cdots & U_n(y_{n-1}) \end{pmatrix} \begin{pmatrix} d_0(s) \\ \vdots \\ d_{n-1}(s) \end{pmatrix} = - \begin{pmatrix} \sum_{j=0}^{n-1} \beta_j^1 r^{(j)}(b - s) \\ \vdots \\ \sum_{j=0}^{n-1} \beta_j^n r^{(j)}(b - s) \end{pmatrix}. \quad (2.3.10)$$

### 2.3. Calcul de la fonction de Green associée à des E.D.O. linéaires à coefficients constants

Puisque le problème (2.3.2) a seulement la solution triviale, le théorème (2.2.1) et le corollaire (2.2.1) assurent que le système précédent admet une et une seule solution.

**Exemple 2.3.1** On considère le problème aux limites du second ordre

$$\begin{cases} y'' = \sigma(t), t \in [0, 1] \\ y(0) = 0, y(1) = 0. \end{cases} \quad (2.3.11)$$

Ici

$$\mathcal{L}_2 y(t) = y''(t) \quad (a_1 = 0, a_2 = 0)$$

$$U_1(y) = y(0) \quad (\alpha_0^1 = 1, \alpha_1^1 = 0, \beta_0^1 = 0, \beta_1^1 = 0)$$

$$U_2(y) = y(1) \quad (\alpha_0^2 = 0, \alpha_1^2 = 0, \beta_0^2 = 1, \beta_1^2 = 0).$$

La fonction de Green associée au problème homogène

$$\begin{cases} y'' = 0, t \in [0, 1] \\ y(0) = 0, y(1) = 0, \end{cases}$$

s'écrit sous la forme

$$G(t, s) = \begin{cases} r(t-s) + y_0(t)d_0(s) + y_1(t)d_1(s), & \text{si } 0 \leq s \leq t \leq 1, \\ y_0(t)d_0(s) + y_1(t)d_1(s), & \text{si } 0 \leq t < s \leq 1, \end{cases}$$

où  $r$  vérifie le problème de Cauchy

$$\begin{cases} r''(t) = 0, t > 0 \\ r(0) = 0 \\ r'(0) = 1. \end{cases}$$

On trouve  $y_0(t) = t$ ,  $y_1(t) = 1$  et  $r(t) = t$ . Par suite, les fonctions continues  $d_0$  et  $d_1$  vérifient le système linéaire

$$\begin{pmatrix} U_1(y_0) & U_1(y_1) \\ U_2(y_0) & U_2(y_1) \end{pmatrix} \begin{pmatrix} d_0(s) \\ d_1(s) \end{pmatrix} = - \begin{pmatrix} 0 \\ 1-s \end{pmatrix},$$

qui est équivalent à

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} d_0(s) \\ d_1(s) \end{pmatrix} = - \begin{pmatrix} 0 \\ 1-s \end{pmatrix}.$$

Alors

$$\begin{cases} d_0(s) = -(1-s) \\ d_1(s) = 0. \end{cases}$$

D'où

$$G(t, s) = \begin{cases} s(t-1), & \text{si } 0 \leq s \leq t \leq 1, \\ t(s-1), & \text{si } 0 \leq t < s \leq 1. \end{cases}$$

**Exemple 2.3.2** Soit le problème aux limites associé à une équation d'ordre 4 :

$$\begin{cases} y^{(4)} = 0, \\ y(0) = 0 \\ y'(0) = 0 \\ y(1) = 0 \\ y'(1) = 0. \end{cases} \quad (2.3.12)$$

Toute solution de (E) est de la forme  $y(t) = A + Bt + Ct^2 + Dt^3$ . Donc

$$\begin{cases} y_0(t) = 1 \\ y_1(t) = t \\ y_2(t) = t^2 \\ y_3(t) = t^3. \end{cases}$$

Et

$$\begin{cases} \mathcal{L}_4 y(t) = y^{(4)}(t), \quad t \in [0, 1] \quad (a_0 = 1 \text{ et les autres coefficients sont nuls}) \\ U_1(y) = y(0) \quad (\alpha_0^1 = 1 \text{ et les autres coefficients sont nuls}) \\ U_2(y) = y'(0) \quad (\alpha_1^2 = 1 \text{ et les autres coefficients sont nuls}) \\ U_3(y) = y(1) \quad (\beta_0^3 = 1 \text{ et les autres coefficients sont nuls}) \\ U_4(y) = y'(1) \quad (\beta_1^4 = 1 \text{ et les autres coefficients sont nuls}) \end{cases}$$

Nous cherchons la fonction de Green sous la forme

$$G(t, s) = \begin{cases} r(t-s) + \sum_{k=0}^3 y_k(t) d_k(s), & \text{si } s \leq t \leq 1 \\ \sum_{k=0}^3 y_k(t) d_k(s), & \text{si } 0 \leq t < s \leq 1 \end{cases}$$

### 2.3. Calcul de la fonction de Green associée à des E.D.O. linéaires à coefficients constants

où  $r(t) = \frac{1}{6}t^3$  est l'unique solution du problème de Cauchy

$$\begin{cases} u^{(4)}(t) = 0, & t > 0 \\ u(0) = 0 \\ u'(0) = 0 \\ u''(0) = 0 \\ u^{(3)}(0) = 1. \end{cases}$$

Il nous reste à déterminer les fonctions continues  $d_k$  pour  $k = 0, \dots, 3$  qui vérifient le système linéaire

$$\begin{pmatrix} U_1(y_0) & U_1(y_1) & U_1(y_2) & U_1(y_3) \\ U_2(y_0) & U_2(y_1) & U_2(y_2) & U_2(y_3) \\ U_3(y_0) & U_3(y_1) & U_3(y_2) & U_3(y_3) \\ U_4(y_0) & U_4(y_1) & U_4(y_2) & U_4(y_3) \end{pmatrix} \begin{pmatrix} d_0(s) \\ d_1(s) \\ d_2(s) \\ d_3(s) \end{pmatrix} = - \begin{pmatrix} \sum_{j=0}^3 \beta_j^1 r^{(j)}(1-s) \\ \sum_{j=0}^3 \beta_j^2 r^{(j)}(1-s) \\ \sum_{j=0}^3 \beta_j^3 r^{(j)}(1-s) \\ \sum_{j=0}^3 \beta_j^4 r^{(j)}(1-s) \end{pmatrix}.$$

