Analyse des données en Science de la Terre

O. de Viron

2009-2010

Table des matières

Chapitre 1

Analyse numérique

> Je vis dans l'approximatif et je
> m'en rapproche de plus en plus
>
> Julos Beaucarne

1.1 Introduction

Pour alergique qu'il puisse être au mathématique, il est souvent demandé à un scientifique de calculer des trucs. Pour cette raison, vous avez été dotés d'un bagage mathématique raisonnable. Vous êtes supposés savoir dériver, intégrer, calculer des racines de polynomes, résoudre des systèmes d'équations, ... Cependant, il est des cas, nombreux, où ce bagage ne vous sera d'aucune utilité. En particulier, deux cas peuvent se présenter : (1) le calcul est trop compliqué pour être résolu, ou (2) vous connaissez la fonction non pas analytiquement, mais par des points de mesure. Dans ce cas, la seule solution est de travailler numériquement. Malheureusement, si la solution analytique d'un problème est exacte, par définition, la solution numérique ne sera, la plupart du temps, qu'approchée. Par conséquent, l'art du numéricien consistera à minimiser les erreurs de calculs pour les amener à des valeurs acceptables compte tenu de l'application qu'il veut faire du résultat. L'autre enjeu de l'analyse numérique, c'est le temps de calcul. La plupart du temps, on peut ne pas s'en inquiéter outre mesure mais, selon la méthode utilisée, on pourra avoir une différence de temps de calcul de plusieurs ordres de grandeur.

Dans ce petit chapitre, nous nous limiterons à trois problèmes élémentaires et classiques, que vous avez déjà abordé lors du cours de méthode numérique de Stéphane Jacquemoud, en L3.

La dérivée numérique : Je connais la valeur d'une fonction en un certain nombre de points. Je souhaite en déduire la valeur de la dérivée en ces mêmes points.

L'intégration numérique : C'est presque la même chose, mais juste le contraire. A partir des valeurs connues, comment calculer les valeurs de l'intégrale d'une fonction.

Intégration numérique d'équation différentielle ordinaire : Assez proche du précédent, on connait le comportement d'un système par son équation différentielle, et on reproduit, de proche en proche, son comportement temporel.

1.2 Dérivation numérique

1.2.1 Dérivation numérique sur deux points

Je supposerai que vous avez tous eu de nombreux cours sur ce que c'est qu'une dérivée. Je rappelle donc simplement que la dérivée s'intéresse à la variation d'une fonction et calcule un taux de variation en fonction du temps, de l'espace,... La variation d'une fonction $f(x)$ est donnée par

$$\Delta f = f(x+a) - f(x),$$

son taux de variation par

$$\frac{\Delta f}{\Delta x} = \frac{f(x+a) - f(x)}{a}.$$

La dérivée est, par définition, le taux de variation quand le temps devient infiniment petit :

$$\frac{df}{dx} = \lim_{a \to 0} \frac{f(x+a) - f(x)}{a}.$$

Si l'on connaît la fonction $f(x)$, il est alors simple de calculer $\frac{df}{dt}$, et vous savez tous le faire. Cependant, si on ne connaît pas $f(x)$ et mais qu'on mesure sa valeur à des instants donnés, on peut calculer le taux de variation entre deux mesures et dire que la dérivée est presque égale à ce taux de variation.

La méthode la plus immédiate pour calculer la dérivée numériquement est donc la suivante :

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

ou, de façon équivalente,

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

Si $x_i - x_{i-1}$ est suffisament petit, c'est une approximation acceptable. En pratique, ce n'est pas excellent. Prenons par exemple la fonction suivante : $\frac{-5}{1+\mathrm{atan}^2 x}$, entre [0, 4]. On en calcule les dérivées dans pour différents échantillonages, comme illustré sur la figure 1.1.

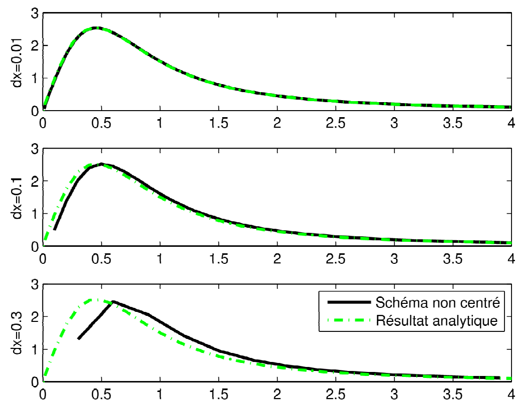


FIGURE 1.1 – Dérivée de la fonction $\frac{-5}{1+\mathrm{atan}^2 x}$ pour différents échantillonage. Le résultat numérique est en trait continu, et le résultat analytique est en trait discontinu.

1.2.2 La dérivée centrée sur trois points

On note, dès le second dessin, qu'il y a un décalage entre la dérivée estimée et calculée. C'est normal, dans la mesure où la formule utilisée estime, en réalité, la dérivée entre deux points de mesure. On peut corriger cela en calculant la formule, un poil plus complexe :

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}$$

Si l'on reprend l'exemple précédent, on obtient la figure 1.2. On voit que la dérivée centrée est toujours plus proche de la solution analytique que la dérivée non centrée.

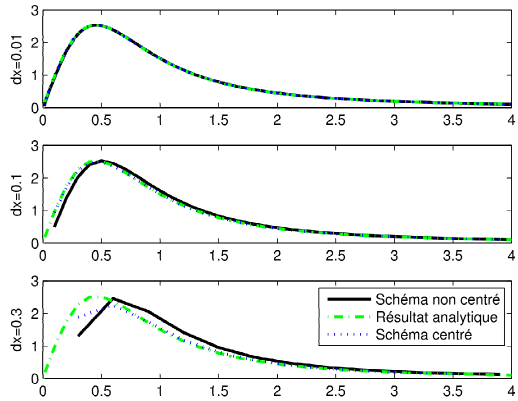


FIGURE 1.2 – Dérivée de la fonction $\frac{-5}{1+\mathrm{atan}^2 x}$ pour différents échantillonage. La dérivée centrée est ajoutée en trait pointillé

Exemple 1.1. *Calculer la dérivée de la fonction x^2 entre 0 et 1, par différence non-centrée et par différence centrée. On part du tableau suivant :*

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
x^2	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1

Par différence non centrée, on trouve :

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$x_i^2 - x_{i-1}^2$	/	0.01	0.03	0.05	0.07	0.09	0.11	0.13	0.15	0.17	0.19
$\frac{x_i^2 - x_{i-1}^2}{0.1}$	/	0.1	0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
$2x$	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2

Ce n'est pas mal, mais décalé de 0.1. Prenons à présent une différence centrée :

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$x_{i+1}^2 - x_{i-1}^2$	/	0.04	0.08	0.12	0.16	0.20	0.24	0.28	0.32	0.36	/
$\frac{x_{i+1}^2 - x_{i-1}^2}{0.2}$	/	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	/
$2x$	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2

L'exemple montre que la solution est exacte avec la différence centréNotons tout de même que l'on a perdu les deux premières valeurs.

