

Systèmes dynamiques et contrôle

Rythme de renouvellement d'un parc automobile

Stéphane BINOIS

October 9, 2016

Contents

1	Division du parc en classes	2
2	Écriture du système dynamique	2
3	Implémentation du système dynamique	3
4	Initialisation des paramètres	4
4.1	Parc initial	4
4.2	Proportions des différentes classes technologiques	5
4.3	Immatriculations	6
4.4	Durée de simulation	7
4.5	Coefficients de survie	7
5	Evolution des différents sous-parcs	9
6	Evolution des parts des différentes classes technologiques	10
7	Visualisation des données 1962 - 1999	10
8	Visualisation du parc projeté	13
9	Visualisation “animée” de l'évolution du parc agrégé	14
10	Visualisation de l'évolution des parts des différentes classes	19

N.B. : Tous les graphiques présentant l'effectif d'un parc sont en milliers de véhicules. Par ailleurs, tous les graphiques montrant la répartition cumulée en types de véhicules présentent de bas en haut le nombre de diesel inférieurs, diesel supérieurs, essences inférieurs, essences moyens, essences supérieurs.

Introduction

L'objet de cette étude est de modéliser l'évolution du parc automobile à l'horizon 2020, d'une part dans son ensemble, mais surtout d'autre part dans sa répartition en classes d'âge et en classes technologiques. Une telle approche permet en effet de rendre compte du vieillissement du parc, ainsi que de l'évolution de la part de chaque classe technologique dans le parc automobile de 2000 à 2020. Toutes ces données peuvent permettre par exemple de modéliser la pollution future, domaine où connaître la répartition du parc selon le type de carburant et l'âge des véhicules est primordial.

Nous allons donc tout au long de ce document voir pas à pas comment réaliser une telle étude à l'aide du logiciel Scilab, tant pour le calcul des différentes composantes du parc année après année que pour la visualisation des divers résultats obtenus. Ce qui nous permettra donc d'aborder à la fois les problèmes de modélisation des systèmes dynamiques sous Scilab et les problèmes de représentation graphique (courbes 3D, superposition de courbes 2D, etc...).

Notons enfin que toutes les données numériques sont tirées des ouvrages cités en bibliographie. Nous ne nous étendrons pas sur les problèmes de recensement des véhicules, le lecteur curieux pouvant se reporter aux dits ouvrages. Quant aux prévisions d'immatriculation, elles correspondent toutes aux projections à l'horizon 2020 du parc automobile français (limité aux seuls véhicules particuliers).

Mise en équation du problème

Avant d'aborder l'utilisation de Scilab à proprement parler, détaillons la manière de modéliser le parc automobile français sous forme d'un système dynamique.

1 Division du parc en classes

Pour modéliser notre parc, nous allons répartir les véhicules en 28 classes d'âge. Chaque classe d'âge recouvre une année complète, et l'on suppose qu'aucune voiture ne survit après la 28ème classe¹.

Nous scindons par ailleurs le parc automobile en cinq classes technologiques :

- les voitures essence de cylindrée inférieure à 1,4 L
- les voitures essence de cylindrée comprise entre 1,4 L et 2 L
- les voitures essence de cylindrée supérieure à 2 L
- les voitures diesel de cylindrée inférieure à 2 L
- les voitures diesel de cylindrée supérieure à 2 L

Ces classes sont dénommées `einf`, `eint`, `esup`, `dinf` et `dsup` dans le programme écrit sous Scilab.

¹Ce qui ne gêne en rien les estimations au vu du nombre négligeable de véhicules particuliers ayant plus de 28 ans.

Comme nous l'avons signalé en introduction, nous scindons le parc en différentes classes technologiques car la finalité de notre étude est d'estimer les émissions polluantes dans le futur, ainsi que l'effet de nouvelles normes sur les dites émissions. Connaître la répartition en tranches d'âge mais aussi en type de motorisation et en cylindrée est donc important.

Chacune des classes technologiques que nous venons d'introduire est ensuite considérée comme un parc à part entière et simulée indépendamment des autres. Cela est possible puisqu'un véhicule ne passe jamais d'une classe technologique à une autre au cours de son existence.

Il ne reste ensuite qu'à sommer toutes les classes technologiques pour avoir l'évolution du parc automobile global. Cette approche peut paraître plus lourde en calculs qu'une approche globale, mais la légèreté des calculs nous permet d'en augmenter sans aucun problème la taille, et une telle approche permet d'avoir des informations sur l'âge des différents véhicules dans chaque classe technologique, d'où des résultats plus pertinents au final.

2 Écriture du système dynamique

Chaque classe d'âge j d'une catégorie technologique i va vieillir d'un an à un certain taux $q_{i,j}(t)$, appelé *taux de survie*, et mourir (donc disparaître du parc) à un taux $1 - q_{i,j}(t)$. Ce taux dépend du temps, même s'il sera pris constant dans le temps dans la majeure partie de cette étude². Par ailleurs, chaque année, on enregistre un nombre d'immatriculations neuves $f_i(t)$.

Cette modélisation n'est pas sans rappeler les modélisations d'évolutions des populations humaines, animales ou végétales, en remplaçant les immatriculations par les naissances. L'appellation *taux de survie* prend alors tout son sens.

En notant $N_i(t)$ le vecteur colonne de 28 lignes correspondant à la population de la classe technologique i au temps t , et en prenant pour t des valeurs discrètes (t correspond à une année, et ne prend donc que des valeurs entières), on peut écrire l'équation d'évolution suivante :

$$N_i(t+1) = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ q_{i,1}(t) & \ddots & \ddots & \ddots & 0 \\ 0 & q_{i,2}(t) & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{i,27}(t) & 0 \end{pmatrix} N_i(t) + \begin{pmatrix} f_i(t) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (1)$$

ce qui peut encore s'écrire, en nommant les deux matrices détaillées ci-dessus $M_i(t)$ et $F_i(t)$:

$$N_i(t+1) = M_i(t)N_i(t) + F_i(t). \quad (2)$$

²Mais la part de chaque technologique étant prise variable dans le temps (en fonction des projections sur le revenu des ménages et le taux de désérialisation), le taux de survie global du parc ne sera pas constant dans nos simulations.

La mise en équation étant effectuée, nous allons à présent pouvoir aborder l'écriture de ce système sous Scilab.

Implémentation et initialisation du parc automobile sous Scilab

3 Implémentation du système dynamique

L'écriture des systèmes dynamiques sous Scilab est assez simple. Pour implémenter notre système dynamique, nous allons tout d'abord définir une fonction récursive, dont la valeur à un temps donné dépend du temps t et de la valeur à $t - 1$. Par exemple, pour le système

$$N(t + 1) = M(t) * N(t) + f(t)$$

le code Scilab sera

```
deff(' [Ndot]=parc(t,N)', 'Ndot=M(t)*N+f(t)');
```

dans lequel la fonction `parc(t,N)` renvoie la valeur de $N(t + 1)$, la fonction `deff(...)` servant à définir les fonctions sous Scilab.

Cette ligne sera déclinée en 5 versions, correspondant aux 5 classes technologiques considérées. Les 5 “sous-parcs” seront donc définis par les lignes suivantes :

```
deff(' [Ndot]=parcdsup(t,N)', 'Ndot=Mdsup(t)*N+imdsup(:,t)');  
deff(' [Ndot]=parcdinf(t,N)', 'Ndot=Mdinf(t)*N+imdinf(:,t)');  
deff(' [Ndot]=parcesup(t,N)', 'Ndot=Mesup(t)*N+imesup(:,t)');  
deff(' [Ndot]=parceint(t,N)', 'Ndot=Meint(t)*N+imeint(:,t)');  
deff(' [Ndot]=parceinf(t,N)', 'Ndot=Meinf(t)*N+imeinf(:,t)');
```

4 Initialisation des paramètres

Les sous-parcs que nous venons de définir font appel à différents paramètres qu'il nous faut à présent initialiser, à savoir :

- les vecteurs initiaux $N_i(t_0)$;
- les parts des différentes classes technologiques année après année;
- les immatriculations;
- la durée de la simulation;
- les coefficients de survie $q_{i,j}$;

N.B.: Toutes les données numériques de parc sont en milliers de véhicules

4.1 Parc initial

Nous avons les données chiffrées du parc automobile français entre 1962 et 1999, réparti en 36 classes d'âge.

