

Atractores del Autómata Celular de Wolfram

Dentro de la literatura se clasifican a los CA en 4 Clases:

- Uniformes
- Periódicos
- Caóticos
- Complejos

Con el análisis de sus atractores podemos interpretar su Clase ademas de ver comportamientos periódicos en las reglas.

Código y Funcionamiento:

```
gio in ~ λ cd Documentos/Complex_Systems/elementary_cellular_automata
gio in ~/Documentos/Complex_Systems/elementary_cellular_automata on main • λ go run main.go
Enter Rule: 110
*****
Rule: 110 Length: 14 Generations: 100
*****
Select One Option:
* 1. Print Rule with #Generations
* 2. Make an Atractor's File
* 3. Print Rule and Make an Atractor's File
* 4. Exit
*****
Option: 1
*****
Select One Option:
* 1. Random Start
* 2. Start with 1
* 3. Digit a Value
*****
Option: 2
*****
```



```
package main

import (
    "fmt"
    "math/big"
    "math/rand"
    "os"
    "strconv"
    "strings"
)

const len_ac int = 14
const len_aa uint = 14
const gen_max = 100

var bucket [1 << len_ac]bool

func main() {
    const cells1 = len_ac
    const cells2 = len_aa

    a := big.NewInt(1)
    a.Lsh(a, cells2/2)
```

```
var rule uint

//rules := []uint{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 40, 41, 42, 43, 44, 45, 46, 50, 51, 54, 56, 57, 58, 60, 62, 72, 73, 74, 76, 77, 78, 90, 94, 104, 105, 106, 108, 110, 122, 126, 128, 130, 132, 134, 136, 138, 140, 142, 146, 150, 152, 154, 156, 160, 162, 164, 168, 170, 172, 178, 184, 200, 204, 232}

fmt.Println("Enter Rule: ")

fmt.Scanln(&rule)

fmt.Println("*****")
fmt.Println("Rule:", rule, " Length:", cells1, " Generations:", gen_max)
fmt.Println("*****")

fmt.Println("Select One Option: *")
fmt.Println("* 1. Print Rule with #Generations *")
fmt.Println("* 2. Make an Atractor's File *")
fmt.Println("* 3. Print Rule and Make an Atractor's File *")
fmt.Println("* 4. Exit *")
fmt.Println("*****")

fmt.Print("Option: ")

var option int

fmt.Scanln(&option)

switch option {

case 1:
    fmt.Println("*****")
}
```

```
fmt.Println("Select One Option: *")

fmt.Println("* 1. Random Start *")

fmt.Println("* 2. Start with 1 *")

fmt.Println("* 3. Digit a Value *")

fmt.Println("*****")
fmt.Print("Option: ")

var option2 int

fmt.Scanln(&option2)

fmt.Println("*****")

switch option2 {

case 1:

a.Rand(rand.New(rand.NewSource(3)), a.Lsh(a, cells2))

print_rule(rule, cells1, gen_max, a)

case 2:

print_rule(rule, cells1, gen_max, a)

case 3:

var value int64

fmt.Print("Value between 0 and ", (1<<cells1)-1, ": ")

fmt.Scanln(&value)

a.SetInt64(value)

print_rule(rule, cells1, gen_max, a)

}

case 2:

make_atractor(rule, cells1)
```

```
case 3:
```

```
fmt.Println("*****")
fmt.Println("Select One Option: *")
fmt.Println("* 1. Random Start *")
fmt.Println("* 2. Start with 1 *")
fmt.Println("* 3. Digit a Value *")
fmt.Println("*****")
fmt.Print("Option: ")
var option3 int
fmt.Scanln(&option3)
fmt.Println("*****")
switch option3 {
case 1:
    a.Rand(rand.New(rand.NewSource(3)), a.Lsh(a, cells2))
    print_rule(rule, cells1, gen_max, a)
case 2:
    print_rule(rule, cells1, gen_max, a)
case 3:
    var value2 int64
    fmt.Print("Value between 0 and ", (1<<cells1)-1, ": ")
    fmt.Scanln(&value2)
    a.SetInt64(value2)
    print_rule(rule, cells1, gen_max, a)
}
```

```
make_atractor(rule, cells1)

case 4:

os.Exit(0)

default:

fmt.Println("Invalid Option")

}

}

func print_rule(rule uint, cells, generations int, a *big.Int) {

output := func() {

fmt.Println(strings.Replace(strings.Replace(
fmt.Sprintf("%0*b", cells, a), "0", " ", -1), "1", "\u25a1", -1))

}

output()

a1 := new(big.Int)

set := func(cell int, k uint) {

a1.SetBit(a1, cell, rule>>k&1)

}

last := cells - 1

for r := 0; r < generations; r++ {

k := a.Bit(last) | a.Bit(0)<<1 | a.Bit(1)<<2

set(0, k)

for c := 1; c < last; c++ {

k = k>>1 | a.Bit(c+1)<<2

```

```
set(c, k)

}

set(last, k>>1|a.Bit(0)<<2)

a, a1 = a1, a

output()

}

}

func make_atractor(rule uint, cells int) {

var path_folder = "r" + strconv.FormatUint(uint64(rule), 10)

errF := os.Mkdir(path_folder, 0755)

if os.Exists(errF) {

println("Folder ya existente")

}

var path = path_folder + "/"

var file_name = "atractor_rule_" + strconv.FormatUint(uint64(rule), 10) + "_len_" + strconv.Itoa(len_ac) + ".m"

var file, err = os.Create(path + file_name)

if isError(err) {

return

}

defer file.Close()
```

```
_ , err2 := file.WriteString("G = digraph;\n")
```

```
if isError(err2) {
```

```
    return
```

```
}
```

```
fmt.Println("Rule:", rule, " Length:", cells)
```

```
var i int64
```

```
for i = 0; i < (1 << len_ac); i++ {
```

```
    a := big.NewInt(i)
```

```
    if !bucket[i] {
```

```
        elem_atractor(rule, cells, a, file)
```

```
    } else {
```

```
        continue
```

```
}
```

```
}
```

```
_ , err3 := file.WriteString("\nplot(G,'Layout','force');")
```

```
if isError(err3) {
```

```
    return
```

```
}
```

```
fmt.Println("Done")
```

```
}
```

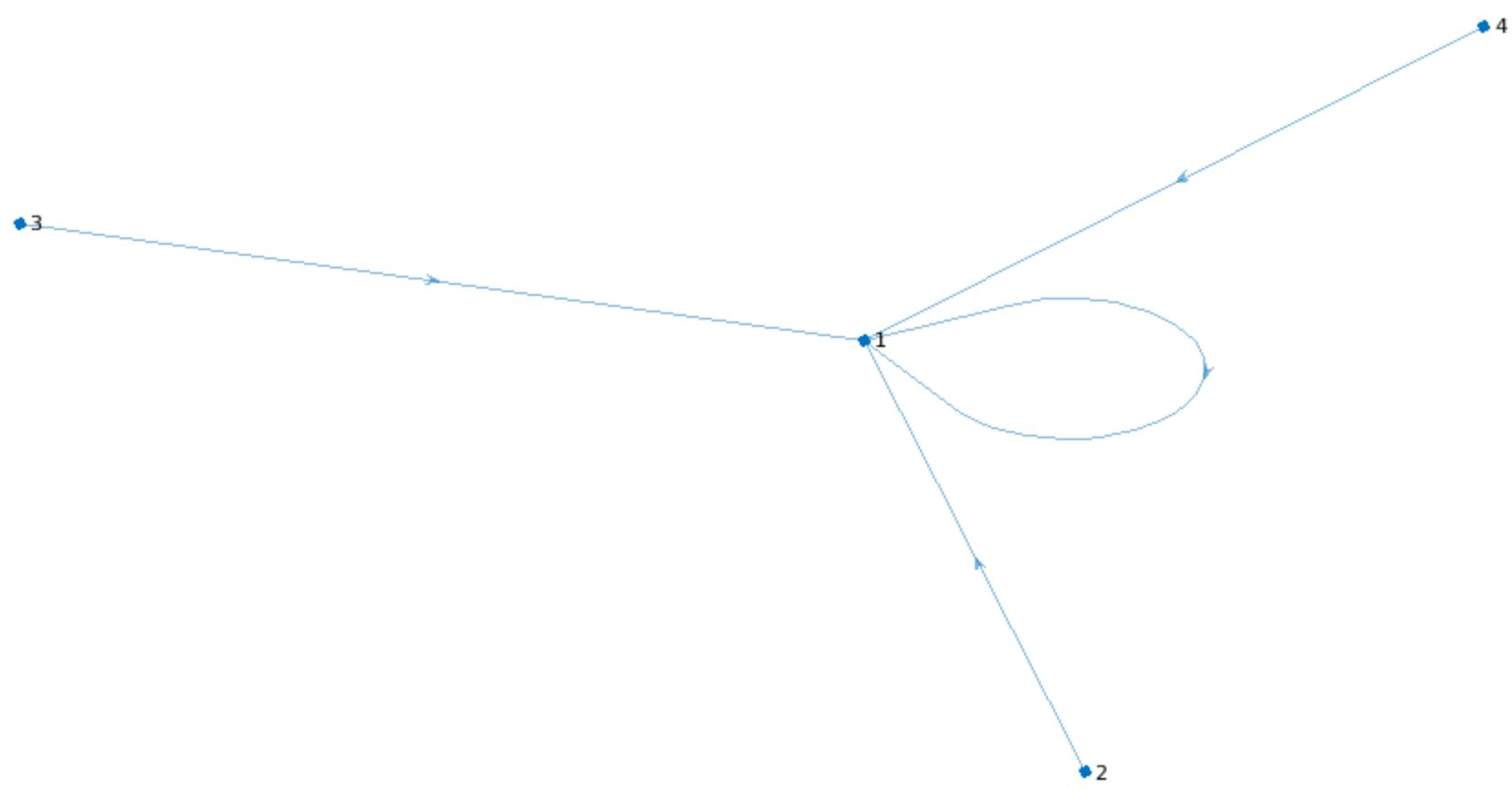
```
func elem_atractor(rule uint, cells int, a *big.Int, file *os.File) {
    bucket[a.Int64()] = true
    a1 := new(big.Int)
    set := func(cell int, k uint) {
        a1.SetBit(a1, cell, rule>>k&1)
    }
    last := cells - 1
    for {
        k := a.Bit(last) | a.Bit(0)<<1 | a.Bit(1)<<2
        set(0, k)
        for c := 1; c < last; c++ {
            k = k>>1 | a.Bit(c+1)<<2
            set(c, k)
        }
        set(last, k>>1|a.Bit(0)<<2)
        astr := strconv.FormatInt(a.Int64()+1, 10)
        a1str := strconv.FormatInt(a1.Int64()+1, 10)
        _, err := file.WriteString("G = addedge(G," + astr + "," + a1str + ");\n")
        a, a1 = a1, a
    }
    if isError(err) {
        return
    }
}
```

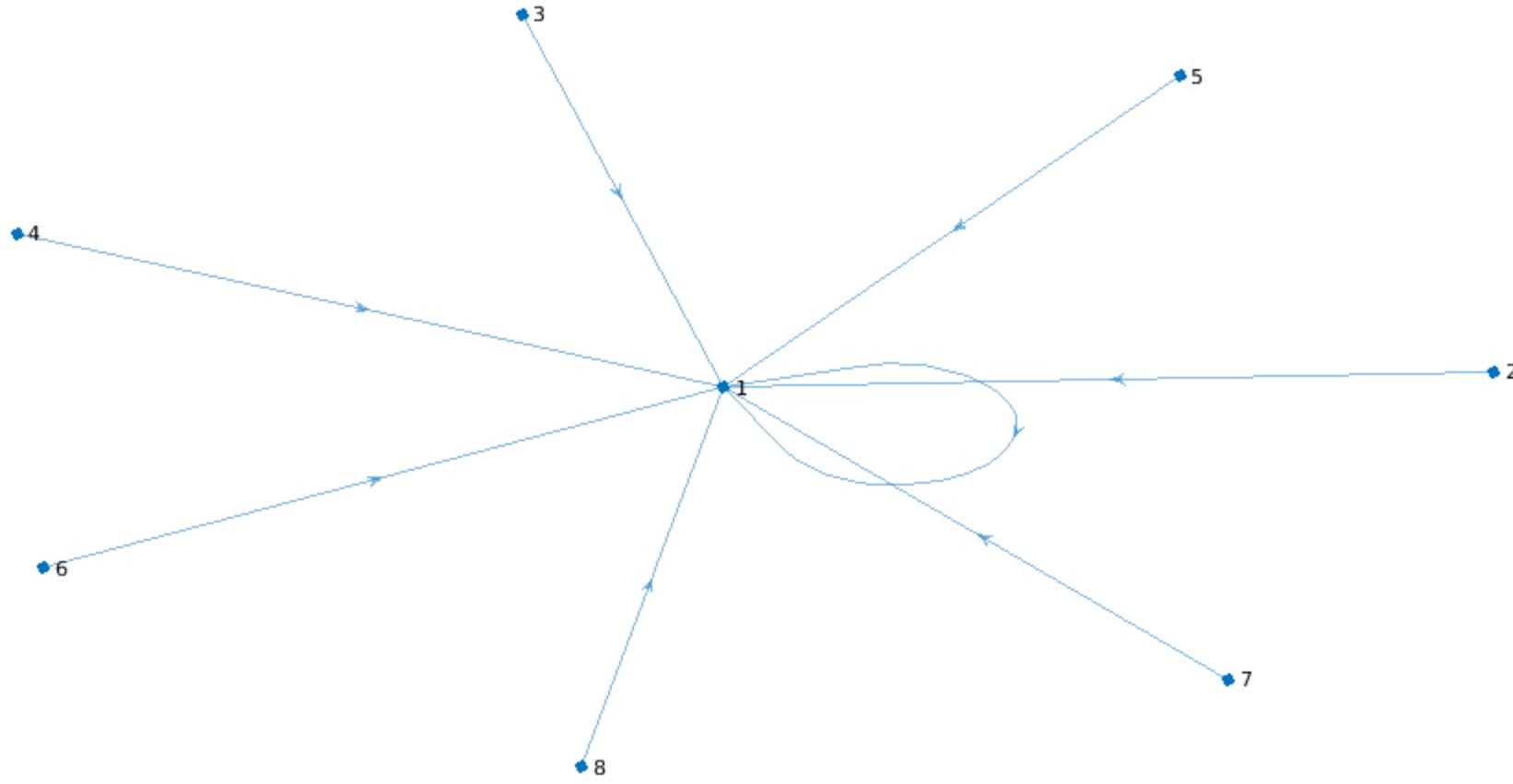
```
if bucket[a.Int64()] {  
    break  
}  
  
bucket[a.Int64()] = true  
}  
}
```

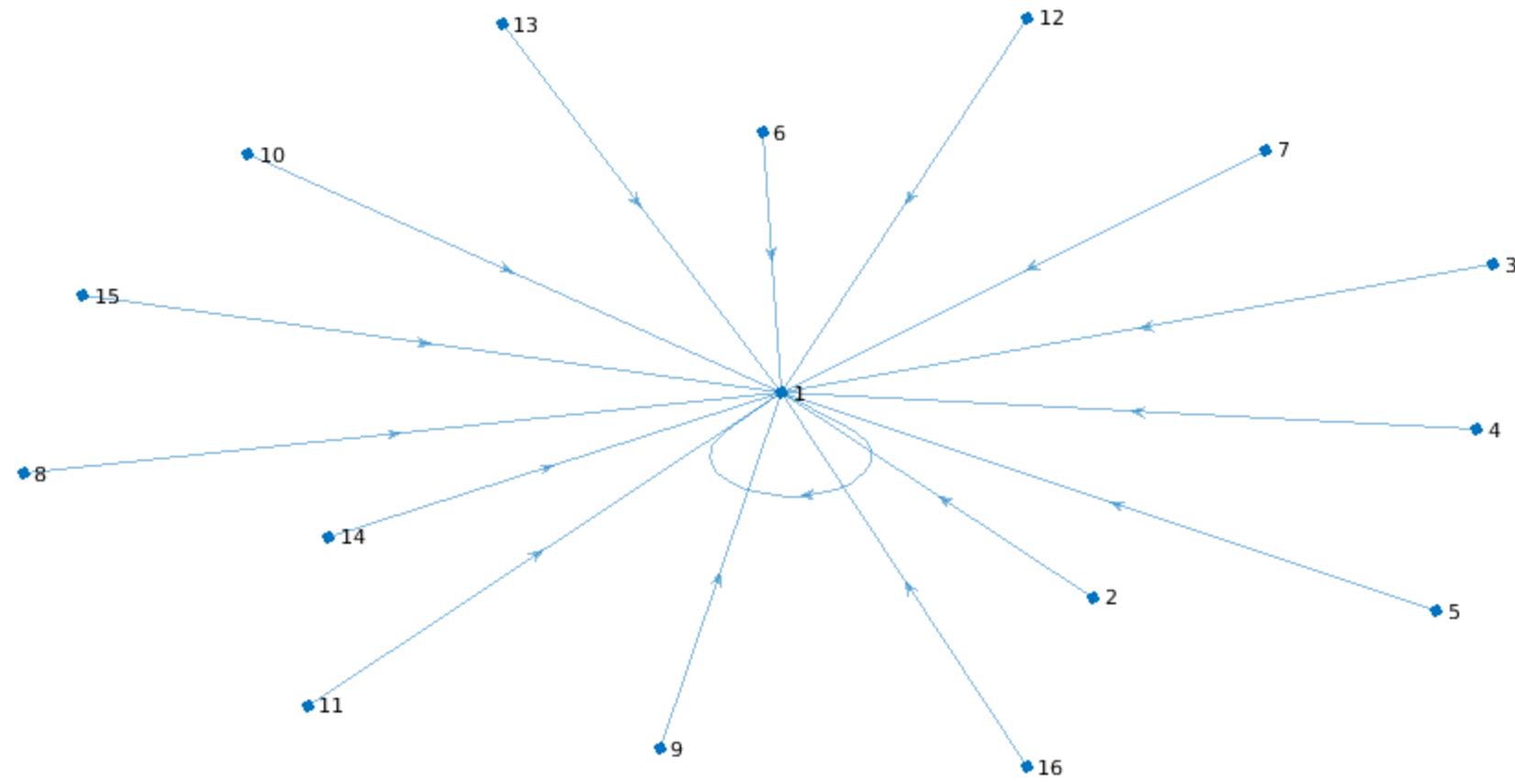
```
func isError(err error) bool {  
    if err != nil {  
        fmt.Println(err.Error())  
    }  
  
    return (err != nil)  
}
```

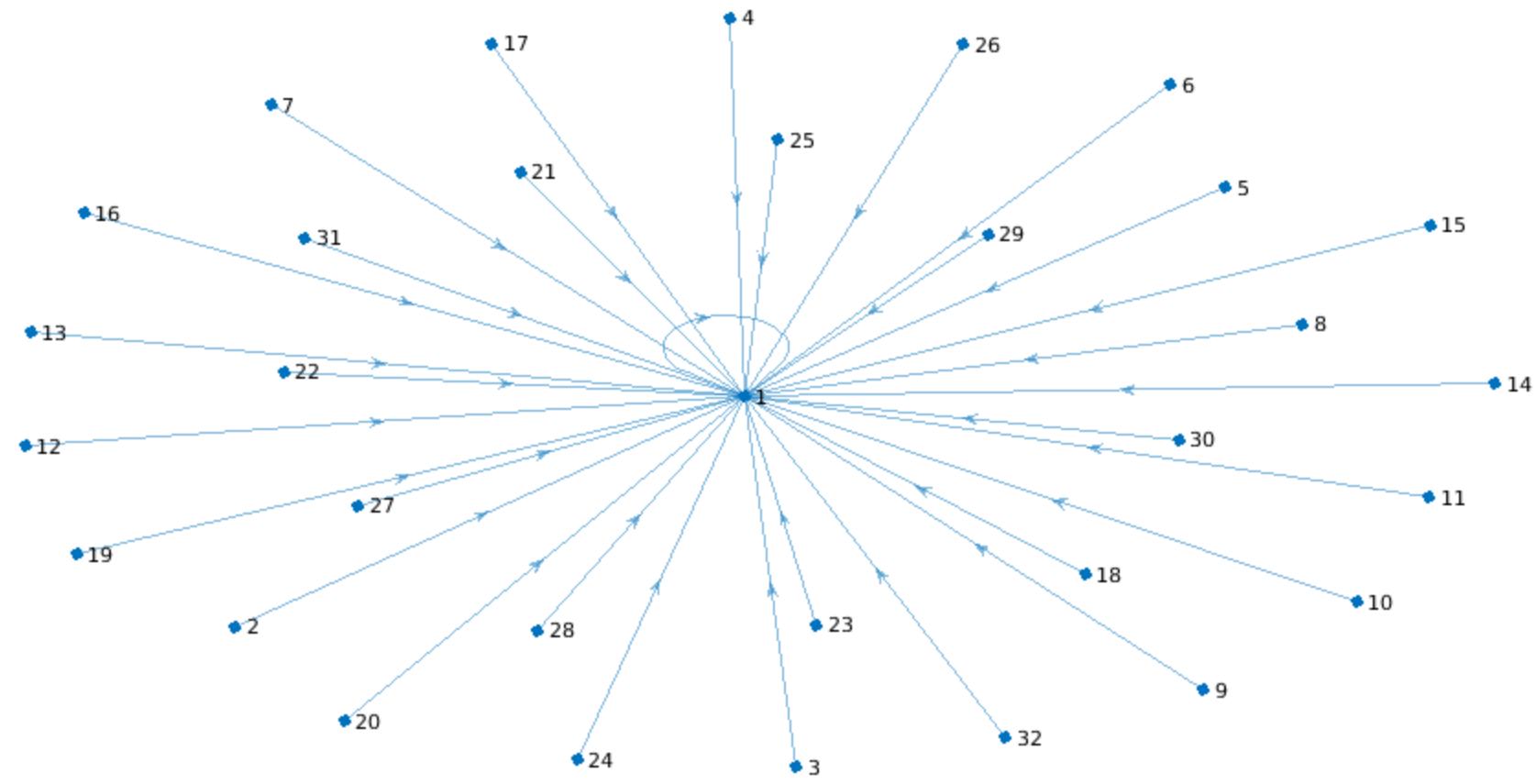
Reglas:

