# RANDOM AUXETIC NETWORK NOTES

I here, I will try to cover up the details and explanations for making my code efficient. In order to locate the bottlenecks in my code, I need to first laydown the flow of information in my code.

Notes:

* I tried implementing my matrix jugglert on FDV but
  + There is no conclusive evidence as such for speed gains
* Conjugate Gradient take most of the time.
  + However, the timigs of code is still not that bad.
* It takes about 0.70 seconds to python to run for 3x3 system. I think the code is itself fast enough as of now for 3x3 system. What I need to check is that how much time does it takes for 4x4, 5x5 and 6x6 systems?
  + Focus now on extending the code for larger systems, then after that I can try a few more things like:
    - Making the matrix mapping efficient.
    - Making the CG efficient.

# 6x6 System

# 6x6 System

59

58

57

56

55

54

53

52

51

50

49

48

47

46

45

44

43

42

41

400

39

38

37

36

35

34

33

32

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

15

14

13

12

11

10

9

8

7

6

5

4

3

22

1

0