Projet Data Science Chaîne de Markov : Tennis match

Projet réalisé par Nicolas Joué et Melvyn Provenzano

Problématique : Est-ce qu'un jeu au tennis amplifie ou diminue la différence de niveau entre les deux joueurs ?

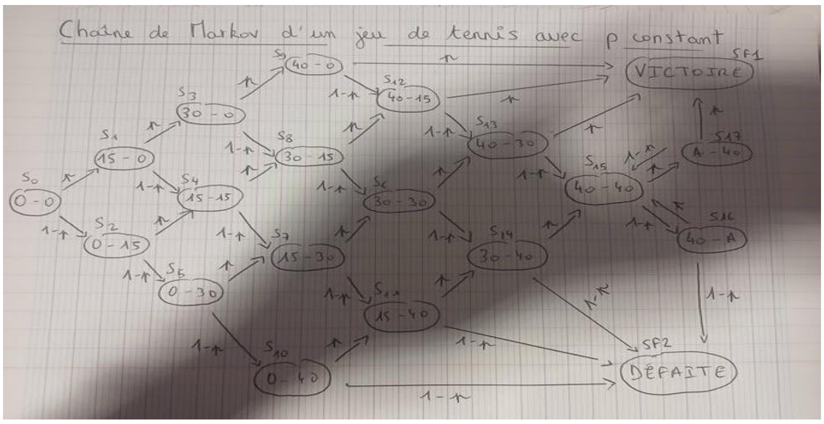
I – Réalisation de la chaîne de Markov du modèle

1. Réalisation de la chaîne de Markov et du modèle pour un jeu

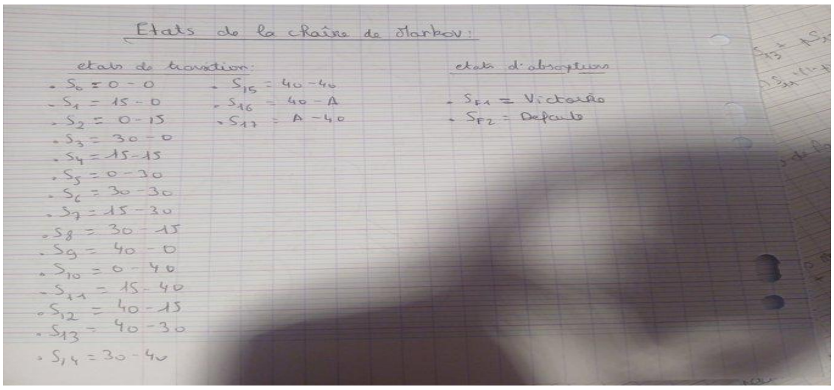
La première étape dans l'étude d'une chaîne de Markov consiste à poser le problème et à définir toutes les étapes de la chaîne. Pour construire le modèle, il est nécessaire de créer la matrice de transition T, où chaque élément T_{ij} représente la probabilité de transition de l'état i vers l'état j, en supposant une probabilité constante p de gagner un point. Cela nécessite au préalable de définir tous les états, qu'ils soient de transition ou d'absorption.

Nous avons suivi ces différentes étapes :

- Chaîne de Markov d'un jeu de tennis (en prenant en compte que p représente la probabilité de gagner un point pour le joueur qui sert) :



- Etats de la chaîne de Markov :



Etats de la chaîne de Markov :

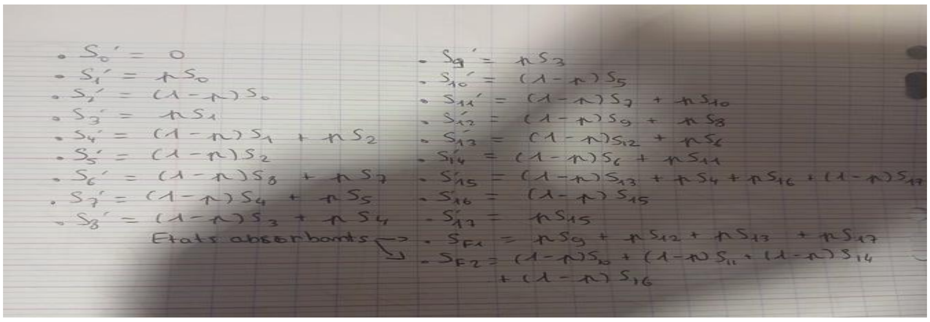
etats de transition :

- $S_0 = 0 - 0$
- $S_1 = 15 - 0$
- $S_2 = 0 - 15$
- $S_3 = 30 - 0$
- $S_4 = 15 - 15$
- $S_5 = 0 - 30$
- $S_6 = 30 - 30$
- $S_7 = 15 - 30$
- $S_8 = 30 - 15$
- $S_9 = 40 - 0$
- $S_{10} = 0 - 40$
- $S_{11} = 15 - 40$
- $S_{12} = 40 - 15$
- $S_{13} = 40 - 30$
- $S_{14} = 30 - 40$
- $S_{15} = 40 - 40$
- $S_{16} = 40 - A$
- $S_{17} = A - 40$

etats d'absorption

- S_{F1} = Victoire
- S_{F2} = Defaite

- Équations de récurrence des probabilités de transition :

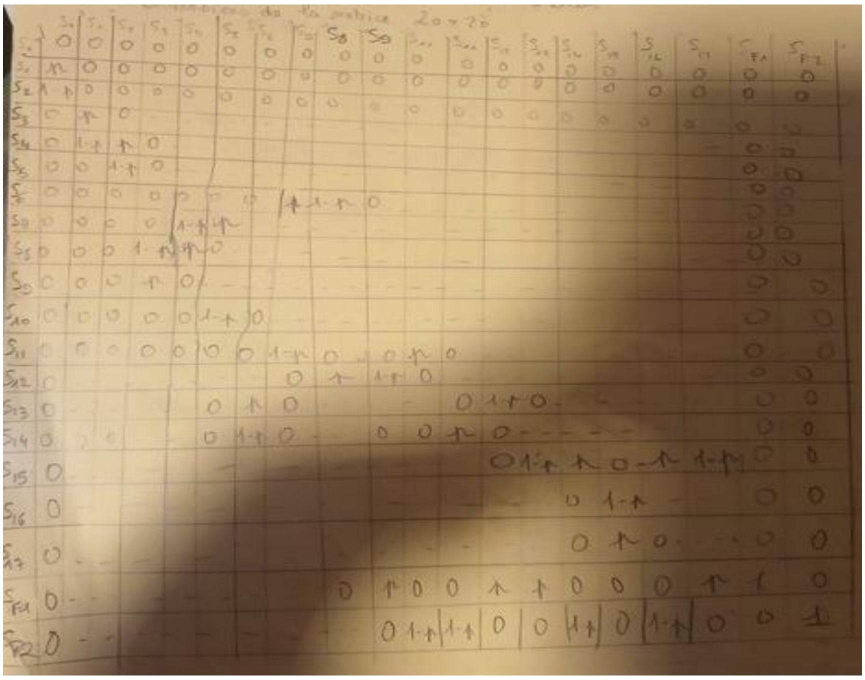


- $S_0' = 0$
- $S_1' = p S_0$
- $S_2' = (1-p) S_0$
- $S_3' = p S_1$
- $S_4' = (1-p) S_1 + p S_2$
- $S_5' = (1-p) S_2$
- $S_6' = (1-p) S_8 + p S_7$
- $S_7' = (1-p) S_4 + p S_5$
- $S_8' = (1-p) S_3 + p S_4$
- $S_9' = p S_3$
- $S_{10}' = (1-p) S_5$
- $S_{11}' = (1-p) S_7 + p S_{10}$
- $S_{12}' = (1-p) S_9 + p S_8$
- $S_{13}' = (1-p) S_{12} + p S_6$
- $S_{14}' = (1-p) S_6 + p S_{11}$
- $S_{15}' = (1-p) S_{13} + p S_4 + p S_{16} + (1-p) S_{17}$
- $S_{16}' = (1-p) S_{15}$
- $S_{17}' = p S_{15}$

Etats absorbants

- $S_{F1} = p S_9 + p S_{12} + p S_{13} + p S_{17}$
- $S_{F2} = (1-p) S_{10} + (1-p) S_{11} + (1-p) S_{14} + (1-p) S_{16}$

- Cela donne alors la matrice de transition suivante, de dimension 20 * 20 (car le nombre d'états est de 20 : 18 états de transition + 2 états d'absorption) :



(avec les pointillés qui correspondent à la continuité des 0)

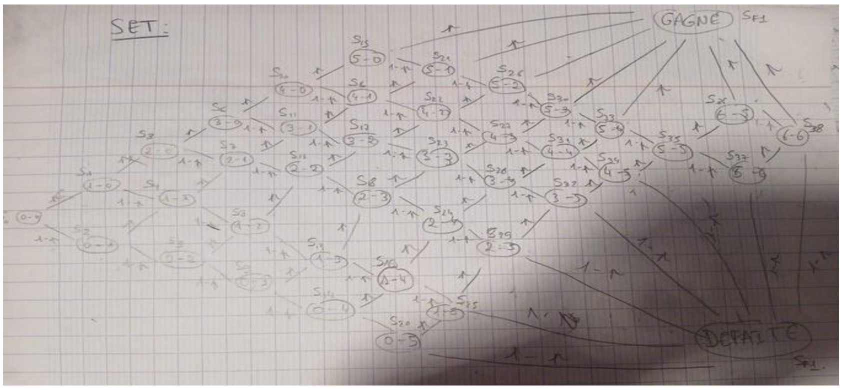
Une fois la matrice construite, il est essentiel de vérifier ses propriétés, notamment que la somme des probabilités de chaque colonne (représentant un état) est égale à 1.

