ministère
de l'Équipement
des Transports
de l'Aménagement
du territoire
du Tourisme
et de la Mer

ENTPE
École nationale
des travaux publics
de l'État

Modélisation

PROJET

Activités d'une station service

Auteur : de Sigaldi / Cardon / Telpic

Promotion : 53

Année : 2006

rue Maurice Audin
69518 Vaulx-en-Velin Cedex
téléphone :
+33 (0)4 72 04 70 70
télécopie :
+33 (0)4 72 04 62 54

http://www.entpe.fr

Sommaire

0 - Généralités sur les chaînes de Markov

0.1 - Processus de naissance et de mort

On considère une file d'attente comportant entre 0 et N objets. Le nombre d'objets présents dans la file à l'instant t représente l'état du processus. On se donne des nombres positifs quelconques $(\lambda_i)_{0\le i\le N-1}$ et $(\mu_i)_{1\le i\le N}$.

On suppose que pendant un intervalle de temps infiniment petit Δt, sachant qu'il y a k objets présent dans la file :

- la probabilité d'apparition d'un objet dans la file pendant Δt est égale à $\lambda_k \Delta t + o(\Delta t)$;
- en négligeant l'apparition de deux objets pendant Δt, (l'intervalle de temps étant choisi dans ce but), la probabilité complémentaire (i.e. le système reste dans son état) est $1-\lambda_k \Delta t + o(\Delta t)$;
- la probabilité qu'un objet sorte de la file pendant Δt est égale à $\mu_k \Delta t + o(\Delta t)$
- les autres éventualités ont une probabilité en $o(\Delta t)$.

Ces probabilités ne dépendent ni du temps t ni de l'état X(t) dans lequel le système se trouve. Définissons par $p_{i,j}(\Delta_t)$, la probabilité de transition, qui ne dépend pas de l'instant t, probabilité que le processus X(t) fasse une transition de i vers j pendant la durée Δt :

$$p_{i,j}(\Delta_t) = p\left(X(t+\Delta t) = j \,|\, X(t) = i\right)$$

Par hypothèse les arrivées et les départs sont indépendants entre eux , il vient, en négligeant les termes d'ordre deux en Δt, les équations *0.1.1* :

Si $n \ge 0, \quad p_{n,n+1}(\Delta t) = \lambda_n \Delta t(1-\mu_{n+1}\Delta t) + o(\Delta t) = \lambda_n \Delta t + o(\Delta t)$

Si $n \ge 1, \quad p_{n,n}(\Delta t) = \lambda_{n-1}\Delta t \mu_{n+1}\Delta t + (1-\lambda_n \Delta t)(1-\mu_n \Delta t) + o(\Delta t) = 1-(\lambda_n + \mu_n)\Delta t + o(\Delta t)$

$p_{0,0}(\Delta t) = 1-\lambda_0 \Delta t + o(\Delta t)$

Si $n \ge 0, \quad p_{n+1,n}(\Delta t) = (1-\lambda_n \Delta t)\mu_{n+1}\Delta t + o(\Delta t) = \mu_{n+1}\Delta t + o(\Delta t)$

Alors que pour $|m-n| \ge 2, \quad p_{n,m}(\Delta t) = o(\Delta t)$

Ainsi le processus ne peut, à partir d'un état donné n, que passer dans l'un des états voisins n-1 ou n+1.

On obtient donc le graphe de transition suivant dans le cas général (i.e. avec $(\lambda_i)_{0\le i\le N-1}$ pour les naissances et $(\mu_i)_{1\le i\le N}$ pour les morts) :

0.2 - Chaînes de Markov en temps continu

On mesure le temps par un nombre réel positif ou nul. On étudie les processus aléatoires dont l'état est un entier positif ou nul (ici nombre de véhicules dans un système (file d'attente et service). On désigne par $\pi_i(t)$, la probabilité que le processus étudié soit dans l'état i à l'instant t, et l'on pose :

$$\pi(t) = (\pi_0(t), \quad \pi_1(t), \quad \ldots, \quad \pi_n(t))$$

0.2.1 - Vecteur stochastique

Un vecteur (ligne) formé de nombres réels $\pi = (\pi_0, \quad \pi_1, \quad \ldots, \quad \pi_n)$ est appelé stochastique si :

- $\forall i \in \{0,\ldots,n\}, \quad \pi_i \geq 0$; et
- $\sum_{i=0}^{n} \pi_i = 1$.

Remarque : le vecteur $\pi(t)$ est stochastique à tout instant t.

0.2.2 - Générateur stochastique infinitésimal

Une matrice réelle carrée $A = (a_{i,j})_{\substack{0 \leq i \leq n \\ 0 \leq j \leq n}}$ est appelée générateur stochastique infinitésimal si :

- $\forall i \in \{0,\ldots,n\}, \quad \sum_{j=0}^{n} a_{i,j} = 0$; et
- $\forall (i,j) \in \{0,\ldots,n\}^2, \quad \begin{cases} a_{i,j} \leq 0, & i = j \\ a_{i,j} \geq 0, & i \neq j \end{cases}$.

0.2.2 - Définition d'une chaîne de Markov (équation d'état)

La fonction $t \mapsto \pi(t)$ est une chaîne de Markov si et seulement si le vecteur stochastique $\pi(t)$ vérifie, à tout instant t, une équation différentielle ordinaire de la forme :

$$\frac{d}{dt}\pi(t) = \pi(t)A(t), \qquad \textit{(0.2.2.1)}$$

où A est un générateur stochastique infinitésimal.