Ce système est équivalent à

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} d_0(s) \\ d_1(s) \\ d_2(s) \\ d_3(s) \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ r(1-s) \\ r'(1-s) \end{pmatrix},$$

dont la résolution donne

$$\begin{cases} d_0(s) = 0, \\ d_1(s) = 0, \\ d_2(s) = -\frac{1}{2}(1-s)^3 + \frac{1}{2}(1-s)^2, \\ d_3(s) = \frac{1}{3}(1-s)^3 - \frac{1}{2}(1-s)^2. \end{cases}$$

D'où

$$G(t, s) = \begin{cases} (\frac{1}{2}t^3 - t^2 + \frac{1}{2}t)s^2 - (\frac{t^3}{3} - \frac{t^2}{2} + \frac{1}{6})s^3 & \text{si } 0 \leq s \leq t \leq 1, \\ (\frac{1}{2}s^3 - s^2 + \frac{1}{2}s)t^2 - (\frac{s^3}{3} - \frac{s^2}{2} + \frac{1}{6})t^3 & \text{si } 0 \leq t \leq s \leq 1. \end{cases}$$

### 2.3.1 Problèmes périodiques

Comme nous l'avons vu dans la preuve du theoreme (2.3.1), l'obtention de l'expression de la fonction de Green associée au problème aux limites (2.3.2) est réduite à la résolution du système algébrique (2.3.10) et du problème (2.3.3). De plus, l'existence et l'unicité d'une telle fonction est équivalente à l'unicité de la solution du système algébrique correspondant.

À partir de la définition des fonctions  $y_k$ ,  $k = 1, \dots, n$ , données dans (2.3.6), nous devons seulement calculer la fonction  $r$  comme solution unique du problème de Cauchy (2.3.3). Dans certains cas, dépendant de l'ordre de l'équation et des paramètres non nuls intervenant dans l'opérateur  $\mathcal{L}_n$  et les fonctions  $U_i$ , les expressions des fonctions continues  $d_k$  peuvent être très compliquées. Pour cette raison, dans certains cas particuliers, par rapport aux conditions aux bords, on peut trouver des moyens plus appropriés pour calculer la fonction de Green. Considérons le cas des conditions aux bords périodiques :

$$U_i(y) \equiv y^{(i)}(b) - y^{(i)}(a), \quad i = 0, \dots, n-1.$$

Dans ce cas, l'expression de la fonction de Green est donnée dans le lemme 2.3.1. Ce lemme a été démontré dans [[2], Lemme 2.1] sur l'intervalle particulier  $[0, 2\pi]$ .

**Lemme 2.3.1** [1] *Supposons que le problème périodique linéaire :*

$$\begin{cases} \mathcal{L}_n u(t) = \sigma(t), & t \in I, \\ u^{(i)}(a) - u^{(i)}(b) = \mu_i, & i = 0, \dots, n-1 \end{cases}$$

*ait une solution unique  $u \in C^n(I)$  pour tout  $\sigma \in C(I)$ . Alors  $u$  est donné par :*

$$u(t) = \int_a^b G(t, s) \sigma(s) ds + \sum_{i=0}^{n-1} r_i(t) \mu_i.$$

*Ici  $r_j \in C^\infty(\mathbb{R})$ ,  $j = 0, 1, \dots, n-1$ , est la solution unique du problème linéaire*

$$\begin{cases} \mathcal{L}_n z(t) = 0, & t \in I, \\ z^{(i)}(a) - z^{(i)}(b) = 0, & i = 0, \dots, n-1; \quad i \neq j, \\ z^{(j)}(a) - z^{(j)}(b) = 1 \end{cases} \quad (2.3.13)$$

et

$$G(t, s) = \begin{cases} r_{n-1}(a + t - s), & \text{si } a \leq s \leq t \leq b \\ r_{n-1}(b + t - s), & \text{si } a \leq t \leq s \leq b. \end{cases} \quad (2.3.14)$$

De plus, pour tout  $i = 0, \dots, n-2$ , on a

$$r_i(t) = r_{n-1}^{(n-1-i)}(t) + \sum_{j=0}^{n-i-1} a_{n-j} r_{n-1}^{(j-i-1)}(t), \quad t \in I.$$

Il est évident qu'il est plus facile de calculer l'expression donnée dans le lemme précédent que celle donnée dans le cas général. Dans ce cas, nous devons seulement obtenir l'expression de la fonction  $r_{n-1}$ , qui est déduite en résolvant un système algébrique linéaire à coefficients constants d'ordre  $n$ .

**Exemple 2.3.3** Considérons le problème aux limites périodique :

$$\begin{cases} y'' + k^2 y = 0, & k > 0 \\ y(0) - y(\omega) = 0 \\ y'(0) - y'(\omega) = 0, & \omega > 0. \end{cases} \quad (2.3.15)$$

Ici

$$\mathcal{L}_2 y(t) = y''(t) + k^2 y(t) \quad (a_1 = 0, \quad a_2 = k^2)$$

$$U_1(y) = y(0) - y(\omega), \quad (\alpha_0^1 = 1, \quad \alpha_1^1 = 0, \quad \beta_0^1 = -1, \quad \beta_1^1 = 0)$$

$$U_2(y) = y'(0) - y'(\omega), \quad (\alpha_0^2 = 0, \quad \alpha_1^2 = 1, \quad \beta_0^2 = 1, \quad \beta_1^2 = -1).$$

Cherchons la fonction de Green sous la forme :

$$G(t, s) = \begin{cases} r_1(t - s), & \text{si } 0 \leq s \leq t \leq \omega, \\ r_1(\omega + t - s), & \text{si } 0 \leq t \leq s \leq \omega, \end{cases}$$

où  $r_1$  est l'unique solution du problème

$$\begin{cases} z''(t) + k^2 z(t) = 0 \\ z(0) - z(\omega) = 0 \\ z'(0) - z'(\omega) = 1 \end{cases}$$

### 2.3. Calcul de la fonction de Green associée à des E.D.O. linéaires à coefficients constants

$$r_1''(t) + k^2 r_1(t) = 0 \implies r_1(t) = c_1 \cos kt + c_2 \sin kt, \quad c_1, c_2 \in \mathbb{R}.$$