1.2.3 La dérivée seconde

Supposons que l'on souhaite calculer la dérivée seconde de la fonction f. Pour cela, il faut différencier deux dérivées. Soit donc f, observée aux points x_i, supposés équidisants. On a dès lors

$$\begin{aligned} f'(x_i) &\simeq \frac{f(x_{i+1}) - f(x_i)}{h} \\ f'(x_{i-1}) &\simeq \frac{f(x_i) - f(x_{i-1})}{h} \end{aligned}$$

Par définition,

$$f''(x) = \lim_{h\to 0} \frac{f'(x+h) - f'(x)}{h},$$

ce qui donne

$$\begin{aligned} f''(x_i) &\simeq \frac{\frac{f(x_{i+1})-f(x_i)}{h} - \frac{f(x_i)-f(x_{i-1})}{h}}{h} \\ &\simeq \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} \end{aligned}$$

Exemple 1.2. *Calculer la dérivée seconde de la fonction x^3 entre 0 et 1. On part du tableau suivant :*

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
x^3	0	0.001	0.008	0.027	0.064	0.125	0.216	0.343	0.512	0.729	1

Par conséquent,

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$x_{i+1}^3 - x_{i-1}^3 - 2x_i^3$	/	0.006	0.012	0.018	0.024	0.030	0.036	0.042	0.048	0.0540	/
$\frac{x_{i+1}^3 - x_{i-1}^3 - 2x_i^3}{0.01}$	/	0.6	1.2	1.8	2.4	3.0	3.6	4.2	4.8	5.4	/
$6x$	0	0.6	1.2	1.8	2.4	3.0	3.6	4.2	4.8	5.4	6

Ici encore, j'ai choisi une fonction que l'on dérive exactement. Ce n'est pas le cas de toutes, bien entendu.

Exemple 1.3. *Calculer la dérivée seconde de la fonction e^{x^2} entre 0 et 1. On part du tableau suivant :*

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
e^{x^2}	1.000	1.010	1.041	1.094	1.174	1.284	1.433	1.632	1.897	2.248	2.718

Par conséquent,

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$f(x_{i+1}) - f(x_{i-1}) - 2f(x_i)$	/	0.021	0.023	0.026	0.031	0.039	0.050	0.065	0.087	0.120	/
$\frac{f(x_{i+1})-f(x_{i-1})-2f(x_i)}{0.01}$	/	2.1	2.3	2.6	3.1	3.9	5.0	6.5	8.7	12.0	/
$2e^{x^2} + 4x^2e^{x^2}$	2.00	2.06	2.25	2.58	3.10	3.85	4.93	6.46	8.65	11.78	16.31

Pour cette fonction plus complexe, le résultat n'est pas trop mauvais.

1.2.4 La dérivation numérique sous matlab

Exemples

Il suffit de faire le même calcul que ce qu'on fait à la main, mais la structure vectorielle de MATLAB nous permettra de calculer très facilement.

Exemple 1.4. *Calculer la dérivée de la fonction x^2 entre 0 et 1, par différence non-centrée et par différence centrée. On génère pour commencer le tableau des* `x`*, par exemple avec un pas de 0.1. On génère ensuite x^2 :*

```
x=0:0.1:1
y=x.^2
```

Pour calculer la dérivée sur deux points, on utilise l'une des deux solutions ci-dessous :

```
dy=(y(2:end)-y(1:end-1))./(x(2:end)-x(1:end-1))
dy=diff(y)./diff(y)
```

Pour estimer la dérivée par différence centrée, on écrit

```
dy=(y(3:end)-y(1:end-2))./(x(3:end)-x(1:end-2))
```

Par exemple,

```
>> x=0:0.1:1;
>> y=x.^2;
>>dy=(y(3:end)-y(1:end-2))./(x(3:end)-x(1:end-2))

dy =

  Columns 1 through 8

    0.2000    0.4000    0.6000    0.8000    1.0000    1.2000    1.4000    1.6000

  Column 9

    1.8000
```

Exemple 1.5. *Calculons à présent la dérivée seconde de x^2, toujours entre 0 et 1.*

```
x=0:0.1:1;
y=x.^2;
d2y=(y(3:end)+y(1:end-2)-2*y(2:end-1))./(x(2:end-1)-x(1:end-2)).^2

On obtient le résultat suivant:
d2y =

  Columns 1 through 8

    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000

  Column 9

    2.0000
```

Si la fonction est à deux variables, c'est tout aussi simple.

Exemple 1.6. *Le potentiel gravitationel exercé par une masse ponctuelle se calcule comme*

$$V = -\frac{GM}{r}.$$

Soit une masse ponctuelle de $\frac{1}{G}$ kg, située en l'origine. Calculer le potentiel pour x et y compris entre -1 et 1. Pour calculer la force exercée en un point, il suffit de dériver le potentiel par rapport à la coordonnée :

$$\boldsymbol{F} = -\boldsymbol{\nabla} V.$$

Représenter le potentiel et la force gravitationnelle (normalisée).

```
x=-1:0.05:1; % Je définis x
y=-1:0.05:1; % Je définis y
[X,Y]=meshgrid(x,y); % Si vous ne savez pas ce que ça fait,
                     % affichez X et Y.
R=sqrt(X.^2+Y.^2); % Je calcule R
V=-1./R;  % Je calcule V
% Je calcule les dérivées de V
dVdx=-(V(:,3:end)-V(:,1:end-2))./(X(:,3:end)-X(:,1:end-2));
dVdy=-(V(3:end,:)-V(1:end-2,:))./(Y(3:end,:)-Y(1:end-2,:));
% Je coupe pour avoir la même grille en X et en Y
dVdx=dVdx(2:end-1,:);
dVdy=dVdy(:,2:end-1);
N=sqrt(dVdx.^2+dVdy.^2); % Je calcule la norme
pcolor(X,Y,V); shading flat % Je dessine le potentiel
colormap hsv % Je change l'échelle de couleur
hold on % Je demande de dessiner la suite par dessus
quiver(X(2:3:end-1,2:3:end-1),Y(2:3:end-1,2:3:end-1),...
    dVdx(1:3:end,1:3:end)./N(1:3:end,1:3:end),...
    dVdy(1:3:end,1:3:end)./N(1:3:end,1:3:end))
% Je dessine le champ de vecteurs, en ne prenant qu'un point sur 3
hold off
```

Exercices

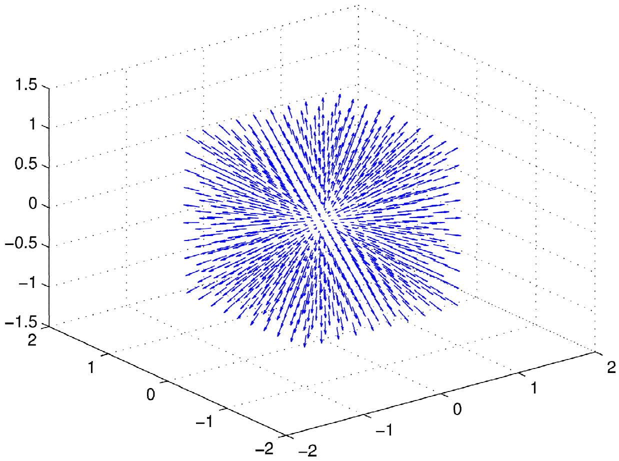
Exercice 1.2.1. Calculer numériquement les dérivées premières et secondes des fonctions suivantes entre -1 et 1, et comparez le résultat avec la solution analytique. Pour représenter une fonction, vous pouvez utiliser la fonction plot.

1. $y = x^3 - 5$
2. $y = \frac{1}{x}$
3. $y = |x|$
4. $y = 3\cos x - 2x^2$
5. $y = e^x$

Exercice 1.2.2. Soit le potentiel défini par la fonction suivante :

$$V = -3\sin x \cos x \cos y.$$

Calculer le pour $x \in [0, \pi]$ et $y \in [0, 2\pi]$. Evaluer ensuite les dérivées premières de V par rapport à x et



y, et estimer en chaque point le vecteur

$$\boldsymbol{v} = (-\frac{\partial V}{\partial x}, \frac{1}{sin(x)}\frac{\partial V}{\partial y})^T.$$

Ce champ de vecteur, lié au terme indivergenciel tangentiel du théorème de Helmoltz (pour ceux à qui cela dit quelque chose), est courant dans la nature. Par exemple, il est associé au vent dominant à nos latitude (dit « vent géostrophique ») : le vent tourne autour des hautes et des basses pressions dans des sens opposés.

