

---

## Compte-rendu

### Performance of CDMA sequences and Capacity of a CDMA System.

---

*Auteurs :*

Marouane BATTACH  
Clément BAUMANN

*Encadrant :*

Mohamad ASSAAD

---

**Résumé :** Code division multiple access (CDMA) est un système de codage des transmissions, utilisant la technique d'étalement de spectre. Il permet à plusieurs liaisons numériques d'utiliser simultanément la même fréquence porteuse.

Le but de ce TP est d'explorer les performances de la transmission d'information multiplexée avec le CDMA et puis l'évaluation de la capacité du système CDMA

**Mots clés :**

Télécommunication, Réseaux cellulaires, Système CDMA, Multiplexage, Traitement du signal, Matlab

---

## Table des matières

<b>1</b>	<b>Performance du multiplexage CDMA</b>	<b>2</b>
1.1	Générateurs de code . . . . .	2
1.2	Transmission des signaux dans un canal sans bruit . . . . .	4
1.3	Transmission sur un canal multivoies . . . . .	6
1.4	Le taux d'erreur binaire . . . . .	7
1.5	propriété du code CDMA . . . . .	9
<b>2</b>	<b>Capacité d'un système CDMA</b>	<b>14</b>
2.1	Génération de K utilisateurs uniformément distribués dans une cellule circulaire . . . . .	14
2.2	Pourcentage d'utilisateurs pour lesquels la condition $SINR_i \geq \gamma$ est satisfaite . . . . .	15
<b>3</b>	<b>Contrôle de la puissance dans un système CDMA à liaison montante : capacité et solution itérative</b>	<b>18</b>
3.1	Mise du système d'équation $SINR_i = \gamma$ sous forme matricielle . .	18
3.2	Génération de K utilisateurs uniformément distribués dans une cellule circulaire . . . . .	18
3.3	Nombre maximum d'utilisateurs tel que le système $(I - F) * P = b$ ait une solution . . . . .	19
3.4	Calcul des puissances allouées . . . . .	20
3.5	Algorithme itératif d'allocation de puissance . . . . .	20
<b>4</b>	<b>Estimation de la capacité de trafic d'un système CDMA par une simulation hybride et une évaluation analytique</b>	<b>22</b>
4.1	Performance en l'absence d'interférences inter-cellule . . . . .	23
4.1.1	Génération des utilisateurs . . . . .	23
4.1.2	Répartition des conditions radio et consommation de puissance . . . . .	23
4.1.3	Evaluation de la performance en utilisant la méthode d'Erlang . . . . .	25
4.2	Prise en compte des interférences inter-cellule . . . . .	27
4.2.1	CDF de la consommation de puissance sur la cellule, intégrant les interférences inter-cellules . . . . .	27

# 1 Performance du multiplexage CDMA

Cette partie s'intéresse à l'étude des performances du multiplexage CDMA. on considère une liaison descendante où une station de base transmet plusieurs signaux en parallèle à différents utilisateurs en codant chaque signal avec un code spécifique de longueur  $SF$ . Les signaux des différents utilisateurs ont différents codes. On suppose que tous les codes ont la même longueur  $SF$ .

## 1.1 Générateurs de code

On présente ici trois façons pour générer les codes CDMA :

1. **Walsh-Hadamard codes** : Les matrices de Walsh-Hadamard de dimension  $SF = 2^k$  pour  $k \in \mathbb{N}$  sont générées récursivement par la formule suivante :

$$H(2^1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$H(2^2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

pour  $k \in \mathbb{N}^*$ ,

$$H(2^k) = \begin{bmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{bmatrix}$$

Dans Matlab, la fonction `hadamard` dans `matlab` génère cette matrice en indiquant la taille de la matrice.

```
SF=4 %16,128
codes=hadamard(SF)
```

On obtient :

$$SF = 4 \longrightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$SF = 16 \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

## 2. Binary random sequence : génération aléatoire des séquences

Dans Matlab, on génère ces séquences de la façon suivante :

```
SF=4 %16,128 ;
code=(round(rand(1,SF))-ones(1,SF)./2)*2;
```

On obtient par exemple :  $SF = 4 \rightarrow \text{code} = [-1, -1, 1, 1]$

$SF = 16 \rightarrow \text{code} = [1, 1, -1, 1, -1, 1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1]$

## 3. PN séquence : génération des pseudo codes aléatoires avec l'algorithme suivant $c_n = c_{n-1} + c_{n-4} \pmod{2}$

On implémente cet algorithme sur Matlab :

```
% PN sequence : Generation of pseudo random codes of ...
length 15
seq=zeros(1,15);
seq(1)=0;
seq(2)=0;
5 seq(3)=0;
seq(4)=1 ;
for n=5:1:15
    seq(n)=mod(seq(n-1)+seq(n-4),2);
end
10 binary_seq=2*seq-1; %sequence of 1 and -1
codes=zeros(15,15);
cr=binary_seq;
for i=1:15
    codes(i,:)=cr ;
```

```

15      cr=circshift(cr,1) ;
      end

```

On obtient une matrice 15x15. Les pseudo codes aléatoires sont les lignes de cette matrice. On obtient par exemple :

$$\text{codes}(0) = [-1, -1, -1, 1, 1, 1, 1, -1, 1, -1, 1, 1, -1, -1, 1]$$

## 1.2 Transmission des signaux dans un canal sans bruit

On considère deux usagers qui transmettent des signaux dans un canal sans bruit. Les deux signaux binaires des deux usagers sont les suivants :

$$x_1 = [-1, 1, -1, 1, -1, -1, -1, 1, 1]$$

$$x_2 = [-1, 1, -1, -1, -1, -1, -1, -1, 1]$$

Chaque usager applique un code CDMA à son signal de longueur  $SF$ . Les deux codes CDMA sont orthogonaux et générés par les matrices de Walsh-hadamard.

Ce graphe reste le même pour différentes valeurs de  $SF$ . On conclut que dans le cas d'un canal sans bruit le signal détecté ne subit aucune distorsion peu importe la longueur de  $SF$  à condition que  $SF$  soit plus grand que le nombre des usagers.