Puis

$$\begin{cases} r_1(0) - r_1(\omega) = 0 \\ r_1'(0) - r_1'(\omega) = 1 \end{cases} \implies \begin{cases} c_1(1 - \cos k\omega) - c_2 \sin k\omega = 0 \\ kc_1 \sin k\omega + c_2(k - k \cos k\omega) = 1, \end{cases}$$

on obtient

$$c_1 = \frac{\begin{vmatrix} 0 & -\sin k\omega \\ 1 & (k - k \cos k\omega) \end{vmatrix}}{2k(1 - \cos k\omega)} = \frac{\sin k\omega}{2k(1 - \cos k\omega)},$$

$$c_2 = \frac{\begin{vmatrix} (1 - \cos k\omega) & 0 \\ k \sin k\omega & 1 \end{vmatrix}}{2k(1 - \cos k\omega)} = \frac{(1 - \cos k\omega)}{2k(1 - \cos k\omega)} = \frac{1}{2k}.$$

Donc

$$r_1(t) = \frac{\sin k\omega}{2k(1 - \cos k\omega)} \cos kt + \frac{1}{2k} \sin kt = \frac{\cos \frac{k}{2}\omega}{2k \sin \frac{k}{2}\omega} \cos kt + \frac{1}{2k} \sin kt.$$

D'où

$$\begin{aligned} G(t, s) &= \frac{1}{2k \sin \frac{k}{2}\omega} \begin{cases} \cos \frac{k}{2}\omega \cos k(t - s) + \sin \frac{k}{2}\omega \sin k(t - s), & \text{si } 0 \leq s \leq t \leq \omega \\ \cos \frac{k}{2}\omega \cos k(\omega + t - s) + \sin \frac{k}{2}\omega \sin k(\omega + t - s), & \text{si } 0 < t < s \leq \omega \end{cases} \\ &= \frac{1}{2k \sin \frac{k}{2}\omega} \begin{cases} \cos(\frac{k}{2}\omega - k(t - s)), & \text{si } 0 \leq s \leq t \leq \omega \\ \cos(\frac{k}{2}\omega - k(\omega - s + t)), & \text{si } 0 < t < s \leq \omega \end{cases} \\ &= \frac{1}{2k \sin \frac{k}{2}\omega} \begin{cases} \cos(\frac{k}{2}\omega - k(t - s)), & 0 \leq s \leq t \leq \omega \\ \cos(k(t - s + \frac{\omega}{2})), & 0 < t < s \leq \omega \end{cases} \\ &= \frac{1}{2k \sin \frac{k}{2}\omega} \begin{cases} \cos(k(-t + s + \frac{\omega}{2})), & 0 \leq s \leq t \leq \omega \\ \cos(k(t - s + \frac{\omega}{2})), & 0 < t < s \leq \omega \end{cases} \\ &= \frac{1}{2k} \begin{cases} \csc(\frac{k}{2}\omega) \cos(k(s - t + \frac{\omega}{2})), & \text{si } 0 \leq s \leq t \leq \omega \\ \csc(\frac{k}{2}\omega) \cos(k(t - s + \frac{\omega}{2})), & \text{si } 0 < t < s \leq \omega. \end{cases} \end{aligned}$$

---

## Calcul de la fonction de Green avec Mathematica

Le but de cette section est de construire un algorithme pour calculer la fonction de Green du problème (2.3.1). Une telle construction est basée sur l'expression (2.3.8). Pour arriver à cette expression, nous devons trouver les fonctions  $r$ ,  $y_k$  et  $d_k$ ,  $k = 0, \dots, n-1$ . Puisque la fonction  $r$  est l'unique solution du problème de Cauchy (2.3.3), la première étape consiste donc à résoudre ce problème. Puis nous obtenons les expressions des fonctions  $y_k$ ,  $k = 0, \dots, n-1$ , solutions uniques de  $n$  problème de Cauchy (2.3.7). L'étape suivante de l'algorithme consiste à résoudre le système algébrique (2.3.9) pour déterminer les  $n$  fonctions continues  $d_k$ ,  $k = 0, \dots, n-1$ . Ici nous devons vérifier, dans un premier temps, que la matrice de ce système est inversible. Sinon, la fonction de Green n'existe pas et cela termine le processus. Une fois le système (2.3.9) est uniquement soluble, nous arrivons à l'expression de la fonction de Green  $G(t, s)$ , à travers l'expression (2.3.8) définie dans les deux triangles  $a \leq s < t \leq b$  et  $a \leq t < s \leq b$ .

Comme nous l'avons constaté dans la section précédente, les calculs utilisés dans ce processus sont très complexes, donc pour les équations d'ordre élevé et avec plusieurs conditions aux limites, la résolution peut être très lente. D'autre part, pour le cas particulier des conditions aux limites périodiques, afin d'avoir l'expression de la fonction de Green donnée dans (2.3.14) nous devons seulement déterminer la fonction  $r_{n-1}$  qui vérifie le problème (2.3.13) pour  $j = n-1$ .



## 3.1 Un module pour calculer la fonction de Green

En utilisant Mathematica 8.0.1.0 et l'algorithme décrit précédemment nous avons mis en place un programme (module) dans lequel, en fournissant l'ordre de l'équation, les coefficients de l'opérateur linéaire et les conditions aux limites de deux points sur l'intervalle  $I$ , la fonction de Green correspondante sera calculée. De plus, nous pouvons voir et manipuler à la fois les résultats analytiques et les résultats graphiques. Dans cette section, nous nous concentrerons sur les aspects techniques du module et nous laissons les aspects de la manipulation à la section suivante.

Le module de Mathematica que nous allons présenter fonctionne une fois que nous donnons les valeurs suivantes :

- 1 . L'ordre de l'équation (**n**)
- 2 . Les coefficients (**vecteur c**)
- 3 . Les extrêmes de l'intervalle (**exta et extb**)
- 4 . Les conditions aux limites en deux points en fonction de  $a$  et  $b$  (**vecteur cc**).

De plus, nous avons l'option **périodicité** ( faux / vrai ).

Ce module est divisé en deux parties, selon les conditions aux limites, sont ou non périodiques. Dans ce dernier cas, on peut choisir l'algorithme expliqué au début de cette section.

Au départ, le programme vérifie que le nombre de coefficients de l'équation et de conditions aux limites sont égaux à l'ordre de l'équation considérée. Puisque les conditions aux limites doivent être évaluées dans le module, afin de trouver les coefficients du système (2.3.9), ces conditions doivent être une fonction réelle et non un vecteur.