Exercice 1.2.3. Générez un champ scalaire qui correspond à la distance depuis l'origine en trois dimensions, entre -1 et 1 dans chaque direction. Calculer la dérivée du champs dans chacune des directions et représentez le vecteur gradient. Pour faire un dessin de champ vectoriel à trois dimensions, la fonction `quiver3` sera utile. Vous devriez obtenir le résultat illustré ci-dessus.

1.3 L'intégration numérique

1.3.1 Introduction

Comme pour la dérivée, je supposerai que vous savez ce que c'est qu'une intégrale. Dans le cas où vous n'avez aucune idée de la définition, je vous propose de retourner à vos cours de licence ou même avant. Je rappelle simplement que le but de l'intégrale est de calculer la surface sous une courbe. A la base, cet usage-là est totalement intéressant. Par extension, l'intégrale permet de calculer l'effet d'une cause variable, ce qui est plus utile. Par exemple, je sais qu'un poluant se déverse dans l'environnement avec une valeur connue et variable dans le temps, je peux calculer à chaque instant le volume total déversé jusque là, simplement en intégrant la valeur instantanée.

Vous vous souvenez sûrement que l'intégrale n'est jamais que la somme des surfaces des rectangles (qui approximent au mieux la surface) dont la base est l'incrément de temps et la hauteur la valeur de la fonction.

Supposons donc la fonction $f(x)$ que l'on ne connaît que via sa valeur en un certain nombre de points x_i. On connaît donc l'ensemble des $f(x_i)$, que l'on notera f_i. On cherche à calculer

$$I = \int_a^b f(x)\, dx,$$

où l'on suppose que a et b sont inclus dans l'intervalle $[x_1, x_N]$. On supposera en pratique que l'on doit calculer

$$I = \int_{x_1}^{x_N} f(x)\,dx.$$

La méthode des rectangles

Comme d'habitude, nous allons commencer pédestrement par appliquer sans trop réfléchir la définition.

$$I = \int_{x_k}^{x_l} f(x)\,dx \simeq \sum_{i=1}^{N} f_k \delta x_k$$

Exemple 1.7. *Calculer l'intégrale de la fonction x^2 entre 0 et 1, par la méthode des rectangles. On part à nouveau du tableau suivant :*

x	*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1*
x^2	*0*	*0.01*	*0.04*	*0.09*	*0.16*	*0.25*	*0.36*	*0.49*	*0.64*	*0.81*	*1*

Les δx_k sont tous égaux à 0.1. *On en a 11. On aura donc*

$$\int_0^1 x^2\,dx = 0.1 \sum_{i=1}^{11} x_i^2 = 0.3850.$$

Ce résultat n'est pas très bon, on sait en effet que le résultat que l'on doit obtenir est 0.3333.

1.3.2 La méthode des trapèzes

On peut raffiner l'estimation en remplaçant les rectangles par des trapèze, qui approximent mieux la fonction, comme l'illustre la Figure 1.3. Le trapèze est basé sur les valeurs de la fonction au début et à

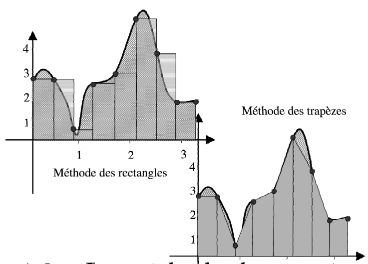


FIGURE 1.3 – La méthode des trapèzes.

la fin de l'intervalle. Comme c'est un trapèze rectangle (trois côtés sont soit horizontaux soit verticaux), la surface d'un trapèze est :

$$S = \delta x \frac{x_i + x_{i+1}}{2}.$$

Donc, parce que l'intégrale se calcule comme la somme de ces surfaces élémentaires :

$$I \simeq \sum_{i=1}^{N} S_i = \sum \delta x_i \frac{h^i_{\text{début}} + h^i_{\text{fin}}}{2}$$

Or, on sait que $h^i_{\text{fin}} = h^{i+1}_{\text{début}}$, donc, si les points sont équidistants ($\delta x_i = \delta x, \forall i$).

$$\begin{aligned} I &\simeq \sum \delta x_i \frac{h^i_{\text{début}} + h^i_{\text{fin}}}{2} \\ &= \sum \delta x \frac{h^i_{\text{début}} + h^{i+1}_{\text{début}}}{2} \\ &= \delta x \frac{h^1_{\text{début}} + h^N_{\text{fin}}}{2} + \delta x \sum_{i=2}^{N-1} h^i_{\text{début}} \\ &= \delta x \frac{f(x_1) + f(x_N)}{2} + \delta x \sum_{i=2}^{N-1} f(x_i) \end{aligned}$$

En comparant avec la méthode des rectangles, on voit que la seule différence consiste à diviser par deux le premier et le dernier terme de la somme. Voyons dans l'exemple ci-dessus si le résultat en est amélioré :
Exemple 1.8. *Calculer l'intégrale de la fonction x^2 entre 0 et 1, par la méthode des trapèzes. On part toujours du tableau suivant :*

x	*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1*
x^2	*0*	*0.01*	*0.04*	*0.09*	*0.16*	*0.25*	*0.36*	*0.49*	*0.64*	*0.81*	*1*

Les δx_k sont tous égaux à 0.1*. On en a 11. On aura donc*

$$\int_0^1 x^2 \, dx = 0.1 \sum_{i=2}^{10} x_i^2 + \frac{0.1}{2}(1+0) = 0.3350.$$

Le résultat est déjà plus conforme à la réalité.

Exemple 1.9. *Comparer les résultats de l'intégrale de la fonction $x^3 - 1$ entre 0 et 1, par la méthode des rectangles et des trapèzes. On part du tableau suivant :*

x	*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1*
$x^3 - 1$	*-1.000*	*-0.999*	*-0.992*	*-0.973*	*-0.936*	*-0.875*	*-0.784*	*-0.657*	*-0.488*	*-0.2710*	*0*

Les δx_k sont toujours tous égaux à 0.1*. On aura donc*

$$I_{\text{rectangle}} = \int_0^1 x^3 - 1 \, dx = 0.1 \sum_{i=1}^{10} (x_i^3 - 1) = -0.7975.$$

$$I_{\text{trapèze}} \int_0^1 x^3 - 1 \, dx = 0.1 \sum_{i=2}^{10} (x_i^3 - 1) + \frac{0.1}{2}(1+0) = -0.7475.$$

On peut calculer une primitive de $f : x \mapsto x^3 - 1$, ce qui permet d'obtenir la valeur exacte de l'intégrale :

$$\int_0^1 x^3 - 1 \, dx = \left[\frac{x^4}{4} - x \right]_0^1 = -0.75$$

Le résultat de la méthode des trapèzes est toujours meilleur.

Les deux méthodes ci-dessus sont exactes pour des polynômes de degré 1 (des droites, quoi !). Pour des fonctions plus compliquées, on peut prendre des formules plus complexes.

1.3.3 La formule de Simpson

La formule de Simpson est faite pour intégrer exactement les polynômes de degré 2 ou moins, dans le cas d'un nombre impair de points. On aura :

$$I_{\text{Simpson}} = \frac{1}{3} (y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots + 2y_{n-2} + 4y_{n-1} + y_n)$$

Testons, sur les exemples vus ci-dessus, les performances de cette méthode.

Exemple 1.10. *Calculer l'intégrale de la fonction x^2 entre 0 et 1, par la méthode des trapèzes. On part toujours du tableau suivant :*

x	*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1.0*
x^2	*0*	*0.01*	*0.04*	*0.09*	*0.16*	*0.25*	*0.36*	*0.49*	*0.64*	*0.81*	*1.00*

Les δx_k *sont tous égaux à* 0.111. *On aura donc*

$$\int_0^1 x^2 \, dx = \frac{0.1}{3}(y_0 + 4y_1 + 2y_2 + 4y_2 + 2y_4 + 4y_5 + 2y_6 + 4y_7 + 2y_8 + 4y_9 + y_1 0) = 0.3333.$$

Le résultat est bien exact.

