Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 1 : Eléments de base de Matlab

Tracés de courbes

1. Donnez les instructions Matlab permettant de tracer, sur un même graphe, les fonctions $f(x) = \sin(x)$ et $g(x) = x\cos(x)$ pour $x \in [0, 2\pi]$ (on prendra soin de découper l'intervalle de façon à obtenir des tracés visuellement "lisses"). Les axes devront porter des noms.

2. Pour tout $n \geq 1$, on pose $u_n = n^2 e^{-n}/(1+n)$. Donnez les instructions Matlab permettant de calculer les 100 premiers termes de cette suite (deux instructions maximum, sans boucle). Tracer u_n en fonction de n.

Suite de Fibonacci

On définit la suite de Fibonacci, telle que $u_1 = 1$, $u_2 = 1$, et $u_n = u_n + u_{n-1}$ pour tout $n \geq 3$.
Tracer cette suite en fonction de n.

Convergence d'un développement limité

Le développement limité de la fonction e^x est

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

On souhaite étudier la convergence de cette somme pour une valeur de x donnée, $x = 1$. On pose donc

$$e_N = \sum_{n=0}^{N} \frac{1}{n!}.$$

Tracer e_N en fonction de N pour N compris entre 1 et 20. (la fonction factorielle $n!$ est `factorial(n)`.)

Manipulation sur les matrices

1. Donner les instructions Matlab permettant de construire la matrice suivante (deux instructions au minimum) :

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 6 & 6 & 0 \\ 0 & 6 & 6 & 0 \end{pmatrix}$$

2. Ecrire le programme permettant de calculer les N premières lignes du triangle de Pascal :

$$\begin{array}{ccccc} 1 & & & & \\ 1 & 1 & & & \\ 1 & 2 & 1 & & \\ 1 & 3 & 3 & 1 & \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

   Le triangle de Pascal $P(i,j)$ est tel que $P(i,j) = P(i-1,j-1) + P(i-1,j)$ où $j \leq i$, avec $P(i,1) = 1$ pour tout i.

Quelques fonctions

Programmez les fonctions suivantes (et testez-les sur des exemples simples) :

1. `b = renverse(a)` : une fonction qui renvoie le vecteur `b` tel que ses éléments soient ceux de `a` ré-ordonnés à l'envers.
2. `m = geomean(x)` : une fonction qui renvoie la moyenne géométrique des éléments du vecteur `x` (c'est-à-dire $m = (x(1)x(2)..x(N))^{1/N}$, où N est la longueur de x).
3. `Y = polymorceau(X)` : une fonction qui renvoie un vecteur Y dont les éléments sont donnés par x^2 pour chaque élément $x \geq 0$ du vecteur X, et $-x$ pour chaque élément $x < 0$ de X. Tracer le résultat pour $x \in [-1, 1]$.

Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 2 : Evolution chaotique d'une population

Introduction : Evolution d'une population

On considère une population de bactéries, décrite par le nombre d'individus x_n à la $n-$ème génération. Ce nombre d'individus est normalisé de telle sorte que $x_n = 1$ corresponde à un milieu saturé. Cette population est décrite par la suite :

$$x_{n+1} = r\, x_n(1 - x_n),$$

où r est un réel entre 0 et 4, qui décrit le taux de croissance de la population en l'abscence de restrictions. On cherche dans ce TP à étudier l'évolution de cette population en fonction de r. On cherchera en particulier à identifier les situations de population stable (suite convergente), d'évolution périodique ou encore chaotique.

Etude des itérés successifs

1. Ecrire une fonction `X=population(r,N,x1)` qui renvoie le vecteur ligne `X` des N premiers itérés de la suite x commençant par $x1$ et de paramètre r.
2. Utiliser cette fonction pour représenter graphiquement x_n en fonction de n pour différentes valeurs de r entre 0 et 4. Observer qualitativement pour quelles valeurs de r la suite converge ou non (en cas d'ambiguité sur la convergence, augmenter la valeur de N). Que pensez-vous de l'influence de la condition intiale x_1 ?
3. Dans le cas d'oscillations périodiques, repérer, pour $3 < r < 3.6$, pour quelles valeurs de r la population change de périodicité (on relèvera au moins 5 valeurs de r pour lesquelles la période des oscillations de x_n passe d'une valeur T à la valeur double $2T$).
4. Pour des valeurs de r dans le régime périodique ou chaotique, représenter l'histogramme des valeurs de x_n, en utilisant la fonction `hist`. Comment se caractérise chacun des régimes ?
5. Facultatif : Représenter la suite $x = (x_n)_{n\geq 1}$ sous la forme d'un "diagramme en escargot" : pour cela, on tracera dans une même figure la courbe $f(x) = r\, x(1-x)$, la droite $y = x$ ainsi que la ligne brisée qui joint les points $(x_1, 0), (x_1, f(x_1)), (x_2, x_2), (x_2, f(x_2))...$