Dans cette première analyse, nous testons le modèle avec une hypothèse simplifiée : une probabilité constante *p* de gagner un point. Cette simplification permet d'explorer les mécanismes fondamentaux par lesquels un jeu de tennis amplifie ou atténue les différences de niveau entre joueurs, sans complexifier le modèle par des variations de *p* selon les états du jeu. En établissant ces tendances générales, cette approche fournit une base pour intégrer, dans les sections suivantes, des éléments plus réalistes reflétant les variations contextuelles et stratégiques propres à un jeu.

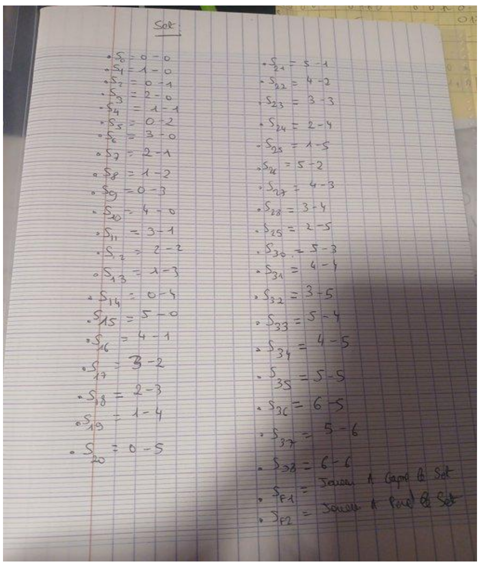
2. Réalisation de la chaîne de Markov et du modèle pour un set

Pour analyser un set, nous avons également commencer par définir les états de la chaîne de Markov et établir sa structure, avec cette fois-ci *p* qui représente la probabilité de gagner un jeu :

- Chaîne de Markov :



- Etats de transitions :


Set :

- $S_0 = 0-0$
- $S_1 = 1-0$
- $S_2 = 0-1$
- $S_3 = 2-0$
- $S_4 = 1-1$
- $S_5 = 0-2$
- $S_6 = 3-0$
- $S_7 = 2-1$
- $S_8 = 1-2$
- $S_9 = 0-3$
- $S_{10} = 4-0$
- $S_{11} = 3-1$
- $S_{12} = 2-2$
- $S_{13} = 1-3$
- $S_{14} = 0-4$
- $S_{15} = 5-0$
- $S_{16} = 4-1$
- $S_{17} = 3-2$
- $S_{18} = 2-3$
- $S_{19} = 1-4$
- $S_{20} = 0-5$
- $S_{21} = 5-1$
- $S_{22} = 4-2$
- $S_{23} = 3-3$
- $S_{24} = 2-4$
- $S_{25} = 1-5$
- $S_{26} = 5-2$
- $S_{27} = 4-3$
- $S_{28} = 3-4$
- $S_{29} = 2-5$
- $S_{30} = 5-3$
- $S_{31} = 4-4$
- $S_{32} = 3-5$
- $S_{33} = 5-4$
- $S_{34} = 4-5$
- $S_{35} = 5-5$
- $S_{36} = 6-5$
- $S_{37} = 5-6$
- $S_{38} = 6-6$
- S_{F1} = Joueur A Gagné le Set
- S_{F2} = Joueur A Perd le Set

Au vu du grand nombre d'états (41 : 39 états de transition et 2 états d'absorption), nous avons décidé cette fois-ci de réaliser la matrice de transition uniquement et directement sur Matlab :

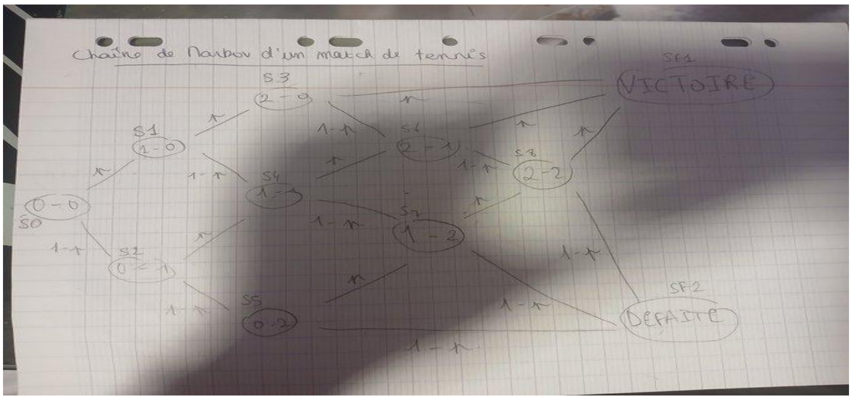
```matlab
% Nombre d'états : 39 transitions + 2 absorbants pour le set
nset = 41;
prob_victoire_set = zeros(size(p_values));

scores_set = [
    0, 0;  % S0
    1, 0;  % S1
    0, 1;  % S2
    2, 0;  % S3
    1, 1;  % S4
    0, 2;  % S5
    3, 0;  % S6
    2, 1;  % S7
    1, 2;  % S8
    0, 3;  % S9
    4, 0;  % S10
    3, 1;  % S11
    2, 2;  % S12
    1, 3;  % S13
    0, 4;  % S14
    5, 0;  % S15
    4, 1;  % S16
    3, 2;  % S17
    2, 3;  % S18
    1, 4;  % S19
    0, 5;  % S20
    5, 1;  % S21
    4, 2;  % S22
    3, 3;  % S23
    2, 4;  % S24
    1, 5;  % S25
    5, 2;  % S26
    4, 3;  % S27
    3, 4;  % S28
    2, 5;  % S29
    5, 3;  % S30
    4, 4;  % S31
    3, 5;  % S32
    5, 4;  % S33
    4, 5;  % S34
    5, 5;  % S35
    6, 5;  % S36
    5, 6;  % S37
    6, 6;  % S38
];

% Trouver l'indice de l'état correspondant au score
find_state_set = @(a, b) find(scores_set(:,1) == a & scores_set(:,2) == b);
SF1_set = 40; % Victoire de A
SF2_set = 41; % Défaite de A

for i = 1:num_p
    p = p_values(i);
    q = 1 - p;
```

```matlab
    % Matrice de Transition du set
    T_set = zeros(nset, nset);

    for j = 1:nset - 2 % On exclut les états absorbants
        current_score = scores_set(j, :);
        a = current_score(1);
        b = current_score(2);

        new_a = a + 1;
        new_b = b;
        if (new_a >= 6 && new_a - new_b >= 2) || (new_a == 7 && new_b == 5)
            T_set(SF1_set, j) = p;
        else
            next_state = find_state_set(new_a, new_b);
            if ~isempty(next_state)
                T_set(next_state, j) = p;
            else
                T_set(SF1_set, j) = p;
            end
        end

        new_a = a;
        new_b = b + 1;
        if (new_b >= 6 && new_b - new_a >= 2) || (new_b == 7 && new_a == 5)
            T_set(SF2_set, j) = q;
        else
            next_state = find_state_set(new_a, new_b);
            if ~isempty(next_state)
                T_set(next_state, j) = q;
            else
                T_set(SF2_set, j) = q;
            end
        end
    end

    state_6_6 = find_state_set(6,6);
    if ~isempty(state_6_6)
        T_set(SF1_set, state_6_6) = p;
        T_set(SF2_set, state_6_6) = q;
    end

    for col = 1:nset
        column_sum = sum(T_set(:, col));
        if column_sum < 1
            T_set(col, col) = 1 - column_sum;
        end
    end

    etat_initial = zeros(nset, 1);
    etat_initial(1) = 1;

    T_long_term_set = T_set^100;
    proba_finale_set = T_long_term_set * etat_initial;

    prob_victoire_set(i) = proba_finale_set(SF1_set);
end
```

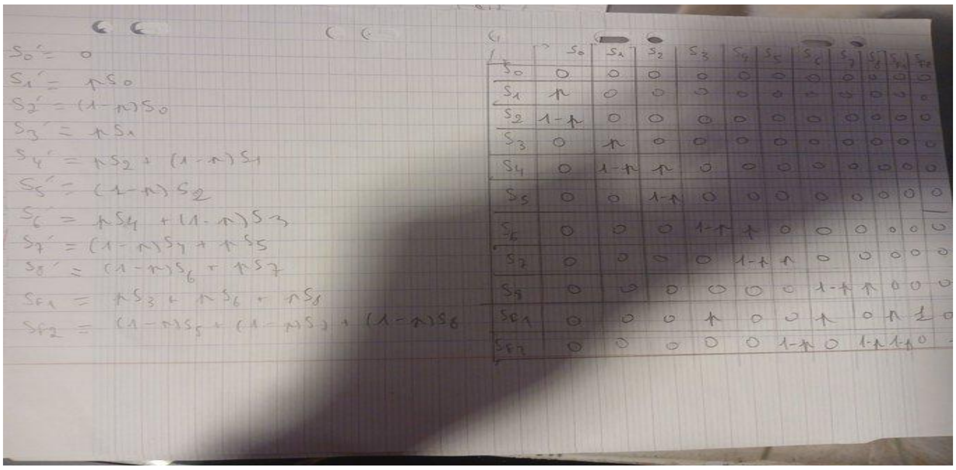
3. Réalisation de la chaîne de Markov et du modèle pour un match

Avant toute chose, nous avons considéré ici que notre match est en 3 gagnant. Ensuite nous avons fait la même chose que pour jeu avec cette fois-ci *p* qui est la probabilité de gagner un set :

- Chaîne de Markov :



- Équations de récurrence des probabilités de transition et matrice de transition (de dimension 11 * 11 (car le nombre d'états est de 11 : 9 états de transition + 2 états d'absorption)) :