Le fichier `matA.dta` contient la composition du parc automobile français total (donc sans tenir compte d'une quelconque séparation en classes technologiques). Pour charger les données sous Scilab, taper la commande

```
load('matA.dta');
```

qui charge en mémoire une matrice A contenant les dites données. L'année 1999 nous servira de conditions initiales. Toutefois, la matrice A comporte en lignes les années de construction des véhicules en ordre croissant et en colonnes les années d'observation (de 1972 à 1999). L'année 1999 correspond donc à la dernière colonne, mais nous voulons les classes par âge croissant (et donc par année de construction décroissant), et seulement pour les 28 premières classes d'âge (contre 37 années de production dans la matrice A). D'où les commandes suivantes

```
//parc de depart
N0=zeros(28,1);           //on crée un vecteur vide pour 28 classes
for l=1:28,               //on ne considère que 28 années de construction
    N0(l)=A((37-l),26);   //on remplit N0
end
```

4.2 Proportions des différentes classes technologiques

Le vecteur N_0 comporte alors la répartition du parc global en 28 classes d'âge en 1999. Il nous faut alors répartir ce parc en différentes classes technologiques. Ce que nous allons faire en définissant d'une part la proportion de diesels dans le parc, et en considérant d'autre part les proportions des différentes classes de cylindrée au sein de chaque type de motorisation (les prévisions étant présentées sous cette forme). Nous créons ainsi les paramètres pour le temps $t = 0$, c'est-à-dire en 1999 :

```
//Initialisation des classes technologiques en 1999
pd0=0.4           //part du diesel en 1999
pe0=1-pd0        //et donc part de l'essence

pdinf0=0.402     //part des diesel de - de 2L en 1999
pdsup0=0.093     //diesel + de 2L
peinf0=0.31      //essence de - de 1.4L
peint0=0.174     //essence entre 1.4 et 2L
pesup0=0.021     //essence de + de 2L
```

Puis nous choisissons les valeurs de ces mêmes paramètres pour les véhicules neufs en 2020 ($t = 20$), avant de procéder à une interpolation linéaire pour déterminer les valeurs des paramètres entre 1999 et 2020. C'est ici que nous pourrions faire varier le paramètre `pd20` selon les prévisions de part du diesel³. Nous considérerons par contre dans tous les cas que les diesels supérieurs représenteront 95% des immatriculations de diesels en 2020, et que les essences supérieures représenteront 55% des essences, les intermédiaires regroupant quant à elles 40% des essences.

```
//Paramètres en 2020

pd20=.25                //part du diesel en 2020
pe20=1-pd20

//parts des cylindrees en 2020
pdinf20=pd20*0.95
pdsup20=pd20*0.05
peinf20=pe20*0.55
peint20=pe20*0.4
pesup20=pe20*0.05

//evolution des parts entre 2000 et 2020 par interpolation linéaire
pdinf=interpln([1 21;pdinf0 pdinf20],1:21);
pdsup=interpln([1 21;pdsup0 pdsup20],1:21);
peinf=interpln([1 21;peinf0 peinf20],1:21);
peint=interpln([1 21;peint0 peint20],1:21);
pesup=interpln([1 21;pesup0 pesup20],1:21);
```

4.3 Immatriculations

Il y avait 2,2 millions d'immatriculations en 2000, et les prévisions sont de 2,5 millions d'immatriculations pour 2020. Bien évidemment, comme pour tous les paramètres de cette étude, on peut donner différentes valeurs aux immatriculations en 2020 pour tester diverses hypothèses, mais nous nous tiendrons au chiffre de 2,5 millions de nouveaux véhicules en 2020 cette simulation. Là encore, le nombre d'immatriculations pour les années intermédiaires est

³Les projections concernant l'évolution de la part du diesel varie d'une étude à l'autre, en particulier parce que ces projections évoluent selon l'année : il y a sept ans, la progression du diesel était très forte (elle aurait atteint 100% des immatriculations neuves en 2003 !), puis il y a eu une amorce de baisse de cette part en raison des problèmes d'émissions polluantes entre autres. D'où des projections différentes entre ? et ?.

Il sera donc intéressant de procéder à deux simulations, selon la part du diesel parmi les immatriculations neuves en 2020 : 25% (scénario bas) ou 50% (scénario haut). Pour cet exposé, nous nous en tiendrons à l'hypothèse basse.

obtenu par interpolation linéaire entre 2000 et 2020. Et la ventilation des immatriculations dans les différentes classes technologiques se fait grâce aux coefficients annuels calculés au paragraphe précédent.

```
//immatriculations dans chaque classe technologique
immat=zeros(28,21);
immat(1,:)=interpLn([1 21;2200 2500],1:21);

imdsup=immat*diag(pdsup); //ventilation des immatriculations
imdinf=immat*diag(pdinf); //le ‘diag’ permet de multiplier les
imeinf=immat*diag(peinf); //coefficients de répartition en classes
imeint=immat*diag(peint); //technologiques d’une année avec les
imesup=immat*diag(pesup); // immatriculations de l’année correspondante
```

On peut se demander pourquoi mettre les immatriculations sous forme d’une matrice 28x21, alors que seule la première ligne est non nulle. Cela vient du fait qu’on ajoute les immatriculations au vecteur N à chaque pas de temps, donc les vecteurs doivent être de même taille, c’est-à-dire 28 lignes. Par ailleurs, chaque colonne représente un pas de temps, d’où les 21 colonnes.

4.4 Durée de simulation

La durée de simulation est implémentée par l’entremise d’un vecteur comportant toutes les valeurs prises par la variable de temps pendant la simulation. Ainsi, pour une simulation allant des années $k = 2000$ à $k = 2020$, on définit le vecteur temps t égal à 1 en 2000 :

```
//duree de la simulation
k=(2000:2020); //on crée un vecteur [2000 2001 2002 ... 2020]
k0=2000;
t=k-k0+1; //on a donc t=[1 2 3 .. 21]
```

4.5 Coefficients de survie

Les coefficients de survie sont très importants, puisque ce sont eux qui permettent d’évaluer le nombre de véhicules passant d’une année à l’autre. Afin d’avoir une simulation pertinente, nous utiliserons donc les taux de survie réels observés en 1999 pour chaque classe technologique. Comme nous l’avons déjà signalé, ces taux seront pris constants tout au long de l’étude, tout en gardant à l’esprit que les parts des différentes classes technologiques varieront, et donc que les taux de survie globaux ne seront pas constants...

Pour charger en mémoire la matrice de survie, télécharger le fichier survie.dta et inclure dans notre fichier Scilab la commande suivante :

```
load('survie.dta');
```

Cette matrice, dénommée `survie`, comporte les taux de survie de chaque tranche d'âge (les lignes de la matrice) dans chaque classe technologique (les colonnes). Nous allons donc extraire chaque colonne⁴, puis la multiplier par un vecteur ligne 1x21, afin de copier cette colonne pour chaque pas de temps. On pourra ainsi modifier les taux de survie (en supposant par exemple que les véhicules sont plus fiables, ou au contraire qu'ils seront plus facilement mis à la casse). Nous nous contenterons toutefois ici de cette matrice aux colonnes identiques. Les lignes de commande sont donc :

```
//Survies de chaque classe technologique
qeinf=survie(:,1)*ones(1,21);
qeint=survie(:,2)*ones(1,21);
quesup=survie(:,3)*ones(1,21);
qdinf=survie(:,5)*ones(1,21);
qdsup=survie(:,6)*ones(1,21);
```

Le lecteur désireux de visualiser ces courbes de survie pourra taper des commandes telles que :

```
plot2d(qdsup)
```

qui renvoie la courbe de la survie en fonction de l'âge du véhicule (en l'occurrence ici pour les diesels supérieurs).

Enfin, il nous faut construire la fonction qui à chaque pas de temps extrait les coefficients de survie des matrices que nous venons de définir, et crée la matrice que nous avons appelé M_i :

$$M_i(t) = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ q_{i,1}(t) & \ddots & \ddots & \ddots & 0 \\ 0 & q_{i,2}(t) & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{i,27}(t) & 0 \end{pmatrix}. \quad (3)$$

Cette disposition des coefficients est très facile à obtenir sous Scilab grâce à une macro qui prend comme argument un vecteur et crée une matrice nulle dont la diagonale est ledit vecteur, et décale cette diagonale si l'on ajoute un deuxième argument. D'où les lignes suivantes, que nous écrirons dans un fichier fonction.sci :

⁴Exceptée la quatrième colonne, qui correspond à une classe technologique non représentée dans cette étude


```

function [vec]=Mdsup(t)
vec=diag(qdsup(:,t),1)';
function [vec]=Mdinf(t)
vec=diag(qdinf(:,t),1)';
function [vec]=Meinf(t)
vec=diag(qeinf(:,t),1)';
function [vec]=Meint(t)
vec=diag(qeint(:,t),1)';
function [vec]=Mesup(t)
vec=diag(qesup(:,t),1)';

```

Il faut alors ajouter dans le fichier de commande une ligne pour charger ces fonctions :

```
getf fonction.sci
```

Simulation de l'évolution du parc automobile

5 Evolution des différents sous-parcs

Nous avons défini notre parc automobile comme un système dynamique en temps discret. Nous allons donc utiliser un solveur d'équations différentielles pour calculer l'évolution du système. Ce solveur a pour nom `ode` et prend comme paramètres, dans l'ordre:

- une chaîne de caractères précisant le caractère discret ou continu du système dynamique;
- l'état initial du système, à savoir donc ici un vecteur comportant la population de chaque classe d'âge;
- le temps initial;
- le vecteur temps contenant toutes les valeurs prises par la variable temps;
- les équations d'évolution du système dynamique, sous forme d'une fonction.

Nous allons utiliser `ode` pour chacune des classes technologiques, puis sommer les cinq classes pour obtenir le parc total. Ce qui donne au final les commandes suivantes:

```

//On simule l'evolution de chaque composante separement, puis on somme
//le tout pour avoir l'evolution globale duparc et donc des proportions
//de chaque classe technologique

```

```

Ndsup=ode('discrete',pdsup0*N0,1,t,parcdsup);
Ndinf=ode('discrete',pdinf0*N0,1,t,parcdinf);
Nesup=ode('discrete',pesup0*N0,1,t,parcesup);
Neint=ode('discrete',peint0*N0,1,t,parceint);
Neinf=ode('discrete',peinf0*N0,1,t,parceinf);

```

```

N=Ndsup+Ndinf+Neinf+Neint+Nesup;

```

6 Evolution des parts des différentes classes technologiques

Maintenant que nous avons calculé l'évolution de chaque classe technologique, nous pouvons nous intéresser à la part de chacune dans le parc agrégé.