Interprétation :
Cette définition peut se justifier de la manière suivante. Si l'on considère un intervalle de temps infiniment petit, (i.e. $\Delta t \to 0^+$), la matrice de transition $P(\Delta t)$ est voisine de l'identité. On suppose en effet que le système évolue très peu pendant ce temps Δt (il suffit de faire tendre Δt vers 0 dans les équations *0.1.1*).
On pose :
$P(\Delta t) = Id + \Delta t A + o(\Delta t)$
L'équation d'état pour un Δt tendant vers 0 s'apparente à l'équation d'état d'une chaîne de Markov dans le cas homogène (i.e. P indépendante de t) ; elle devient :
$\pi(t + \Delta t) = \pi(t)P(\Delta t)$
soit, $\pi(t + \Delta t) = \pi(t)(Id + \Delta t A) = \pi(t) + \Delta t \pi(t) A$,
donc $\dfrac{\pi(t + \Delta t) - \pi(t)}{\Delta t} = \pi(t)A$, il vient *(0.2.2.1)*

L'équation d'état a pour solution $\pi(t) = \pi(0)\exp(tA)$, avec $\exp(tA) = \sum_{k=0}^{+\infty} \frac{(tA)^k}{k!}$

Le calcul de l'exponentielle de A pouvant être laborieux on calcule $\pi(t)$ par discrétisation de la chaîne de Markov en temps continu ou bien par utilisation de la transformée de Laplace (et de sa transformée inverse).

1 – Première activité

1. 0 - Modélisation théorique du problème

Pour classifier les systèmes d’attente, on utilise la notation symbolique A/B/s/N où :
A est la distribution du temps entre deux arrivées successives
B est la distribution des durées de service
s est le nombre de stations services montées en parallèle
N est la capacité du système (serveur + file d’attente)

Ici on fait l’hypothèse que les N places de stationnement regroupent les places aux pompes (r places) et le nombre de places dans la file, étant donné que le nombre de véhicules X(t) servis ou en attente de l’être est compris entre 0 et N.

Le modèle proposé dans le projet correspond donc au schéma suivant :

$$P(\lambda)/P(\mu)/r/N$$

Ce système est modélisé par un processus de naissance et de mort tel que (avec les notations du § 0.1) :

$$\begin{cases} \lambda_k = \lambda, & \forall k \in \{0,1,\dots,N\} \\ \mu_k = k\mu, & \forall k \in \{1,\dots,r-1\} \\ \mu_k = r\mu, & \forall k \in \{r,\dots,N\} \end{cases}$$

On obtient la matrice de transition $N \times N$ pendant Δt suivante :

$$P(\Delta t) = \begin{array}{cc}
\begin{array}{ccccccc} & & & & r & & \\ & & & & \vdots & & \end{array} & \\
\begin{bmatrix}
1-\lambda\Delta t & \lambda\Delta t & 0 & \cdots & 0 & \cdots & 0 \\
\mu\Delta t & 1-(\lambda+\mu)\Delta t & \lambda\Delta t & & & & \\
0 & 2\mu\Delta t & 1-(\lambda+2\mu)\Delta t & & \vdots & & \\
 & 0 & 3\mu\Delta t & \ddots & & & \vdots \\
 & & 0 & \ddots & 0 & & \\
\vdots & & & \ddots & \lambda\Delta t & & \\
 & \vdots & & & 1-(\lambda+r\mu)\Delta t & \ddots & \\
 & & \vdots & & r\mu\Delta t & \ddots & 0 \\
 & & & & \vdots & \ddots & \lambda\Delta t \\
0 & 0 & 0 & \cdots & 0 & \cdots & 1-r\mu\Delta t
\end{bmatrix} &
\begin{array}{c} \\ \\ \\ \\ \\ \\ \cdots r \\ \\ \\ \\ \end{array}
\end{array}$$

1.1 – Calcul d’une distribution stationnaire

Théorème :
Une chaîne de Markov finie admet au mois une distribution stationnaire

La chaîne considérée étant finie elle admet au moins un distribution stationnaire.
On part de la formule $\dot{\pi}(t) = \pi(t)A(t)$. Le vecteur stochastique $\pi(t)$ est invariant dans le temps lorsque $\dot{\pi}(t) = 0$. C’est-à-dire lorsque $\pi(t)A(t) = 0$. Cela revient donc à résoudre le système linéaire

$$\boxed{\pi A = 0}$$

L’équation d’état nous donne les équations suivantes pour l’indice variant de 0 à r:

$$\begin{cases} \pi_0'(t) = -\lambda\pi_0(t) + \mu\pi_1(t) \\ \pi_1'(t) = \lambda\pi_0(t) - (\lambda+\mu)\pi_1(t) + 2\mu\pi_2(t) \\ \quad\vdots \\ \pi_r'(t) = \lambda\pi_{r-1}(t) - (\lambda + r\mu)\pi_r(t) + r\mu\pi_{r+1}(t) \end{cases} \quad (*)$$

et

$$\begin{cases} \pi_k'(t) = \lambda\pi_{n-1}(t) - (\lambda + r\mu)\pi_n(t) + r\mu\pi_{k+1}(t) \quad ; \quad \forall k \in \{r+1,\dots,N-1\} \\ \pi_N'(t) = \lambda\pi_{N-1}(t) - r\mu\pi_N(t) \end{cases}$$

soit en régime permanent

$\pi_0'(t) = \pi_1'(t) = \cdots = \pi_N'(t) = 0$;

i. e. $\pi_1 = \dfrac{\lambda}{\mu}\pi_0$; $\pi_2 = \dfrac{1}{2}\left(\dfrac{\lambda}{\mu}\right)^2 \pi_0$; et par récurrence $\forall k \in \{1,\dots,r\}$; $\pi_k = \dfrac{1}{k!}\left(\dfrac{\lambda}{\mu}\right)^k \pi_0$