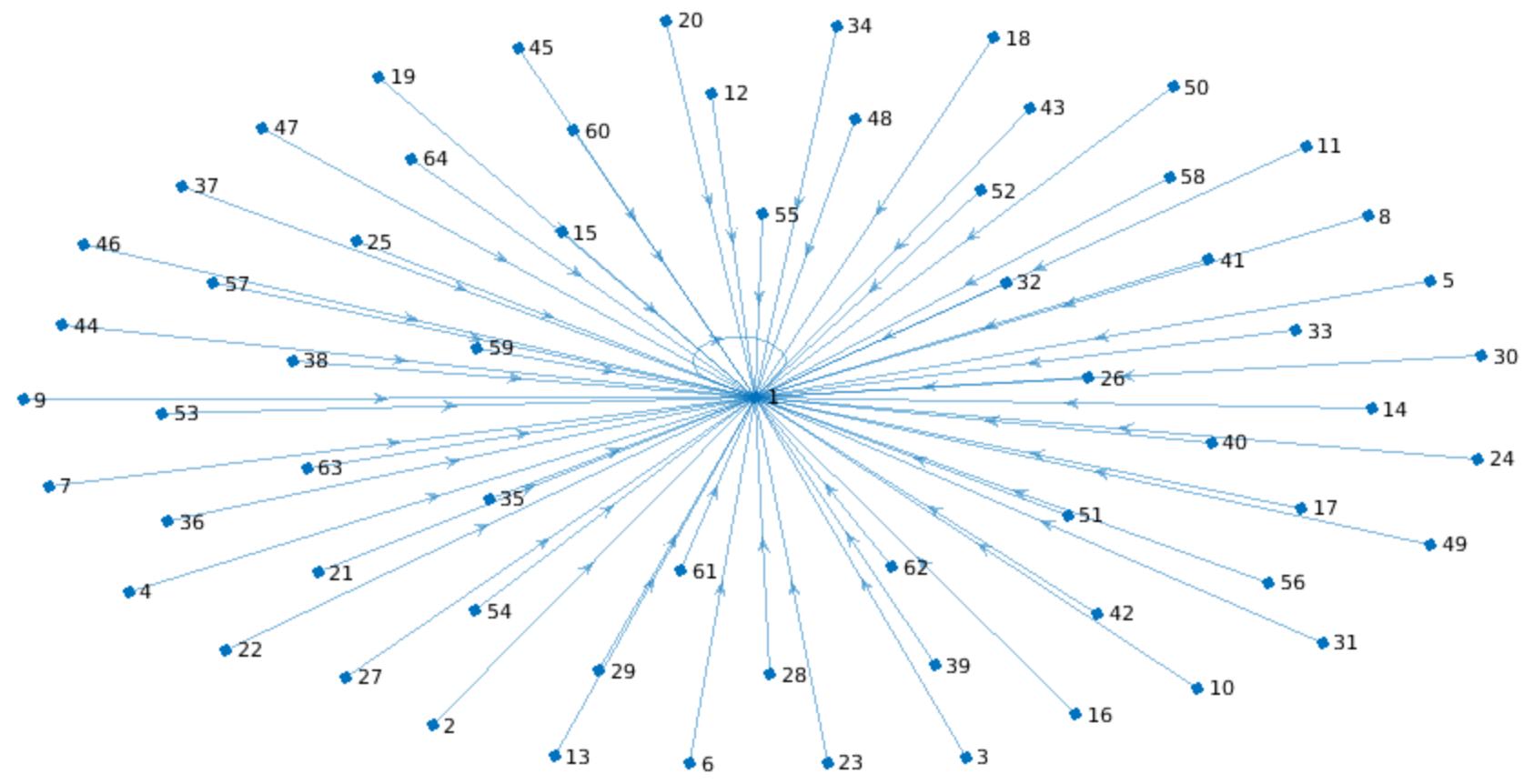
0

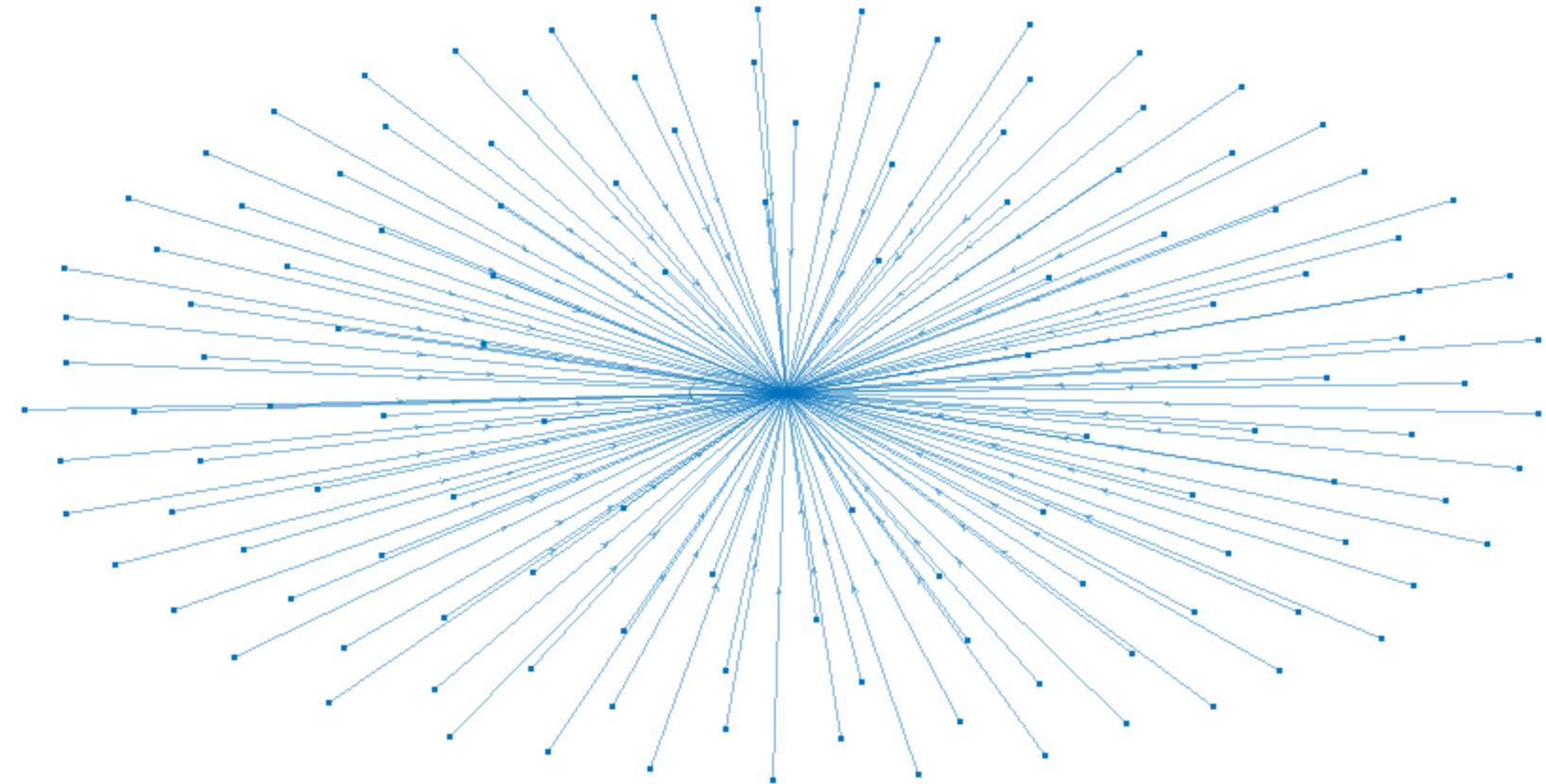


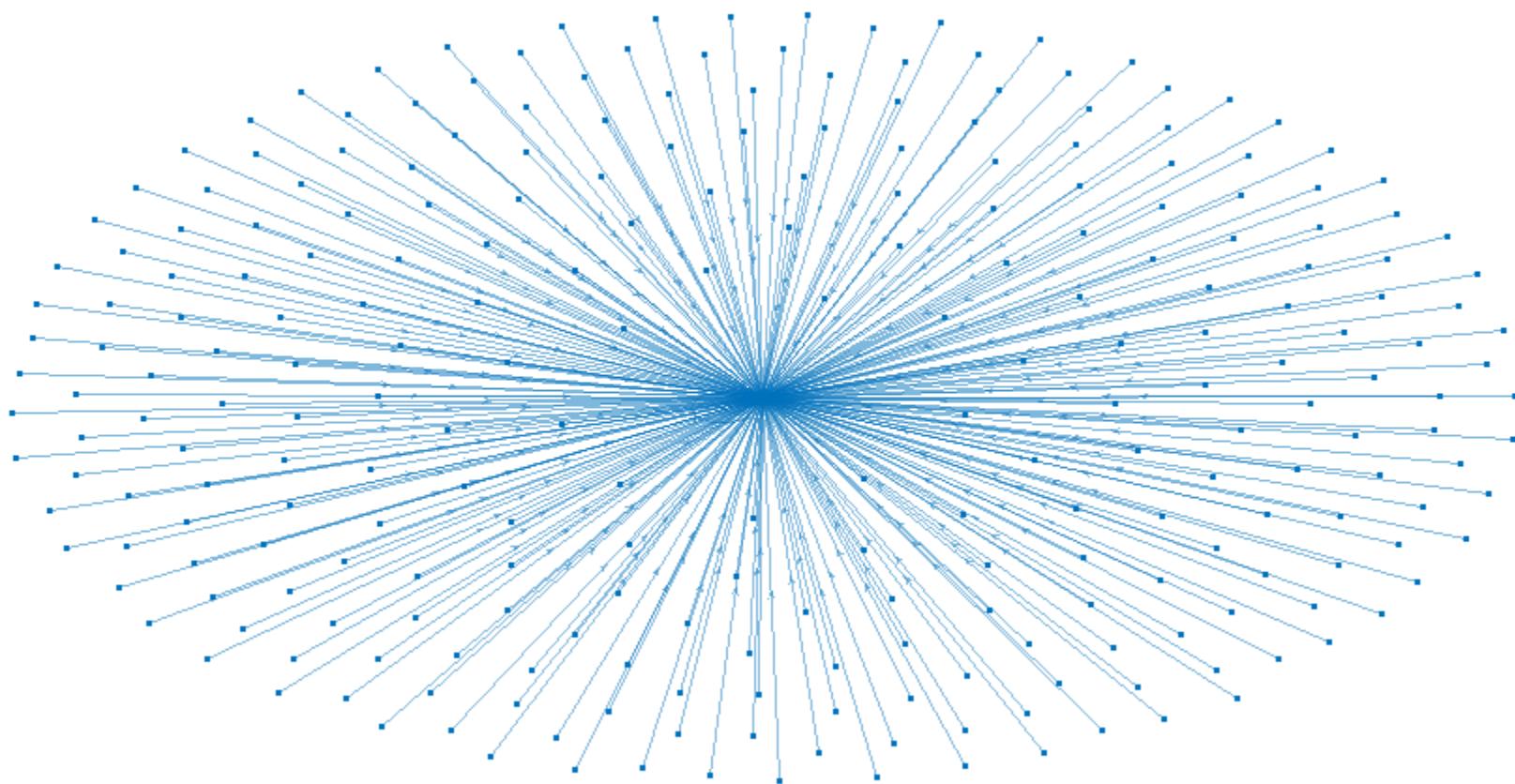


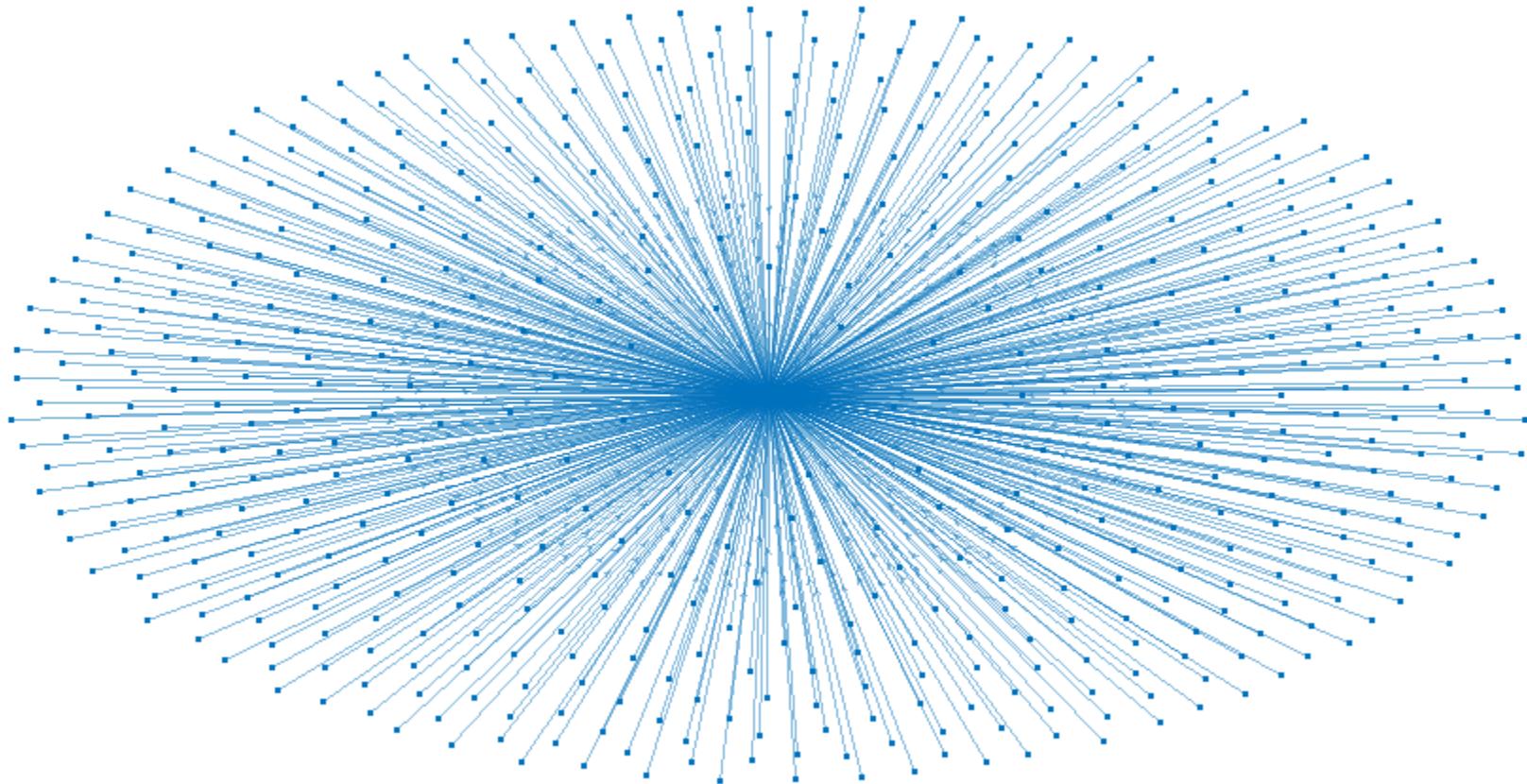


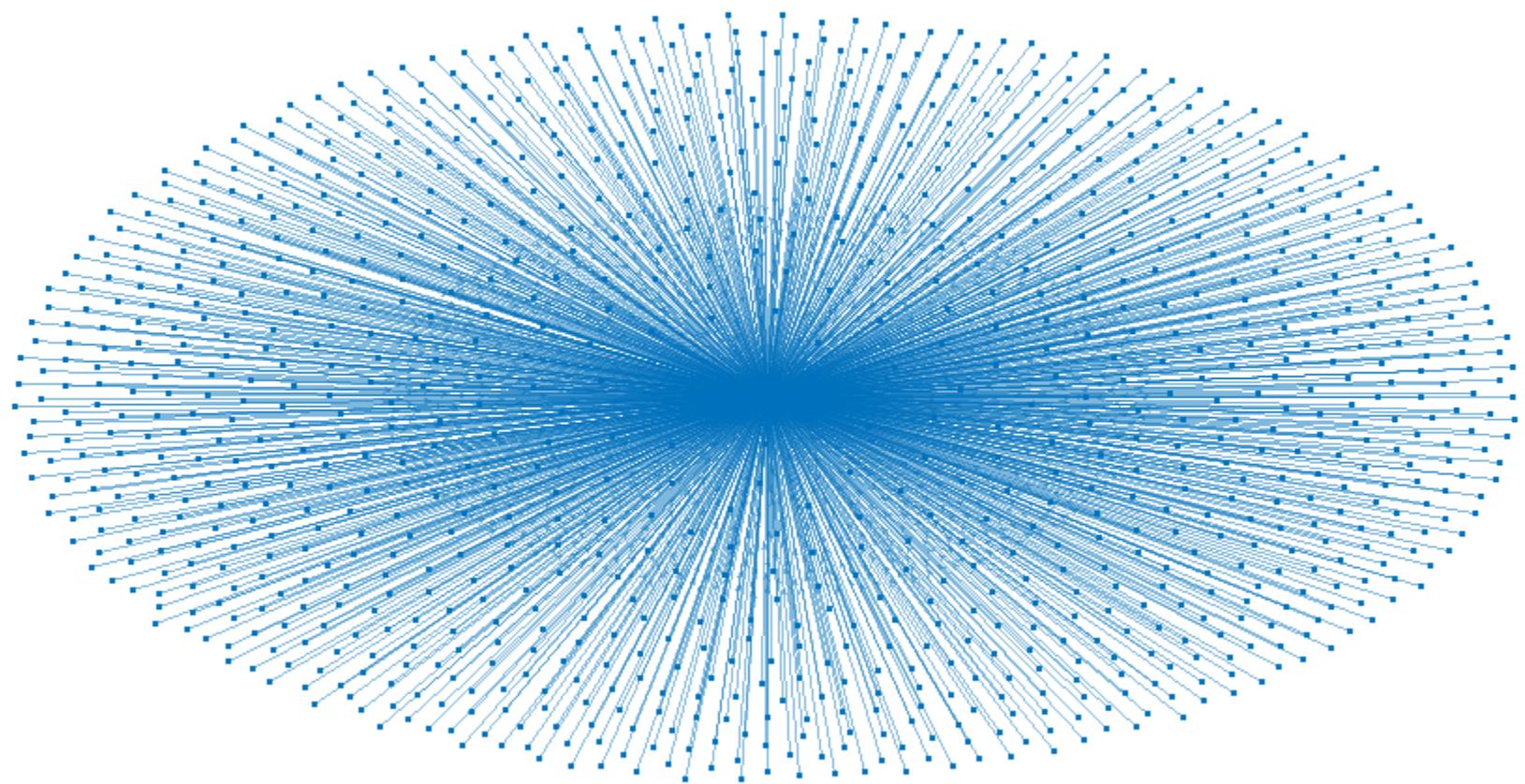


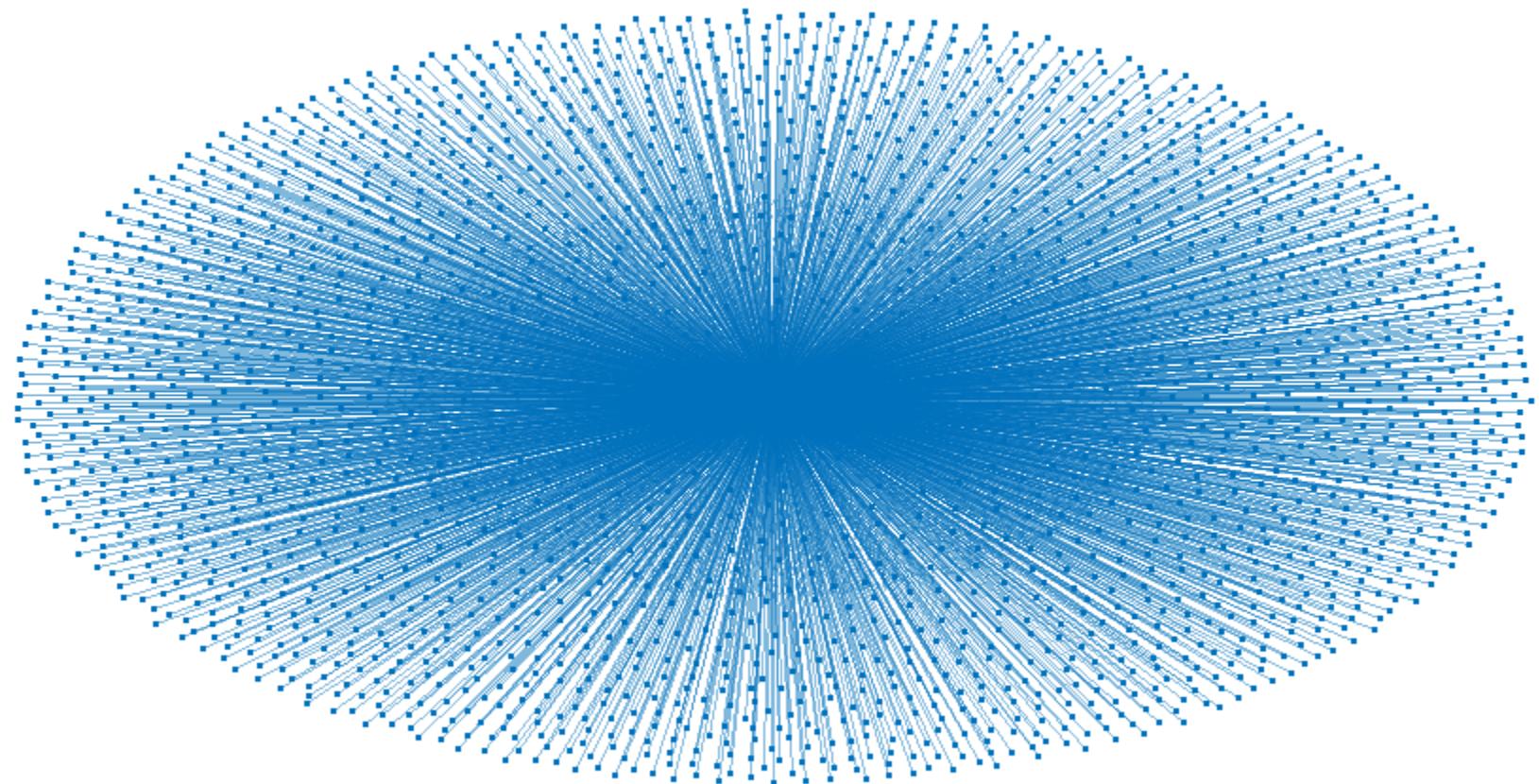


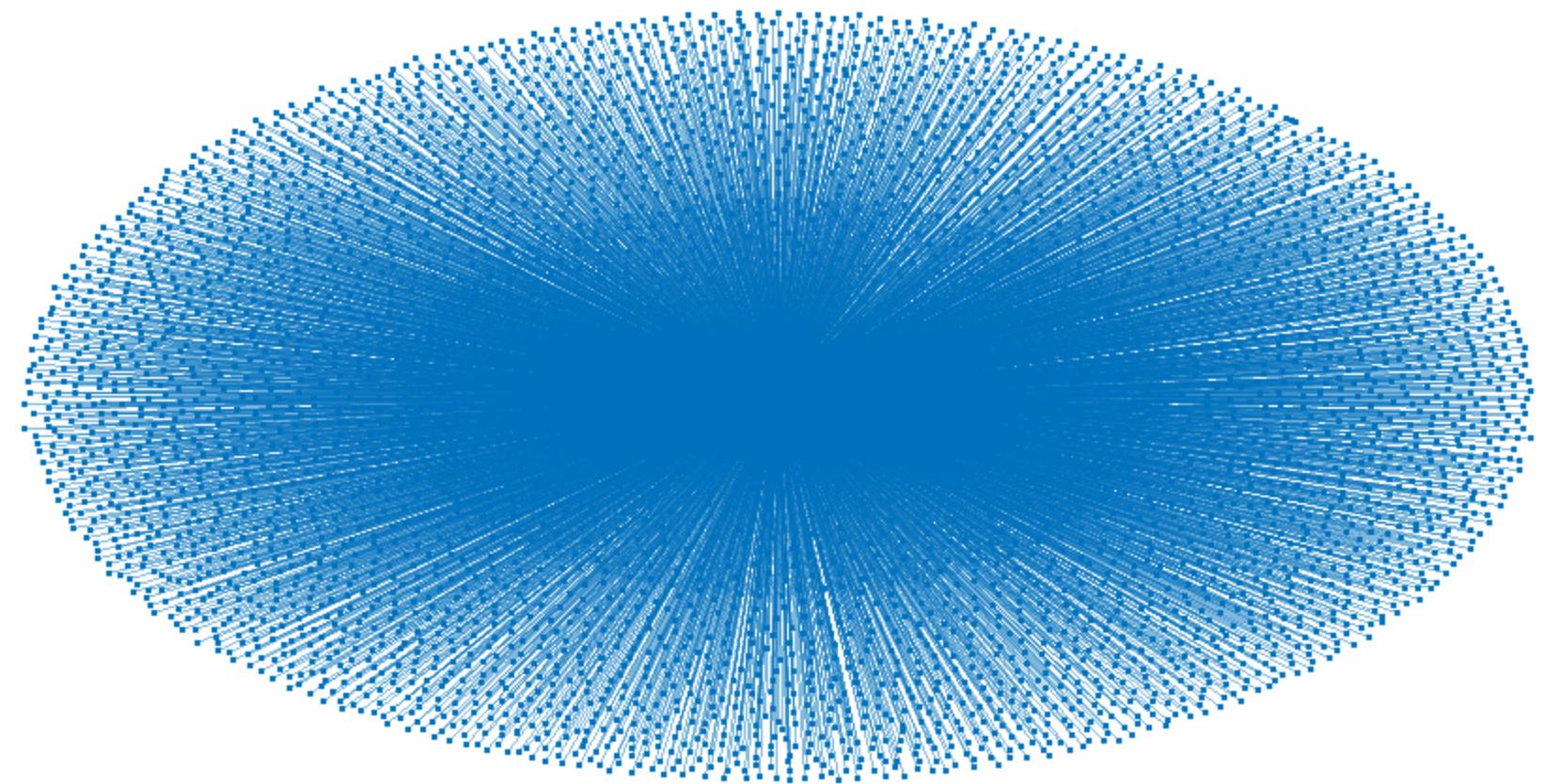


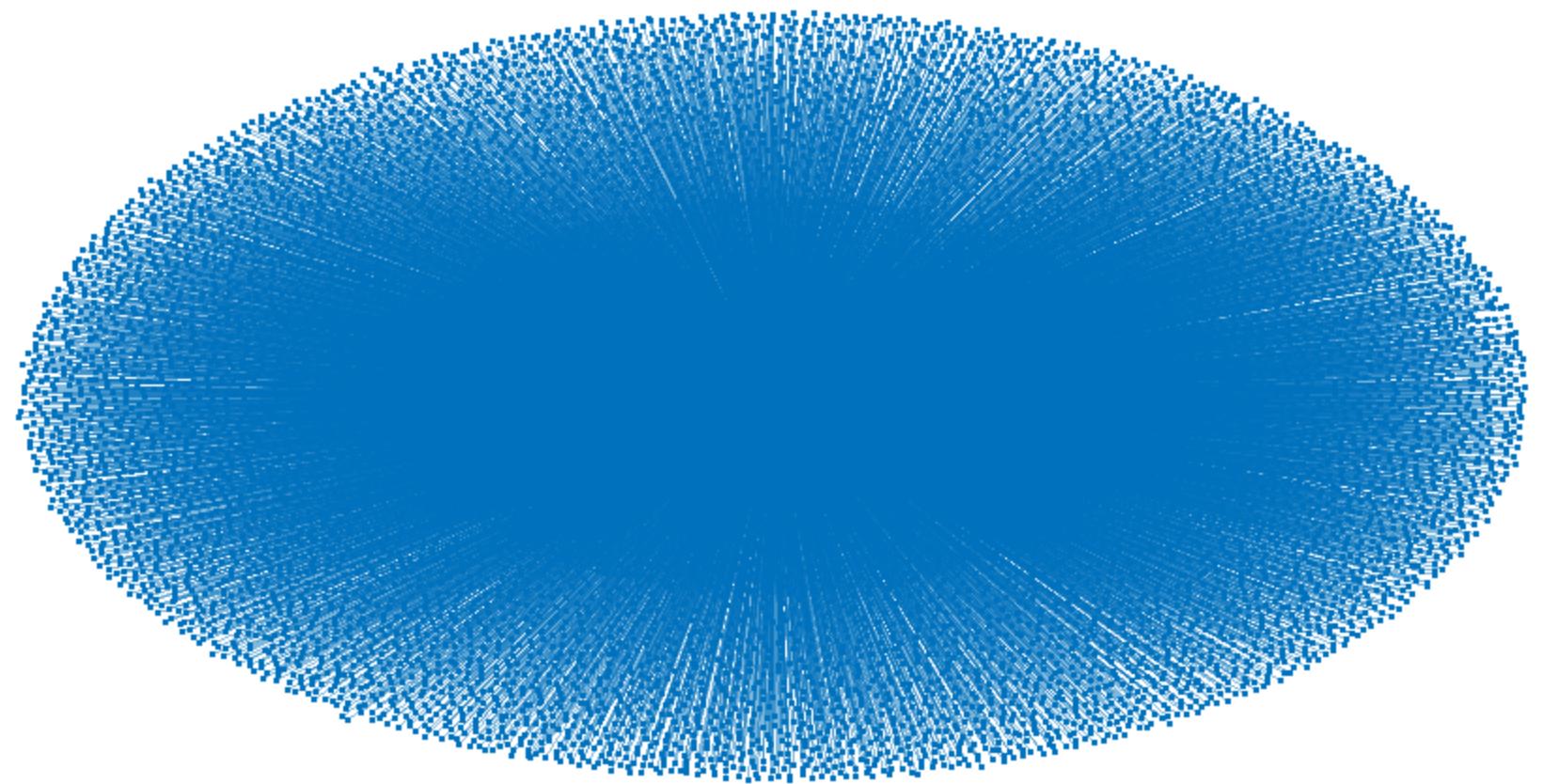




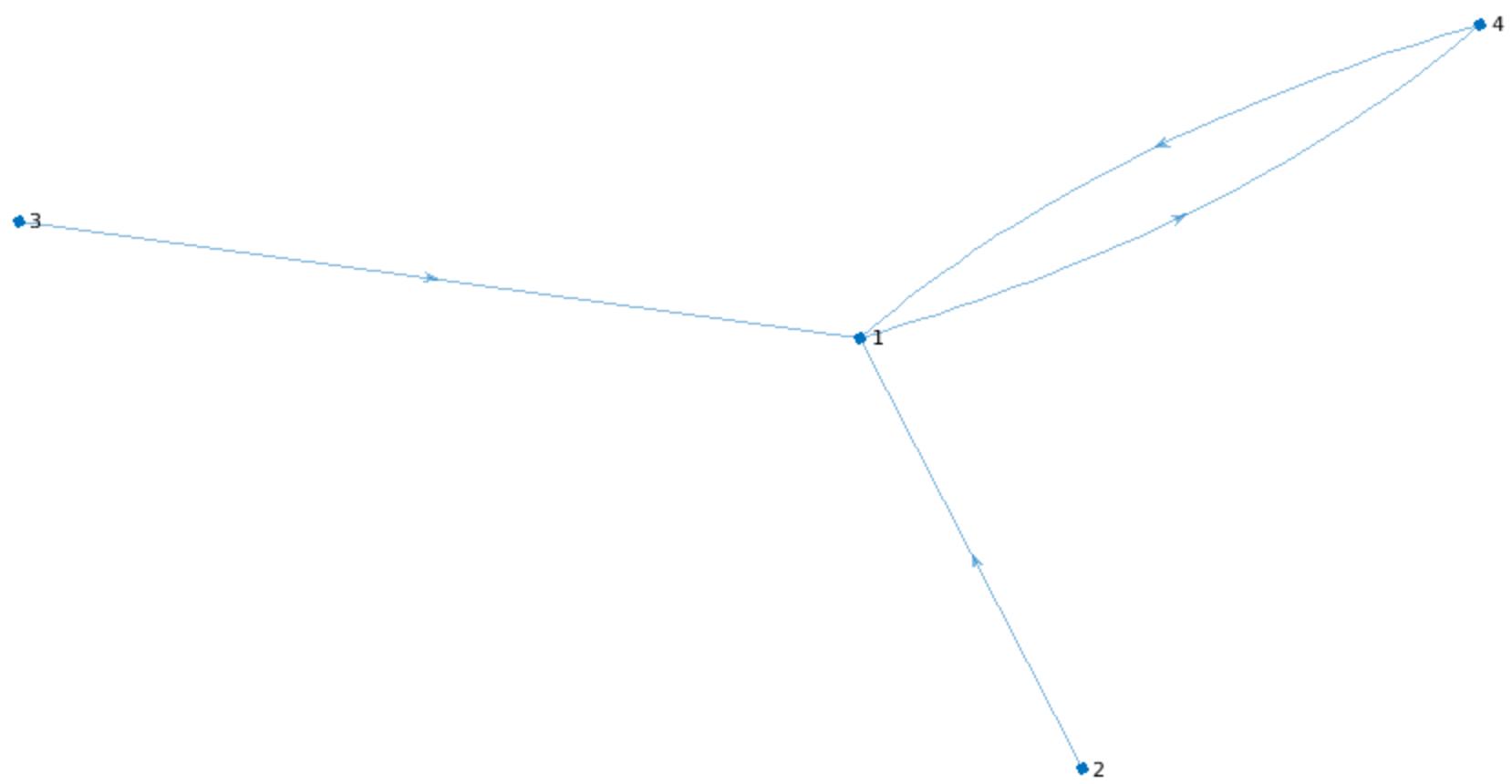


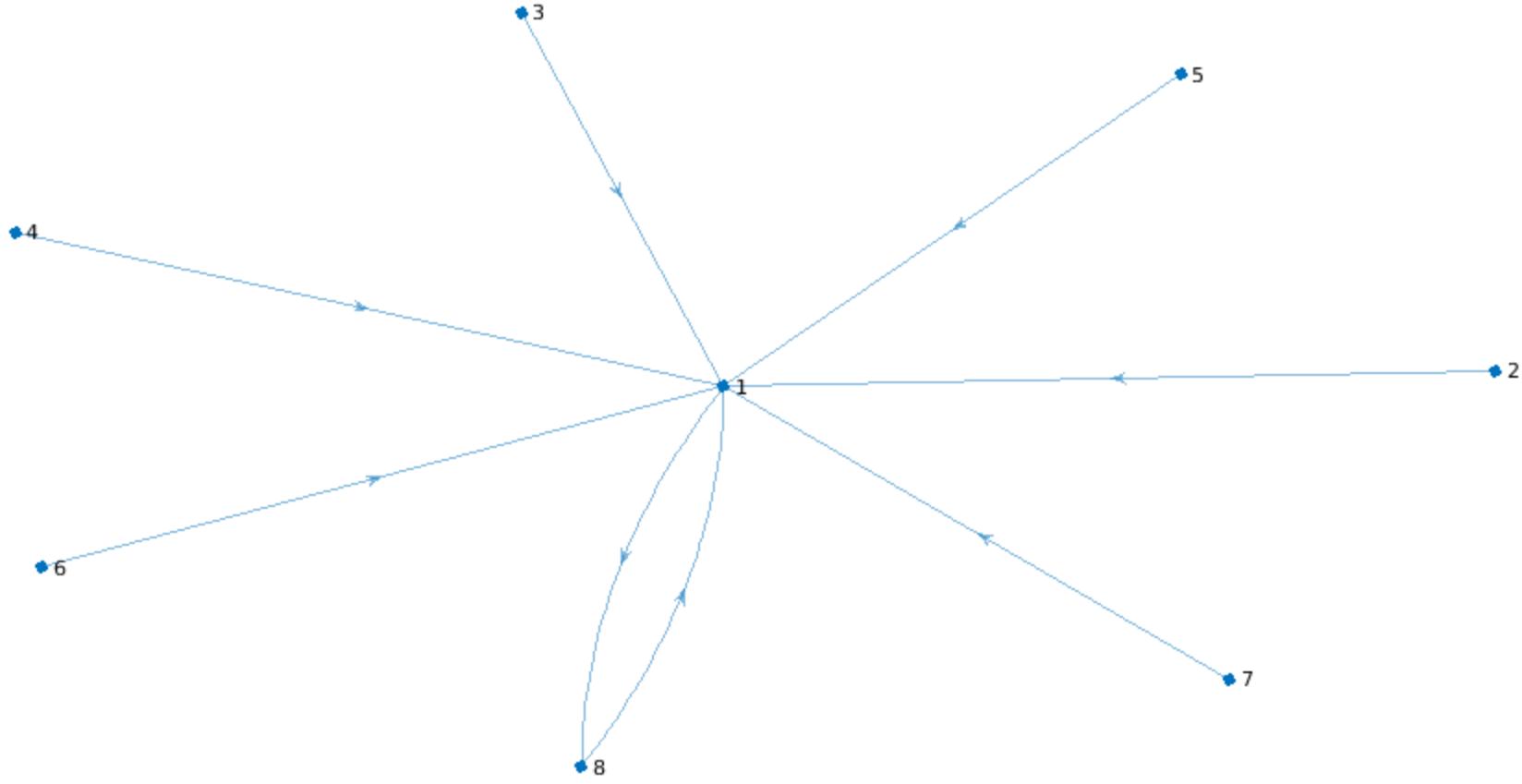


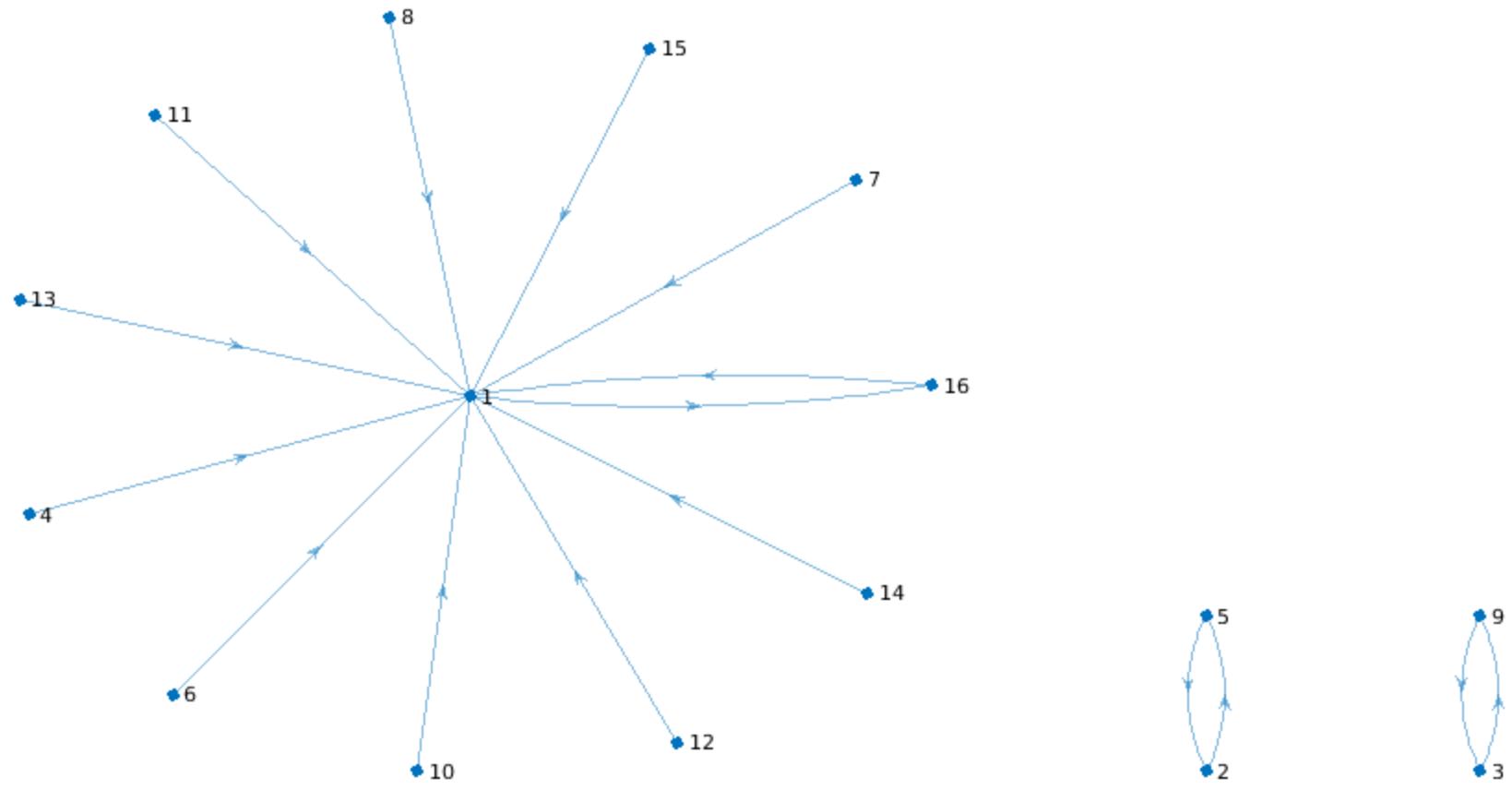


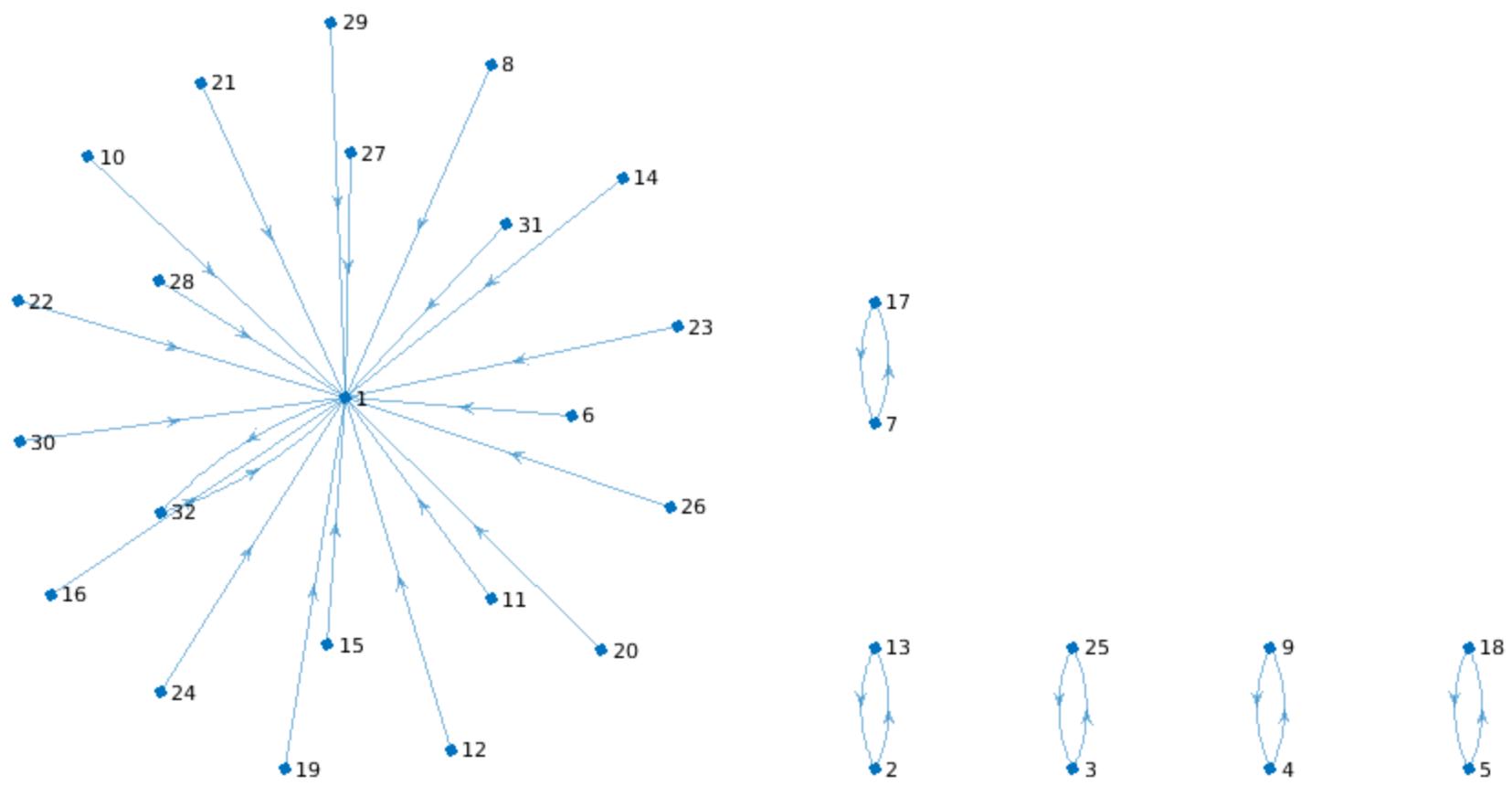


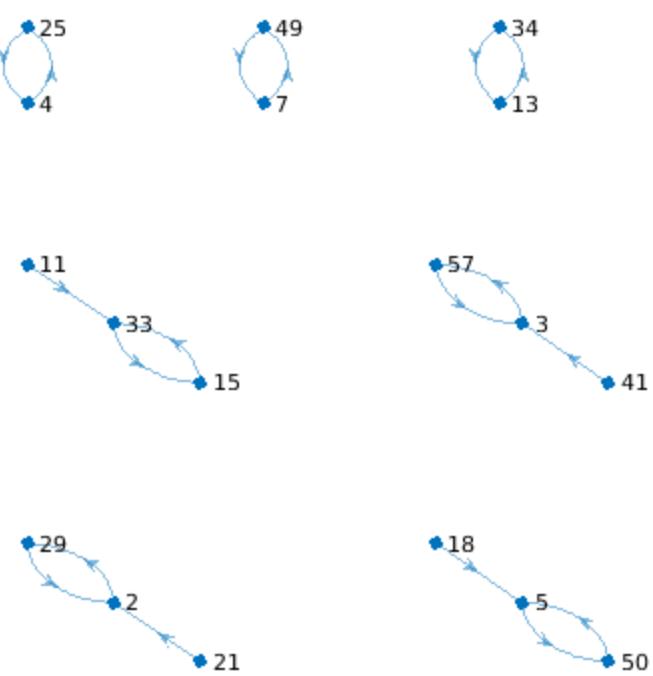
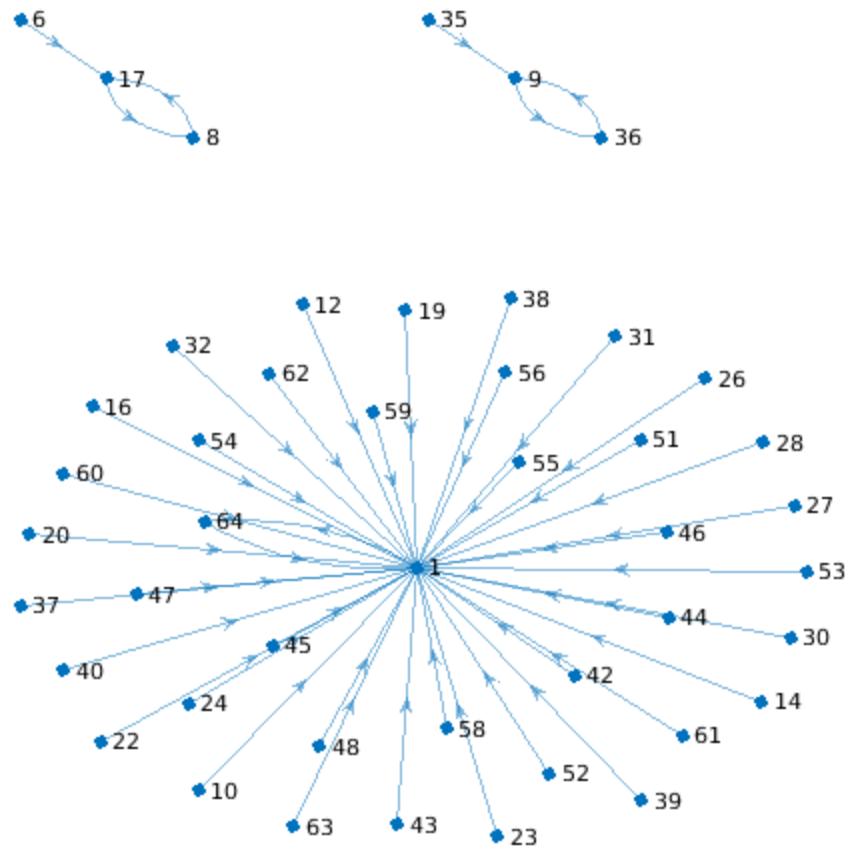
1

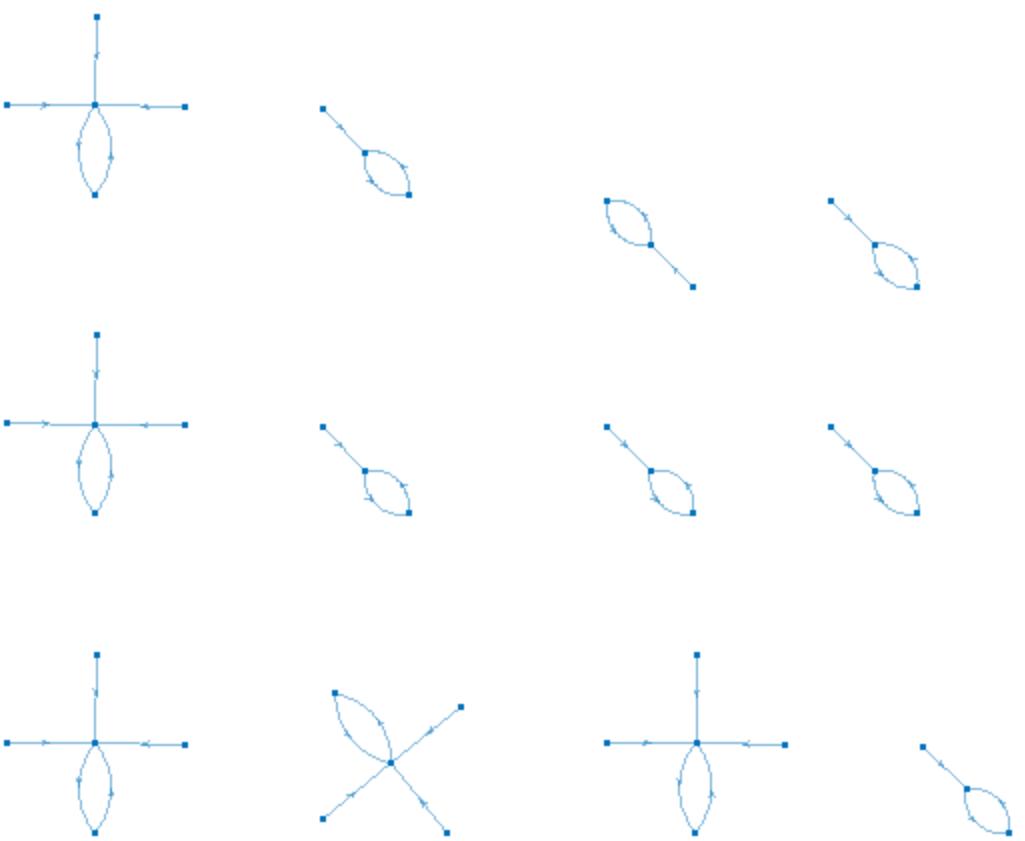
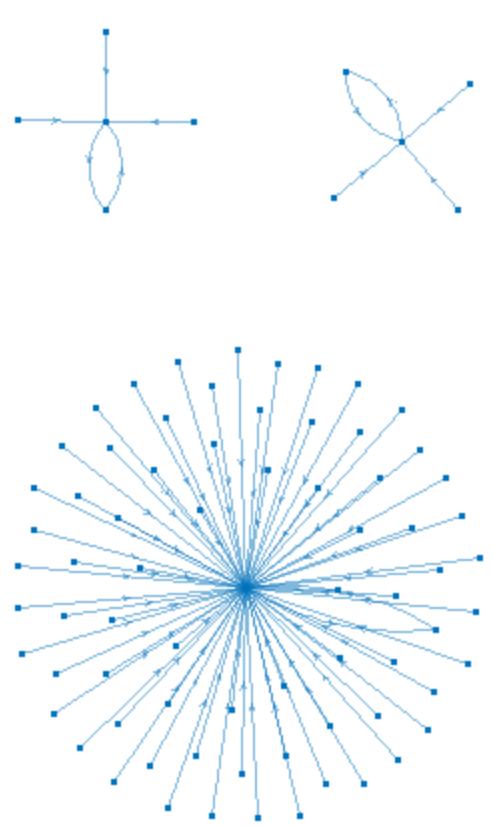


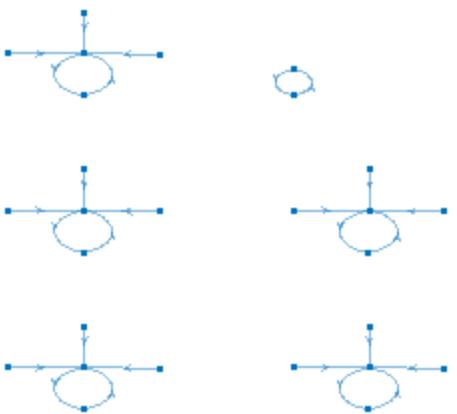
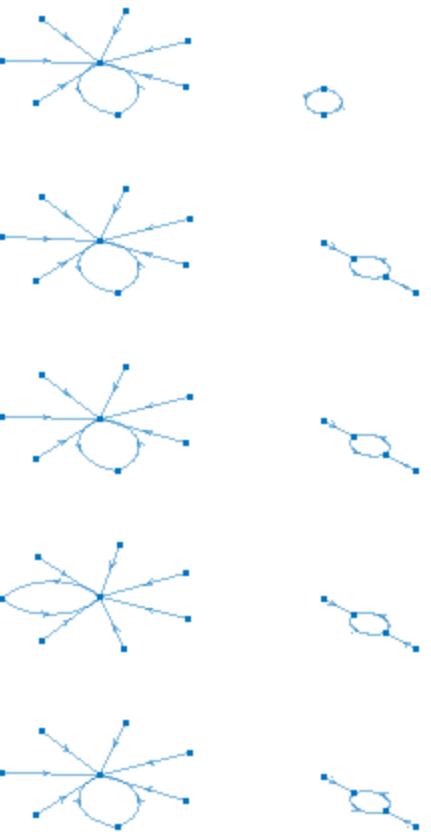
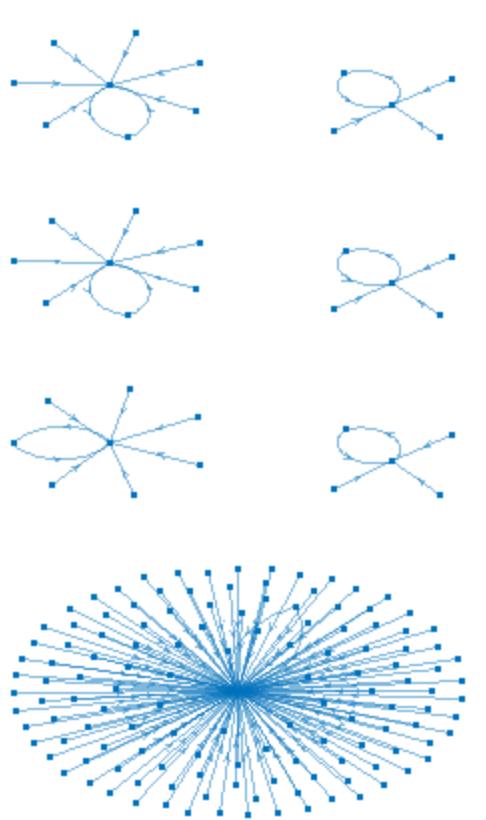


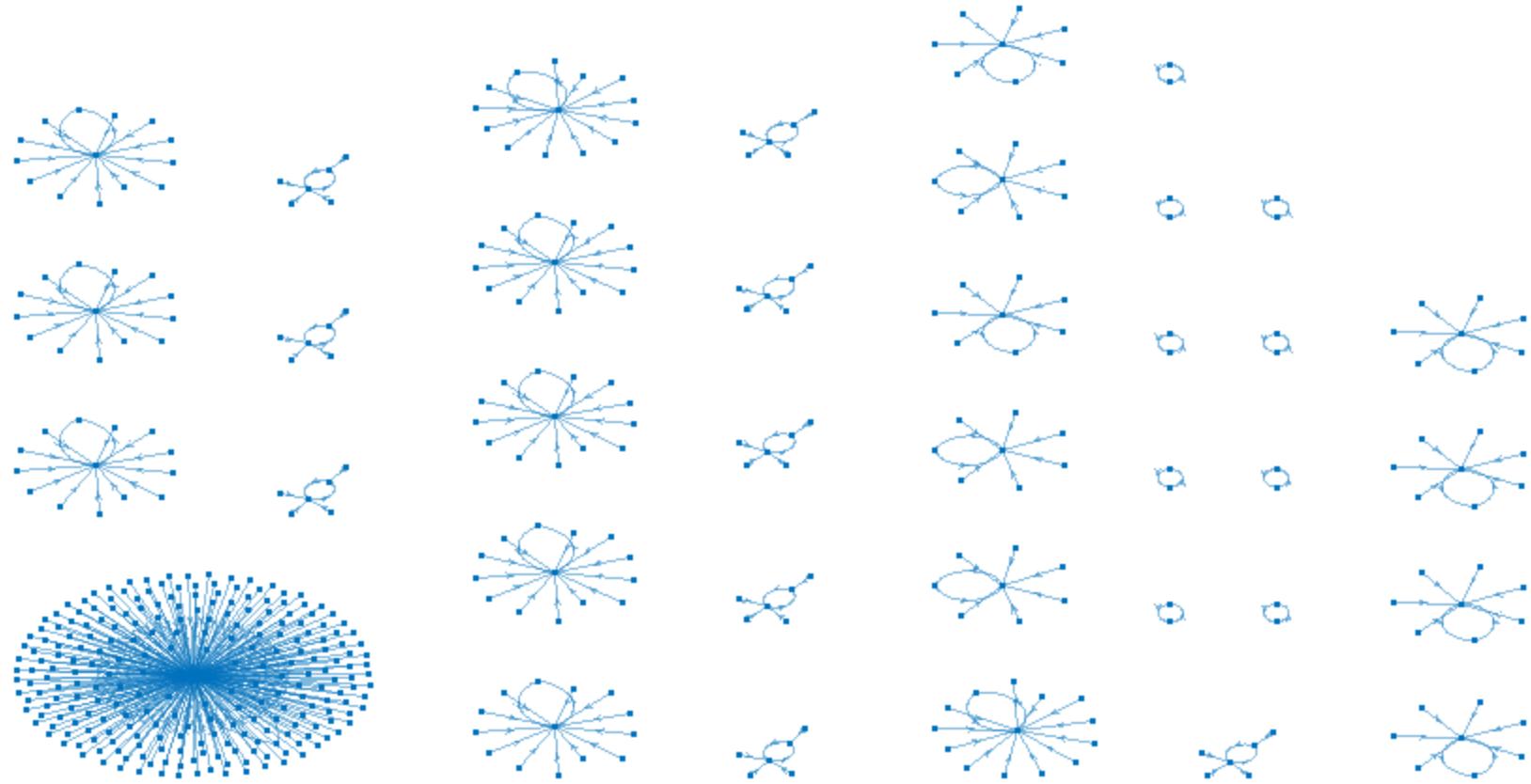


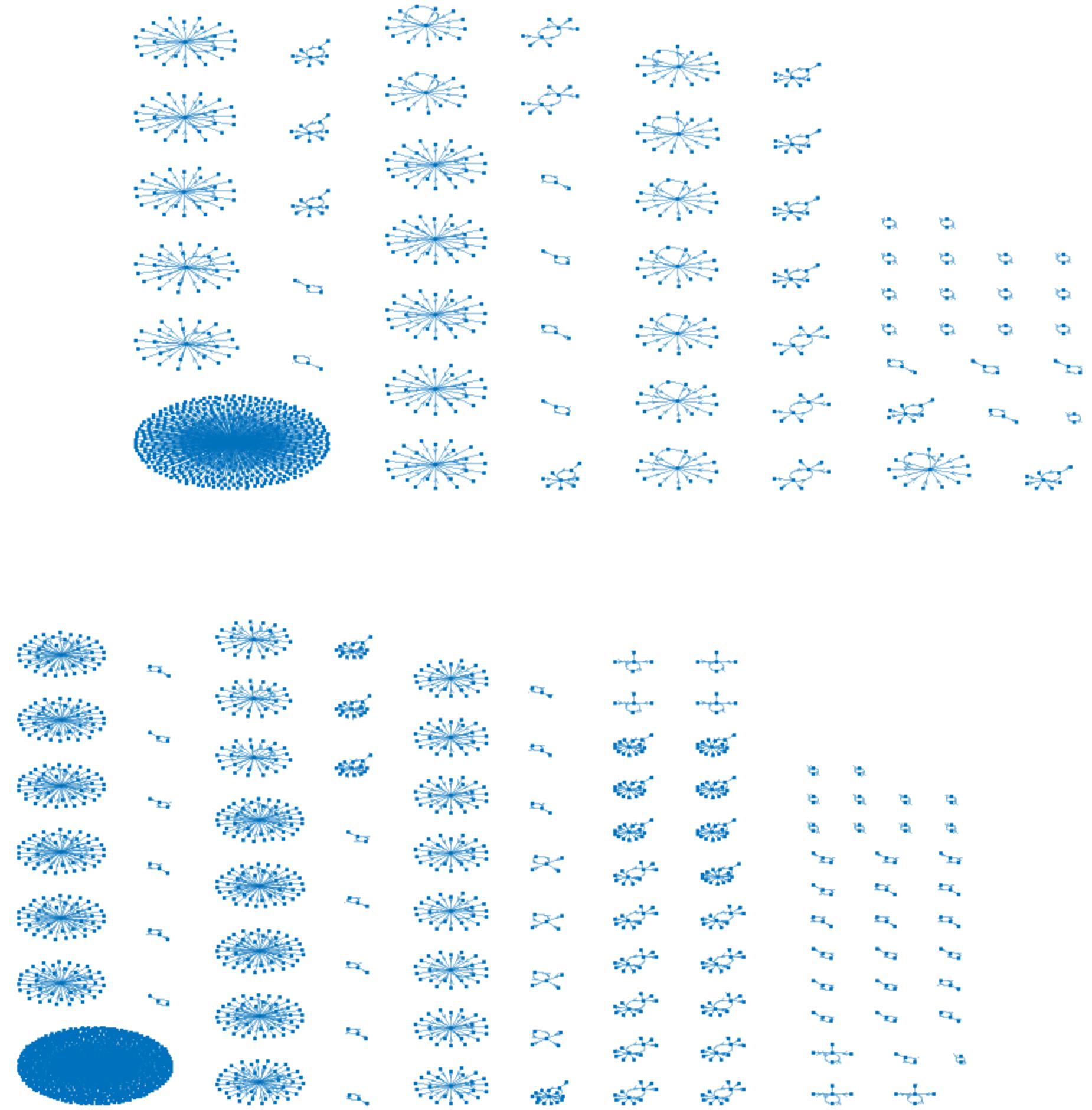


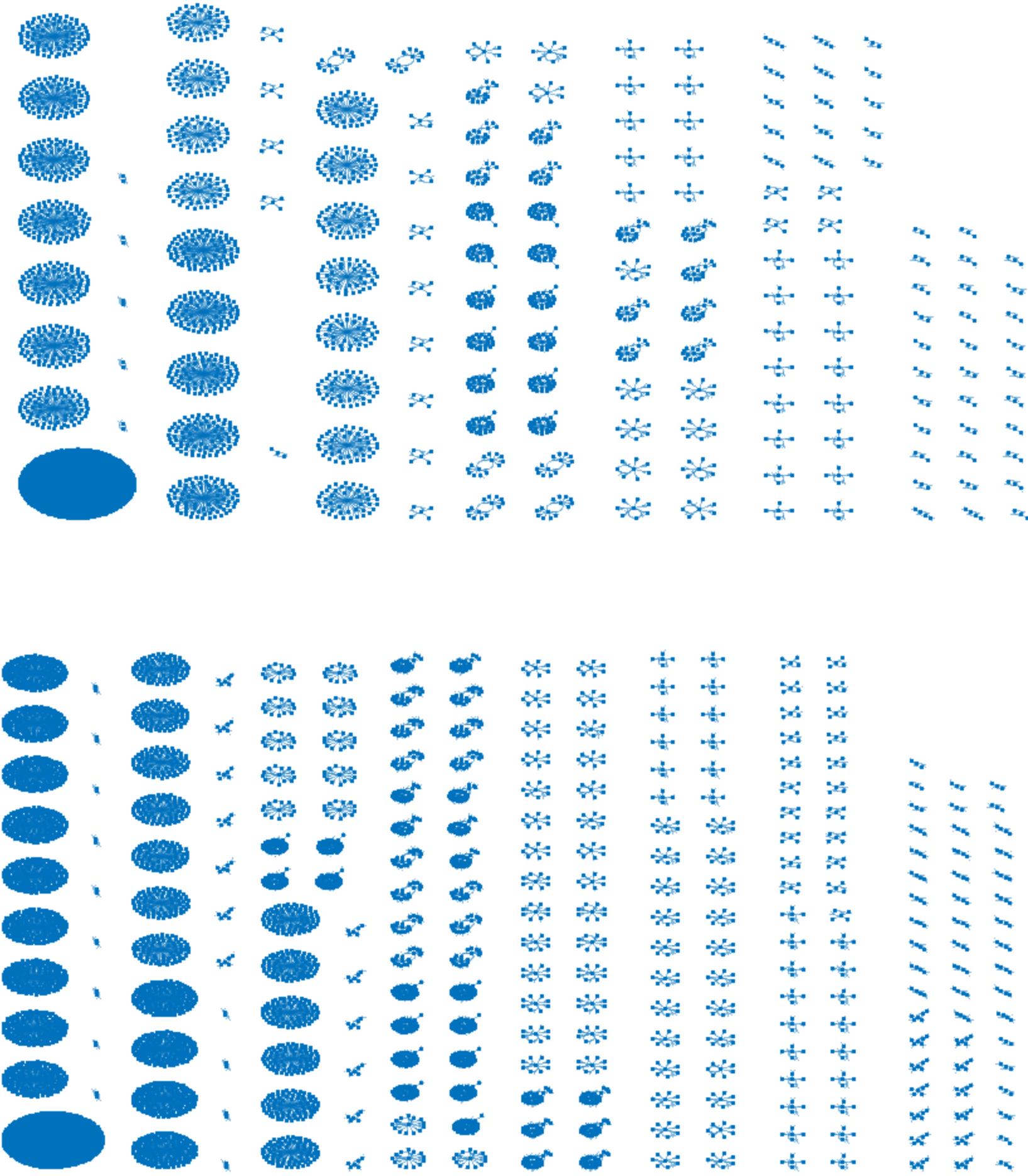




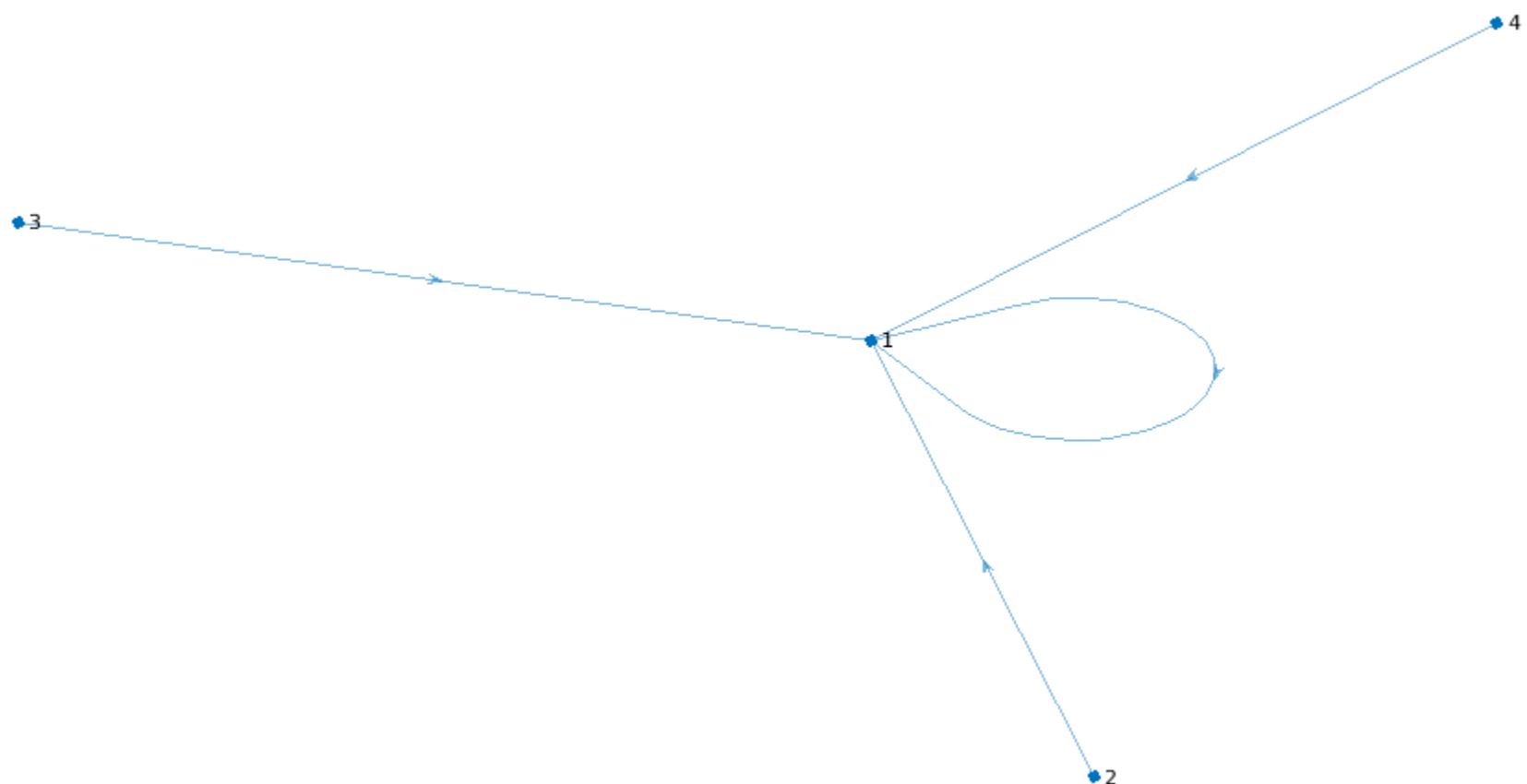


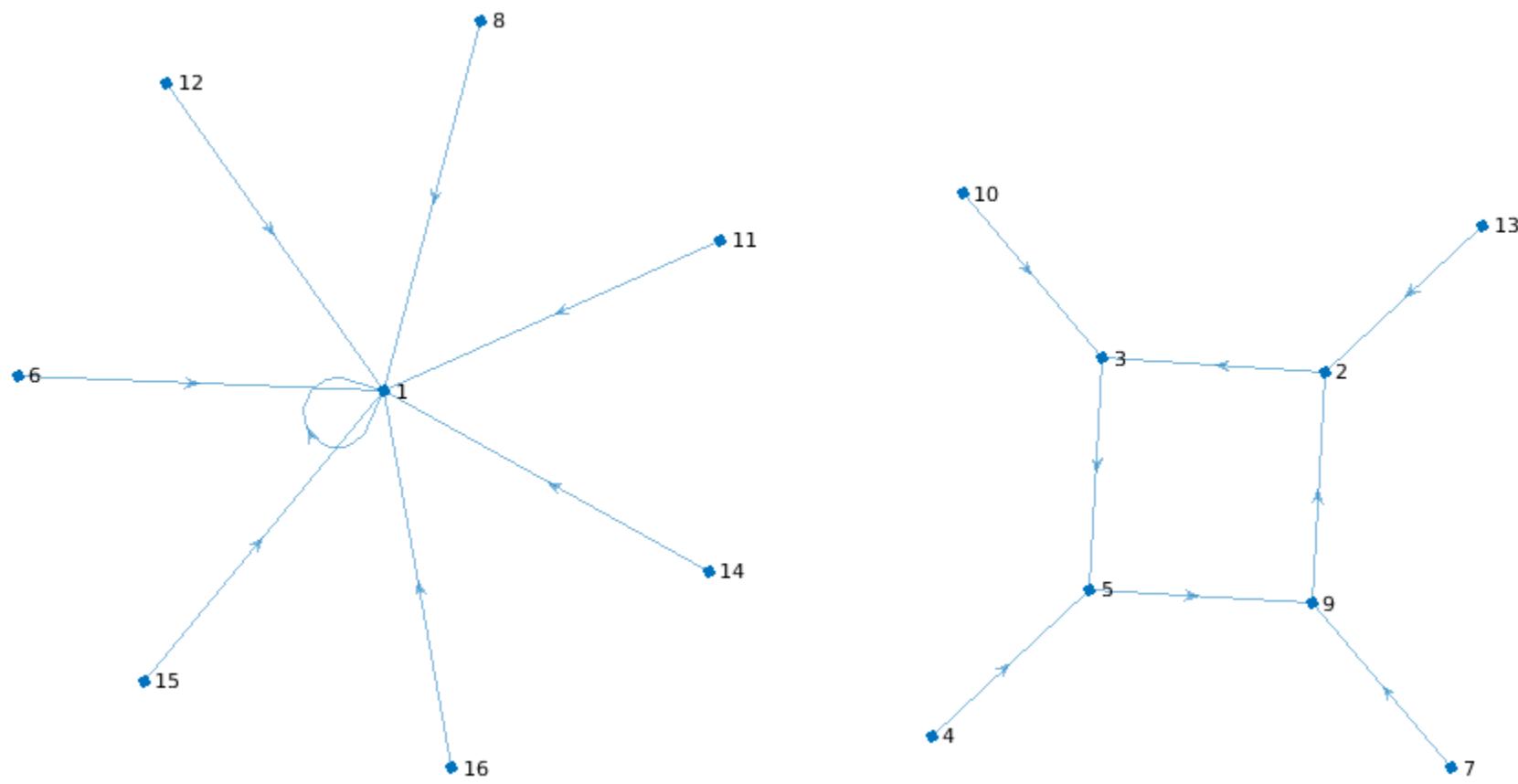
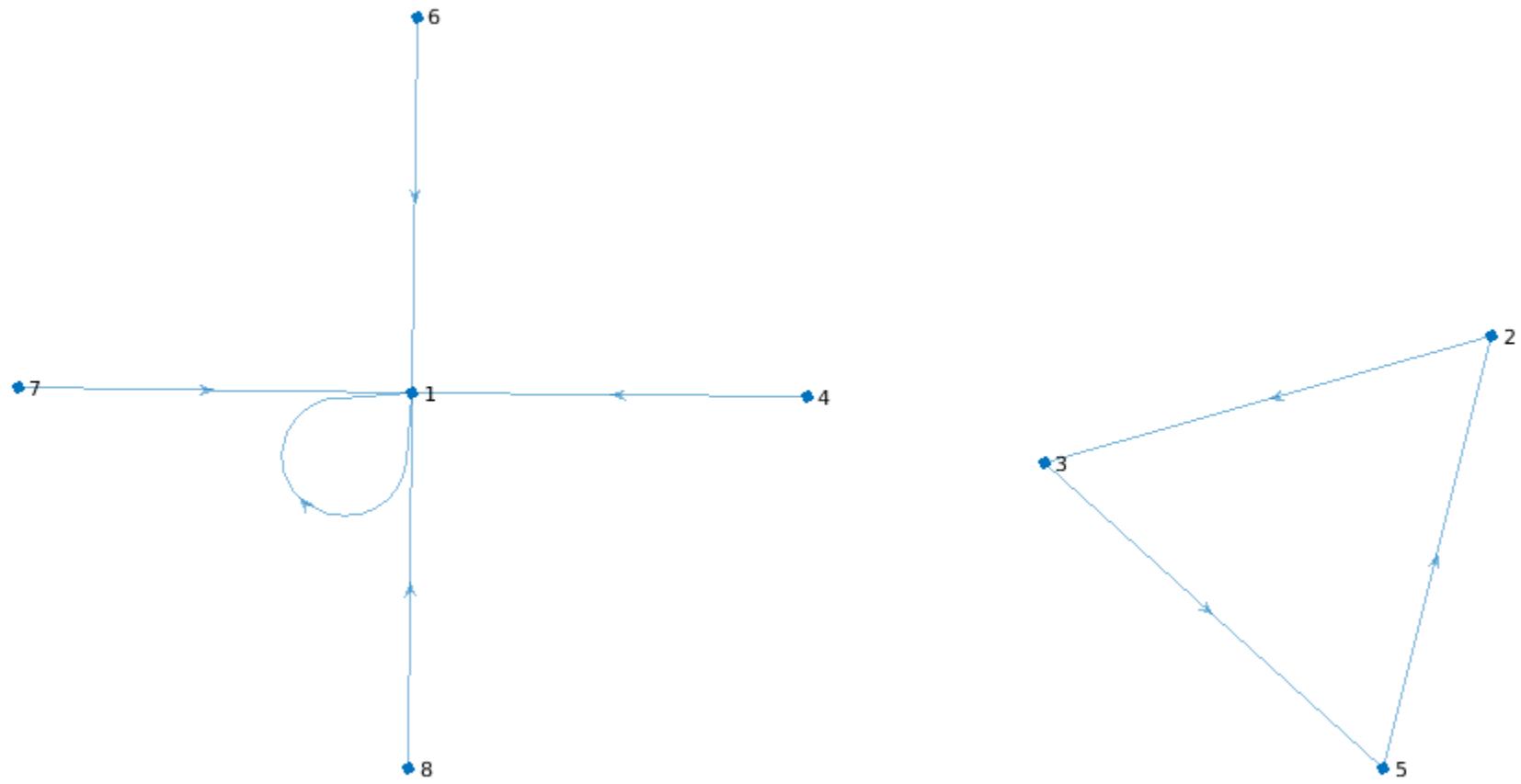


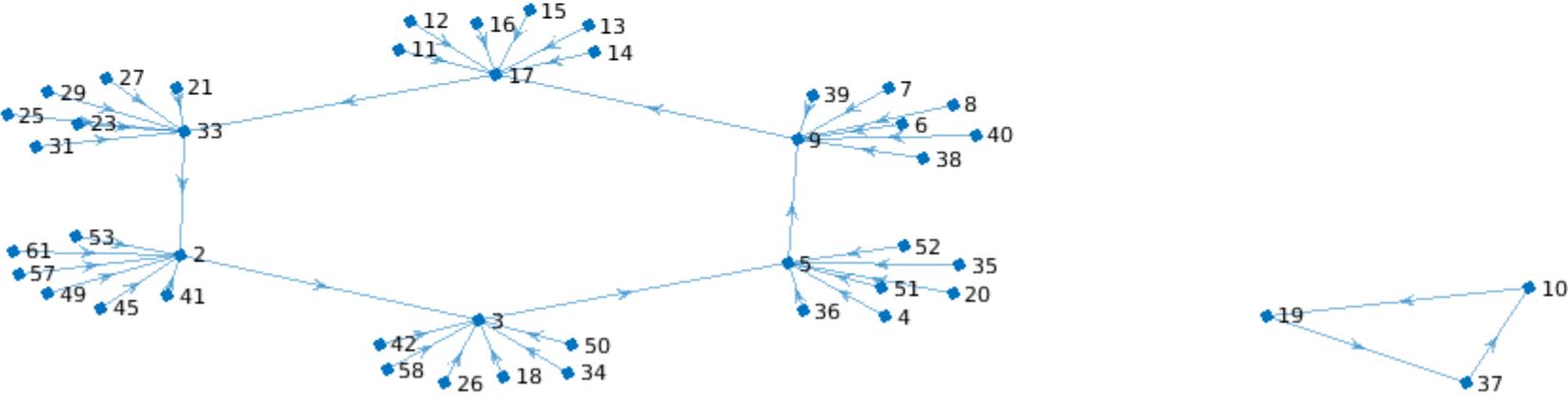
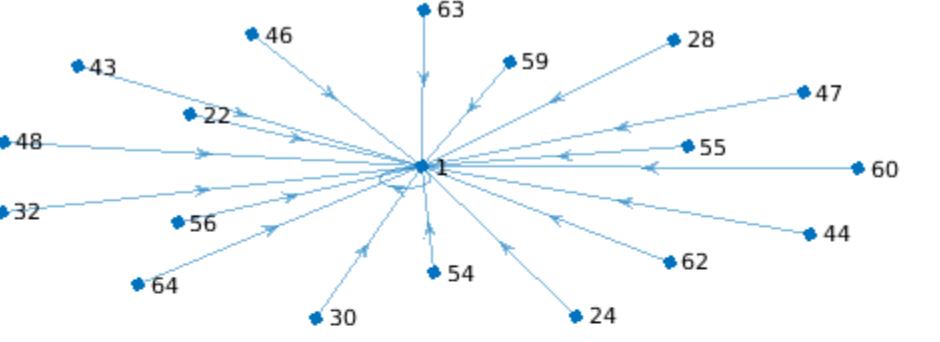
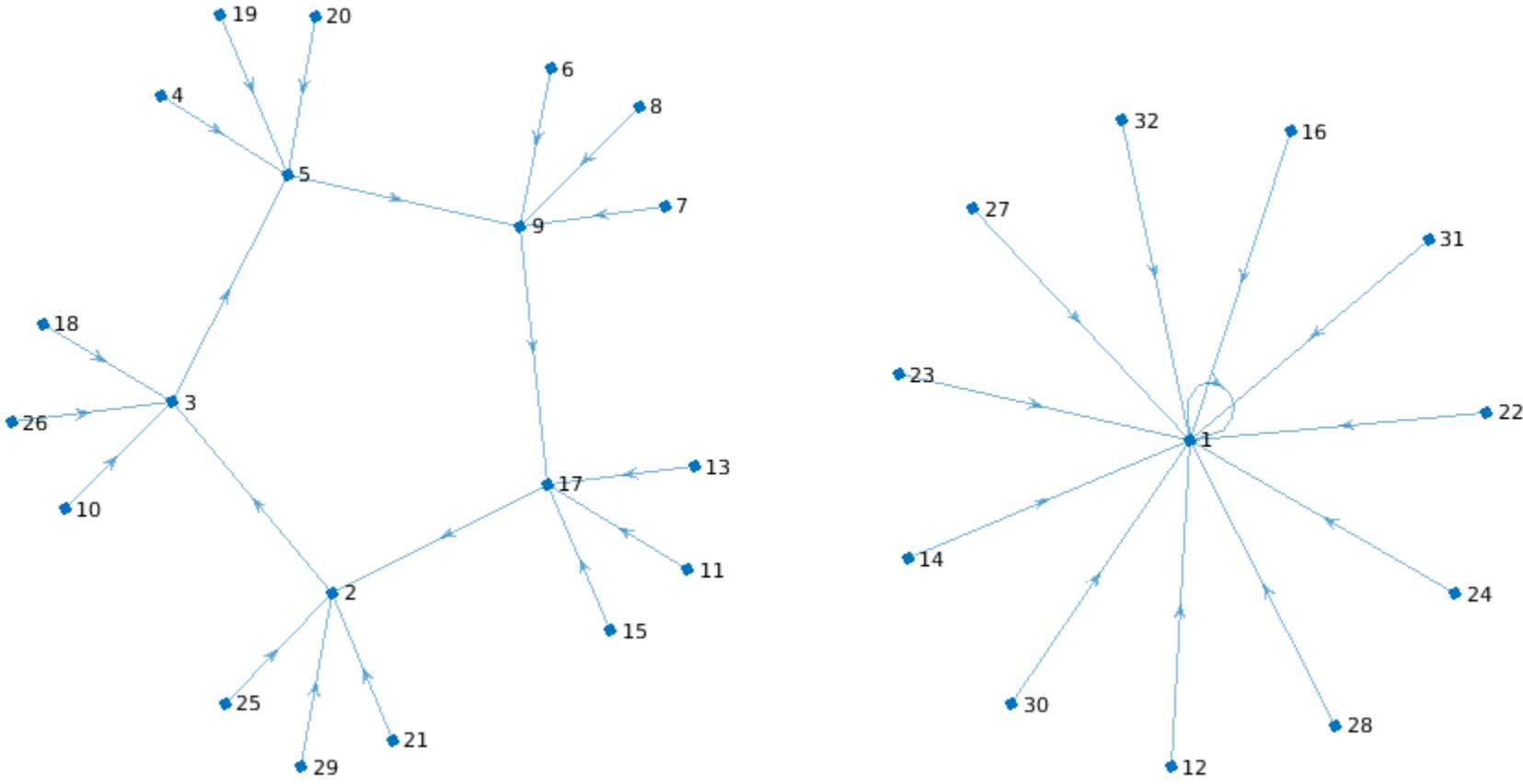