Exemple 1.11. *Calculer l'intégrale de la fonction $x^3 - 1$ entre 0 et 1, par la méthode des trapèzes. On part toujours du tableau suivant :*

x		*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1.0*
$x^3 - 1$	*-1.000*	*-0.999*	*-0.992*	*-0.973*	*-0.936*	*-0.875*	*-0.784*	*-0.657*	*-0.488*	*-0.2710*	*0*	

On trouve alors

$$\int_0^1 x^3 - 1 \, dx = \frac{0.1}{3}(y_0 + 2y_1 + 4y_2 + 2y_3 + 4y_4 + 2y_5 + 4y_6 + 2y_7 + 4y_8 + y_9) = -0.75.$$

Le résultat est exact aussi, dans ce cas.

1.3.4 L'intégration numérique sous matlab

Exemples

Exemple 1.12. *Calculer l'intégrale de* $\frac{x^3-1}{x^3+1}$ *entre 0 et 1, par les méthodes des rectangles, des trapèzes et de Simpson.*
Commençons par nous donner x *et* $f(x)$*.*

```
x=0:0.1:1;
y=(x.^3-1)./(x.^3+1);
```

Ensuite, on peut calculer l'intégrale par simple somme. Pour les rectangles, c'est immédiat :

```
>> I=sum(y*0.1)

I =

   -0.7200
```

Pour les trapèzes, il suffit de retirer la moitié du premier et du dernier :

```
>> I=sum(y*0.1)-0.05*(y(1)+y(end))

I =

   -0.6700
```

Pour Simpson, c'est un poil plus compliqué :

```
>> I=1/3*(y(1)+y(end)+4*sum(y(2:2:end-1))+2*sum(y(3:2:end-2)))*0.1

I =

   -0.6713
```

La solution analytique de cette intégrale n'est pas vraiment jolie :

$$x - \frac{2}{3}\ln(1+x) + \frac{1}{3}ln(x^2-x+1) - \frac{2\sqrt{3}}{3}\arctan(\frac{\sqrt{3}}{3}(2x-1))$$

Entre 0 et 1, la valeur numérique de l'intégrale est -0.6712976974.

Exemple 1.13. *Calculer l'intégrale de la fonction* $f(x) = \arctan\frac{e^{x^2}}{\cos 3x}$ *pour* x *entre 0 et 1.*
On calcule les x *et les* $f(x)$*, puis on utilise la méthode de Simpson :*

```
>> x=0:0.1:1;
>> y=atan((exp(x.^2))./(cos(3*x)));
>> I=1/3*(y(1)+y(end)+4*sum(y(2:2:end-1))+2*sum(y(3:2:end-2)))*0.1

I =

    0.0792
```

Cette intégrale n'a pas de solution analytique, et sa solution numérique est -0.05612530377. Dans ce cas-ci l'approximation n'est pas terrible. Le problème vient de la fonction elle-même, dessinez-là, et vous comprendrez... Le pas que l'on a choisi est trop petit. Avec un pas de 0.01, on obtient -0.0570, et avec un pas de 0.001, -0.05591.

Exercices

Exercice 1.3.1. Vérifier numériquement les intégrales suivantes :

1.
$$\int_0^1 xe^x \, dx = 1$$

2.
$$\int_{-1}^1 \sin x \cos x \, dx = 0$$

3.
$$\int_{-\infty}^{\infty} \begin{cases} \frac{\sin x}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases} dx = \pi$$

4.
$$\int_{-1}^1 \int_0^x (x^2 + y^2) \, dy \, dx = 0$$

   Attention, dans cet exemple, il faut parfois créer y avec un pas positif et parfois avec un pas négatif. La méthode des trapèzes est, en outre, préférable à la méthode de Simpson, car on n'a pas toujours un nombre impair de valeurs.

5.
$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} (1 + \tan^2 x) \, dx = \infty$$

   Pour ceci, montrez que plus le pas diminue et plus on calcule l'intégrale proche des vraies bornes, plus l'intégrale est grande.

1.4 La résolution numérique d'équations différentielles

1.4.1 Méthode

Supposons que l'on ait à résoudre

$$\frac{dy(x)}{dx} - 3 = x^2, \quad \text{avec } y(0) = 0.$$

On cherche y sur [-1,1], avec un point tous les 0.1. Soient $x_i = i \times 0.1$ et $y_i = y(x_i)$. On sait que, en y_1, on peut approximer

$$\frac{dy(x)}{dx} \simeq \frac{y_1 - y_0}{x_1 - x_0}.$$

Par conséquent, l'équation différentielle peut s'écrire

$$\begin{aligned} \frac{y_1 - y_0}{x_1 - x_0} - 3 &= x^2 \\ \frac{y_1}{0.1} &= 0.1^2 + 3 \\ 10y_1 &= 3.01 \\ y_1 &= 0.301 \end{aligned}$$

Similairement,

$$\begin{aligned} \frac{y_2 - y_1}{x_2 - x_1} - 3 &= x^2 \\ \frac{y_2}{0.1} &= 0.2^2 + 3 + \frac{0.301}{0.1} \\ 10y_2 &= 0.304 + 0.301 \\ y_2 &= 0.605 \end{aligned}$$

etc.

La solution analytique de ce problème est

$$y(x) = \frac{x^3}{3} + 3x.$$

On obtient le tableau suivant :

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$y(x)$ numérique	0	0.301	0.605	0.914	1.230	1.555	1.891	2.240	2.604	2.985	3.385
$y(x)$ analytique	0	0.300	0.603	0.909	1.221	1.542	1.872	2.214	2.571	2.943	3.333

La Figure 1.4 compare la solution analytique à la solution numérique. Toutefois, on note une diver-

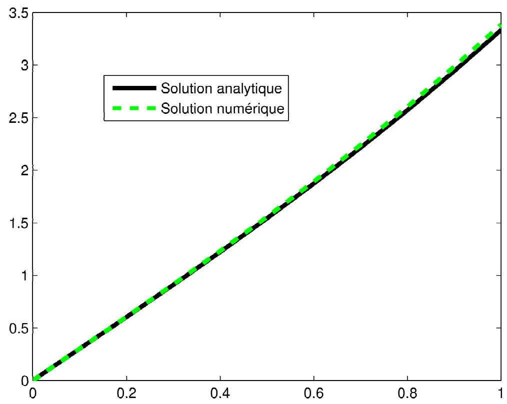


FIGURE 1.4 – Comparaison des solutions analytique et numérique de l'équation $\frac{dy(x)}{dx} - 3 = x^2$.

gence vers la fin. Une partie du problème tient à ce qu'on évalue la dérivée, en réalité, en $x - \frac{\Delta x}{2}$, et pas en x. Pour éviter ce problème, on va travailler au point $x + \frac{\Delta x}{2}$. En ce point, la dérivée de y se calcule uniquement par des points où l'on a des mesures :

$$\frac{dy(x + \frac{\Delta x}{2})}{dx} = \frac{y(x) - y(x + \Delta x)}{\Delta x}.$$

Dès lors,

$$y(x + \Delta x) = y(x) + \Delta x \frac{dy(x + \frac{\Delta x}{2})}{dx}.$$

Le problème, c'est qu'on ne connaît pas encore $\frac{dy(x+\frac{\Delta x}{2})}{dx}$. On peut toujours écrire l'équation différentielle comme

$$\frac{dy(x)}{dx} = f(x, y).$$

On l'évalue alors la dérivée de y un demi-pas plus loin par un développement de Taylor à l'ordre 1 en Δx, et on obtient, après quelques calculs élémentaires :

$$y_{i+1} = y_i + \Delta x \; f(x_i + \frac{\Delta x}{2}, y_i + \frac{\Delta x}{2} f(x_i, y_i)).$$

Dans l'exemple précédent, c'est facile, puisque $f(x,y) = f(x) = x^2 + 3$. On aura donc :

$$\begin{aligned}
y_1 &= y_0 + 0.1 \times f(0.05) \\
&= 0 + 0.1(0.05^2 + 3) \\
&= 0.3003 \\
y_2 &= y_1 + 0.1 \times f(0.15) \\
&= 0.3003 + 0.3023 \\
&= 0.6026
\end{aligned}$$

On obtient finalement :

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$y(x)$ numérique	0	0.300	0.603	0.909	1.221	1.541	1.872	2.214	2.570	2.942	3.333
$y(x)$ analytique	0	0.300	0.603	0.909	1.221	1.542	1.872	2.214	2.571	2.943	3.333

Le résultat par cette méthode est nettement meilleur.