Grâce à l’équation (*), on tire $\pi_{r+1} = \dfrac{1}{r(r!)}\left(\dfrac{\lambda}{\mu}\right)^{r+1}\pi_0$ et de proche en proche :

$\forall k \in \{r+1,\dots,N\}$; $\pi_k = \dfrac{1}{r^{k-r}r!}\left(\dfrac{\lambda}{\mu}\right)^k \pi_0$

On a donc la distribution stationnaire suivante, $\forall k \in \{1,\dots,r\}$ *et* $\forall p \in \{r+1,\dots,N\}$:

$$\boxed{\frac{\pi_\infty}{\pi_0} = \left(\frac{\lambda}{\mu};\ \dots ;\ \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k ;\ \dots ;\ \frac{1}{r!}\left(\frac{\lambda}{\mu}\right)^r ;\ \frac{1}{(r+1)!}\left(\frac{\lambda}{\mu}\right)^{r+1} ;\ \dots ;\ \frac{1}{r^{p-r}r!}\left(\frac{\lambda}{\mu}\right)^p ;\ \dots ;\ \frac{1}{r^{N-r}r!}\left(\frac{\lambda}{\mu}\right)^N\right.}$$

<u>Calcul de π_0</u>

On sait que $\sum_{k=0}^{n} \pi_k = 1$; avec les expressions de π_k déterminées précédemment, on trouve que

$$\pi_0 = \frac{1}{\displaystyle\sum_{k=0}^{r} \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \sum_{k=r+1}^{N} \frac{1}{r^{k-r}r!}\left(\frac{\lambda}{\mu}\right)^k}.$$

1.2 – Calcul d'espérances – Formules de Little

X(t) représentant le nombre de véhicules présents dans le système (file + service) au temps t, on détermine la longueur moyenne de la file totale (i.e. le nombre moyen de véhicules dans le système) en calculant E(X(t)). Quand le régime permanent est atteint,

$$E(X)=\sum_{k=0}^{N}k\pi_k=\sum_{k=1}^{r}k\pi_k+\sum_{k=r+1}^{N}k\pi_k\ ,$$

donc

$$E(X)=\sum_{k=1}^{r}\frac{k}{k!}\left(\frac{\lambda}{\mu}\right)^k\pi_0+\sum_{k=r+1}^{N}k\,\frac{1}{r^{k-r}r!}\left(\frac{\lambda}{\mu}\right)^k\pi_0$$

et

$$E(X)=\frac{\lambda\pi_0}{\mu}\sum_{k=1}^{r}\frac{1}{(k-1)!}\left(\frac{\lambda}{\mu}\right)^{k-1}+\frac{\pi_0 r^r}{r!}\sum_{k=r+1}^{N}k\left(\frac{\lambda}{r\mu}\right)^k .$$

Finalement, le *nombre moyen de véhicules dans le système* est :

$$\boxed{E(X)=\pi_0\left\{\frac{\lambda}{\mu}\sum_{k=0}^{r-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k+\frac{r^r}{r!}\sum_{k=r+1}^{N}k\left(\frac{\lambda}{r\mu}\right)^k\right\}}$$

Formules de Little

Avec les définitions suivantes

λ_e : inverse du temps moyen séparant deux clients consécutifs entrants effectivement dans le système

X : nombre de véhicules dans le système

T : temps de séjour dans le système

X_f : nombre de véhicules dans la file d'attente

T_f : temps de séjour dans la file

On a :

$$\begin{cases}E(X)=\lambda_e E(T)\\ E(X_f)=\lambda_e E(T_f)\end{cases}\quad \text{et}\quad E(T)=E(T_f)+\frac{1}{\mu}$$

avec $\lambda_e=\sum_{k=0}^{N-1}\lambda\pi_k^\infty=\lambda(1-\pi_N^\infty)=\lambda\left(1-\frac{\pi_0}{r^{N-r}r!}\left(\frac{\lambda}{\mu}\right)^N\right).$

Il vient donc que le *temps moyen de séjour dans le système* est :

$$\boxed{E(T)=\frac{\pi_0\left\{\frac{\lambda}{\mu}\sum_{k=0}^{r-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k+\frac{r^r}{r!}\sum_{k=r+1}^{N}k\left(\frac{\lambda}{r\mu}\right)^k\right\}}{\lambda\left\{1-\frac{\pi_0}{r^{N-r}r!}\left(\frac{\lambda}{\mu}\right)^N\right\}}}$$

De plus, $E(X_f)=E(X)-\frac{\lambda_e}{\mu}$

donc la *taille moyenne de la file d'attente* est :

$$\boxed{E(X_f)=\pi_0\left\{\frac{\lambda}{\mu}\sum_{k=0}^{r-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k+\frac{r^r}{r!}\sum_{k=r+1}^{N}k\left(\frac{\lambda}{r\mu}\right)^k\right\}-\frac{\lambda}{\mu}\left(1-\frac{\pi_0}{r^{N-r}r!}\left(\frac{\lambda}{\mu}\right)^N\right)}$$

On définit le taux d'occupation du système fini $P(\lambda)/P(\mu)/r/N$ par $\rho=\frac{\lambda_e}{r\mu}$