Diagramme de bifurcation

Pour synthétiser les observations précédentes, on va chercher à construire le *diagramme de bifurcation*, qui représente un ensemble de valeurs visitées par la suite x_n en fonction de r.

1. En utilisant la fonction `population` écrite précédemment, faire une boucle sur r, permettant d'afficher `n` valeurs de `x` pour chaque valeur de r. On fera varier r entre 0 et 4, par pas `dr` judicieusement choisis.
2. Pour des valeurs de r telles que la suite converge, il peut être intéressant de n'afficher que les valeurs de x_n atteintes asymptotiquement par la suite, afin de ne pas surcharger le diagramme de bifurcation avec les régimes transitoires. Que proposez-vous pour réaliser un tel diagramme ?
3. Observer plus précisemment le diagramme au voisinage d'un changement de régime, au voisinage de $r = 3$ par exemple. Que pensez-vous des régimes transitoires pour de telles valeurs de r ?

Sensibilité aux conditions initiales

Une propriété remarquable de cette suite dans le régime chaotique est sa "sensibilité aux conditions initiales" : partant de deux conditions initiales très proches, séparées de $\delta x_1 \ll 1$, l'écart entre les deux évolutions $\delta x_n = |x_n^{(2)} - x_n^{(1)}|$ augmente exponentiellement ave n. Mettez en évidence ce phénomène. Montrer graphiquement que, pour une précision donnée sur la condition initiale, il existe un nombre d'itérations N au-delà duquel le comportement de la suite est imprévisible.

Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 3 : Ondes propagatives, interférences

Ondes à 1 dimension

On considère dans un premier temps une onde propagative à 1 dimension, de longueur d'onde λ, de période T et d'ampliture A, se propageant vers les x croissants :

$$u(x,t) = A\cos(kx - \omega t) = A\cos\left(\frac{2\pi}{\lambda}(x - ct)\right).$$

$k = 2\pi/\lambda$ est le nombre d'onde et $\omega = 2\pi/T$ la pulsation. La grandeur u peut représenter la pression dans le cas d'une onde acoustique, une composante du champ électrique dans le cas d'une onde électromagnétique (la lumière par exemple), ou encore la déviation transverse d'une corde pour une onde mécanique. La vitesse de propagation (ou vitesse de phase) est donnée par $c = \omega/k = \lambda/T$. Dans le cas d'une onde "non dispersive" (cas considéré dans ce TP), cette vitesse de phase est une constante indépendante de la longueur d'onde.

1. Réaliser une animation de l'onde, en faisant varier le temps t par petits intervalles d'une image à l'aute. Entre deux images on utilisera la fonction `pause` (attend l'appui d'une touche), ou `drawnow` ("rafraîchit" la figure pour une animation en continu ; voir l'annexe).

2. Une onde stationnaire – par exemple la vibration d'une corde de guitare – est la superposition d'une onde propagative "droite" (de célérité c) et d'une onde propagative "gauche" (de célérité $-c$). Pouvez-vous le vérifier visuellement ? Que se passe-t-il si les deux ondes n'ont pas même amplitude ?

3. On peut modéliser le phénomène d'atténuation de l'onde, associé à une dissipation d'énergie lors de la propagation : on remplace dans l'équation précédente A par $A_0 e^{-x/x_0}$, soit

   $$u(x,t) = A_0 e^{-x/x_0}\cos\left(\frac{2\pi}{\lambda}(x - ct)\right),$$

   où x_0 est la longueur d'atténuation. Modifier l'animation précédente pour représenter une telle onde atténuée. On pourra observer l'onde pour x_0 petit, comparable ou grand par rapport à la longueur d'onde.