Exemple 1.14. *Résoudre l'équation*

$$\frac{dy(x)}{dx} = y^2 + x + 1$$

numériquement entre 0 et 1, avec $y(0) = 0$.
On applique la méthode ci-dessus, mais avec $f(x,y) = y^2 + x + 1$.

$$\begin{aligned}
y_1 &= y_0 + 0.1\left((y_0 + \frac{0.1}{2} f(0,0))^2 + (0 + \frac{0.1}{2}) + 1\right) \\
&= 0 + 0.1\left((0 + \frac{0.1}{2}(0+0+1))^2 + 1.05\right) \\
&= 0.1 \times 1.0525 = 0.1053 \\
y_2 &= y_1 + 0.1\left((y_1 + \frac{0.1}{2} f(0.1, 0.1053))^2 + (0.1 + \frac{0.1}{2}) + 1\right) \\
&= 0.1053 + 0.1\left((0.1053 + 0.05(0.1053^2 + 0.1 + 1))^2 + 1.15\right) \\
&= 0.1053 + 0.1(0.1609^2 + 1.15)) = 0.1053 + 0.1176 = 0.2229
\end{aligned}$$

On trouve finalement :

x	*0*	*0.1*	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*	*0.7*	*0.8*	*0.9*	*1.0*
$y(x)$ *numérique*	*0*	*0.105*	*0.223*	*0.356*	*0.509*	*0.689*	*0.906*	*1.177*	*1.529*	*2.016*	*2.746*
$y(x)$ *analytique*	*0*	*0.105*	*0.223*	*0.357*	*0.510*	*0.691*	*0.910*	*1.185*	*1.544*	*2.049*	*2.83*

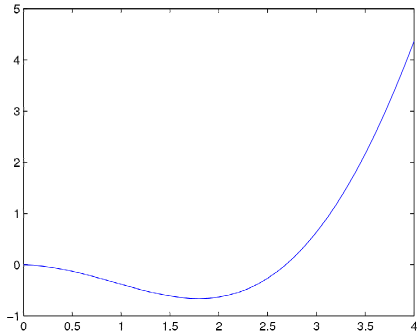


FIGURE 1.5 – LA POSITION EN RÉPONSE À UNE ACCÉLÉRATION $a = t - cos(t)$.

1.4.2 Résolution numérique d'équations différentielles sous Matlab

Exemples

Exemple 1.15. *Calculer la position d'une particule, au départ au repos, soumise à l'accélération $a = t - cos(t)$. On intègre une première fois pour obtenir la vitesse, puis une seconde pour la position. Si l'on prend la méthode la plus simple, on obtient le résutat de la Figure 1.5*

```
t=0:0.1:4;
a=t/2-cos(t);
v(1)=0; x(1)=0;
for i=2:length(t)
    v(i)=v(i-1)+a(i)*0.1;
    x(i)=x(i-1)+v(i)*0.1;
end
plot(t,x)
```

Si l'on travaille avec le point millieu, on aura :

```
t=0.05:0.10:4.05;
a=t/2-cos(t);
v(1)=0; x(1)=0;
for i=2:length(t)
    v(i)=v(i-1)+a(i-1)*0.1;
    x(i)=x(i-1)+v(i-1)*0.1;
end
```

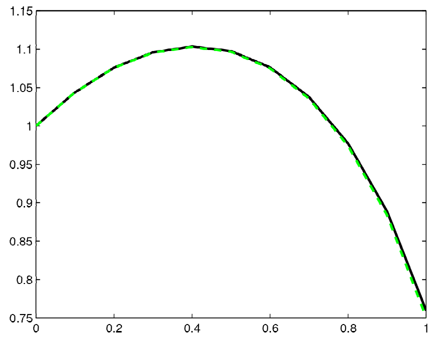


FIGURE 1.6 – SOLUTIONS NUMÉRIQUE ET ANALYTIQUE DE L'ÉQUATION DIFFÉRENTIELLE $\frac{dy}{dt} = \frac{y^2-3t}{2y}$.

Exemple 1.16. *Intégrer l'équation différentielle suivante pour t entre 0 et 1.*

$$\frac{dy}{dt} = \frac{y^2 - 3t}{2y},$$

avec $y(0) = 1$. La solution analytique est $y = \sqrt{3 + 3t - 2e^t}$. On écrit le programme suivant :

```
t=0:0.1:1;
y(1)=1;
for i=2:length(t)
    f=(y(i-1)^2-3*t(i-1))/2/y(i-1);
    yb=y(i-1)+0.05*f;
    y(i)=y(i-1)+0.1*(yb^2-3*(t(i-1)+0.05))/2/yb;
end
plot(t,y,'k',t,sqrt(3+3*t-2*exp(t)),'g--','linewidth',2)
```

qui donnera le joli résultat représenté sur la Figure 1.6

Exercices

Exercice 1.4.1. Comparez le résultat numérique des résolutions d'équations différentielles (entre 0 et 1) à la solution analytique :

1. $\frac{dy}{dt} + 3t = 1$, avec $y(1) = 0$. $\left(\text{Solution : } y(t) = -\frac{3}{2}t^2 + t + 1/2\right)$.
2. $\frac{dy}{dt} + y^2 = 1$, avec $y(0) = 0$. (Solution : $y(t) = \tanh t$).
3. $\frac{dy}{dt} + 5\cos(y + t) = 0$, avec $y(0) = 0$. $\left(\text{Solution : } y(t) = -t - 2\arctan\left(\frac{\sqrt{21}}{7}\tanh(\frac{\sqrt{21}}{2}t)\right)\right)$.
4. $\frac{dy}{dt} + \frac{dy}{dx} = y$, avec, comme conditions aux bords $y(0, x) = x$ et $y(t, 0) = -te^t$ (Solution : $y(t, x) = (x - t)e^t$).

Chapitre 2

La lecture de données sous MATLAB

2.1 Les données formatées

2.1.1 Le cas le plus simple : les données sans entêtes, rangées en colonnes

Le cas le plus simple, c'est un fichier avec seulement les données (pas d'entête) rangées par colonnes, avec un nombre constant de nombres par ligne, tout le long du fichier. Par exemple, prenons le fichier `d1.dat`

```
1.0 5.45
1.5 2.34
2.0 9.48
... ...
```

Dans ce cas, la lecture se fait simplement par la commande `a=load('d1.dat')`. Si on ne spécifie pas le nom de la variable, le logiciel, de lui-même, crée une variable selon le nom du fichier. Dans le cas présent, il créera la variable `d1`. Si il y a quelques lignes d'en-tête, on peut les passer en les faisant précéder d'un symbole `%`.

Note : pour écrire des données formatées dans un fichier, la commande à utiliser est `fprintf`.