Pour réduire le nombre ligne de code , on définit des fonctions de transmission et de décodage comme suit :

```

%signal transmis dans le canal
%paramtres : x--->s quence des caract res      transmettre
%              c---> code CDMA
%output : s---> signaltransmis dans le canal
5 function s=transmitted_signal(x,c)
s=[];
for i=x
    signal= i*c;
    s=[s signal];
10 end

```

```

% fonction de d codage du signal re u
%paramtres : s--> signal d tect par le r cepteur
%              c---> code CDMA
%output : d---> signal d cod
5 function d=decode(s,c)
    L=length(c);
    start=1;
    ending=L;
    d=[];
10 while ending<=length(s)

        signal=s(start:ending)*c'*1/L;
        if abs(signal-1)>abs(signal+1)
            signal=-1;
15         else
            signal=1;
        end
        d=[d signal];
        start=ending+1;
20         ending=ending+L;
    end

```

```

%% transmission et d codage

x1=[-1 1 -1 1 -1 -1 -1 1 1];
x2 = [-1 1 -1 -1 -1 -1 -1 -1 1 ];
5

SF=4;
c=code(SF,1); %Generation des codes par les matrices de ...
    Walsh-Hadamard
signal_transmis_1=transmitted_signal(x1,c{1});
10 signal_transmis_2=transmitted_signal(x2,c{2});
    detected_signal=signal_transmis_1+signal_transmis_2;

    decoded_signal_1=decode(detected_signal,c{1});
    decoded_signal_2=decode(detected_signal,c{2});
15

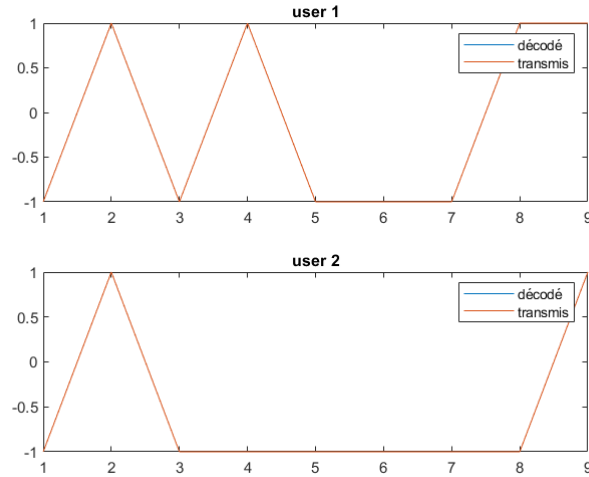
t1=[1:length(x1)];
subplot(2,1,1);plot(t1,decoded_signal_1,t1,x1)
title('user 1')
legend('d cod ','transmis')

```

```

20 t2=[1:length(x1)];
   subplot(2,1,2);plot(t2,decoded_signal_2,t2,x2)
   title('user 2')
   legend('d cod ', 'transmis')
   hold off

```



**FIGURE 1** – signaux transmis et détectés pour les deux usagers

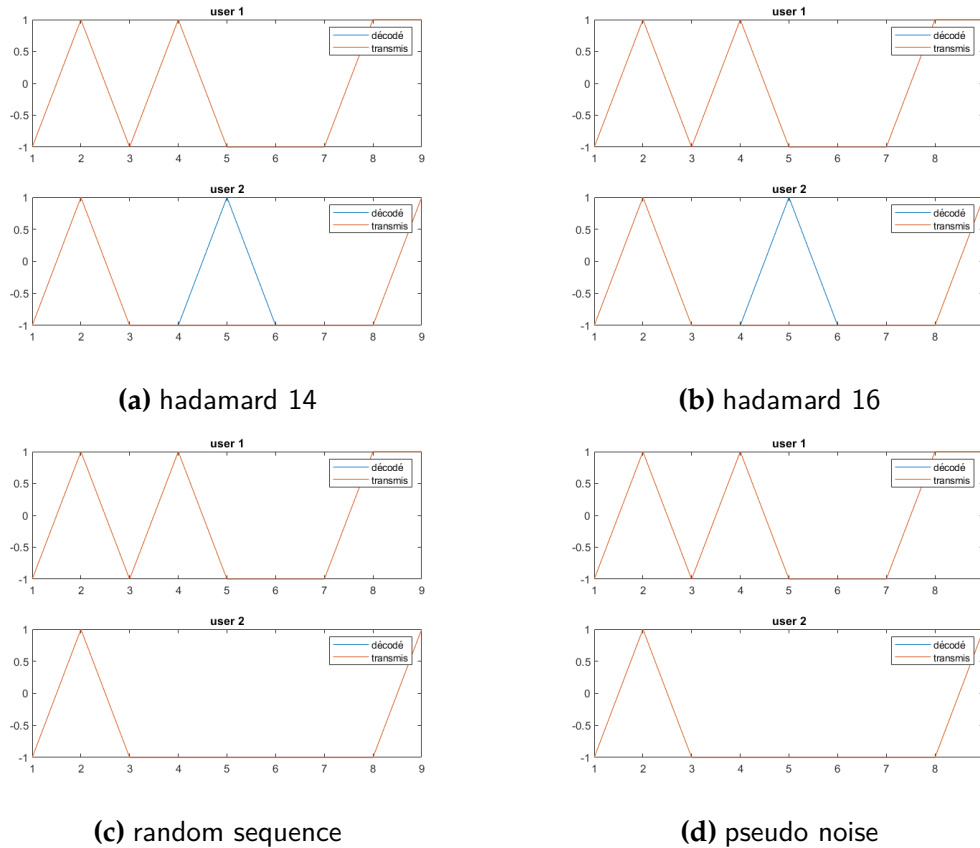
### 1.3 Transmission sur un canal multivoies

On considère maintenant que les deux signaux sont transmis dans un canal avec deux voix différentes de la forme  $h(t) = \delta(t) + 0.9\delta(t - T_c)$  tel que  $T_c$  est le retard subi par le signal dans la deuxième voie. On utilise toujours le signal des deux usagers de la première partie.

$$x_1 = [-1, 1, -1, 1, -1, -1, -1, 1, 1]$$

$$x_2 = [-1, 1, -1, -1, -1, -1, -1, -1, 1]$$

La simulation nous montre les signaux transmis et les signaux détectés pour les différentes méthodes de génération de Code qu'on a vu dans la première partie. On observe que les codes de Walsh-Hadamard ne sont pas assez performants pour dans cette situation contrairement aux codes PN et Random Sequences.



**FIGURE 2** – signaux transmis et détectés en multicanal

## 1.4 Le taux d'erreur binaire

Pour comparer les différents codes, on va calculer le nombre moyen d'erreurs dans les signaux détectés ainsi que le taux d'erreur. D'abord on génère deux signaux aléatoires de longueur  $N = 1000$  ou  $N = 10000$ , puis on suit le processus de transmission et de décodage comme fait des les parties précédentes. on répète ce processus plusieurs fois ( $T$  itérations) pour calculer une moyenne. Dans notre comparaison, on considère la moyenne de l'erreur des deux utilisateurs.