Ondes circulaires à 2 dimensions

On considère maintenant une onde propagative circulaire à 2 dimensions, excitée par une source ponctuelle située en (x_s, y_s). Il peut s'agir d'une onde électromagnétique émise par une antenne dans un plan, ou bien d'une vague à la surface de l'eau engendrée par la chute une goutte. En coordonnées polaires, l'onde est maintenant décrite par ($r \neq 0$) :

$$u(r,\theta,t) = \frac{A}{\sqrt{r}}\cos\left(\frac{2\pi}{\lambda}(r - ct)\right)$$

où r est la distance à la source, $r = \sqrt{(x - x_s)^2 + (y - y_s)^2}$. L'angle θ n'intervient pas car l'onde se propage de la même façon dans toutes les directions. Le facteur $1/\sqrt{r}$ est une conséquence de la conservation de l'énergie (en effet, le flux d'énergie sur un cercle de rayon r moyenné sur une période, qui est égal

à $2\pi r u^2(r,t)$, doit être indépendant de r). La fonction n'est pas définie en $r = 0$ (divergence de l'énergie). On ne considère plus dans cette partie le phénomène d'atténuation de l'onde (terme en e^{-x/x_0} du cas 1D).

1. On considère une région carrée du plan, de dimension $[0,1] \times [0,1]$, sur laquelle on va visualiser l'onde. On place une source ponctuelle au centre du domaine, en $(x_s, y_s) = (1/2, 1/2)$. On souhaite visualiser, pour l'instant $t = 0$ fixé, l'image d'une onde de longueur d'onde $\lambda = 0.1$. Pour cela, on discrétise le domaine de calcul à une certaine résolution, et l'on construit la matrice `U(i,j)` contenant les valeurs prises par u en chaque point (x_i, y_i) du domaine. On utilisera la fonction `imagesc` pour afficher la matrice sous forme d'une image. Pour le calcul de `U(i,j)` on pourra utiliser la fonction `meshgrid` (voir l'annexe).

2. Réaliser maintenant une animation de cette même onde. Faites varier la longueur d'onde, la célérité, et observer les résultats.

3. Réaliser maintenant une animation de l'interférence de deux ondes émises depuis deux sources ponctuelles, placées en (0.3, 0.5) et (0.7, 0.5). Observer les figures obtenues lorsque la longueur d'onde est petite, comparable ou grande par rapport à la distance entre les deux sources.

4. Pour simuler la diffraction d'une onde plane passant à travers un orifice, on peut simuler l'interférence d'un grand nombre de "sources secondaires" disposées le long de cet orifice. Observez la reconstruction de l'onde plane ainsi que la figure de diffraction sur le bord de l'orifice.

5. S'il reste du temps, essayez d'autres configurations : interférences de sources de célérité ou d'amplitude différentes, etc.

Annexe : quelques fonctions Matlab pour l'affichage d'images

Seule une description sommaire est donnée ici. Taper `help fonction` ou `doc fonction` pour plus d'information.

- `imagesc(C)` : Affiche la matrice `C` en ajustant la palette de couleur entre le plus petit et le plus grand élément de `C`. Attention : La convention d'ordonner les 2 dimensions d'une matrice en ligne-colonne implique que la dimension `I` soit représentée *verticalement* et que la dimension `J` soit représentée *horizontalement*. On pourra préférer représenter `C'` (transposée de `C`, inversion de l'ordre des 2 dimensions) pour une représentation selon la convention `X` horizontal - `Y` vertical.

- `imagesc(C,[min max])` : Idem, en imposant que la palette de couleur soit ajustée entre `min` et `max`.

- `imagesc(AXEI,AXEJ,C,[min max])` : Idem, en utilisant les vecteurs `AXEI` et `AXEJ` pour graduer les échelles verticales et horizontales. La matrice `C` doit être de taille `length(AXEI)` $\times$ `length(AXEJ)`.

- `axis ij` et `axis xy` : Bascule entre le mode d'affichage "matrice" (axe vertical descendant) et le mode "graphe" (axe vertical montant).

- `axis([xmin xmax ymin ymax])` : Fixe les échelles horizontales et verticales de la figure à `[xmin xmax]` et `[ymin ymax]`.

- `axis image` : Impose le même facteur d'échelle entre les abscisses et les ordonnées. Ainsi, une matrice de dimension `N` $\times$ `N` apparaitra carrée quelque soit la dimension de la fenêtre.

- `drawnow` : Met à jour les dernières modifications apportées à la figure courante. Par défaut, Matlab ne met pas à jour les figures au fur et à mesures que les modifications y sont apportées afin de gagner du temps. Il est donc important d'utiliser `drawnow` pour afficher des animations, afin d'obliger Matlab à rafraichir la figure en temps réel.