Pour effectuer la transformation de chaque condition aux limites en une fonction réelle, nous définissons une fonction auxiliaire au début du module appelée (**aux**). Cette fonction permet d'enregistrer le vecteur qui contient les conditions aux limites comme une fonction dépendant de  $u$ ,  $exta$  et  $extb$ . Après sa transformation en un système, les coefficients de  $u^{(j)}(exta)$  et  $u^{(j)}(extb)$  sont extraits comme  $\alpha_j^i$  et  $\beta_j^i$ , respectivement. La partie du code qui réalise cette étape est présentée ci-dessous.

Do[alfa[i, j]=Coefficient[aux[u][exta, extb][[i]], u<sup>(j)</sup>[exta]], {j, 0, Length[c]-1}, {i, 1, Length[c]}];  
 Do[beta[i, j]=Coefficient[aux[u][exta, extb][[i]], u<sup>(j)</sup>[extb]], {j, 0, Length[c]-1}, {i, 1, Length[c]}];  
 Une fois que les coefficients  $\alpha_j^i$  et  $\beta_j^i$  ont été extraits, les fonctionelles  $U_i$ , qui dépendent

de  $u$ ,  $exta$  et  $extb$ , sont définis comme suit :

$$\text{Do}[U_i[u\_][exta\_, extb\_]=\sum_{j=0}^{Length[c]-1}(\text{alfa}[i, j]*u^{(j)}[exta]+\text{beta}[i, j]*u^{(j)}[extb]), \{i, 1, Length[c]\}]$$

La première étape de l'algorithme mentionné ci-dessus est de trouver la solution unique du problème de Cauchy (2.3.3). Ce problème est résolu en employant la commande de Mathematica "DSolve". Le résultat est sauvegardé comme une fonction  $r$  qui dépend de  $t$ .

Notons par  $\mathbf{n}$  l'ordre de l'équation et par  $\mathbf{c}$  le vecteur qui contient les coefficients de cette équation. L'équation du problème (2.3.3) est présentée dans le programme comme suit :

$$\text{ec:} = \{y^{(Length[c])}[t] + \sum_{i=1}^{Length[c]} \mathbf{c}[[i]] y^{(Length[c]-i)}[t].$$

Le problème de cauchy (2.3.3) est résolu comme suit :

$$\text{DSolve}[\text{join}[\{\text{ec} == 0\}, \text{Table}[y^{(i)}[0] == 0, \{i, 0, n-2\}, y^{(n-1)}[0] == 1], y, t];$$

Notons que ce rendement est une liste, ainsi nous devons extraire la partie correspondante de la fonction avec  $y[t]/.\text{DSolve}[\dots][[1]]$ . La commande DSolve renvoie la solution simplifiée.

La deuxième étape de l'algorithme consiste à résoudre les  $n$  problèmes de Cauchy (2.3.7). La solution est obtenue directement en employant la commande DSolve ou à travers l'expression (2.3.6), selon que toutes les données sont des nombres réels ou des paramètres. L'exécution dans Mathematica se fait comme suit :

$$\begin{aligned} &\text{If}[(\mathbf{c} \in \text{Reals} \wedge \text{extb} \in \text{Reals} \wedge \text{exta} \in \text{Reals}) == \text{True}, \\ &\text{Do}[\text{soluci}[k] = \text{DSolve}[\text{join}[\{y^{(Length[c])}[t] + \sum_{i=1}^{Length[c]} \mathbf{c}[[i]] y^{(Length[c]-i)}[t] == 0\} \\ &\text{Table}[y^{(i)}[0] == 0, \{i, 0, k-2\}], \{y^{(i)}[0] == 1\} \\ &\text{Table}[y^{(i)}[0] == 0, \{i, k, Length[c]-1\}], y, t]; \\ &yk[k][t\_]=\text{FullSimplify}[\text{ComplexExpand}[y[t]/.\text{Extract}[\text{soluci}[k], \{1, 1\}]]]; \\ &\{k, 1, Length[c]\}] \\ &\text{Do}[yk[k][t\_]=r^{(Length[c]-k)}[t] + \sum_{j=k}^{Length[c]-1} \mathbf{c}[[Length[c]-j]] r^{(j-k)}[t]; \end{aligned}$$

$\{k, 1, \text{Length}[c]\}$

Les coefficients du système (2.3.9) sont donnés par les conditions aux bords  $U_i$  évaluées en  $y_k$ . Pour le résoudre nous employons la commande de Mathematica `Solve`, et les inconnues sont données dans un vecteur  $d$ .

$$\text{Solve}[\text{Table}[0 == \sum_{j=0}^{\text{Length}[c]-1} \text{beta}[i, j] * r^{(j)}[\text{extb} - s] + \sum_{j=1}^{\text{Length}[c]} d_j[s] U_i[y[j]] [\text{exta}, \text{axtb}], \{i, 1, \text{Length}[c]\}], \text{Table}[d_i[s], \{i, 1, \text{Length}[c]\}]]$$

Finalement, pour avoir l'expression de la fonction de Green donnée dans (2.3.8) nous devons extraire les coefficients de la solution du système (2.3.9) retourné par Mathematica sous forme d'une liste. Dans les lignes suivantes du code nous présentons l'extraction de tels coefficients et la définition de la fonction  $h$  qui correspond à la partie d'addition dans les deux côtés de l'expression (2.3.8).

```
coef := Sort[Extract[ecuacion, {1}]];
Do[e[i][s_] := [d_i[s]/.Extract[coef, {1}];, {i, 1, n}];
h[t_, s_] := Sum_{i=1}^{Length[c]} Simplify[e[i][s]] y[i][t];
```

Nous rappelons que ce module comporte deux parties, la première est l'algorithme général, décrit ci-dessus, et la deuxième partie se compose d'un algorithme spécifique pour résoudre des équations avec des conditions périodiques. L'option **Périodicité** rend le programme exécutable d'une manière ou d'une autre. Dans l'algorithme spécifique pour les conditions aux limites périodiques, nous avons besoin seulement de résoudre un problème aux limites et de définir la fonction de Green comme dans (2.3.14).