On exécute le code suivant pour toutes les méthodes de génération du code CDMA vu précédemment.

```
%% Bit error rate
```



```

SF=16; % longueur des codes
N=1000;

5
%G nration des signaux      transmettre alatoirement
x1=(round(rand(1,N))-ones(1,N)./2)*2; % binary random code {-1,1}
x2=(round(rand(1,N))-ones(1,N)./2)*2; % binary random code {-1,1}

10 %generation de codes
c1=code(SF,1);
c2=code(SF,2);
c3=code(SF,3);

15 %Transmission par code CDMA
%hadamard
signal_transmis_1_1=transmitted_signal(x1,c1{1});
signal_transmis_2_1=transmitted_signal(x2,c1{2});
%random noise
20 signal_transmis_1_2=transmitted_signal(x1,c2{1});
signal_transmis_2_2=transmitted_signal(x2,c2{2});
%pseudo noise
signal_transmis_1_3=transmitted_signal(x1,c3{1});
signal_transmis_2_3=transmitted_signal(x2,c3{2});

25
%Ajout des signaux venant des autres canaux
signal_transmis_1_1=[signal_transmis_1_1 0]+0.9*[0 ...
    signal_transmis_1_1];
signal_transmis_2_1=[signal_transmis_2_1 0]+0.9*[0 ...
    signal_transmis_2_1];

30 signal_transmis_1_2=[signal_transmis_1_2 0]+0.9*[0 ...
    signal_transmis_1_2];
signal_transmis_2_2=[signal_transmis_2_2 0]+0.9*[0 ...
    signal_transmis_2_2];

signal_transmis_1_3=[signal_transmis_1_3 0]+0.9*[0 ...
    signal_transmis_1_3];
signal_transmis_2_3=[signal_transmis_2_3 0]+0.9*[0 ...
    signal_transmis_2_3];

35

detected_signal_1=signal_transmis_1_1+signal_transmis_2_1;
detected_signal_2=signal_transmis_1_2+signal_transmis_2_2;
detected_signal_3=signal_transmis_1_3+signal_transmis_2_3;

40
%D codage des signaux

```

```

decoded_signal_1_1=decode(detected_signal_1,c1{1});
decoded_signal_2_1=decode(detected_signal_1,c1{2});

45 decoded_signal_1_2=decode(detected_signal_2,c2{1});
decoded_signal_2_2=decode(detected_signal_2,c2{2});

decoded_signal_1_3=decode(detected_signal_3,c3{1});
decoded_signal_2_3=decode(detected_signal_3,c3{2});
50
%error
avg_error_1=N-(sum(decoded_signal_1_1==x1)+sum(decoded_signal_2_1==x2))/2;
error_rate_1=avg_error_1/N

55 avg_error_2=N-(sum(decoded_signal_1_2==x1)+sum(decoded_signal_2_2==x2))/2;
error_rate_2=avg_error_2/N

avg_error_3=N-(sum(decoded_signal_1_3==x1)+sum(decoded_signal_2_3==x2))/2;
error_rate_3=avg_error_3/N

```

Après avoir exécuté

N	Hadamard 4	Hadamard 16	Random sequence	Pseudo noise
1000	0.0632	0.0645	0	0
10000	0.0622	0.0609	0.0629	0

Le pseudo noise et la random sequence semblent plus performants que les codes de walsh hadamard dans le cas du multitrajet.

## 1.5 propriété du code CDMA

Pour comprendre pourquoi le Pseudo noise est plus performant que les autres codes CDMA, On compare les corrélations croisée et les auto-corrélations des différents codes.

L'auto-corrélation montre la mesure de la similarité entre le code et sa copie décalée d'un lag de temps  $K$ . Pour cette raison, les codes qui ont les meilleures propriétés d'auto-corrélation ont été fréquemment utilisés pour supprimer l'asynchronicité dans les systèmes de communication. L'auto-corrélation peut être exprimée comme ci-dessous.

$$\text{corr 11}(n) = \frac{1}{SF} \sum_{k=0}^{SF-1} c_k^1 c_{k-n}^1$$

$$\text{corr22}(n) = \frac{1}{SF} \sum_{k=0}^{SF-1} c_k^2 c_{k-n}^2$$

la corrélation croisée est la mesure de la similarité entre deux codes différents. En d'autres termes, la corrélation croisée décrit l'interférence entre les codes  $c^1$  et  $c^2$

$$\text{corr12}(n) = \frac{1}{SF} \sum_{k=0}^{SF-1} c_k^1 c_{k-n}^2$$

```

%% CDMA codes propreties
SF=16;

%hadamard
5  c_H=code(SF,1);
   auto_Corr_1=xcorr(c_H{1})
   auto_Corr_2=xcorr(c_H{2})
   Corr_12=xcorr(c_H{1},c_H{2})
   subplot(1,3,1);plot(auto_Corr_1)
10  title('autocorrlation-1')

   subplot(1,3,2);plot(auto_Corr_2)
   title('autocorrlation-2')

15  subplot(1,3,3);plot(Corr_12)
   title('corrlation crois 1 et2 ')
   hold off
   %% noise
   c_H=code(SF,2);
20  auto_Corr_1=xcorr(c_H{1})
   auto_Corr_2=xcorr(c_H{2})
   Corr_12=xcorr(c_H{1},c_H{2})
   subplot(1,3,1);plot(auto_Corr_1)
   title('autocorrlation-1')
25  subplot(1,3,2);plot(auto_Corr_2)
   title('autocorrlation-2')

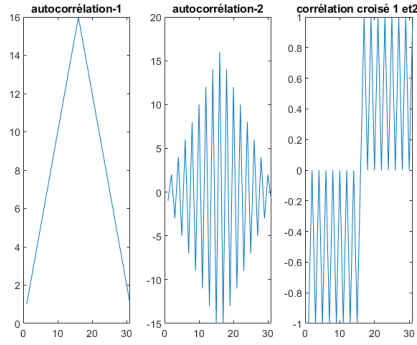
   subplot(1,3,3);plot(Corr_12)
30  title('corrlation crois 1 et2 ')
   hold off

   %% pseudo noise
   c_H=code(15,3);

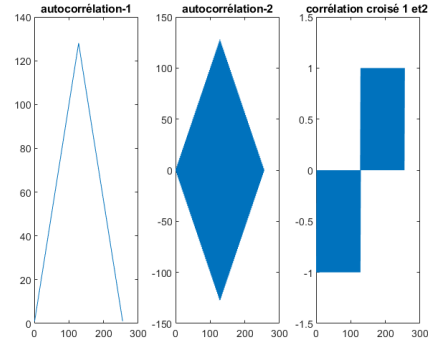
```

```
35 auto_Corr_1=xcorr(c_H{1})
   auto_Corr_2=xcorr(c_H{2})
   Corr_12=xcorr(c_H{1},c_H{2})
   subplot(1,3,1);plot(auto_Corr_1)
   title('autocorrlation-1')
40
   subplot(1,3,2);plot(auto_Corr_2)
   title('autocorrlation-2')

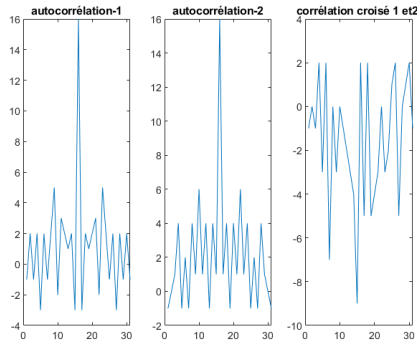
   subplot(1,3,3);plot(Corr_12)
45 title('corrlation crois 1 et2 ')
   hold off
```