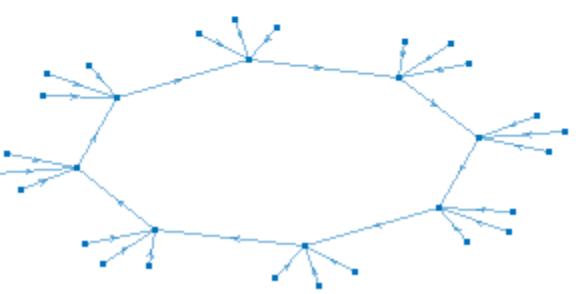
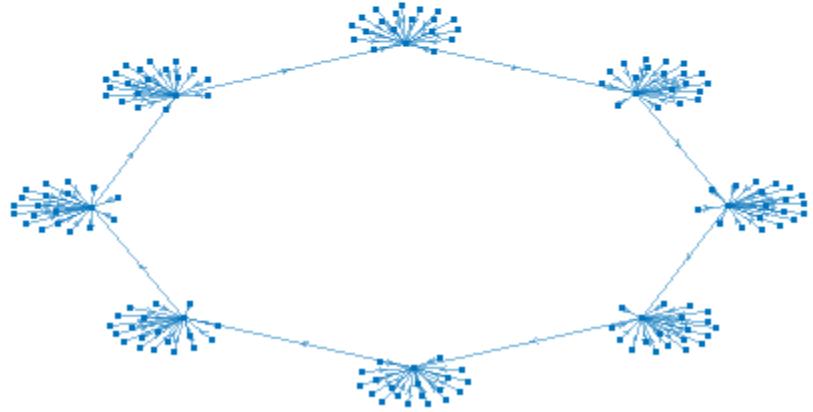
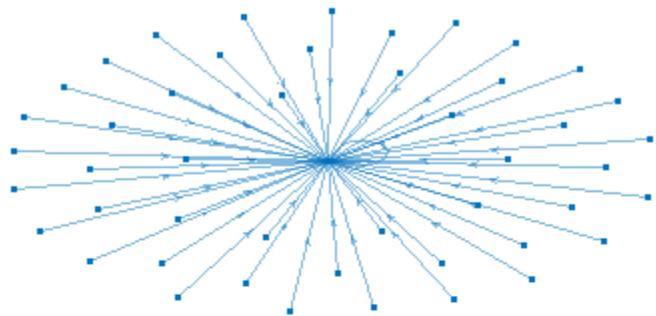
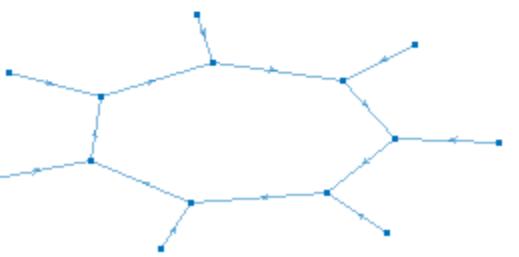
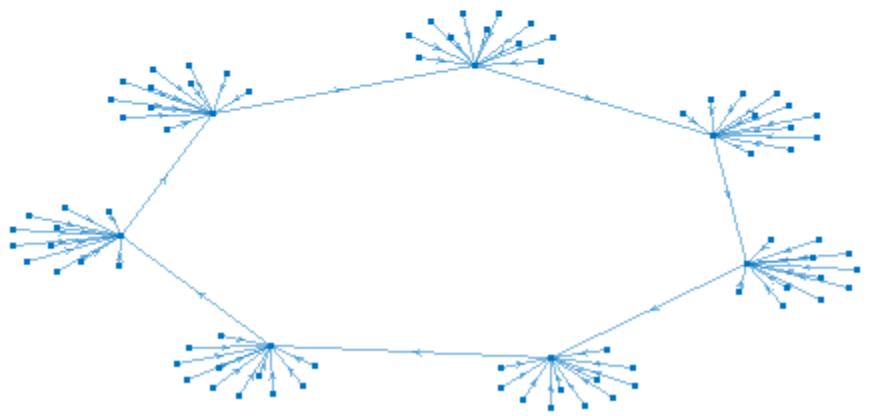
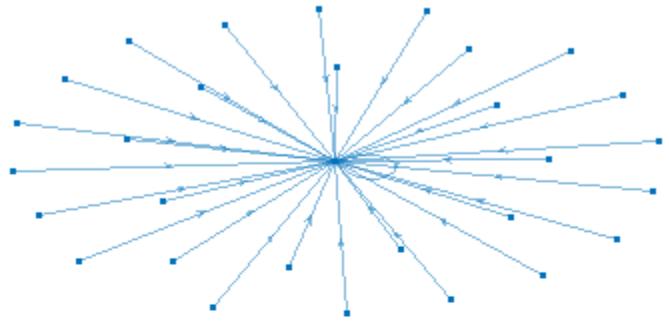


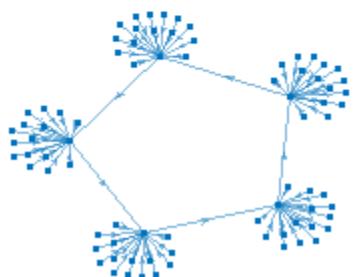
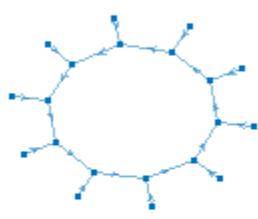
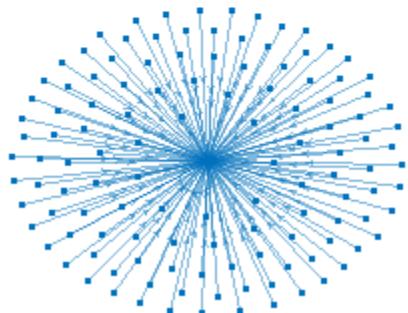
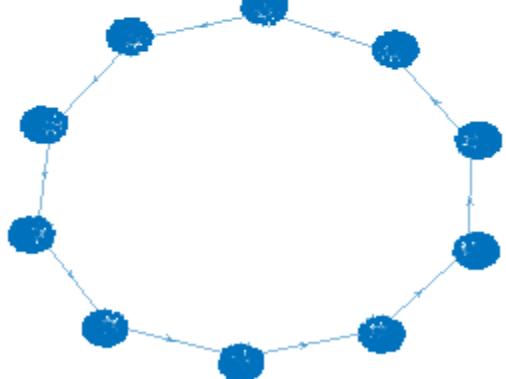
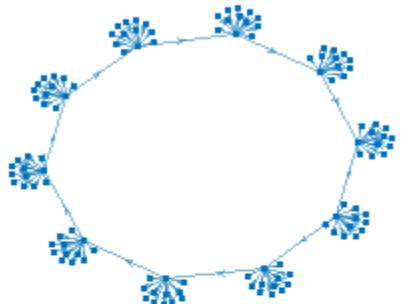
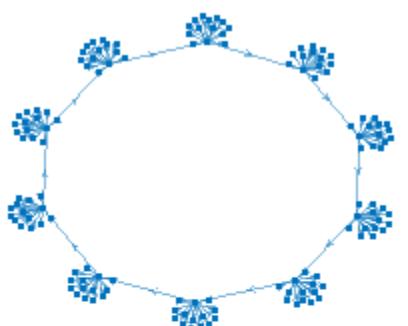
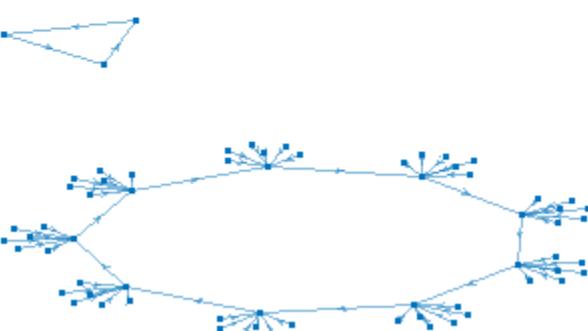
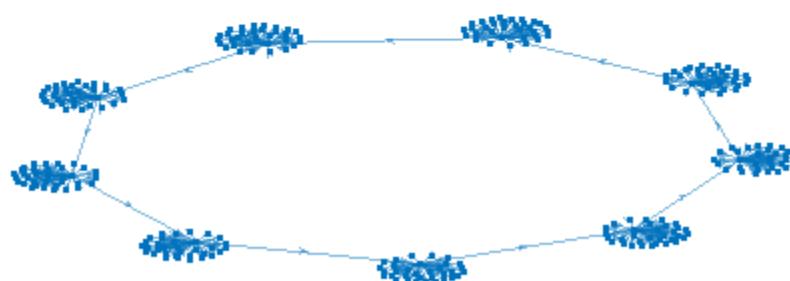
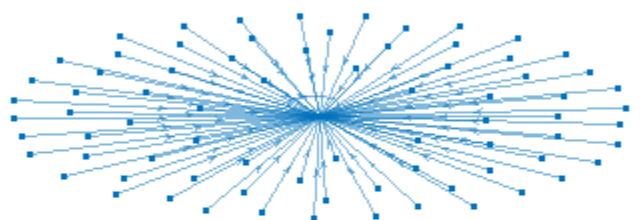
2

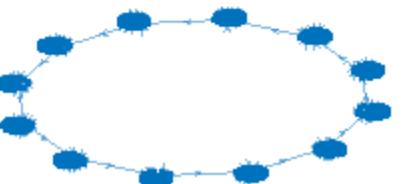
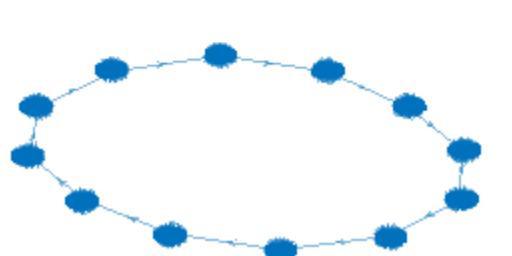
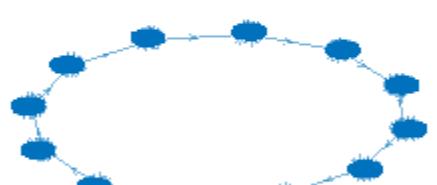
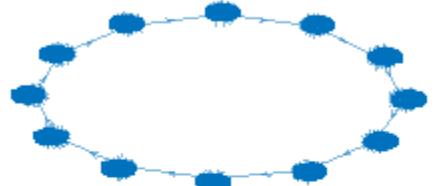
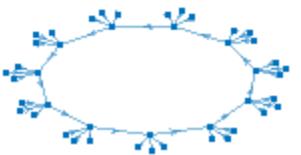
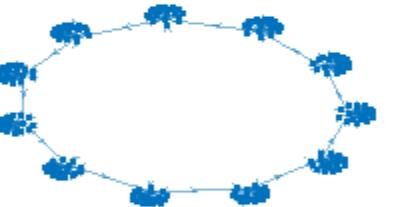
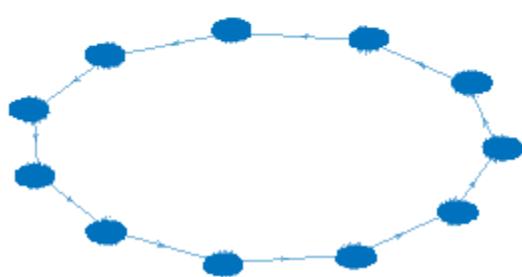
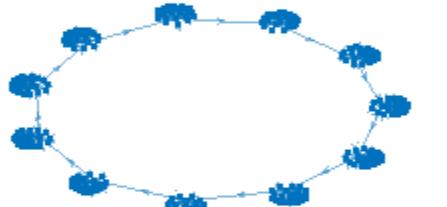
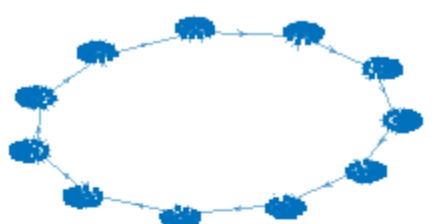


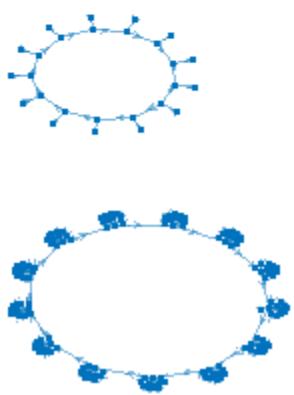
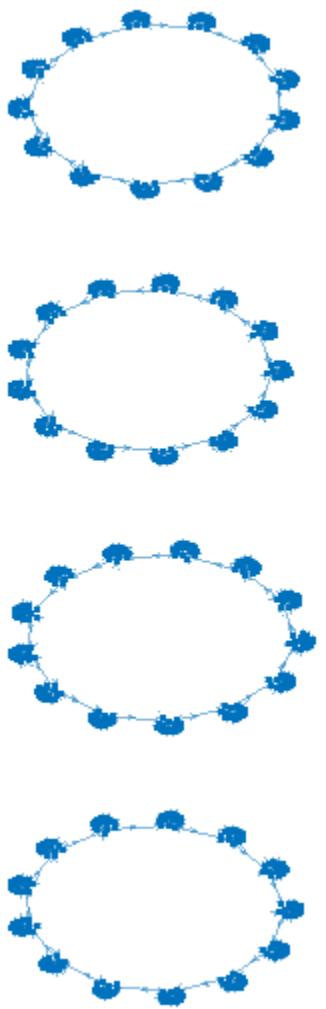
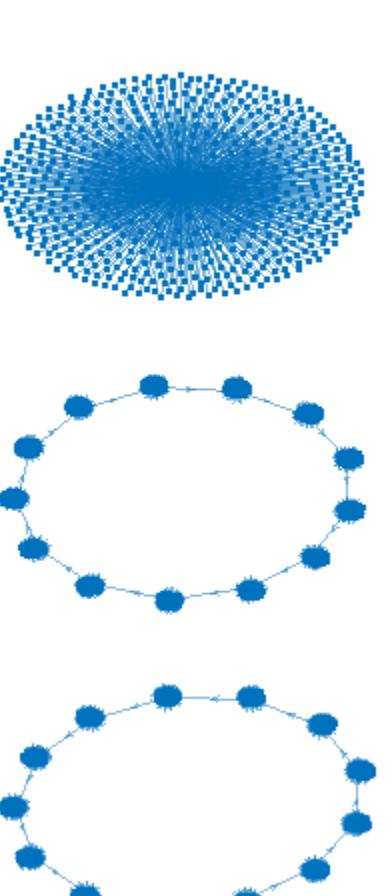
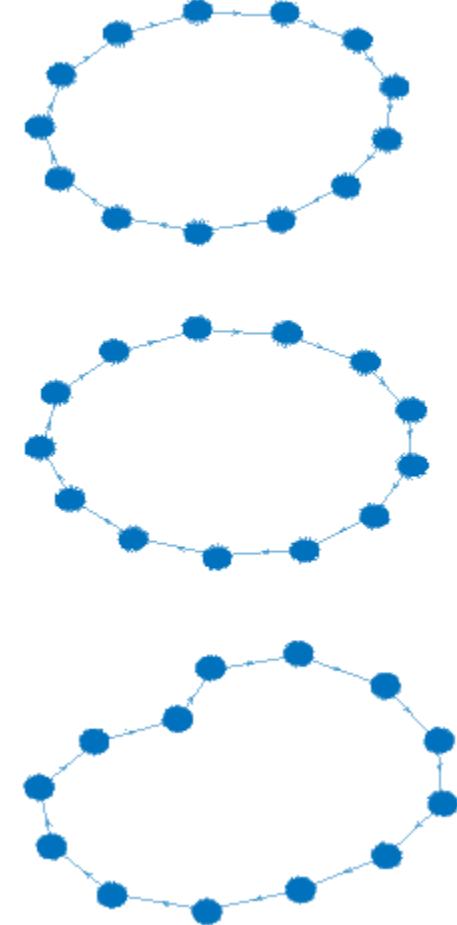




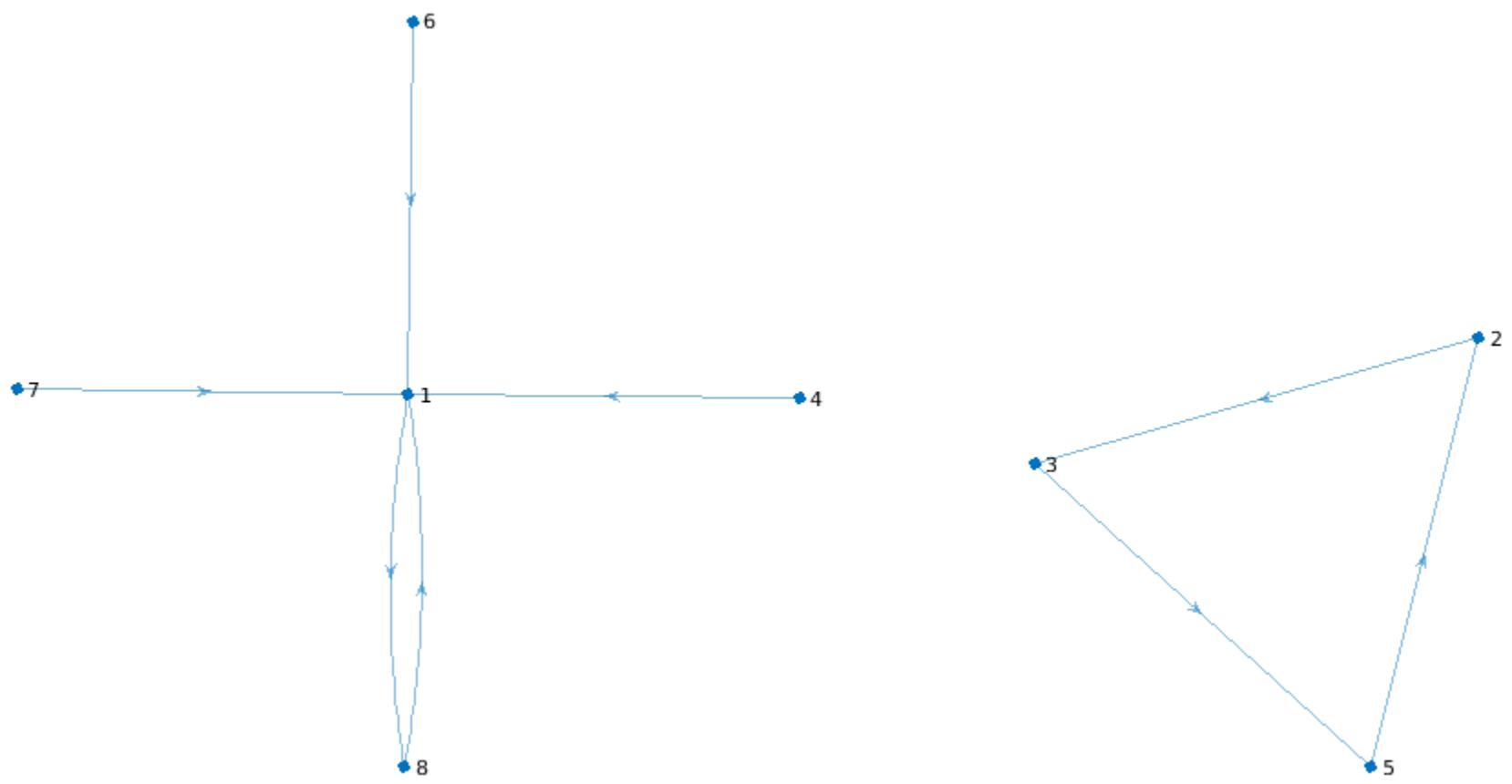
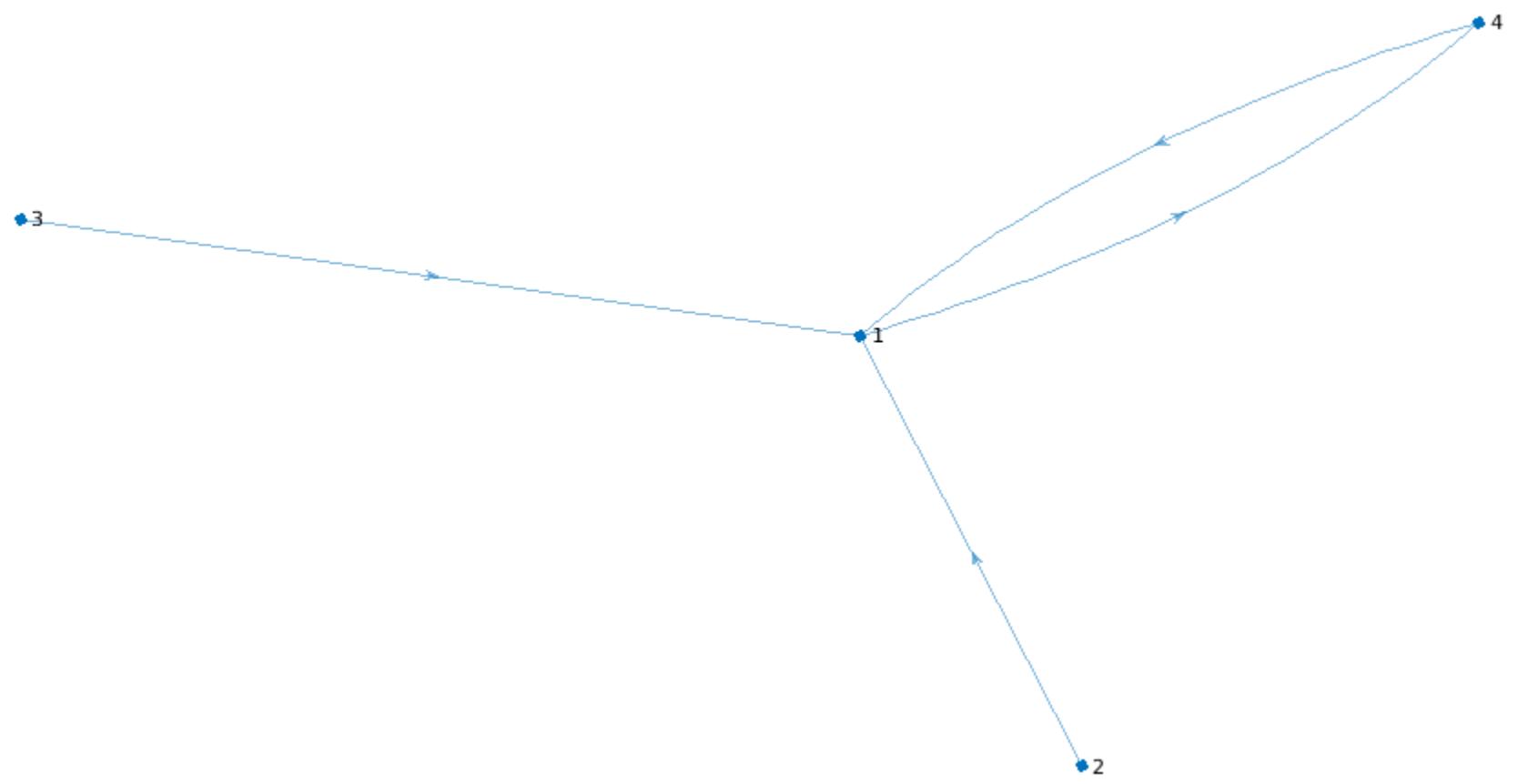


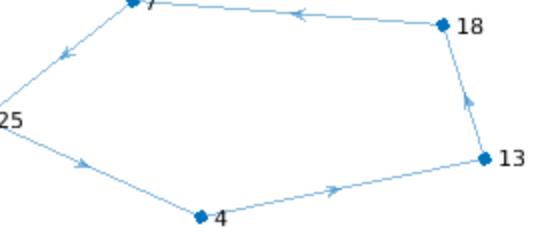
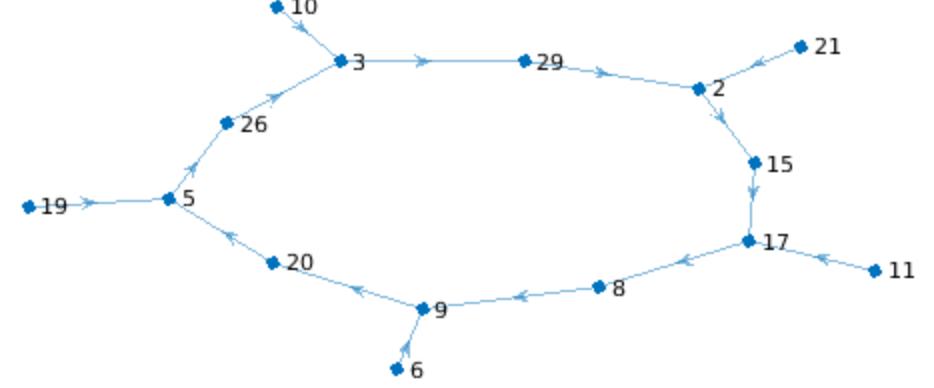
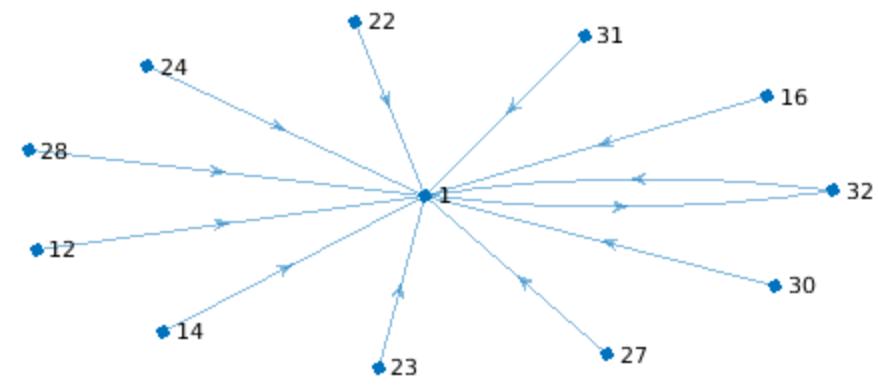
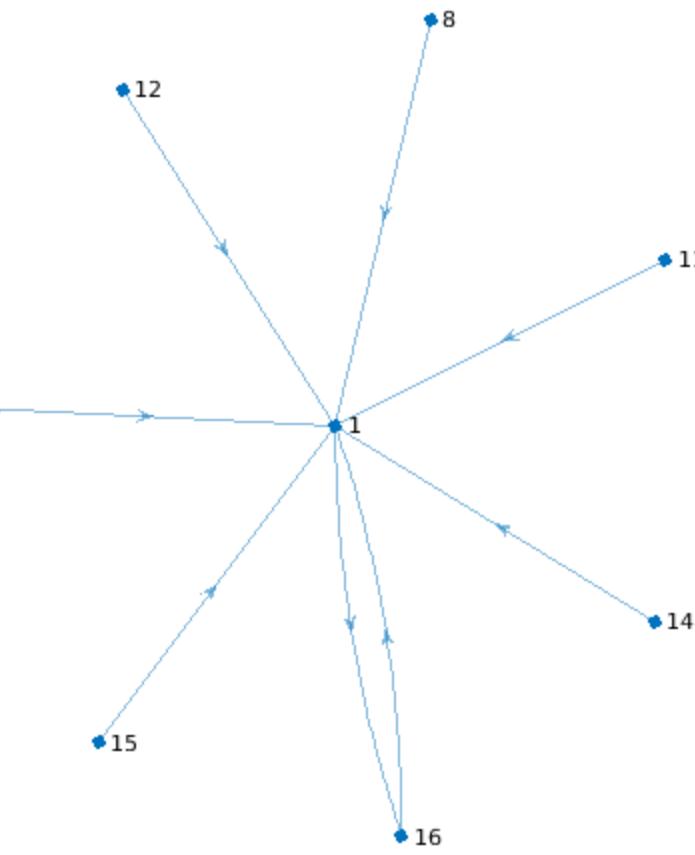
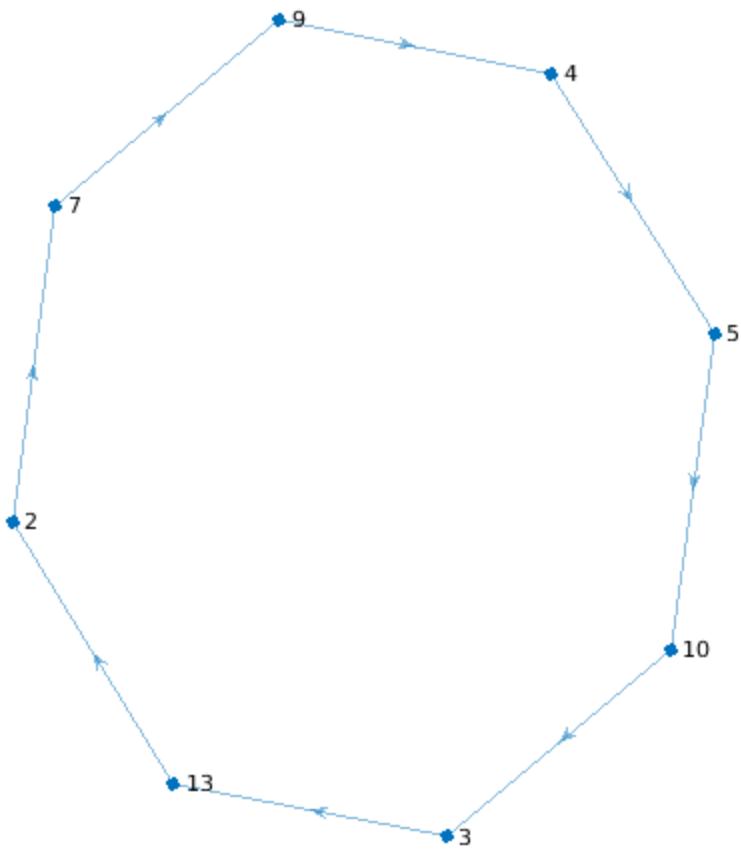


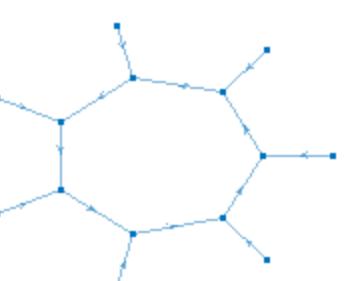
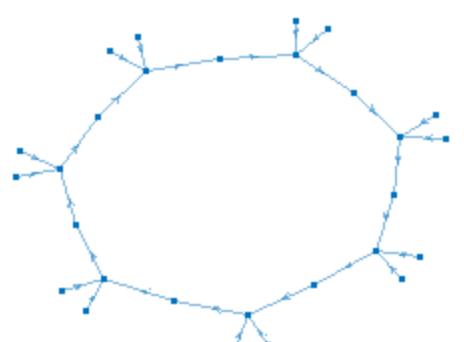
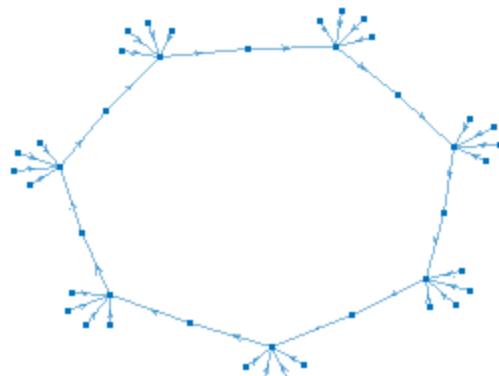
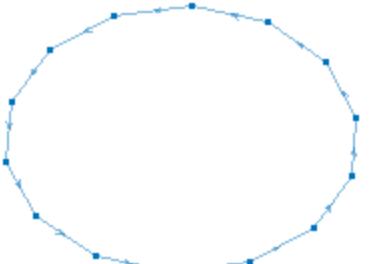
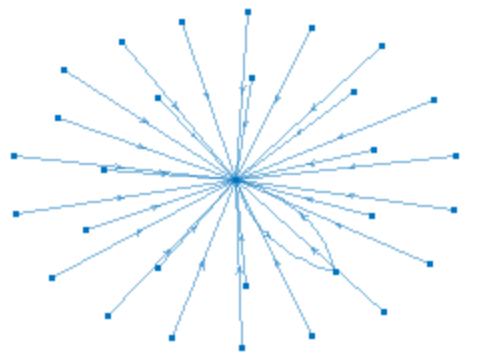
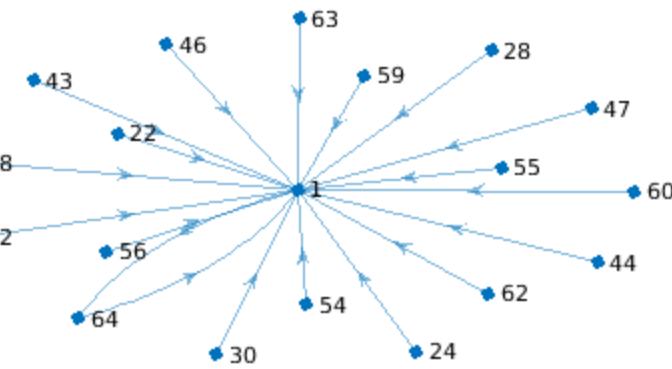
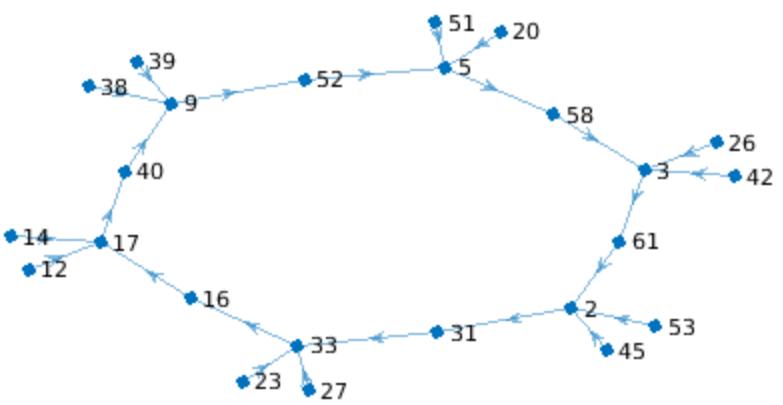
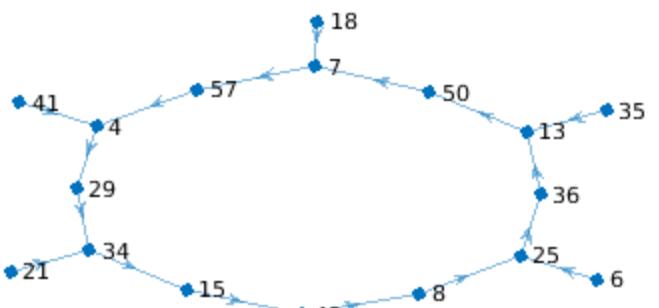


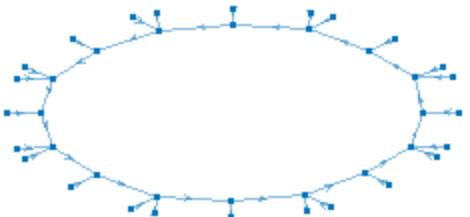
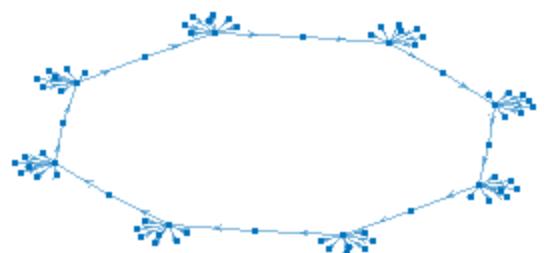
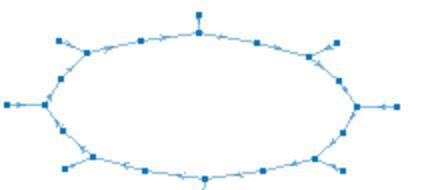
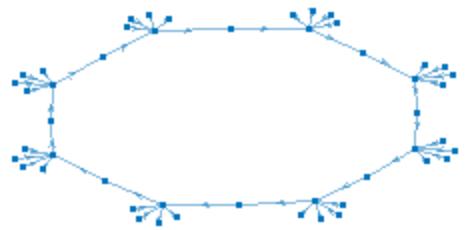


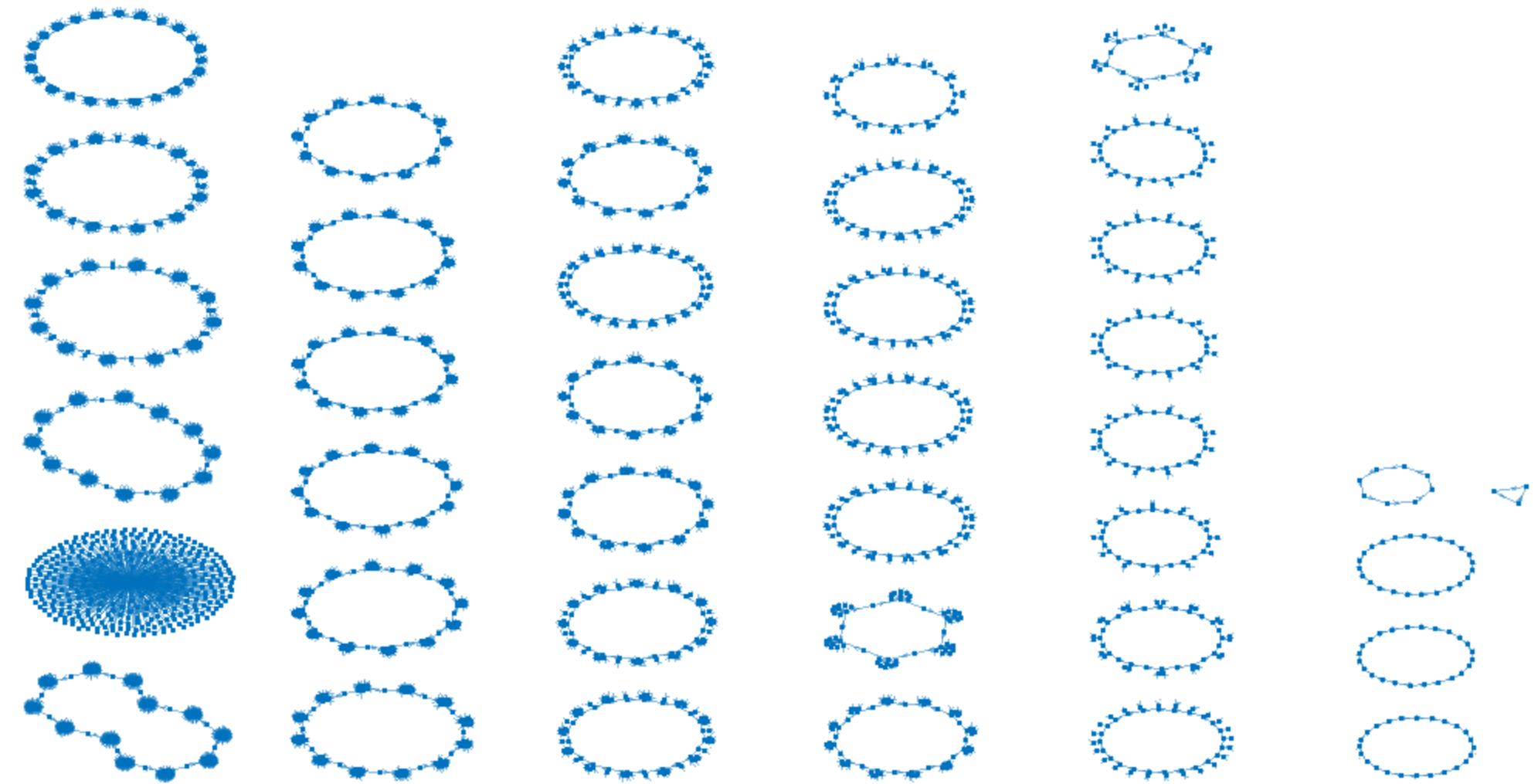
3











2000

1000

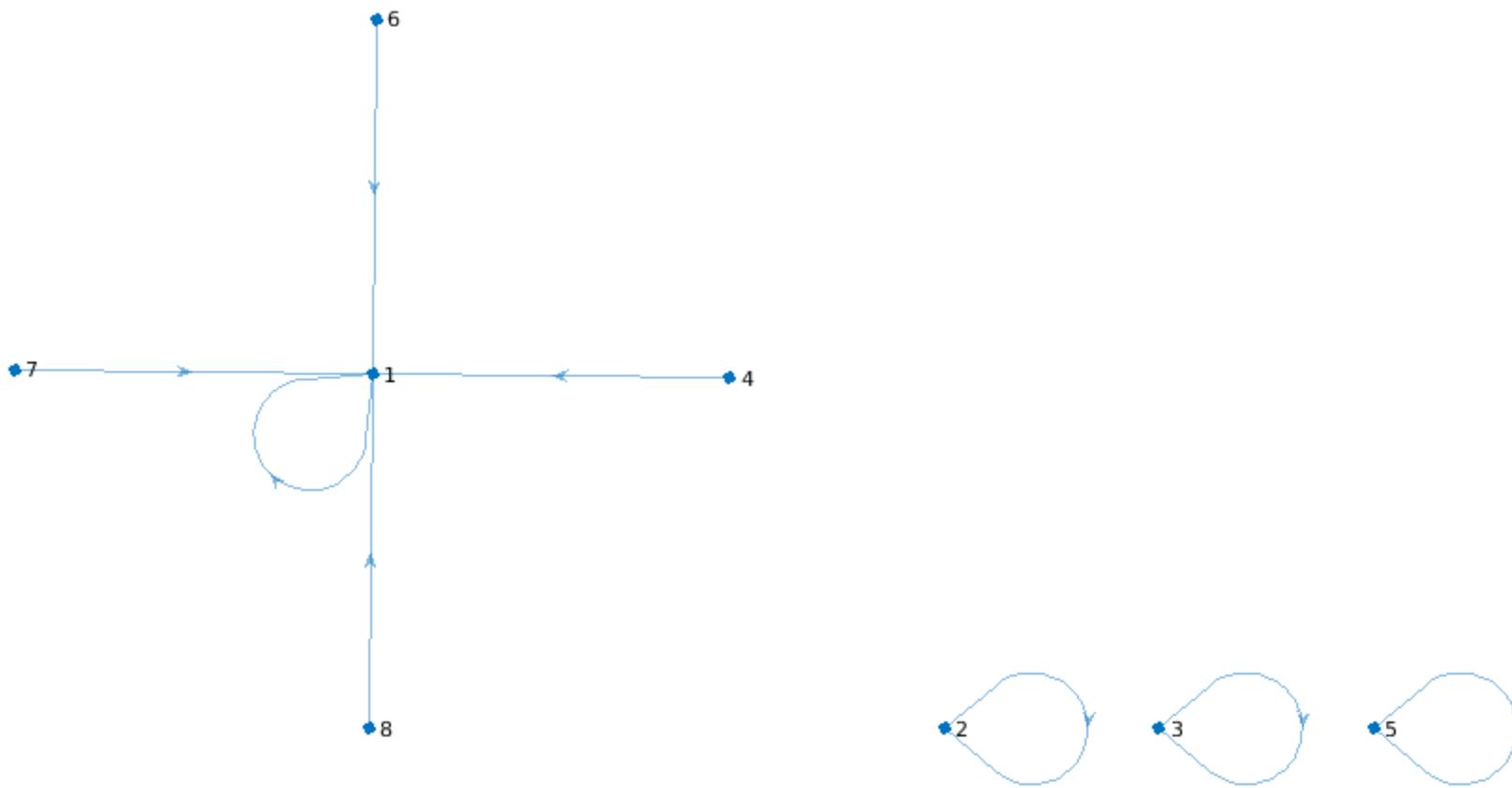
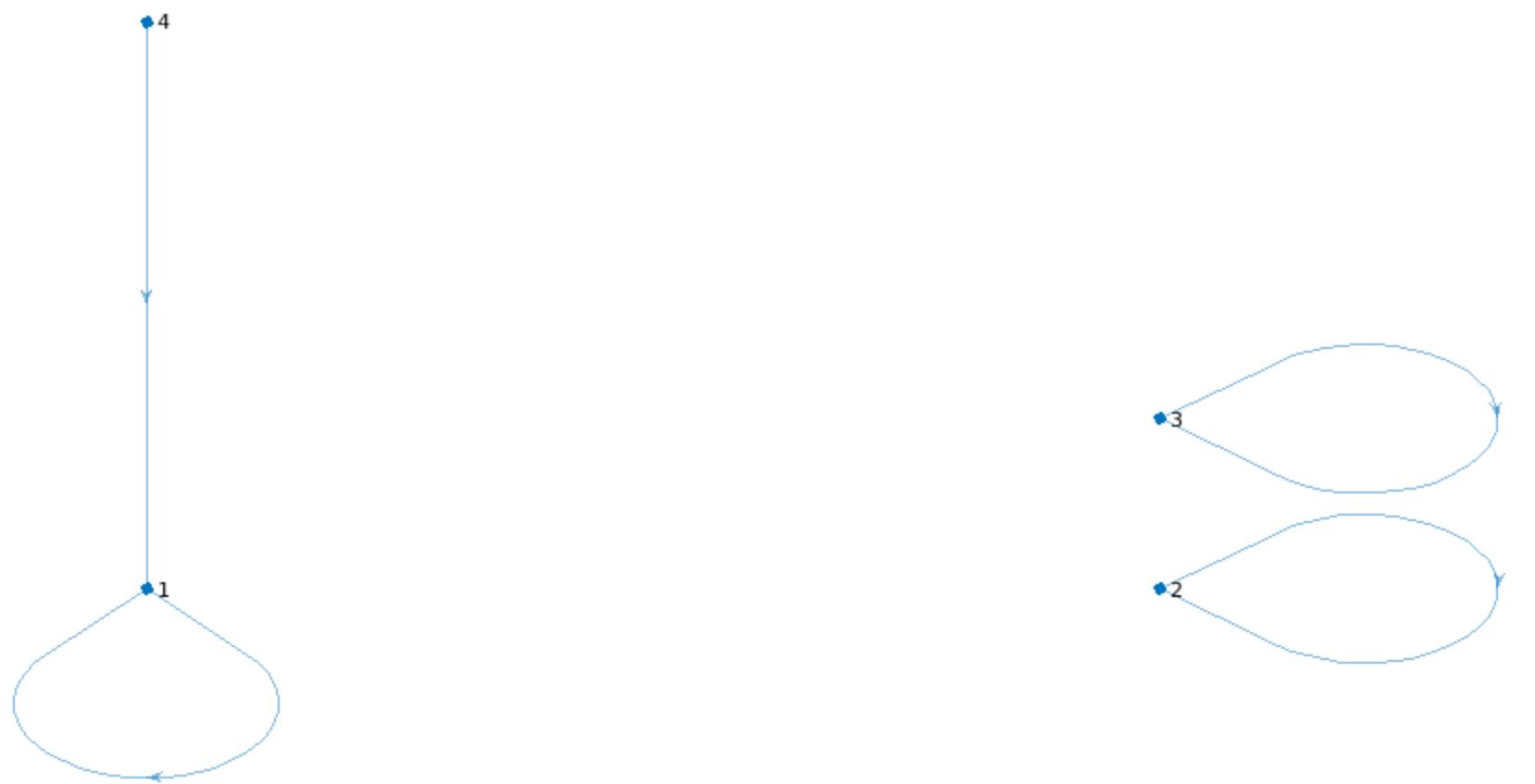
800

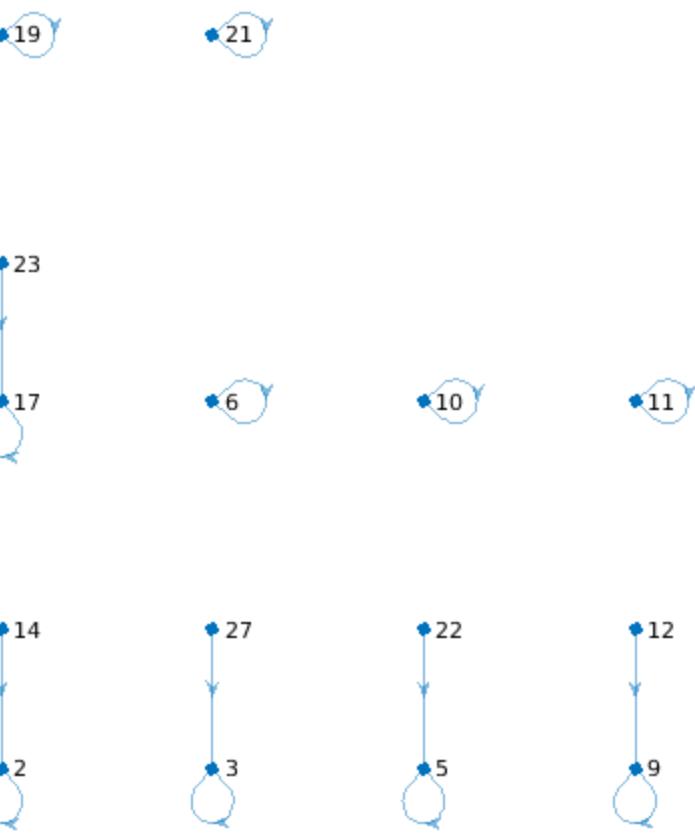
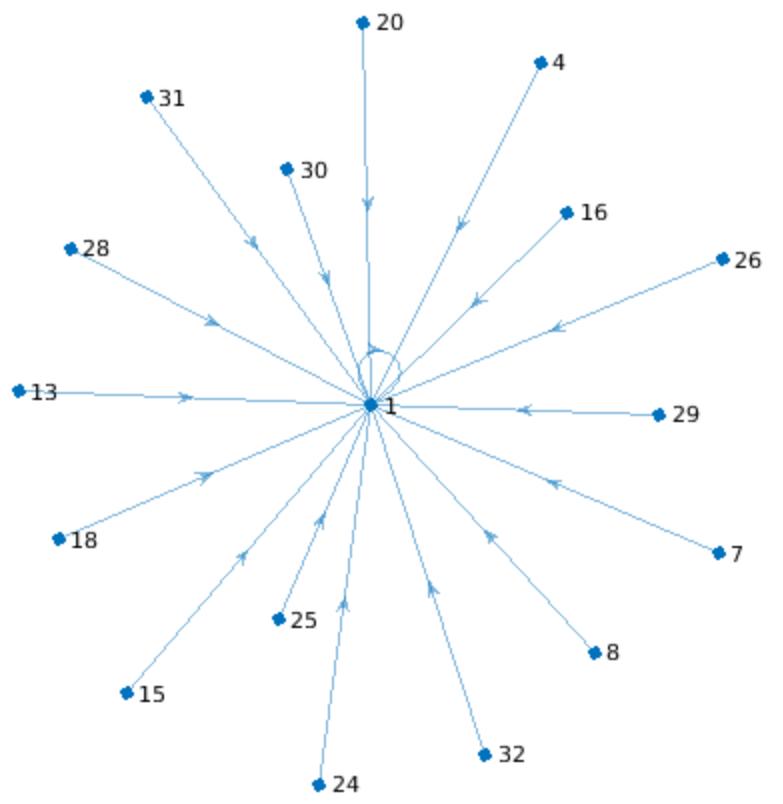
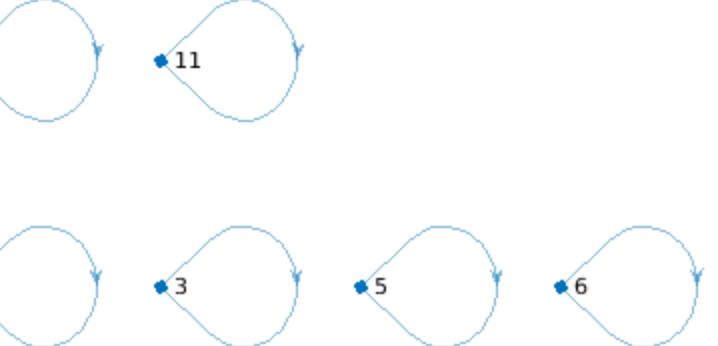
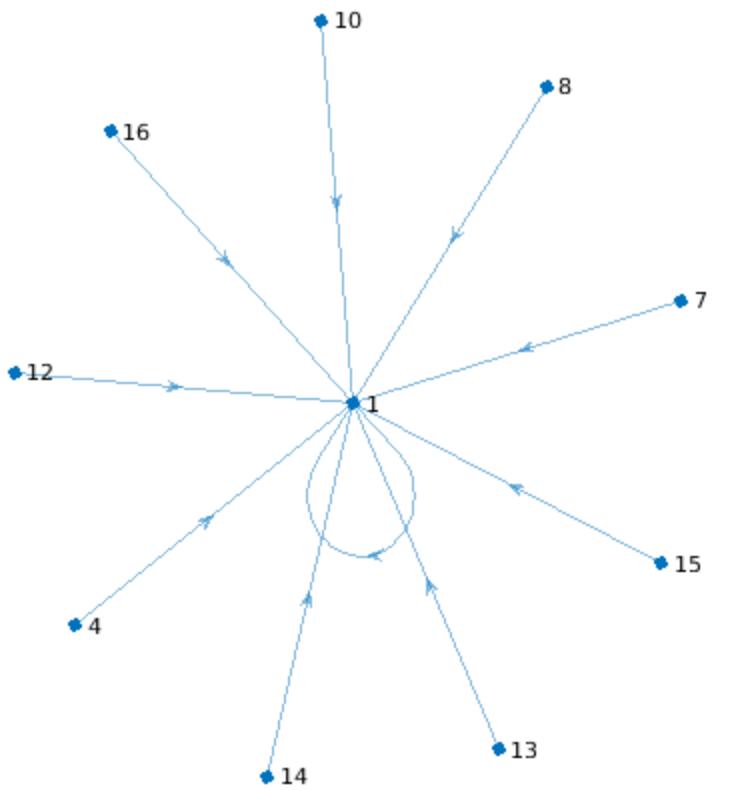
600

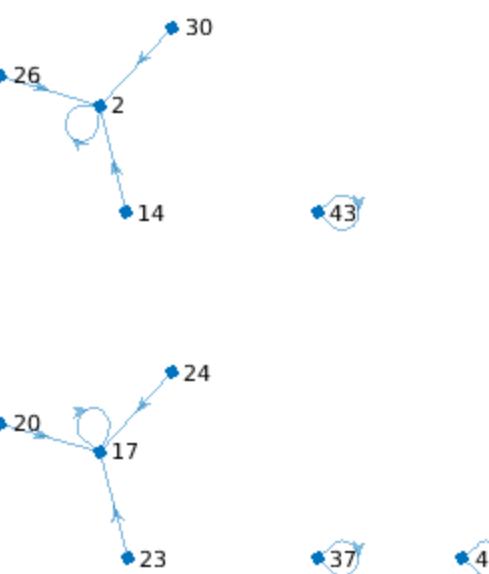
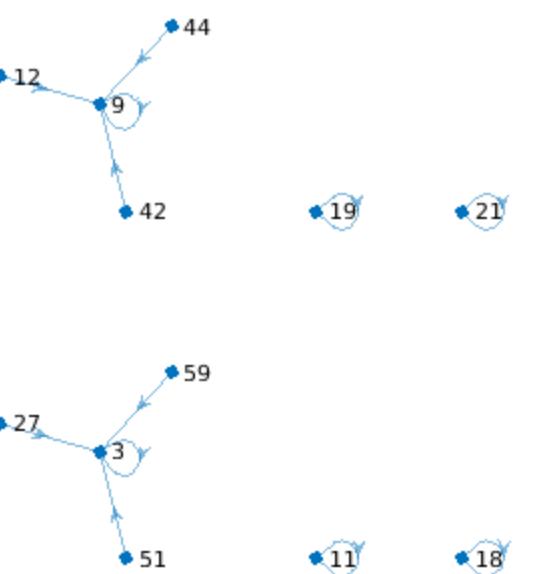
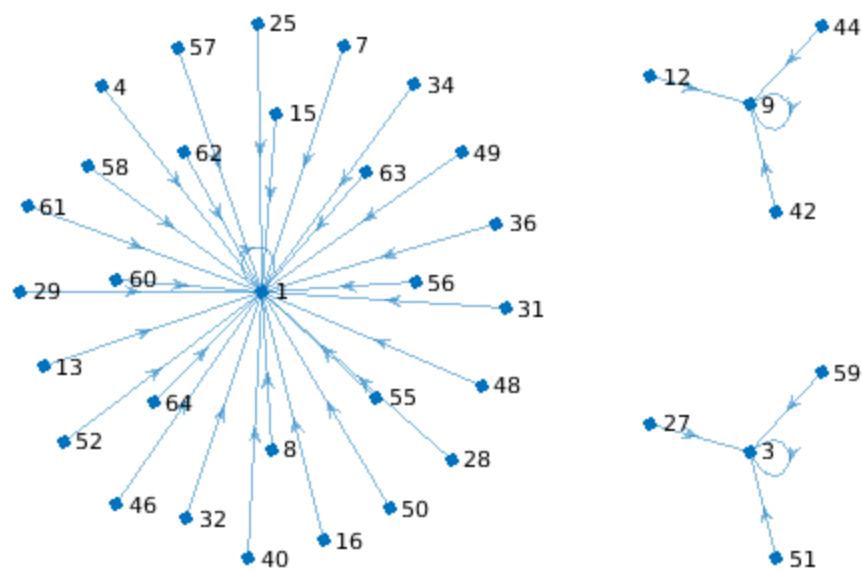
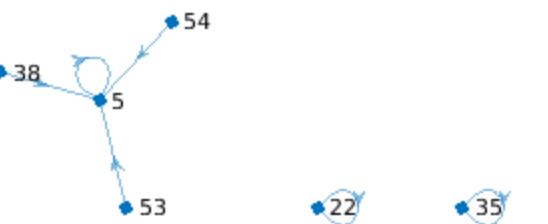
400