1.3 – Comparaison des valeurs théoriques et simulées

Le programme dont le code est donné en annexe permet de comparer les valeurs des taux moyens (longueur de la file, temps de séjour dans le système et taux d'occupation) calculées grâce aux formules ci-dessus.
Le programme modélise l'arrivée des véhicules suivant un processus de Poisson de paramètre lambda et leur départ suivant un processus de Poisson de paramètre mu. Un vecteur garde en mémoire l'évolution de la file d'attente du parking et de la file de la pompe sur un période T avec un pas Δt .
Le tableau suivant montre la forte corrélation entre les valeurs théoriques (aussi calculées par le programme) et les valeurs expérimentales. De plus le choix d'une période d'étude élevée influe sur la précision des résultats comme le montre les simulations n°2 et 3.

λ=1	**μ=2**	**N =10**	**r= 5 T= 10000**	**Δt=0.01**
Valeurs théoriques	Nombre moyen de voitures dans le système	Temps moyen de séjour dans le système	Taux d'occupation des pompes	Temps d'attente (1/μ)
	0.5	**0.5**	**0.1**	
Valeurs simulées	**0.5039**	**0.5081**	**0.1008**	**0.5**
λ=0.1	**μ=0.5**	**N =20**	**r=5 T= 1000**	**Δt=0.01**
Valeurs théoriques	Nombre moyen de voitures dans le système	Temps moyen de séjour dans le système	Taux d'occupation des pompes	Temps d'attente (1/μ)
	0.2	**2**	**0.04**	
Valeurs simulées	**0.1850**	**2.2835**	**0.0370**	**2**
λ=0.1	**μ=0.5**	**N =20**	**r=5 T= 10000**	**Δt=0.01**
Valeurs théoriques	Nombre moyen de voitures dans le système	Temps moyen de séjour dans le système	Taux d'occupation des pompes	Temps d'attente (1/μ)
	0.2	**2**	**0.04**	
Valeurs simulées	**0.2092**	**2.0514**	**0.0418**	**2**
λ=0.1	**μ=0.7**	**N =20**	**r=5 T=10000**	**Δt=0.01**
Valeurs théoriques	Nombre moyen de voitures dans le système	Temps moyen de séjour dans le système	Taux d'occupation des pompes	Temps d'attente (1/μ)
	0.1429	**1.4286**	**0.0286**	
Valeurs simulées	**0.1449**	**1.5019**	**0.029**	**1.4286**
λ=0.5	**μ=0.1**	**N =20**	**r=5 T=100000**	**Δt=0.01**
Valeurs théoriques	Nombre moyen de voitures dans le système	Temps moyen de séjour dans le système	Taux d'occupation des pompes	Temps d'attente (1/μ)
	11.2127	**23.7061**	**0.9460**	
Valeurs simulées	**11.1579**	**23.5252**	**0.9458**	**10**

1.4 - Recherche du couple (r,N) optimum

Avec les hypothèses suivantes :
C le coût d'entretien d'une place ;
D le coût d'entretien d'une pompe ;
b le bénéfice dégagée par un véhicule.

Il s'agit de minimiser la fonction suivante :

$$\boxed{Bénéfice \quad total = E(X)_p \, b - [(N - r)C + rD]}$$

avec $E(X)_p = \sum_{k=1}^{r} \mu k P(X=k) + \sum_{k=r+1}^{N} \mu r P(X=k) = E(X) - E(X_f)$ le nombre moyen de véhicules aux pompes.

(On teste pour des valeurs données de ***b, C, D*** **la valeur de la fonction définie ci-dessus sur un maillage de *(r,N)*.)**

2– Deuxième activité

Hypothèses :
(s ; S) représente les valeurs limites du stock
Z_n représente l'état du stock à t_n
$(D_n)_{n \in N}$ est la suite de variables aléatoires représentant la demande et qui suit les lois $(u_n)_{n \in N}$.

On note :
$\forall i \in N, \quad p(i)$ est la probabilité que i articles soient demandés.
Ensuite, $\forall i \in N, \quad r(i)$ est la probabilité qu'au moins i articles soient demandés.
Les évènements {i articles sont demandés} et {j articles sont demandés} sont indépendants pour $i \neq j$, on a donc :

$$\boxed{r(i) = 1 - p(0) - \cdot - p(i-1)}$$

Les probabilités que Z_n passe d'un état i à un état j étant connues, la suite $(Z_n)_{n \in N}$ est une chaîne de Markov prenant ses valeurs dans $\{0, \dots, S\}$. Sa matrice de transition de dimension S+1est la suivante, pour une période de temps donnée :

$$P_{(s,S)} = \left[\begin{array}{c|ccccccccc}
 & 0 & 1 & \cdots & s-1 & s & s+1 & & \cdots & S \\
\hline
0 & r(S) & p(S-1) & \cdots & p(S-s+1) & p(S-s) & p(S-s-1) & & \cdots & p(0) \\
1 & r(S) & p(S-1) & \cdots & p(S-s+1) & p(S-s) & p(S-s-1) & & \cdots & p(0) \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & & \vdots \\
s-1 & r(S) & p(S-1) & \cdots & p(S-s+1) & p(S-s) & p(S-s-1) & & \cdots & p(0) \\
s & r(s) & p(s-1) & \cdots & p(1) & p(0) & 0 & 0 & \cdots & 0 \\
s+1 & r(s+1) & p(s) & \cdots & \cdots & p(1) & p(0) & 0 & & 0 \\
\vdots & \vdots & \vdots & & & & & \ddots & \ddots & \vdots \\
 & & & & & & & & \ddots & 0 \\
S & r(S) & p(S-1) & \cdots & p(S-s+1) & p(S-s) & p(S-s-1) & \cdots & & p(0)
\end{array}\right]$$