2.1.2 Ca se complique un peu, le nombre de colonnes n'est pas constant

Parfois, les données sont formatées avec un nombre de colonnes non constant. C'est par exemple le cas si l'on écrit une carte du monde avec une résolution de 2 degré par 2 degré, sur 8 colonnes. Chaque "ligne" de la carte fait 180 points, soit donc 22 lignes du fichier, plus 4 valeurs. Donc, toutes les 23 lignes, il y en aura une avec seulement 4 colonnes, ce que le `load` ne peut pas gérer. Il faut alors utiliser la commande `fscanf` qui permet les lectures formatées. De toutes les commandes MATLAB que je connais, c'est celle qui est la plus mal programmée, du point de vue de la souplesse. Malheureusement, il n'y a le plus souvent pas d'autre possibilité que de l'utiliser le moins mal possible. Dans le cas mentionné ci-dessus, voici la syntaxe.

```
fid=fopen('monfichier.dat'); % On ouvre le fichier pour la lecture.
map=fscanf(fid,'%f',[180,91]); % On lit dans le fichier 91 lignes de 181 données
fclose(fid); % On ferme le fichier.
```

Quelques remarques :

- Contrairement à d'autres langages, MATLAB ne va pas à la ligne après exécution de l'instruction (ni en lecture ni en écriture). Donc, on ne peut pas, et c'est très génant, lire juste les 2 premières colonnes et passer à la ligne suivante.
- Le format se met entre ' avec un format par colonne à lire. La liste de tous les formats possibles avec les symbôles correspondants sont donnés dans l'aide en ligne.

TABLE 2.1 – Les principaux formats de lecture/écriture dans MATLAB

%c	simple caractère
%d	nombre décimal (avec signe)
%e	nombre décimal en notation exponentielle (3.1415e+00)
%f	nombre décimal avec nombre de décimales fixé
%i	entier (avec signe)
%s	chaîne de caractères

- Souvent, le programme aura un problème de lecture au milieu d'une boucle, et donc avant la fermeture du fichier. Le pointeur n'est pas remis à ce moment au début du fichier. On peut, par exemple, commencer par fermer le fichier avant de le rouvrir.
- Attention, MATLAB lit *à l'envers*, puisque la matrice apparaît transposée. Cependant, comme elle relit de la même façon, cela ne prête pas à conséquence.

2.1.3 Les données hétérogènes par colonnes

Supposons cette fois que j'ai des données formatées, mais dont les formats sont différents d'une colonne à l'autre. Cela peut, parfois, ne pas poser de problème. Par exemple

```
1.0 3 5.45
1.5 5 2.34
2.0 1 9.48
```

se lira bien avec fscanf(fid,'%f',[3,3]). Par contre, si l'on a

```
Pierre 1.0 3 5.45
Paul 1.5 5 2.34
Jacques 2.0 1 9.48
```

cela ne marchera pas. Deux solutions possibles :

```
fid=fopen('d1.dat');
for i=1:3
  nom=fscanf(fid,'%s',1)
  line(i,:)=fscanf(fid,'%f',3)
end
```

ou alors, on peut utiliser la commande textread.

```
>> [name,l1,l2,l3]=textread('d1.dat','%s %f %f %f')

name =

    'Pierre'
    'Paul'
    'Jacques'


l1 =

    1.0000
    1.5000
    2.0000
```

```
l2 =

     3
     5
     1


l3 =

    5.4500
    2.3400
    9.4800
```

2.1.4 Le problème des dates

Souvent, la mesure est associée à un indice de temps, par exemple une date. Souvent, à ce moment-là, les ennuis commencent. En effet, la plupart des méthodes d'analyses ont besoin, sinon de données échantillonnées régulièrement, de données dont le temps varie de façon continue.

Exemple 2.1. Supposons que l'on ait des données sur un jour en heure-minute-seconde, qu'en fait-on ? La solution consiste à tout ramener sur un même indice de temps. Par exemple, en seconde. On aura alors :

```
aa=load('data.txt');
heure=aa(:,1);
minute=aa(:,2);
seconde=aa(:,3);
sec=seconde+minute*60+heure*3600;
data=aa(:,4);
```

On peut aussi tout reconvertir en heures ou en jours décimaux.

Exemple 2.2. Soit un fichier qui comprend année jour mois donnée. C'est un peu moins simple, parce que le nombre de jours par mois est un peu pénible. Une solution simple est de chercher sur l'internet une fonction qui convertit des dates en un format continu (jour julien modifié, par exemple). Sinon, il faut un peu chipoter :

```
nj=[0,31,28,31,30,31,30,31,31,30,31,30]; % Nombres de jours écoulés à la fin du mois
aa=load('data.txt');
annee=aa(:,1);
jour=aa(:,2);
mois=aa(:,3);
data=aa(:,4);
for j=1:length(annee)
   if (mod(annee(j),4)==0) % On considère qu'on travaille entre 1901 et 2099.
     nj(2)=29;
     njy=366;
   else
     nj(2)=28;
     njy=365;
   end
   ny(j)=annee(j)-1960;
   snj=sum(nj);
   yd(j)=annee+(snj(mois)+jour)/(njy);
   d1960(j)=ny*365+fix(ny/4)+snj(mois)+jour;
end
```

La première solution, en année décimale, a l'avantage de la lisibilité. Elle a l'inconvénient que des données équidistantes ne le sont plus, puisqu'elles seront plus espacées dans une année non-bissextile.

2.2 La lecture automatique

Exemple 2.3. Parfois, on reçoit 100 fichiers du même type, avec un nom qui change peut. Par exemple, le fichier s'appelle donnees.mois.année.txt. Le premier s'appelle donnees.janv.1950.txt, puis on a donnees.fev.1950.txt, etc... Il est hautement préférable de ne pas les ouvrir à la main, ce qui entraine facilement des erreurs de manipulation. La meilleure solution est de créer un programme de lecture automatique.

```
mois=['jan';'fev';'mar';'avr';'mai';'jui';'jul';'aou';'sep';'oct';'nov';'dec'];
for i=1950:2000
   for j=1:12
      name=['donnees.',mois(j,:),'.',num2str(i),'.txt'];
      fid=fopen(name);
      ...
      ...
      fclose(fid);
   end
end
```

Notez (1) les accolades pour concaténer les variables caractères. ['pif','paf']='pifpaf', ['pif';'paf']= $\begin{array}{c}\text{pif}\\\text{paf}\end{array}$; (2) l'usage du num2str(i) qui transforme la variable i (numérique) en son équivalent caractère.

Exemple 2.4. Supposons que vous deviez recevoir un grand nombre de fichiers de données. Pour chaque fichier, vous avez besoin de l'information de la latitude, la longitude et la technique d'observation, plus des données au cours du temps. La méthode la plus simple est la suivante. On commence par nommer (ou faire nommer) les fichiers avec un nom qui comprend les infos autres que les données, par exemple data.lon.lat.tech.txt, avec 3 chiffres pour la longitude, 2 chiffres et une lettre (N ou S) pour la latitude, un chiffre pour la technique. Supposons un même nombre de données par fichier. On peut alors utiliser l'algoritme suivant :

```
gg=ls('data.*.txt'); %met dans gg une matrice des noms de fichiers.
for i=1:size(gg,1)
   lon(i)=str2num(gg(i,6:8));
   lat(i)=str2num(gg(i,10:11));
   if (gg(i,12)=='S')
     lat(i)=-lat(i);
   end
   tech(i)=str2num(gg(i,14));
   aa=load(gg(i,:));
   data(i,:)=aa(:,1)';
   ...
end
```

Notons (1) l'usage du nom de fichier informatif qui permet d'utiliser le load pour charger les données, (2) l'usage du str2num qui convertit en nombre les caractères du nom du fichier. Dans le cas où je ne veux traiter que les données obtenues avec la technique 3, je ferai simplement : data3=data(find(tech==3), :) ;.

2.3 Exercices sur la lecture élémentaire

Vous aurez sans doute besoin des fonctions suivantes :
- load et textread pour la lecture des données ;

- plot, semilogy, pcolor pour les représentations graphiques ;
- log, exp, abs pour le traitement des données ;
- interp1 pour l'interpolation des données.