Nous allons donc définir les sommes des lignes des matrices d'évolution, afin d'avoir le nombre total de véhicules d'une classe technologique à un instant donné, sans se préoccuper des classes d'âge. Les lignes de commande sont assez intuitives :

```

//Evolution des parts des differentes classes technologiques
sumN=sum(N,"row");
sumdsup=sum(Ndsup,"row");
sumdinf=sum(Ndinf,"row");
sumeinf=sum(Neinf,"row");
sumeint=sum(Neint,"row");
sumesup=sum(Nesup,"row");

```

On se reportera au chapitre suivant pour tout ce qui concerne la visualisation de ces résultats.

Visualisation des données

7 Visualisation des données 1962 - 1999

Nous avons au début de cette étude chargé en mémoire une matrice A (issue du fichier `matA.dta`) contenant la répartition en classes d'âge du parc global de 1962 à 1999.

Il suffit de taper la commande `A` pour que la matrice s'affiche à l'écran. Mais il est clairement plus agréable d'avoir une sortie graphique des données de cette matrice.

L'idée développée dans ce paragraphe va donc être de réaliser une représentation en trois dimensions, à savoir le nombre de véhicules en fonction d'une part de l'année d'observation, d'autre part de l'âge du véhicule. On pressent bien que la surface obtenue sera décroissante

avec l'âge (les véhicules disparaissant année après année du parc), mais croissante avec l'année d'observation (les immatriculations augmentent d'année en année).

Avant toute chose, il faut taper les lignes suivantes qui définissent une palette de couleurs, pour un meilleur rendu des couleurs de la surface :

```
//palette pour de jolis dessins 3d
m=228;
nn = fix(3/8*m);
r = [(1:nn)'/nn; ones(m-nn,1)];r(1)=1;
g = [zeros(nn,1); (1:nn)'/nn; ones(m-2*nn,1)];g(1)=1;
b = [zeros(2*nn,1); (1:m-2*nn)'/(m-2*nn)];b(1)=1;
h = [r g b];
xset('colormap',h);
```

Ceci étant fait, il nous faut traiter les données de la matrice A . En particulier, nous allons supprimer la première ligne de la matrice, car elle contient le total des voitures construites avant 1962, et regroupe donc plus de voitures que les autres lignes (qui ne concernent qu'une seule année de construction chacune). La matrice obtenue comporte alors 35 lignes. La deuxième modification provient du fait que la matrice A est ordonnée par année de construction, alors que notre graphique sera en fonction de l'âge des véhicules. Nous créons donc une nouvelle matrice, B , de 35 lignes et 26 colonnes que l'on remplit en partant de la dernière ligne de A ⁵. Les commandes à taper sont regroupées ci-dessous :

```
A2=A(2:36,:);
//on enleve la premiere ligne qui correspond au total des voitures
//construites avant 1962, et qui donc est plus grand que le total des
//voitures construites en 1962 (ce qui fait un pic en fin de courbe
//qui ne correspond à rien)

B=zeros(35,26);
for i=(1:26)
    j=35;
    while A2(j,i)==0,j=j-1; end
    for k0=(1:j)
        B(k0,i)=A2(j-k0+1,i);
    end
end
```

⁵Le script proposé contient une boucle while visant à sauter les coefficients nuls de A . Cette boucle permet de tenir compte du fait qu'une voiture ne peut pas avoir été construite une année postérieure à l'année d'observation, ce qui se traduit par une matrice A triangulaire. Rappelons que l'âge d'un véhicule est égal à la différence entre année d'observation et année de construction

Nous pouvons désormais taper la commande de représentation graphique en 3D. Il s'agit de la fonction `plot3d`. Pour simplifier, je vais donner la commande à taper, puis expliquer chacun des termes.

```
xbasc();
plot3d1(1:35,1974:1999,B,30,2.5,"age@annee@population");
halt();
```

`xbasc()` efface les éventuels graphiques dans la fenêtre graphique. `halt()` permet de faire une pause dans le programme, pause qui prend fin si l'on appuie sur la touche "entrée". Cette fonction évite ainsi que les différents graphiques s'enchaînent si l'on exécute le script⁶. Les différents arguments de `plot3d` sont ici :

- `1:35` et `1974:1999` qui sont les vecteurs correspondant aux axes x et y . Attention, la taille de ces deux vecteurs doit correspondre à la taille de la matrice à afficher, sans quoi un doux message d'erreur sera renvoyé ;
- `B` qui est tout simplement la matrice à visualiser ;
- `30` et `2.5` qui sont les angles de visualisation de la surface tracée (modifiez ces paramètres pour vous en rendre compte !). Notez toutefois que le bouton `3D Rot.` de la fenêtre des graphiques permet de faire varier ces angles une fois la surface tracée (appuyez sur le bouton, puis cliquez avec la souris sur la figure à tourner, déplacez la souris jusqu'à ce que les angles vous conviennent et appuyez alors sur le bouton gauche de la souris pour redessiner la surface);
- et enfin `"age@annee@population"` qui correspond aux légendes des différents axes séparées par des `@` et mises globalement entre guillemets.

Si cette représentation est esthétiquement intéressante, elle soulève un problème assez gênant : elle n'est pas toujours facile à lire, les différentes cotes des points de la surface n'étant pas aisées à déterminer. Cette représentation permet plus de voir les mouvements d'ensemble que les données pour une année donnée.

Pour pallier cette difficulté, nous allons utiliser parallèlement une autre représentation, se présentant sous forme d'*ailerons*. L'idée est ainsi de dessiner côte à côte les courbes en 2D du parc automobile pour des années données, par exemple tous les 5 ans. Pour ce faire, nous reprenons `B` et nous annulons quatre lignes sur cinq. Ce qui peut s'écrire :

⁶Pour exécuter le script d'un coup, il y a deux possibilités : soit l'on copie le script donné à la fin de ce document, qui reprend toutes les lignes expliquées pas à pas dans ce document, soit l'on enregistre le même script dans un fichier, par exemple `script.sce`, et l'on tape `exec script.sce` sous Scilab, ce qui exécutera le script contenu dans le fichier. La mise en place de poses est alors indispensable.

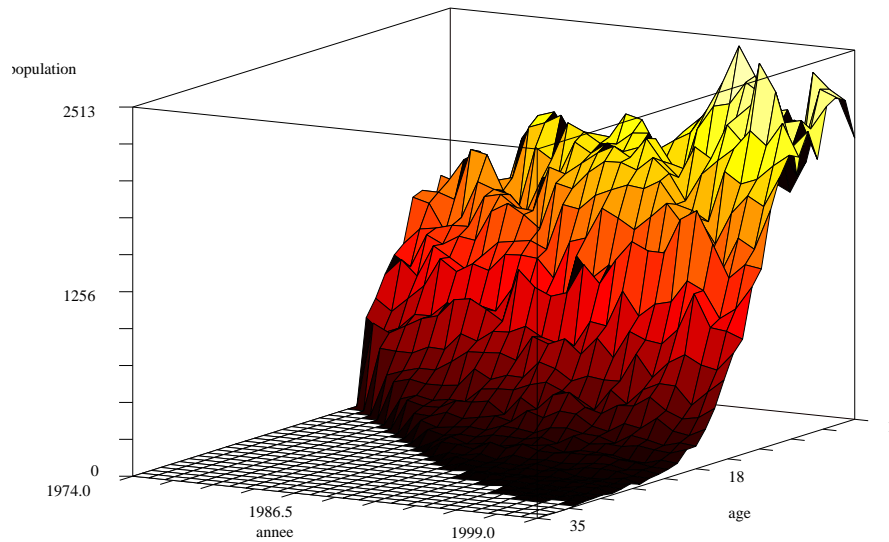


Figure 1: Évolution du parc des véhicules particuliers de 1974 à 1999

```
//graphiques à ailettes
```

```
B2=zeros(B);
for i=(1:5:26), B2(:,i)=B(:,i); end
B2(:,27)=0;
```

```
xbasc()
plot3d1(1:35,1974:2000,B2,30,2.5,"age@annee@population");
halt();
```

La boucle `for i=(1:5:26)` permet d'attribuer à `i` les valeurs de 1 à 26 par pas de 5 (c'est-à-dire 1, 6, 11, etc...).

8 Visualisation du parc projeté

Visualiser le parc projeté ne pose pas de problème particulier dès lors que l'on a réussi à visualiser le parc passé.

Ainsi, au lieu de visualiser `B`, on visualise directement `N`, qui est, rappelons-le, la matrice contenant les effectifs de chaque année (colonnes), pour chaque classe d'âge (lignes).