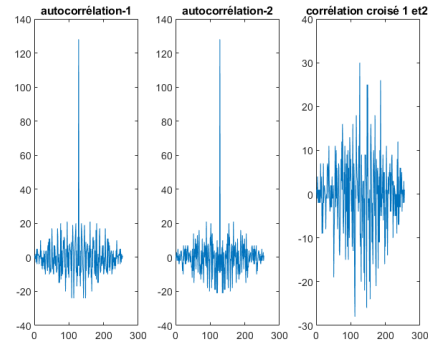
(a) hadamard 16



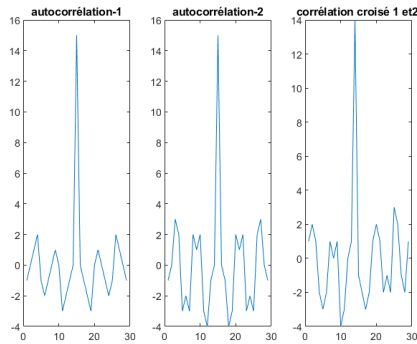
(b) hadamard 128



(c) random noise 16



(d) random noise 128



(e) pseudo noise

**FIGURE 3** – corrélation des séquences CDMA

L'orthogonalité est la propriété la plus importante des codes Hadamard-Walsh. En raison de cette propriété d'orthogonalité, la corrélation croisée entre deux codes Hadamard-Walsh de la même matrice est nulle, lorsque le système est parfaitement synchronisé. L'étalement de code de Walsh peut être utilisé

si tous les utilisateurs du même canal sont synchronisés dans le temps, car la corrélation croisée entre différents décalages de codes de Walsh n'est pas nulle.

Une séquence de pseudo-bruit (PN) est une séquence binaire de 1 et de 0 et elle est périodique. Ce sont certaines caractéristiques qui sont similaires aux séquences binaires aléatoires (ayant un nombre égal de 0 et de 1), une très faible corrélation entre deux versions décalées de la séquence et une faible corrélation croisée entre deux séquences. La séquence pseudo-aléatoire n'est pas aléatoire (déterministe) mais elle semble aléatoire pour l'utilisateur qui ne connaît pas le code.

## 2 Capacité d'un système CDMA

### 2.1 Génération de $K$ utilisateurs uniformément distribués dans une cellule circulaire

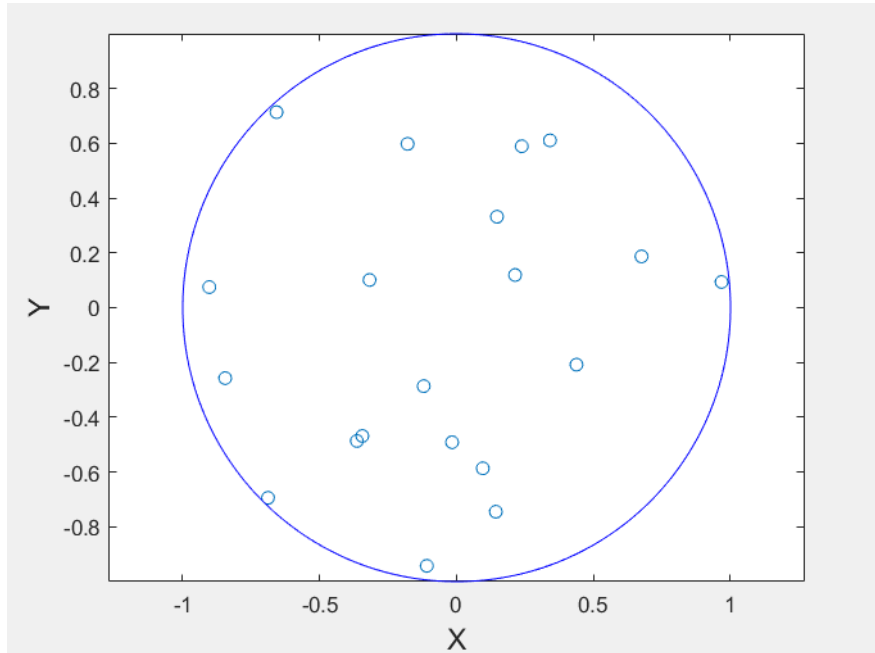
L'implémentation sous Matlab est la suivante :

```
% we will consider that the center of the circle is the point ...
    (0,0)
center = [0 0];
radius = 1;

5 angles = linspace(0, 2*pi, 500);
  rx = radius * cos(angles) ;
  ry = radius * sin(angles) ;

K=20; % 30,50,90,100
10 % we create the two lists of cartesian coordinates of the K points
  rho = radius*sqrt(rand(1,K));
  phi = 2*pi*rand(1,K);
  x = rho.*cos(phi);
  y = rho.*sin(phi);
15
  % we plot the points
  plot(x,y, 'o');
  hold on;
  plot(rx, ry, 'b-')
20 axis equal;
  xlabel('X', 'FontSize', 14);
  ylabel('Y', 'FontSize', 14);
```

On obtient :



**FIGURE 4** – Positions de 20 utilisateurs uniformément distribués sur un disque de rayon 1

## 2.2 *Pourcentage d'utilisateurs pour lesquels la condition $SINR_i \geq \gamma$ est satisfaite*

Tout d'abord, on génère un coefficient de canal pour chaque utilisateur. Le code Matlab est le suivant :

```
% we first compute the path loss effect coefficient
l = 10.^(-12.81-3.76*log10(rho)); % rho is indeed the ...
    distances from the center matrix
% then we compute the fading effect coefficient
h = exprnd(1/lambda,1,K);
5 % finally we compute the channel coefficients matrix
g = l.*h;

% we compute the SINR for each user
SINR = zeros(1,K);
10 for i=1:K
    SINR(i) = ...
        ((W/R)*(P/K)*g(i))/(theta*(K-1)*(P/K)*g(i)+sigma_carre);
```



end

Ensuite, on calcule le pourcentage d'utilisateurs satisfaisant la condition  $SINR_i \geq \gamma$  :