(2)

2.1 – Probabilité stationnaire

Z étant indexé par $n \in N$, nous sommes dans le cas d'une chaîne de Markov en temps discret dont l'équation d'état est dans notre cas :

$$\boxed{\forall n \in N, \quad \pi_{n+1} = \pi_n P_{(s,S)}}$$

soit par récurrence et pour $Z_0=e_q=(0,.....,1,......0)$

$$\forall n \in N, \quad \pi_n = Z_0 P_{(s,S)}{}^n$$

On note toujours π^∞ la probabilité stationnaire. Si elle existe, elle est solution de l'équation

$$\boxed{\pi^\infty = \pi^\infty P_{(s,S)}}$$

2.1.1 Chaîne de Markov irréductible

Définition :
On dit qu'une chaîne de Markov est irréductible si la probabilité, partant d'un point k quelconque de l'espace des états, d'atteindre un point p quelconque de l'espace des états en un nombre $n_{k,p}$ d'étapes est strictement positive.

Théorème :
Une chaîne de Markov irréductible et finie (i.e. dont la variable aléatoire prend ses valeurs dans un ensemble fini) possède une unique probabilité stationnaire.

La chaîne de Markov dont la matrice de transition est donnée en *(2)* est irréductible, en effet on a toujours :

$$P\left(\exists n_{k,p} \geq 1, Z_{n_{k,p}} = k \middle| Z_0 = p\right) > 0$$

De plus l'ensemble $\{0, \ldots, S\}$ est fini, donc

$$\boxed{\text{la chaîne de Markov } (Z_n)_{n \in N} \text{ possède une unique probabilité stationnaire } \pi^\infty}$$

La probabilité stationnaire est telle que :

$\forall n \in N, P_{(s,S)}{}^n \xrightarrow[n \to \infty]{} U_S \, {}^t\!\left(\pi_0^\infty; \quad \cdots \quad ; \pi_S^\infty\right)$ où U_S est la matrice de dimension S+1 constituée de 1.

Une fois la loi $(u_i)_{i \in N}$ déterminée il suffit de chercher la limite de $P_{(s,S)}{}^n$ (analytiquement si c'est possible ou numériquement sinon) pour déterminer l'expression (ou la valeur) de π^∞

2.2 - Simulation de la chaîne avec des exemples pour la suite $(u_i)_{i \in N}$

En fait la suite $(p(i))_{i \in N}$ représente la suite des lois des demandes $(u_i)_{i \in N}$.

En effet, comme la suite $(D_n)_{n \in N}$ est identiquement distribuée on a :

$\forall (n;k) \in N^2, \quad P(D_n = k) = u_k$.

Il vient donc la matrice de transition suivante avec $\forall k \in N, \quad p(k) = u_k$

$$
P_{(s,S)} = \left[\begin{array}{c|ccccccccc}
 & 0 & 1 & \cdots & s-1 & s & s+1 & & \cdots & S \\
\hline
0 & 1-\sum_{n=0}^{S-1} u_n & u_{S-1} & \cdots & u_{S-s+1} & u_{S-s} & u_{S-s-1} & & \cdots & u_0 \\
1 & 1-\sum_{n=0}^{S-1} u_n & u_{S-1} & \cdots & u_{S-s+1} & u_{S-s} & u_{S-s-1} & & \cdots & u_0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & & \vdots \\
s-1 & 1-\sum_{n=0}^{S-1} u_n & u_{S-1} & \cdots & u_{S-s+1} & u_{S-s} & u_{S-s-1} & & \cdots & u_0 \\
s & 1-\sum_{n=0}^{s-1} u_n & u_{s-1} & \cdots & u_1 & u_0 & 0 & 0 & \cdots & 0 \\
s+1 & 1-\sum_{n=0}^{s} u_n & u_s & \cdots & \cdots & u_1 & u_0 & 0 & & 0 \\
\vdots & \vdots & \vdots & & & & & \ddots & \ddots & \vdots \\
 & & & & & & & & \ddots & 0 \\
S & 1-\sum_{n=0}^{S-1} u_n & u_{S-1} & \cdots & u_{S-s+1} & u_{S-s} & u_{S-s-1} & \cdots & & u_0
\end{array}\right]
$$

2.2.1 - En choisissant une loi géométrique $G_0(p)$ pour la suite $(u_i)_{i\in N}$

Dans ce cas si p est le probabilité du succès $G_0(p)$ est la loi de la variable aléatoire égale au nombre d'échec avant le premier succès. C'est la loi du temps d'attente de la réalisation d'un évènement donné.
On a pour cette loi :

$$\forall k \in N, \quad p(k) = \rho(1-\rho)^k$$

Donc sachant que $(u_i)_{i\in N}$ est géométrique de raison $(1-\rho)$, on a $\sum_{n=0}^{S-1} u_n = \frac{u_0 - u_S}{\rho} = 1-(1-\rho)^S$ et donc :