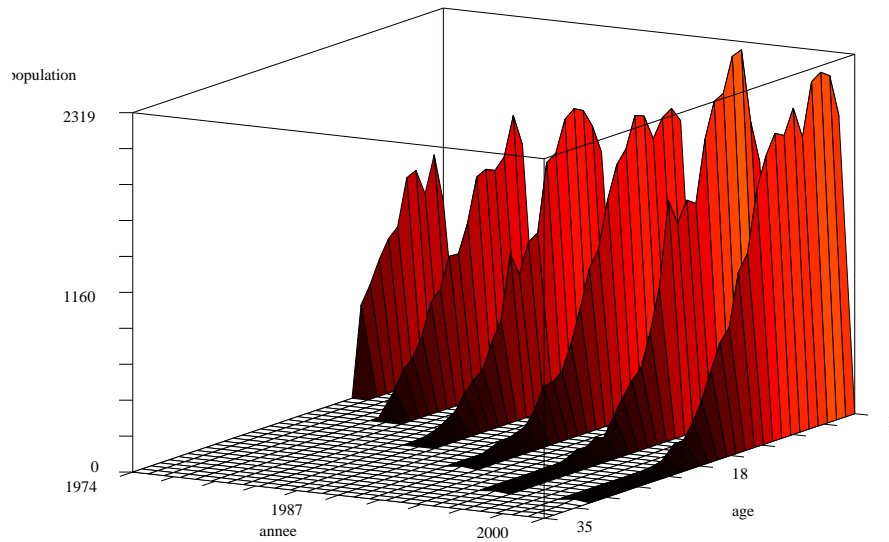


Figure 2: Évolution du parc des véhicules particuliers par pas de 5 ans de 1974 à 1999

Quant au graphique par ailettes, on procède de la même manière que précédemment, c'est-à-dire en recopiant qu'une année (donc colonne) sur 5 parmi celles de N , et en rajoutant une dernière colonne nulle, pour que la dernière ailette soit complète⁷.

Les lignes de commande sont finalement les suivantes :

```
//Visualisation du parc projeté

//lisse...
xbasc();
plot3d1(1:28,2000:2020,N,30,2.5,"age@annee@population");
halt();

//... ou ailerons
xbasc();
N2=zeros(N);
for i=(1:5:21), N2(:,i)=N(:,i); end
N2(:,22)=0;
```

⁷Si l'on ne rajoute pas cette colonne, la dernière ailette n'est pas fermée, puisqu'il n'y a pas d'autres données à tracer après cette ailette.

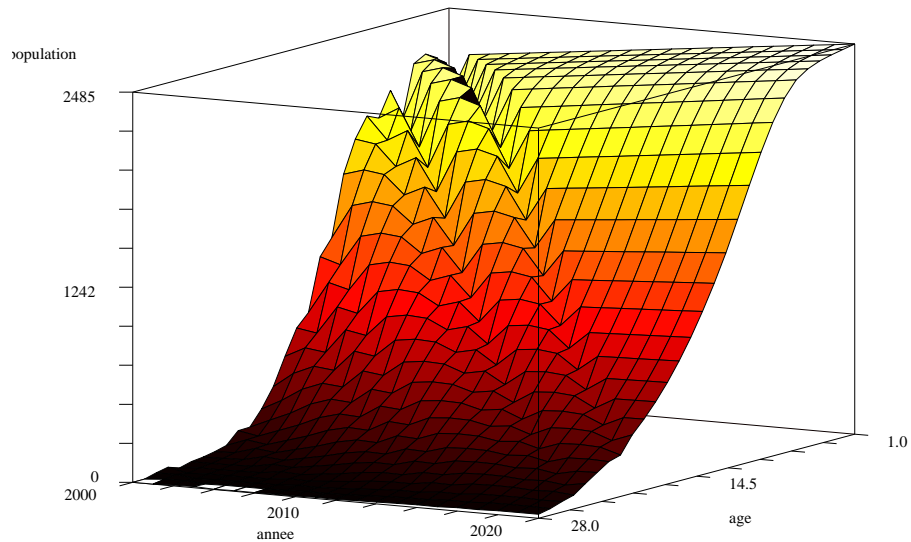


Figure 3: Évolution du parc de 1999 à 2020 - Scénario bas

```
plot3d1(1:28,1999:2020,N2,30,2.5,"age@annee@population");
halt();
```

9 Visualisation “animée” de l’évolution du parc agrégé

Les paragraphes précédents permettent certes de réaliser de jolis graphiques, mais il peut être préférable de recourir au “classique” graphique en deux dimensions, ne serait-ce que par souci de lisibilité, et pour tout simplement mettre l’accent sur un problème (et donc des courbes) que l’on juge important. Les paragraphes qui suivent ne traiteront donc plus que des graphiques en deux dimensions.

La première chose que nous allons donc faire est de lancer une nouvelle fenêtre graphique, ce qui permettra en outre de retrouver les couleurs d’origine dans cette fenêtre, alors que nous avons modifié la palette de la fenêtre pour les graphiques en 3D : en particulier, le tracé par défaut se fait en blanc sur fond blanc, ce qui est bien pour une surface de dégradés de couleurs (pour laquelle la lisibilité tient pour beaucoup aux dégradés dessinés) mais beaucoup moins pour une ligne monochrome, qui est alors confondu avec le fond... Bref, nous ouvrons une deuxième fenêtre, numérotée 1 (la première étant la 0) en tapant la commande :

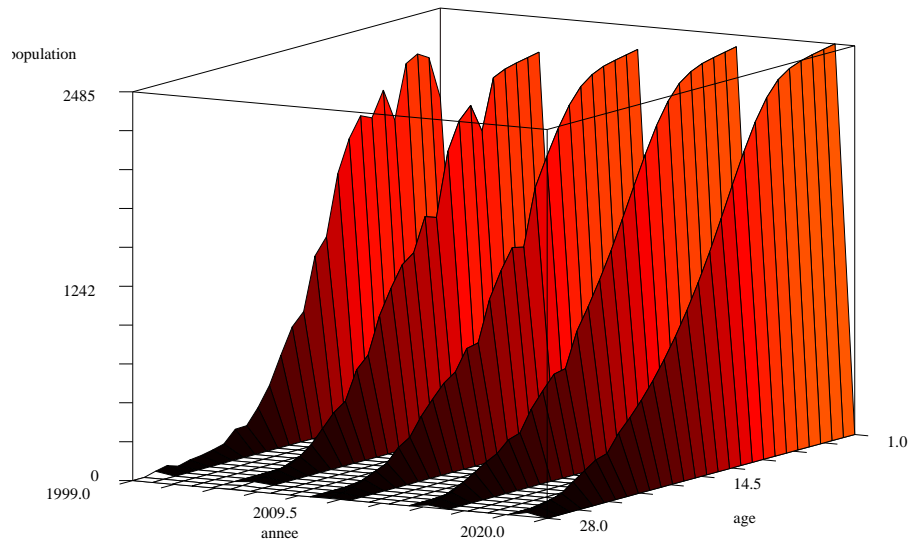


Figure 4: Évolution du parc de 1999 à 2020 - Scénario bas

```
xset('window',1);
```

Dans ce paragraphe, nous allons voir comment visualiser l'évolution du parc automobile autrement qu'avec des surfaces.

La simulation de l'évolution nous donne le parc pour chaque pas de temps, c'est-à-dire chaque année. Nous allons donc afficher l'une après l'autre chaque année : on visualisera ainsi la répartition du parc par tranches d'âge, d'abord pour la première année de simulation, puis pour la suivante, et ainsi de suite jusqu'à la dernière année :

```
xbasc();
for i=1:21,
    plot2d((1:28)',N(:,i),1,"011","a@b@c",[1,0,28,2500]),
    halt();
    xbasc();
end
```

Nous utilisons la commande `plot2d`, dont le rôle est de tracer des graphiques en 2D. Les différents paramètres sont *grosso modo* les mêmes que pour `plot3d`, l'argument `[1,0,28,2500]` que nous ajoutons ici étant en fait le domaine de visualisation, le rectangle dans lequel est

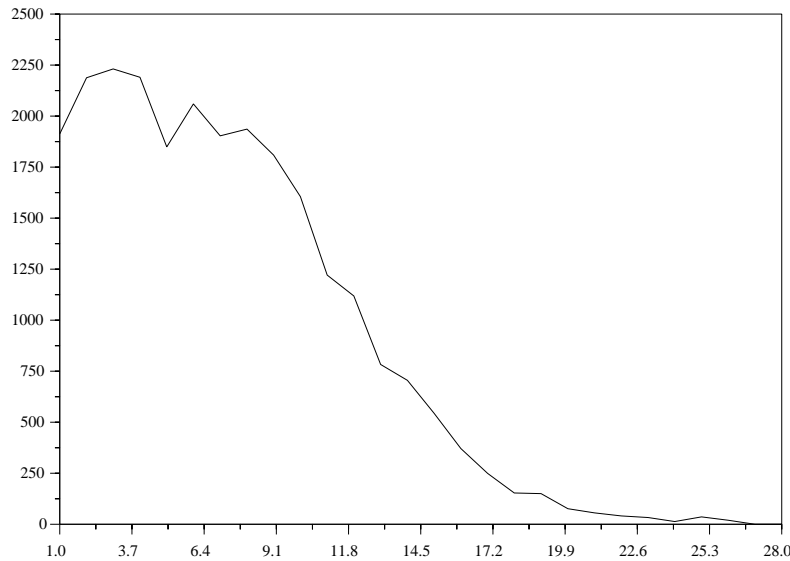


Figure 5: Répartition par classes d'âge du parc en 2000 - Scénario bas

tracée la courbe, ce qui correspond dans cet exemple à $x = 1$ à 28 et $y = 0$ à 2500. On pourra se reporter à l'aide en ligne pour plus de précisions.