## 3.2 Code de manipulation

```
Clear ["Global *"] ;
Manipulate[
Module[{d, e, coef, aux, i, j, k, alfa, beta, U, yk, ecrnt, cadenatexto,
cadenatexto2, rango, condcont2, ecinicial, determinante, soluci}, off[] ;
```

---

```

If [(c ∈ Reals ∧ extb ∈ Reals ∧ exta ∈ Reals)===True,
Graphic = True, Graphic = False];
If[(n ∈ Integers && n >0) ===True,
If[Periodicity = False,
If[(extb ∈ Reals ∧ exta ∈ Reals ∧ exta ≥extb) ===True,
MessageDialog["a must be less than b"]; ecu=0; 1=0,
If[n=Length[c] && n=Length[cc],
1=0; aux[u_] [exta_, extb_] : =cc;
condcont = Table [aux[u] [exta, extb] [[i]] == 0, {i, 1, Length[c] }];
Do[alfa[i, j]=Coefficient[aux[u] [exta, extb] [[i]], u(j)[exta]],
{j, 0, Length[c]-1 },{i, 1, Length[c] }];
Do[beta[i, j]=Coefficient[aux[u] [exta, extb] [[i]], u(j)[extb]],
{j, 0, Length[c]-1 },{i, 1, Length[c] }];
rango=MatrixRank[Join[Table[alfa[i, j], {i, 1, n},{j, 0, n-1}],
Table[beta[i, j], {i, 1, n}, {j, 0, n-1}],2]];
Do[Ui[u_] [exta_, extb_]= $\sum_{j=0}^{Length[c]-1}$ (alfa[i, j]* u(j)[exta])+
beta[i, j]* u(j)[extb]), {i, 1, Length[c] }];
condcont2=Table[Ui[u] [exta, extb] ==0, {i, 1, Length[c] }]; contador=0;
If[ condcont ===condcont2, Label [volver]; contador = contador+1;
ecinicial = DSolve[Join[{y(Length[c]) [t]+ $\sum_{i=1}^{Length[c]}$ c[[i]] y(Length[c]-i)[t]==0},
Table[y(i)[0] == 0, {i, 0, Length[c]-2 }], {y(Length[c]-1)[0] == 1}],
y, t]; cadenatexto= ToString[ecinicial];
If[StringMatchQ[cadenatexto, "*Root*"]== False,
Style[Column[{Row[{Panel[
Column[{Style[" ", Bold], Style["PROBLEM : ", Bold], Style[" ", Bold],
Row[{problema[t_]=TraditionlForm[u(Length[c])[t]+
 $\sum_{i=1}^{Length[c]}$ c[[i]] u(Length[c]-i)[t]; problema[t]== $\sigma[t]$ ,
", t ∈ ["exta, ", "extb, "]}], Style[" ", Bold],
Style["With boundary conditions", Bold], Style[" ", Bold],

```

---

```

condcont2, r[t_]=ComplexExpand[Re[y[t] /. ecinicial[[1]]]; ,
Style[" ", Bold], Style["The Green's function is given by : ",
Bold],
If[c ∈ Reals ∧ extb ∈ Reals ∧ exta ∈ Reals ) ===True,
Do[soluci[k]=
DSolve[Join[{y(Length[c]) [t]+∑i=1Length[c] c[[i]] y(Length[c]-i) [t] == 0},
Table[y(i) [0]==0,{i, 0, k-2}],{y(k-1) [0] == 1},
Table[y(i) [0]==0, {i, k, Length[c]-1}]], y, t];
yk[k] [t_] =FullSimplify[ComplexExpand [y[t] /.
Extract[soluci[k], {1, 1}]]]; ,{k, 1, Length[c]}],
Do [yk[k] [t_]=r(Length[c]-k) [t] + ∑j=kLength[c]-1 c[[Length[c]-j]] r(j-k) [t]; ,
{k, 1, Length[c]}]; determinante = Chop[
1-Det[Table[Ui[yk[j]][exta, extb],{i, 1, n}, {j, 1, n}]]];
If[rango == n,
If[(determinante == 0 &&
(c ∈ Reals ∧ extb ∈ Reals ∧ exta ∈ Reals) ) ===True,
MessageDialog["There is not Green's function"]; Graphic =
False; 1=0; G[t_, s_]="There is not unique solution"; ecu=0;
aa=exta; bb=extb; , ecuacion := Solve[Table[0=∑j=0Length[c]-1 beta[
i, j] * r(j) [extb-s]+∑j=1Length[c] dj[s] Ui[yk[j]][exta, extb],
{i, 1, Length[c]}], Table[di[s], {i, 1, Length[c]}]];
If[ecuacion == {},
MessageDialog["There is not Green's function"]; Graphic =False;
1=0; G[t_, s_]="There is not unique solution";
ecu=0; aa=exta; bb=extb; ,
ecu=1; coef :=Sort[Extract[ecuacion, {1}]];
Do[e[i] [s_] :=di[s] /. Extract[coef, {i}], {i, 1, n}];
h[t_, s_] := ∑i=1Length[c] Simplify [e[i] [s]] yk[i][t];
G1[t_, s_] = Simplify[Chop[r[t-s]+h[t, s]]];

```

---

```

G2[t_, s_] = Simplify[Chop[h[t, s]]];
G[t_, s_] =  $\begin{cases} G1[t, s] & exta \leq s \leq t \leq extb \\ G2[t, s] & exta < t < s \leq extb \end{cases}$ ;
aa=exta; bb=extb;]],
MessageDialog["The boundary condicions are linearly dependent"];
Graphic = False; 1=0; G[t_, s_]="There is not unique solution";
ecu=0; aa=exta; bb=extb; ]
Row[{Style["G[t, s] = ", Bold], TraditionalForm [G[t, s]]}],
Style[ " ", Bold], Style[ " ", Bold], Style[ " ", Bold]}, Center]],
Style[ " ", Bold], If[Graphic == False, , Manipulate[If[ecu == 1, Plot3D[
G[t, s], {s, aa, bb}, {t, aa, bb}], "Cannot show the graphic"],
FrameLabel→ " Graphic of the Green's function"]]]]],
c = N[c]; If[contador == 2, MessageDialog[
"The program cannot calculate the Green's function"];
1=0; ecu=0; aa=exta; bb=extb; ,Goto[volver]] ],
MessageDialog["The boundary condicions are not valid"];
1=0; ecu=0; aa=exta; bb=extb; ],
MessageDialog["Vector of coefficients or Boundary condicions :
LENGTH INCORRECT"]; 1=0; ecu=0; aa=exta; bb=extb; ]],
If[(extb ∈ Reals ∧ exta ∈ Reals ∧ exta ≥ extb) == True,
MessageDialog["a must be less than b"]; ecu=0; 1=0,
contador2 =0;
If [n=Length[c],
Off[DSolve : : bvnul]; 1=0; Label[volver2];
ecrnt = DSolve[Join[{y(Length[c]) [t]+ $\sum_{i=1}^{Length[c]} c[[i]] y^{(Length[c]-i)}[t] == 0$ },
Table[y(i)[exta]-y(i)[extb] == 0, {i, 0, Length[c]-2}],
{y(Length[c]-1)[exta]-y(Length[c]-1)[extb] ==1}], y, t];
cadenatexto2 = ToString[ecrnt]; contador2= contador2+1;
If[StringMatchQ[cadenatexto2, "*Root*"]==False,
Style[Column[{Row[{Panel[[Column[{Style[ " ", Bold], Style["PROBLEM : ",