Exercice 2.3.1. A l'aide du fichier *dipole_ref.txt*, représenter l'inclinaison (colonne 2) et l'amplitude (colonne 3) du dipôle magnétique en fonction du temps (colonne 1) dans deux sous-fenêtres distinctes.

Exercice 2.3.2. A l'aide des fichiers *dipole_ref.txt* (séquence de référence) et *dipole_pert.txt* (séquence perturbée), calculer la différence entre les amplitudes du dipôle lorsque cela est possible. Représenter la valeur absolue de cette différence dans un diagramme semi-logarithmique.

Exercice 2.3.3. A l'aide du fichier *spectre.txt*, représenter l'amplitude des modes harmoniques du champ magnétique (colonne 3) en fonction de leurs indices (colonne 2). A l'aide d'une régression linéaire, calculer l'atténuation du spectre. On peut définir l'atténuation comme la pente du logarithme de l'amplitude des modes en fonction de leurs indices.

Exercice 2.3.4. Le fichier *anomalie.txt* contient les données d'une carte avec le repère suivant l'axe des x dans la première colonne, le repère suivant l'axe des y dans la deuxième colonne et la valeur de la mesure dans la troisième colonne. Après avoir réorganisées les données sous forme matricielle, représenter graphiquement la carte à l'aide d'un pcolor.

2.4 Exercices corrigés

Exercice 2.4.1. Le fichier lecture1.txt contient les données de l'indice du El-Niño. La première colonne contient l'année, et les 12 suivantes les valeurs mensuelles de cet indice. Ecrire un programme qui lit et dessine ces données en fonction du temps (exprimé en années décimales). Nous négligerons les années bissextiles dans ce problème.

```
clear all                                  % On efface la mémoire
aa=load('lecture1.txt');                   % On charge l'ensemble du ficher dans la variable aa
SOI=reshape(aa(:,2:13)',1,1692);             % Le SOI, ce sont les colonnes 2 à 13
% on transforme les dimensions. On impose que la première dimension = 1
dm=[1,31,28,31,30,31,30,31,31,30,31,30];   % On définit l'indice de temps
sdm=(cumsum(dm)+15)/365;                   % ici aussi. On prend le milieu du mois comme repère
for j=1:12
   t(j:12:12*size(aa,1))=aa(:,1)'+sdm(j);
end
t=t(SOI>-5);                               % Les valeurs -10 indiquent qu'il n'y a pas de données. On
SOI=SOI(SOI>-5);
plot(t,SOI)                                % Et on dessine.
title('Le SOI en fonction du temps')
xlabel('Temps (année)')
ylabel('SOI')
```

Notons que l'on peut, en MATLAB, le laisser choisir la seconde dimension du reshape, en mettant SOI=reshape(aa(:,2 :13)',1,[]) ;, mais octave n'accepte pas cette facilité.

Exercice 2.4.2. Le fichier lecture2.pdf comprend un tableau de données. Importez-le dans matlab et dessinez-en les différentes séries temporelles. Voila une exemple courant. Les données sont disponibles sous format pdf, et on voudrait éviter de retaper le tableau, bien sûr. Le travail se fait en plusieurs étapes :

1. On utilise la fonction de saisie de texte de l'acrobat reader (voir figure 2.1), pour prendre les données et les entêtes de colonnes. C'est un peu de travail parce que les entêtes sont en général rentrés, comme c'est le cas ici, dans le mauvais sens.
2. On commente ces deux lignes-là dans le tableau en les faisant précéder d'un sympole %.
3. On sauve le fichier avec un joli nom, genre lecture2.txt.

Figure 2.1 – Bouton de saisie de text sur acrobat reader.

4. On essaye comme ça, et cela rate. En effet, les données ont été entrées avec un espace entre le 1 et les chiffres suivants pour les valeurs supérieures à mille. On corrige ce problème. On resauve.

A ce moment-là, on peut utiliser un programme de lecture tout simple, du type :

```
aa=load('c:/données/cours/lecture2.txt');
annee=aa(:,1);
data=aa(:,2:end);
plot(annee,data,'o')
```

Notons que, MATLAB, par défaut, changera de couleur pour chaque colonne, alors qu'il faudra l'imposer dans OCTAVE.

Exercice 2.4.3. Le fichier lecture3.txt contient des données de rotation de la Terre. Les deux premières lignes donnent les informations sur les données. On est en 2006, et les données sont à minuit (temps universel (i.e. UTC)), et les colonnes sont les dates (mois, jours, et date julienne modifiée), le mouvement du pôle (en seconde de degré), x et y, les variations du taux de rotation de la Terre ($UT1$) est la mesure de l'orientation de la Terre, c'est le temps universel, UTC et $UT1R$ sont les valeurs de l'orientation de la Terre dans l'hypothèse où elle tournerait de façon régulière, D est la durée du jour (moins vingt-quatre heures), et $dPsi$ et $dEpsilon$ les angles de nutation de la Terre. Dessiner, dans le plan $x-y$ le mouvement du pôle en affichant les dates tous les 10 jours, en année décimale.

```
clear all                    % On vide la mémoire.
fid=fopen('c:/données/cours/lecture3.txt'); % On ouvre le fichier en lecture
aa=fscanf(fid,'%s',1);       % On lit le premier ensemble de caractères (qui sera 'JUN')
while(aa(1)~='J')            % Tant que la première lettre n'en est pas 'J'...
   aa=fscanf(fid,'%s',1);    % ... on continue à lire les ensembles de caractères...
end                          % ... et puis on arrete.
for k=1:55                   % On sait qu'il y a 55 lignes de données
    aa=fscanf(fid,'%i %i',2); % On lit les 2 entiers correspondants au jour du mois et à la MJD
    x(k)=fscanf(fid,'%f',1);  % On lit x(k)
    y(k)=fscanf(fid,'%f',1);  % On lit y(k)
    aa=fscanf(fid,'%f %f %f %f %f',5); % On lit le reste de la ligne...
    bb=fscanf(fid,'%s',1);    % et les caractères du mois suivant
end
t=2006+(31+28+31+30+31+2+(1:55))/365; % On définit le temps en année décimale
plot(x,y,'k.')                % On dessine x en fonction de y
for k=3:10:57                 % Une fois tous les 10 jours...
   text(x(k),y(k),num2str(t(k),6)) % ... on ajoute les labels de temps sur le graphe
end                                % le 6 du num2str indique la précision gardée.
```

Exercice 2.4.4. Le répertoire lecture4 contient 258 fichiers qui donnent, chacun, les coefficients des bas degrés du champ de pesanteur pour un mois donné. Le nom du fichier est composé du préfixe grvfld.ref, suivi de deux chiffres pour l'année, deux chiffres pour le mois, et deux chiffres pour le jour de référence. Le but de l'exercice est de représenter l'évolution au cours du temps de l'un de ces coefficents.