Nous réalisons par ailleurs une boucle, qui parcourt les colonnes (donc les années) les unes après les autres. Cette boucle contient, en plus de la fonction de tracé, la fonction `halt()`, qui insère une pause (pour bien voir la courbe nouvellement tracée, sans quoi tout irait trop vite), ainsi que la fonction `xbasc()` qui efface la fenêtre graphique, les différentes figures étant superposées si l'on n'insère pas cette fonction dans la boucle.

10 Visualisation de l'évolution des parts des différentes classes

Nous avons, dans le chapitre précédent, calculé les effectifs globaux de chaque classe technologique pour chaque année de simulation, afin d'observer l'évolution des parts des différentes classes technologiques dans le temps.

Il nous faut à présent tracer ces données.

On pourrait tout d'abord les tracer les unes après les autres dans une même fenêtre, et en superposition. On taperait alors les commandes suivantes, où l'on note que le troisième paramètre de la fonction `plot2d` change à chaque ligne : il s'agit de la couleur du trait...

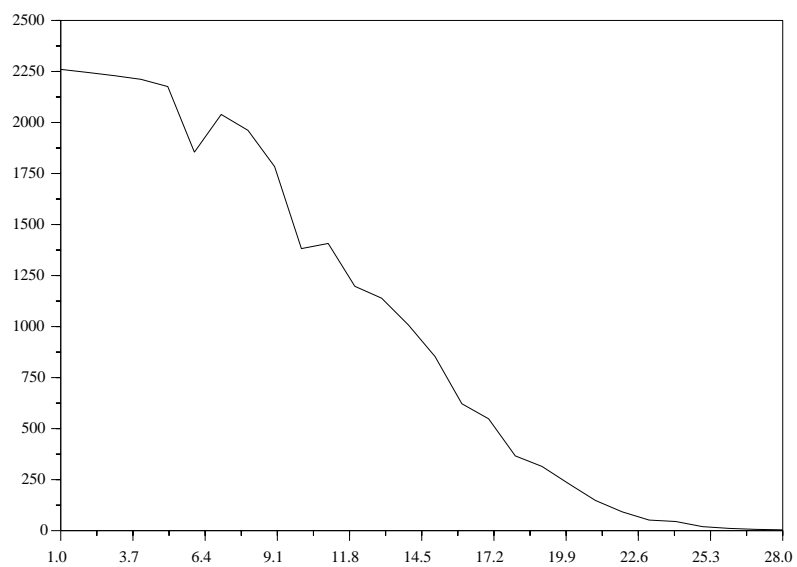


Figure 6: Répartition par classes d'âge du parc en 2005 - Scénario bas

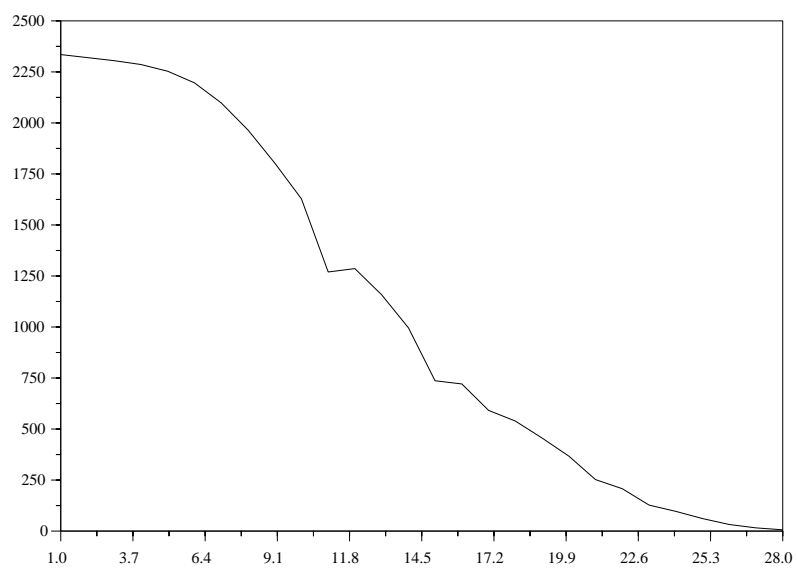


Figure 7: Répartition par classes d'âge du parc en 2010 - Scénario bas

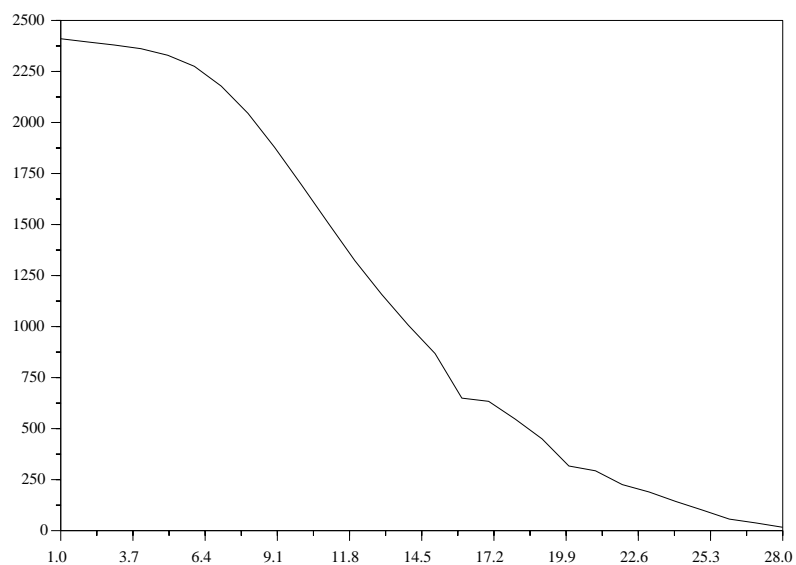


Figure 8: Répartition par classes d'âge du parc en 2015 - Scénario bas

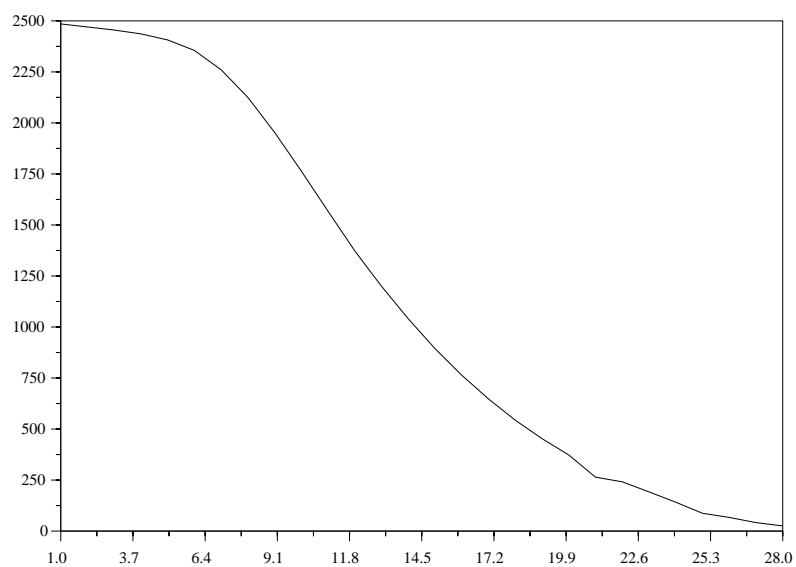


Figure 9: Répartition par classes d'âge du parc en 2020 - Scénario bas

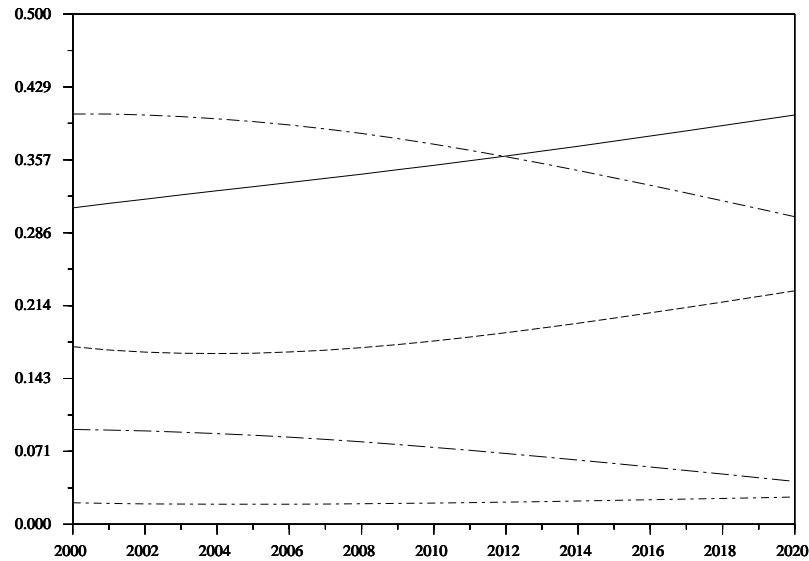


Figure 10: Évolution de la part de chaque classe technologique - Scénario bas

```

xbasc();
plot2d(k,sumeinf./sumN,1,"011","a@b@c",[2000,0,2020,0.5]);halt();
plot2d(k,sumeint./sumN,2,"011","a@b@c",[2000,0,2020,0.5]);halt();
plot2d(k,sumesup./sumN,3,"011","a@b@c",[2000,0,2020,0.5]);halt();
plot2d(k,sumdinf./sumN,4,"011","a@b@c",[2000,0,2020,0.5]);halt();
plot2d(k,sumdsup./sumN,5,"011","a@b@c",[2000,0,2020,0.5]);halt();