```

---

```

    Bold], Style[ " ", Bold], Row[{problema[t_] =TraditionalForm[
u(Length[c])[t] +  $\sum_{i=1}^{Length[c]}$  c[[i]] u(Length[c]-i)[t]]; problema[t] == $\sigma$ [t],
    ",      t  $\in$  [", exta, ", " , extb, "]" }], Style[ " ", Bold],
    Style["With boundary condicions ", Bold], Style[ " ", Bold],
condcont =Table[u(i)[exta]- u(i)[extb] == 0, {i, 0, Length[c]-1}];
condcont, Style[ " ", Bold], Style[
"the Green's function is given by: ", Bold],
If[ecrnt=={ },
MessageDialog["there is not Green's function"];
Graphic = False; G[t_, s_]="There is not unique solution";
ecu =0; aa=exta; bb=extb; ,
ecu =1; r[t_] = ComplexExpand[Re[y[t] /.ecrnt[[1]]]];
ecu =1; G1[t_, s_]=Simplify[Chop[r[exta+t-s]]];
G2[t_, s_] =Simplify[Chop[r[extb+t-s]]];
G[t_, s_] =  $\begin{cases} G1[t, s] & exta \leq s \leq t \leq extb \\ G2[t, s] & exta < t < s \leq extb \end{cases}$ ; aa=exta; bb=extb; ];
Row[{Style["G[t, s]= ", Bold], TraditionalForm[G[t, s]]}],
Style[ " ", Bold] , Style[ " ", Bold], Style[
" ", Bold]}, Center]], Style[ " ", Bold],
If[Graphic == False, , Manipulate[If[ecu ==1, Plot3D[G[t, s],
{s, aa, bb}, {t, aa, bb}, "Cannot show the graphic"], FrameLabel→
" Graphic of the Green's function"]]]]], c = N[c]; If [contadot2 == 2,
MessageDialog["the program cannot calculate the Green's function"];
1=0; ecu=0; aa=exta; bb=extb; , Goto[volver2]]],
MessageDialog["Length of coefficients vector or boundary condicions
INCORRECT"]; 1=0; ecu=0; aa=exta; bb=extb;]],
MessageDialog["Order must be a positive integer"]; 1=0; ecu=0;
aa=exta; bb=extb;]],
Style[" Program to compute the Green's function of the equation :
un + c1un-1 + c2un-2 + ... + cnu =  $\sigma(t)$ , t  $\in$  [a, b]", Bold],

```

```

Style[Column[{
    "
    With the two-point boundary
    conditions :  $U_i[u] : = \sum_{j=0}^{n-1} \alpha_i^j u^{(j)}(a) + \beta_i^j u^{(j)}(b) = 0, \quad 1 \leq i \leq n$ ", Bold],
    " , Bold],
    {{n, 2, "
    Order"}},
    {{c, {0, 0}, "Coefficients"}},
    {{exta, 0, "a"}}, {extb, 1, "b"}},
    {{Periodicity, False, "Periodic conditions " }, { False, True}, ControlType→Checkbox},
    {{cc, If[Periodicity = True, , {u[exta], u[extb]}], " Boudary conditions"},
    FieldHint→ "Periodic", Enabled→Not [Periodicity == True]},
    Delimiter, {{1, " "}, {ENTER}, ControlType→Setter},
    ControlPlacement→Top, TrackedSymbols→{1}, Alignment→Center].

```

### 3.3 Exemples d'application

Dans cette section on donne l'illustration de l'application de ce code sur les problèmes (2.3.11), (2.3.12) et (2.3.15) étudiés au chapitre précédent.







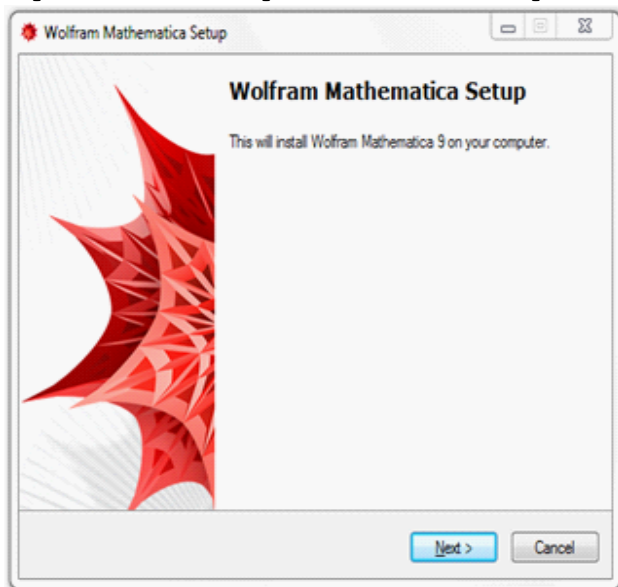


---

# Annexes

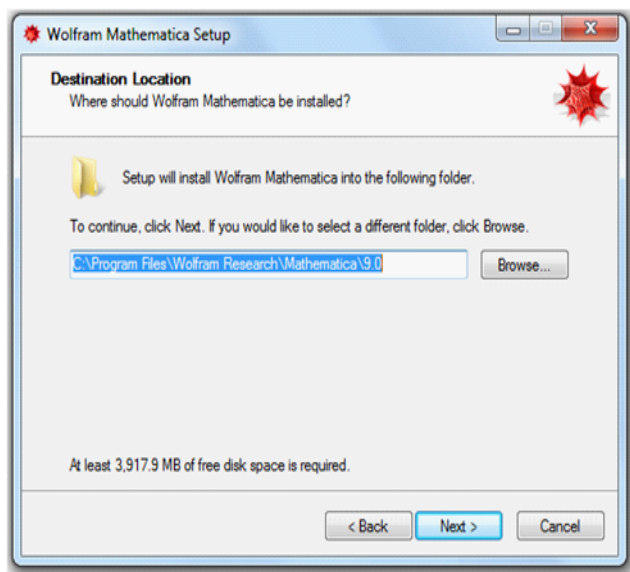
## Installation de Mathematica

1. Arrêter toutes les installations précédentes de Mathematica qui fonctionnent sur votre machine et relancer le Windows;
2. Si vous installez Mathematica à partir d'un dossier exécutable téléchargé, double clic sur dossier exécutable pour le lancer. Si vous installez d'un DVD, insérer le DVD dans votre ordinateur pour commencer l'installation;
3. Le message d'installation de Wolfram Mathematica apparaît sur votre écran. Cliquer après sur **Next >** pour commencer le procédé d'installation;