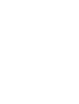
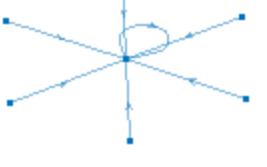
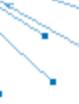
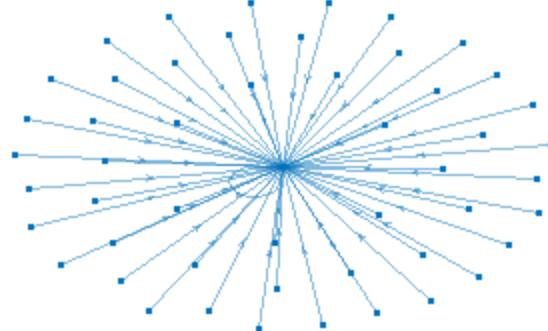
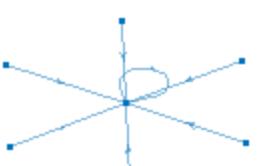
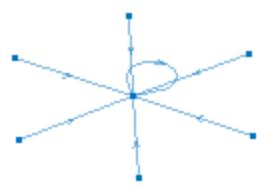
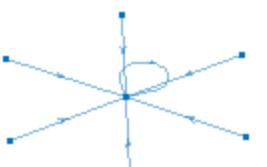
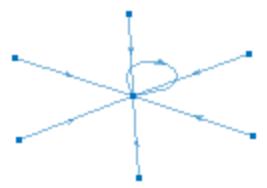
200

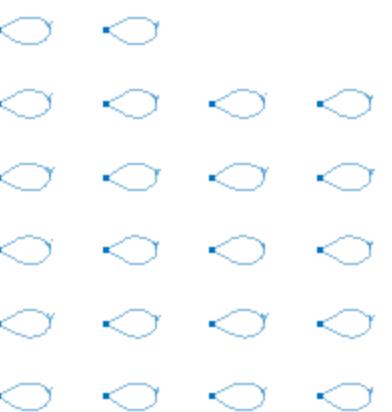
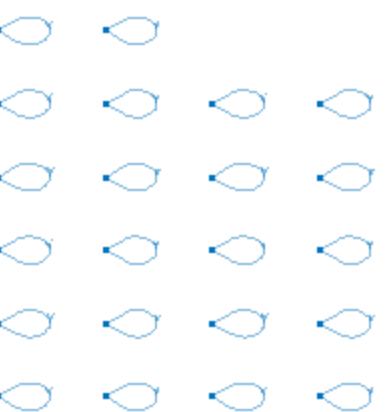
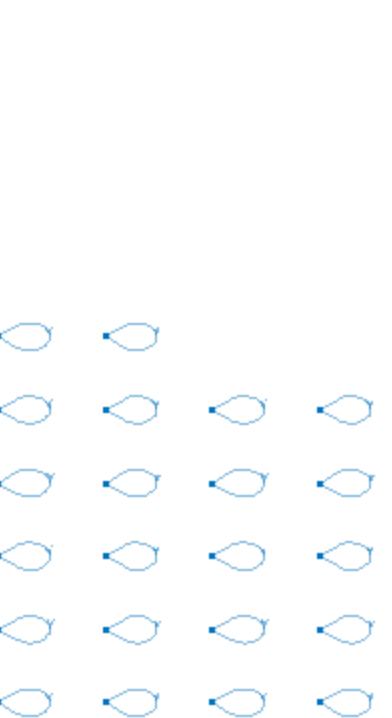
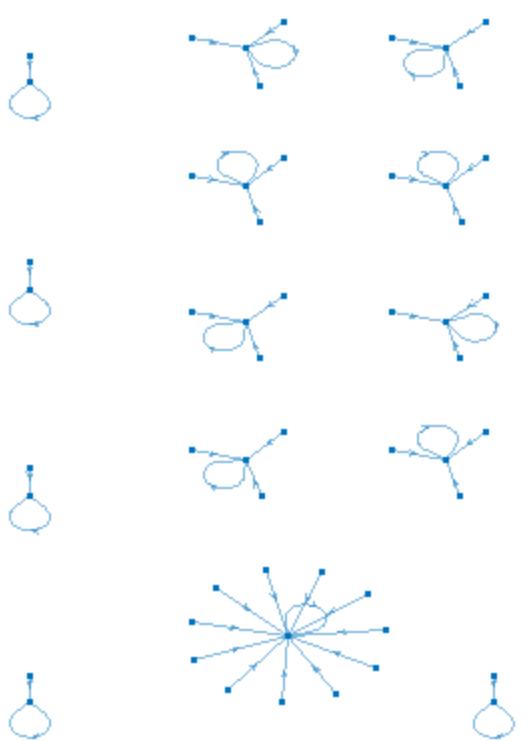
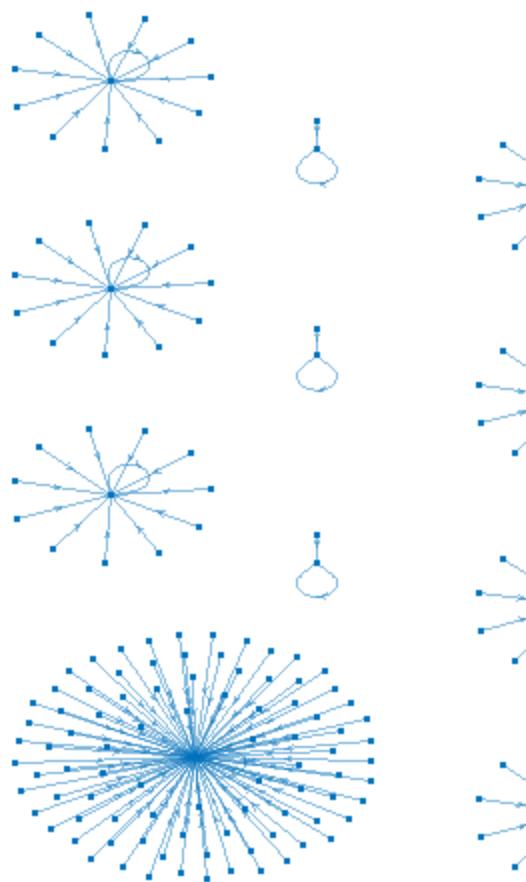
100

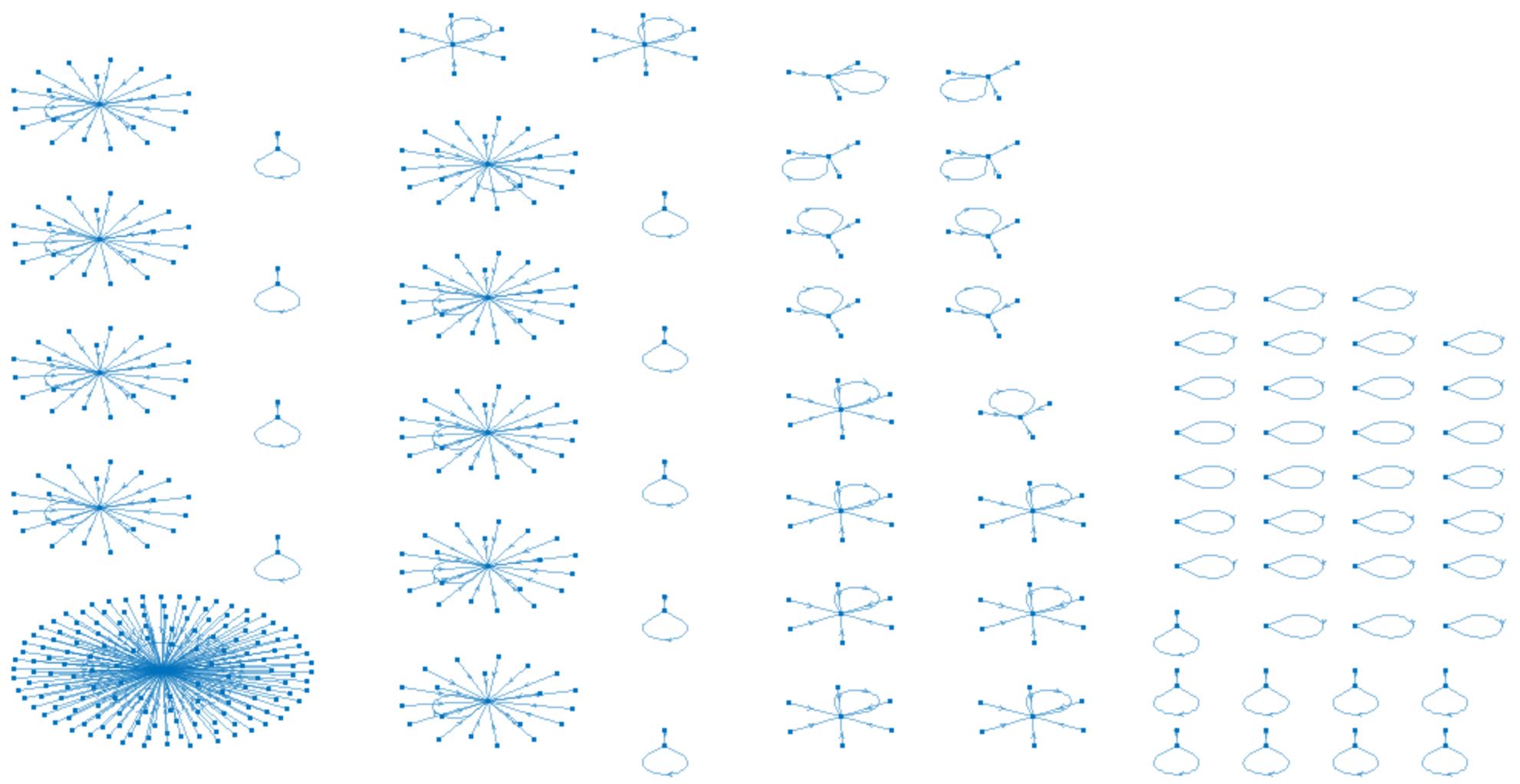


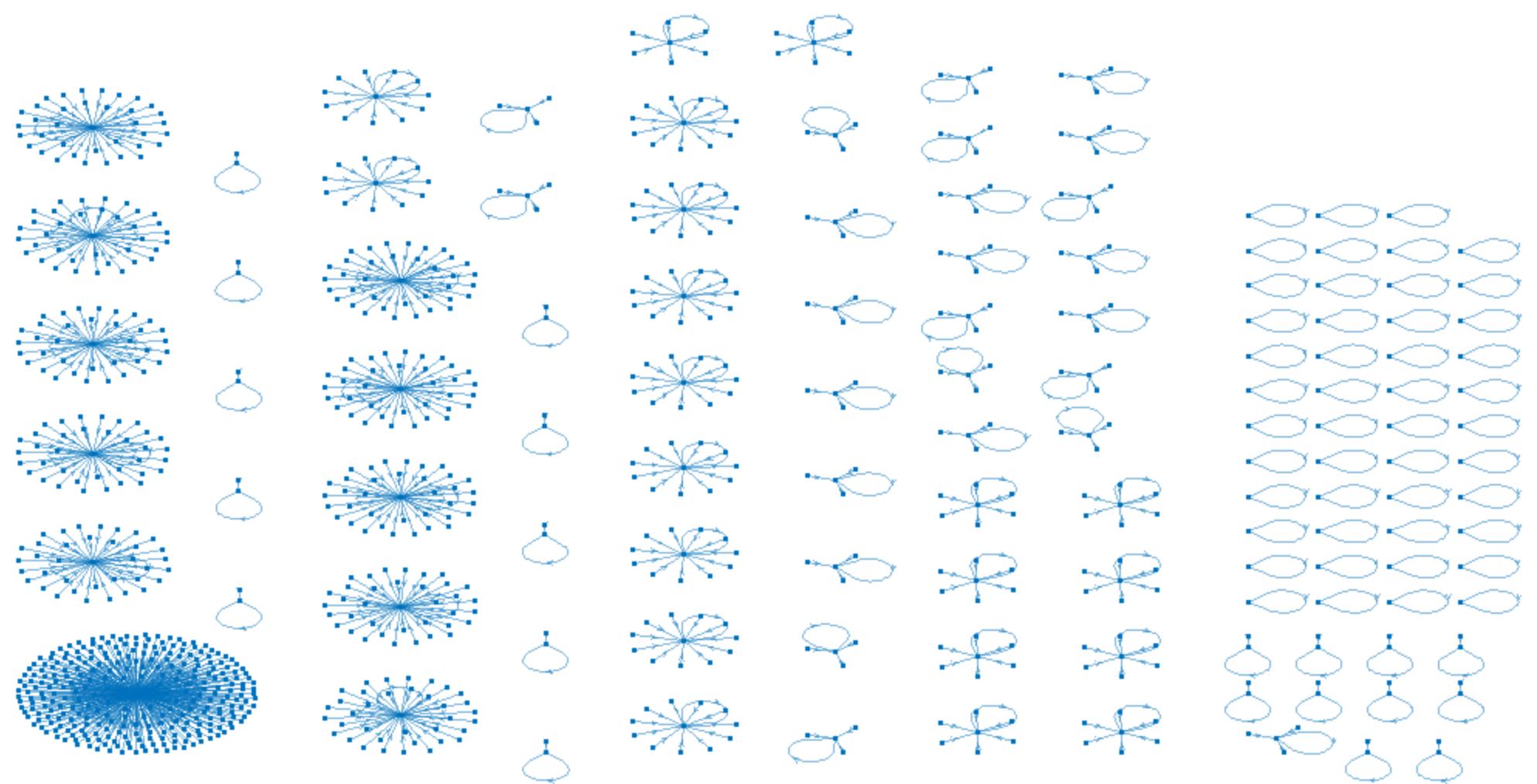


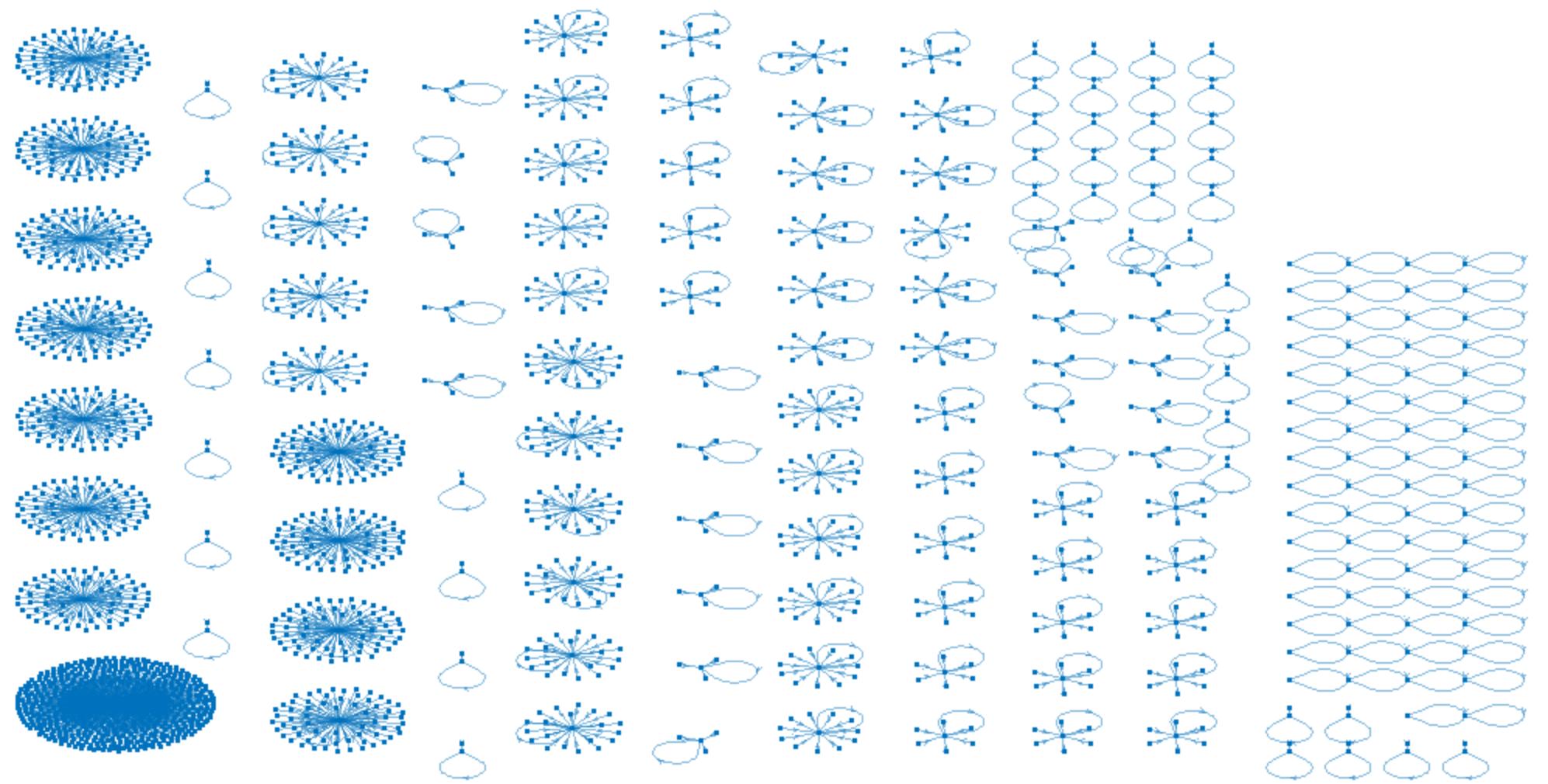


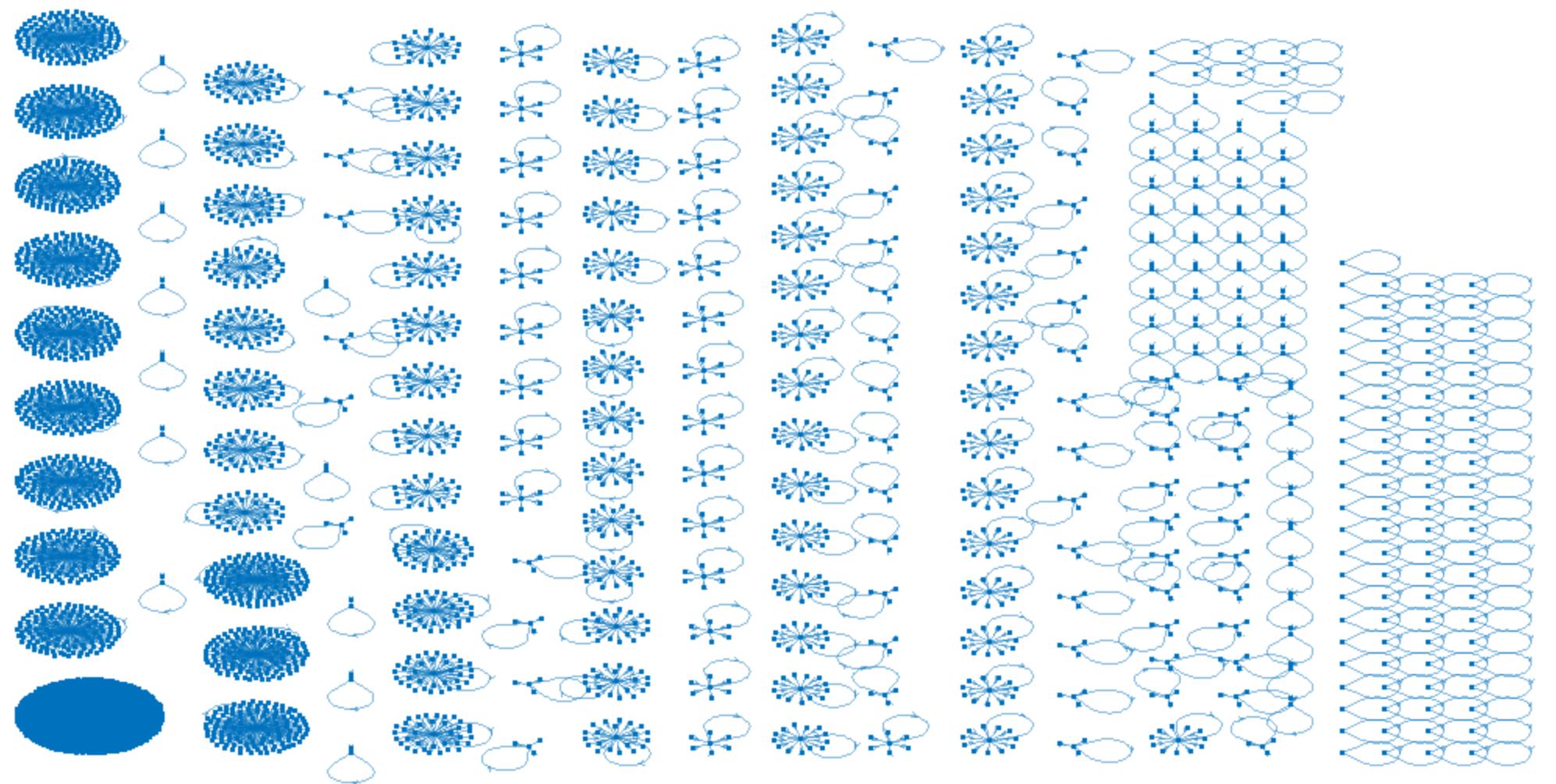


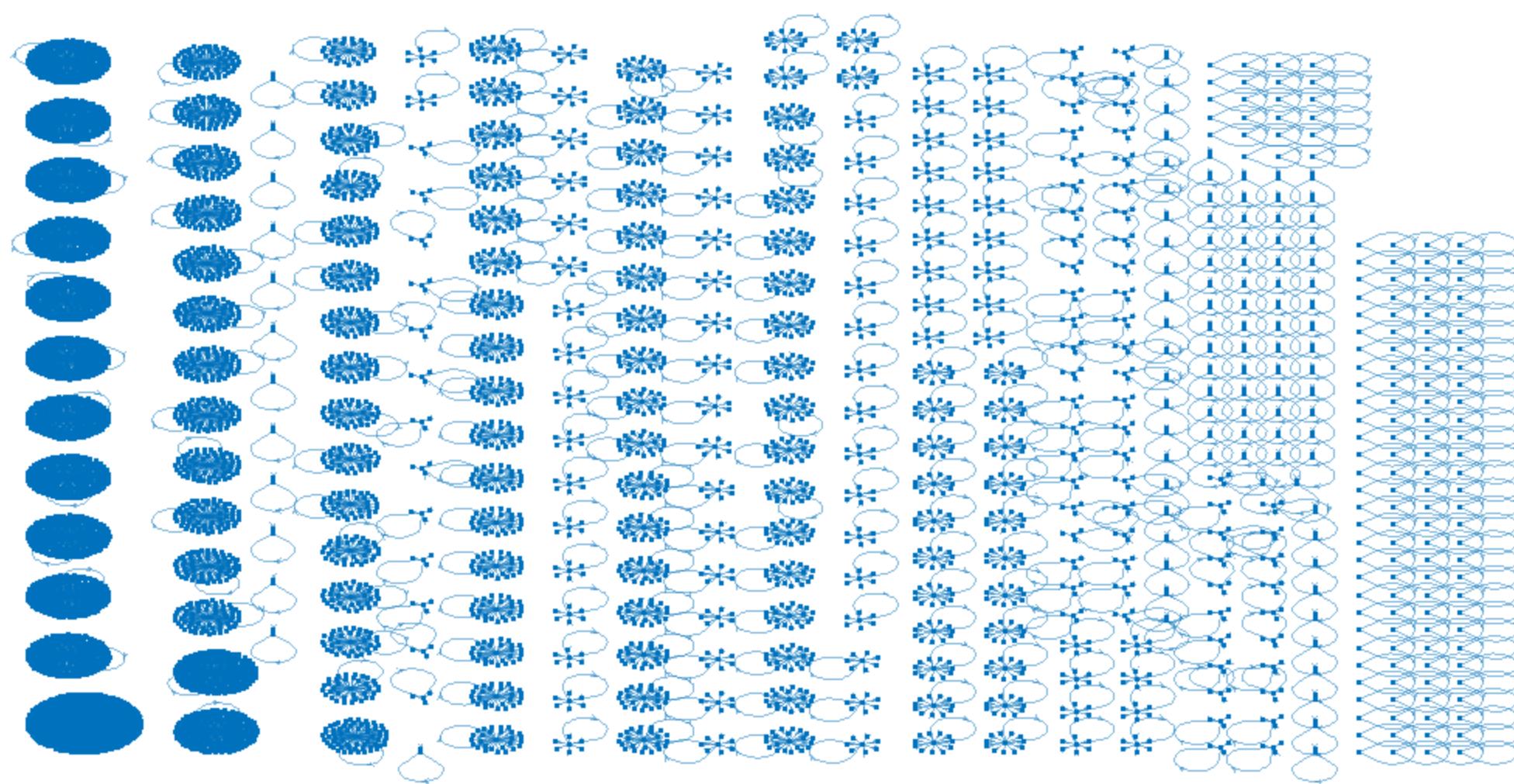




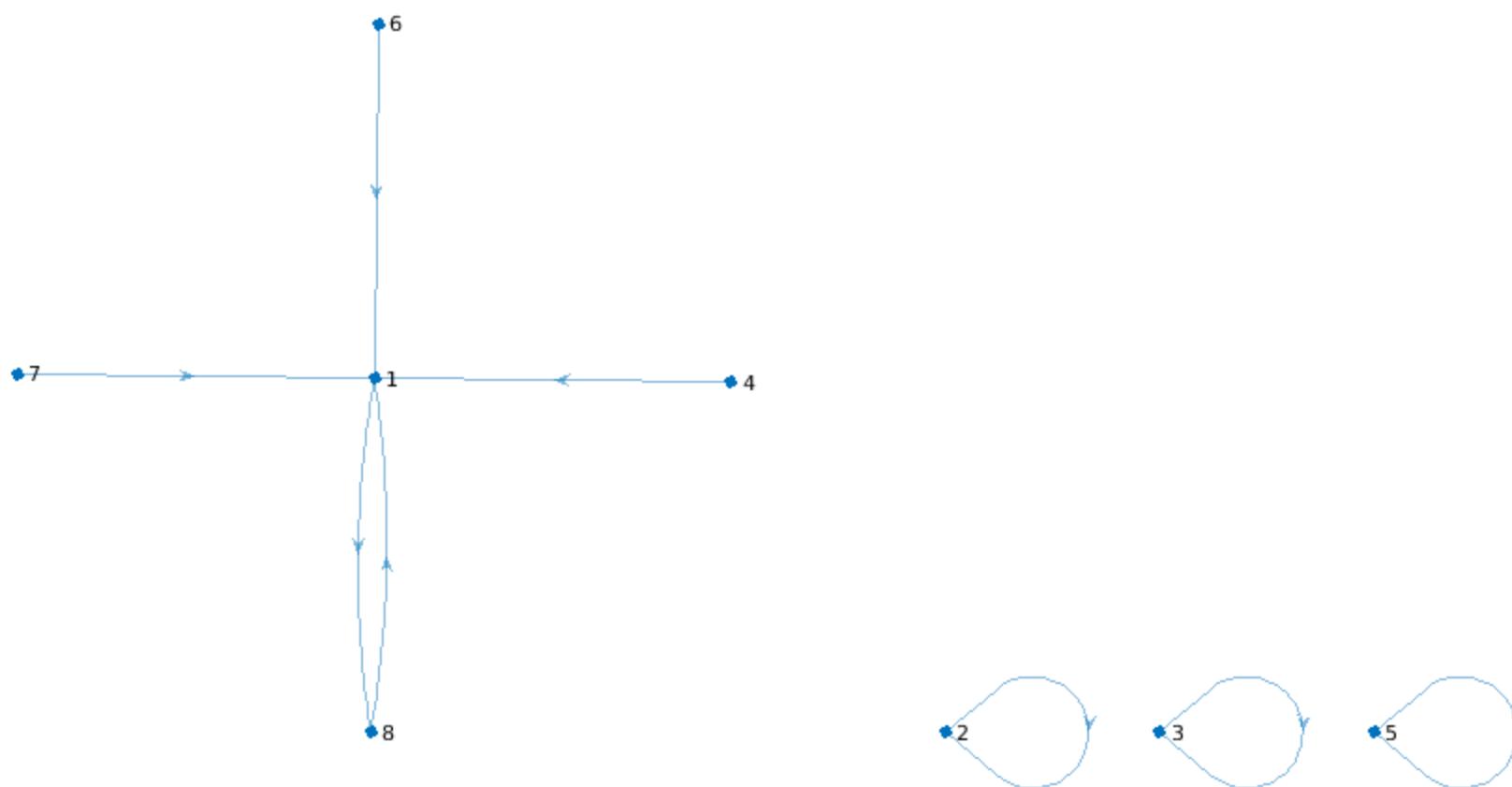


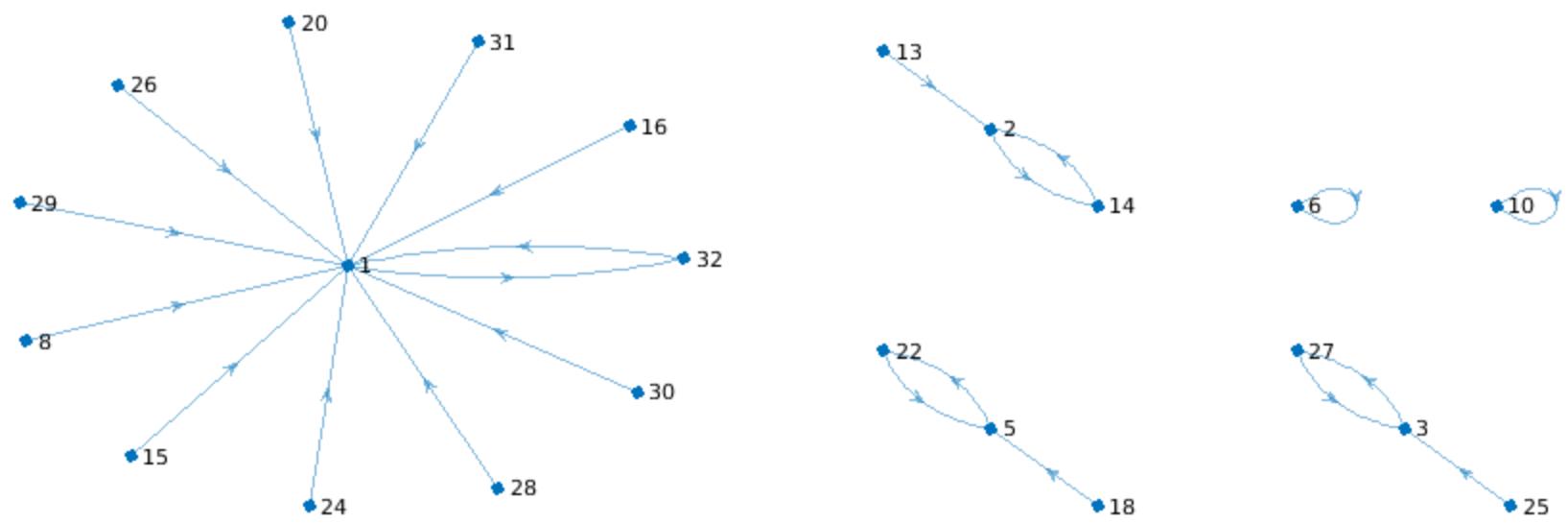
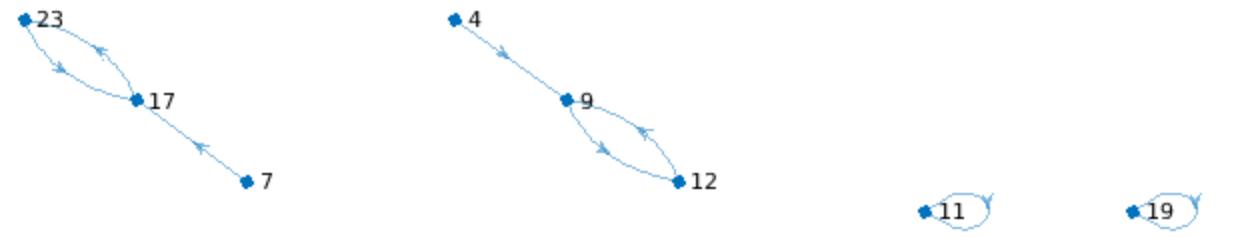
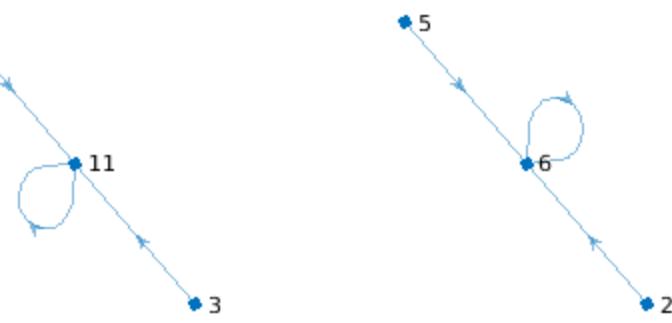
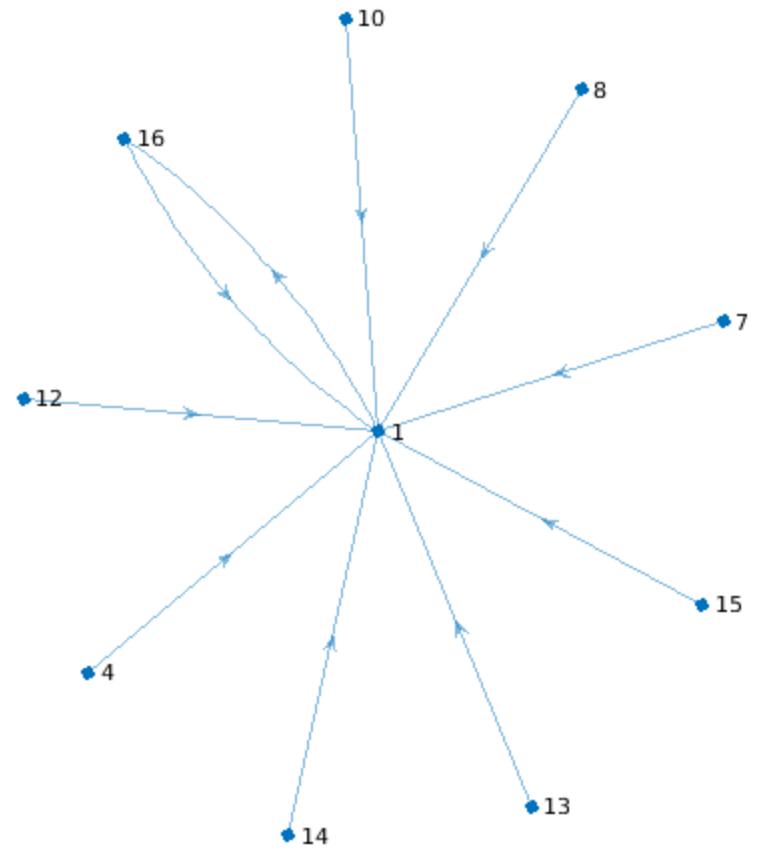


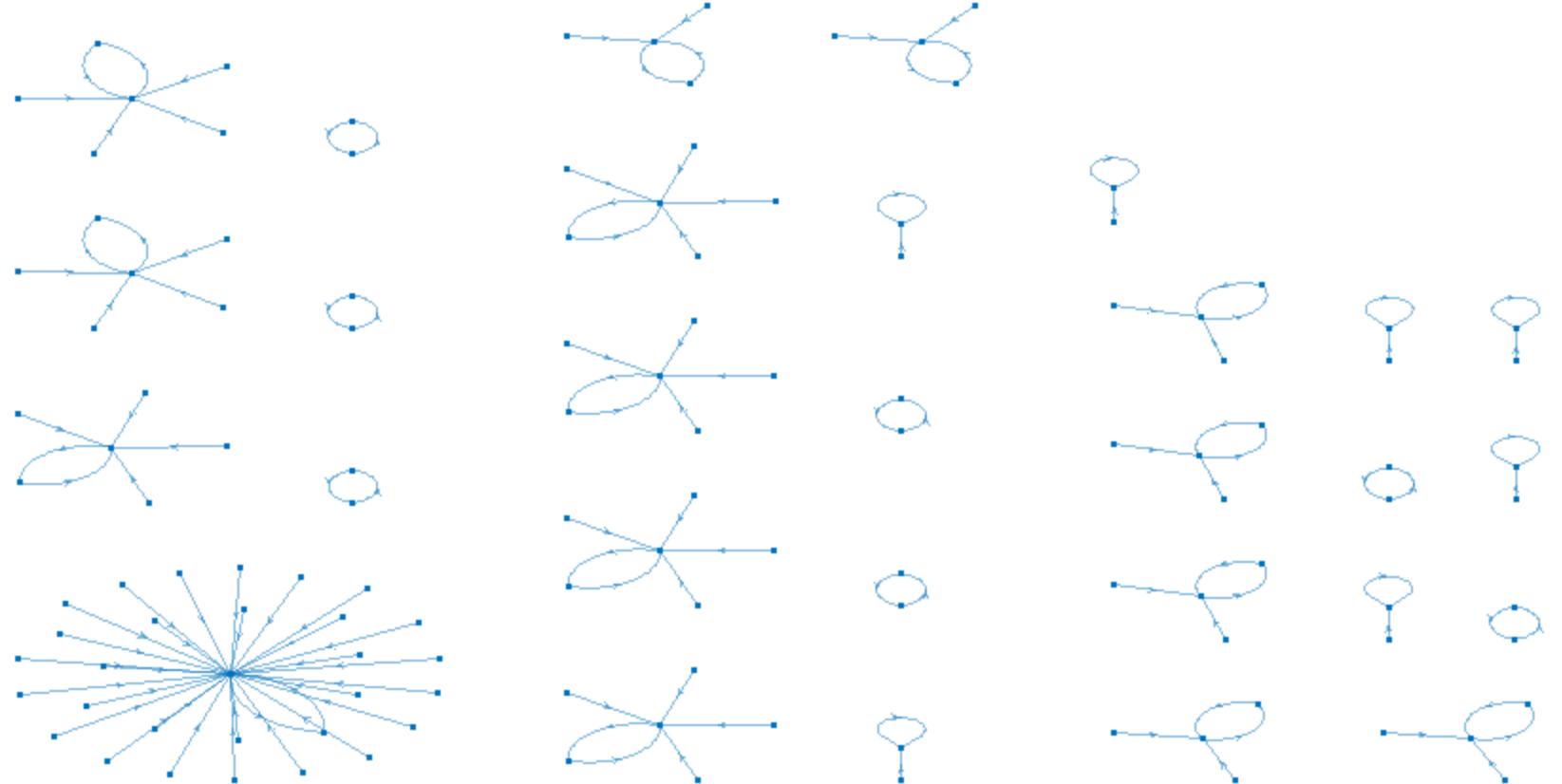
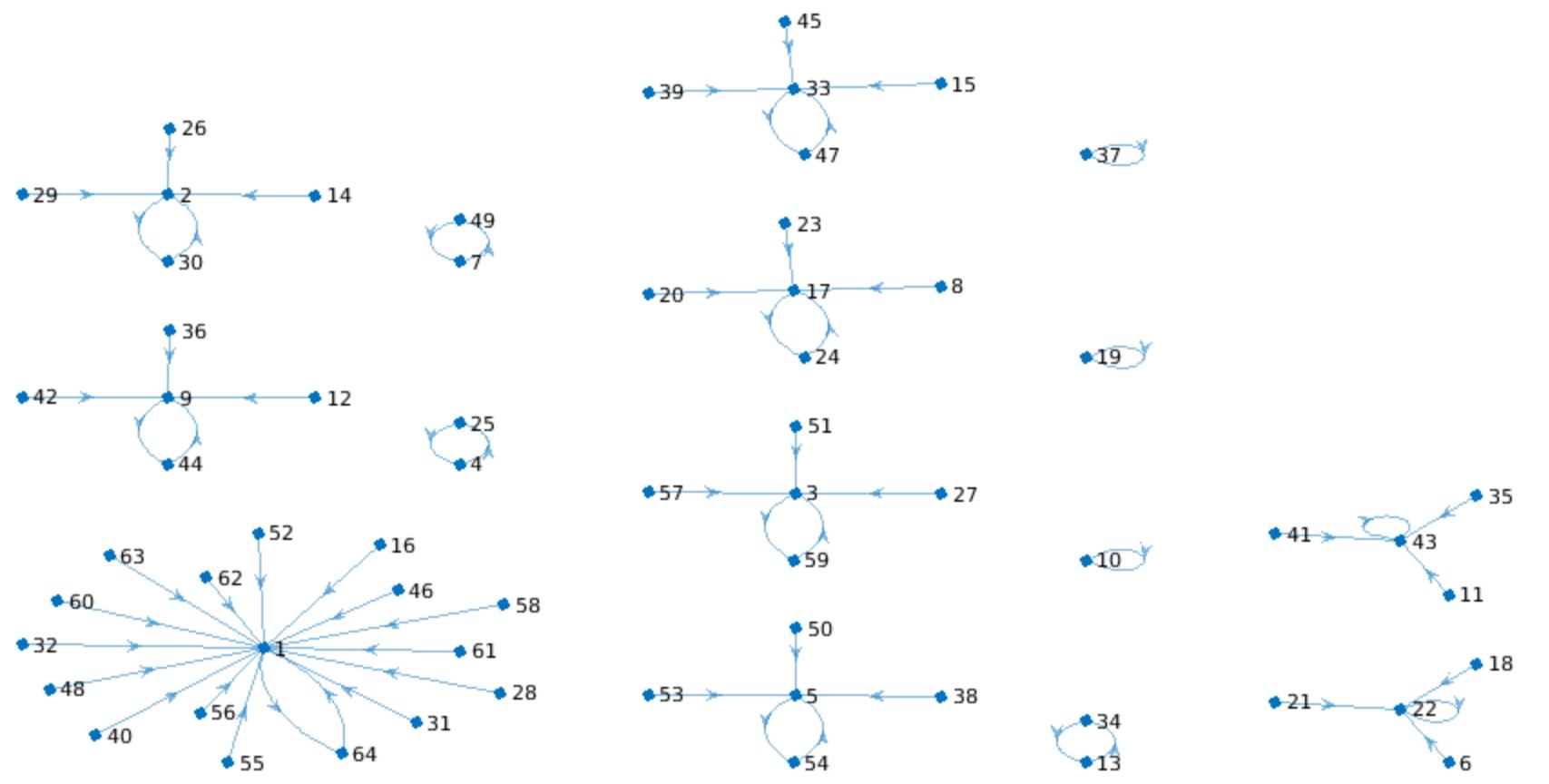


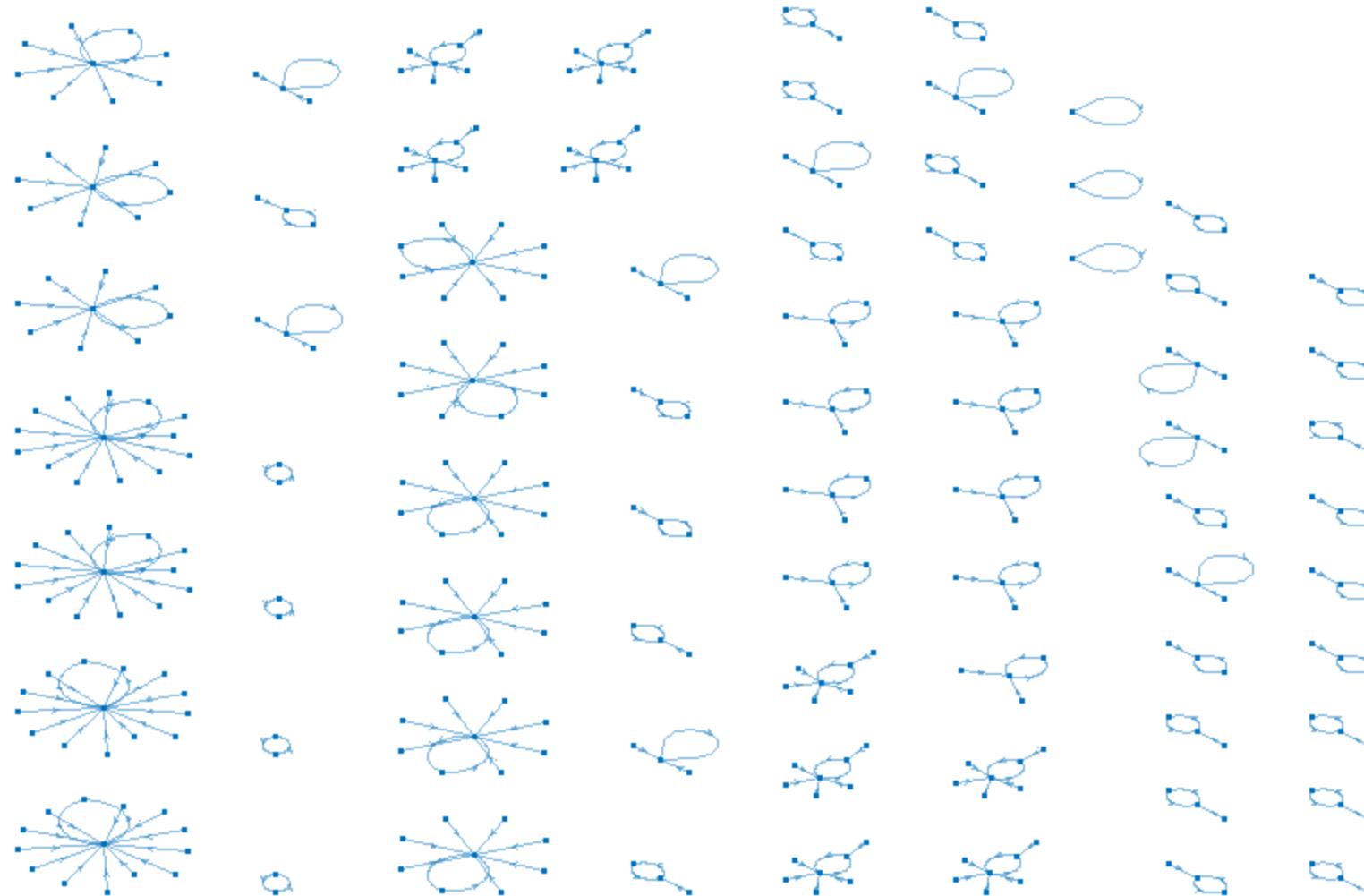
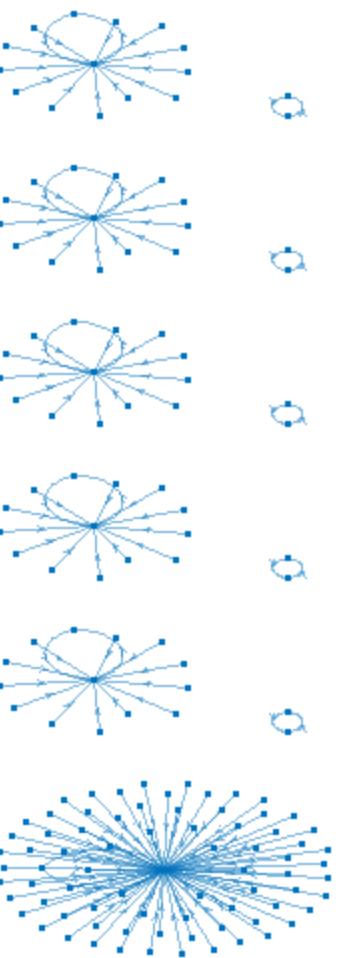
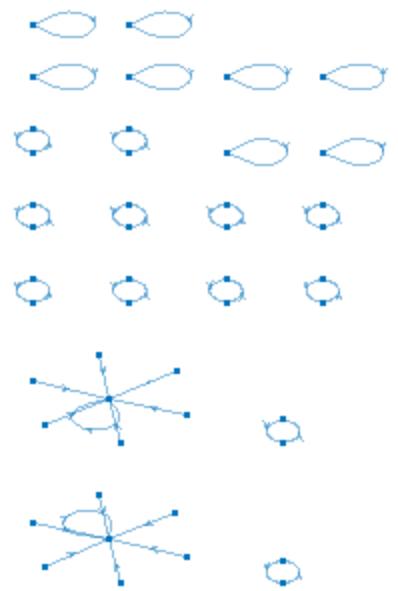
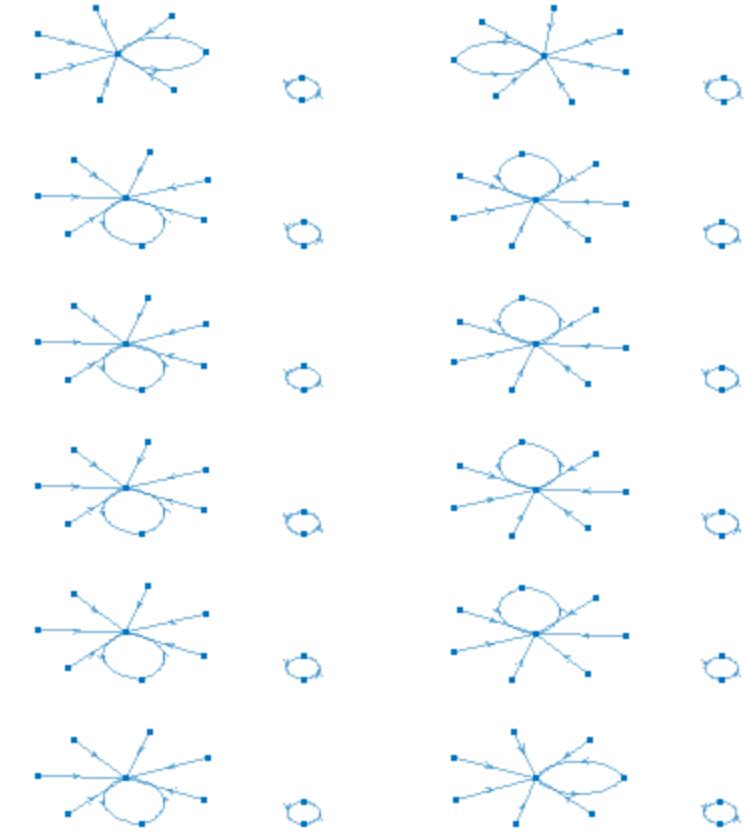
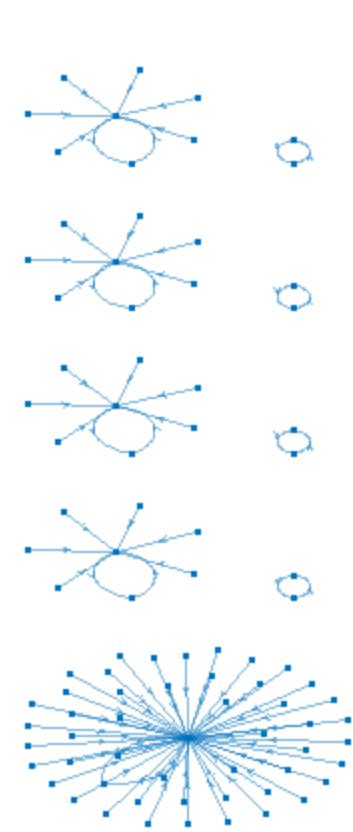


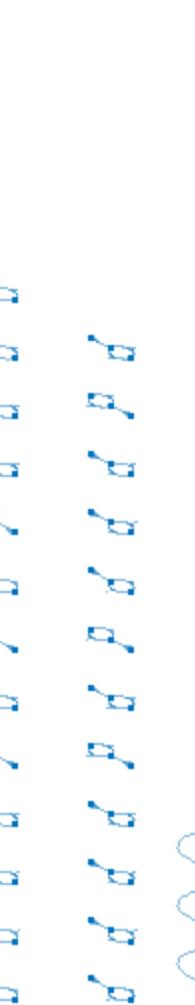
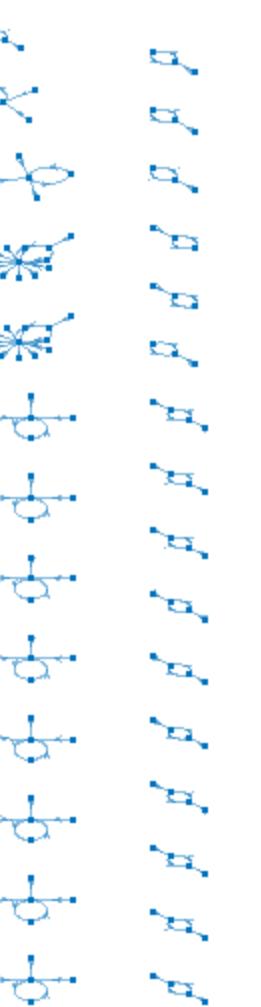
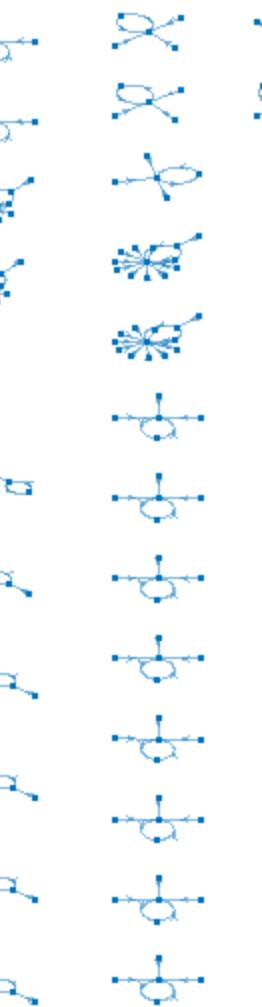
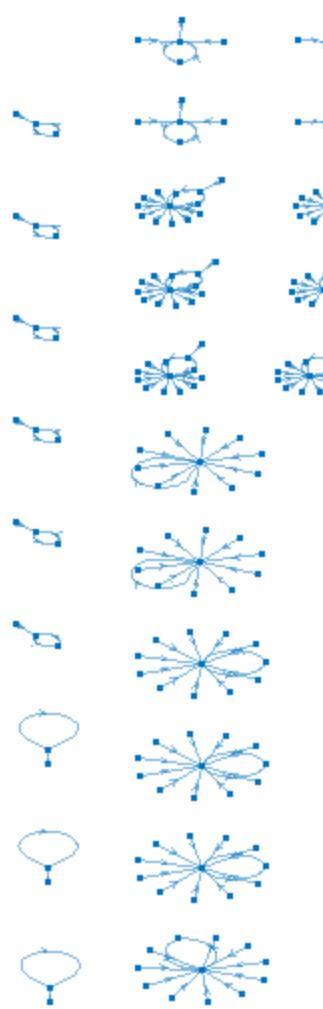
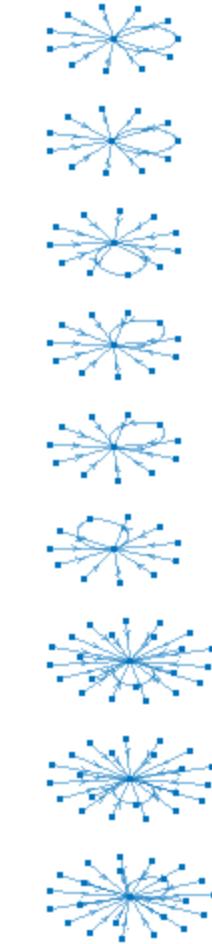
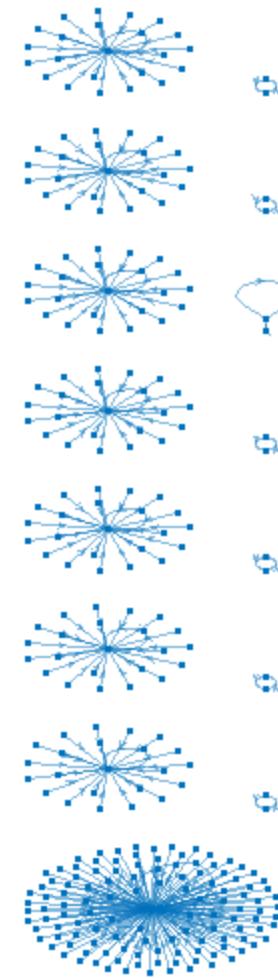
5