II- Implémentation des modèles dans MATLAB

1. Analyse de la probabilité de victoire

a. Jeu

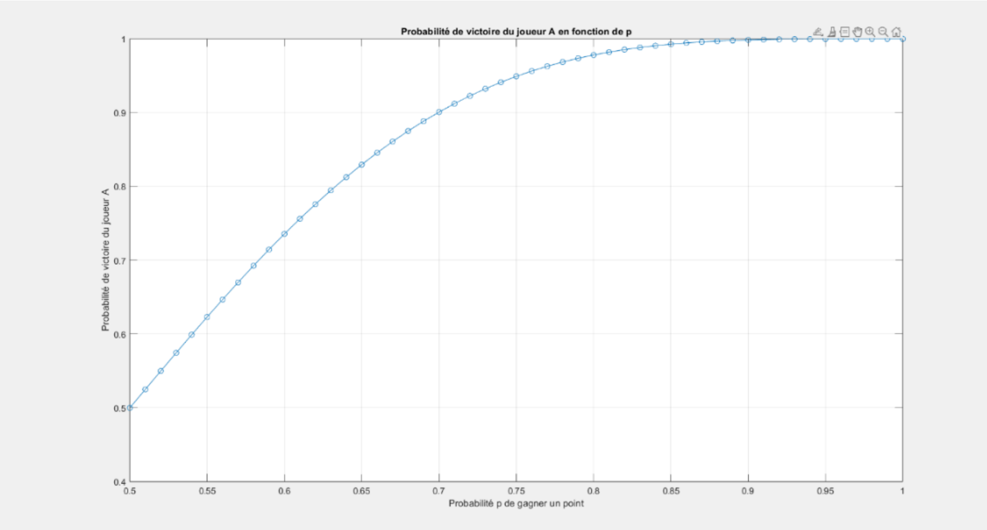
Dans un premier temps, nous avons calculé les probabilités de gagner le jeu pour différentes valeurs de *p*. L'objectif est de comprendre comment la probabilité de gagner un jeu varie en fonction de *p*. Ainsi, on pourrait faire un lien entre la différence de niveau entre les deux joueurs et la probabilité de gagner le jeu. Pour ce faire, nous avons utilisé la matrice de transition dans MATLAB pour calculer la probabilité que le joueur A (on part du principe que *p* représente la probabilité du joueur A de gagner un point, lorsque A est le joueur qui sert durant le jeu) gagne le jeu pour des valeurs de p variant de 0.5 à 1 (avec incrémentation de 0.01). Ainsi, pour chaque p on calcule la probabilité de victoire du joueur A. On l'a fait avec le code suivant :

```
n = 20;  % Nombre d'états
p_values = 0.5:0.01:1.0;  % Valeurs que va prendre p (p va de 0.5 à 1 par incrémentation de 0.01)
prob_victoire_A = zeros(size(p_values));  % Tableau pour stocker les probabilités de victoire de A

% Vecteur état initial pour le score 0-0 (S0)
v0 = zeros(n, 1);
v0(1) = 1;

% Pour chaque valeur de p
for i = 1:length(p_values)
    p = p_values(i);
    % Matrice de transition T
    T = [
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S0
        p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S1
        1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S2
        0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S3
        0,   1-p, p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S4
        0,   0,   1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S5
        0,   0,   0,   0,   0,   0,   0,   p,   1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S6
        0,   0,   0,   0,   1-p, p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S7
        0,   0,   0,  1-p,  p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S8
        0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S9
        0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S10
        0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S11
        0,   0,   0,   0,   0,   0,   0,   0,   p, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S12
        0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0;  % S13
        0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0;  % S14
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   p,   0,   p, 1-p,   0,   0;  % S15
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0;  % S16
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0;  % S17
        0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   p,   p,   0,   0,   0,   p,   1,   0;  %SF1 (victoire de A)
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p, 1-p,   0,   0, 1-p,   0, 1-p,   0,   0,   1;  %SF2
    ];

 % Calcul des probabilités finales après un grand nombre de transitions
    T_long_term = T^100;
    proba_finale = T_long_term * v0;

    % Stocke la probabilité de gagner un jeu pour A
    prob_victoire_A(i) = proba_finale(19);  % Probabilité d'atteindre l'état SF1 et donc de gagner le jeu pour le joueur A
end
```

Pour le calcul des probabilités d'atteindre les états absorbants, nous avons défini un vecteur d'état initial *v0*, de dimension 20×1, représentant le début du jeu à 0–0, avec une probabilité de 1 pour cet état et 0 pour les autres. La matrice de transition *T* a été élevée à une puissance élevée T^{100} pour simuler un grand nombre de transitions et garantir une convergence suffisante des probabilités vers les états absorbants. La multiplication $v0 \times T^{100}$ fournit alors les probabilités finales dans chaque état absorbant, permettant de calculer directement les chances du joueur A de remporter le jeu.

Dans un deuxième temps, nous avons visualisé ces résultats à l'aide de graphique. En effet, le but était d'observer la relation entre *p* et la probabilité de gagner le jeu. Pour ce faire, nous avons tracer un graphique avec *p* en abscisse et la probabilité de gagner le jeu pour A en ordonnée.

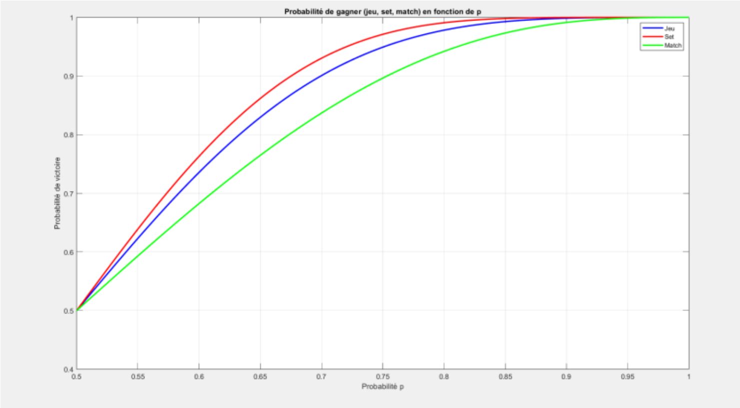
```
% Graphique traçant la probabilité de gagner un jeu en fonction de p
figure;
plot(p_values, prob_victoire_A, '-o');
xlabel('Probabilité p de gagner un point');
ylabel('Probabilité de victoire du joueur A');
title('Probabilité de victoire du joueur A en fonction de p');
grid on;
```



On constate que la probabilité de gagner un jeu augmente logiquement avec *p* : plus un joueur a de chances de remporter un point, plus la probabilité de remporter l'ensemble du jeu (qui comprend plusieurs points) s'accroît. Nous pouvons étendre ce raisonnement au set et au match en reproduisant la même analyse.

b. Comparaison avec un set et un match

En appliquant la même méthodologie, on obtient en plus les courbes de la probabilité de gagner un set ou un match en fonction de *p* (*p* étant respectivement la probabilité de gagner un point, un jeu ou un set, selon le niveau considéré).



Comme pour le jeu, ces probabilités augmentent avec *p*. Toutefois, cette simple constatation ne suffit pas à conclure sur l'amplification ou la réduction des écarts de niveau entre les joueurs. En effet, les formes des courbes diffèrent : par exemple, la courbe du match croît plus lentement, alors que celle du set est plus raide. Cela suggère des mécanismes d'amplification différents selon le niveau (jeu, set ou match).

Pour approfondir cette analyse, il est nécessaire d'examiner la pente de ces courbes, c'est-à-dire la dérivée de la probabilité de victoire par rapport à *p*. Cette démarche, en quantifiant l'effet amplificateur, permettra de mieux comprendre dans quelle zone de *p* l'amplification des petites différences de compétence est la plus marquée.

2. Effet amplificateur

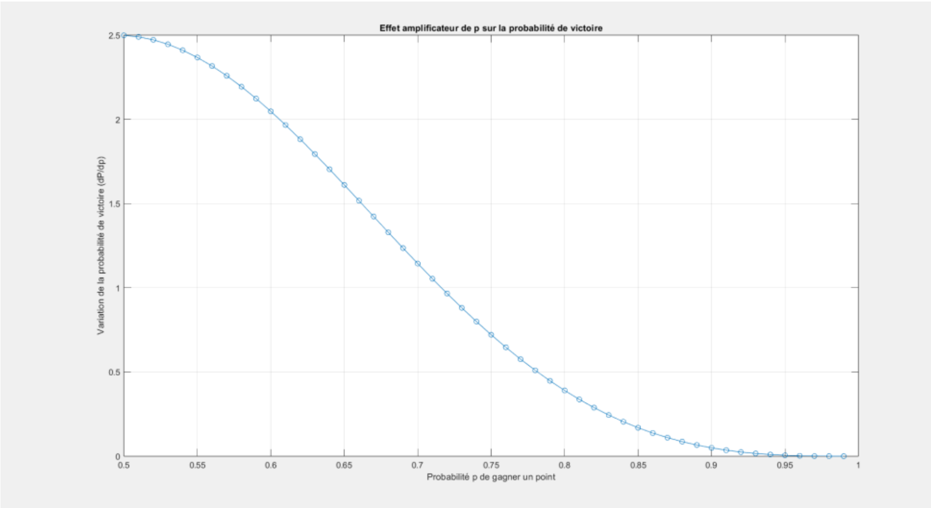
a. Jeu

Ainsi pour mesurer et analyser l'effet amplificateur nous avons calculés la dérivée de la courbe de probabilité de victoire par rapport à *p* et tracer son graphe. Cela permet de voir l'influence de *p* et aide à visualiser la zone où l'effet amplificateur est plus fort.