```

Mais on peut aussi visualiser l'évolution de la répartition en classes technologiques en réalisant un graphique des parts cumulées. On part ainsi du tracé de la part des essences inférieures, puis on superpose celui des essences intermédiaires auxquelles on a ajouté les essences inférieures, et ainsi de suite jusqu'au diesel supérieures. Une telle représentation rend un peu plus difficile l'appréciation de l'évolution des parts de chaque classe, mais a l'avantage de refléter l'évolution en valeur absolue.

```

halt();
plot2d([k' k' k' k' k'],[sumeinf' sumeinf'+sumeint' sumeinf'+sumeint'+sumesup'...
sumeinf'+sumeint'+sumesup'+sumdinf' sumeinf'+sumeint'+sumesup'+sumdinf'+sumdsup']);
halt();xbasc()

```

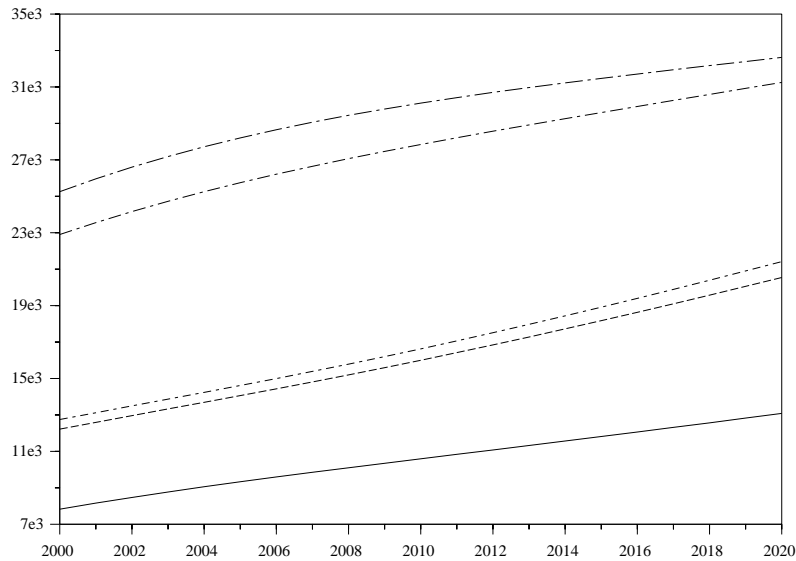


Figure 11: Évolution du nombre total de véhicules dans chaque classe technologique - Scénario bas

Pénétration des nouvelles normes sur les émissions Estimer la pénétration d’une nouvelle norme sur les émissions polluantes, ou tout simplement d’un nouveau matériel (pot catalytique, filtre à particules), c’est estimer l’évolution de la proportion de véhicules répondant à la norme, et ainsi estimer les effets de l’introduction d’une norme sur les émissions du parc automobile.

Pour estimer la pénétration d’une nouvelle norme ou d’un nouveau matériel dans le parc automobile, il suffit de simuler le parc constitué par les véhicules répondant à cette norme ou par les véhicules de ce nouveau type, et de procéder ensuite comme il a été fait pour étudier l’évolution de la part de chaque classe technologique. Ainsi, on recense chaque année le nombre total de voitures dans le parc “nouvelle norme” et on le compare au nombre total de voitures dans le parc global.

On peut ainsi étudier la pénétration d’une nouvelle norme selon le nombre d’immatriculations neuves projetées pour cette norme.

Les normes sur les émissions sont supposées être présentes sur tous les véhicules neufs l’année de la rentrée en vigueur du texte. On suppose par ailleurs qu’aucune voiture ne répond à la nouvelle norme au départ. Cette hypothèse nous fait donc partir d’un état initial nul. Ce n’est pas tant pour simplifier la problème que pour ne pas avoir à inventer des données initiales qui ne correspondent à rien. Le lecteur pourra toujours se reporter à la simulation des classes technologiques s’il désire compliquer la donne, en introduisant par exemple un taux d’équipement des véhicules neufs variable avec le temps.

Nous allons en ce qui nous concerne tester les normes Proposal 2000 et Proposal 2010, chacune contenant dans sa dénomination son année d'entrée en vigueur, et la deuxième remplaçant la première (qui s'arrête donc en 2010). On peut ainsi voir le taux de pénétration selon l'évolution du nombre d'immatriculations, taux qui est bien évidemment d'autant plus élevé qu'il y a plus d'immatriculations neuves.

Nous introduisons donc de nouveaux sous-parcs, à savoir un par norme et par classe technologique. L'implémentation est ici aisée puisque la simplicité des conditions nous ramène à un simple décalage du temps initial à l'année de rentrée en vigueur de la nouvelle norme.

D'où les lignes suivantes, où l'on introduit au passage les parts des différentes normes dans le parc :

```
//Reglementation Proposal2010
```

```
Nproeinf=ode('discrete',zeros(28,1),7,(7:21),parceinf);
Nproeint=ode('discrete',zeros(28,1),7,(7:21),parceint);
Nproesup=ode('discrete',zeros(28,1),7,(7:21),parcesup);
Nprodinf=ode('discrete',zeros(28,1),7,(7:21),parcdinf);
Nprodsup=ode('discrete',zeros(28,1),7,(7:21),parcdsup);
```

```
sproeinf=sum(Nproeinf,'row');
sproeint=sum(Nproeint,'row');
sproesup=sum(Nproesup,'row');
sprodinf=sum(Nprodinf,'row');
sprodsup=sum(Nprodsup,'row');
```

```
proeinf=(sproeinf./sumN(7:21))'
proeint=(sproeint./sumN(7:21))'
proesup=(sproesup./sumN(7:21))'
prodinf=(sprodinf./sumN(7:21))'
prodsup=(sprodsup./sumN(7:21))'
```

```
//Reglementation Proposal2000
```

```
Nproeinf2000=ode('discrete',zeros(28,1),1,1:21,parceinf)-[zeros(28,6) Nproeinf];
Nproeint2000=ode('discrete',zeros(28,1),1,1:21,parceint)-[zeros(28,6) Nproeint];
Nproesup2000=ode('discrete',zeros(28,1),1,1:21,parcesup)-[zeros(28,6) Nproesup];
Nprodinf2000=ode('discrete',zeros(28,1),1,1:21,parcdinf)-[zeros(28,6) Nprodinf];
Nprodsup2000=ode('discrete',zeros(28,1),1,1:21,parcdsup)-[zeros(28,6) Nprodsup];
```

```
proeinf2000=sum(Nproeinf2000,'row')./sumN;
proeint2000=sum(Nproeint2000,'row')./sumN;
proesup2000=sum(Nproesup2000,'row')./sumN;
```

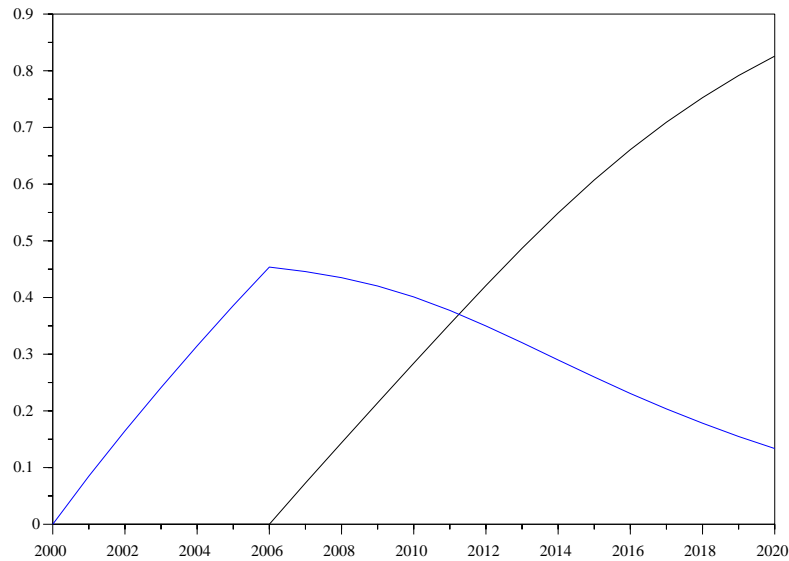


Figure 12: Pénétration des normes Proposal - 1,7 million d'immatriculations neuves en 2020

```
prodinf2000=sum(Nprodinf2000,'row')./sumN;
prodsup2000=sum(Nprodsup2000,'row')./sumN;
```

La visualisation est alors identique à celles déjà abordées :

```
xbasc();
plot(2000:2020',proeinf2000+proeint2000+proesup2000+prodinf2000+prodsup2000)
halt();
```

Les graphiques suivants montrent la vitesse de pénétration de ces deux normes selon le nombre d'immatriculations projeté en 2020. On notera tout de même que les courbes restent très proches l'une de l'autre⁸.