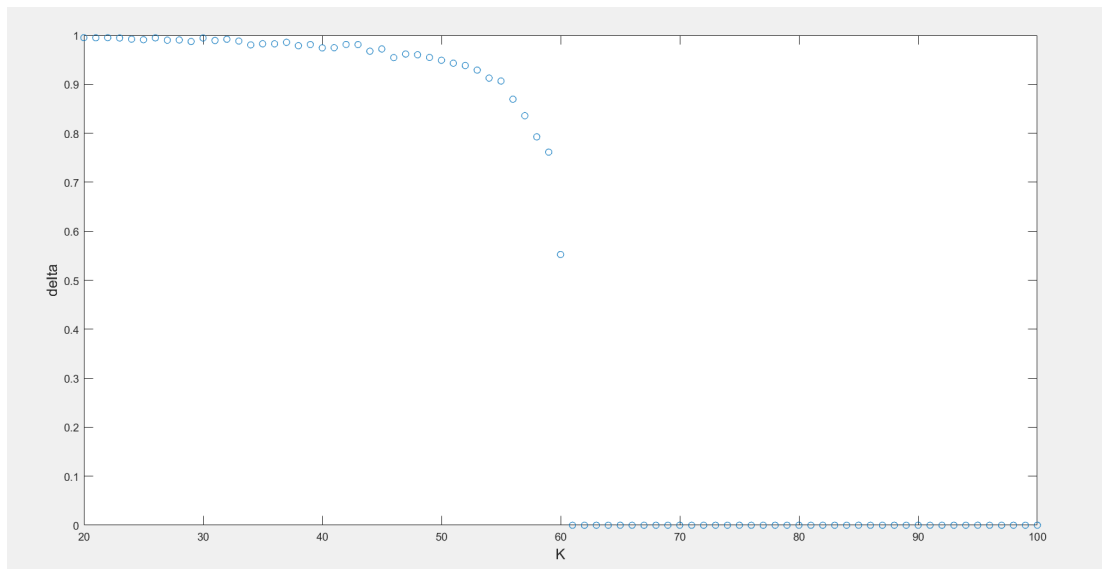
```
% percentage of users for which SINRi ≥ gamma
nb_sataisfied = 0;
for i=1:K
    if SINR(i) ≥ gamma
5       nb_sataisfied = nb_sataisfied+1;
    end
end
satisfied_rate = nb_sataisfied/K
```

Pour  $K = 20$ , on trouve toujours un pourcentage de satisfaction de 100%.

On répète le calculs des coefficients de canal et du pourcentage de satisfaction 3100 fois. On calcule ensuite la moyenne des pourcentages de satisfaction obtenus, notée  $\delta$ .

Pour  $K = 20$ , on trouve  $\delta = 1$ .

On répète pour des valeurs de  $K$  allant de 20 à 100 :



**FIGURE 5** – Pourcentage moyen de satisfaction de la condition  $SINR_i \geq \gamma$  en fonction du nombre d'utilisateurs

Le nombre maximum d'utilisateurs pour lequel pour lequel la condition  $\delta \geq 90\%$  est  $K = 55$ .

### 3 Contrôle de la puissance dans un système CDMA à liaison montante : capacité et solution itérative

#### 3.1 Mise du système d'équation $SINR_i = \gamma$ sous forme matricielle

la condition  $SINR_i = \gamma$  pour tous les usagers peut être formulé comme suit :

$$SINR_i = \frac{W}{R} \frac{p_i g_i}{\theta \sum_{j \neq i}^K p_j g_j + \sigma^2} = \gamma \quad \text{pour } i = 1, \dots, K$$

Ce qui implique

$$p_i g_i = \frac{\gamma R}{W} \left( \theta \sum_{j \neq i}^K p_j g_j + \sigma^2 \right) \quad \text{pour } i = 1, \dots, K$$

Donc

$$p_i - \frac{\gamma R}{W g_i} \left( \theta \sum_{j \neq i}^K p_j g_j \right) = \frac{\gamma R}{W g_i} \sigma^2 \quad \text{pour } i = 1, \dots, K$$

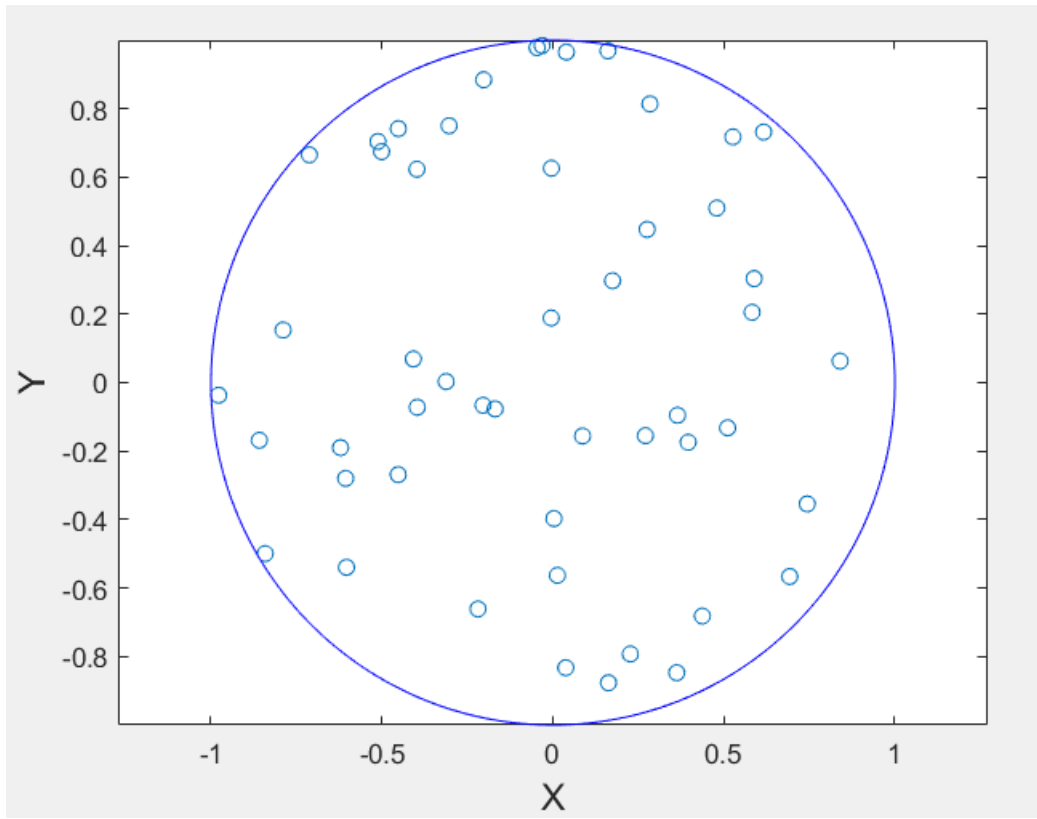
Ce système d'équations peut s'écrire alors comme suit