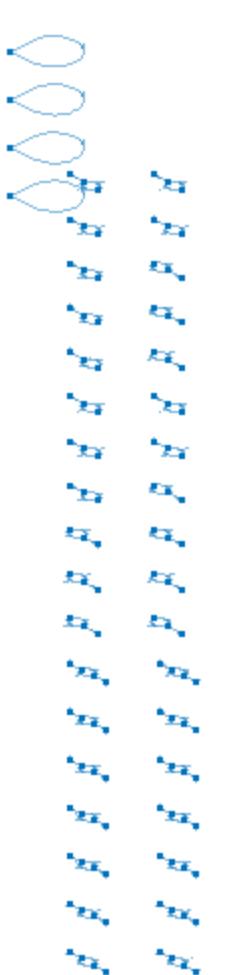
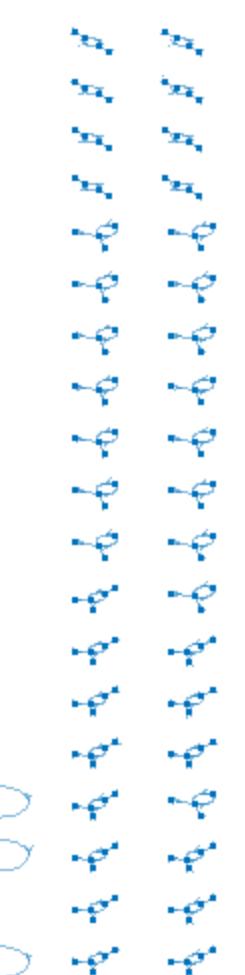
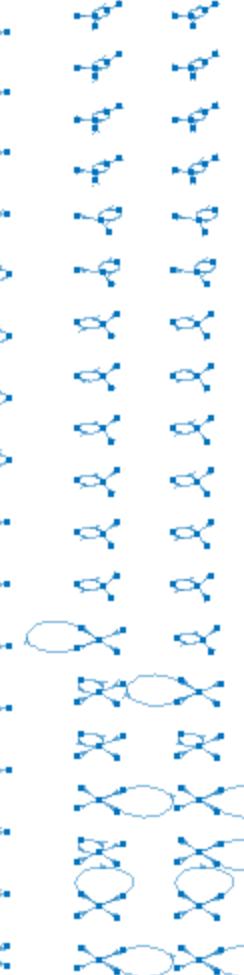
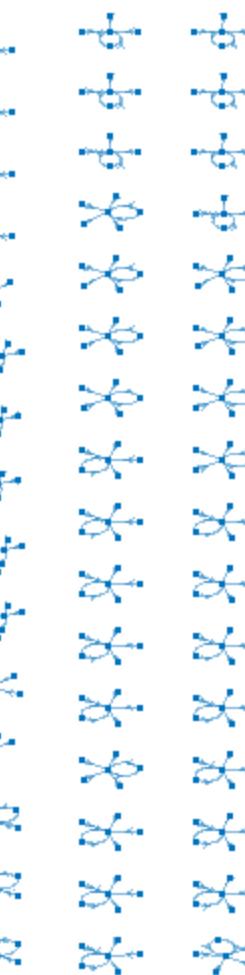
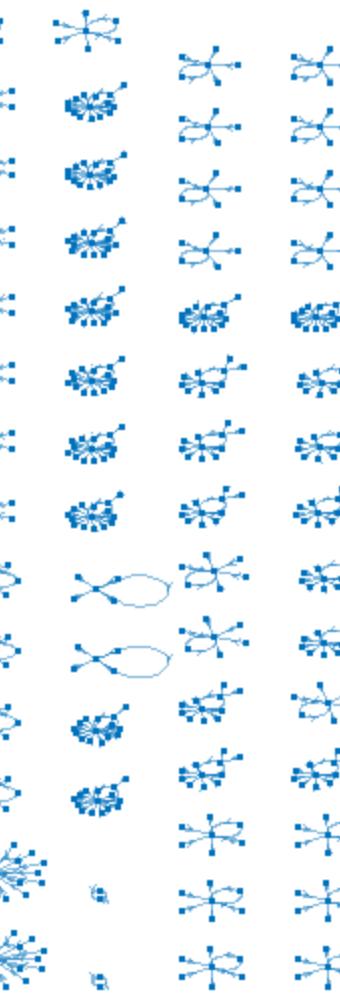
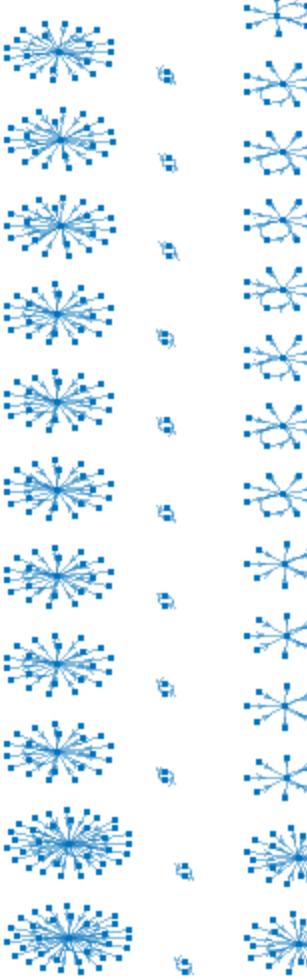
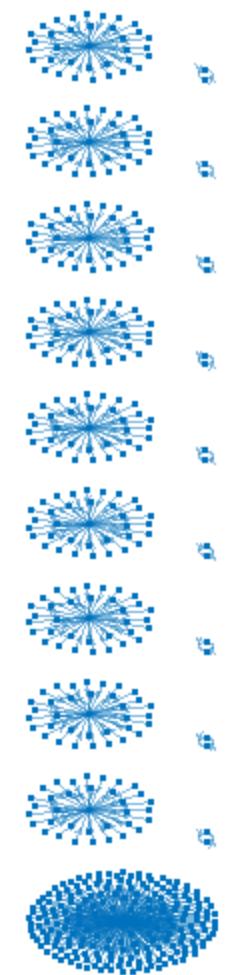


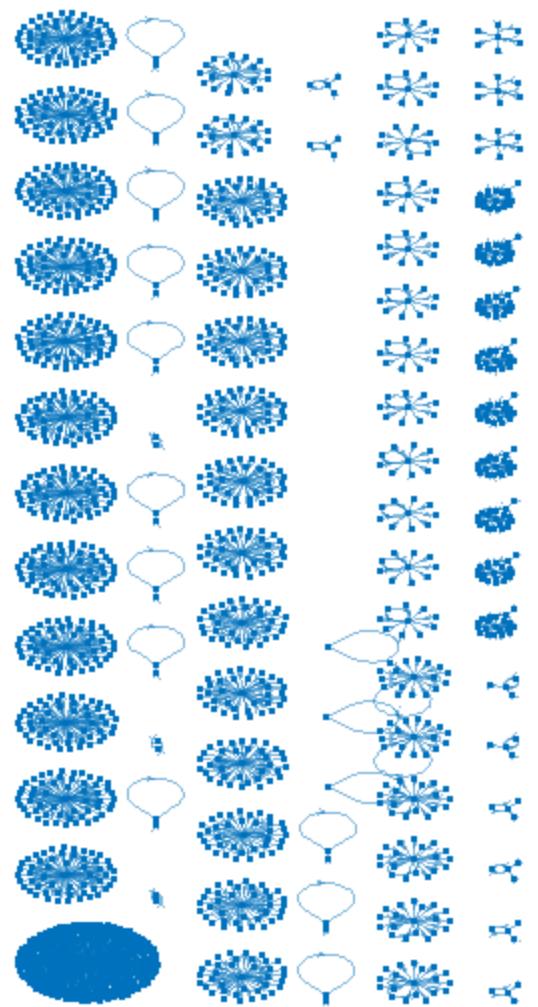








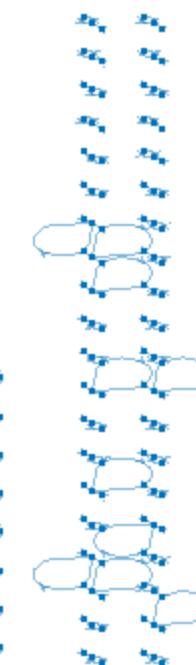


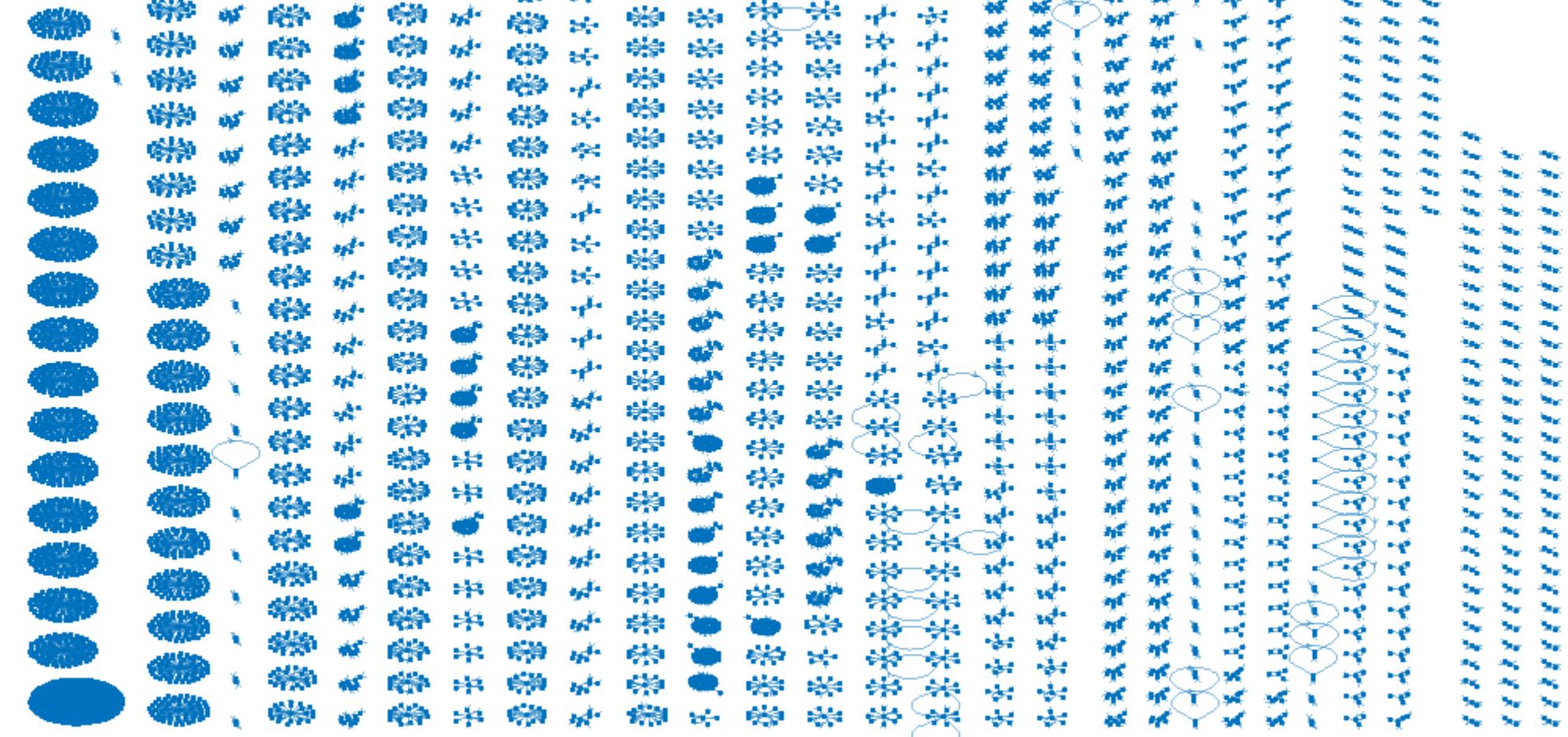


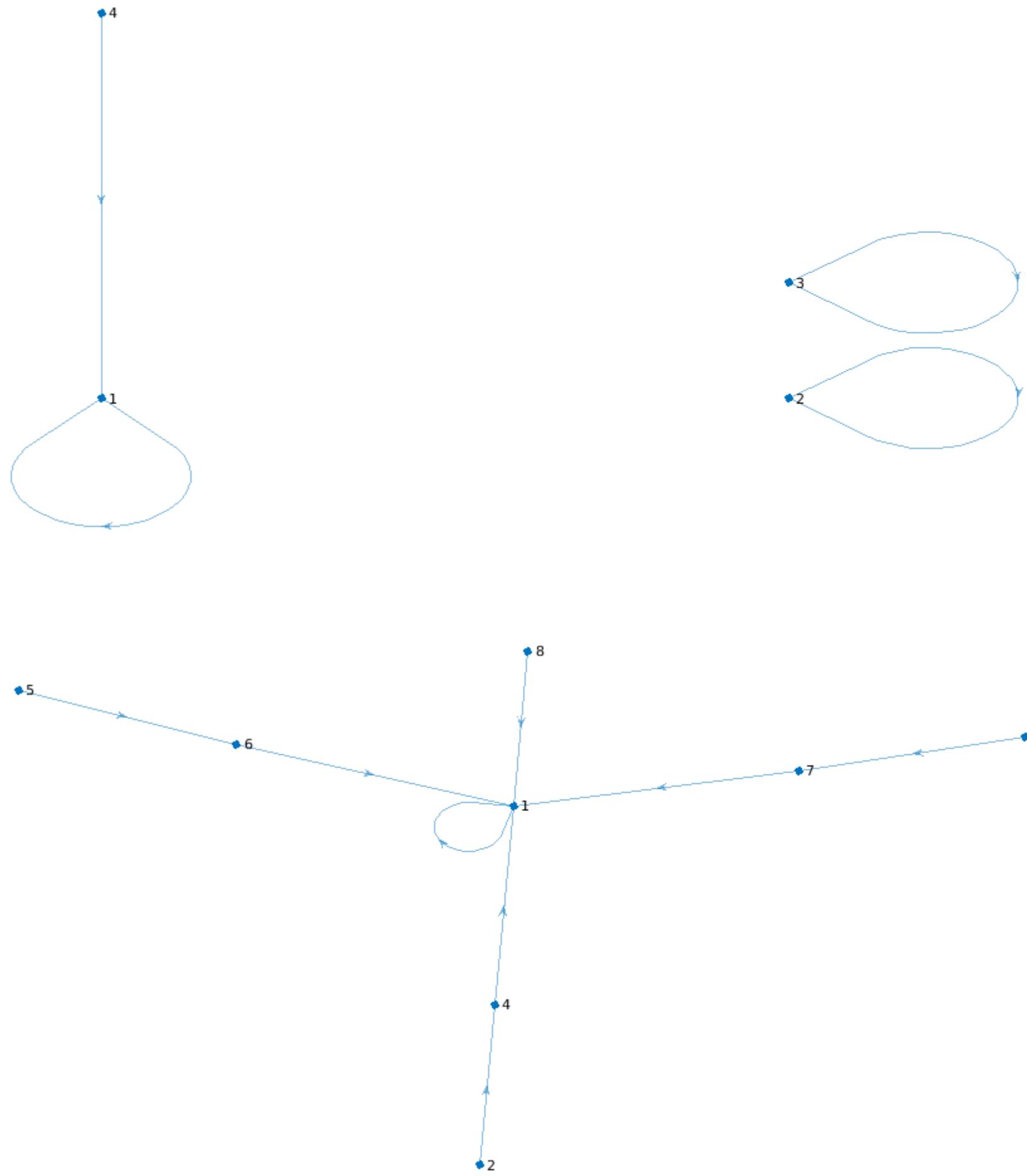
A decorative vertical border pattern consisting of a repeating sequence of blue stylized floral or star-like motifs on a white background.

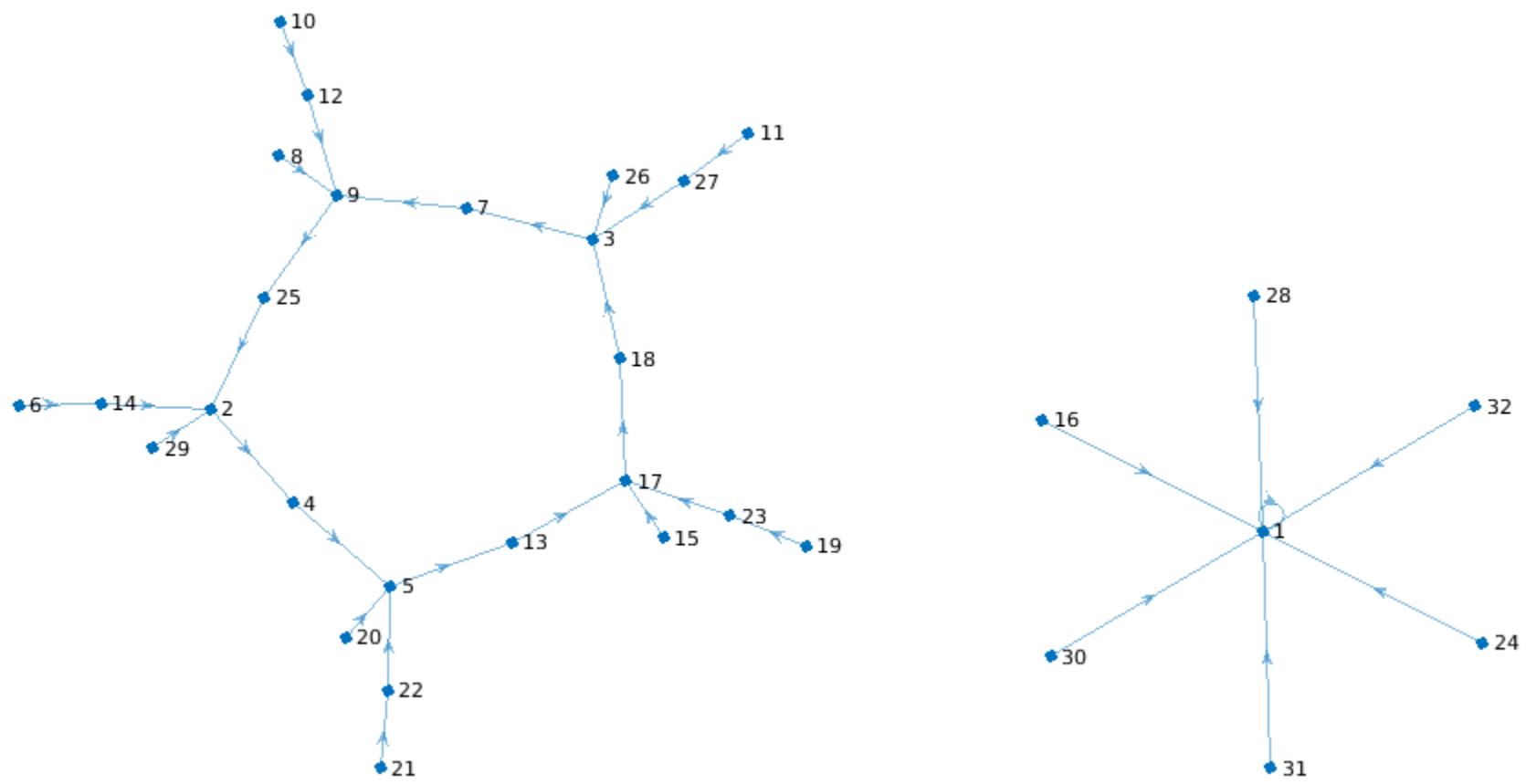
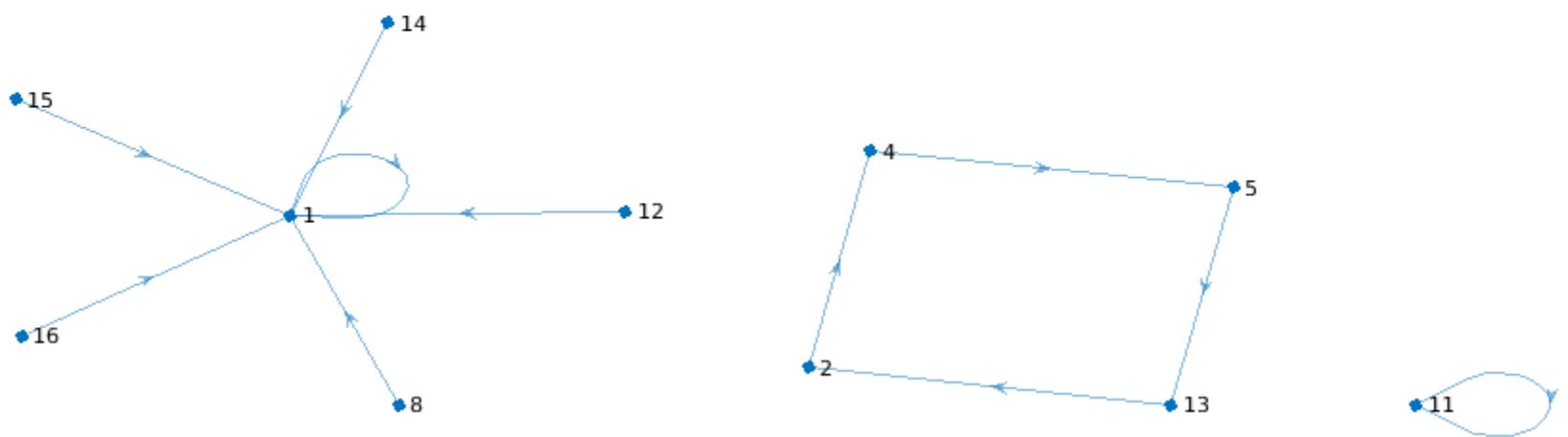
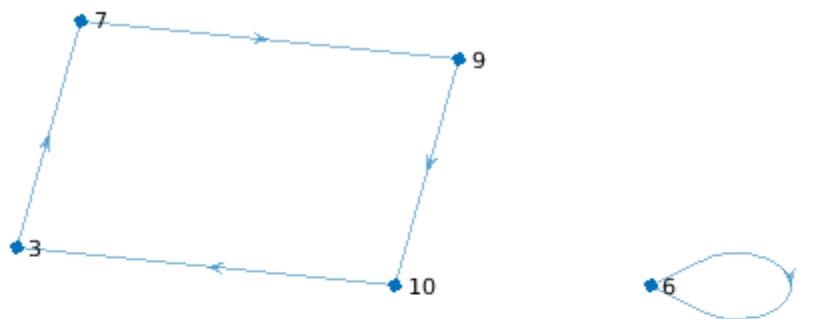
卷之三

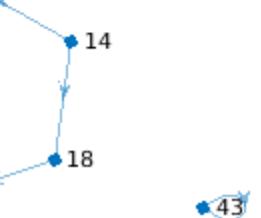
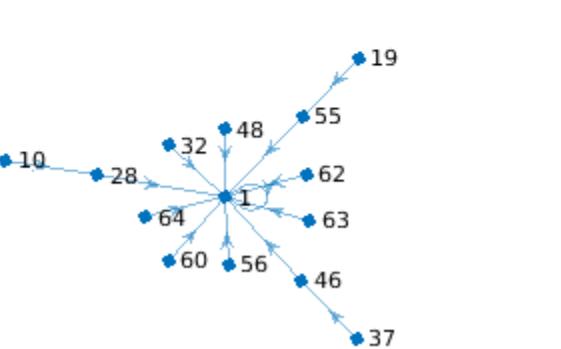
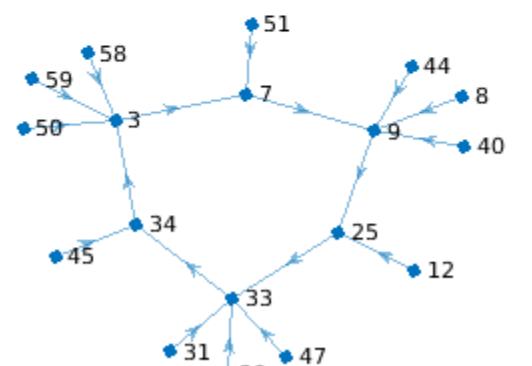
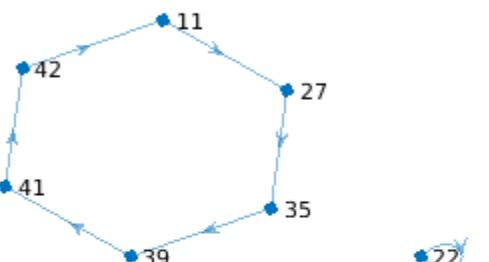
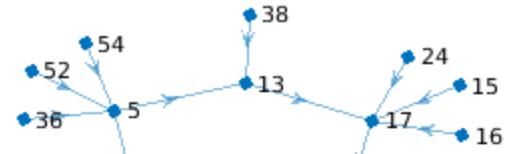
卷之三

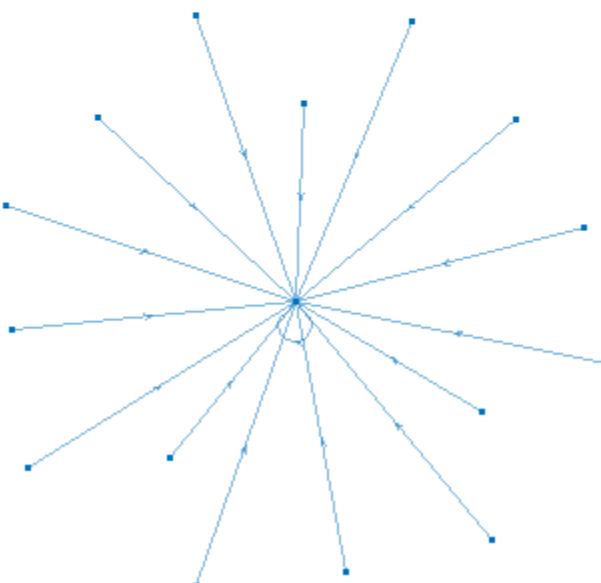
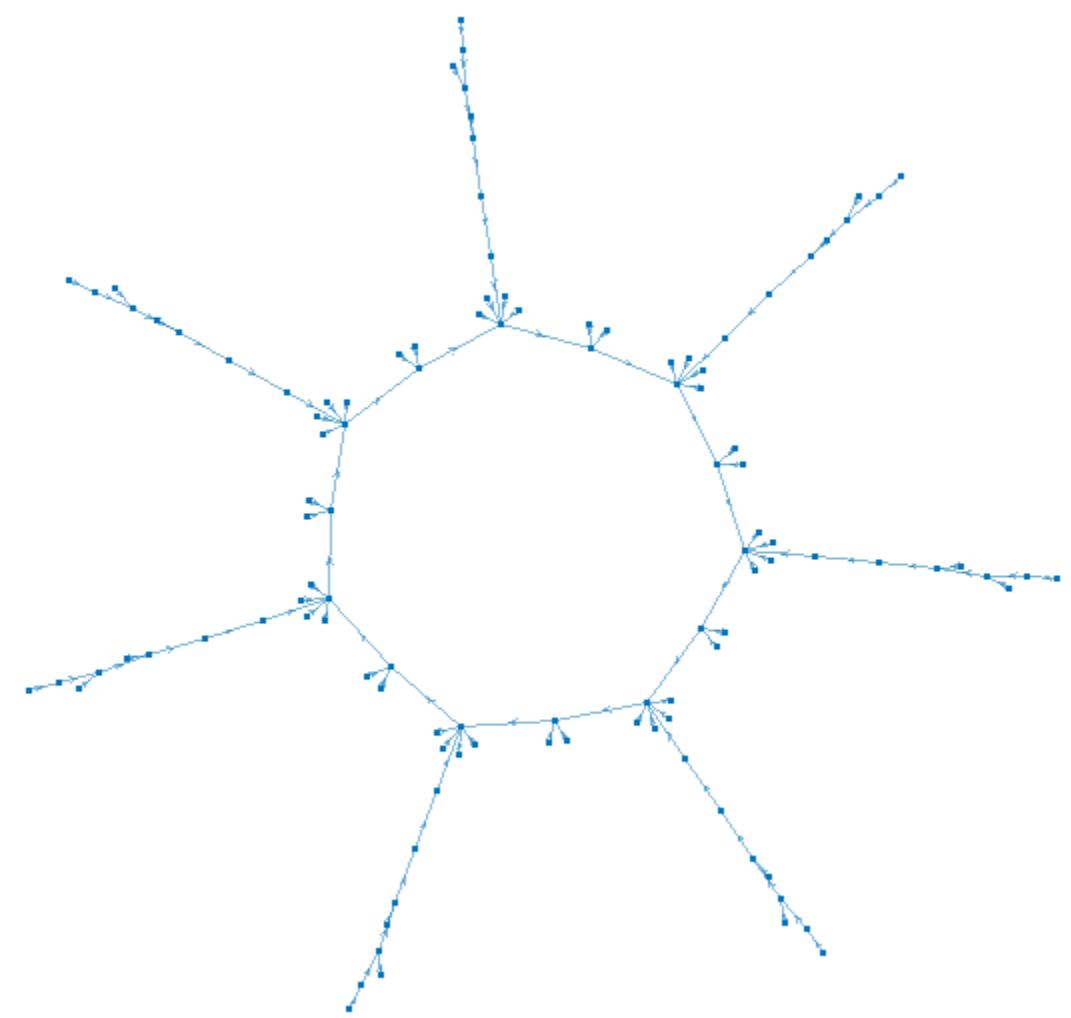


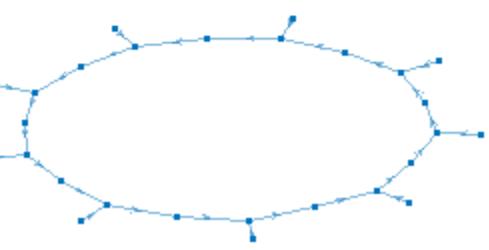
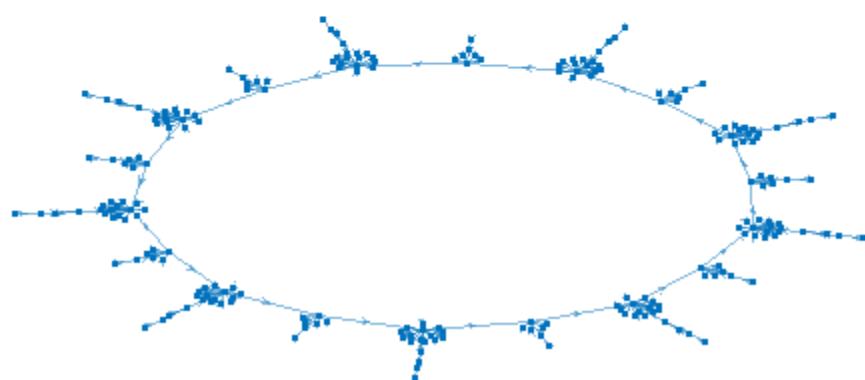
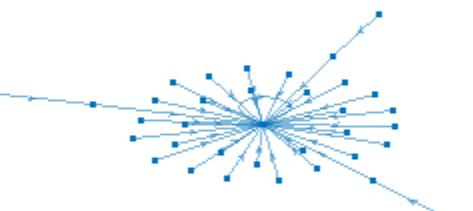
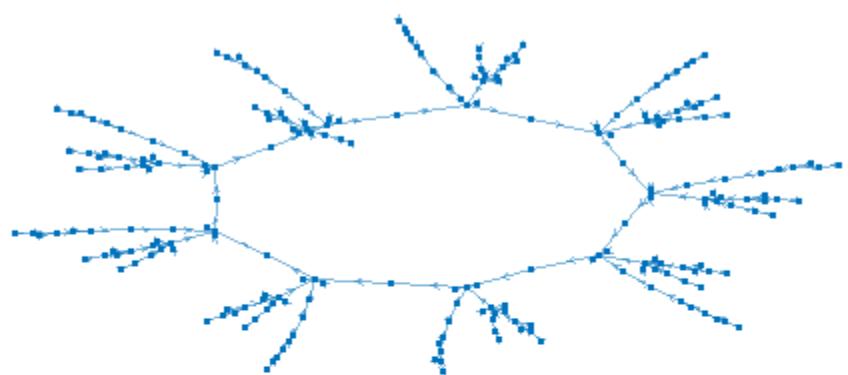
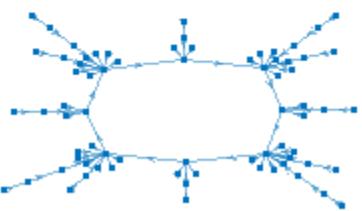
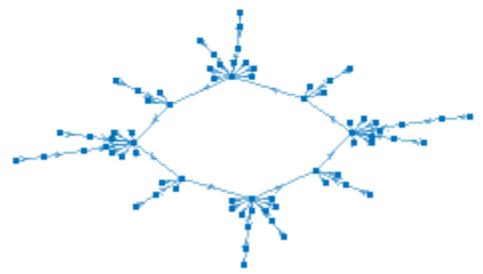
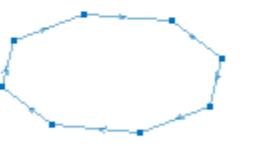
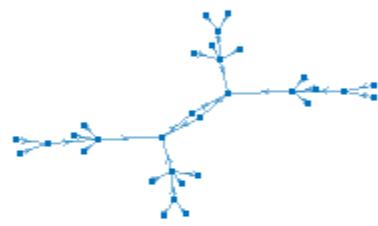
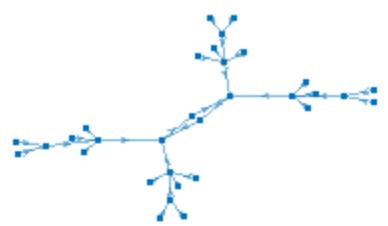


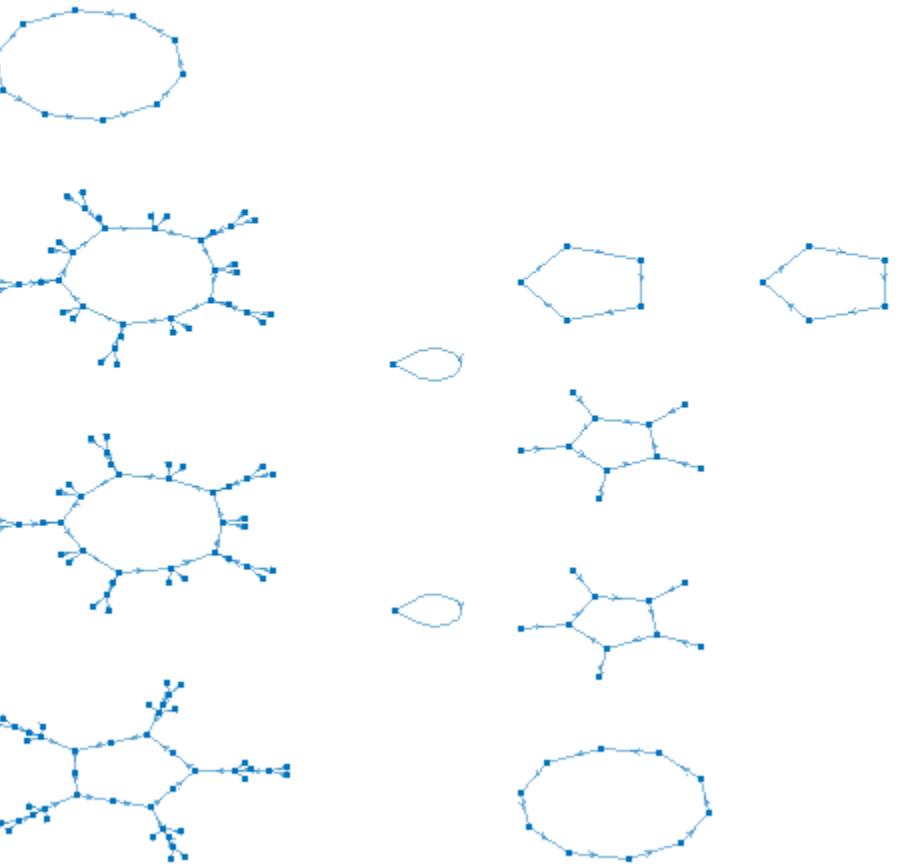
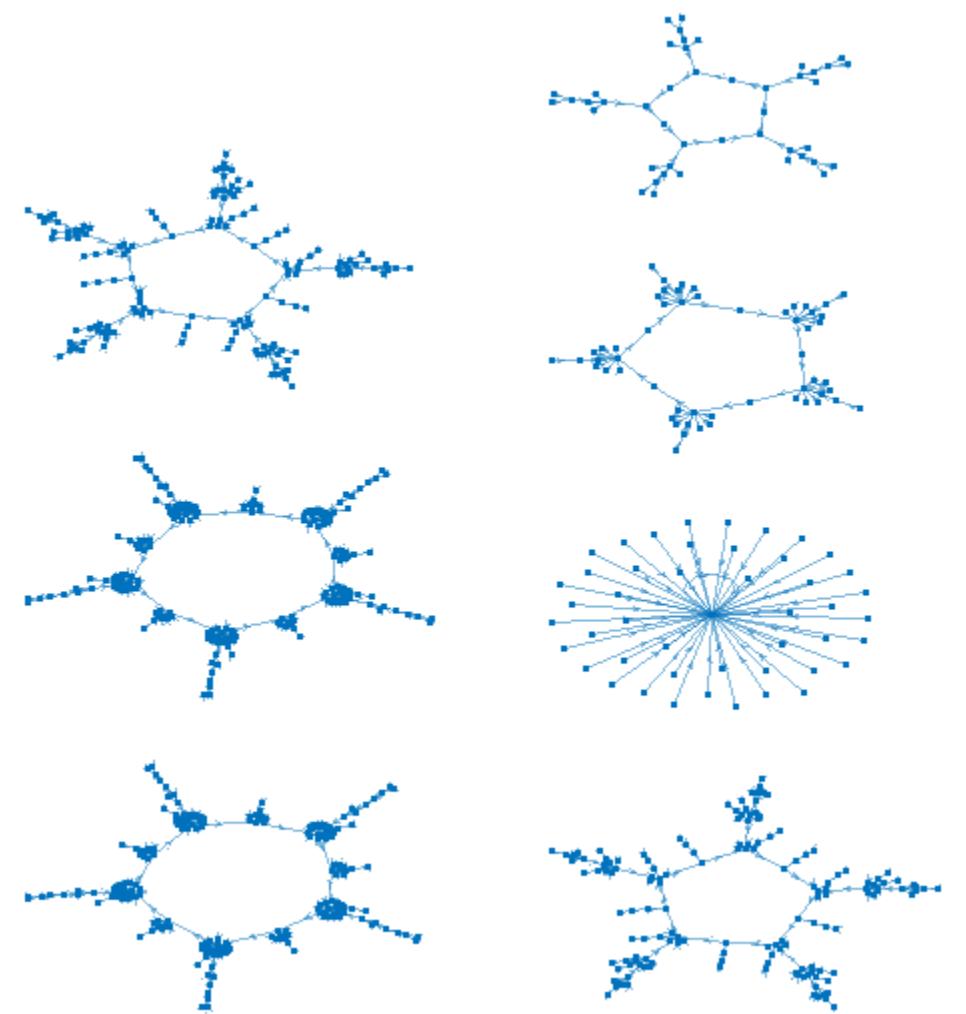


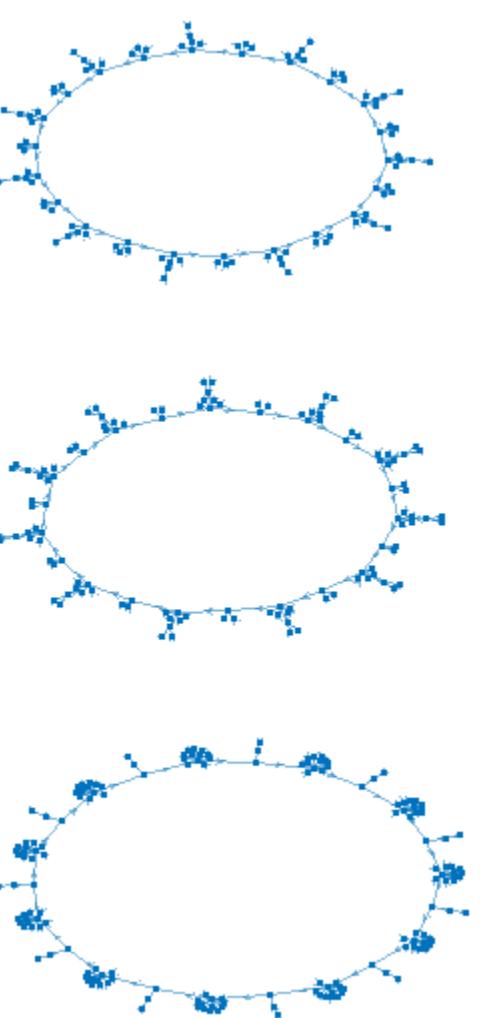
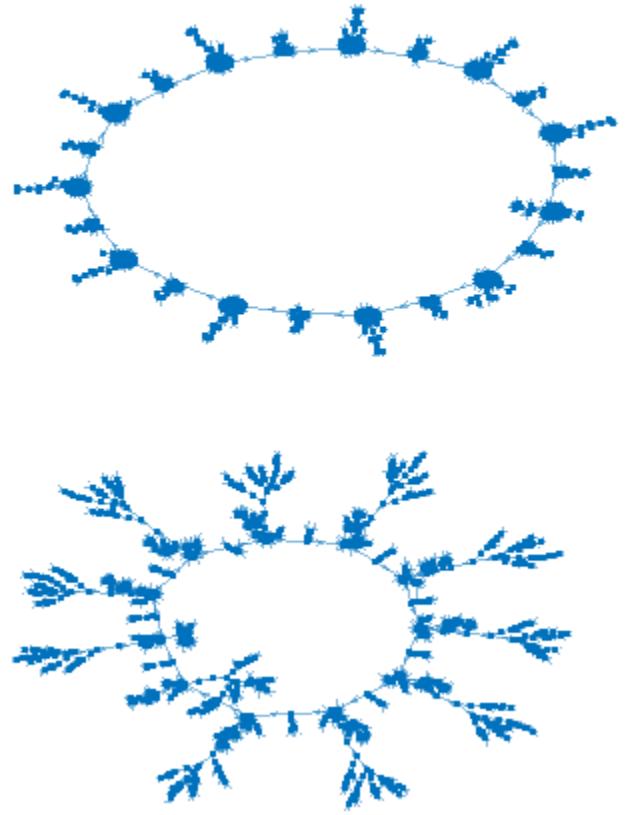