On peut également simuler la pénétration d'un équipement, comme par exemple le pot catalytique essence nouvelle génération, par rapport au nombre total de voitures en circulation et par rapport aux nombres de voitures essence en circulation, en supposant que toutes les nouvelles voitures essence le possèdent à partir de 2000. On obtient ainsi le graphique 16, correspondant aux scénarios bas et haut. Enfin, le dernier graphique [Fig 17] nous montre

⁸Il faut savoir que plus de 8% du parc est renouvelé chaque année

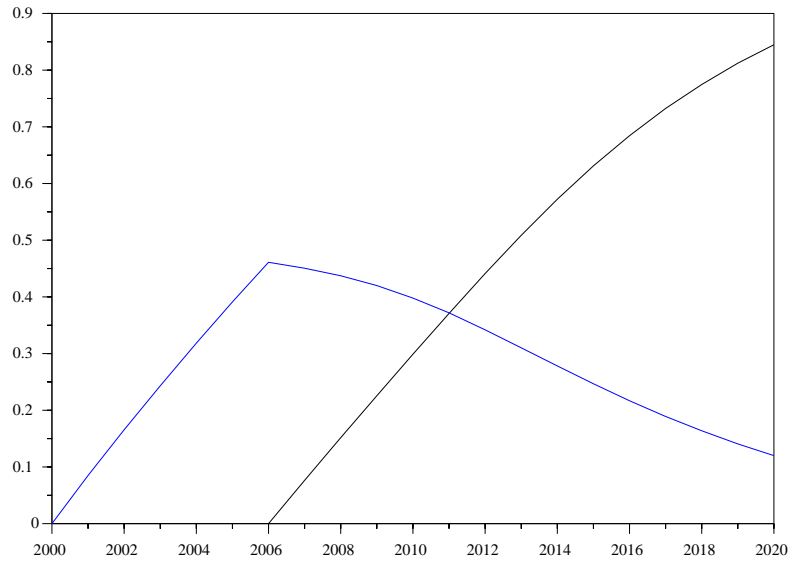


Figure 13: Pénétration des normes Proposal - 2,2 millions d'immatriculations neuves en 2020

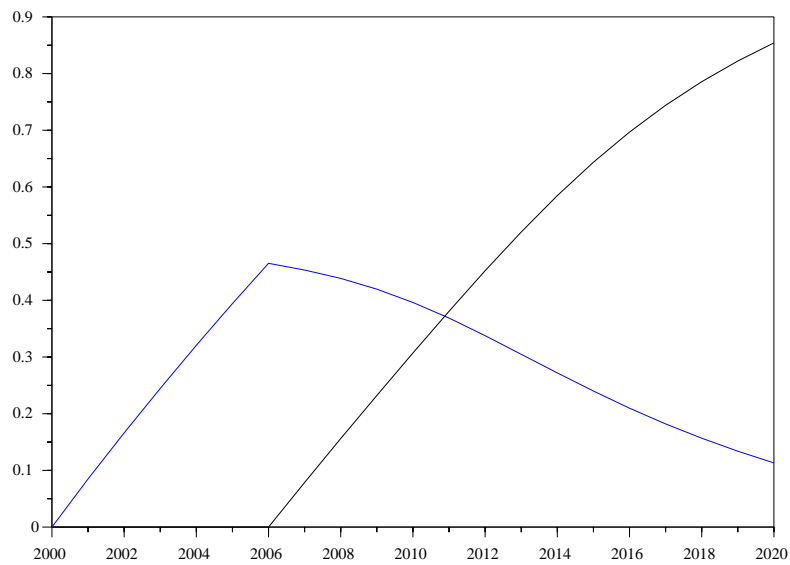


Figure 14: Pénétration des normes Proposal - 2,5 millions d'immatriculations neuves en 2020

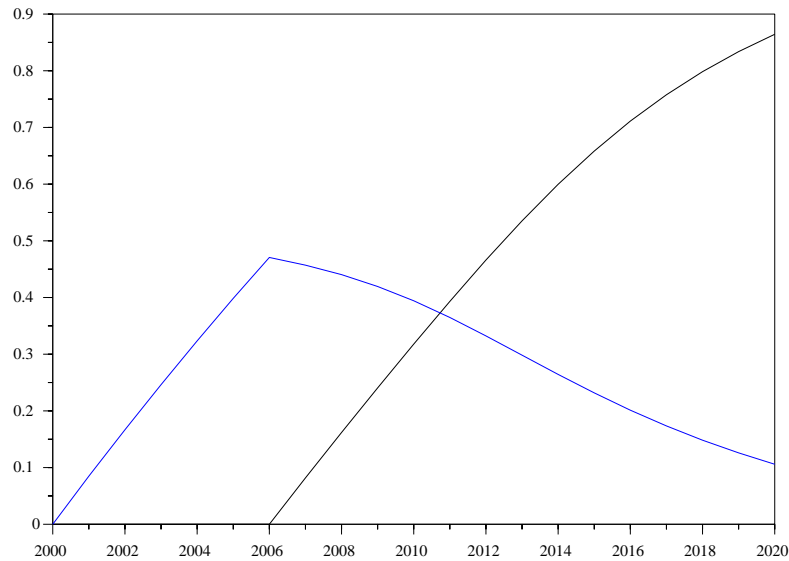


Figure 15: Pénétration des normes Proposal - 2,9 millions d'immatriculations neuves en 2020

la pénétration d'une norme généralisée à tous les nouveaux véhicules (comme par exemple un pot catalytique essence ou diesel). Cela correspond donc à la pénétration des nouvelles générations par rapport au total du parc. On observe ainsi qu'au bout d'un an, plus de 8% du parc répond à la norme, mais que ce n'est qu'au bout de 16 ans que plus de 90% du parc est aux normes.

Le code source final Le code présenté dans cette partie reprend tous les morceaux disséminés au fil des explications dans les chapitres précédents. Il n'est donc pas "complet" en ceci qu'il faudrait le relancer plusieurs fois de suite en changeant les paramètres (part future des diesels, etc.) pour pouvoir faire des comparaisons intéressantes. L'objet de cet exposé étant l'utilisation de Scilab pour simuler l'évolution d'une population (ici, des automobiles), nous laisserons le soin au lecteur de poursuivre les essais de différents paramètres pour approfondir tout ce qui n'a pas trait directement à Scilab.

```
load('matA.dta');

//parc de depart
N0=zeros(28,1);           //on crée un vecteur vide
for l=1:26,                //on ne considère que 26 classes d'âge
```