- `colormap` : Spécifie la palette de couleur à utiliser. Ex: `colormap gray` pour une palette en niveaux de gris.

- `colorbar` : Affiche la palette de couleur utilisée dans l'image.
- `meshgrid` : Cette fonction réalise un "maillage" du plan. Elle est utile pour évaluer une fonction `F(X,Y)` sur tout un domaine `X` $\times$ `Y`. Si l'on souhaite par exemple calculer la fonction $f(x,y) = x^2 + y^2$ pour tous $x \in [0,1,2]$, $y \in [0.1, 0.2, 0.3, 0.4]$, on écrira

```
X = [0:2];
Y = [1:4]/10;
[MX,MY] = meshgrid(X,Y);
F = MX.^2 + MY.^2;
```

Ici les 2 matrices `MX` et `MY` sont de taille 4×3, et sont données par :

$$\texttt{MX} = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 1 & 2 \end{pmatrix}, \qquad \texttt{MY} = \begin{pmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \\ 0.4 & 0.4 & 0.4 \end{pmatrix}$$

La matrice `F` est de taille 4×3 (on rappelle que les données pour x sont représentées en colonne, celles pour y en ligne). L'élément de matrice $\texttt{F}(3,1)$ correspond donc au point de coordonnées $x = \texttt{X}(1) = 0$, $y = \texttt{Y}(3) = 0.3$, et vaudra donc dans cet exemple $f(0, 0.3) = 0^2 + 0.3^2 = 0.09$.

Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 4 : La croissance dendritique

1 Introduction

L'objectif de ce TP est de réaliser une simulation numérique du phénomène de croissance dendritique, c'est-à-dire de la croissance d'un solide par agrégations successives d'ions à la surface d'une électrode, qui conduit spontanément à la formation de dendrites. La méthode utilisée est celle de la "diffusion limitée par agrégation" (DLA) : il s'agit de simuler des marcheurs aléatoires qui se déplacent un par un sur un réseau 2D jusqu'à ce qu'ils "touchent" l'agrégat auquel ils se collent.

On représente l'espace par une matrice $m(x, y)$, de dimension $nx \times ny$, dont les éléments valent 0 (false) si le site est vide et 1 (true) s'il est occupé. On considère le cas où l'agrégat initial (le "germe") est une ligne horizontale, et l'on ne considère que le demi-plan supérieur à cette ligne. Pour que les effets de bord ne soient pas trop prononcés, on propose d'imposer des conditions aux limites périodiques selon x, c'est-à-dire telles que `m(x,y) = m(x-nx,y)` si `x>nx`.

2 Algorithme

L'algorithme de DLA peut s'écrire sous la forme suivante (les tests divers, comme la sortie du domaine par le marcheur, ne sont pas indiqués) :

- Initialiser la matrice booléenne `m`, de dimension `nx` $\times$ `ny`, avec l'agrégat initial.
- Boucle : tant que l'agrégat tient dans le domaine `nx` $\times$ `ny`,
  - Choisir la position initiale du marcheur `n`.
  - Boucle : tant que le marcheur numero `n` ne touche pas l'agrégat
    - Attribue une nouvelle position au marcheur (un pas à gauche, à droite, en haut, ou en bas)
  - Ajouter le marcheur `n` à l'agrégat
  - Tracer le nouvel agrégat (cette opération peut être effectuée une fois tous les 10 marcheurs, pour économiser du temps de calcul).
- Fin.

Idéalement, la position initiale de chaque marcheur doit être très loin de l'agrégat. Cependant, pour économiser du temps de calcul, on pourra choisir de faire partir les marcheurs légèrement au-dessus du point le plus haut de l'agrégat.

3 Pour aller plus loin...

On pourra étudier la loi de croissance de l'agrégat, par exemple la hauteur moyenne ou la hauteur de la plus grande "excroissance" en fonction du temps.

On pourra aussi observer l'influence de "biais" dans la croissance de l'agrégat, par exemple via une direction privilégiée dans le mouvement aléatoire des particules.

Enfin, on pourra observer l'influence de la géométrie du germe initial : plan, disque, etc.

4 Annexe : Fonctions utiles

4.1 Fonctions booléennes

- `true` et `false` : synonymes de 1 et 0, mais stockés de façon "booléenne" (économise de la place mémoire).
- `A=(cond)` : renvoie `true` ou `false` selon que le test `cond` est vérifié ou non (avec `cond` de la forme `var1 OPERATION var2`).