$$(I - F)P = b \quad \text{tel que } P = [p_1, \dots, p_K]^T ; b = \left[ \frac{R\gamma\sigma^2}{Wg_1}, \dots, \frac{R\gamma\sigma^2}{Wg_K} \right]^T ;$$

$$F = \theta \frac{R\gamma}{W} \begin{bmatrix} 0 & \frac{g_2}{g_1} & \dots & \frac{g_K}{g_1} \\ \frac{g_1}{g_2} & 0 & \dots & \frac{g_K}{g_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{g_1}{g_K} & \frac{g_2}{g_K} & \dots & 0 \end{bmatrix}$$

#### 3.2 Génération de $K$ utilisateurs uniformément distribués dans une cellule circulaire

On réutilise le même code Matlab qu'à la question 2.1. Pour  $K = 50$ , on obtient par exemple :



### 3.3 Nombre maximum d'utilisateurs tel que le système $(I - F) * P = b$ ait une solution

On sait que ce système possède une solution si  $\rho(F) < 1$ .  
Le code Matlab pour déterminer le rayon spectral de F est :

```
% first we compute the pathloss coefficients
g = 10.^(-12.81-3.76*log10(rho));

% for R and gamma, we give the values (R1, gamma1) to the ...
% first half of the
5 % users and the values (R2, gamma2) to the other half
R=zeros(1,K);
gamma=zeros(1,K);
for i=1:K
    if i<=K/2
10        R(i)=R1;
```

```

        gamma(i)=gamma1;
    else
        R(i)=R2;
        gamma(i)=gamma2;
15    end
end

% then, we compute the matrix F
F = zeros(K,K);
20 for i=1:K
    for j=1:K
        if i≠j
            F(i,j) = (R(i)*gamma(i)*theta*g(j))/(W*g(i));
        end
    end
25 end

% Finally, the spectral radius is
spec_rad = max(abs(eig(F)));

```

On répète ces calculs pour différentes valeurs de  $K$ , le nombre d'utilisateurs. On trouve que le nombre maximal d'utilisateurs satisfaisant  $\rho(F) < 1$  est  $K_{max} = 93$ .

### 3.4 Calcul des puissances allouées

On calcul la matrice des puissance allouées  $P$  sur Matlab à l'aide de la formule  $P = (I - F)^{-1} * b$ .

```

% computation of matrix b
b = ((sigma_carre/W)*R.*gamma./g).';

% computation of the allocated power matrix (F has already ...
    been computed in
5 % the previous question
P = inv(eye(K)-F)*b;

```

### 3.5 Algorithme itératif d'allocation de puissance

On implémente l'algorithme itératif d'allocation de puissance, puis on compare le résultat avec la matrice obtenue avec la formule matricielle.

```

n = 300; % number of iterations

P_it = 0.01*ones(K,1); % initialisation of the allocated ...
    power matrix : 0.01W

5 t=0; % iteration counter
  while t<n

      new_P_it = zeros(K,1);
      for i=1:K
10         new_P_it(i) = ...
            ((R(i)*gamma(i))/(W*g(i)))*(theta*(sum(P_it.*(g.'))-P_it(i)*g(i))+sigma)
      end

      P_it = new_P_it;
      t = t+1;
15 end

% Compute the medium percentage of error over the power allocated
error = abs(P-P_it)./P ;
mean_error = sum(error)/K

```

On cherche le nombre d'itérations nécessaires pour obtenir une erreur inférieure à 1% ou 0.1% pour 20, 30, 50 et 93 utilisateurs.

	erreur $\leq 1\%$	error $\leq 0.1\%$
$K = 20$	$n \geq 6$	$n \geq 7$
$K = 30$	$n \geq 8$	$n \geq 10$
$K = 50$	$n \geq 18$	$n \geq 20$
$K = 93$	$n \geq 30$	$n \geq 500$

## 4 Estimation de la capacité de trafic d'un système CDMA par une simulation hybride et une évaluation analytique

Les utilisateurs sont répartis aléatoirement dans la cellule, considérée comme un cercle de rayon 1 Km. Le BS est situé au centre de la cellule à étudier et la cellule est isolée (pas d'interférence inter-cellule).

On considère la liaison descendante où la BS transmet les données aux utilisateurs à un débit  $R$ . Les paquets reçus par chaque utilisateur peuvent être décodés correctement si le SINR reçu (au niveau du bit) est supérieur ou égal à un seuil  $g_b$ . Ce seuil est de 2,8 dB pour un débit  $R = 128$  kbit/s. La puissance de la transmission totale de la station de base est  $P_{max} = 10$  Watt. La puissance d'émission pour l'utilisateur  $i$  est notée  $P_i$ . La puissance reçue est alors donnée par  $P_{r,i} = P_i g_i$  où  $g_i$  est la perte de signal composée de les composants suivants :

$$g_i = \frac{G h_i s_i}{l_i L}$$

- La perte de trajet est exprimée par  $l_i = \alpha d_i^\beta$ ,  $\alpha = 10^{13.6}$ ,  $\beta = 3.6$ ,  $d_i$  est la distance en Km entre l'utilisateur et la BS.
- Le gain d'antenne égal à  $G = 2$  (3 dB)
- La perte due aux différentes imperfections des équipements est égale à  $L = 10^{0.7}$  (7dB).
- L'évanouissement lent est modélisé comme une variable log-normale :  $s_i = 10^{\frac{N(0,S)}{10}}$ , avec  $S = 3,5$  (l'évanouissement en dB est une variable gaussienne de moyenne 0 et d'écart type 3,5 dB).
- L'effet d'évanouissement rapide est modélisé comme une variable aléatoire exponentielle de paramètre  $\frac{1}{2}$

En l'absence d'interférence intercellulaire, la puissance à transmettre pour l'utilisateur  $i$  est calculée comme suit :

$$P_i = \frac{\theta \gamma_c}{1 + \theta \gamma_c} P_{max} + \frac{\gamma_c}{1 + \theta \gamma_c} \frac{N}{g_i}$$

