**TD - Rendu du TD n°2**

**Comportement dynamique d’un réseau récurrent**

2.2. Dessinez les trajectoires pour chaque état table, pour une activation synchrone :

(0,0,0,0)->(0,0,0,0) C’est un état stable sur (0,0,0,0)

(0,0,0,1)->(1,0,1,0)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(0,0,1,0)->(1,1,0,1)->(0,0,1,0) C’est un cycle de taille 2 :

[1, 1, 0, 1] -> [0, 0, 1, 0]

(0,0,1,1)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(0,1,0,0)->(0,0,1,0)->(1,1,0,1) C’est un cycle de taille 2 :

[0, 0, 1, 0] -> [1, 1, 0, 1]

(0,1,0,1)->(0,0,1,0)->(1,1,0,1) C’est un cycle de taille 2 :

[0, 0, 1, 0] -> [1, 1, 0, 1]

(0,1,1,0)->(0,1,1,0) Se stabilise sur (0,1,1,0)

(0,1,1,1)->(1,0,1,0)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(1,0,0,0)->(0,0,1,1)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(1,0,0,1)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(1,0,1,0)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(1,0,1,1)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(1,1,0,0)->(0,0,1,0)->(1,1,0,1) C’est un cycle de taille 2 :

[0, 0, 1, 0] -> [1, 1, 0, 1]

(1,1,0,1)->(0,0,1,0)->(1,1,0,1) C’est un cycle de taille 2 :

[0, 0, 1, 0] -> [1, 1, 0, 1]

(1,1,1,0)->(0,0,1,1)->(1,0,1,1) Se stabilise sur (1,0,1,1)

(1,1,1,1)->(1,0,1,1) Se stabilise sur (1,0,1,1)

2.3.1 Dessinez les trajectoires d'états correspondants à une activation asynchrone

(0,0,0,0)-> C’est un état stable

(0,0,0,1)->(1,0,1,1)-> Se stabilise

(0,0,1,0)->(1,0,1,1)-> Se stabilise

(0,0,1,1)->(1,0,1,1)-> Se stabilise

(0,1,0,0)->(0,0,0,0)-> Se stabilise

(0,1,0,1)->(0,0,1,1)->(1,0,1,1)-> Se stabilise

(0,1,1,0)-> C’est un état stable stable

(0,1,1,1)->(1,0,1,1)-> Se stabilise

(1,0,0,0)->(0,0,0,0)-> Se stabilise

(1,0,0,1)->(1,0,1,1)-> Se stabilise

(1,0,1,0)->(1,0,1,1)-> Se stabilise

(1,0,1,1)-> C’est un état stable stable

(1,1,0,0)->(0,0,0,0)-> Se stabilise

(1,1,0,1)->(0,0,1,1)->(1,0,1,1)-> Se stabilise

(1,1,1,0)->(0,1,1,0)-> Se stabilise

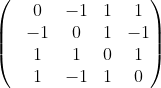
(1,1,1,1)->(1,0,1,1)-> Se stabilise

2.3.2. Comparez les trajectoires synchrones et asynchrones

On remarque que le réseau à activation synchrone possède des cycles de taille 2, alors que les activations asynchrones ne provoquent que des cycles de taille 1 au bout de quelques itérations, ou stable dès le début.

4. Conclusion

1. **Ecrire la matrice des poids de notre petit réseau**



1. **Que peut-on dire de cette matrice ?**

Nous pouvons constater que cette matrice est symétrique.

**3. Faire des essais avec le programme et différentes matrices de poids**

Cf. JeuDeTest.txt

**4. En fonction des observations, proposer des hypothèses sur la forme que doit avoir la matrice des poids pour que le réseau converge vers des points fixes**

On remarque qu’avec une matrice qui n’est pas symétrique, le réseau à activation synchrone génère énormément de cycles de taille 2, et met en moyenne plus d’itérations à se stabiliser. Aussi on remarque que le nombre de configuration déjà stable est grandement réduit. Quant au réseau à activation asynchrone, il se stabilise très rapidement et on retombe régulièrement sur les mêmes configurations

Avec une matrice symétrique, on remarque beaucoup de configurations qui se stabilisent et deviennent des points fixes

Il semble donc cohérent de formuler l’hypothèse qu’une matrice de poids symétrique est nécessaire afin que le réseau converge vers des points fixes