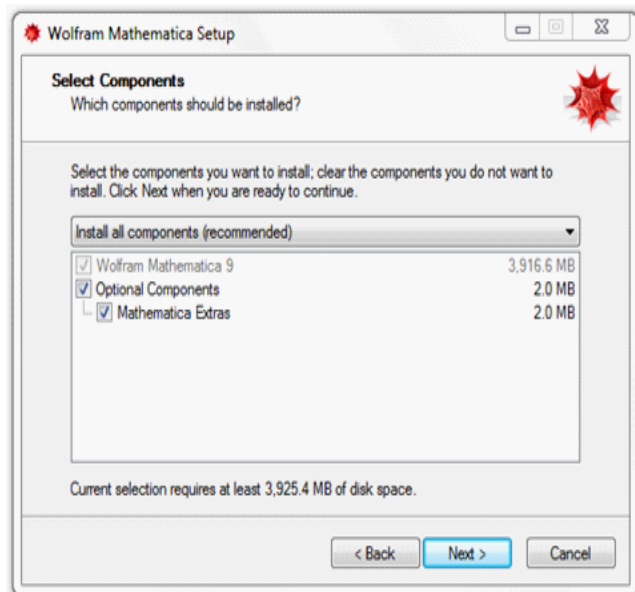
4. Par défaut

Mathematica installe sur l'annuaire C:\Programme\FilesWolfram\Research\Mathematica\9.0. Cliquer après sur **Next >** pour installer Mathematica sur l'annuaire de défaut.



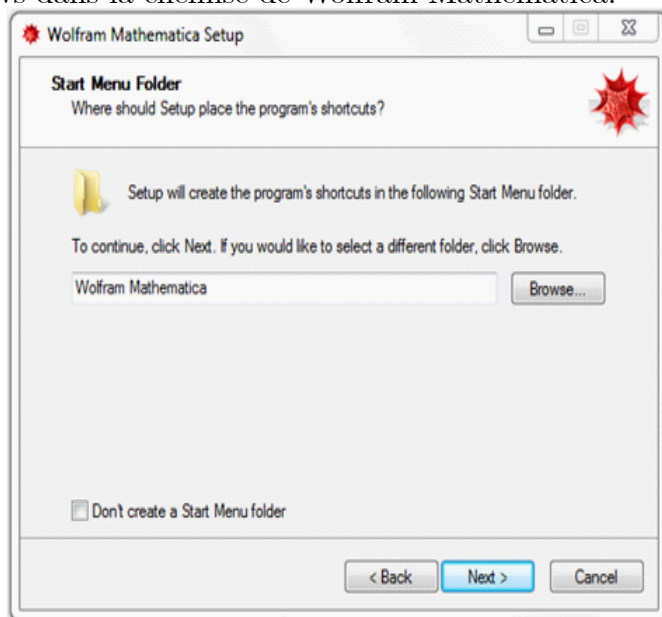
Pour installer mathematica sur les différents annuaires, choisir un autre annuaire avec **Browse...** . Puis cliquer sur **Next >** pour continuer.

5. Cliquer sur **Next >** pour installer les composants facultatifs de Mathematica. Ces composants supplémentaires permettent à des navigateurs de Web d'afficher les cahiers interactifs de Mathematica;



Si vous préférez ne pas installer les composants supplémentaires, décocher le checkbox qui est à côté des composants et cliquer sur **Next >** pour continuer.

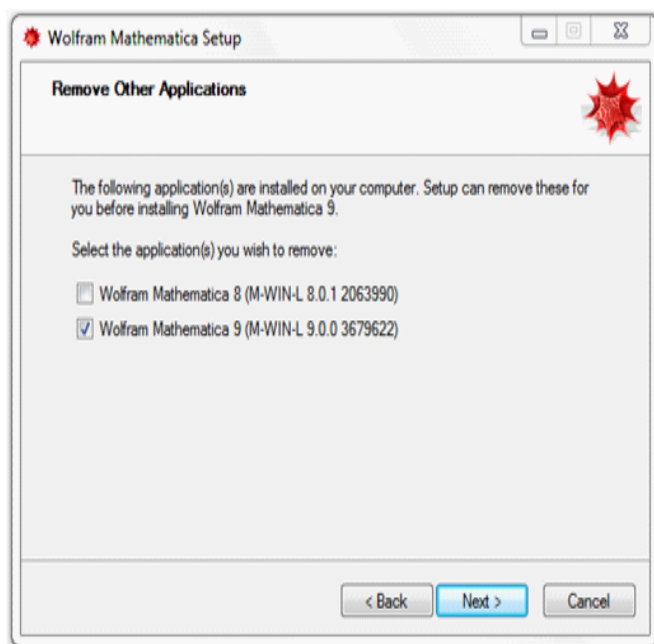
6. L'installateur ajoute des raccourcis de Mathematica au menu de début de Windows dans la chemise de Wolfram Mathematica.



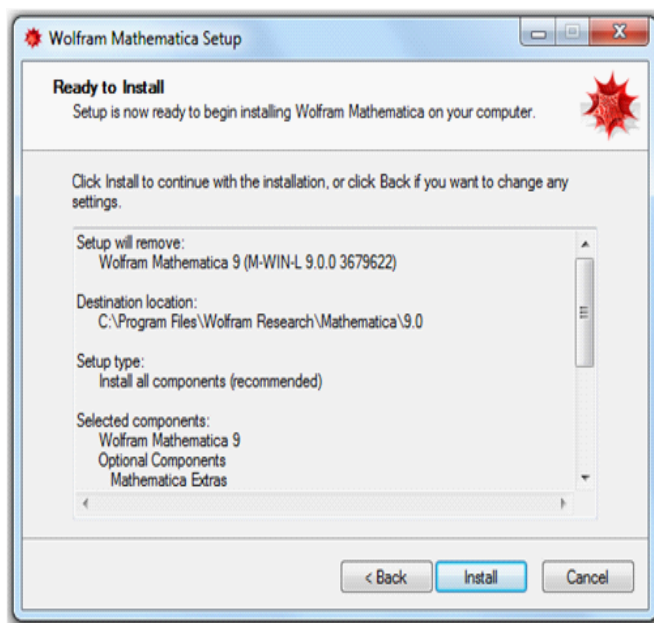
Pour faire apparaître les raccourcis dans une chemise différente, le clic passent en revue et choisis un dossier. Si vous préférez ne pas faire un raccourci dans le menu de début vérifier la boîte à côté de « ne crée pas une chemise de menu de début ». Cliquer alors sur **Next >** pour continuer l'installation.