Voici notre démarche :

```
% Calcul de la dérivée numérique
dp = diff(prob_victoire_A) ./ diff(p_values);

% Graphique de la dérivée
figure;
plot(p_values(1:end-1), dp, '-o');
xlabel('Probabilité p de gagner un point');
ylabel('Variation de la probabilité de victoire (dP/dp)');
title('Effet amplificateur de p sur la probabilité de victoire');
grid on;
```

Nous avons alors obtenu :



En observant ce graphique nous pouvons voir que l'effet amplificateur de *p* diminue en fonction de la probabilité p de gagner un point. En effet, la courbe est élevée autour de *p* = 0.5 (ce qui donne environ 2.5), et diminue de façon monotone jusqu'à devenir quasiment nulle autour de p = 0.95.

Interprétation :

- **Effet amplificateur maximal autour de *p* = 0.5** : La valeur élevée de la dérivée à ce point montre que lorsque les joueurs sont proches en compétence, même une petite différence de *p* a un impact important sur la probabilité de victoire. Cela démontre qu'un jeu en tennis amplifie fortement les différences de compétence lorsque celles-ci sont faibles. Par exemple, lorsque le joueur A a une probabilité de 0,6 de gagner un point, on peut observer que la dérivée de la probabilité de victoire vaut environ 2. Cela signifie que si *p* augmente de 0,01 (donc, passant de 0,6 à 0,61), la probabilité de gagner le jeu augmente de 2×0,01=0,02, soit 2 %. En d'autres termes, même une petite amélioration des compétences (ou un léger avantage) peut avoir un effet amplifié sur les chances de gagner un jeu.
- **Effet amplificateur diminué pour des p élevés** : A mesure que *p* augmente vers 1, la courbe descend vers 0, indiquant que l'effet amplificateur devient presque inexistant. Cela signifie que pour un joueur qui a déjà un avantage marqué (ex : *p* proche de 1), la probabilité de gagner le jeu est presque assurée, et un jeu n'amplifie plus ou moins la différence de compétence. En effet, le joueur a déjà une très forte probabilité de gagner le jeu.

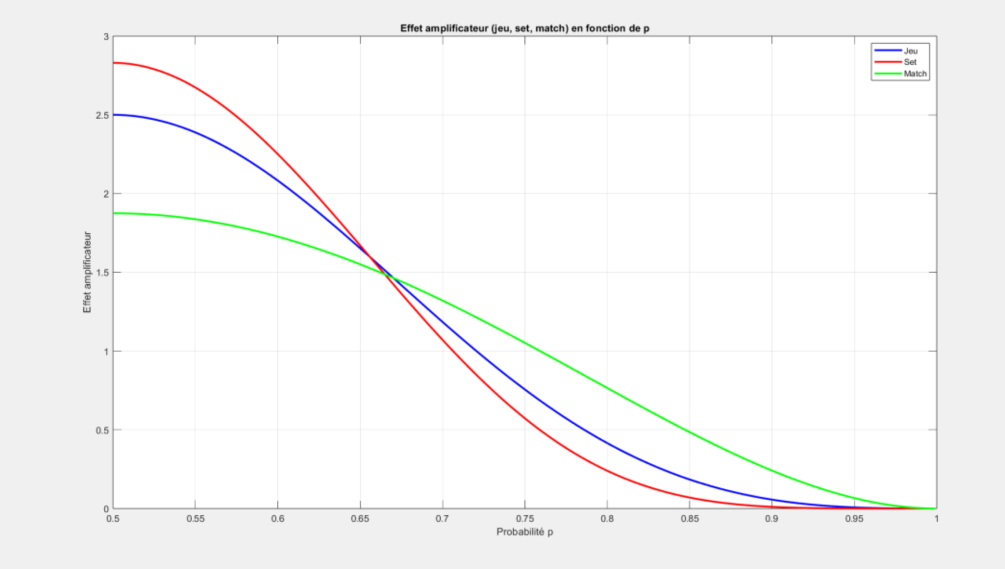
Pour conclure, cette analyse démontre clairement que le système de score d'un jeu de tennis amplifie significativement les petites différences de compétence, et dans une moindre mesure, tout écart de niveau entre les joueurs.

Il est alors intéressant de se demander si ce phénomène s'observe également pour un set et un match.

b. Comparaison avec un set et un match

En appliquant la même méthode que pour le jeu, nous avons également calculé l'effet amplificateur pour le set et le match en fonction de *p*.

Nous avons ainsi obtenu le graphique suivant :



Nous observons que, tout comme pour un jeu, l'effet d'amplification est maximal autour de p=0.5 et diminue progressivement pour se rapprocher de zéro à mesure que p augmente. Ainsi, les mêmes conclusions générales peuvent être tirées. Cependant, en raison des différences de pentes entre les courbes, il est pertinent de nuancer cette conclusion et d'explorer davantage ces phénomènes pour mieux les comprendre.

- **Amplification plus marquée au niveau du set** : Nous pouvons observer que la pente pour un set est plus raide. En effet, autour de p=0.5, le set, avec ses 41 états, amplifie davantage les légers écarts de compétence que le jeu (20 états) et le match (11 états). Comme un set se compose de nombreux jeux, chaque différence de compétence s'accumule, conduisant à une probabilité de victoire encore plus amplifiée qu'au niveau du jeu ou du match.
- **Changement de tendance au-delà de $p \approx 0.65$** : Pour des p plus élevés, chaque petit avantage impacte davantage la probabilité de gagner un jeu, et donc le jeu finit par dépasser le set puis le match en termes d'amplification. Bien que le set reste globalement plus amplificateur, à très forte probabilité, le jeu puis le match passent au-dessus, même si la probabilité de gagner un set par rapport à p reste tout de même pratiquement tout le long supérieure à celles du match et du set).
- **Effet amplificateur plus faible pour le match ($0.5 < p < 0.66$)** : Entre p=0.5 et p=0.66, le match, avec ses 11 états, offre moins d'opportunités de cumuler les écarts de compétence (car en effet un match comporte moins de sets qu'un set

comporte de jeu et moins de sets qu'un jeu comporte de points). Ainsi, dans cette plage et avec ce modèle simplifié, le match amplifie moins les différences que le set ou même le jeu.

- **Après $p \approx 0.66$, avantage du match :** Cependant, à partir de $p \approx 0.66$, la courbe de l'effet amplificateur pour le match dépasse celles du set et du jeu. Ce phénomène s'explique par le fait que, pour des valeurs de p élevées, un joueur devient rapidement favori pour remporter un jeu ou un set, car chacun de ces éléments amplifie directement les petites différences. Le joueur devient alors favori plus rapidement dans un jeu ou un set que dans un match, où la montée de la probabilité de victoire est plus progressive. Ainsi, pour le match, la pente de l'effet amplificateur reste plus douce, mais les petites variations de p à partir de ce point commencent à impacter plus directement la probabilité de victoire, car un avantage aux sets se traduit plus nettement par une victoire de match.

Pour conclure, cette analyse montre de façon convaincante que le système de score en tennis amplifie les petites différences de compétence, et en réalité amplifie la différence de niveau entre 2 joueurs. Mais il serait intéressant de se demander comment un jeu de tennis amplifie cette différence.

III – Comment le système de points au tennis amplifie-t-il cette différence de compétence ?

1. Etude de la durée de vie

Pour approfondir notre analyse de l'effet amplificateur, il serait encore plus intéressant d'examiner l'évolution de la durée moyenne d'un jeu, d'un set et d'un match en fonction de *p*. En effet, en prenant l'exemple du jeu, le but est d'examiner si la durée moyenne d'un jeu (nombre moyen de points joués) varie en fonction de *p*, et voir comment cela pourrait influencer l'amplification des différences de compétence. Pour ce faire, nous avons utilisé la matrice de transition afin de calculer l'espérance de la durée d'un jeu pour différentes valeurs de *p*. En effet, en calculant le nombre moyen de points pour chaque *p*, nous pouvons observer comment la durée moyenne du jeu varie avec *p*. Pour calculer cette durée moyenne du jeu, nous avons utilisé la matrice fondamentale $N = (I - Q)^{-1}$ pour calculer la moyenne du nombre de transitions (nombre de points) avant d'atteindre la fin du jeu (un état absorbant), avec *I* la matrice identité et *Q* qui est la sous-matrice des états non absorbants. Ensuite, en multipliant *N* par un vecteur de 1, on obtient la moyenne du nombre de pas à partir de chaque état non absorbant jusqu'à l'absorption. Ce qui nous donnera le « potentiel » d'amplification des différences de compétence.