  Exemple 1 : `A=(3>pi)` renvoie `false`. Cette syntaxe compacte est équivalent à : `if (3>pi), A=1; else A=0; end;`

  Exemple 2 : `A=(rand(1,20)>0.5)` renvoie un vecteur de la forme `[0 0 1 1 1 0 1 0 0 ...]`, où chaque 0 ou 1 est un nombre booléen présent avec une probabilité 1/2 dans le vecteur `A`.
- $\sim$ `X` : complémentaire de `X` ($\sim 0 = 1$, $\sim 1 = 0$).
- `logical(X)` : converti `X` en booléen (`false` si `X` est nul, `true` sinon). Par exemple, `logical([3 1 0 4 0 0 2]) = [1 1 0 1 0 0 1]`.
- `find(X)` : renvoie les indices de X comprenant des éléments non nuls (équivalent à `find(X ~= 0)`). Exemple : `find([3 1 0 4 0 0 2]) = [1 2 4 7]`.

4.2 Fonctions graphiques

- `image(m)` et `imagesc(m)` : représente la matrice `m` comme une "image" à l'écran, où chaque élément de matrice code pour la couleur du pixel correspondant. `imagesc` normalise (*scale*) la palette de couleur pour qu'elle soit adpatée aux valeurs minimales et maximales des éléments de matrice à représenter (ici noir et blanc pour 0 et 1).

  Note : Matlab utilise les indices `x,y` en représentation ligne-colonne, tandis que les graphes sont généralement représentés en colonne-ligne. Il est donc nécessaire de transposer la matrice à afficher, en utilisant `m'` au lieu de `m`.
- `axis image` : impose un rapport 1:1 à l'image, de sorte que les pixels soient bien carrés et non rectangulaires.
- `axis ij` ou `axis xy` : se met en mode "matrice" (axe vertical descendant) ou "graphe" (axe vertical montant).
- `colormap(gray)` : utilise une palette de couleur noir et blanc.
- `drawnow` : Demande à Matlab de retracer (mettre à jour) la figure.

Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 5 : Désintégration radioactive

5 La désintégration radioactive

Un atome radioactif peut se désintégrer naturellement au cours du temps. C'est le cas par exemple de l'isotope ^{14}C (carbone 14), qui se désintègre en ^{12}C (carbone 12, le carbone "classique"), sur une échelle de quelques milliers d'années. Pour un atome de ^{14}C isolé, il n'est pas possible de prévoir quand exactement cette désintégration aura lieu. En revanche, pour un ensemble d'atomes, on peut prédire statistiquement l'évolution du nombre d'atomes qui ne se sont pas encore désintégrés :

$$N(t) = N_0 e^{-\lambda t}, \tag{1}$$

avec N_0 le nombre initial d'atomes (à $t = 0$), et λ une constante qui dépend de l'atome considéré. On définit en général la demi-vie τ, telle que $N(\tau) = N_0/2$, ce qui donne

$$\tau = \frac{\ln 2}{\lambda}.$$

Dans le cas de la désintégration $^{14}\text{C} \rightarrow ^{12}\text{C}$, la demi-vie correspondante est de 5700 ans (soit $\lambda = 5.6\ 10^{-12}\ \text{s}^{-1}$), ce qui en fait un outil de datation pratique pour l'archéologie.

Dans ce TP, nous allons reproduire numériquement cette loi de décroissance radioactive par deux méthodes : Monte-Carlo et par résolution d'une équation différentielle.

6 Résolution par la méthode de Monte-Carlo

Une méthode "Monte-Carlo" consiste à résoudre un problème par tirages aléatoires. Dans cette approche, on considère N_0 atomes, chaque atome $a(i)$ étant repésenté par un nombre binaire, 1 (atome pas encore désintégré) ou 0 (atome désintégré). On discrétise le temps par intervalles dt. A chaque pas de temps, un atome donné i sera "désintégré" avec une probabilité λdt, ou bien laissé indemne avec une probabilité $1 - \lambda dt$.

Réaliser la simulation numérique décrivant l'évolution temporelle du nombre d'atomes $N(t)$ pas encore désintégrés, et observer l'évolution de $N(t)$ en fonction de t pour différentes valeurs du nombre d'atomes initial N_0 (de l'ordre de 10, 100, 1000...) Comparer avec la loi théorique (1).

Remarque : étant donnée la valeur de λ, il est pertinent de choisir l'année plutôt que la seconde comme unité de temps dans ce problème.