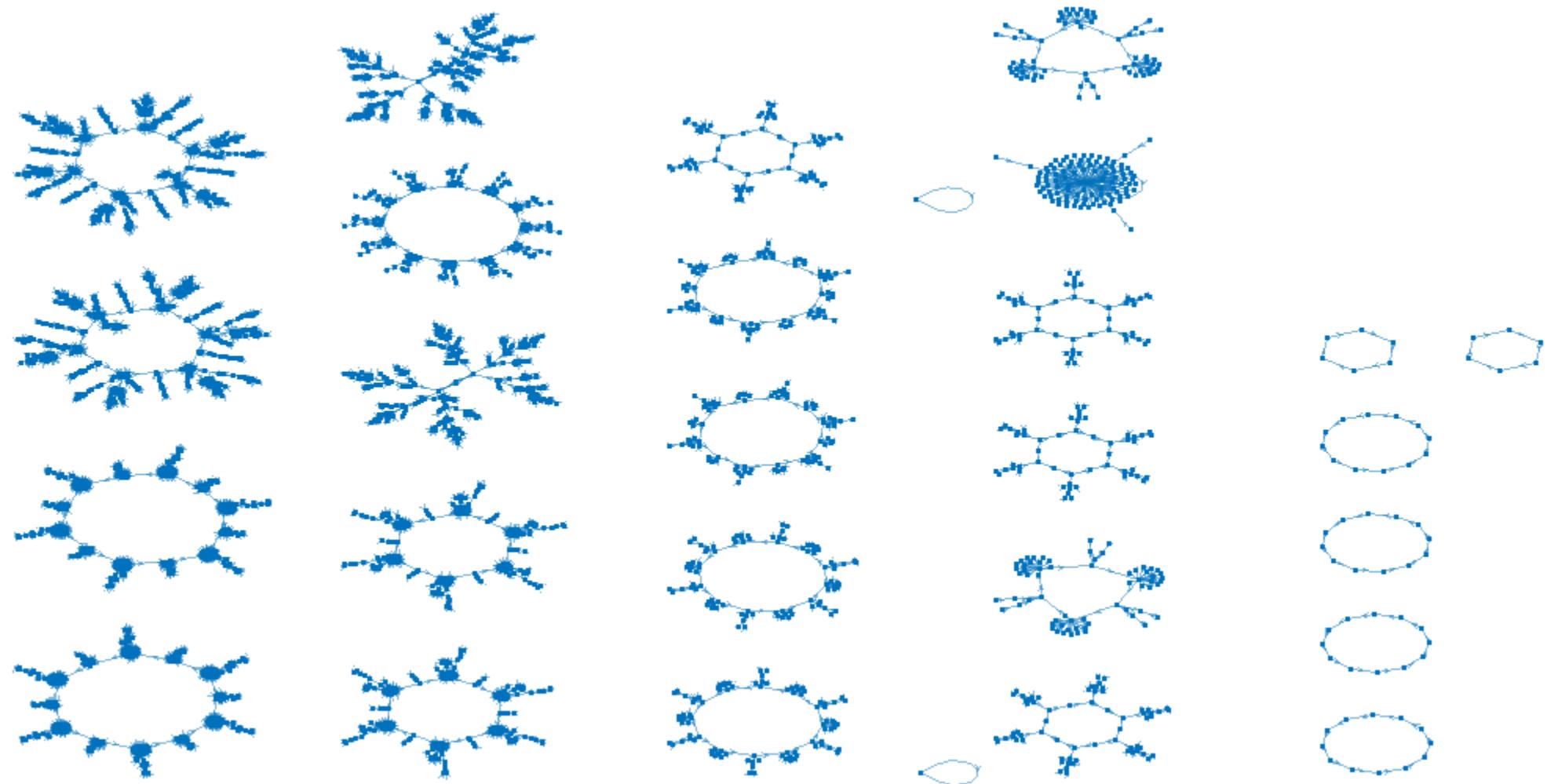


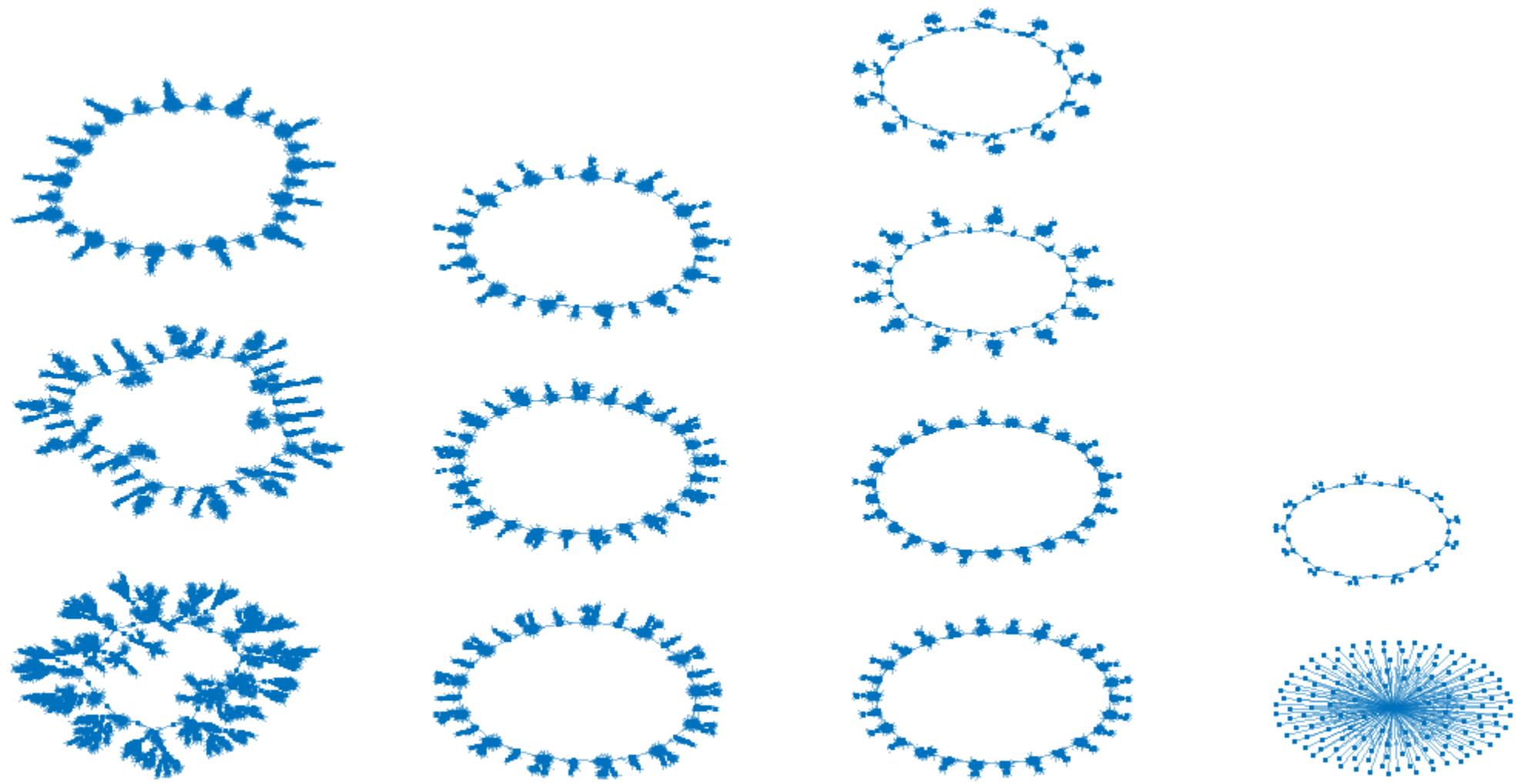


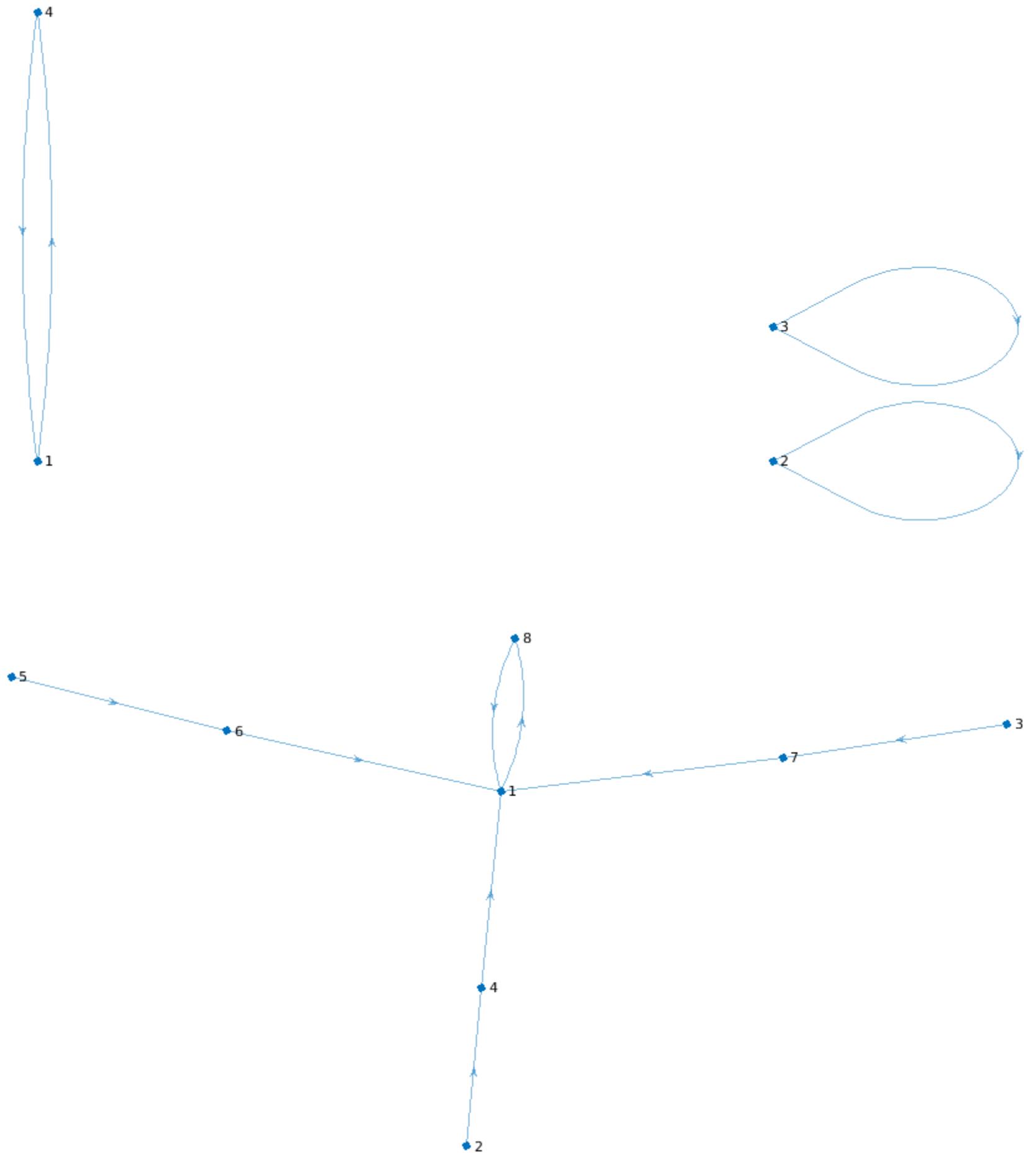


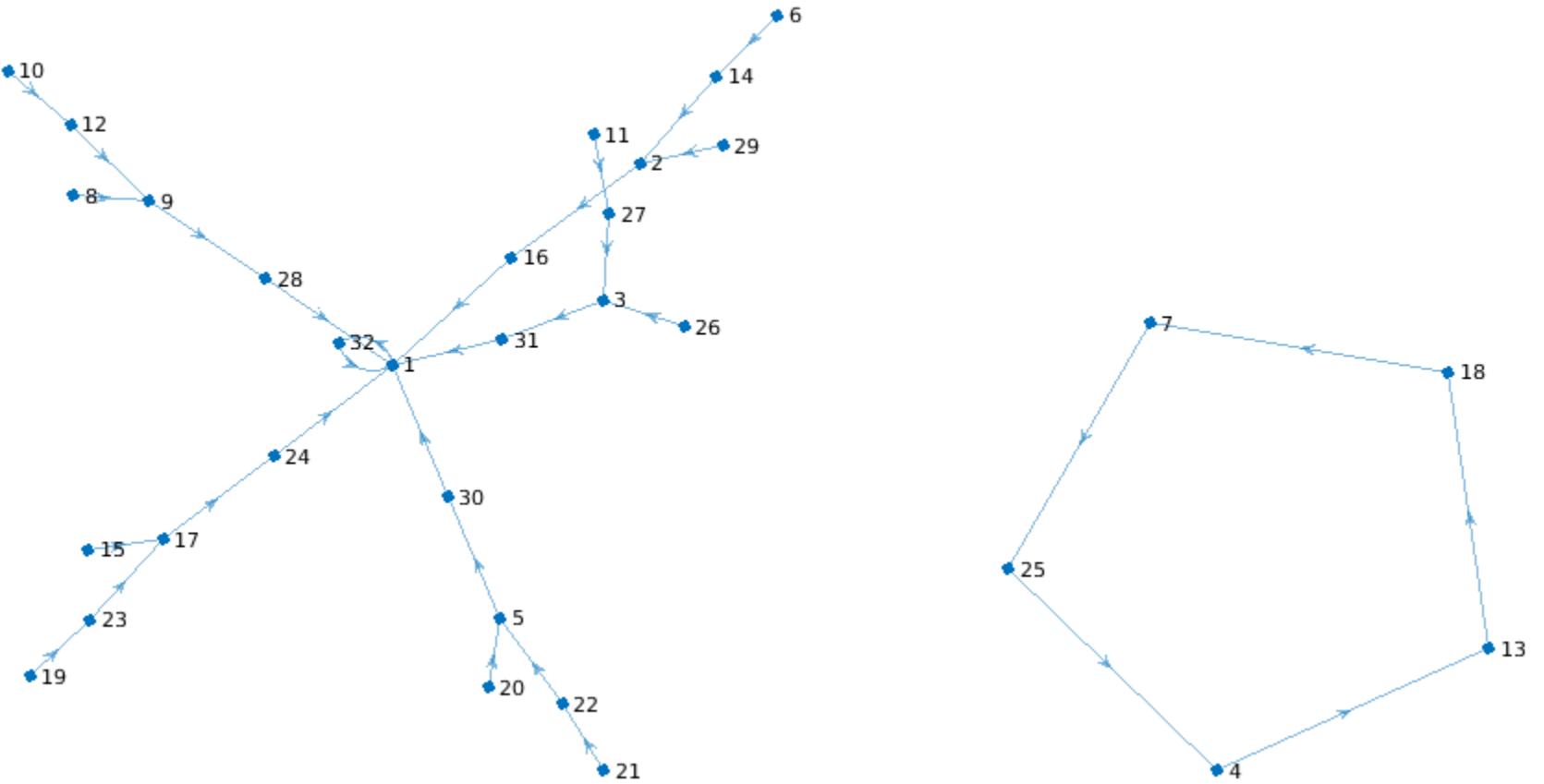


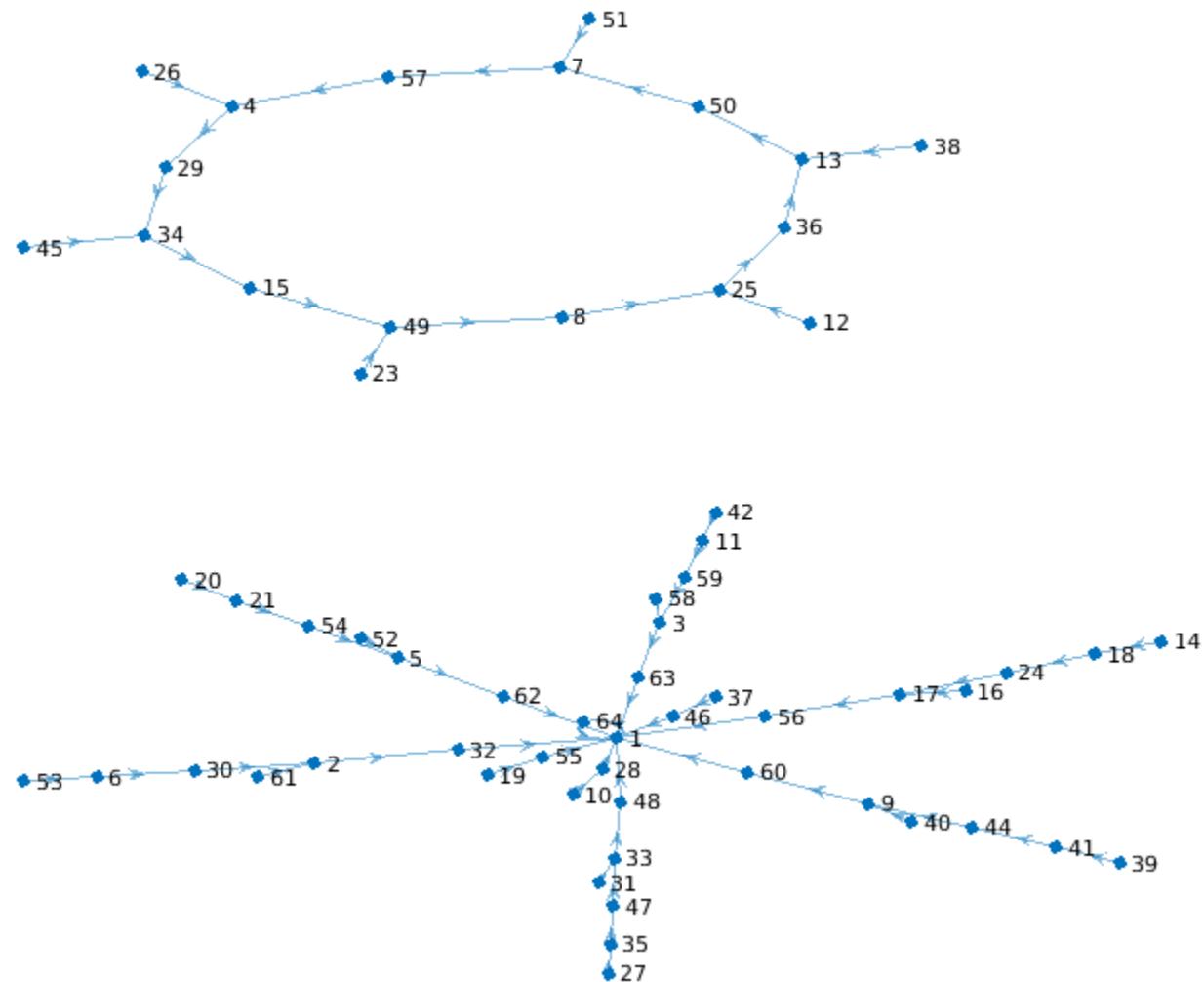


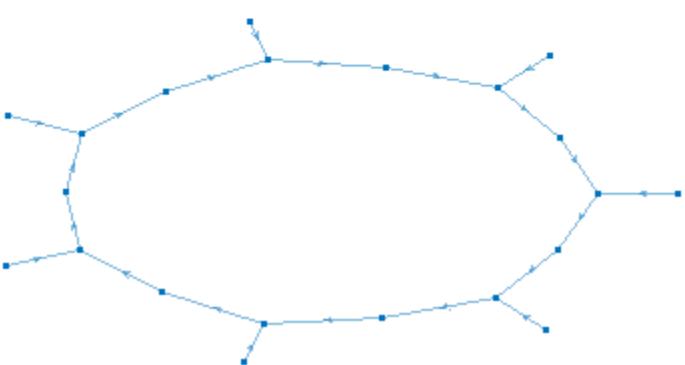
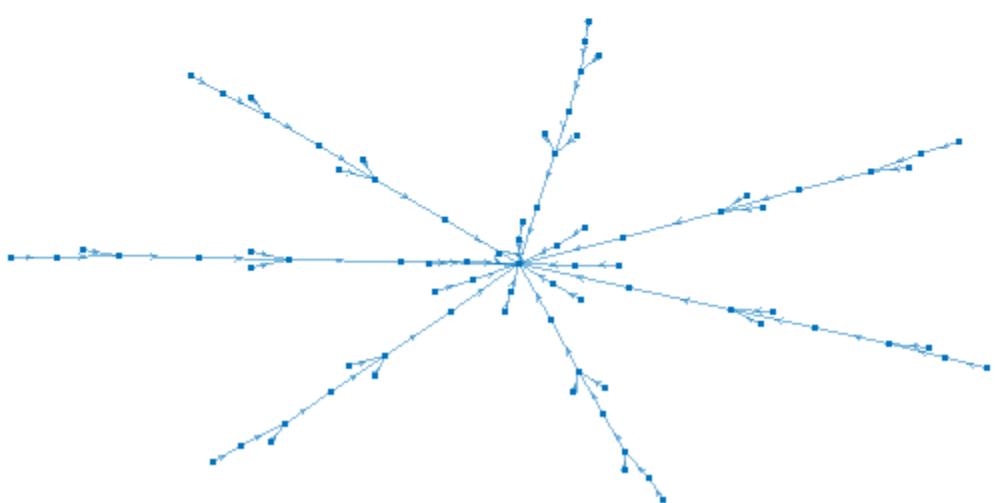
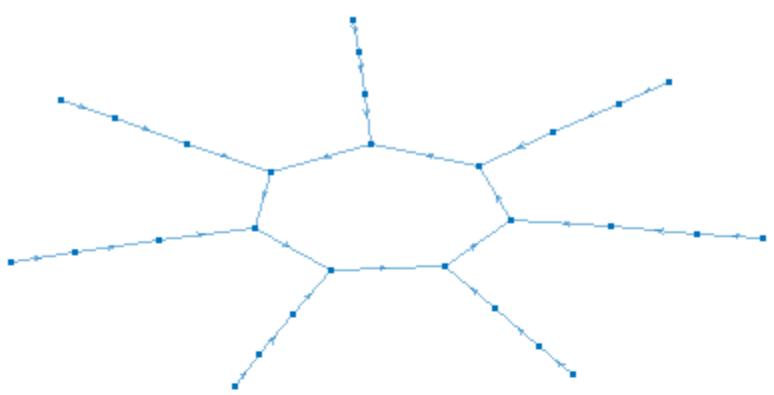


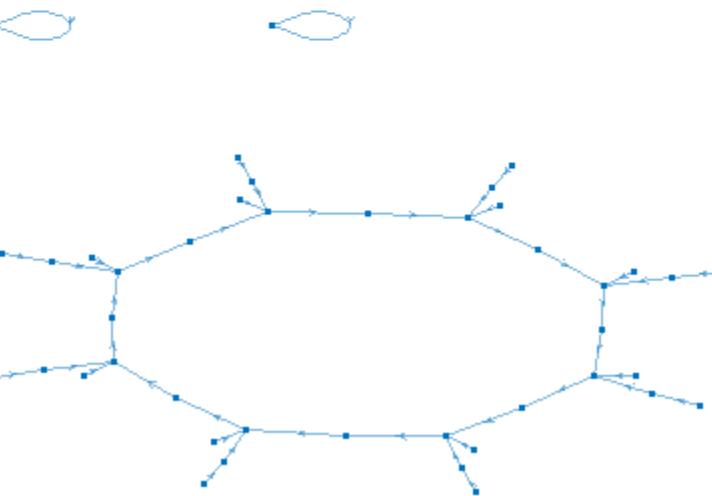
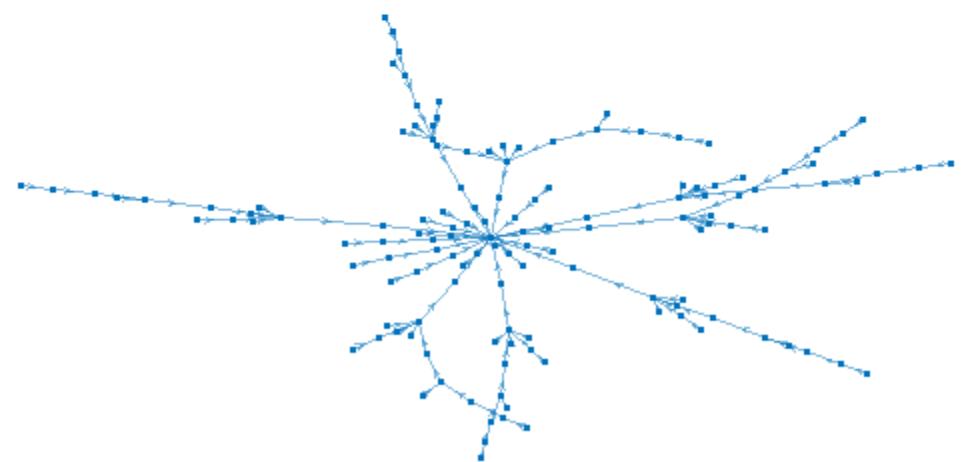
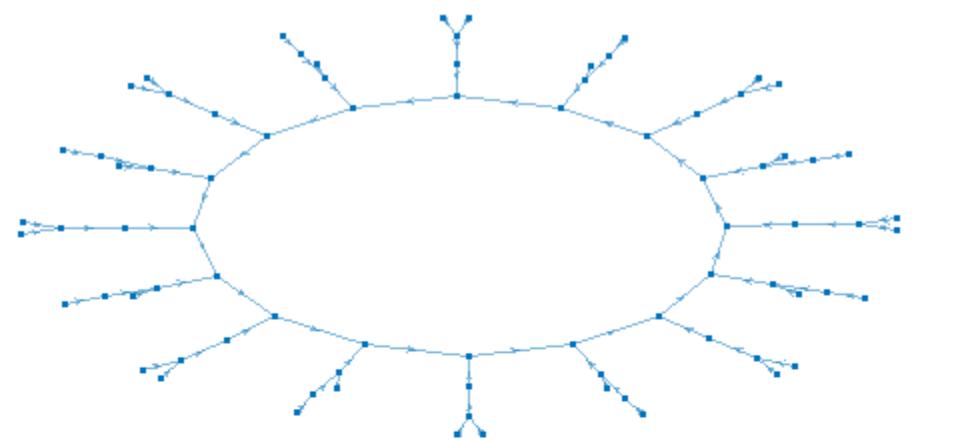


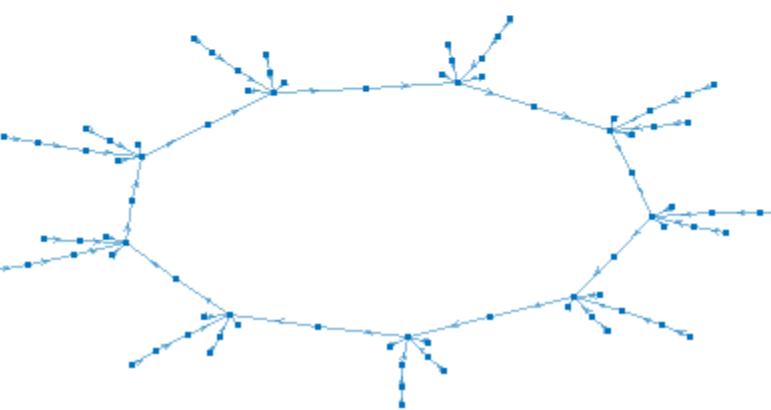
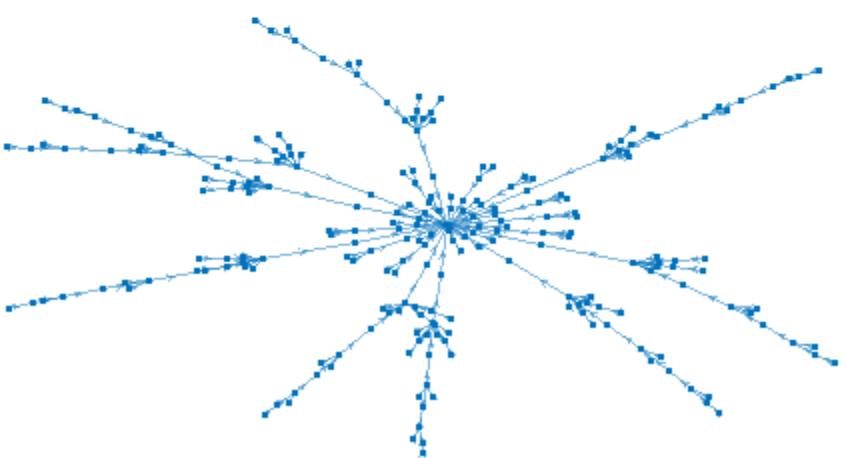
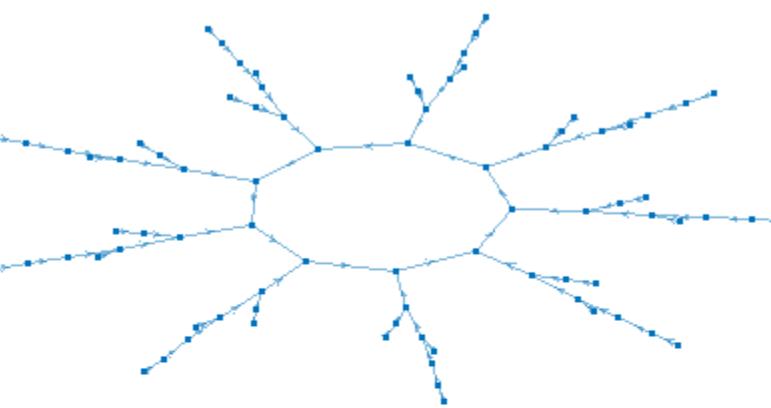
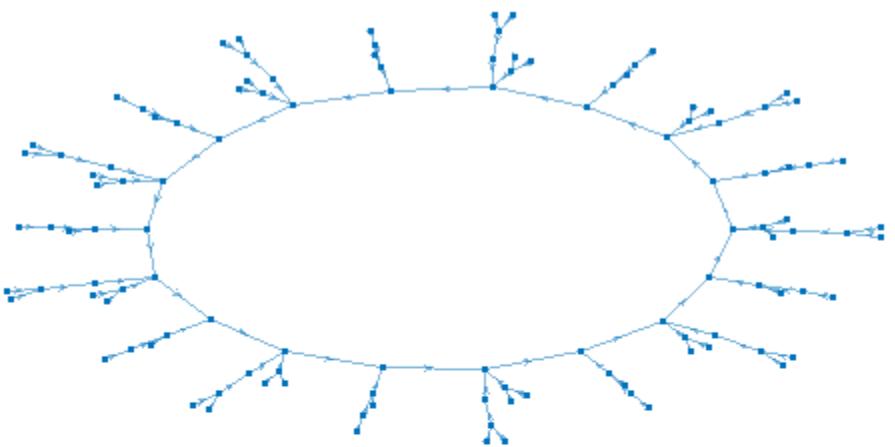


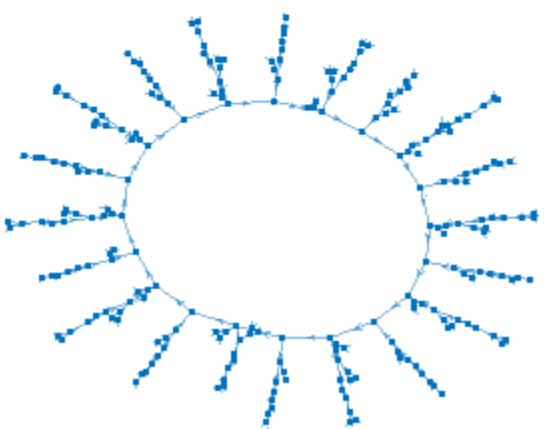
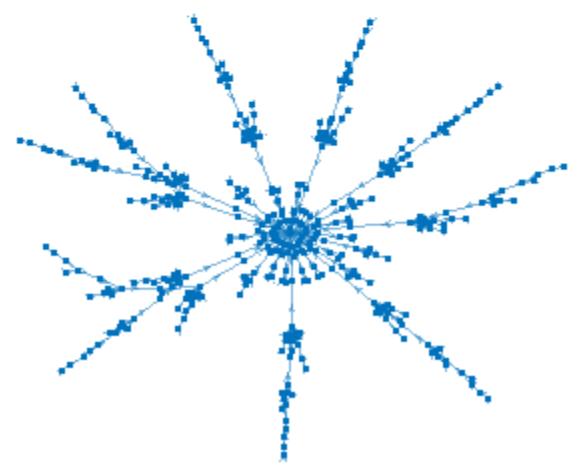
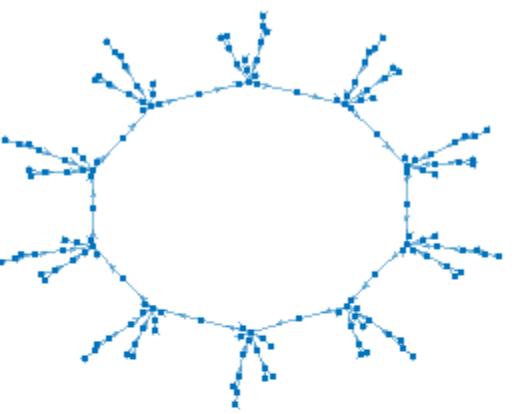
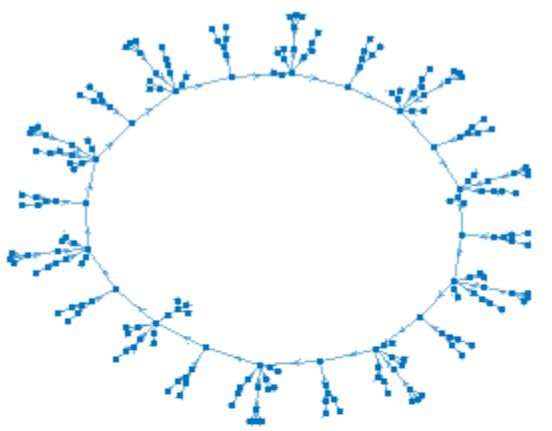


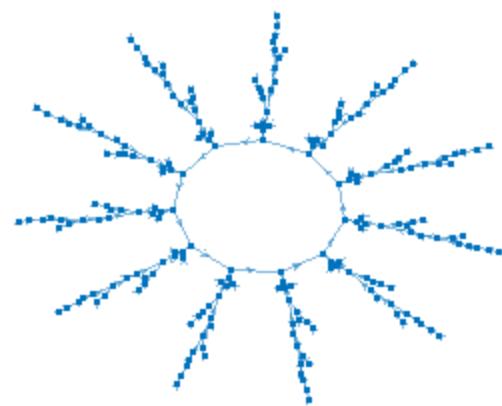
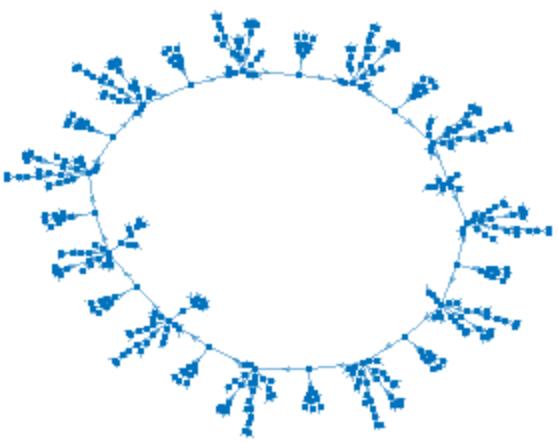
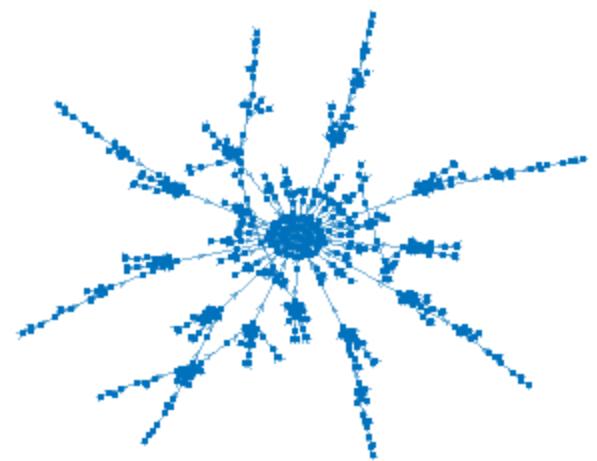
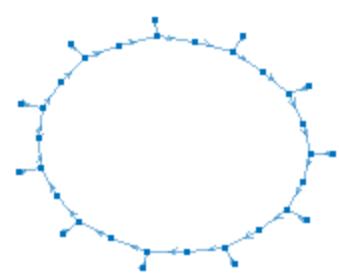
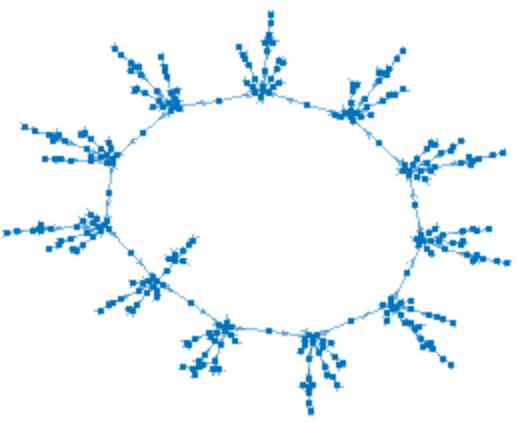
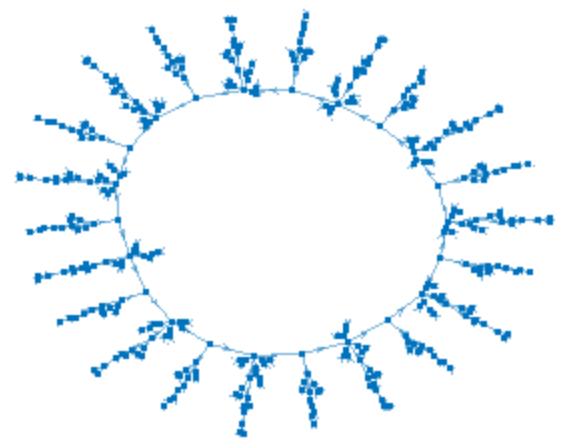


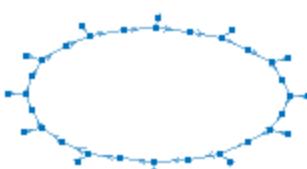
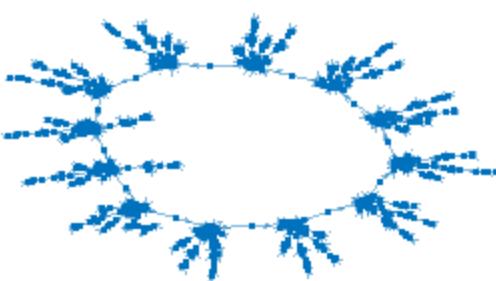
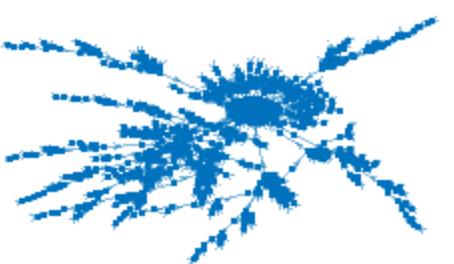
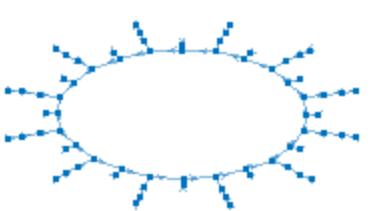
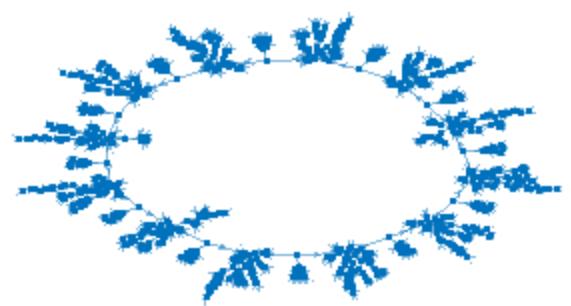
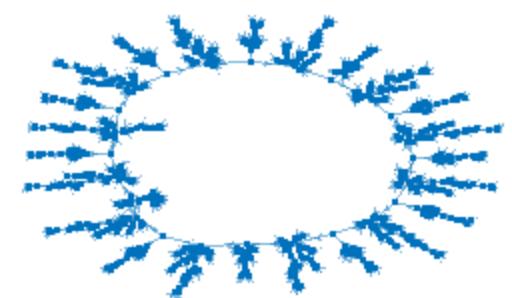
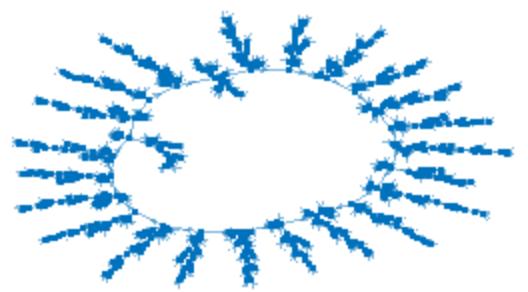


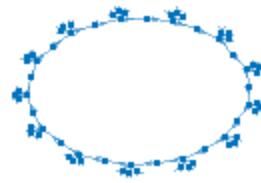
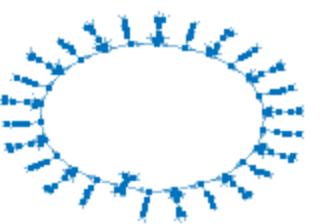
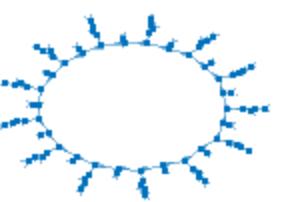
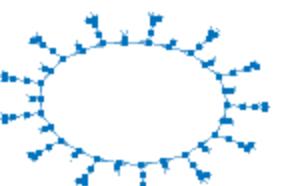
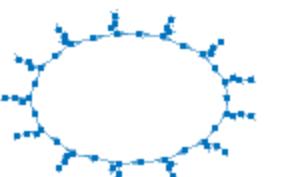
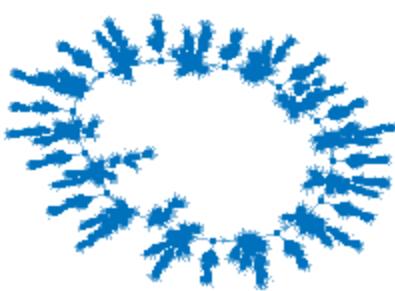
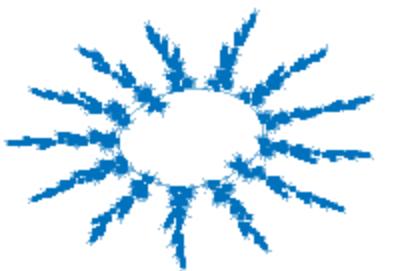
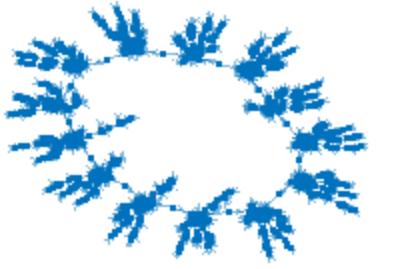
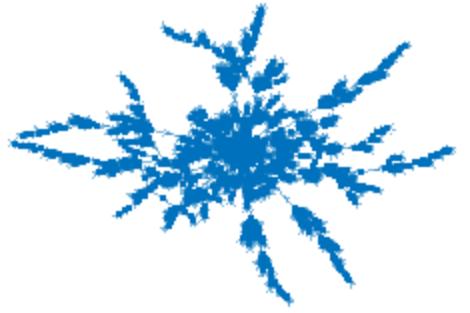
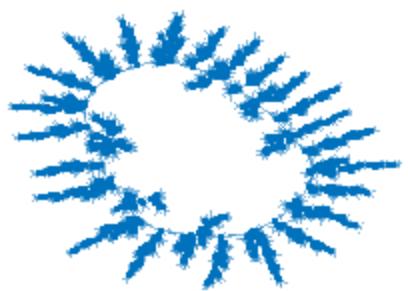
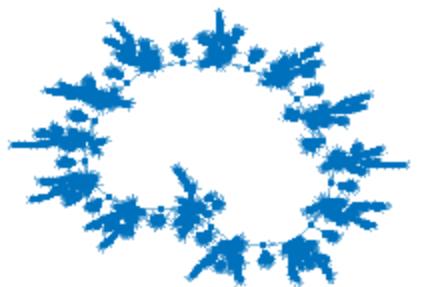


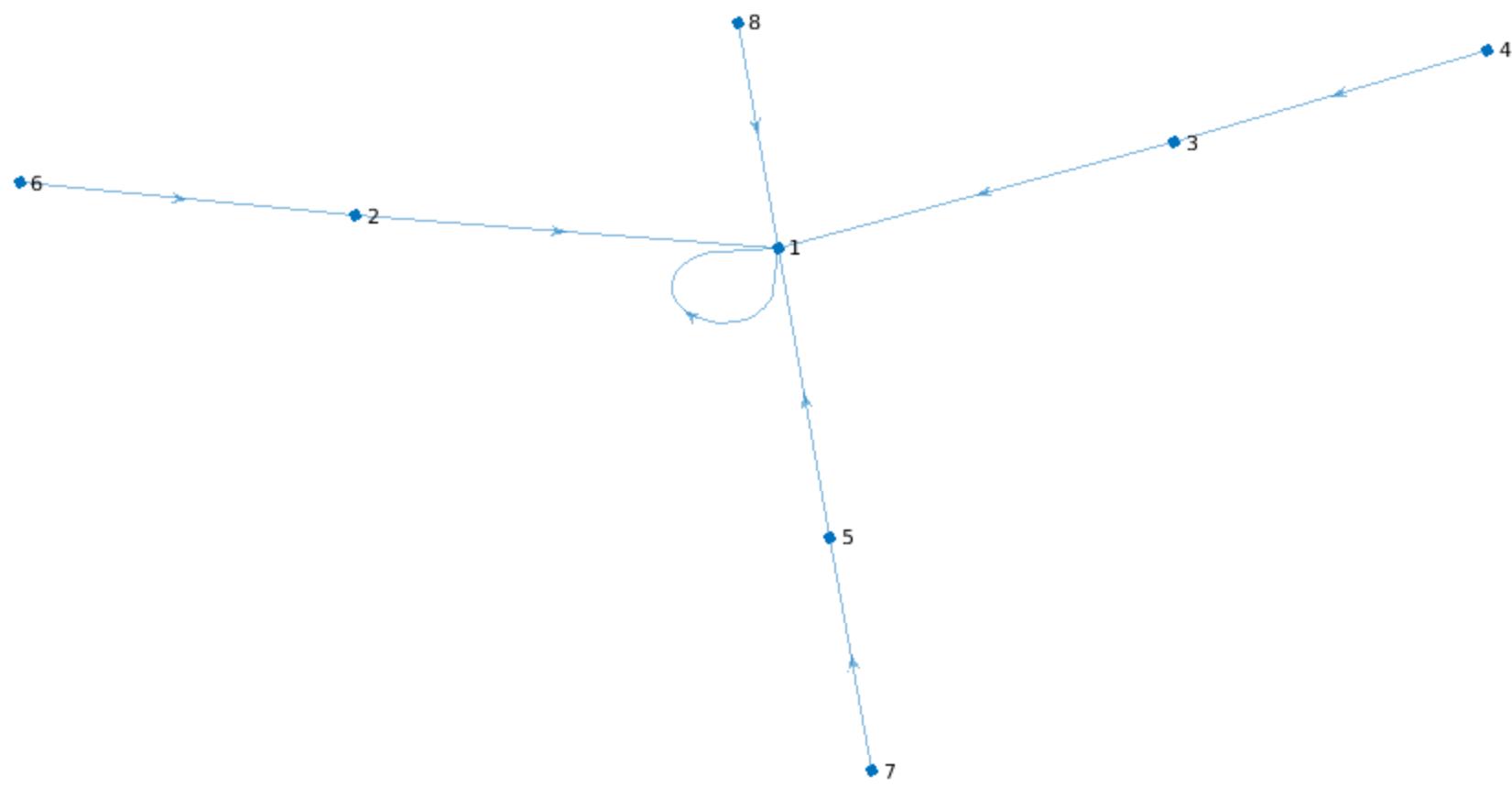
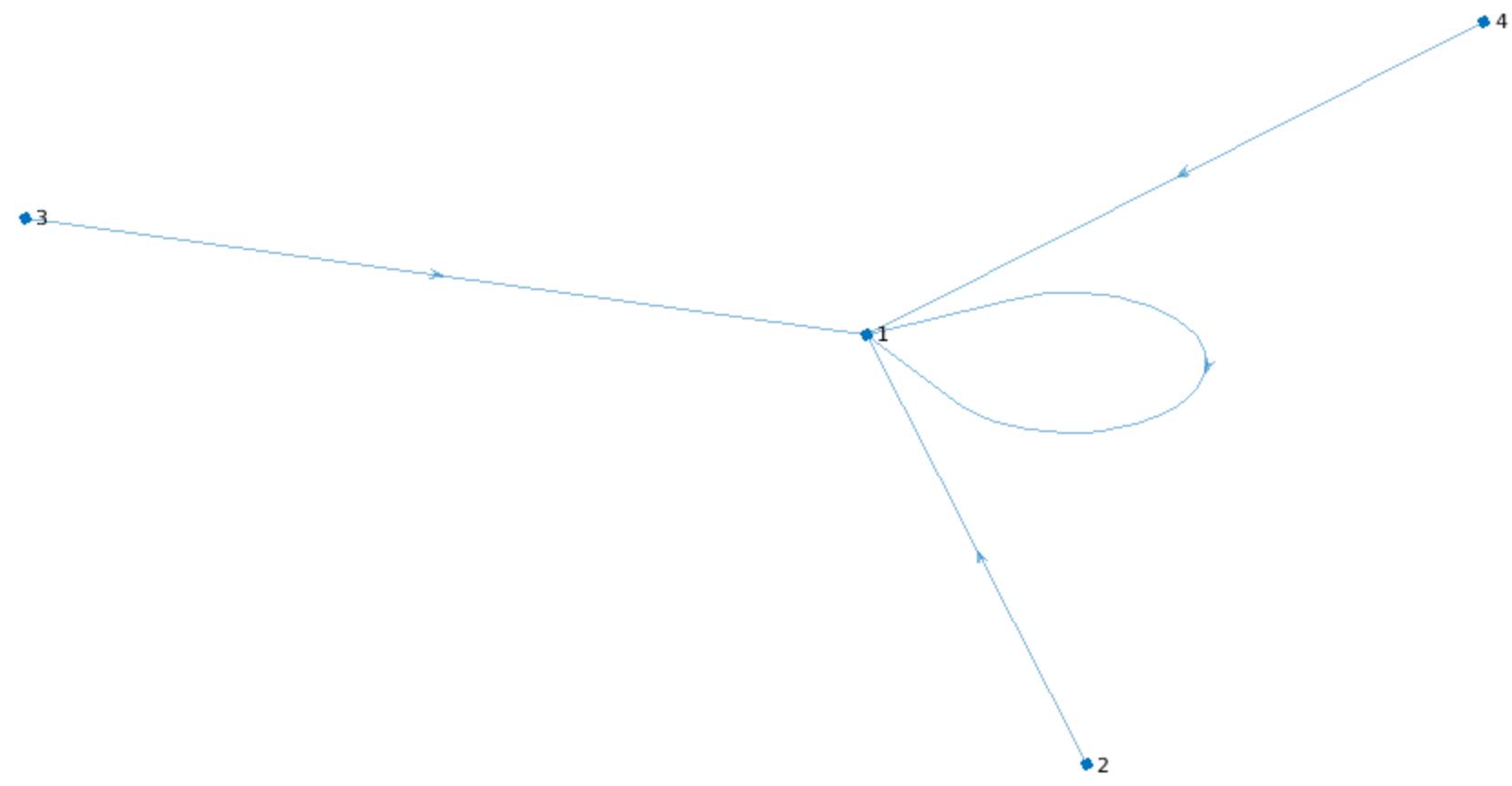


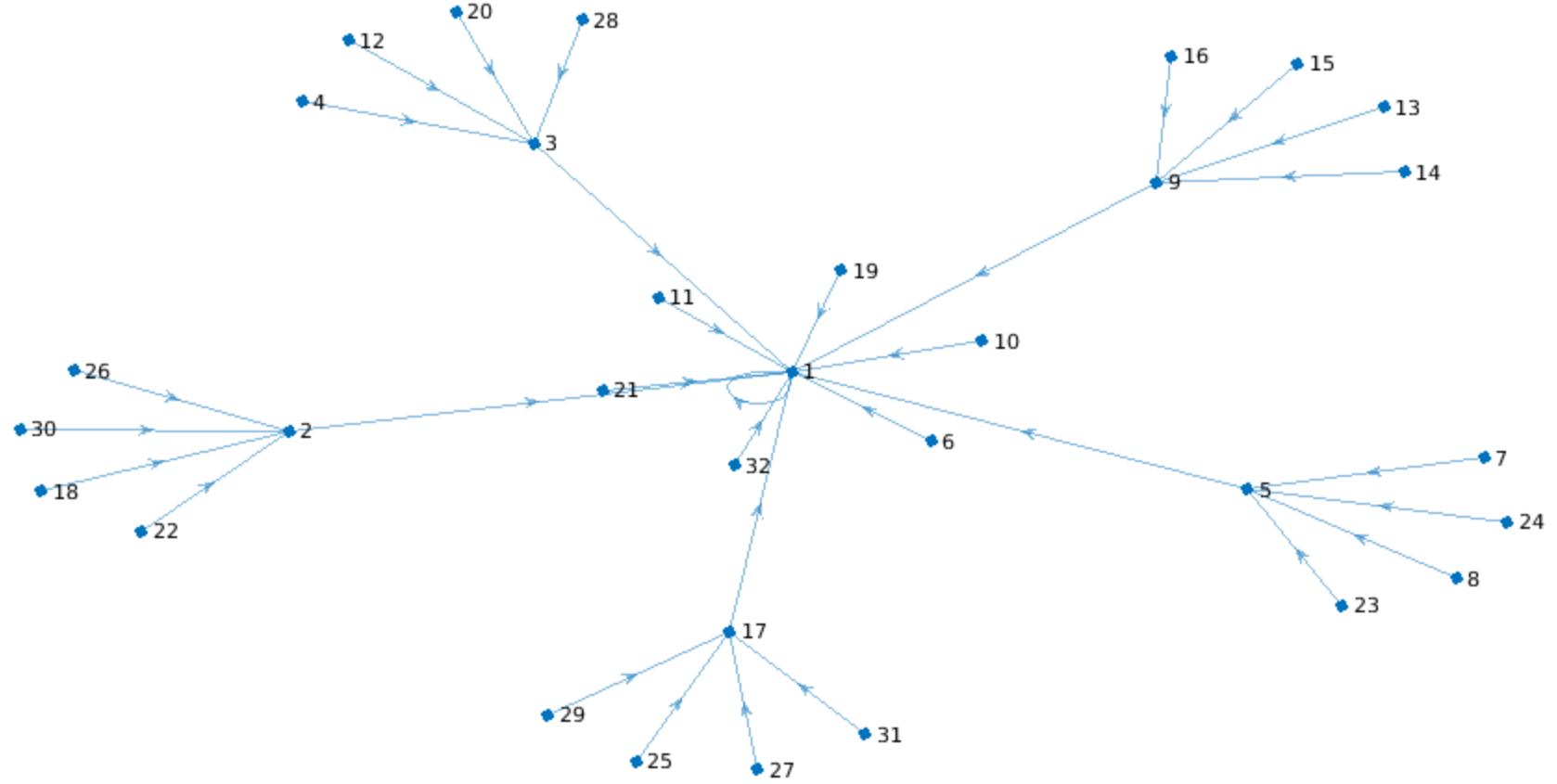
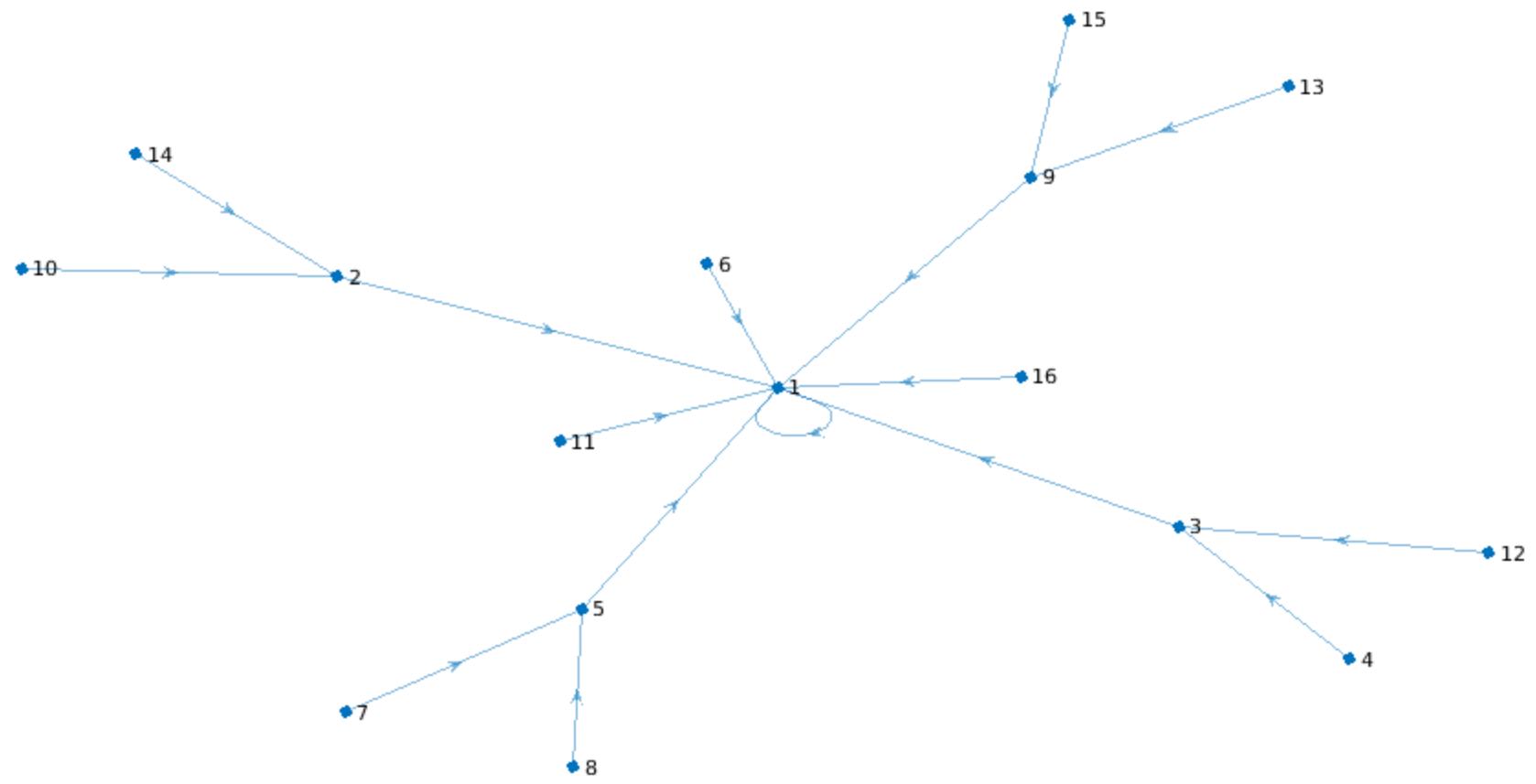