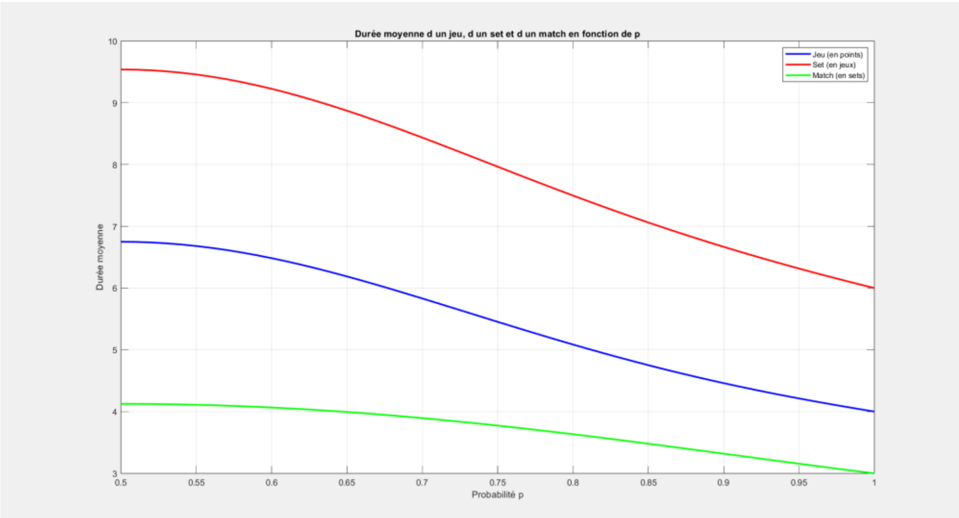
Pour ce faire nous avons fait ceci :

```
p_values = 0.5:0.01:1.0;
duree_moyenne_jeu = zeros(size(p_values));  % Durée moyenne du jeu (en points)
```

```
%% Calcul de la durée moyenne du jeu
for i = 1:length(p_values)
    p = p_values(i);

    % Matrice de transition T pour le jeu
    T_jeu = [
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S0
        p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S1
        1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S2
        0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S3
        0, 1-p,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S4
        0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S5
        0,   0,   0,   0,   0,   0,   0,   p, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S6
        0,   0,   0,   0, 1-p,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S7
        0,   0,   0, 1-p,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S8
        0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S9
        0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S10
        0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S11
        0,   0,   0,   0,   0,   0,   0,   0,   p, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S12
        0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0;  % S13
        0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0;  % S14
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   p,   0,   p, 1-p,   0,   0;  % S15
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0;  % S16
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0;  % S17
        0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   p,   p,   0,   0,   0,   p,   1,   0;  % SF1 (Victoire A)
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p, 1-p,   0,   0, 1-p,   0, 1-p,   0,   0,   1;  % SF2
    ];

    T_jeu = T_jeu';

    % États non absorbants (les 18 premiers états)
    Q_jeu = T_jeu(1:18, 1:18);

    % Matrice fondamentale N pour le jeu
    N_jeu = inv(eye(size(Q_jeu)) - Q_jeu);

    % Durée moyenne du jeu (nombre moyen de points)
    nomb_moy_points = N_jeu * ones(size(Q_jeu,1),1);
    duree_moyenne_jeu(i) = nomb_moy_points(1);
end
```

```
%% Tracé des durées moyennes
figure;
plot(p_values, duree_moyenne_jeu, 'b-', 'LineWidth', 2);
hold on;
plot(p_values, duree_moyenne_set, 'r-', 'LineWidth', 2);
plot(p_values, duree_moyenne_match, 'g-', 'LineWidth', 2);
xlabel('Probabilité p');
ylabel('Durée moyenne');
title('Durée moyenne d un jeu, d un set et d un match en fonction de p');
legend('Jeu (en points)', 'Set (en jeux)', 'Match (en sets)');
grid on;
hold off;
```

Pour comparer ces durées moyennes, on va comparer la durée de vie de chacune des 3 courbes, ce qui permettra d'avoir une vision globale, voici le graphique :



Nous pouvons observer que :

- Pour un jeu, par exemple, à *p*=0.5, la durée moyenne est d'environ 7,8 points.
- Pour un set, à *p*=0.5, la durée moyenne atteint 9,5 jeux. Le plus grand nombre d'états (41) prolonge le set, ce qui augmente les occasions d'amplifier les écarts de compétence. Un léger avantage se transforme ainsi plus facilement en probabilité accrue de victoire au fur et à mesure que les situations s'enchaînent.
- Pour un match, la durée moyenne à p=0.5 est d'environ 4,1 sets, ce qui est moins long que le « potentiel » d'un set ou d'un jeu (en considérant leur échelle respective). Le match, avec moins d'états globaux (11), offre moins d'opportunités d'accumuler les différences de compétence que le set, bien qu'il y ait tout de même une amplification.

Ces observations complètent celles faites sur l'effet amplificateur. L'amplification maximale autour de *p*=0.5 observée à tous les niveaux, est directement liée à la capacité du système à accumuler les différences en fonction de la « longueur » du système (un set étant plus long et complexe qu'un jeu, et un match finalement plus court en moyenne dans ce modèle simplifié).

De manière similaire aux trois systèmes de points (jeu, set et match), on observe également que plus le niveau est proche, plus le nombre d'états est grand. Alors qu'au contraire, plus la différence de niveaux entre les deux joueurs est grande, plus le nombre d'états diminue. Par exemple, un joueur nettement supérieur à son adversaire (il serait toutefois plus précis de dire *"un joueur qui a une probabilité beaucoup plus élevée de gagner un point face à son adversaire"*, car un joueur globalement très supérieur pourrait avoir une probabilité moindre dans des situations spécifiques, comme lorsqu'il ne sert pas) gagnera souvent rapidement un jeu (souvent avec un score de 40-0), sans laisser le temps à des scores serrés de s'installer.

Ainsi, le système de points au tennis tend à augmenter les situations où la probabilité p est mise à l'épreuve, particulièrement lorsque le match est serré. Cela favorise les joueurs ayant un léger avantage, renforçant l'idée que ce système amplifie efficacement les différences de compétence, comme observé précédemment.

Pour poursuivre cette réflexion, il est pertinent d'examiner un aspect clé du système de points au tennis : les situations d'égalité, telles que 40-40 dans un jeu, 5-5 dans un set ou 2-2 dans un match, et leur rôle dans l'amplification des différences de compétence.

2. Situations d'égalité

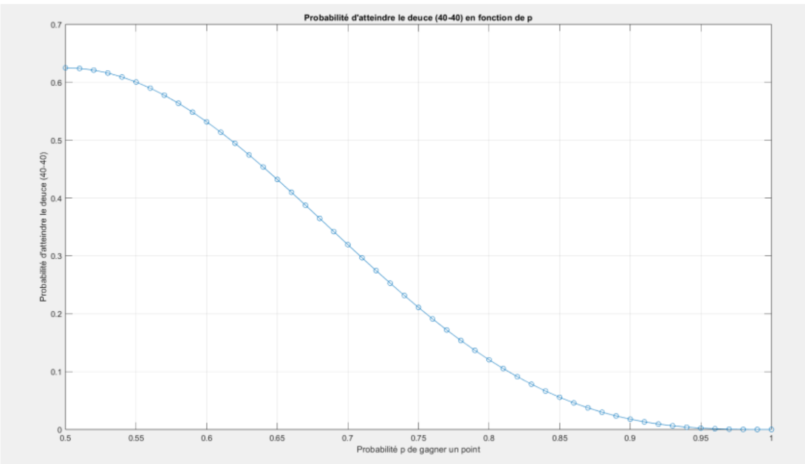
a. Jeu

Ensuite, il est intéressant d'examiner le pourcentage de chances d'atteindre des situations d'égalité en fonction de p, comme le score de 40-40 dans un jeu. Ce score est particulier, car lorsqu'il survient, il entraîne des points "avantage" où les joueurs doivent gagner deux points consécutifs pour conclure le jeu. Cette dynamique prolonge le jeu et peut donc influencer la durée du match.

Pour analyser cela, nous avons utilisé la matrice de transition pour simuler un jeu en partant du score initial (0-0). En effectuant plusieurs itérations, nous avons mesuré la probabilité d'atteindre 40-40 pour différentes valeurs de *p* :

```matlab
p_values = 0.5:0.01:1.0;
prob_deuce = zeros(size(p_values));

for i = 1:length(p_values)
    p = p_values(i);

    % Définir la matrice de transition T pour chaque valeur de p
    T = [
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S0
        p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S1
        1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S2
        0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S3
        0,   1-p, p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S4
        0,   0,   1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S5
        0,   0,   0,   0,   0,   0,   0,   p,   1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S6
        0,   0,   0,   0,   1-p, p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S7
        0,   0,   0,  1-p,  p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S8
        0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S9
        0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S10
        0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S11
        0,   0,   0,   0,   0,   0,   0,   0,   p, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S12
        0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0;  % S13
        0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0;  % S14
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   p,   0,   p, 1-p,   0,   0;  % S15
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0;  % S16
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0;  % S17
        0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   p,   p,   0,   0,   0,   p,   1,   0;  % SF1
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p, 1-p,   0,   0, 1-p,   0, 1-p,   0,   0,   1;  % SF2
    ];

    v = zeros(20, 1);
    v(1) = 1;

    % Calculer la probabilité d'atteindre le 40-40 en itérant T
    iterations = 100;
    proba_deuce_actuelle = 0;

    for k = 1:iterations
        v = T * v;  % Mettre à jour le vecteur état
        proba_deuce_actuelle = proba_deuce_actuelle + v(16);  % Somme des probabilités d'atteindre S15 (40-40)
    end

    prob_deuce(idx) = proba_deuce_actuelle;
end

% Graphique de la probabilité d'atteindre le deuce en fonction de p
figure;
plot(p_values, prob_deuce, '-o');
xlabel('Probabilité p de gagner un point');
ylabel('Probabilité d''atteindre le deuce (40-40)');
title('Probabilité d''atteindre le deuce (40-40) en fonction de p');
grid on;
```

Voici le graphique obtenu :



Probabilité d’atteindre 40-40

Le graphique montre une courbe décroissante :

- La probabilité d’atteindre 40-40 est maximale lorsque *p* ≈ 0,5, car les niveaux des joueurs sont proches, rendant les situations serrées plus fréquentes. Dans ces cas, le jeu s’étire avec des points "avantage" (40-A ou A-40), offrant plus d’opportunités d’amplifier de légers écarts de compétence (par exemple, avec p = 0,51).
- À l’inverse, lorsque *p* est très élevé (proche de 1), le joueur plus fort gagne rapidement la majorité des points, réduisant les chances d’atteindre des scores

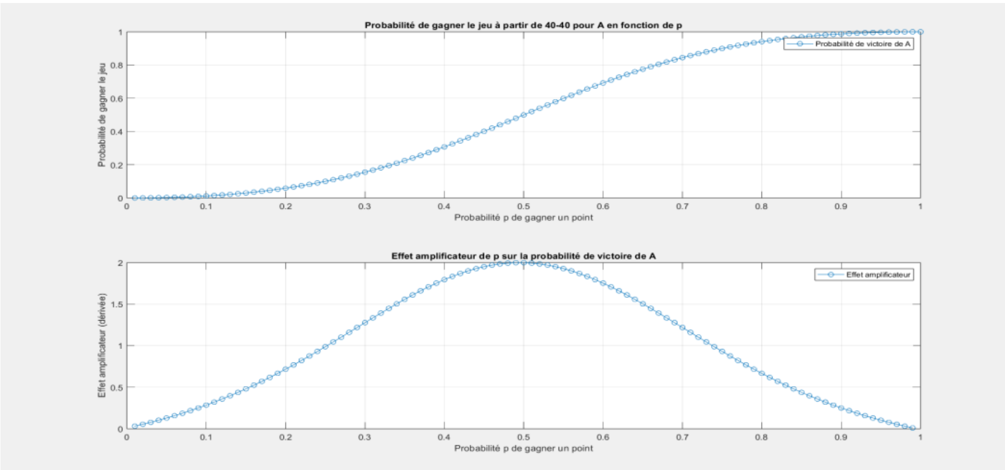
serrés comme 40-40. Cela explique pourquoi, dans ces situations, le jeu amplifie moins les petites différences de compétence.