7. L'élimination des fenêtres d'applications apparaît seulement s'il y a des versions plus anciennes de Mathematica installées sur l'ordinateur.

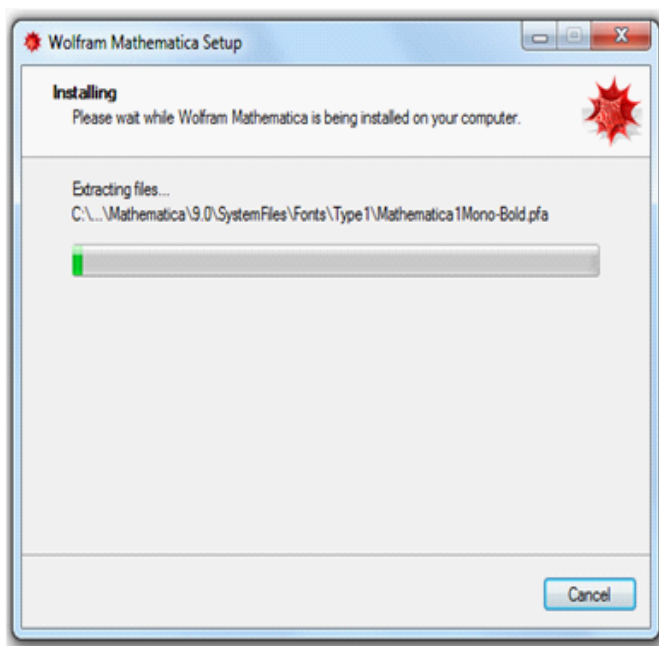
Choisir toutes les vieilles versions de Mathematica que vous voulez désinstaller et choisir par suite **Next >**



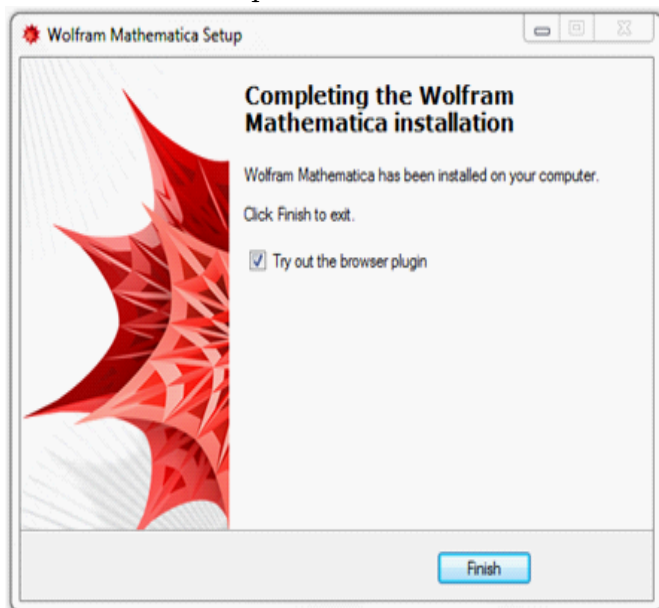
8. La configuration de l'installation de Mathematica est finie. Puis cliquer sur **Instal** pour commencer l'installation du Mathematica.



Attendre le progrès d'installation à accomplir.



9. Pour accomplir l'installation et lancer Mathematica, cliquer sur **Finish**.



L'installation est maintenant complète. Après, activer Mathematica pour commencer à l'employer.



---

# Conclusion

Ce travail nous a permis de comprendre que Mathematica est un logiciel développé par et pour des mathématiciens, puis pour tous les scientifiques pouvant avoir besoin de l'aide d'un logiciel performant de calcul formel. Tout a été fait pour que le langage soit le plus naturel possible, aussi près que possible de ce que le mathématicien écrit sur sa feuille de travail.

Comme une application on a choisi d'écrire un programme avec Mathematica qui permet de calculer la fonction de Green, associée à certaines classes de problème aux limites, qui est un outil mathématique important dans l'étude des problèmes différentiels.

---

# Bibliographie

- [1] A. Cabada, J.A. Cid, B.M. Villamarin, *Computation of Green's functions for boundary value problems with Mathematica*. Applied Mathematics and Computation 219 (2012), 1919-1936.
- [2] A. Cabada, *The method of lower and upper solutions for third-order periodic boundary value problems*, J. Math. Anal. App. 195 (1995) 568-589.
- [3] A. Cabada, *Green's Functions in the Theory of Ordinary Differential Equations*. Springer Briefs in Mathematics. 2014.
- [4] É. Gaul, D. Boire, I. Lizon, *Tutoriel Mathematica-introduction*. WOL FRAM, stephen, Wolfram Mathematica documentation center, <http://reference.wolfram.com/mathematica/guide/Mathematica.html>, 2009.
- [5] H. Ruskeepaa, *Mathematica Navigator Mathematics, Statistics and Graphics*, Elsevier, 2009.
- [6] J.L. Raimbault, D. Boulanger, *Introduction à Mathematica*, Université de paris sud, [http://www.lct.jussieu.fr/pagesperso/toulouse/enseignement/cours\\_mathematica.pdf](http://www.lct.jussieu.fr/pagesperso/toulouse/enseignement/cours_mathematica.pdf), 2005.
- [7] J. Toulouse, *Introduction à Mathematica*, UPMC, Paris, [http://www.lct.jussieu.fr/pagesperso/toulouse/.../cours\\_mathematica.pdf](http://www.lct.jussieu.fr/pagesperso/toulouse/.../cours_mathematica.pdf).
- [8] P. Young, *Introduction to Mathematica*, [http://www.physics.ucsc.edu/~peter/115/math\\_intro.pdf](http://www.physics.ucsc.edu/~peter/115/math_intro.pdf), 2013.