Facultatif : Lorsque N_0 n'est pas très élevé, on constate que la simulation peut s'écarter significativement de la théorie. En effectuant plusieurs realisations d'une même simulation, en déduire l'écart quadratique moyen $\Delta N(t)$ décrivant la dispersion entre les différentes réalisations.

7 Résolution directe de l'équation différentielle

L'origine de la loi exponentielle (1) est la suivante : Pour un nombre d'atomes $N(t)$ à un instant t donné, le nombre d'atomes qui va se désintégrer entre t et $t + dt$ est proportionnel à N et à dt ; ce nombre d'atomes est donné par $\lambda N dt$. Le nombre d'atomes "restant" $N(t)$ va donc diminuer d'une quantité

$$dN = -\lambda N dt.$$

Il s'agit d'une équation différentielle linéaire du premier ordre, dont la solution bien connue est (1).

7.1 Désintégration simple

Implémenter sous Matlab la résolution de cette équation différentielle, et vérifier que la solution numérique coïncide bien avec la solution analytique (on pourra calculer l'écart entre les 2 solutions pour caractériser l'erreur d'intégration).

Facultatif : Observer ce que devient le nombre d'atomes si l'on imagine que la demi-vie n'est plus constante, mais varie légèrement dans le temps. Par exemple, on pourra poser $\lambda(t) = \lambda_0(1 + \alpha t)$, avec $\alpha < \lambda$.

Annexe : Résolution d'une équation différentielle du 1er ordre sous Matlab

Pour intégrer une équation différentielle du type $y' = f(t, y)$, où $y' = dy/dt$, il faut

- Créer une fonction Matlab dans laquelle est codé $f(t, y)$, par exemple dans un fichier `f.m`. Attention : l'ordre des arguments d'entrée, (t, y), est important. Même si la fonction ne dépend pas de t (système dit "autonome"), la fonction doit quand même avoir ces 2 arguments d'entrée.
- Passer la fonction comme argument à un "solveur" d'équation différentielle, à l'aide de l'opérateur `@`. Le solveur usuel est `ode45` (Runge-Kutta explicite), qui permet d'intégrer la plupart des équations différentielles avec une bonne précision. La syntaxe est :

  ```
  [t, y] = ode45(@F, [t0 tf], y0)
  ```

  où `[t0 tf]` est l'intervalle de t sur lequel y doit être calculé, et `y0 = y(t0)` est la condition initiale. Le solveur renvoie un ensemble de points $y(t_i)$, en choisissant lui-même l'échantillonage (qui n'est pas nécessairement uniforme).
- Pour pouvoir accéder à une valeur $y(t1)$ quelconque, utiliser la syntaxe suivante

  ```
  sol = ode45(@F, [t0 tf], y0)
  y = deval(sol, t1)
  ```

7.2 Désintégration multiple

Dans certains cas, la désintégration d'un isotope peut donner lieu à un nouvel isotope lui-même radioactif, pouvant à son tour se désintégrer. Ainsi, si l'on a la séquence $A \rightarrow B \rightarrow C$, on peut généraliser l'équation différentielle précédente au système d'équations différentielles :

$$\begin{aligned} dN_A &= -\lambda_A N_A dt, \\ dN_B &= \lambda_A N_A dt - \lambda_B N_B dt. \end{aligned}$$

Dans ce cas, la condition initiale $t = 0$ est $N_A(0) = N_0$ et $N_B(0) = 0$. Afin de résoudre numériquement un tel système, on reprend la méthode précédente, mais la fonction scalaire $y(t)$ doit être représentée par une fonction vectorielle $Y(t)$, où Y est un vecteur à deux composantes, $Y = [N_A N_B]$.

Ecrire le système différentiel sous la forme

$$dY/dt = MY,$$

où M est une matrice 2×2 dont on identifiera les éléments. Intégrer ce système numériquement pour différentes valeurs de λ_A et λ_B, et observer l'évolution des 2 populations d'isotopes.

Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 6 : Trajectoires ballistiques avec frottement

8 Système sans frottement

On souhaite étudier la trajectoire d'un projectile dans le champ de gravité terrestre, avec ou sans force de frottement. On note m la masse du projectile, et $\vec{r}$ sa position. Le Principe Fondamental de la Dynamique s'écrit

$$m\frac{d^2\vec{r}}{dt^2} = m\vec{g} + \vec{F}_f, \tag{2}$$

où $\vec{F}_f$ est la force de frottement avec l'air. On se place dans le plan vertical, avec $(\vec{e}_x, \vec{e}_z)$ les vecteurs unitaires, et $\vec{g} = -g\vec{e}_z$. Le projectile est lancé du point $x = 0$, $z = h_0$, avec une vitesse initiale $\vec{v}_0 = v_0(\cos\alpha\vec{e}_x + \sin\alpha\vec{e}_z)$.