Il est également intéressant d'analyser les "avantages" un peu plus profondeur.

Analyse des avantages

Lorsque le score atteint 40-40, les joueurs doivent gagner deux points consécutifs pour conclure le jeu, sinon le score revient à 40-40. Intuitivement, cette situation favorise un joueur légèrement supérieur, mais il est utile de confirmer cette intuition avec des graphiques.

En appliquant les mêmes méthodes que précédemment, nous avons obtenu deux résultats :

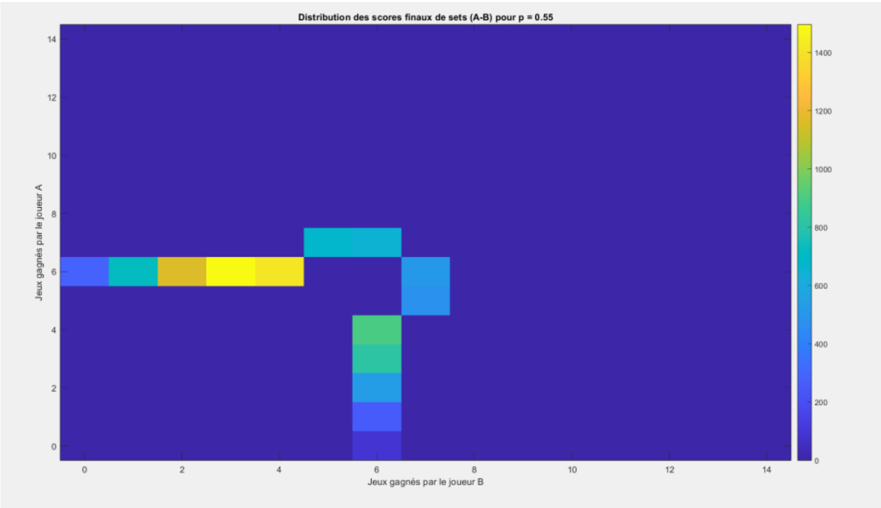


- **Probabilité de gagner le jeu à partir de 40-40 en fonction de *p***
  La probabilité de victoire augmente continuellement avec p, comme attendu. La courbe est plus raide autour de p = 0,5, montrant une forte sensibilité aux petites variations de compétence dans cette zone.
- **Effet amplificateur de *p* à partir de 40-40**
  L'effet amplificateur atteint un maximum autour de p = 0,5, soulignant que les situations serrées (40-40) favorisent l'amplification des petites différences de compétence.

Donc dans la continuité d'avant, ces deux graphiques confirment que le système de score d'un jeu au tennis amplifie les petites différences de compétences. Notamment, dans des situations serrées (ici de 40-40) où le fait qu'il faut gagner deux points de suite, tend à faire ressortir l'avantage du joueur légèrement plus fort.

b. Set

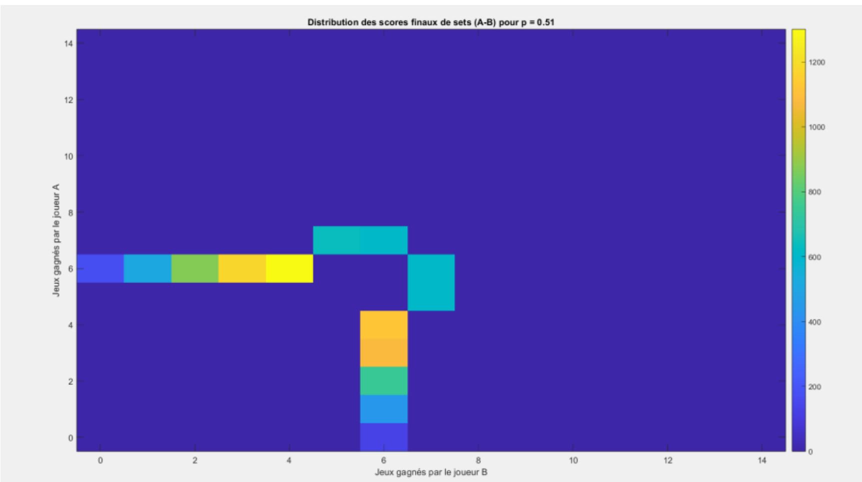
Pour le set, nous sommes partis sur une autre approche. En effet, nous avons d'abord fait une simulation de plusieurs sets (10 000), avec une petite différence de compétence (ici *p* = 0.55) afin de voir les scores plus probables :



Les observations principales que nous pouvons relevées sont :

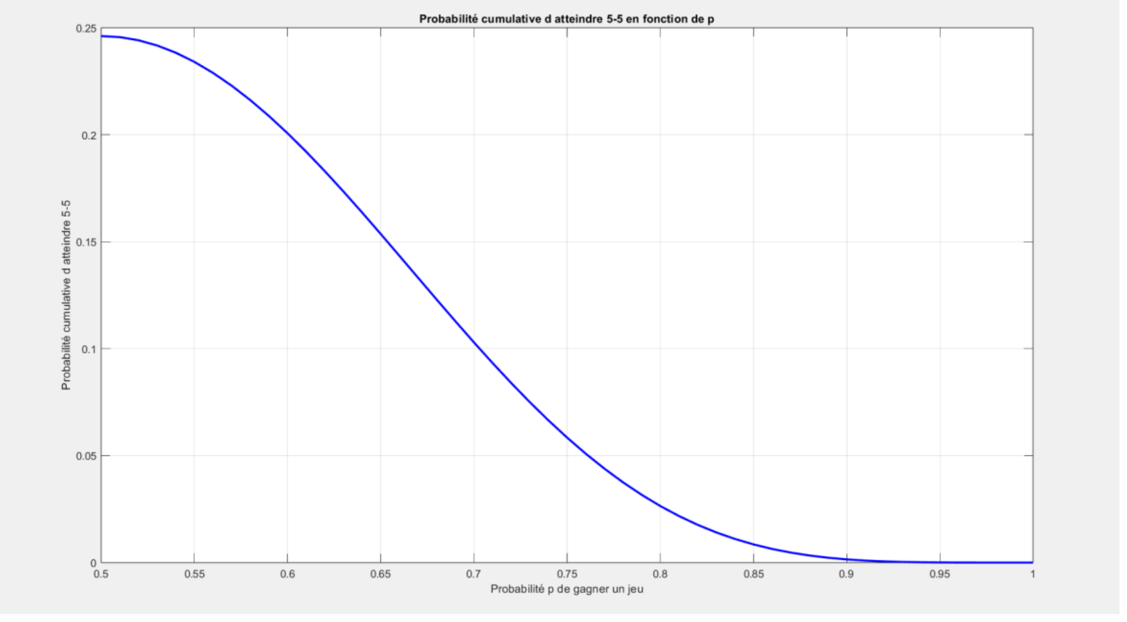
- Le score **6-3** est le plus fréquent, confirmant que le joueur A, avec un léger avantage de 5 %, tend à gagner le set avec une certaine marge.
- Les scores **6-2** et **6-4** pour A sont également plus probables que **6-4** pour B, ce qui illustre comment un léger avantage dans la probabilité de gagner un jeu peut se traduire par une augmentation significative de la probabilité de remporter un set.
- Les scores serrés, comme **7-6** ou **7-5** pour A, restent relativement fréquents, montrant que même avec un léger avantage, un set peut être disputé.

Nous avons fait la même analyse pour *p*=0.51 :



Pour ***p* = 0.51**, l'analyse montre un avantage clair pour le joueur A, malgré une très petite différence de niveau. Par exemple, la probabilité d'un score **6-4** en faveur de A est supérieure à celle d'un **4-6** pour B. De plus, les scores serrés comme **7-6** ou **6-7** sont plus fréquents qu'avec ***p* = 0.55**, ce qui est logique étant donné la proximité des niveaux des deux joueurs.