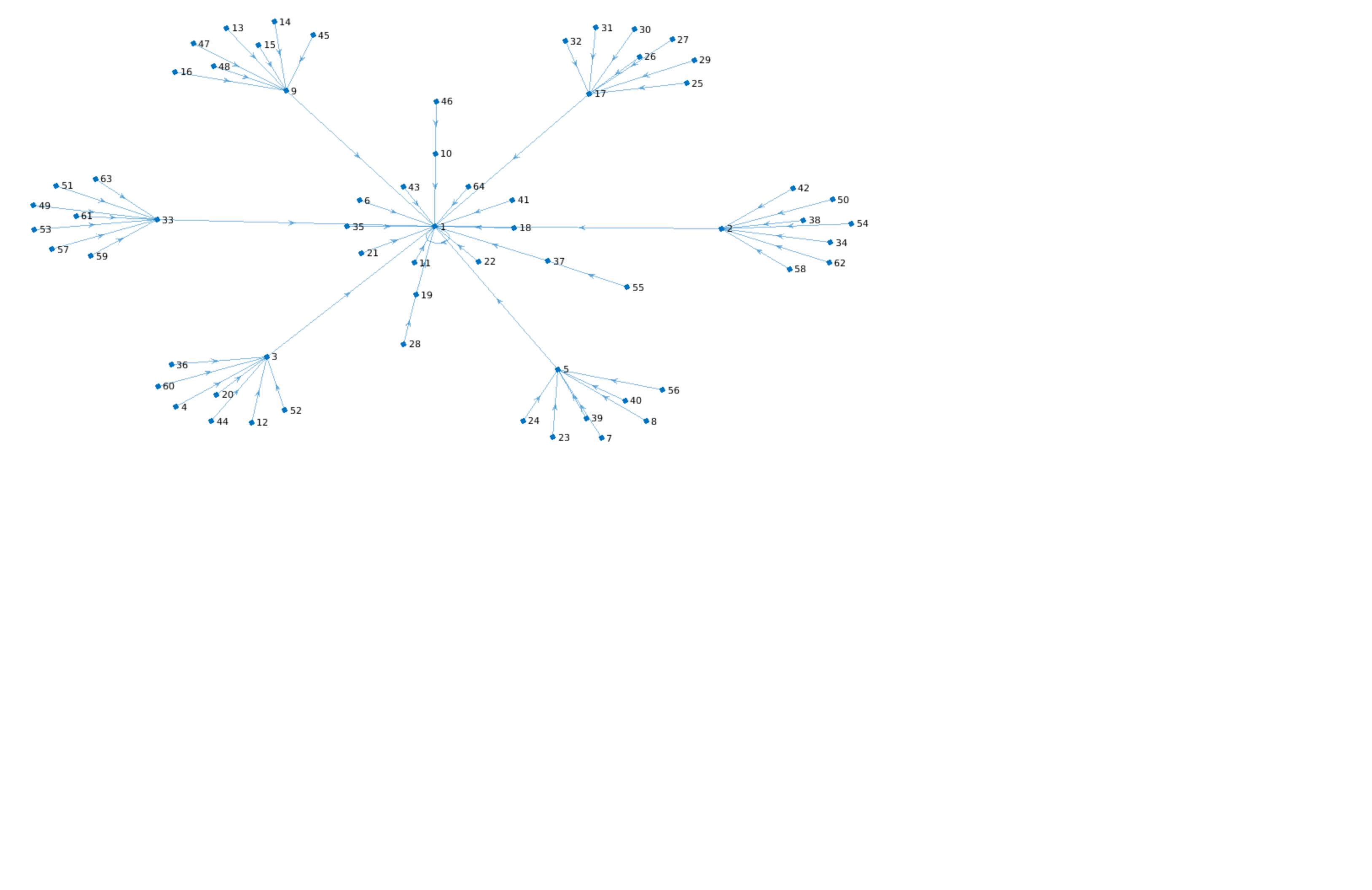


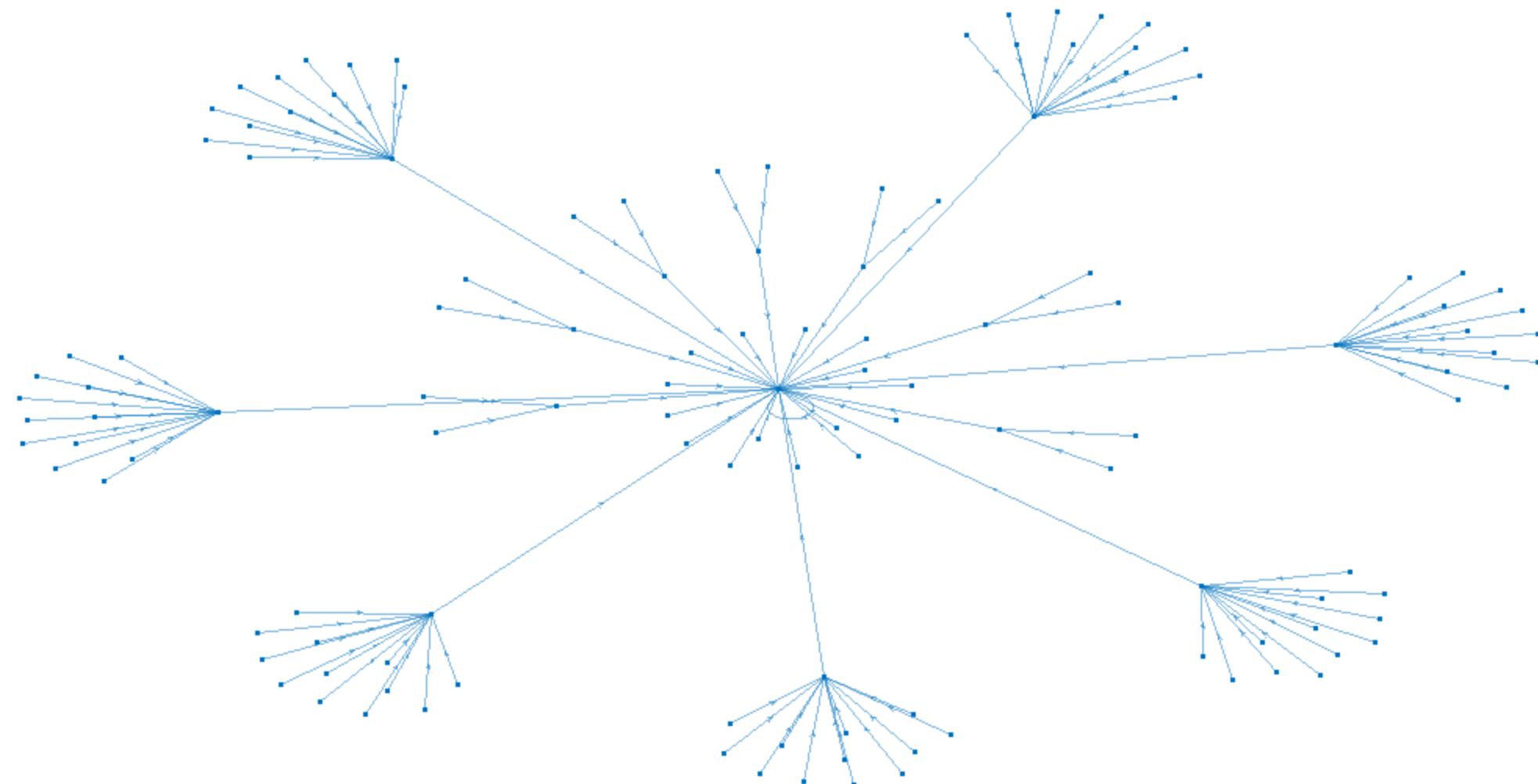




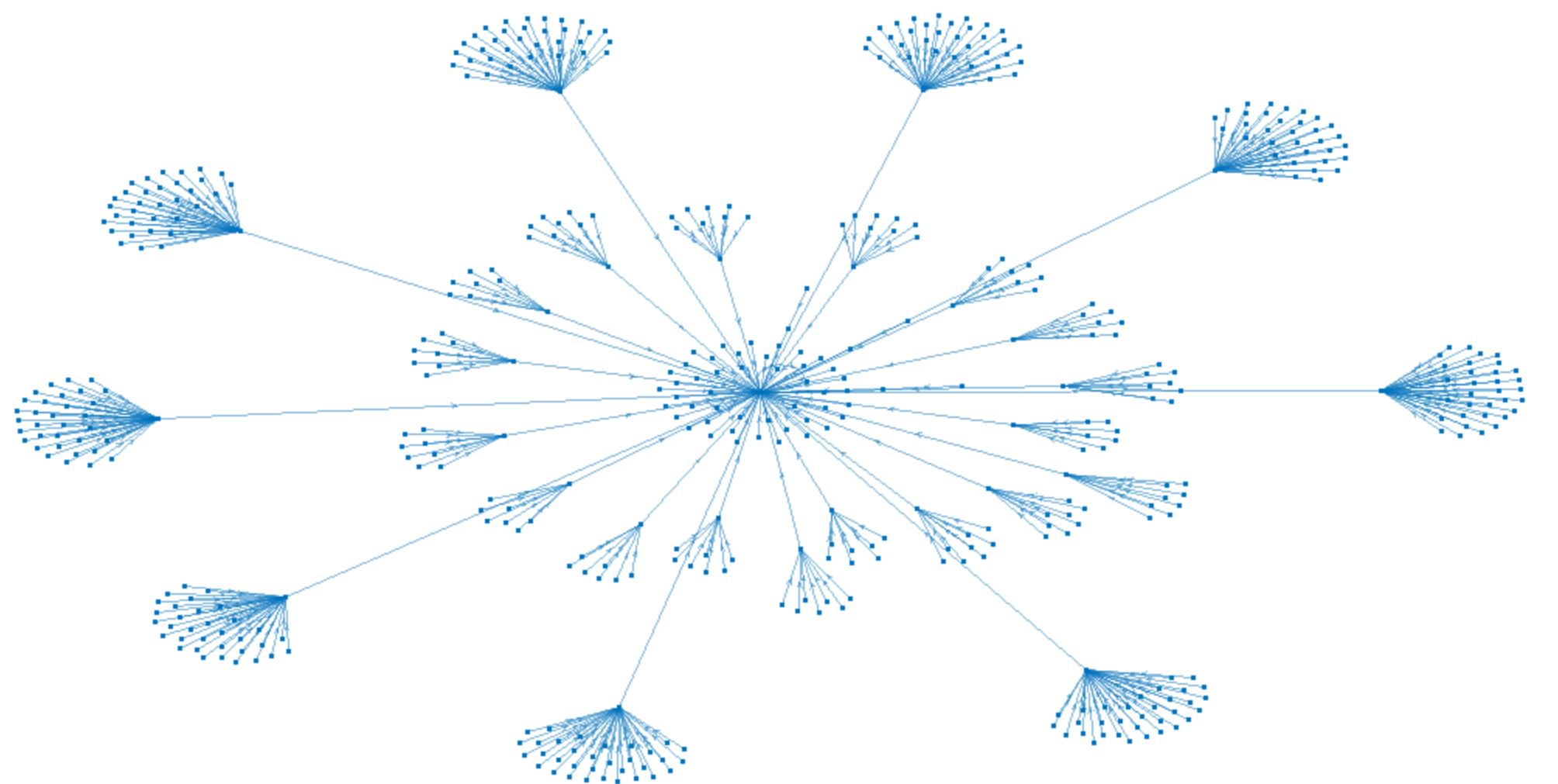


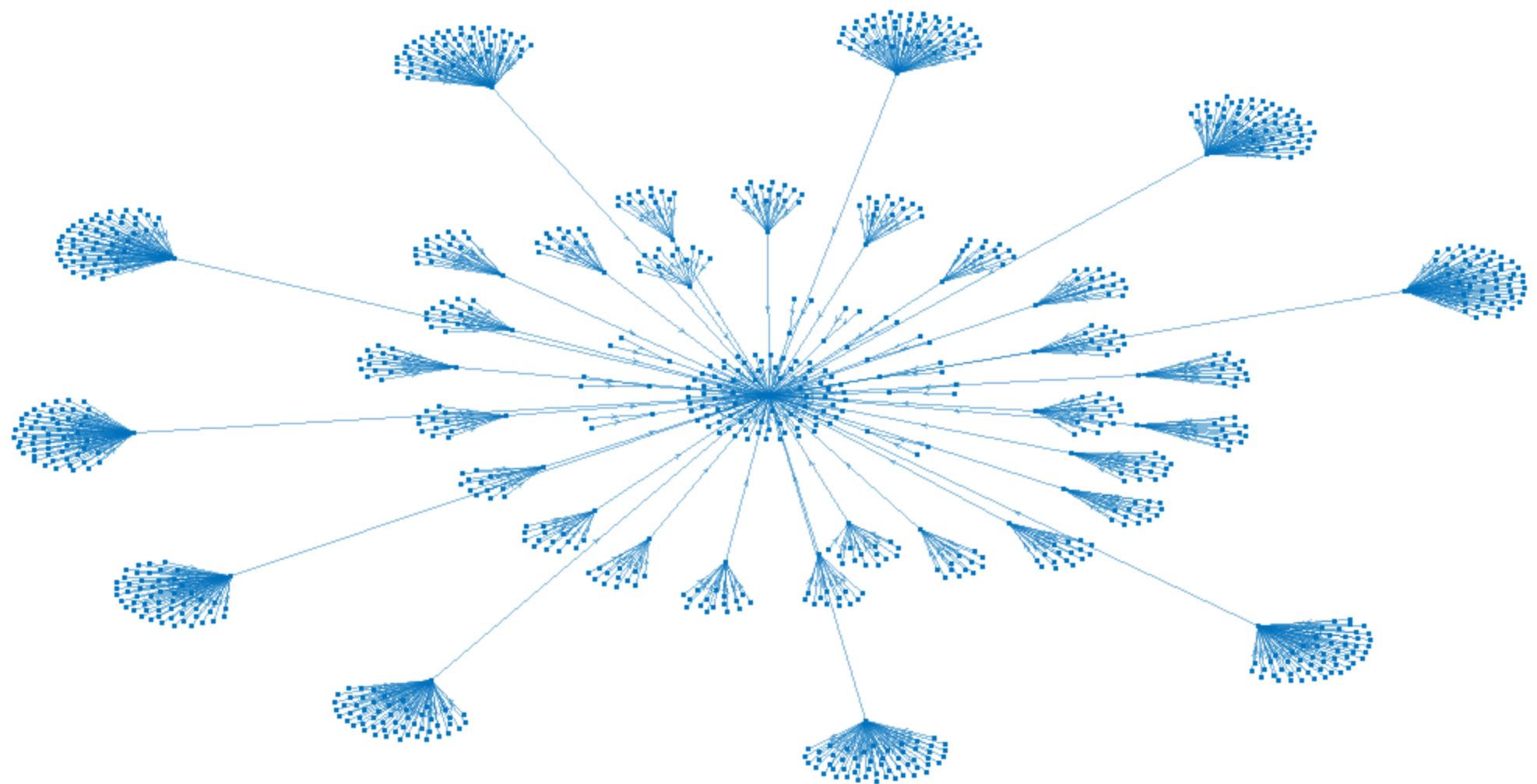


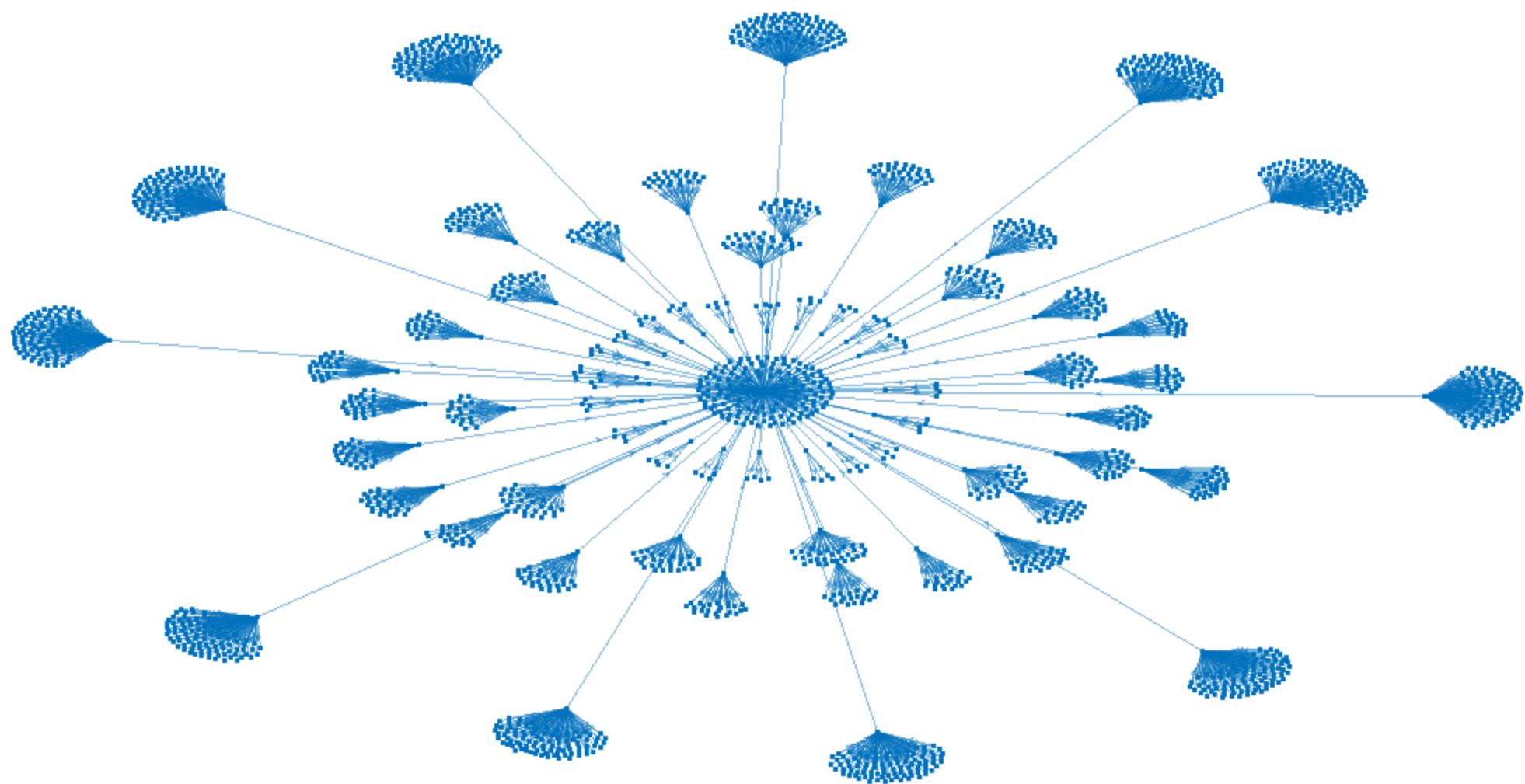


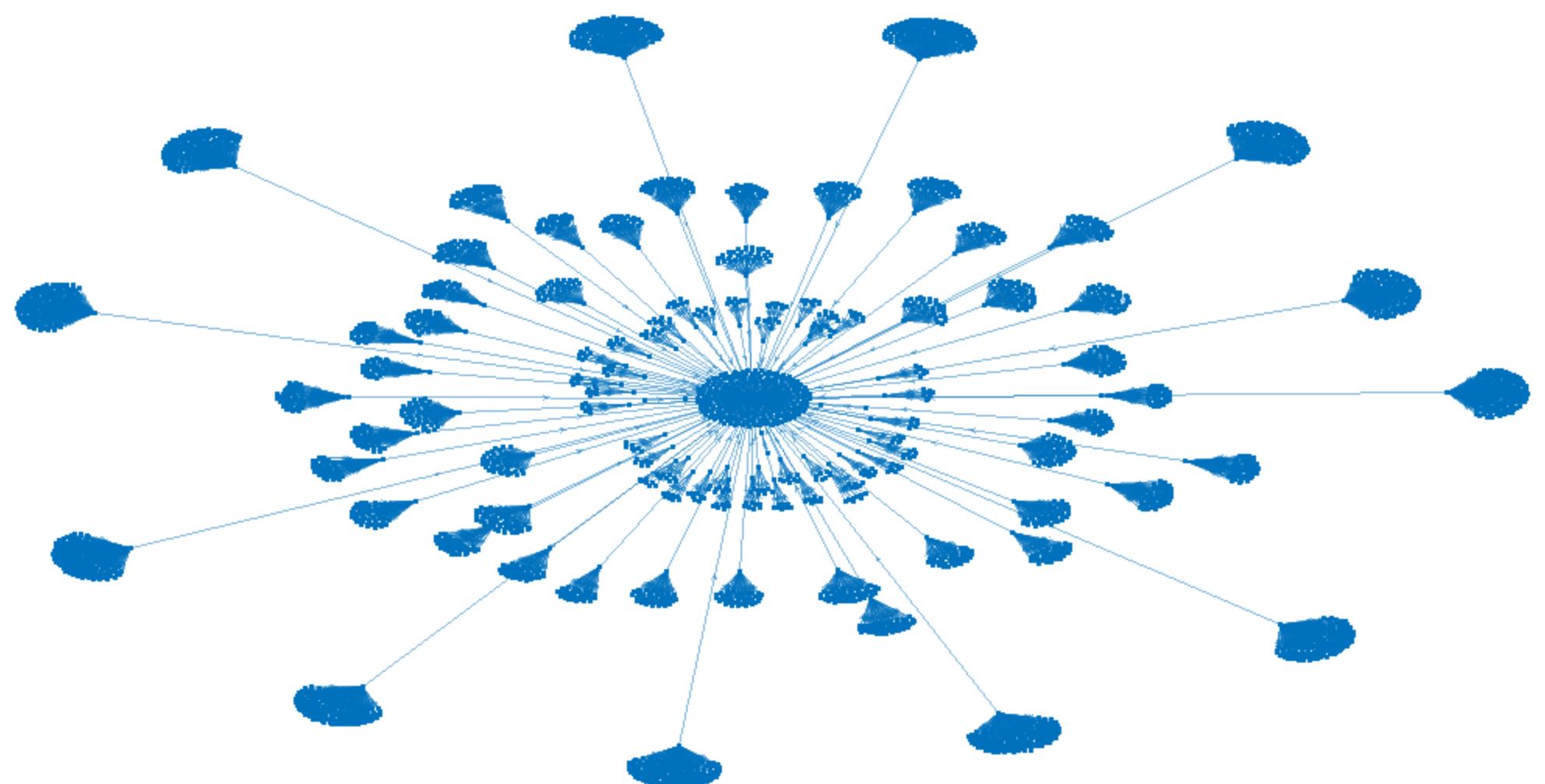


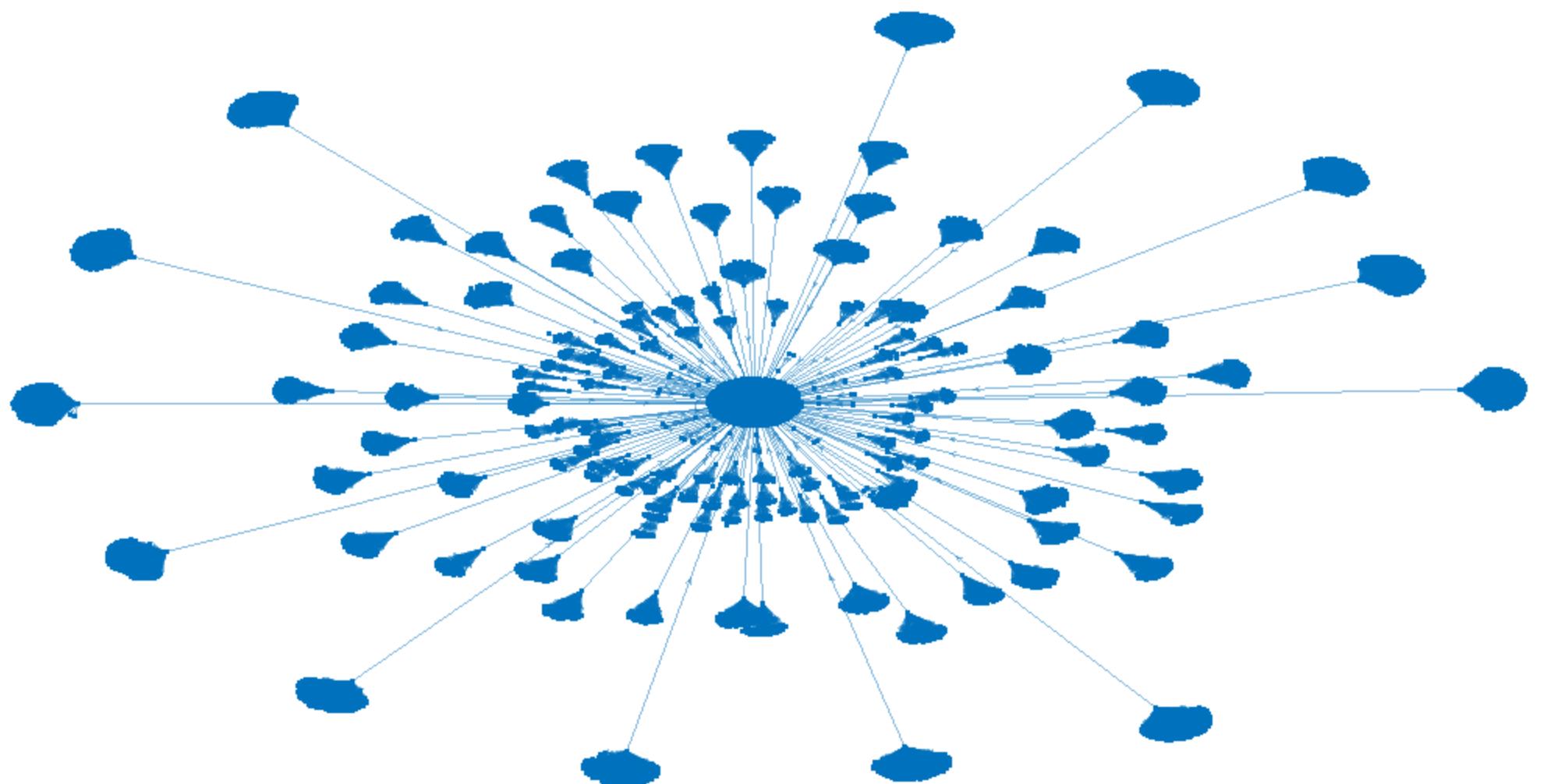


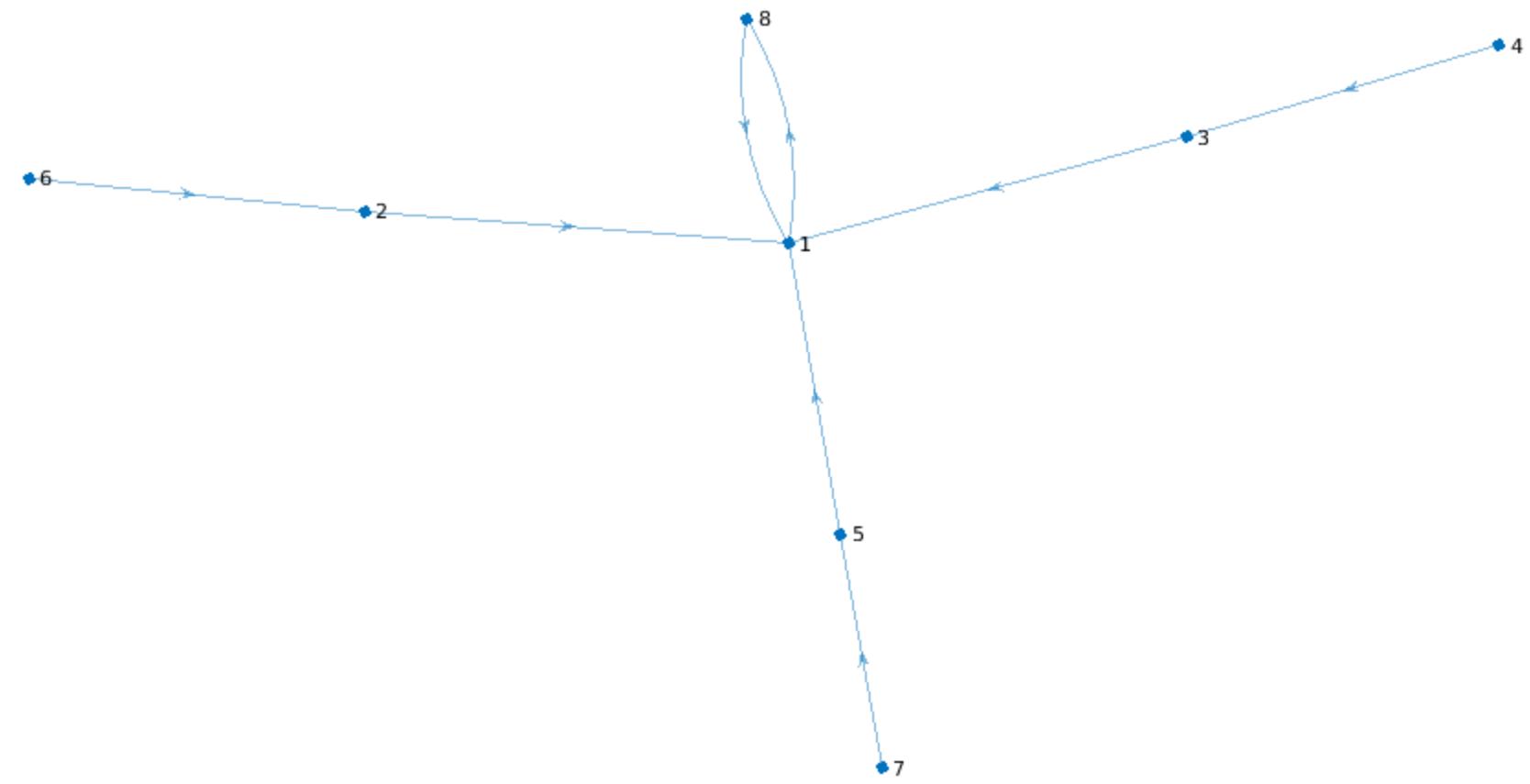
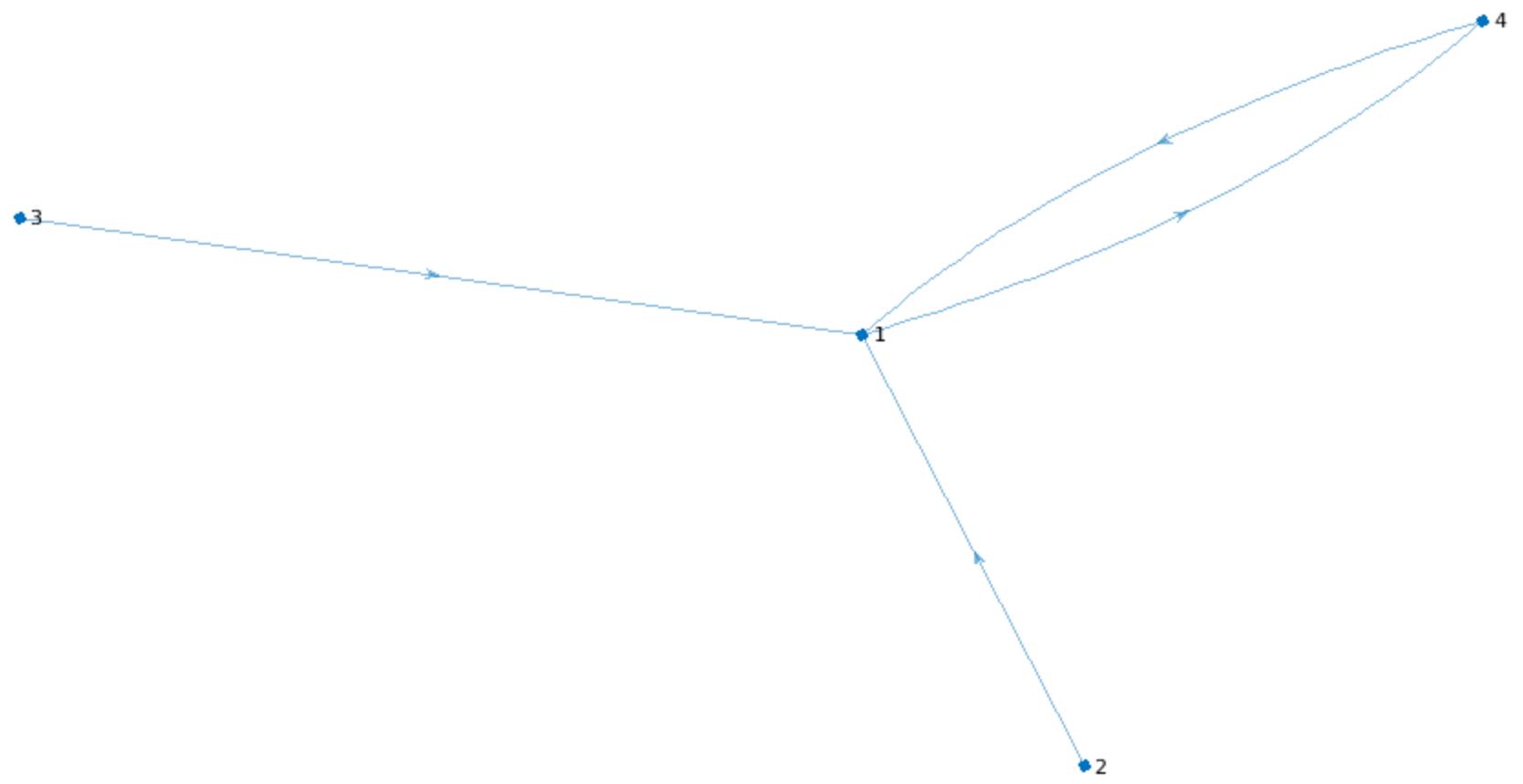


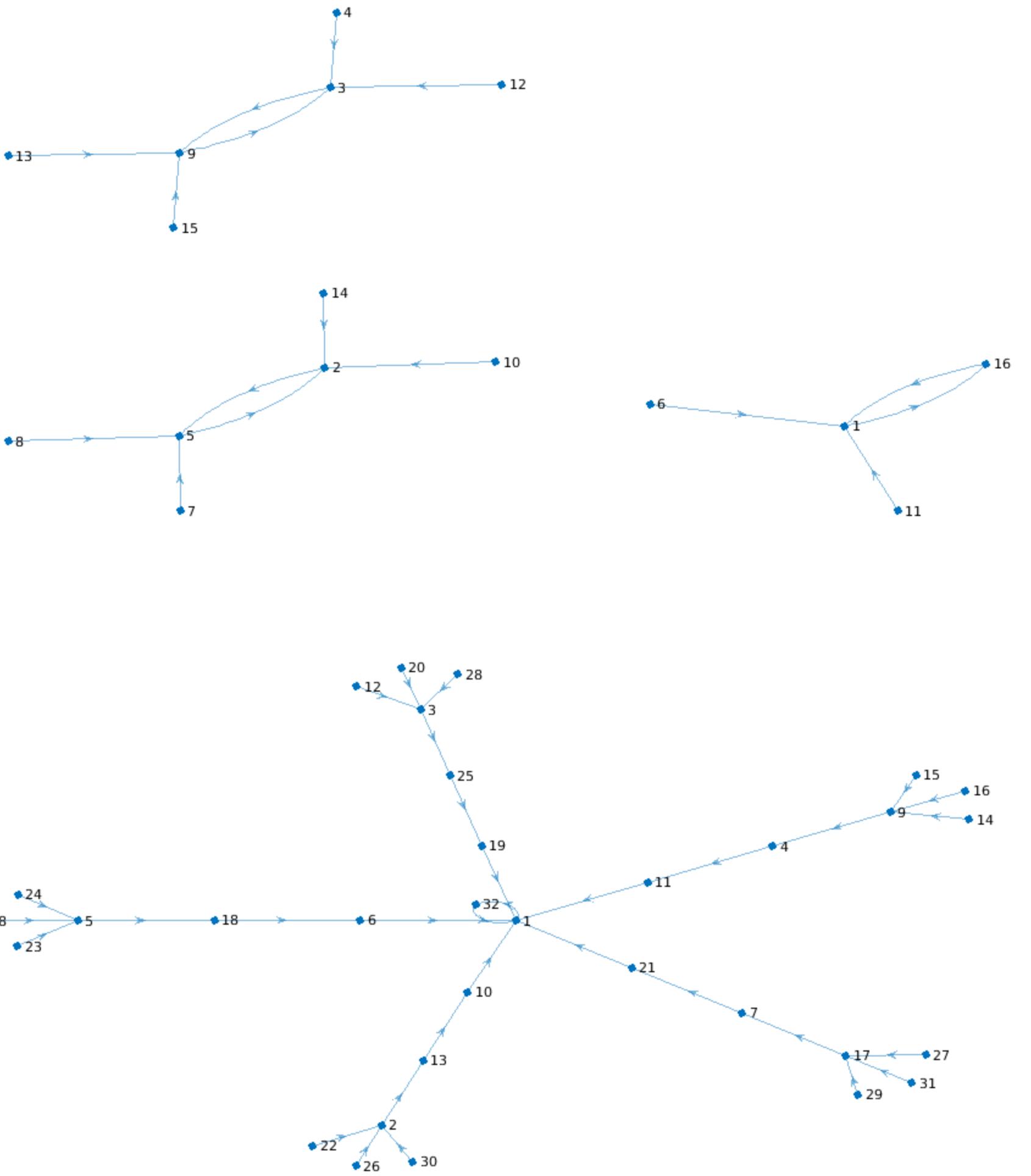


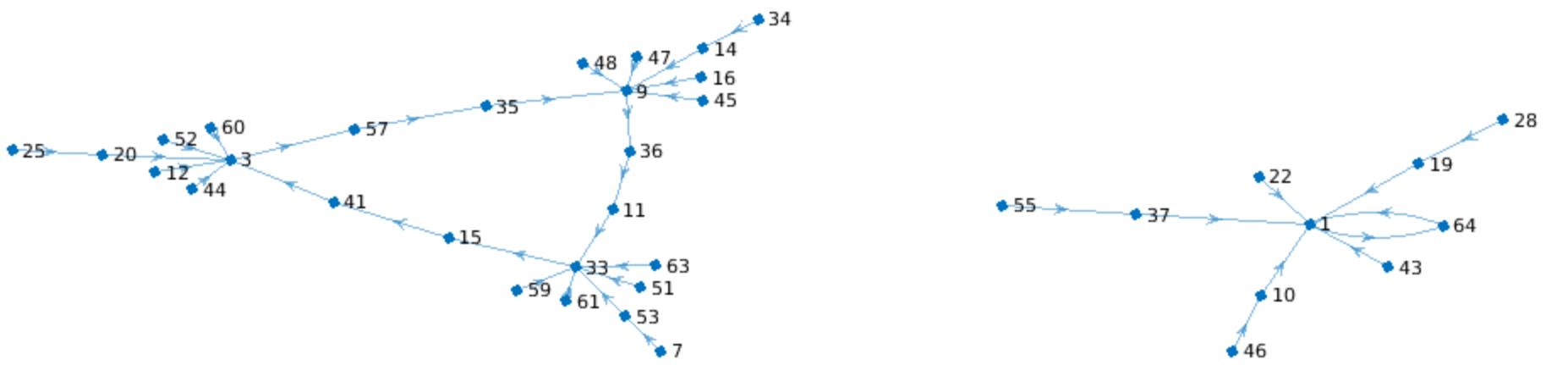
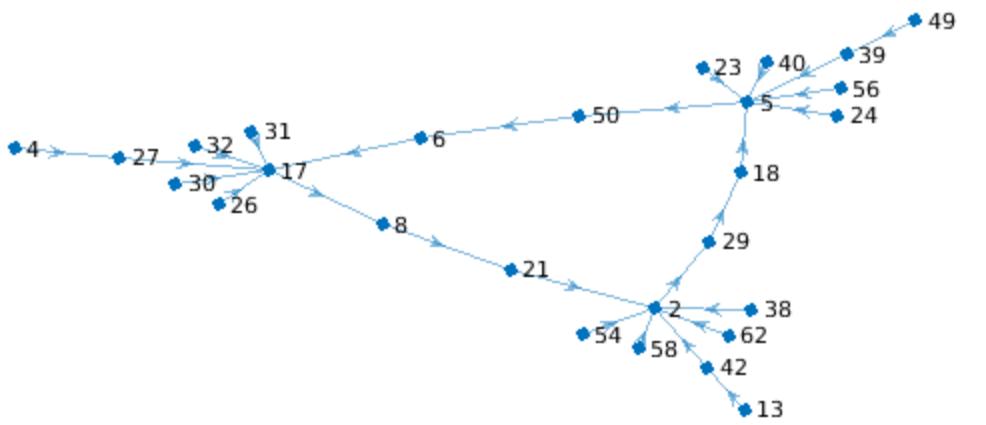


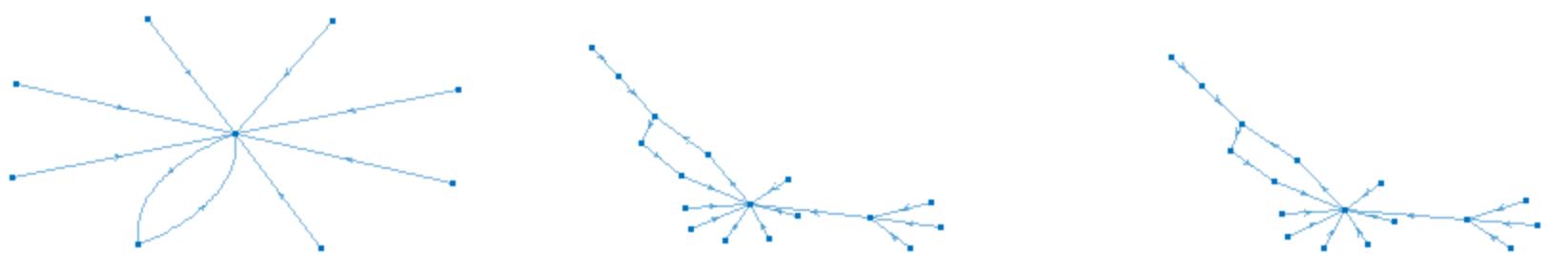
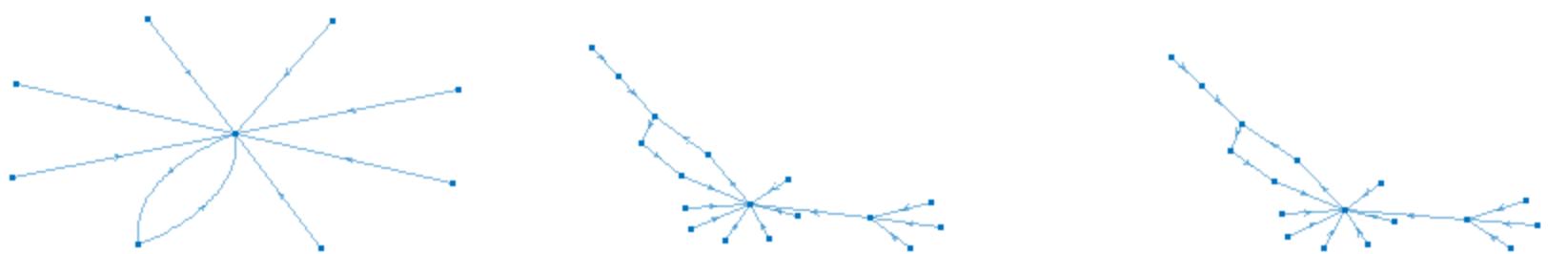
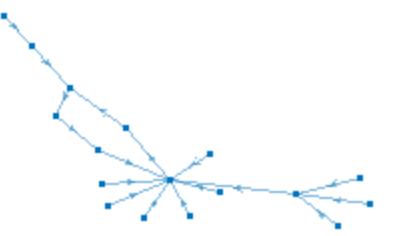
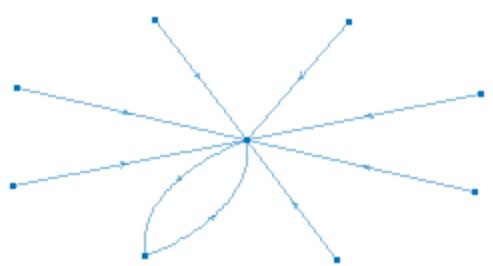
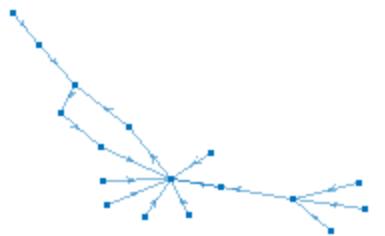
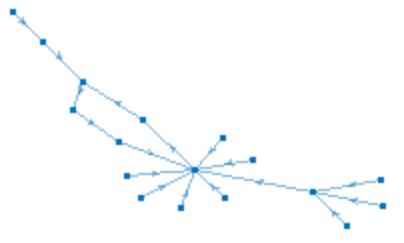


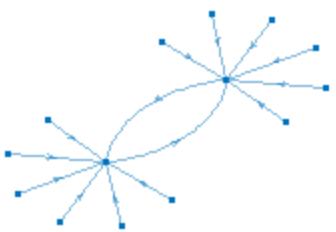
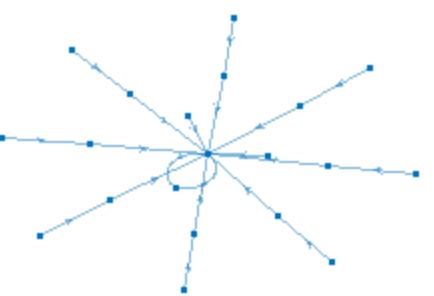
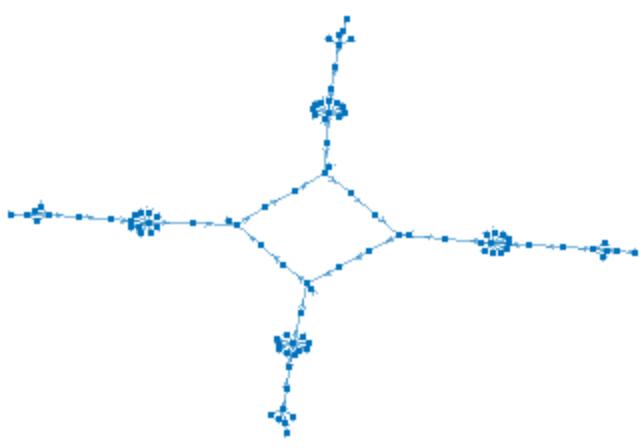
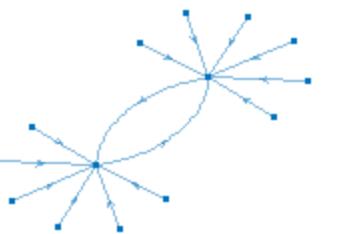
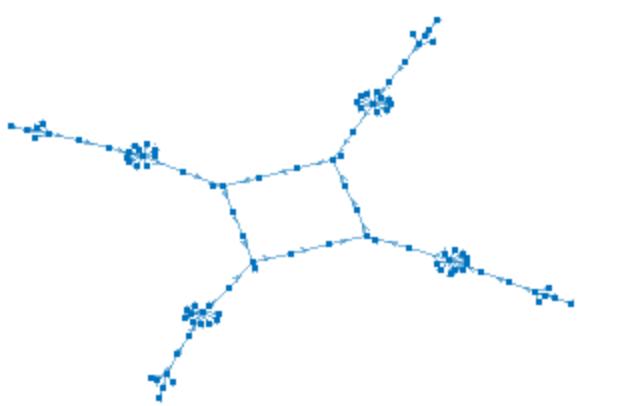


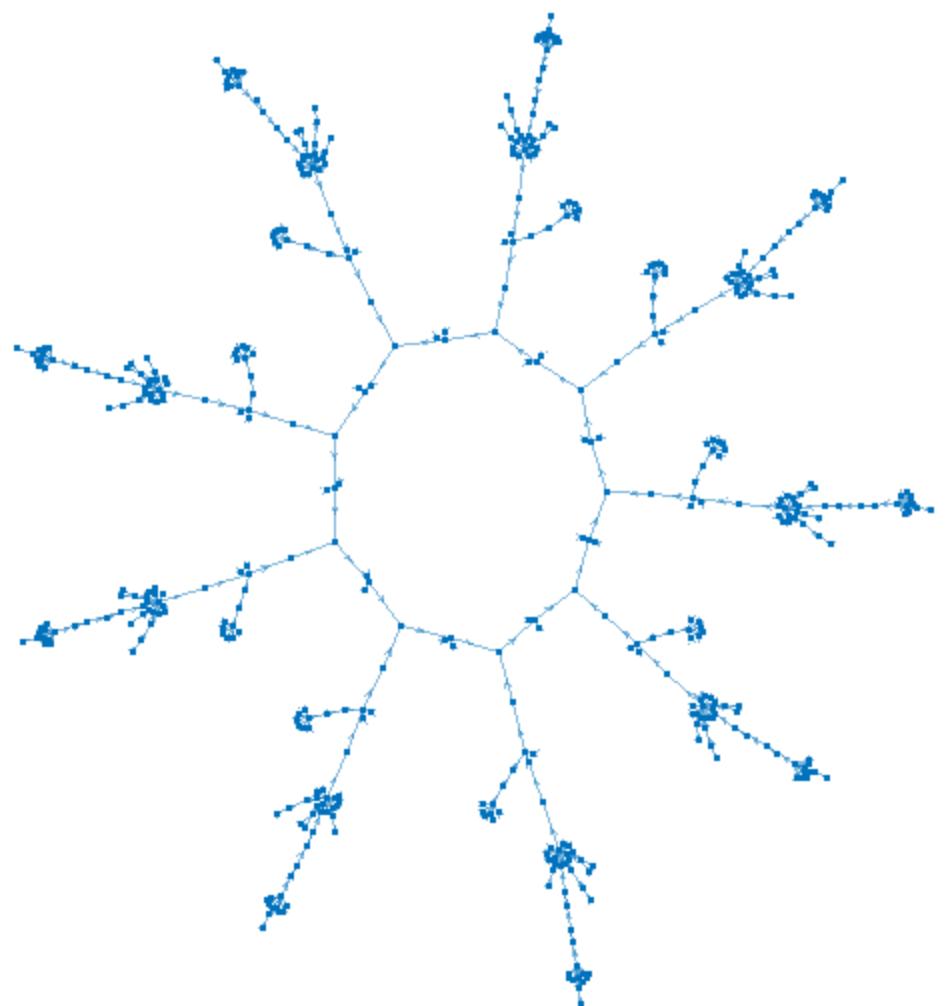


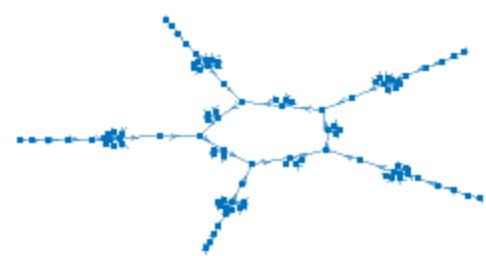
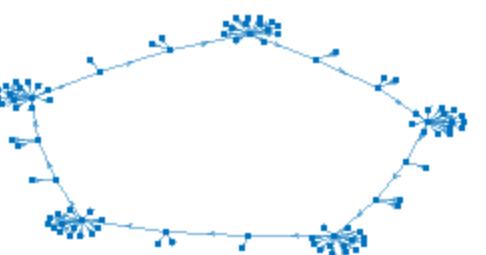
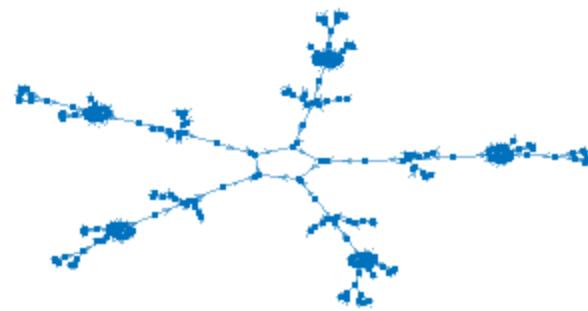
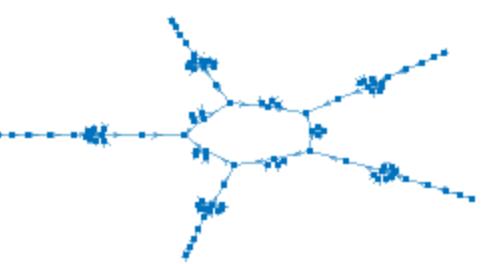
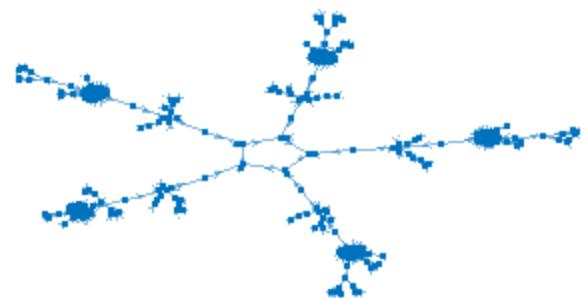
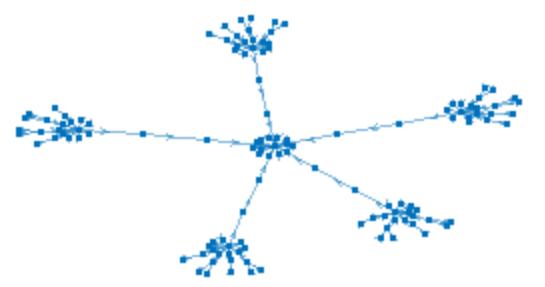


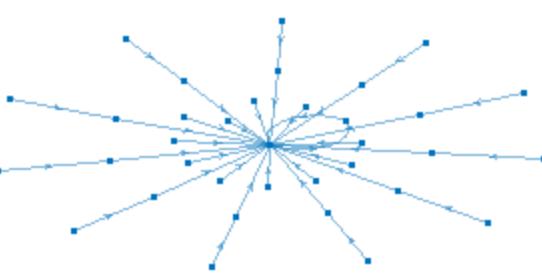
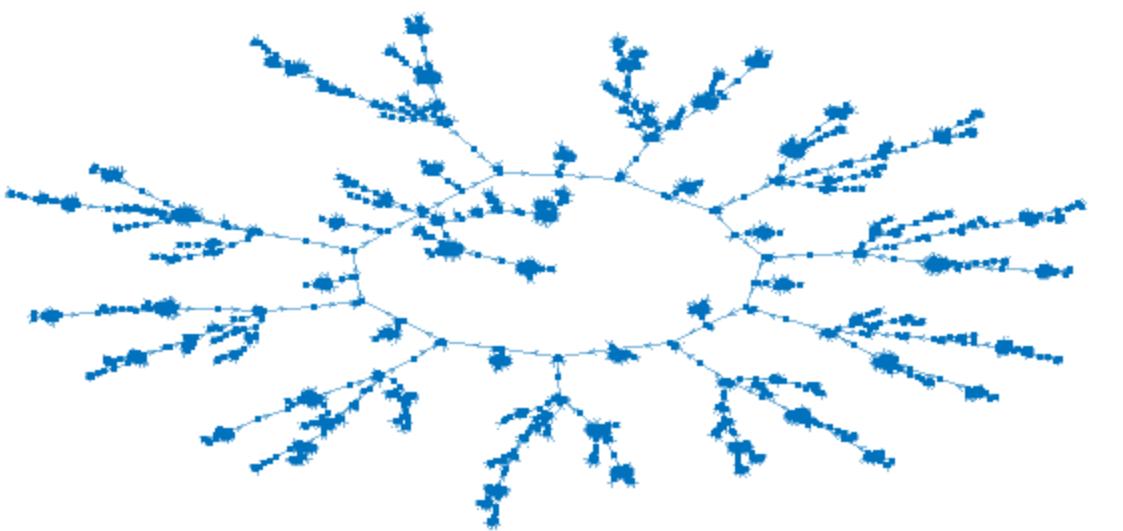
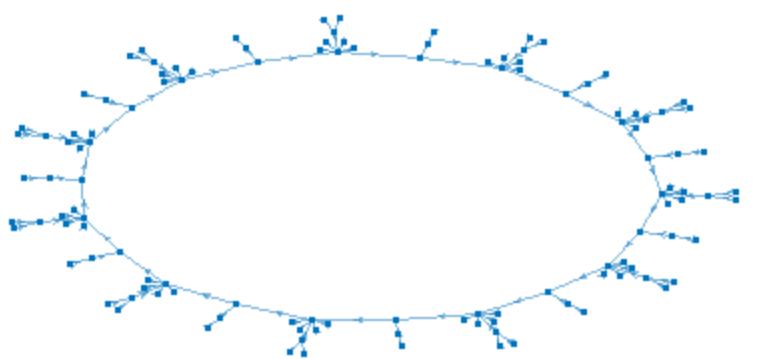


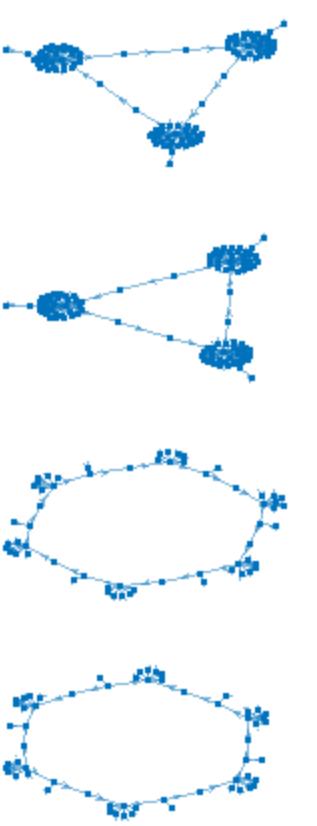
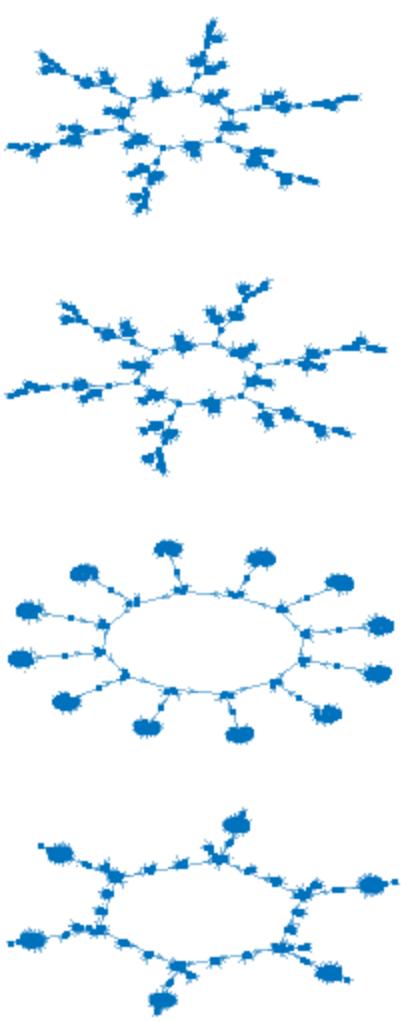
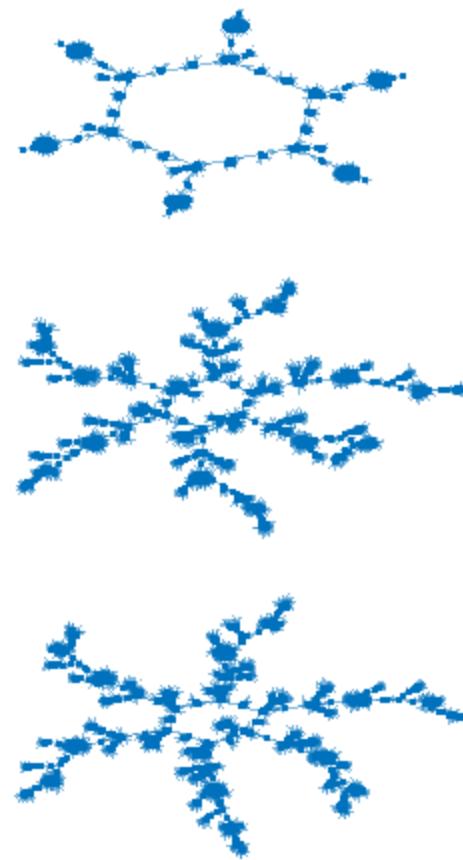