1. Les frottements sont négligés dans un premier temps. Rappeler dans ce cas l'expression analytique $(x(t), z(t))$ de la trajectoire du projectile.
2. En posant le vecteur $Y = (x, z, u, v)$, avec $u = dx/dt$ et $v = dz/dt$, ré-écrire l'équation différentielle du 2nd ordre (2) comme un système différentiel du 1er ordre de dimension 4,
$$\frac{dY}{dt} = \mathcal{F}_0(Y).$$
3. Coder sur Matlab, avec l'aide du solver `ode45`, la résolution de ce système. Représenter graphiquement la trajectoire, et comparer la solution analytique et la solution numérique.

9 Système avec frottement

On tient maintenant compte du frottement avec l'air. Si l'on considère l'écoulement de l'air autour du projectile comme étant turbulent, on peut modéliser la force de frottement par la loi suivante,

$$\vec{F}_f = -C_x S\frac{1}{2}\rho_a|\vec{v}|\vec{v},$$

où C_x est le coefficient de traînée, S la surface "frontale" (surface du projectile perpendiculaire au déplacement), et ρ_a la densité de l'air. On considère ici que le projectile est une sphère de rayon R et de densité ρ_p.

1. Montrer que l'équation (2) peut se mettre sous la forme
$$\frac{d^2\vec{r}}{dt^2} = \vec{g} - \beta|\vec{v}|\vec{v},$$
avec β (en m^{-1}) une constante que l'on exprimera en fonction de ρ_a/ρ_p, R et C_x.
2. Mettre cette nouvelle équation différentielle sous forme
$$\frac{dY}{dt} = \mathcal{F}(Y),$$
et la coder sous Matlab.

3. Dans la suite, on simulera ce problème dans le cas d'une balle de tennis, de rayon $R = 3.35$ cm et de masse $m = 58$ g. A suffisamment grande vitesse, le coefficient de trainée est égal à 0.4 pour une sphère. Visualiser les trajectoires obtenues pour des vitesses initiales de l'ordre de 50, 100, 150 km/h. Comparer les trajectoires obtenues avec et sans frottement.

4. Tracer sur un même graphique l'énergie cinétique, l'énergie potentielle de pesanteur, et l'énergie mécanique totale. Commenter ces courbes dans les cas avec et sans frottement. On pourra prendre $z = h_0$ comme origine de l'énergie potentielle, et normaliser les énergies par l'énergie cinétique initiale.

10 Le service d'Andy Roddick

On considère plus particulièrement la situation du service. Dans le cas d'Andy Roddick (record de vitesse de frappe au service), on a une hauteur initiale de frappe de $h_0 \simeq 2.60$ m et une vitesse maximale de 249 km/h.

1. Trouver l'angle α optimal pour que la balle soit "bonne" (passage au-dessus du filet et rebond dans la zone de service). On considère pour simplifier que la frappe est strictement perpendiculaire au filet. On cherchera sur internet les dimensions d'un terrain de tennis.

2. Facultatif : Tracer l'angle α optimal pour que le service vienne impacter juste sur la limite de zone de service en fonction de la vitesse initiale, pour une gamme de vitesse initiale comprise entre 50 et 250 km/h. Comparer la courbe avec et sans frottement.

Université Paris-Sud **Année 2010-11**

Licence 3 de Mécanique Physique

Expériences Numériques pour la Physique

TP 7 : Thermodynamique hors d'équilibre

On considère un gaz parfait bi-dimensionnel contenu dans une boîte de dimension $[0,1] \times [0,1]$. Cette boîte est séparée en son milieu (en $x = 1/2$) en deux compartiments par une paroi percée d'un trou centré, de taille D. On note (1) la partie gauche, et (2) la partie droite de la boîte.