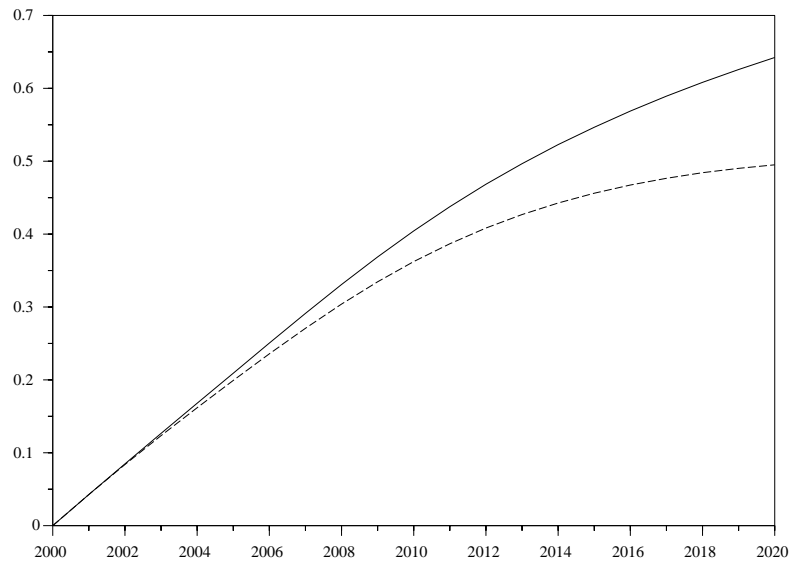


Figure 16: Pénétration des pots catalytiques essence de 2000 à 2020

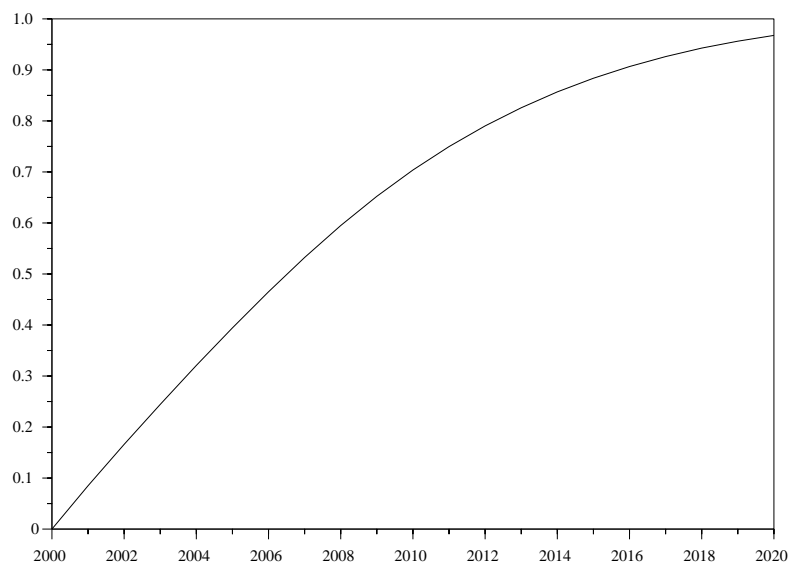


Figure 17: Pénétration des nouvelles générations dans le parc

```

    N0(1)=A((37-1),26); //on remplit N0
end

//Initialisation des classes technologiques en 1999
pd0=0.4 //part du diesel en 1999
pe0=1-pd0 //et donc part de l'essence

pdinf0=0.402 //part des diesel de - de 2L en 1999
pdsup0=0.093 //diesel + de 2L
peinf0=0.31 //essence de - de 1.4L
peint0=0.174 //essence entre 1.4 et 2L
pesup0=0.021 //essence de + de 2L

//Paramètres en 2020

pd20=.25 //part du diesel en 2020
pe20=1-pd20

//parts des cylindrees en 2020
pdinf20=pd20*0.95
pdsup20=pd20*0.05
peinf20=pe20*0.55
peint20=pe20*0.4
pesup20=pe20*0.05

//evolution des parts entre 2000 et 2020 par interpolation linéaire
pdinf=interpLn([1 21;pdinf0 pdinf20],1:21);
pdsup=interpLn([1 21;pdsup0 pdsup20],1:21);
peinf=interpLn([1 21;peinf0 peinf20],1:21);
peint=interpLn([1 21;peint0 peint20],1:21);
pesup=interpLn([1 21;pesup0 pesup20],1:21);

//immatriculations dans chaque classe technologique
immat=zeros(28,21);
immat(1,:)=interpLn([1 21;2200 2500],1:21);

imdsup=immat*diag(pdsup); //ventilation des immatriculations
imdinf=immat*diag(pdinf);
imeinf=immat*diag(peinf);
imeint=immat*diag(peint);

```

```

imesup=immat*diag(pesup);

//duree de la simulation
k=(2000:2020); //on crée un vecteur [2000 2001 2002 ... 2020]
k0=2000;
t=k-k0+1;      //on a donc t=[1 2 3 .. 21]

//Matrice de survie
load('survie.dta');

//Survies de chaque classe technologique
qeinf=survie(:,1)*ones(1,21);
qeint=survie(:,2)*ones(1,21);
qesup=survie(:,3)*ones(1,21);
qdinf=survie(:,5)*ones(1,21);
qdsup=survie(:,6)*ones(1,21);

getf new.sci

//Définition des parcs technologiques
deff(' [Ndot]=parcdsup(t,N)', 'Ndot=Mdsup(t)*N+imdsup(:,t)');
deff(' [Ndot]=parcdinf(t,N)', 'Ndot=Mdinf(t)*N+imdinf(:,t)');
deff(' [Ndot]=parcesup(t,N)', 'Ndot=Mesup(t)*N+imesup(:,t)');
deff(' [Ndot]=parceint(t,N)', 'Ndot=Meint(t)*N+imeint(:,t)');
deff(' [Ndot]=parceinf(t,N)', 'Ndot=Meinf(t)*N+imeinf(:,t)');

//On simule l'évolution de chaque composante separement,
//puis on somme le tout pour avoir l'évolution globale du parc
//et donc des proportions de chaque classe technologique

Ndsup=ode('discrete',pdsup0*N0,1,t,parcdsup);
Ndinf=ode('discrete',pdinf0*N0,1,t,parcdinf);
Nesup=ode('discrete',pesup0*N0,1,t,parcesup);
Neint=ode('discrete',peint0*N0,1,t,parceint);
Neinf=ode('discrete',peinf0*N0,1,t,parceinf);

N=Ndsup+Ndinf+Neinf+Neint+Nesup;

```

```

//Evolution des parts des differentes classes technologiques
sumN=sum(N,"row");
sumdsup=sum(Ndsup,"row");
sumdinf=sum(Ndinf,"row");
sumeinf=sum(Neinf,"row");
sumeint=sum(Neint,"row");
sumesup=sum(Nesup,"row");

//visualisation des données 1962-1999

//palette pour de jolis dessins 3d
m=228;
nn = fix(3/8*m);
r = [(1:nn)'/nn; ones(m-nn,1)];r(1)=1;
g = [zeros(nn,1); (1:nn)'/nn; ones(m-2*nn,1)];g(1)=1;
b = [zeros(2*nn,1); (1:m-2*nn)'/(m-2*nn)];b(1)=1;
h = [r g b];
xset('colormap',h);

A2=A(2:36,:);
//on enleve la premiere ligne qui correspond au total des voitures
//construites avant 1962, et qui donc est plus grand que le total des
//voitures construites en 1962 (ce qui fait un pic en fin de courbe
//qui ne correspond à rien)

B=zeros(35,26);
for i=(1:26)
    j=35;
    while A2(j,i)==0,j=j-1; end
    for k0=(1:j)
        B(k0,i)=A2(j-k0+1,i);
    end
end

xbasc()
plot3d1(1:35,1974:1999,B,30,2.5,"age@annee@population");

```

```

halt()

//graphiques à ailettes

B2=zeros(B);
for i=(1:5:26), B2(:,i)=B(:,i); end
B2(:,27)=0;

xbasc()
plot3d1(1:35,1974:2000,B2,30,2.5,"age@annee@population");
halt();

//Visualisation du parc projeté

//lisse...
xbasc();
plot3d1(1:28,2000:2020,N,30,2.5,"age@annee@population");
halt();

//... ou ailerons
xbasc();
N2=zeros(N);
for i=(1:5:21), N2(:,i)=N(:,i); end
N2(:,22)=0;

plot3d1(1:28,1999:2020,N2,30,2.5,"age@annee@population");
halt();

//on lance une nouvelle fenetre pour les traces 2D

xset('window',1);

// visualisation animée de l'évolution du parc
xbasc();
for i=1:21,
    plot2d((1:28)',N(:,i),1,"011","a@b@c",[1,0,28,2500]),
    halt();
    xbasc();
end

```

```
end
```

```
//visualisation de l'évolution de la répartition du parc  
// selon les classes technologiques
```

```
xbasc();  
plot2d(k,sumeinf./sumN,1,"011","a@b@c",[2000,0,2020,0.5]);halt();  
plot2d(k,sumeint./sumN,2,"011","a@b@c",[2000,0,2020,0.5]);halt();  
plot2d(k,sumesup./sumN,3,"011","a@b@c",[2000,0,2020,0.5]);halt();  
plot2d(k,sumdinf./sumN,4,"011","a@b@c",[2000,0,2020,0.5]);halt();  
plot2d(k,sumdsup./sumN,5,"011","a@b@c",[2000,0,2020,0.5]);halt();
```

```
halt();  
plot2d(['k' 'k' 'k' 'k' 'k'],[sumeinf' sumeinf'+sumeint' sumeinf'+sumeint'+sumesup'...  
sumeinf'+sumeint'+sumesup'+sumdinf' sumeinf'+sumeint'+sumesup'+sumdinf'+sumdsup']);  
halt();xbasc()
```

```
//Pénétration de nouvelles normes
```

```
//Reglementation Proposal2010
```

```
Nproeinf=ode('discrete',zeros(28,1),7,(7:21),parceinf);  
Nproeint=ode('discrete',zeros(28,1),7,(7:21),parceint);  
Nproesup=ode('discrete',zeros(28,1),7,(7:21),parcesup);  
Nprodinf=ode('discrete',zeros(28,1),7,(7:21),parcdinf);  
Nprodsup=ode('discrete',zeros(28,1),7,(7:21),parcdsup);
```

```
sproeinf=sum(Nproeinf,'row');  
sproeint=sum(Nproeint,'row');  
sproesup=sum(Nproesup,'row');  
sprodinf=sum(Nprodinf,'row');  
sprodsup=sum(Nprodsup,'row');
```

```
proeinf=(sproeinf./sumN(7:21))'  
proeint=(sproeint./sumN(7:21))'  
proesup=(sproesup./sumN(7:21))'  
prodinf=(sprodinf./sumN(7:21))'
```

```

prodsup=(sprodsup./sumN(7:21))'

//Reglementation Proposal2000
Nproeinf2000=ode('discrete',zeros(28,1),1,1:21,parceinf)-[zeros(28,6) Nproeinf];
Nproeint2000=ode('discrete',zeros(28,1),1,1:21,parceint)-[zeros(28,6) Nproeint];
Nproesup2000=ode('discrete',zeros(28,1),1,1:21,parcesup)-[zeros(28,6) Nproesup];
Nprodinf2000=ode('discrete',zeros(28,1),1,1:21,parcdinf)-[zeros(28,6) Nprodinf];
Nprodsup2000=ode('discrete',zeros(28,1),1,1:21,parcdsup)-[zeros(28,6) Nprodsup];

proeinf2000=sum(Nproeinf2000,'row')./sumN;
proeint2000=sum(Nproeint2000,'row')./sumN;
proesup2000=sum(Nproesup2000,'row')./sumN;
prodinf2000=sum(Nprodinf2000,'row')./sumN;
prodsup2000=sum(Nprodsup2000,'row')./sumN;

xbasc();
plot(2000:2020',proeinf2000+proeint2000+proesup2000+prodinf2000+prodsup2000)
halt();

//FIN

```