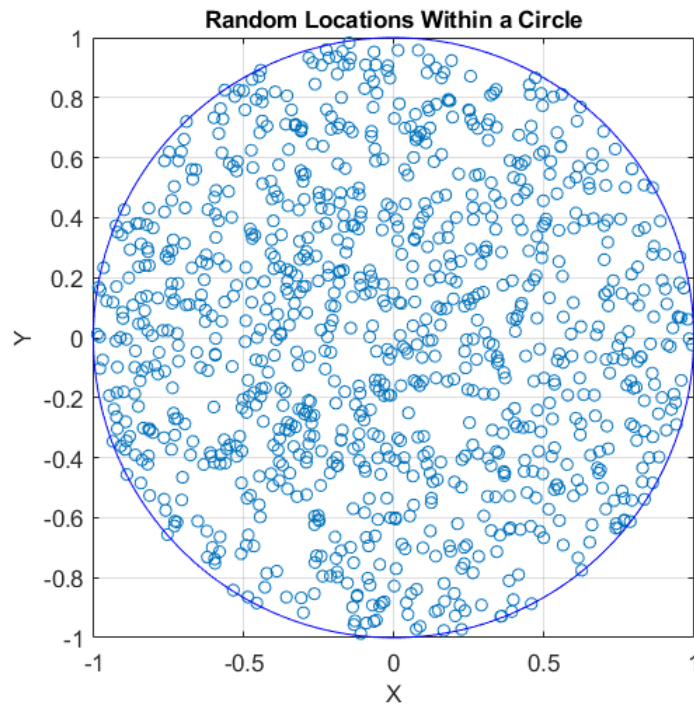
tel que  $\gamma_c = \gamma_b \frac{R}{W}$  est le SINR cible au niveau de la puce,  $W = 3,84$  Mchip/s est le débit de la puce.  $\theta = 0,4$  est un facteur représentant la perte d'orthogonalité entre les codes CDMA et  $N = \sigma^2$  est le bruit.

La densité spectrale de la puissance de bruit est de -174 dBm/Hz et la bande passante de la porteuse est de 5 MHz

## 4.1 Performance en l'absence d'interférences inter-cellule

### 4.1.1 Génération des utilisateurs

<on génère 1000 usagers dans un cercle de rayon 1km comme vu précédemment.



**FIGURE 6** – Random location within a circle

### 4.1.2 Répartition des conditions radio et consommation de puissance

On calcule la fonction de distribution cumulative (CDF) empirique de la consommation d'énergie au-dessus de la cellule.

```

%% Generating a channel coefficient g
beta=3.6;
alpha=10^13.6;
gain_antenna=2;%db
5 P=10;%watt
  L=10^0.7;%db

```



```
R=128*1000;%bit/s
G=2;
gamma_b=10^(2.8/10);%db
10 W=3.84*10^6;%chip/s
    theta=0.4;
    bandwidth=5*10^6;%HZ
    spectral_density_noise=-174;%dbm/hz

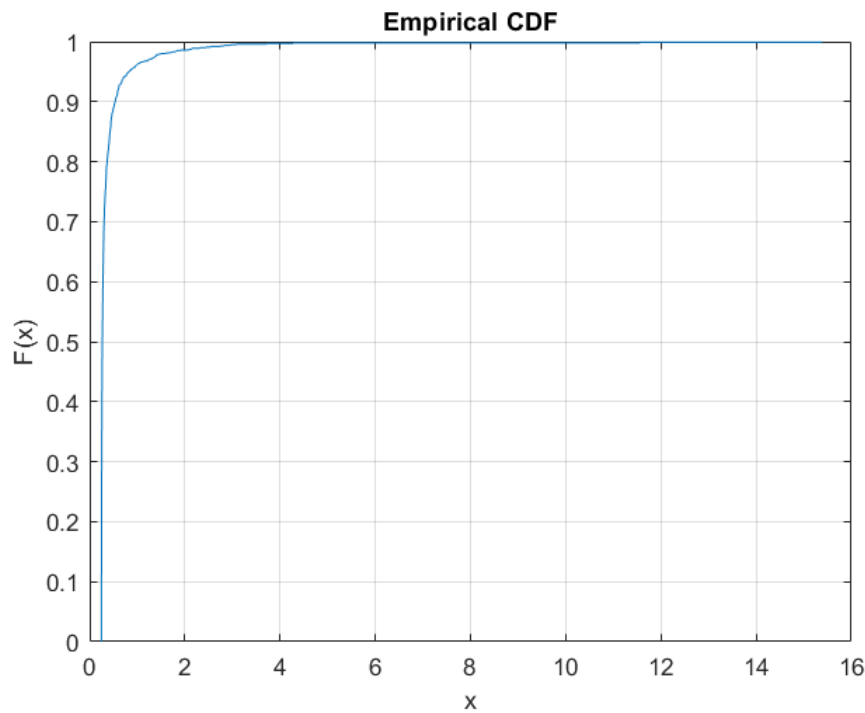
15
    N=10^(spectral_density_noise/10)*bandwidth/10^(3);

    s=10.^(normrnd(0,3.5,1,K)./10);
    l=alpha.*(rho.^beta);
20 h=exprnd(2,1,K);

    g=(G.*h.*s)./(l.*L);

    gamma_c=gamma_b*R/W;
25 transmitted_power=((theta*gamma_c)/(1+theta*gamma_c))*P + ...
    (((gamma_c*N)/(1+theta*gamma_c))./g);

    cdfplot(transmitted_power)
    percentile_90=prctile(transmitted_power,90);
    median=prctile(transmitted_power,50);
```

**FIGURE 7** – Empirical CDF

On déduit la médiane et la 90-percentile de la cosommation de puissance.