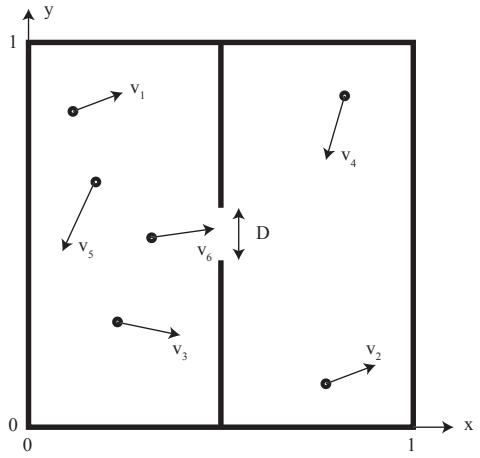

Figure 1: Gaz parfait dans une boîte compartimentée. Ici, la molécule numéro 6 va passer de gauche à droite, ce qui va uniformiser les densités dans chaque compartiment.

Partant d'une situation où le gaz est hors d'équilibre thermodynamique, on souhaite étudier numériquement la façon dont ce système atteint l'équilibre thermodynamique. On va considérer successivement 2 situations initiales hors d'équilibre :

A. La partie (1) contient initialement N molécules et la partie (2) est vide ;

B. Chaque partie contient $N/2$ molécules, mais les températures T_1 et T_2 sont différentes.

1. Trajectoire d'une molécule unique

On commence par considérer le mouvement d'une molécule unique. Cette molécule est caractérisée par 4 quantités : sa position (x, y) et sa vitesse (u, v). En l'absence de collisions, la molécule suit une trajectoire rectiligne à chaque pas de temps Δt,

$$x(t + \Delta t) = x(t) + u\,\Delta t, \qquad y(t + \Delta t) = y(t) + v\,\Delta t.$$

On suppose les collisions avec les parois parfaitement élastiques. Par exemple, lors d'une collision avec une paroi verticale, on a

$$u \to -u, \qquad v \to v.$$

Simuler la trajectoire de cette molécule unique, et vérifier visuellement que la propagation rectiligne et les collisions avec chacune des parois sont correctement reproduites.

2. Comportement collectif d'un gaz parfait

On va maintenant simuler les trajectoires d'un ensemble de N molécules, caractérisées par $4N$ quantités ("degrés de libertés"). On note $x_i(t), y_i(t), u_i(t)$ et $v_i(t)$ les composantes de la position et de la vitesse de chaque molécule $i = 1..N$. Le gaz est suffisamment dilué, de sorte que seules les collisions avec les parois sont considérées (les collisions inter-molécules sont négligées). A l'instant initial, on répartit les N molécules uniformément (en utilisant la fonction `rand`). On attribue à chaque molécule une vitesse aléatoire, de telle sorte que chaque composante u et v est distrubuée selon une loi Gaussienne : on a $\langle u \rangle = \langle v \rangle = 0$, et l'on note la température (normalisée) :

$$T = \frac{1}{2}(\langle u^2 \rangle + \langle v^2 \rangle).$$

On pourra utiliser la fonction `randn`, qui renvoie des nombres aléatoires de moyenne 0 et d'écart-type 1.

Simuler et visualiser les trajectoires des N molécules (on pourra prendre N de l'ordre de 10 à 10^4).

A. Mise à l'équilibre de densité

Afin de simuler la situation A, on part d'une situation où les N molécules sont initialement toutes à gauche. On note $N_1(t)$ et $N_2(t)$ le nombre de molécules dans chaque compartiment. Afin de caractériser la mise à l'équilibre du gaz ($N_1 = N_2 = N/2$), on introduit le coefficient de disymétrie,

$$K(t) = \frac{N_1 - N_2}{N_1 + N_2} \in [-1, 1]$$

de telle sorte que $K = 0$ à l'équilibre thermodynamique.

Tracez l'évolution temporelle de K pour différentes valeurs de N, et observez la vitesse de convergence vers l'équilibre. Que pensez-vous de l'influence de la taille du trou D ?

Facultatif : On pourra réaliser plusieurs expériences pour la même valeur de N et moyenner les résultats pour plus de précision.

B. Thermalisation

Dans la situation B, on part maintenant d'une situation où chaque compartiment contient initialement $N/2$ molécules, mais dont les vitesses sont maintenant telles que $T_1 > T_2$ (compartiment gauche chaud).

Etudier l'évolution de la température T de chaque compartiment. Montrer qu'avant d'atteindre l'équilibre thermodynamique ($T_1 = T_2$ et $N_1 = N_2$), le système passe par un transitoire tel que $N_1(t) > N_2(t)$. De quoi dépend la valeur maximale prise par le coefficient d'asymétrie K ?