– Commençons par la lecture d'un seul fichier.

```
clear all
name='grvfld.ref991228';
jour=str2num(name(end-1:end));   % On lit les infos dans le nom de fichier
mois=str2num(name(end-3:end-2));
an=str2num(name(end-5:end-4));
```

```
if (an>10)
    an=an+1900;
else
    an=an+2000;
end
dm=cumsum([0,31,28,31,30,31,30,31,31,30,31,30]);
t=an+(dm(mois)+jour)/365;        % Et on génère le temps
fid=fopen(name);        % On ouvre le fichier en lecture
aa=fscanf(fid,'%s',1);  % On lit le premier mot, qui commence par G
aa=fscanf(fid,'%s',1);  % Les données commencent au second mot qui commence par G
while(aa(1)~='G')
   aa=fscanf(fid,'%s',1);
end
while(length(aa)>0)              % tant qu'on est pas à la fin du fichier...
   l=fscanf(fid,'%i',1);         % on lit l
   m=fscanf(fid,'%i',1);         % et m
   if (aa(6)=='C')               % si c'est un coefficient 'C'os
       C(l+1,m+1)=str2num(fscanf(fid,'%s',1));  % on le met en partie réelle
   else                          % sinon
       C(l+1,m+1)=C(l+1,m+1)+i*str2num(fscanf(fid,'%s',1)); % on le met en partie imaginaire
   end
   aa=fscanf(fid,'%s',1);        % et on lit le ``GCOEF suivant''
end
```

– Maintenant, on peut automatiser pour les lire tous d'un coup. (

```
clear all
gg=ls('gr*');                       % On met tous les noms de fichiers dans un tableau...
for k=1:size(gg,1);                 % On on les prends tous les uns apres les autres...
   name=gg(k,:);
   ...
   t(k)=an+(dm(mois)+jour)/365;     % On crée un vecteur temps
   fid=fopen([chemin,name]);
   ...
     if (aa(6)=='C')
         C(l+1,m+1,k)=str2num(fscanf(fid,'%s',1));  % Et la matrice C a maintenant 3 dimensions
     else
         C(l+1,m+1,k)=C(l+1,m+1,k)+i*str2num(fscanf(fid,'%s',1));
     end
     ...
   end
   fclose(fid);
end
```

Exercice 2.4.5. Sur le site http ://www.albumdufutur.com/lda/lda.php3 se trouve les dates de naissance, noms et profession de personnalités. Copier une page de ces données dans un fichier texte. Ecrire un programme qui lit ces données, et crée des variables nom, profession, qui contiennent les chaines de caractères, et les variables jours, mois, années. Trouver les professions des gens nés entre midi et 17 heures.

– Commençons par ouvrir un fichier éditeur. On copie dans ce fichier la partie de la page contenant les informations.
– Il faut un petit peu éditer, de façon à n'avoir qu'une seule chaine de caractère pour le nom, et une seule pour le lieu de naissance. On obtient alors le fichier lecture5.txt.
– Comme précédement, on commence par lire un seul enregistrement.

```
fid=fopen('c:\données\cours\lecture5.txt');     % On ouvre le fichier en lecture
nom=fscanf(fid,'%s',1);                         % On lit le nom
prenom=fscanf(fid,'%s',1);                      % On lit la ville...
```

```
ville=fscanf(fid,'%s',1);
pays=fscanf(fid,'%s',1);
date=fscanf(fid,'%s',1);
heure=fscanf(fid,'%s',1);
jour=str2num(date(1:2))+(str2num(heure(1:2))+str2num(heure(4:5))/60)/24; %On combine les infos
mois=str2num(date(4:5));
annee=str2num(date(7:8));
profession=fscanf(fid,'%s',1);                % et puis c'est tout...
```

– Maintenant, il s'agit de faire une liste avec tout le monde... On a envie de faire simplement :

```
while(length(nom)~=0)
nom(k,:)=fscanf...
```

mais cela ne marche pas, car les noms n'ont pas la même longueur. On peut, par exemple, utiliser l'algoritme suivant :

```
clear all
fid=fopen('c:/données/cours/lecture5.txt');
nom=fscanf(fid,'%s',1);
k=1;
while(length(nom)~=0)
    n(k,1:length(nom))=nom;       % ceci permet de ne pas s'inquiéter de la longueur des noms.
    prenom=fscanf(fid,'%s',1);
    ville=fscanf(fid,'%s',1);
    pays=fscanf(fid,'%s',1);
    date=fscanf(fid,'%s',1);
    heure=fscanf(fid,'%s',1);
    jour(k)=str2num(date(1:2))+(str2num(heure(1:2))+str2num(heure(4:5))/60)/24;
    mois(k)=str2num(date(4:5));
    annee(k)=str2num(date(7:8));
    sexe=fscanf(fid,'%s',1);
    profession=fscanf(fid,'%s',1);
    p(k,1:length(profession))=profession;
    nom=fscanf(fid,'%s',1);
    k=k+1;
end
p(find(mod(jour,1)>12/24&mod(jour,1)>17/24),:) % on récupère les parties décimales de jours ent
```

Exercice 2.4.6. Le fichier lecture6.bin contient des données du tracé des côtes dans la région de Sumatra, en binaire. Les lire et les représenter. Les données sont écrites comme une suite de paires longitude-latitude, en "little endian", codée comme float sur 32 bits.

```
fid=fopen('lecture6.bin','r','l');  %On ouvre en lecture. Le 'r' spécifie l'autorisation en ...
for k=1:62633                       % ... lecture, le 'l' le little endian
    Lon(k)=fread(fid,1,'float');    % on lit en spécifiant l'unité de lecture, le nombre et ...
    Lat(k)=fread(fid,1,'float');    % ... le format des données
end
plot(Lon,Lat,'.')
```

2.5 Exercices non corrigés

Exercice 2.5.1. Le fichier lecture7.txt contient une carte des hauteurs d'eau pour la marée M2. La représenter en fonction de la latitude et de la longitude. Remplacer les données manquantes (999.00) par zéro. *Indication : la fonction* pcolor(x,y,z) *fait de jolis dessins. Faire suivre de* shading flat.

Exercice 2.5.2. Représenter l'indice Nino1 en fonction du temps. La première colonne du fichier lecture8.txt donne le temps de début et de fin de la donnée (format yyyymmddyyyymmdd), et la seconde la valeur.

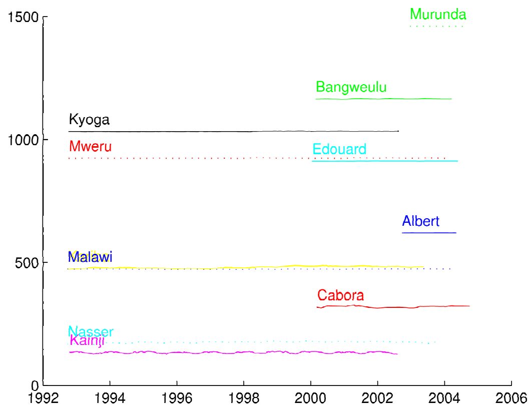


FIGURE 2.2 – RÉSULTAT QUE J'AI OBTENU POUR L'EXERCICE 12.12

Exercice 2.5.3. Le fichier lecture9.txt contient les dates (années décimales), les fluctuations de la longueur du jour (LOD), et l'erreur a priori sur ces valeurs. Représenter le LOD en fonction du temps avec ces bornes d'erreurs. *Indication : voir la fonction* errorbar.

Exercice 2.5.4. Le fichier lecture10.txt contient des données atmosphériques. Représenter l'évolution au cours de temps de la moyenne à chaque latitude (donc moyenne sur la longitude) du vent u.

Exercice 2.5.5. Ecrire un programme qui crée 100 fichiers des valeurs des polynomes de degré 2 entre -1 et 1, avec les coefficients variant de 1 à 10 pour a et b, et $c = 1$ dans l'expression $ax^2 + bx + c$. Le nom du fichier doit comporter l'indication de la valeur de a et de b. Chaque fichier comprendra 2 colonnes, les valeurs de x et les valeurs du polynome.

Exercice 2.5.6. Le répertoire lecture 12 contient des fichiers de données d'évolution de la hauteur d'eau en fonction du temps pour un certain nombre de rivières africaines. Représentez sur un même graphe ces évolutions avec des lignes différentes pour chaque courbe, en indiquant le nom de chaque rivière près du début de la courbe correspondante, dans la couleur appropriée. Le résultat devrait ressembler à la figure 2.2.

Exercice 2.5.7. Le fichier Bathy_EPR9N.xyz contient les données de bathymétrie d'un segment de la dorsale Est-Pacifique autour de 9°N. La première colonne est la longitude, la seconde donne la latitude, et la dernière la profondeur. Lire ces données et les représenter sur une carte (toujours avec pcolor).