$$
P_{(s,S)} = \rho \left[\begin{array}{c|ccccccccc}
 & 0 & 1 & & s-1 & s & s+1 & & & S \\
\hline
0 & \frac{1}{\rho}(1-\rho)^S & (1-\rho)^{S-1} & & & \cdots & & & (1-\rho) & 1 \\
1 & \frac{1}{\rho}(1-\rho)^S & (1-\rho)^{S-1} & & & \cdots & & & (1-\rho) & 1 \\
\vdots & \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\
s-1 & \frac{1}{\rho}(1-\rho)^S & (1-\rho)^{S-1} & & & \cdots & & & (1-\rho) & 1 \\
s & \frac{1}{\rho}(1-\rho)^s & (1-\rho)^{s-1} & \cdots & (1-\rho) & 1 & 0 & 0 & \cdots & 0 \\
s+1 & \frac{1}{\rho}(1-\rho)^{s+1} & (1-\rho)^s & & \cdots & (1-\rho) & 1 & 0 & & 0 \\
\vdots & \vdots & \vdots & & & & & \ddots & \ddots & \vdots \\
 & & & & & & & & \ddots & 0 \\
S & \frac{1}{\rho}(1-\rho)^S & (1-\rho)^{S-1} & & & \cdots & & \cdots & (1-\rho) & 1
\end{array}\right]
$$

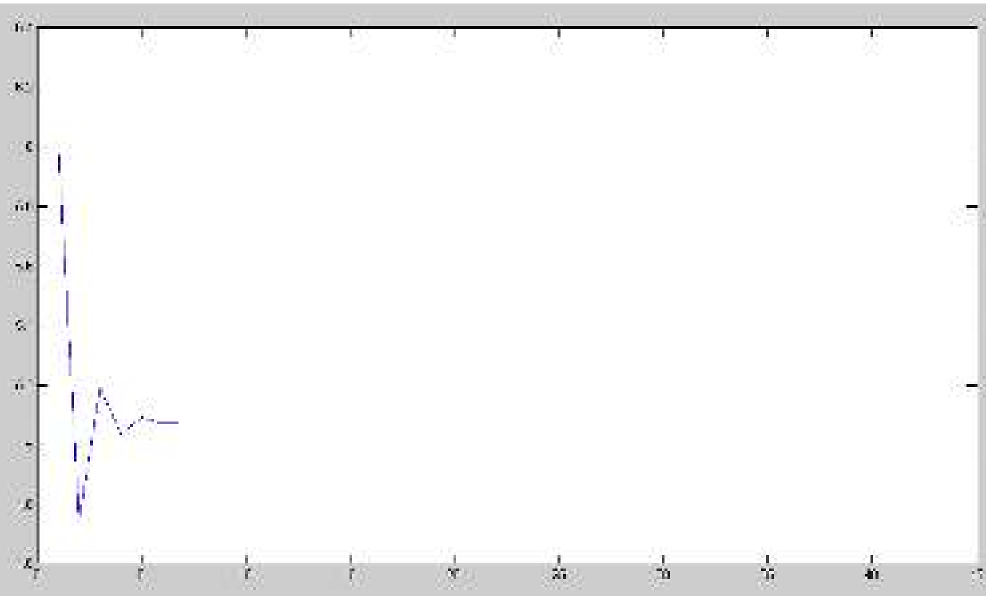
2.2.2 - En choisissant une loi de poisson pour la suite $(u_i)_{i\in N}$

Dans ce cas :

$$\forall k \in N, \quad p(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

$$P_{(s,S)} = e^{-\lambda}\left[\begin{array}{c|ccccccccc}
 & 0 & 1 & \cdots & s-1 & s & s+1 & & \cdots & S \\
\hline
0 & e^{\lambda}-\sum_{k=0}^{S-1}\frac{\lambda^k}{k!} & \frac{\lambda^{S-1}}{(S-1)!} & & & \cdots & & & \cdots & 1 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & & \vdots \\
s-1 & e^{\lambda}-\sum_{k=0}^{S-1}\frac{\lambda^k}{k!} & \frac{\lambda^{S-1}}{(S-1)!} & \cdots & & & & & \cdots & 1 \\
s & e^{\lambda}-\sum_{k=0}^{s-1}\frac{\lambda^k}{k!} & \frac{\lambda^{s-1}}{(s-1)!} & \cdots & & 1 & 0 & 0 & \cdots & 0 \\
s+1 & e^{\lambda}-\sum_{k=0}^{s}\frac{\lambda^k}{k!} & \frac{\lambda^{s}}{(s)!} & & \cdots & & 1 & 0 & & 0 \\
\vdots & \vdots & \vdots & & & & & \ddots & \ddots & \vdots \\
 & & & & & & & & \ddots & 0 \\
S & e^{\lambda}-\sum_{k=0}^{S-1}\frac{\lambda^k}{k!} & \frac{\lambda^{S-1}}{(S-1)!} & \cdots & & & & \cdots & & 1
\end{array}\right]$$

Dans les codes, nous avons utilisé la transposée de la matrice ci-dessus.
Pour étudier le comportement asymptotique de la chaîne, nous avons programmé un graphe qui montre une convergence vers un stock moyen.
Dans l'exemple nous avons choisi s=5, S=9, λ=3, Z_0=9 et n=45



Pour comparer les valeurs théoriques et expérimentales, nous avons utilisé le programme "comparaison". Nous y insérons les valeurs théoriques de la matrice, puis nous avons effectué des simulations à l'aide du programme "simulation". Après avoir créé une demande aléatoire qui suit la loi de poisson et qui calcule un nouveau stock à partir d'un stock initial, le programme génère un vecteur donnant les fréquences de chaque valeur de stock possible partant d'un stock initial donné.
Au bout de 100 000 étapes, les deux vecteurs sont égaux à 10^{-4} près.