Pour un set, l'amplification des petites différences de niveau révèle un point intéressant : contrairement au jeu, où un score de **40-40** est très probable (supérieur à 60 % pour ***p* = 0.5**), un score de **5-5** dans un set est rare, même à ***p* = 0.5**, comme illustré sur ce graphique :



Ce phénomène, qui peut sembler contre-intuitif, renforce l'idée que l'accumulation d'un léger écart de niveau s'accentue au fil du set. Cela s'explique par plusieurs facteurs liés à la structure du set :

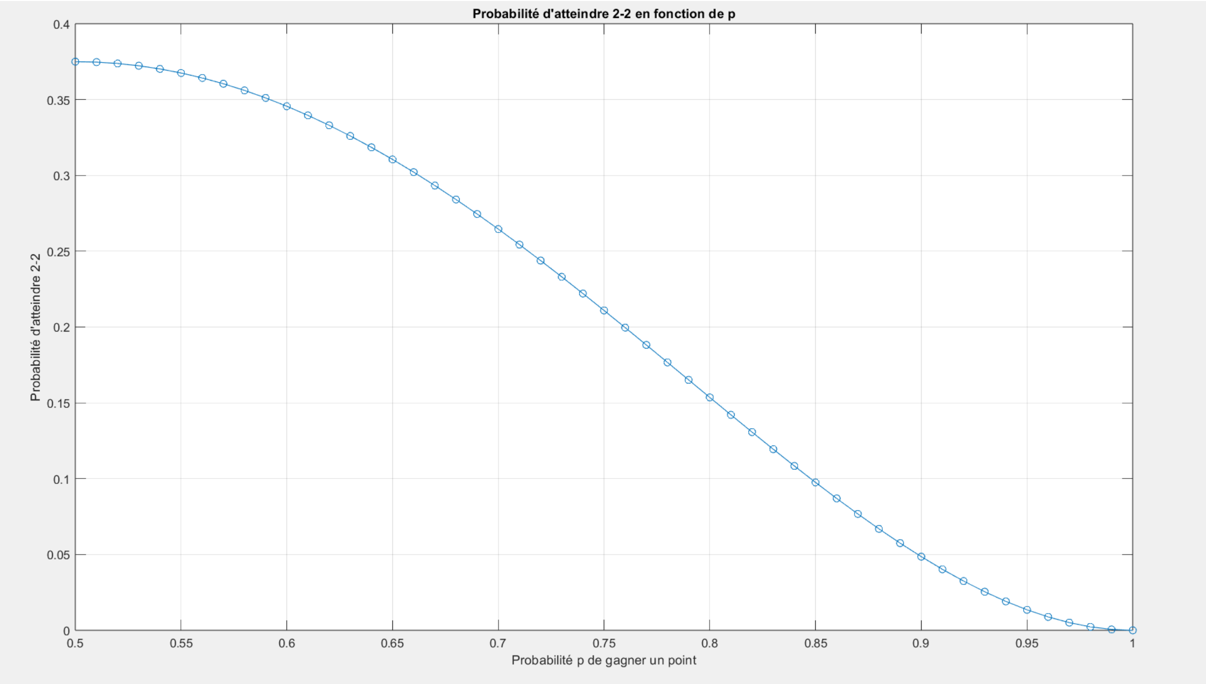
- **Accumulation des avantages** : Dans un set, un joueur doit remporter six jeux avec un écart de deux, ce qui accentue l'impact de chaque jeu gagné. Une fois un joueur en avance, il se rapproche de la victoire sans nécessairement passer par des scores serrés comme **5-5**, réduisant la probabilité d'atteindre ces égalités.

- **Comparaison avec un jeu** : Dans un jeu, des égalités comme **40-40** surviennent fréquemment car les points progressent petit à petit, et les joueurs peuvent revenir plusieurs fois à égalité. En revanche, un set compte des états plus marqués (jeux gagnés), rendant des égalités intermédiaires comme **5-5** bien moins fréquentes (le 5-5 ne peut être atteint qu'une seule fois durant un set).

- **Effet cumulatif des petits avantages** : Chaque jeu remporté dans un set a un impact plus significatif qu'un point gagné dans un jeu, et les écarts se comblent moins facilement. Cet effet cumulatif réduit encore la probabilité d'atteindre des scores serrés comme **5-5**.

c. Match

Pour le match, nous avons analysé la situation d'égalité **2-2** (deux sets partout). Voici le graphique obtenu :



On observe que la probabilité d'atteindre cette égalité tardive se situe entre celle d'un jeu (**40-40**) et celle d'un set (**5-5**). Cela s'explique par :

- **Fréquence des égalités** : Un score de **40-40** dans un jeu peut survenir plusieurs fois, contrairement à **2-2** dans un match ou **5-5** dans un set, qui ne peuvent être atteints qu'une seule fois.
- **Nombre réduit d'états dans un match** : Comparé à un set, un match comporte moins d'états (11 contre 41 pour un set), permettant aux joueurs d'atteindre des égalités comme **2-2** plus rapidement.

Ces résultats confirment que, dans ce modèle simplifié, un match amplifie les différences de compétence moins fortement qu'un set, en rendant les égalités intermédiaires moins fréquentes.

Cependant, ce modèle reste théorique : il repose sur une probabilité constante de gagner un set, sans prendre en compte les variations dues à la performance, au contexte ou à l'adversaire.

Pour affiner cette analyse et la rendre plus représentative des matchs réels, nous intégrerons maintenant des données spécifiques à des joueurs. Cela permettra d'évaluer comment le modèle reflète les différences de compétence avec des probabilités qui correspondent aux performances réelles de ces joueurs.

III- Comparaison avec la réalité

1. Jeu

Pour rendre l'analyse plus concrète, nous avons cherché et utilisé les statistiques réelles de joueurs professionnels de tennis, répartis par catégorie :

- **Joueur d'exception** : Rafael Nadal
- **Bon joueur** : Gilles Simon
- **Joueur moyen** : Corentin Moutet

Ces catégories offrent des points de comparaison intéressants pour évaluer la robustesse du modèle théorique. Comparer les prédictions du modèle avec les performances réelles de joueurs de niveaux différents permet de vérifier sa précision et sa capacité à représenter la réalité.

Voici les statistiques utilisées :

	Rafael Nadal (Joueur d'exception)	Gilles Simon (bon joueur)	Corentin Moutet (joueur moyen)
% de jeu gagnés en carrière	59.7	51.3	47.1
% de points gagnés en carrière	54.4	50.7	48.6

Pour pouvoir comparer, nous avons repris le premier graphique (celui qui affiche la probabilité de gagner de le jeu pour le joueur A en fonction de *p*) sauf que nous avons affiché cette fois dès 0.47 (pour voir les scores de Moutet) et avec une incrémentation de 0.001 :

```
joueurs = {'Nadal', 'Simon', 'Moutet'};
p_values = [0.544, 0.507, 0.486];  % Probabilités de points gagnés en carrière (vrais stats)
```

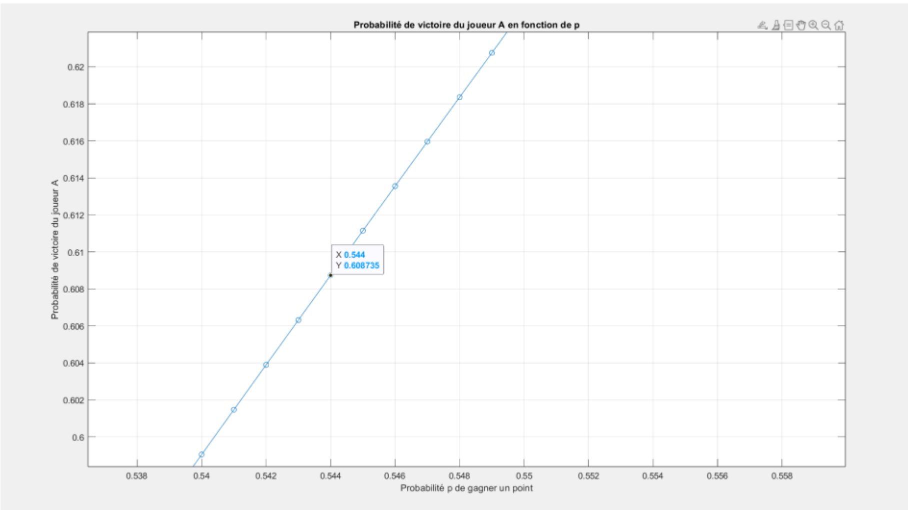
Ainsi, nous avons fait ce code afin de pouvoir comparer notre modèle à des valeurs réelles :

```matlab
joueurs = {'Nadal', 'Simon', 'Moutet'};
p_values = [0.544, 0.507, 0.486];  % Probabilités de points gagnés en carrière (vrais stats)
pourcentages_reels = [59.7, 51.3, 47.1];  % Pourcentage de jeux gagnés en carrière (vrais stats)
pourcentage_predits = zeros(size(p_values));

% Simulation pour chaque joueur
for i = 1:length(p_values)
    p = p_values(i);

    % Matrice de transition T pour le joueur actuel
    T = [
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S0
        p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S1
        1-p, 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S2
        0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S3
        0, 1-p,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S4
        0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S5
        0,   0,   0,   0,   0,   0,   0,   p, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S6
        0,   0,   0,   0, 1-p,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S7
        0,   0,   0, 1-p,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S8
        0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S9
        0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S10
        0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S11
        0,   0,   0,   0,   0,   0,   0,   0,   p, 1-p,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0;  % S12
        0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   0,   0,   0;  % S13
        0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0,   p,   0,   0,   0,   0,   0,   0,   0,   0;  % S14
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   p,   0,   p, 1-p,   0,   0;  % S15
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p,   0,   0,   0,   0;  % S16
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   0,   0;  % S17
        0,   0,   0,   0,   0,   0,   0,   0,   0,   p,   0,   0,   p,   p,   0,   0,   0,   p,   1,   0;  % SF1 (Victoire A)
        0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 1-p, 1-p,   0,   0, 1-p,   0, 1-p,   0,   0,   1;  % SF2
    ];

    v = zeros(20, 1);
    v(1) = 1;  % État initial (0-0)
    prob_victoire_jeu = (T^100) * v;
    pourcentage_predits(i) = prob_victoire_jeu(19) * 100;
end
fprintf('Comparaison des probabilités de gagner un jeu pour chaque joueur :\n');
for i = 1:length(joueurs)
    fprintf('%s - Modèle : %.2f%%, Statistiques réelles : %.2f%%\n', joueurs{i}, pourcentage_predits(i), pourcentages_reels(i));
end
```

Ainsi nous avons obtenus les résultats suivants :

```
Comparaison des probabilités de gagner un jeu pour chaque joueur :
Nadal - Modèle : 60.87%, Statistiques réelles : 59.70%
Simon - Modèle : 51.75%, Statistiques réelles : 51.30%
Moutet - Modèle : 46.50%, Statistiques réelles : 47.10%
```

- Avec Nadal qui a un *p* = 0.544. Ainsi en reprenant le schéma du modèle avec la probabilité de A de gagner le jeu en fonction de *p* :



(donc bien 60.87 % pour le modèle)

- Pour Gilles Simon, avec un ***p* = 0.507**, le modèle prédit une probabilité de victoire de **51.75 %** pour un jeu.
- Pour Corentin Moutet, avec un ***p* = 0.486**, le modèle donne une probabilité de **46.50 %**.