**La médiane :**  $\bar{P} = 0.2709$

**Le percentile à 90% :**  $\hat{P} = 0.5002$

#### 4.1.3 Evaluation de la performance en utilisant la méthode d'Erlang

le nombre des circuits est modélisé par :

$$M = \left\lceil \frac{P_{\max}}{\bar{P}} \right\rceil$$

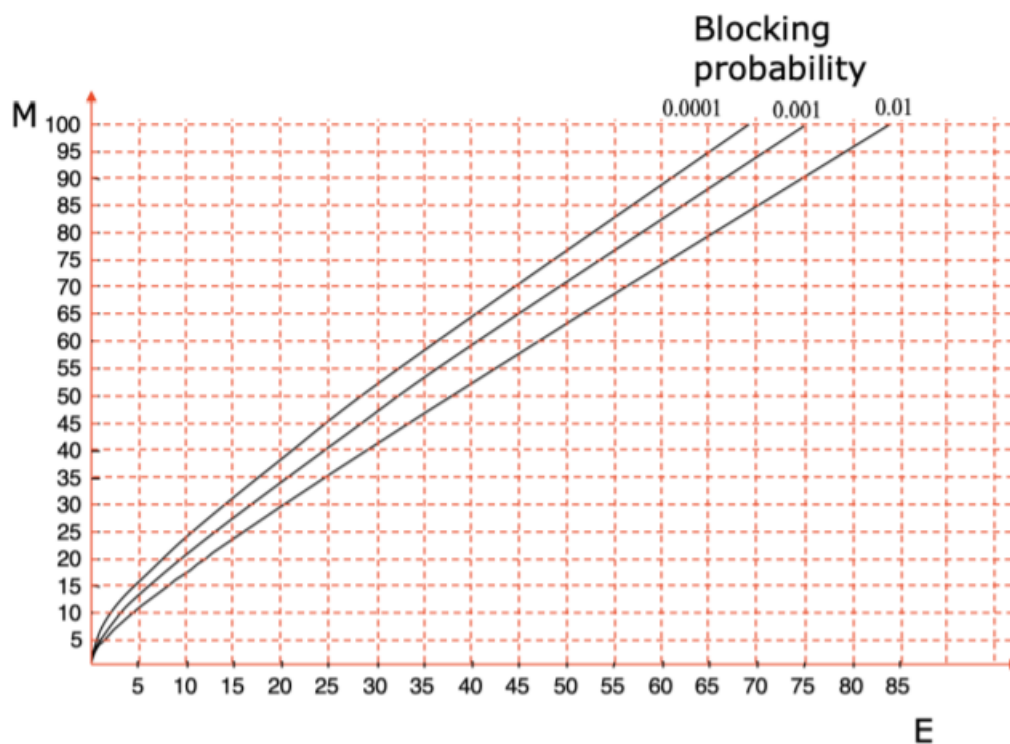
En première approche, on considère que les utilisateurs consomment une la puissance médiane. Une seconde approche, plus pessimiste, est de considérer que les utilisateurs consomment tous une puissance égale au percentile à 90%.

```

%%Performance evaluation using Erlang method
%1.3.1
M=floor(P/median);
%1.3.2
5 M_90=floor(P/percentile_90);

```

En utilisant l'abaque d'Erlang ,



**FIGURE 8** – Abaque d'Erlang

On trouve :

**cas optimiste : maximal-erlang-capacity = 25**

**cas pessimiste : maximal-erlang-capacity-90 = 13**

## 4.2 Prise en compte des interférences inter-cellule

Lorsque les interférences intercellulaires sont prises en compte, la puissance à transmettre pour l'utilisateur  $i$  est calculé comme suit :

$$P_i = \frac{\theta\gamma_c}{1 + \theta\gamma_c} P_{max} + \frac{\gamma_c}{1 + \theta\gamma_c} \frac{N}{g_i} + \frac{\gamma_c}{1 + \theta\gamma_c} \rho P_{max} F_i$$

Où  $\rho$  est la charge moyenne des cellules interférentes.  $F_i$  est le facteur d'interférence, calculé comme la somme des rapports des gains de canal entre la cellule de desserte et les cellules interférentes. le facteur d'interférence dépend de la distance relative à la base station. Pour une cellule de rayon  $R_c$  et un utilisateur situé à distance  $d$ , le facteur  $F$  est approché par :

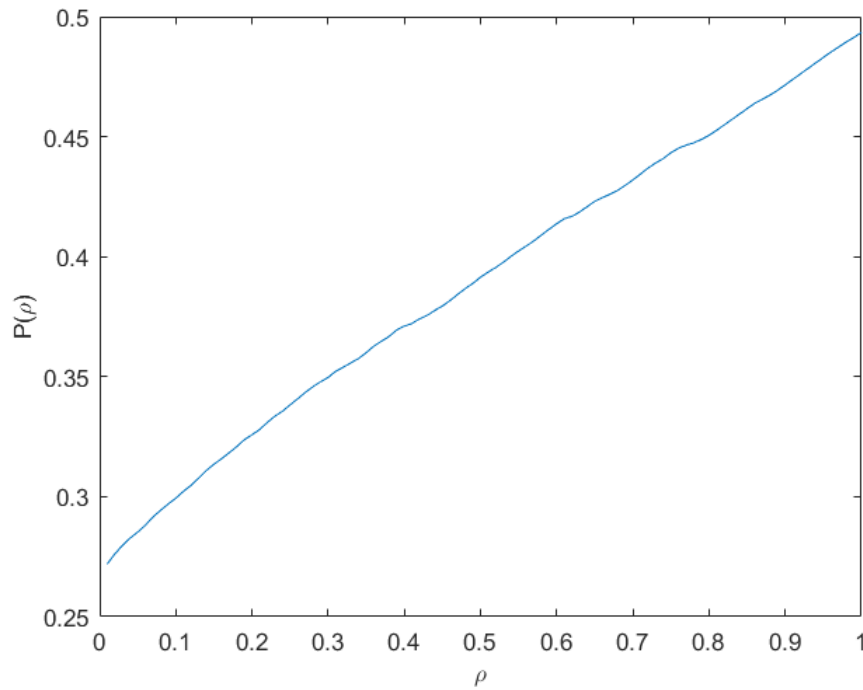
$$F(d) = \frac{\pi}{\sqrt{3}(\beta - 2)} \left(\frac{d}{R_c}\right)^\beta \left(2 - \frac{d}{R_c}\right)^{(2-\beta)}$$

### 4.2.1 CDF de la consommation de puissance sur la cellule, intégrant les interférences inter-cellules

On calcule la puissance médiane de la consommation d'énergie sur la cellule, en intégrant les interférences inter-cellules, pour une charge allant de 0 à 1.

```
%% Accounting for inter-cell interference
Rc=1;
F=pi/(sqrt(3)*(beta-2))*((rho/Rc).^beta).*((2-rho/Rc).^(2-beta));
m=100;
5 interference_ro=zeros(1,m);

for i=1:m
    interference_power=transmitted_power+(gamma_c/(1+theta*gamma_c))*((i/m)*P.*F;
    interference_ro(i)=prctile(interference_power,50);
10 end
plot([1:m]/m,interference_ro)
xlabel('\rho', 'FontSize', fontSize);
ylabel('P(\rho)', 'FontSize', fontSize);
```



**FIGURE 9** – median power consumption

On remarque que la consommation médiane est presque linéaire par rapport à la charge  $\rho$ . Ceci pourrait être déduit en observant que  $P_i$  dépend linéairement de la charge  $\rho$ .