2.3 – Temps moyen d'épuisement

Pour calculer, les temps moyen, nous avons d'abord utilisé un programme qui calcule la probabilité qu'il faille n étapes pour obtenir un stock nul. En entrée, il y a la matrice P, issue du programme "mat2". On calcule en fait :

$$P(X_n = 0 \cap X_{n-1} \neq 0) = \sum_{j=1}^{q} P(X_{n-1} = j).P(X_n = 0 \cap X_{n-1} = j)$$

On cherche ensuite l'espérance de la variable t qui suit la loi P_n

2.4 – Bénéfice moyen espéré en commandant x articles

Avec les hypothèses suivantes :
commander x articles coûte C_3+C_4x
le prix de vente d'un article est b
le coût de stockage est C_1
le coût de pénurie est C_2

Soit $b_q(x)$ ce bénéfice avec q le stock initial

Alors si $\forall n \in N, \quad D_n \leq q + x \Rightarrow b_q(x) = b(q+x)r(q+x) + b\sum_{i=0}^{q+x-1} ip(i) - C_1(q+x) - C_3 - C_4x$

$\forall n \in N, \quad D_n \geq q + x$ on introduit un coût de pénurie C2 et il vient :

$$b_q(x) = b(q+x)r(q+x) + b\sum_{i=0}^{q+x-1} ip(i) - C_1(q+x) - C_3 - C_4x - (D_n - (q+x))C_2$$

Le stock optimal est donné par la relation suivante :

$$\boxed{Stock_{optimal} = \max\left\{ i \in N \,\middle|\, r(i) > \frac{C_1 + C_4}{C_2 + b} \right\}}$$

Code de la question I-2 (Activité 1)

```
%Initialisation des variables et saisie des paramètres
lambda=input('Quelle est la valeur de lambda');
mu=input(''Quelle est la valeur de mu');
r=input(''Quelle est la valeur de r');
N=input('Quelle est la valeur de N');
delta_t=input(''Quelle est la valeur du pas de temps de l''étude');
T=input('Quelle est la valeur du temps de l''étude');
i=1;
parking_place_libre=N-r;
pompe_place_libre=r;
file_pompe_évolution=[];
file_parking_évolution=[];
nombre_voitures_sorties=0;
nombre_voitures_sorties_du_parking=0;
tps_moy_parking=0;
tps_moy_total=0;
nombre_voitures_pompes=0;
nombre_voitures_système=0;


while i<=T/delta_t
    % on teste si un véhicule va entrer dans le parking suivant un
processus de poisson de paramètre lambda et on fait évoluer la file_parking
et le nombre de places libres sur le parking en conséquence, s'il rentre.
    lambdatest=rand;
    if (parking_place_libre>0)&&(lambdatest<=lambda*delta_t)
        file_parking_évolution=[file_parking_évolution,i];
        parking_place_libre=parking_place_libre-1;
    end

    % on teste si un véhicule va sortir des pompes suivant un processus de
poisson de paramètre mu et on calcul les temps moyens de séjours et on
construit la file d'attente de la station, s'il sort.
    mutest=rand;
    if (pompe_place_libre~=r) && (mutest<=(r-pompe_place_libre)*mu*delta_t)
       tps_séjour_voiture=(i-file_pompe_évolution(1))*delta_t;
       tps_moy_total=(nombre_voitures_sorties*tps_moy_total+tps_séjour_voit
       ure)/(nombre_voitures_sorties+1);
        nombre_voitures_sorties=nombre_voitures_sorties+1;
        file_pompe_évolution1=[];
        for p=1:size(file_pompe_évolution,2)-1

file_pompe_évolution1=[file_pompe_évolution1,file_pompe_évolution(p+1)];
        end
        file_pompe_évolution=file_pompe_évolution1;
        pompe_place_libre=pompe_place_libre+1;
    end
```

```
% On teste si un véhicule du parking peut entrer dans la station
    if (pompe_place_libre~=0) && (parking_place_libre~=N-r)
        tps_séjour_voiture=(i-file_parking_évolution(1))*delta_t;
        tps_moy_parking=(nombre_voitures_sorties_du_parking*tps_moy_parking+
        tps_séjour_voiture)/(nombre_voitures_sorties_du_parking+1);

        nombre_voitures_sorties_du_parking=nombre_voitures_sorties_du_parkin
        g+1;

file_pompe_évolution=[file_pompe_évolution,file_parking_évolution(1)];
        file_parking_évolution1=[];
        for l=1:size(file_parking_évolution,2)-1

         file_parking_évolution1=[file_parking_évolution1,file_parking_évol
         ution(l+1)];
        end
        file_parking_évolution=file_parking_évolution1;
        parking_place_libre=parking_place_libre+1;
        pompe_place_libre=pompe_place_libre-1;
    end
     nombre_voitures_système=nombre_voitures_système+N-parking_place_libre-p
     ompe_place_libre;

     nombre_voitures_pompes=nombre_voitures_pompes+size(file_pompe_évolution
     ,2);
    i=i+1;
end
nbre_moyen_voiture=nombre_voitures_système/(T/delta_t)
tps_moy_total
tps_moy_parking
tps_moy_pompe=tps_moy_total-tps_moy_parking
taux_occupation=(nombre_voitures_pompes/(T/delta_t))/r


sigma1=0;
for k=r+1:N
    sigma1=sigma1+(1/(r^(k-r)*factorial(r)))*(lambda/mu)^k;
end
sigma2=0;
for k=0:r
    sigma2=sigma2+(1/factorial(k))*(lambda/mu)^k;
end
pizéro=1/(sigma1+sigma2);
sigma3=0;
for k=r+1:N
    sigma3=sigma3+k*(lambda/(r*mu))^k;
end
sigma4=0;
for k=0:r-1
    sigma4=sigma4+(1/factorial(k))*(lambda/mu)^k;
end
EX=pizéro*((lambda/mu)*sigma4+((r^r)/(factorial(r)))*sigma3)
Lambda_effectif=lambda*(1-(pizéro/(factorial(r)*r^(N-r)))*(lambda/mu)^N);
ET=EX/lambda_effectif
Tauxoccupth=min(lambda_effectif/(r*mu),1)
Attente=1/(mu)
```