Ces résultats montrent une correspondance étroite entre le modèle et les données réelles, avec un écart d'environ **1 % au maximum**, ce qui valide la précision du modèle pour des jeux.

De plus, le test sur des joueurs de niveaux variés (exceptionnel, bon, moyen) renforce la robustesse du modèle et sa capacité à représenter fidèlement la réalité.

2. Set

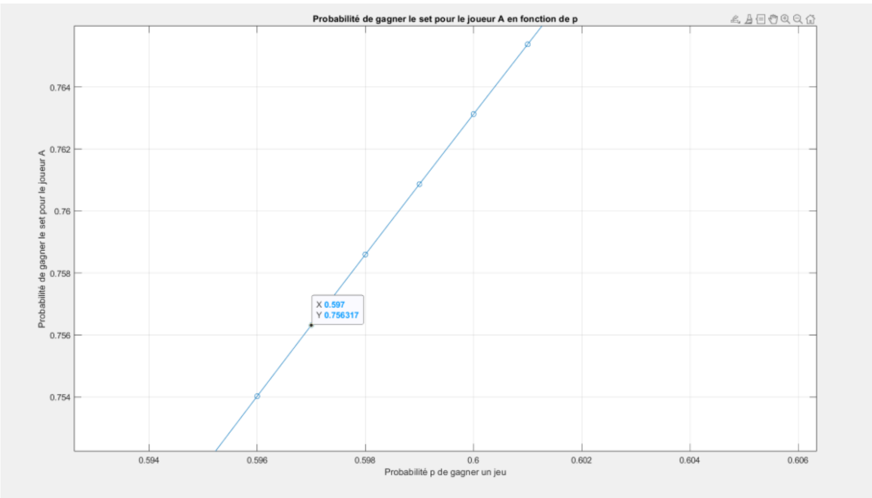
Comme pour le jeu, nous avons appliqué le modèle aux mêmes joueurs, cette fois en utilisant leurs statistiques spécifiques pour les sets :

	Rafael Nadal (Joueur d'exception)	Gilles Simon (bon joueur)	Corentin Moutet (joueur moyen)
% de jeu gagnés en carrière	59.7	51.3	47.1
% de set gagnés en carrière	77.1	53.6	43.2

En utilisant la même technique que pour le jeu, nous avons obtenus :

Avec un p = 0.597, Rafael Nadal atteint une probabilité de victoire de 75.63 % selon le modèle, comme le montre le graphique ci-dessous :

```
Comparaison des probabilités de gagner un set pour chaque joueur :
Nadal - Modèle : 75.63%, Statistiques réelles : 77.10%
Simon - Modèle : 53.67%, Statistiques réelles : 53.60%
Moutet - Modèle : 41.84%, Statistiques réelles : 43.20%
```



Pour Gilles Simon, avec un $p = 0.513$, le modèle prédit une probabilité de 53.67 %, et pour Corentin Moutet, avec un $p = 0.471$, une probabilité de 41.84 %.

Ces résultats montrent que, comme pour le jeu, le modèle représente bien la réalité pour différents types de joueurs, avec des écarts inférieurs à 2 % par rapport aux statistiques réelles des joueurs, et seulement 0.07 % pour Gilles Simon.

3. Match

Comme pour le jeu et le set, cette partie vise à rendre le modèle plus concret en utilisant les statistiques réelles des joueurs, mais cette fois appliquées à un match. Une subtilité importante réside dans le fait qu'en tennis, certains matchs se jouent en 3 sets gagnants (notamment lors des Grands Chelems), tandis que d'autres se jouent en 2 sets gagnants (dans les tournois classiques). Pour rester cohérents avec notre modèle simplifié de match en 3 sets gagnants, nous avons utilisé uniquement les statistiques issues des Grands Chelems.

Voici les statistiques utilisées :

	Rafael Nadal (Joueur d'exception)	Gilles Simon (bon joueur)	Corentin Moutet (joueur moyen)
% de set gagnés en Grand Chelem	80.7	55.7	46.3
% de match gagnés en Grand Chelem	87.3	59.9	47.1

Ainsi, en utilisant la même technique que pour les jeux et pour les stats, nous avons obtenus :

```
Comparaison des probabilités de gagner un match pour chaque joueur en Grand Chelem :
Nadal - Modèle : 94.73%, Statistiques réelles : 87.30%
Simon - Modèle : 60.60%, Statistiques réelles : 59.90%
Moutet - Modèle : 43.09%, Statistiques réelles : 47.10%
```

Le modèle simplifié pour les matchs en Grand Chelem montre des résultats globalement proches des statistiques réelles, surtout pour Gilles Simon, avec un écart de seulement 0.70%. Pour Nadal et Moutet, les écarts sont légèrement plus importants mais raisonnables, ce qui est cohérent avec les simplifications du modèle.

Plusieurs facteurs peuvent expliquer ces écarts :

- **Probabilité Constante** : Le modèle suppose une probabilité fixe *p* de gagner un set, sans intégrer les variations liées à l'adversaire, au contexte ou à l'évolution du match. Cela surestime les performances de Nadal, souvent confronté à des adversaires de haut niveau en Grand Chelem, tandis que pour Moutet, les écarts peuvent refléter des performances ponctuelles au-dessus de la moyenne.
- **Amplification des Écarts** : Les petites différences de probabilité s'amplifient à chaque niveau (jeu, set, match), rendant le modèle plus précis pour les jeux, un peu moins pour les sets, et encore moins pour les matchs, où les approximations s'accumulent.
- **Filtrage des Statistiques** : Les données pour les matchs concernent uniquement les Grand Chelem (en 3 sets gagnants), contrairement aux jeux et sets, qui s'appuient sur des statistiques globales de carrière. Cela peut expliquer cet écart.
- **Précision pour Gilles Simon** : L'écart faible pour Simon (+0.70 %) s'explique par sa régularité et une probabilité *p* proche de 0.5, où l'amplification des écarts est moindre, limitant les imprécisions.

Pour conclure par rapport à la comparaison avec les statistiques réelles, le modèle simplifié offre des prédictions globalement précises, surtout pour les joueurs avec des *p* proches de 0.5. Cependant, il est moins fiable pour les matchs que pour les jeux ou les sets, en raison des approximations accumulées et de l'hypothèse d'une probabilité constante, qui ne reflète pas les variations réelles de performance en Grand Chelem.

IV- Limite des modèles

1. Jeu

Le modèle pour un jeu reste une simplification. Des facteurs externes comme la surface, la fatigue ou la pression ne sont pas pris en compte. De plus, il n'intègre pas la distinction entre jeux de service et de retour, alors que le service confère généralement un avantage significatif. Sans ajuster la probabilité *p* selon le score (par exemple 15-15, 40-40), on ne peut pas saisir pleinement la dynamique réelle d'un jeu de tennis. Introduire un *p* variable rendrait le modèle plus fidèle, en montrant mieux comment un joueur légèrement supérieur profite de situations décisives.

2. Set

Au niveau du set, les limites restent similaires : le modèle suppose une probabilité constante pour gagner un jeu, n'intègre pas les tie-breaks et ignore le service. Ainsi, il ne capture pas les variations de performance dans les jeux décisifs (par exemple à 5-4) ni la complexité du format des sets en tournoi. Un modèle plus détaillé, prenant en compte le service, les moments clés et les tie-breaks, renforcerait sa pertinence.

3. Match

Le modèle du match, tel que simplifié, paraît indiquer une moindre amplification des différences de compétence, car il considère une probabilité constante de gagner un set sans moduler p selon la fatigue, la stratégie ou les conséquences d'un set difficilement gagné. Or, cette conclusion est liée à la simplification adoptée. En réalité, un match est une structure hiérarchique (match → sets → jeux → points) : chaque niveau accumule les différences déjà amplifiées par le niveau inférieur. Un match complet, intégrant plusieurs sets et donc de nombreux jeux, amplifie de fait plus encore les différences de compétence qu'un set ou un jeu pris isolément. L'absence de prise en compte de cette hiérarchie dans le modèle simplifié conduit donc à une vision incomplète.

4. Limites globales d'un modèle simplifié

Les trois niveaux (jeu, set, match) partagent des limites communes :

- Hypothèse de probabilités constantes, ignorant la fatigue, la pression ou les stratégies.
- Absence de distinction entre service et retour, pourtant cruciale dans la dynamique du tennis.
- Manque de contexte (balles de break, points de set ou de match) et d'adaptation entre les niveaux (un set difficilement gagné n'influence pas le suivant dans le modèle).

Pour conclure, bien que ces modèles simplifiés offrent un éclairage sur les mécanismes d'amplification de la différence de compétence, ils nécessitent des améliorations pour capturer la réalité du tennis professionnel, en particulier l'importance du service, des conditions variables et la hiérarchie des niveaux (jeu, set, match).