Code de la question I-3 (Activité 1)

```
function [roptim,Noptim,benoptim]=optimisation(lambda,mu,b,C,D,n) %renvoie
une matrice donnant en (r,N) le bénéfice correpondant au couple (r, N)
P=-Inf*ones(n,2*n);
benoptim=-Inf;
for r=1:size(P,1)
    for N=r:size(P,2)
        P(r,N)=benefice(N,r,lambda,mu,b,C,D);
        if P(r,N)>benoptim
            roptim=r; Noptim=N;
            benoptim=P(r,N);
        end
    end
end
```

```
function [pn,Xt,e]=vectpi(lambda,mu,r,N)
%regime stationnaire
%lambda, mu paramètres d'entrée sortie
%r nombre de pompes
%N file d'attente plus nombre de pompes (nombre de places)
%Xt, espérance du nombre de personnes dans la station service
% e écart-type
% pn renvoie le vecteur où en ligne i on a la probabilité d'avoir i
véhicules dans le système

eta=lambda/mu;
u=zeros(1,(r+1));
for i=1:(r+1)
    u(1,i)=eta^(i-1)/fact(i-1);
end % i

v=zeros(1,N-r);
for j=1:(N-r)
    v(1,j)=(eta/r)^j;
end

p= 1/(sum(u)+eta^(r)*(sum(v))/fact(r));
%p= po= probabilité qu'il n'y ait aucun véhicule dans le système
for i=1:r+1
    pn(i,1)=p*u(1,i);% probabilité qu'il y ait  i véhicules dans le système
end
for i=(r+2):(N+1)
    pn(i,1)=pn(r+1,1)*(eta/r)^(i-r-1); % cas de saturation

end

y=0:1:N;
Xt=y*pn;
E=zeros(1,N+1);
for i=1:(N+1)
    E(1,i)=(i-1)^2;
end
e=((E*pn)-Xt^2)^(1/2);
```

```
function benef=benefice(N,r,lambda,mu,b,C,D) %bénéfice par unité de temps
[pn,Xt,e]=vectpi(lambda,mu,r,N);
mus=mu*[1:r]*pn(2:r+1)+mu*r*sum(pn(r+2:N+1)); %taux du processus de sortie
benef=b*mus-C*(N-r)-D*r;
```

Code de la question II-1 (Activité 2)

```
function M=graphe(s,S,lambda,Z0,n)
%Z0: stock initial
%n: nombre d'étapes
PX=zeros(S+1,1);
PX(Z0+1)=1;
M=[];
P=rep1(s,S,lambda);
for k=1:n
    PX=P*PX;
    N=[0:S];
    moy=N*PX;
    M=[M moy];
end
figure
plot(M);
```

Code de la question II-2 (Activité 2)

```
function [th,sim]=comparaison(s,S,lambda,e,stock0)
n=100000;
sim=simulation(n,lambda,s,S,stock0,e);
th0=zeros(S+1,1);
th0(stock0+1)=1;
P=ques1(s,S,lambda);
Pe=P^e;
th=Pe*th0;


function sim=simulation(n,lambda,s,S,stock0,e)
%n nombre de simulations
%e nbre d'étapes
%stock0 stock initial
for k=1:n
    stock1=stock0;
    for m=1:e
        stock1=demal(stock1,lambda,s,S);
    end
    stock(k)=stock1;
end
M=max(stock);
sim=zeros(M+1,1);
for m=1:n
    sim(stock(m)+1)=sim(stock(m)+1)+1/n;
end


function stock2=demal(stock1,lambda,s,S)
demande=poissrnd(lambda);
if stock1<=s
    if demande>S
        stock2=0;
    else stock2=S-demande;
    end
else
    if demande>stock1
        stock2=0;
    else stock2=stock1-demande;
    end
end
```

Code de la question II-3 (Activité 2)

```
function t=ques2(q,lambda)
u=loiu(q,lambda);
P=mat2(q,u);
t=stocknul(P);


function P=mat2(q,u)
P=zeros(q+1);
for j=1:q+1
    for i=2:q+1
        if j<i
            P(i,j)=0;
        else
            P(i,j)=u(j-i+1);
        end
    end
    P(1,j)=1-sum(P(2:q+1,j));
end


function m=stocknul(P)
m=0;
q=size(P,1)-1;
for k=1:2*q
    m=m+k*Pn(P,k);
end


function t=Pn(P,n)
%probabilité qu'il faille n essais pour obtenir zero, avec comme entrée la
matrice de%transition P et n,
% n nombre de tour
%q stock initial

t=0;
q=size(P,1)-1;
p=zeros(q+1,1);
p(q+1)=1;
p=P^(n-1)*p;
for k=2:q+1
    t=t+p(k)*P(1,k);
end
```