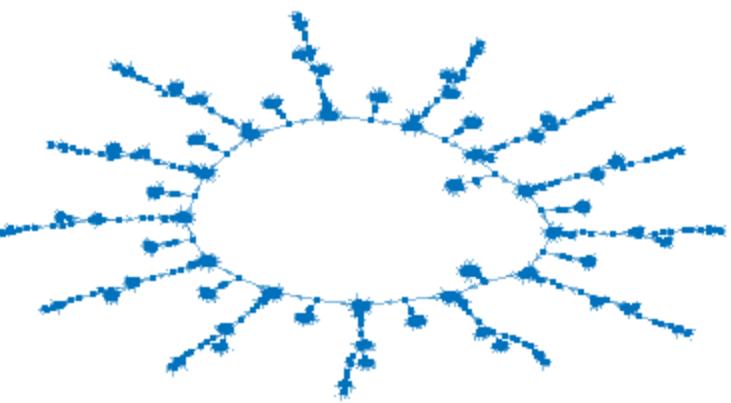
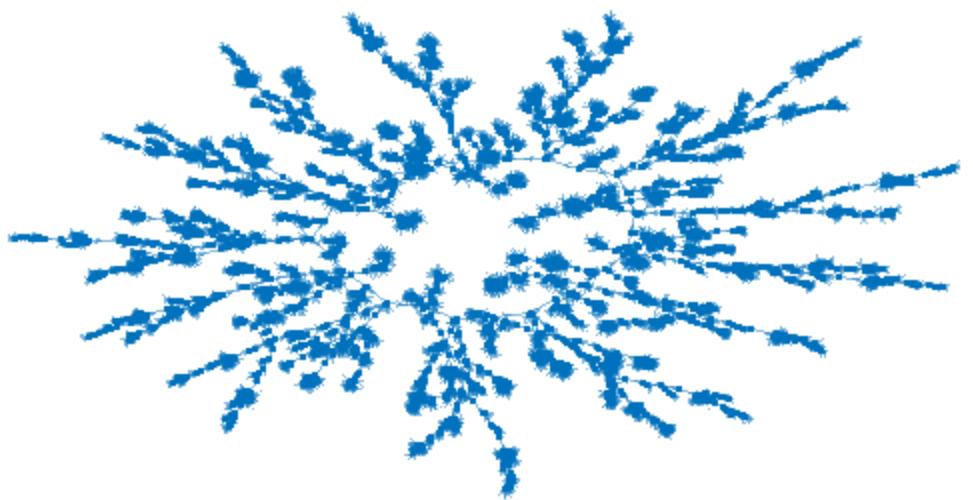
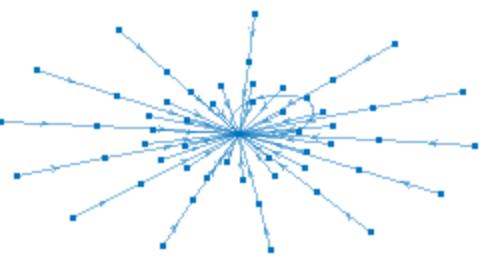
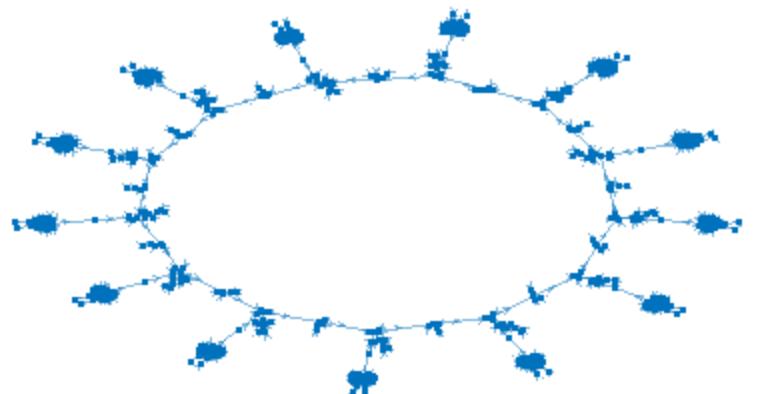




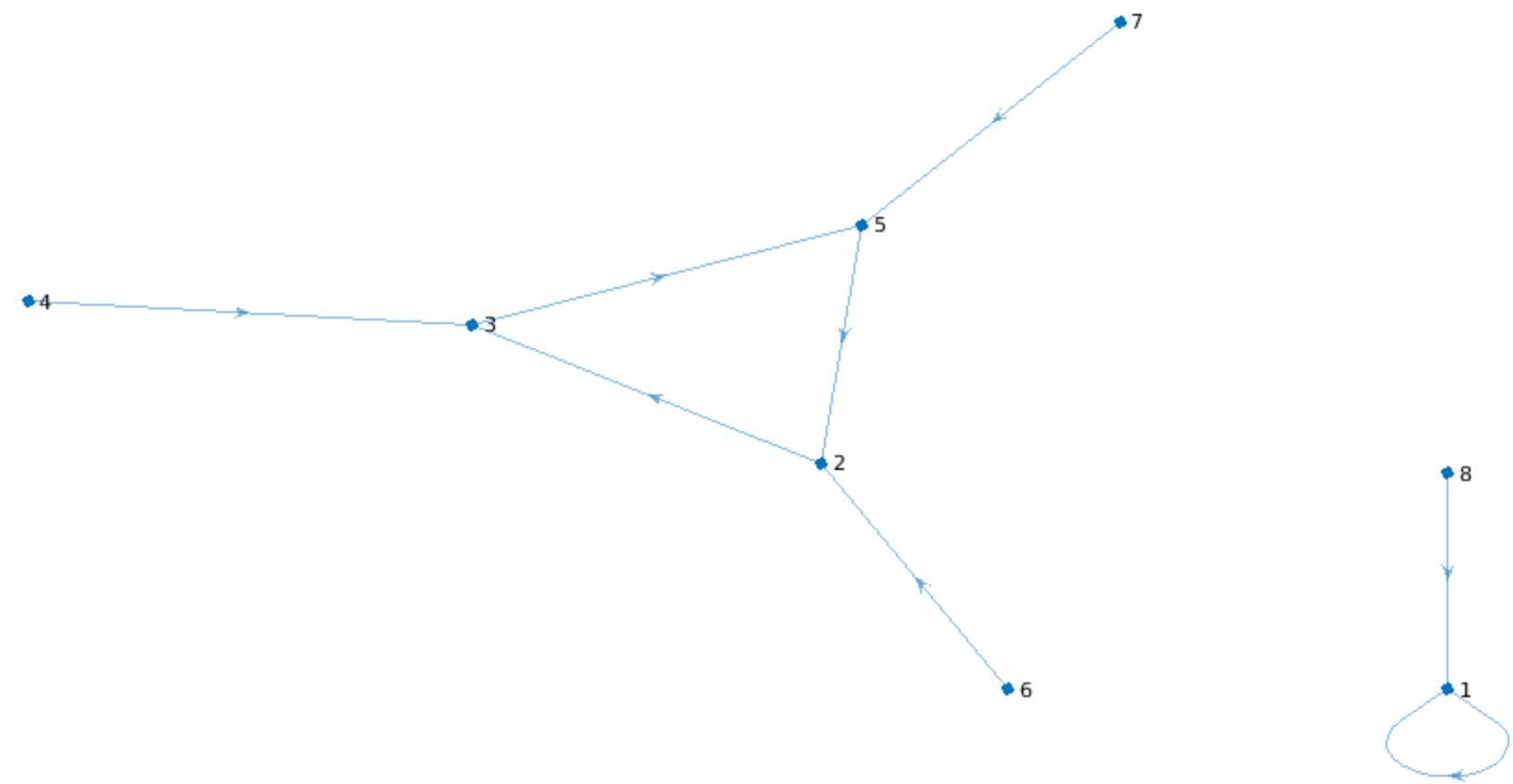
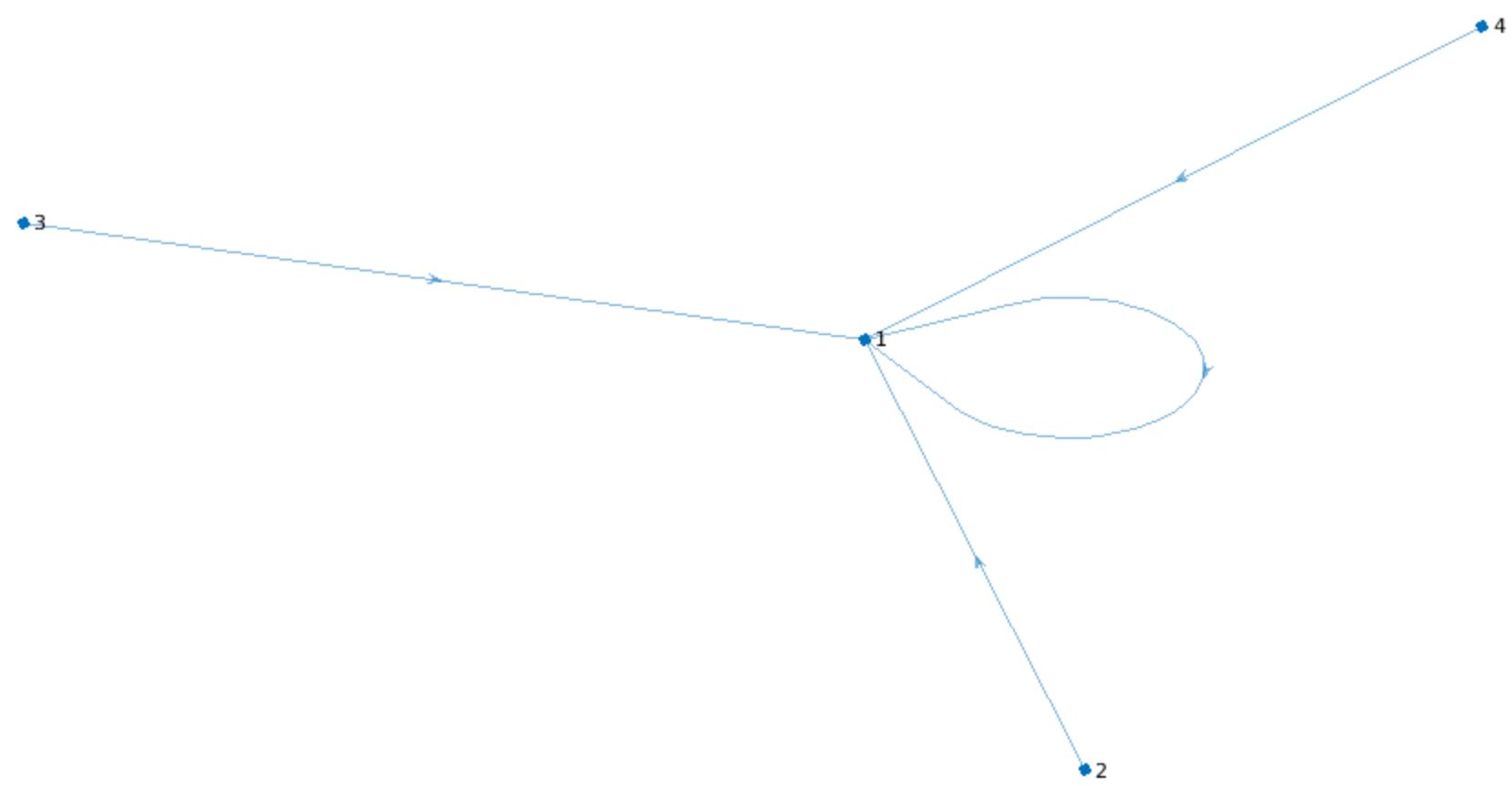


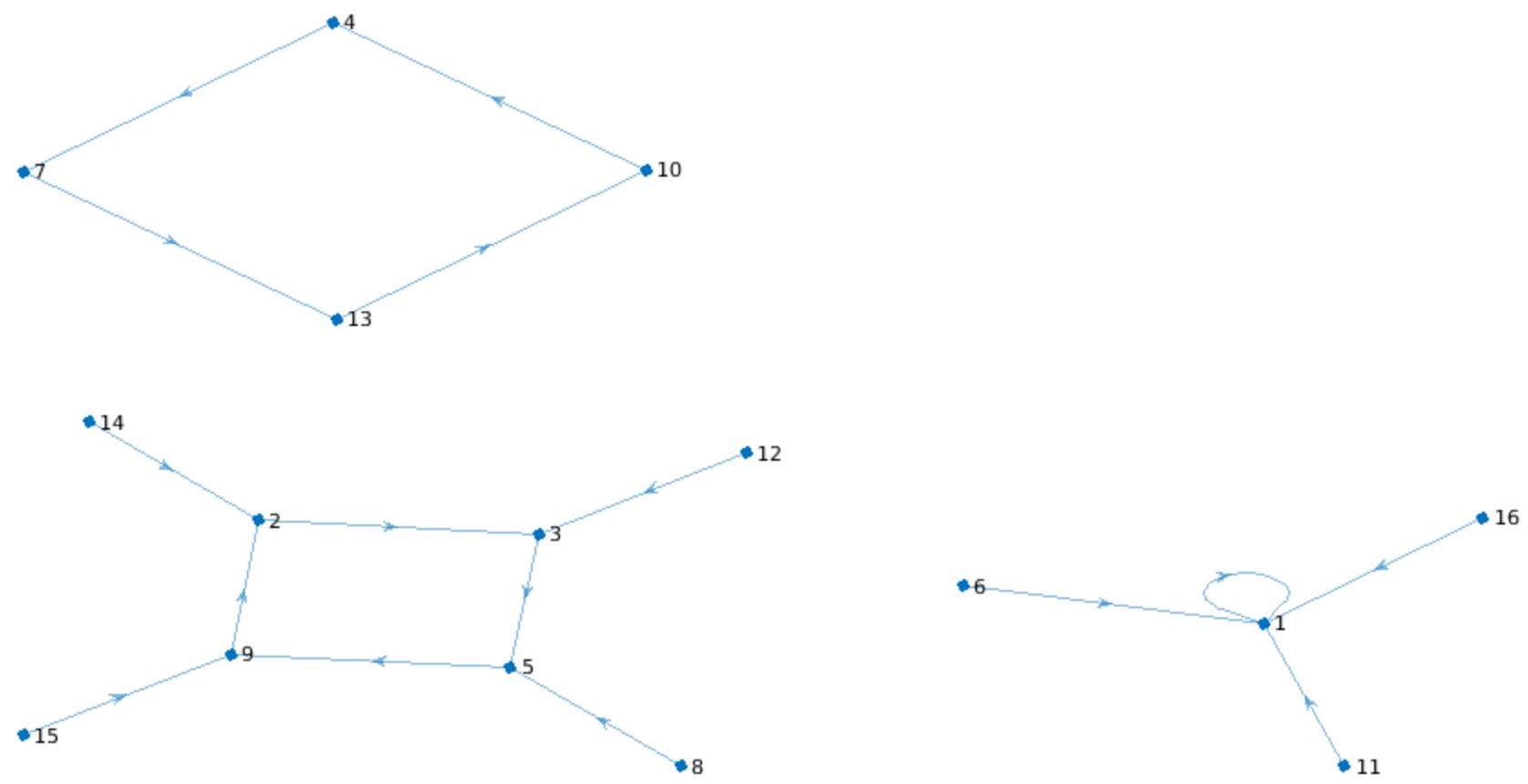


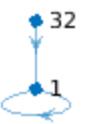
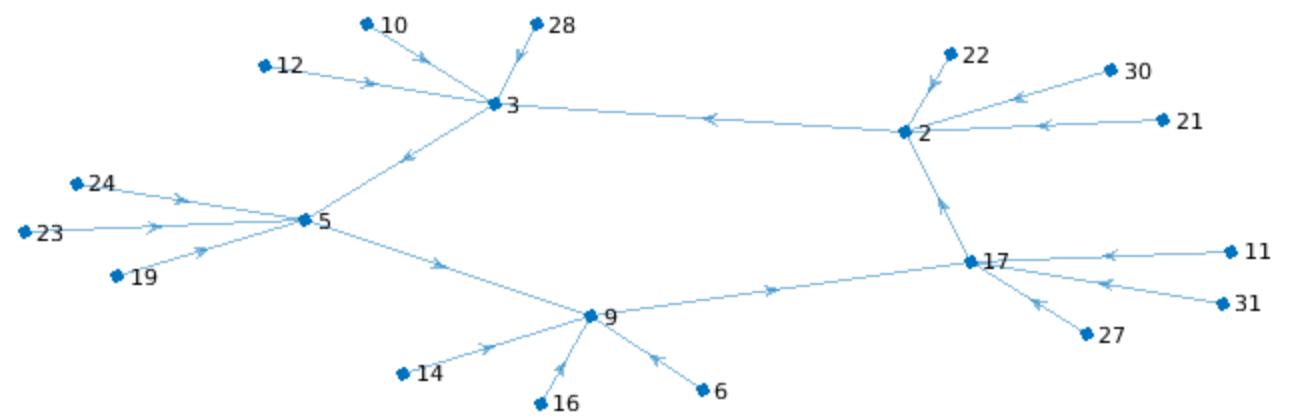
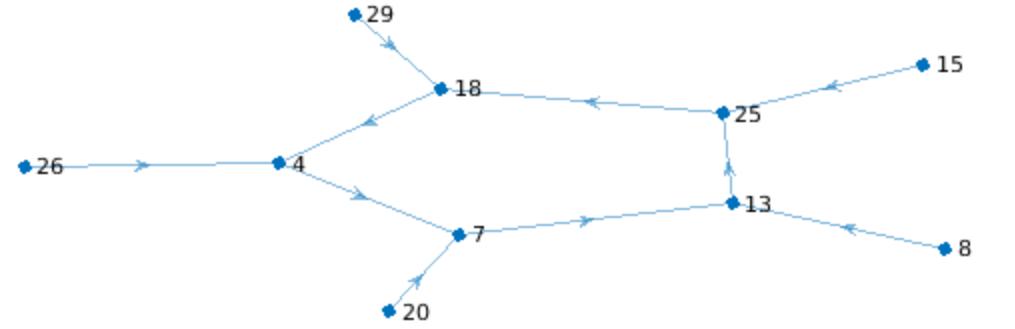


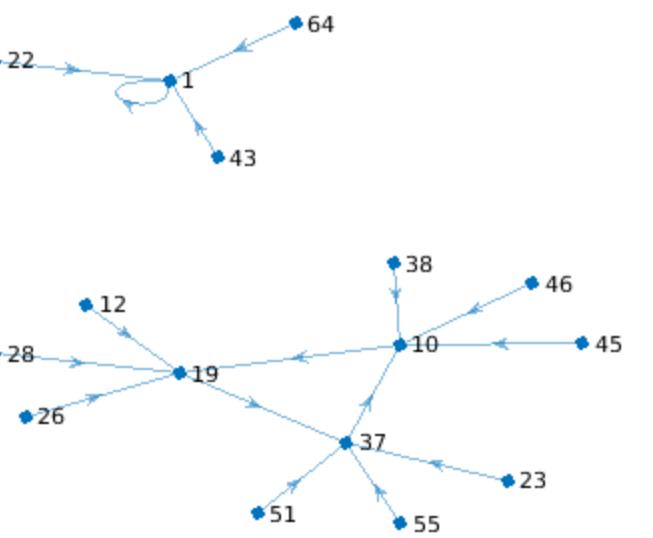
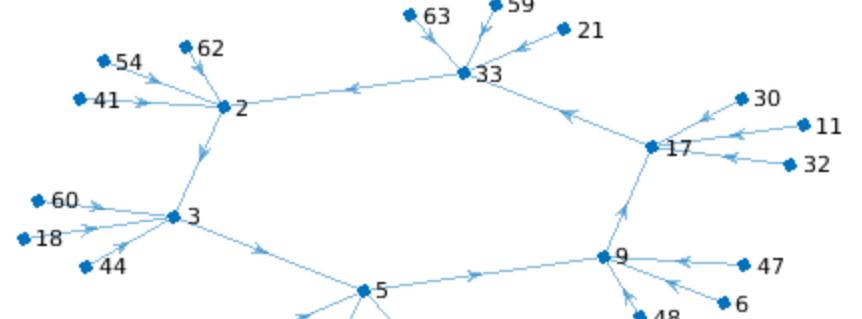


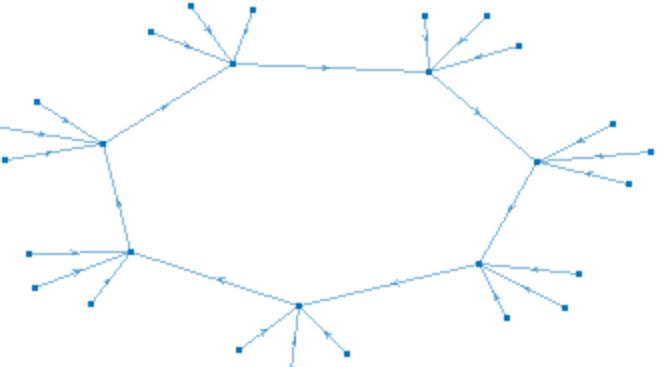
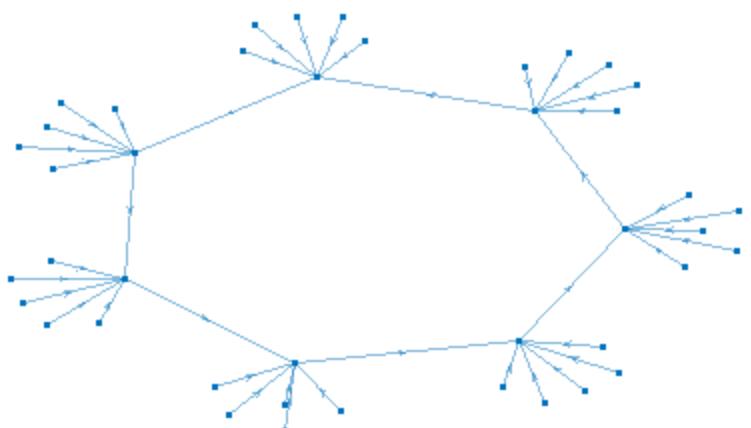
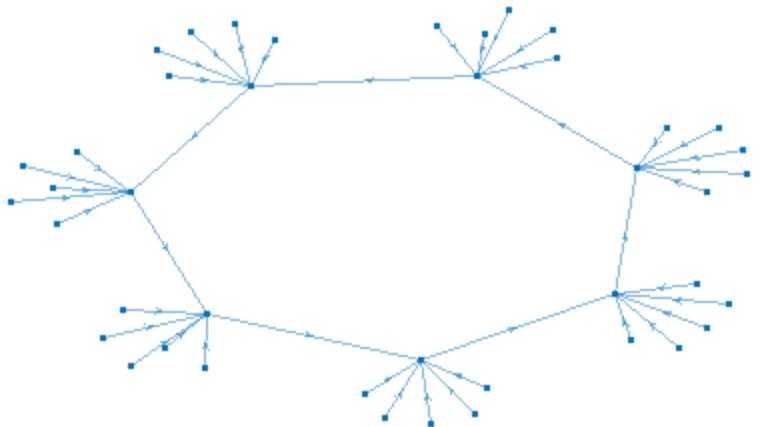
10

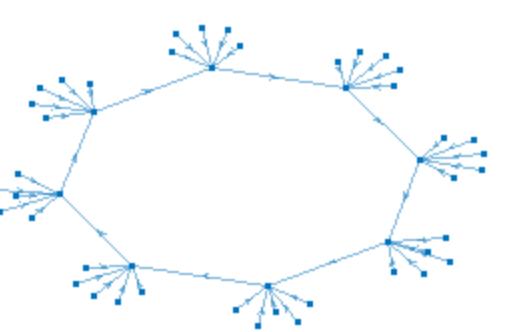
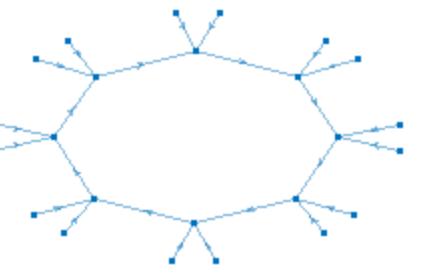
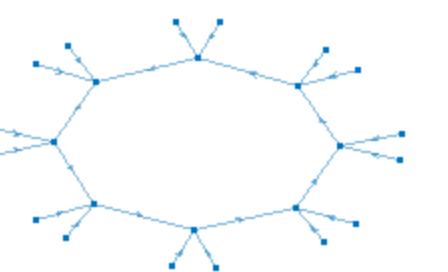
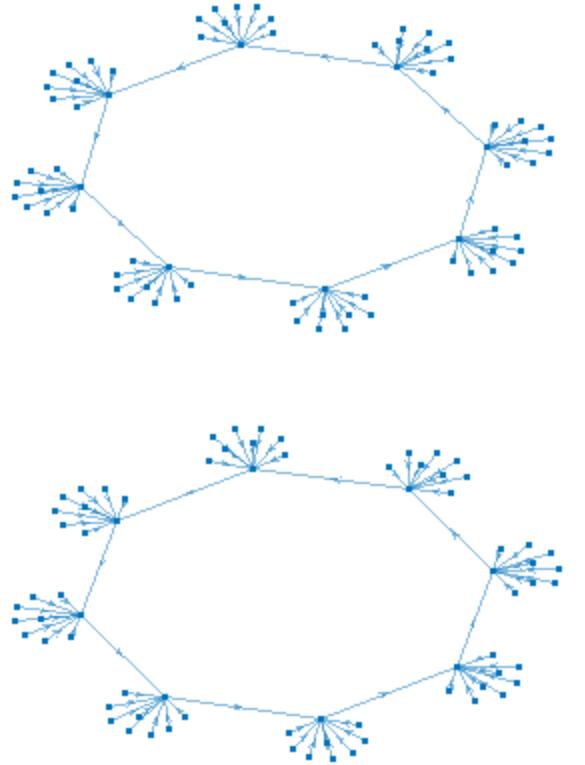


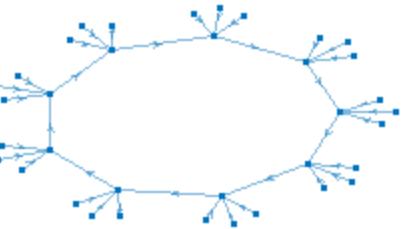
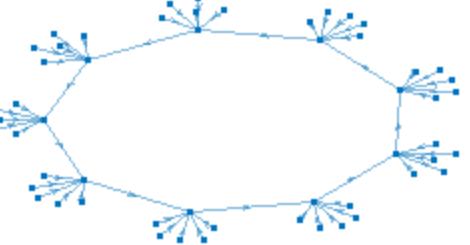
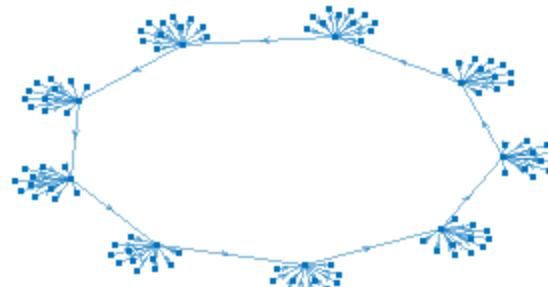
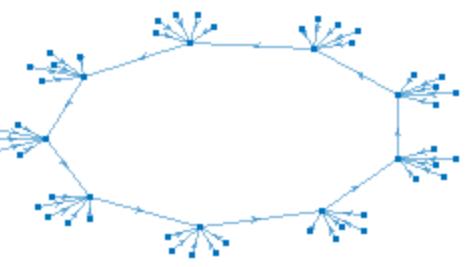
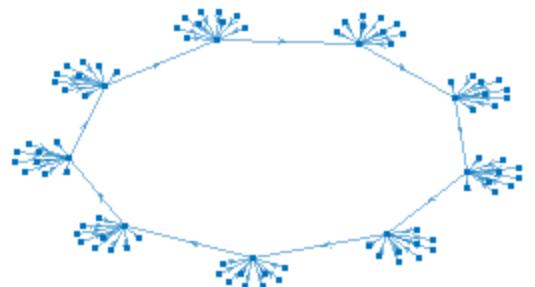
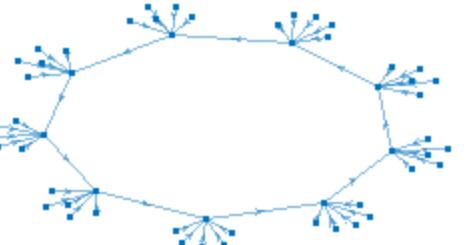
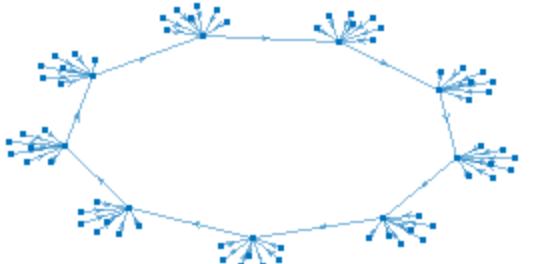


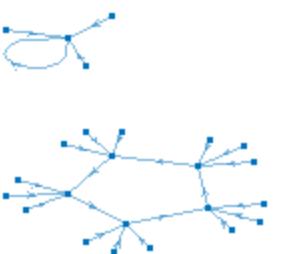
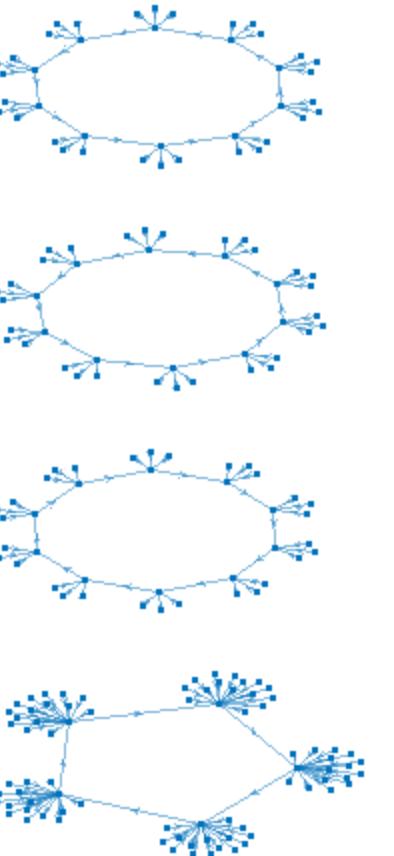
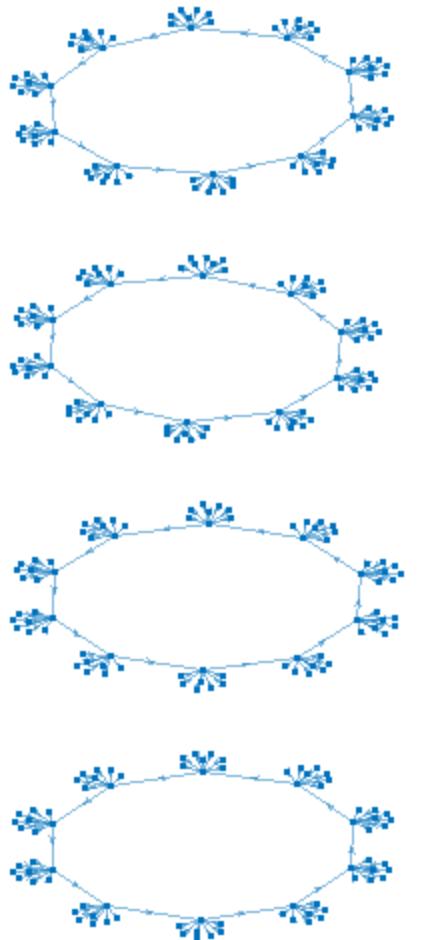
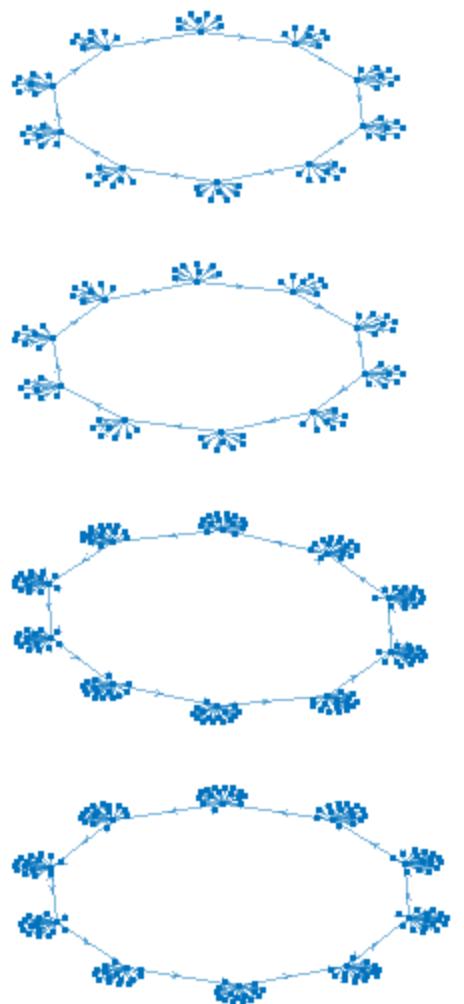


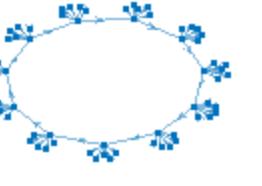
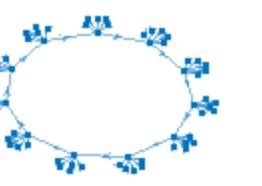
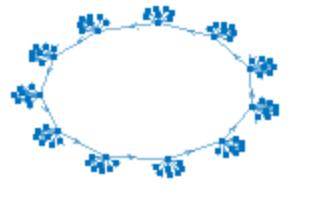
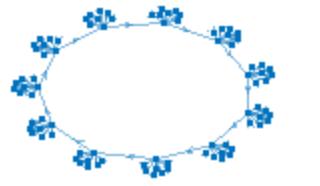
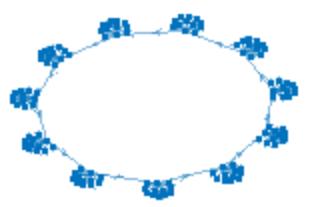
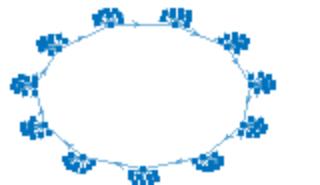
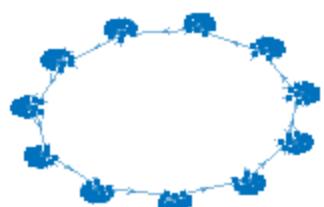
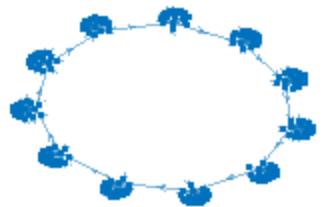
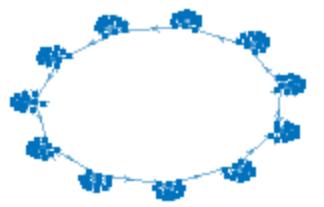


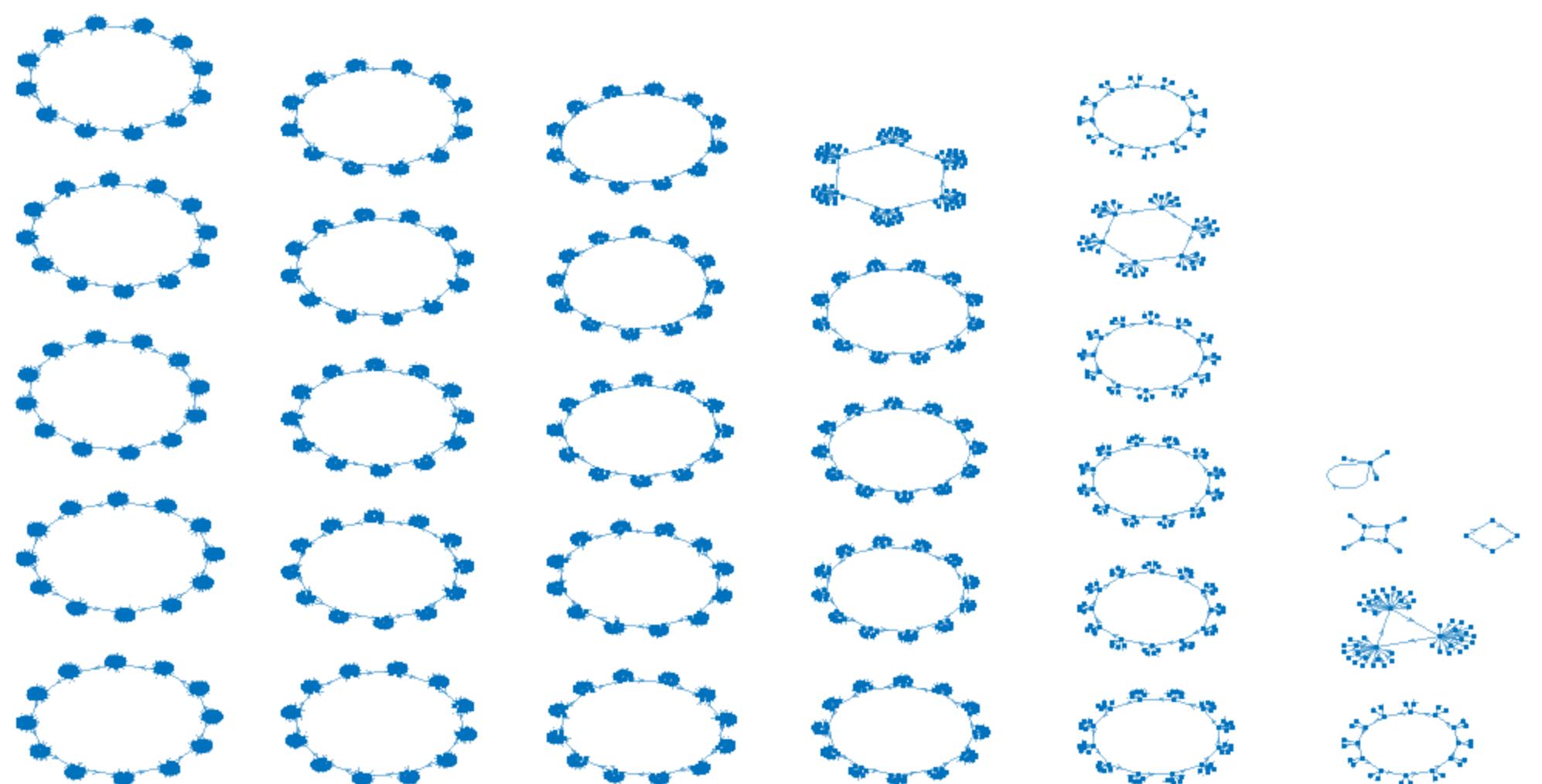


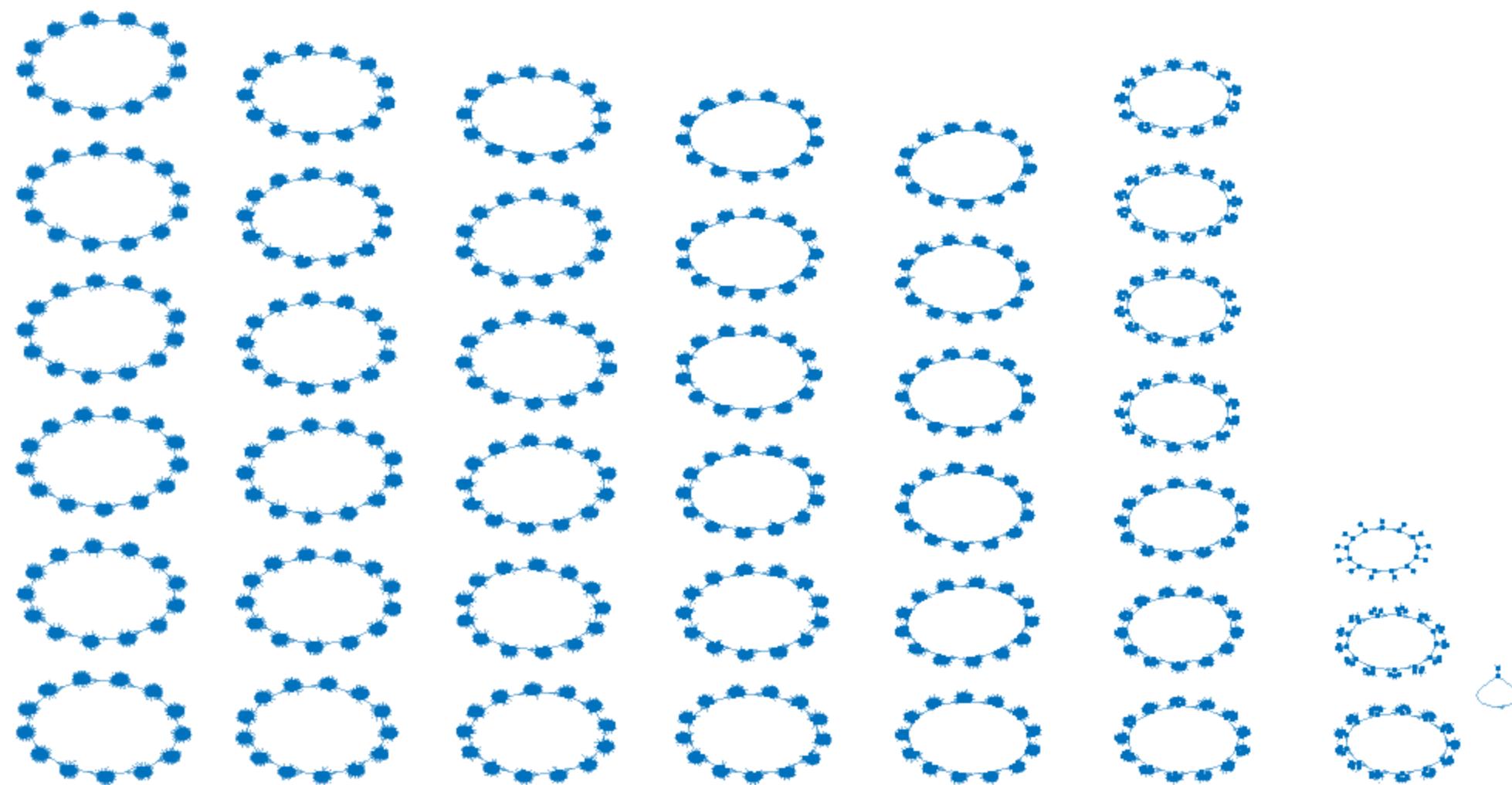




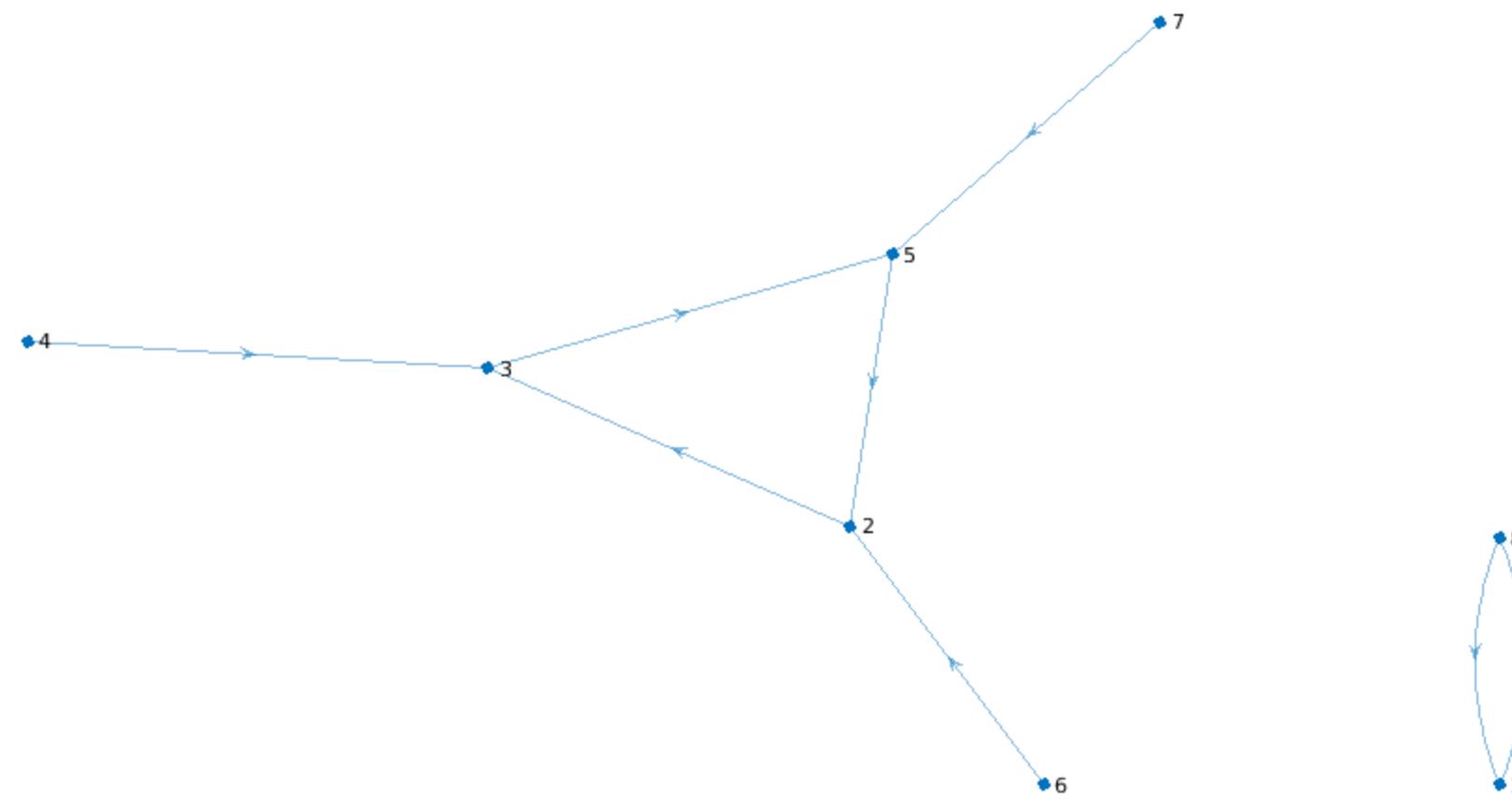
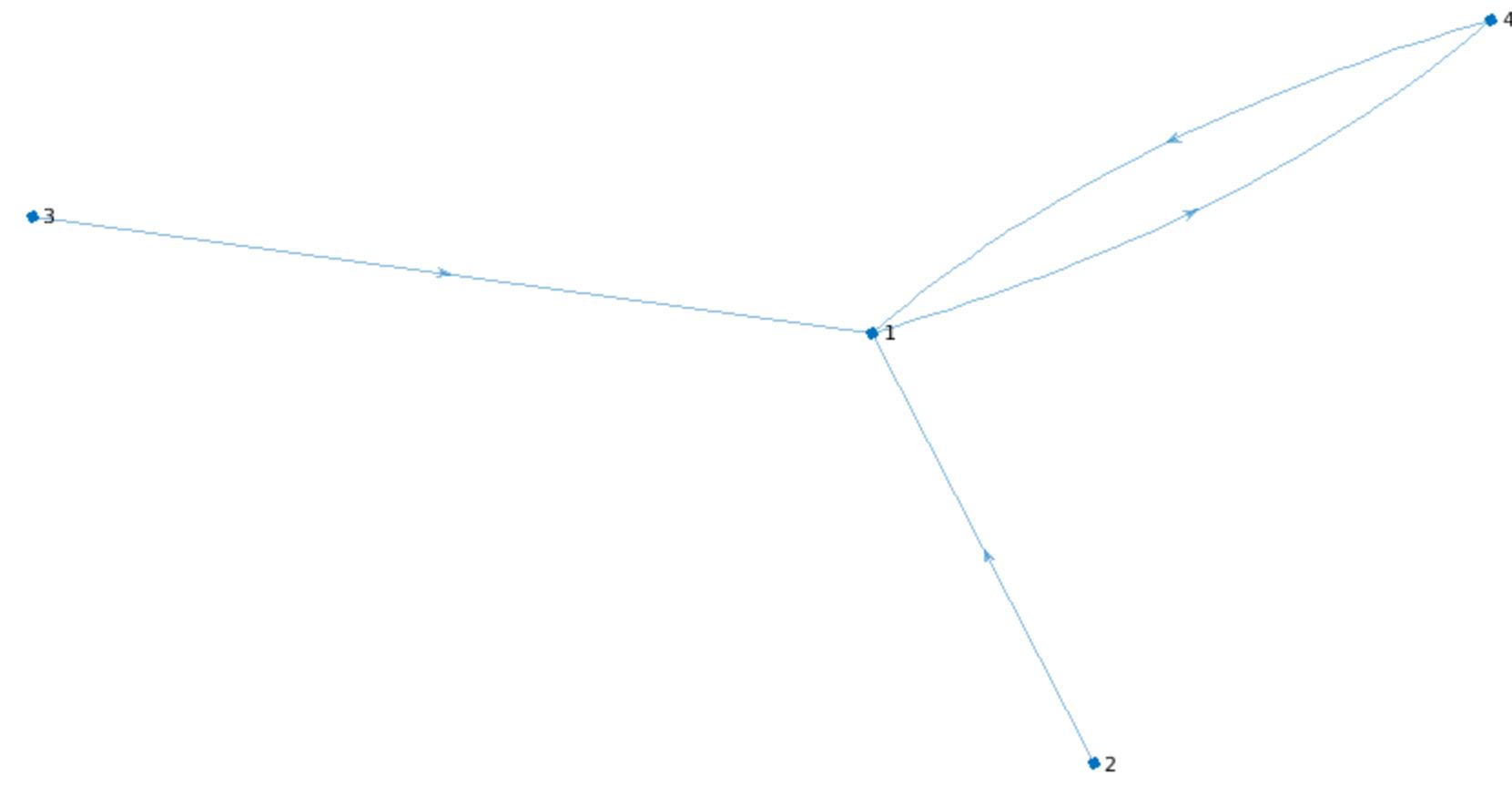


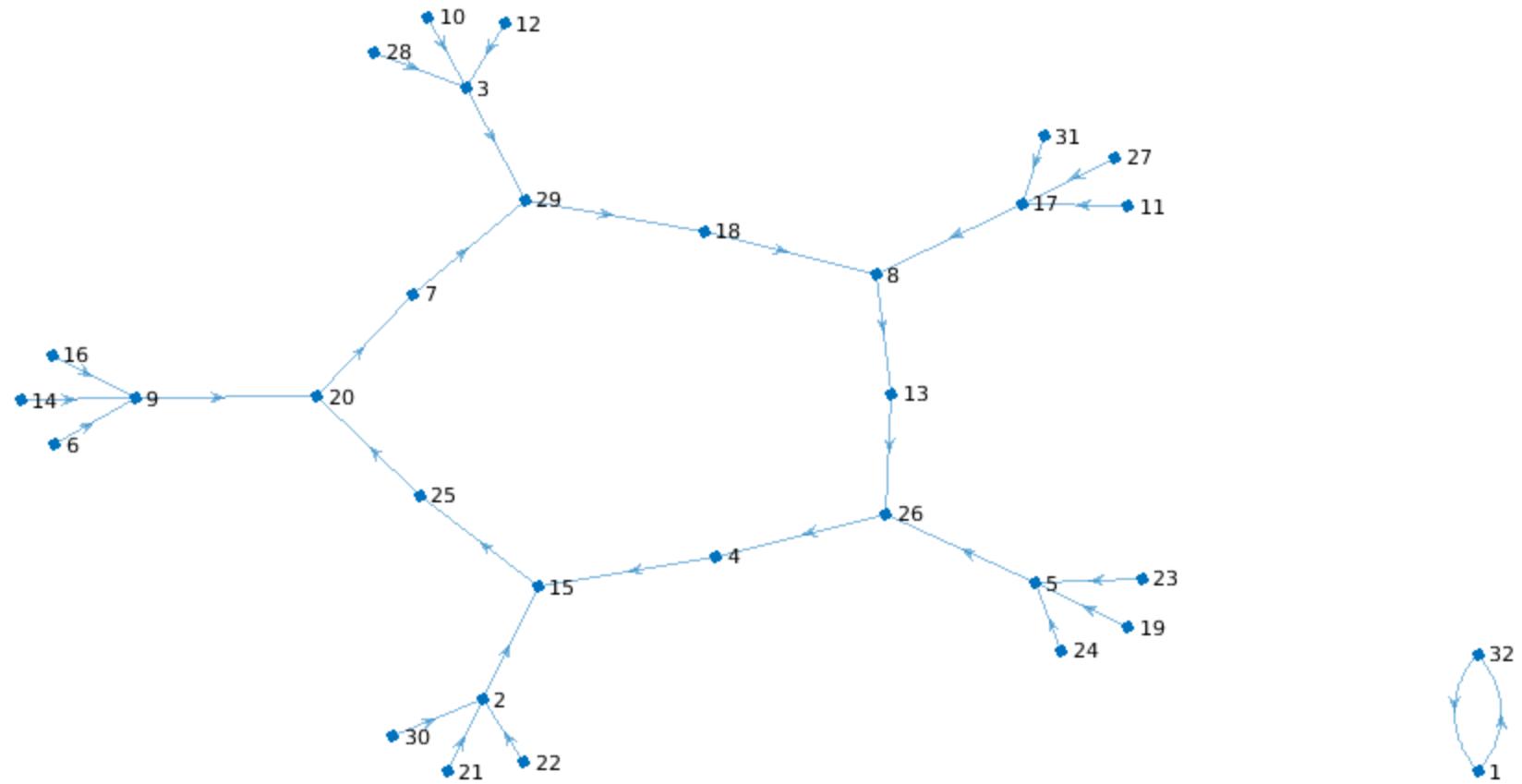
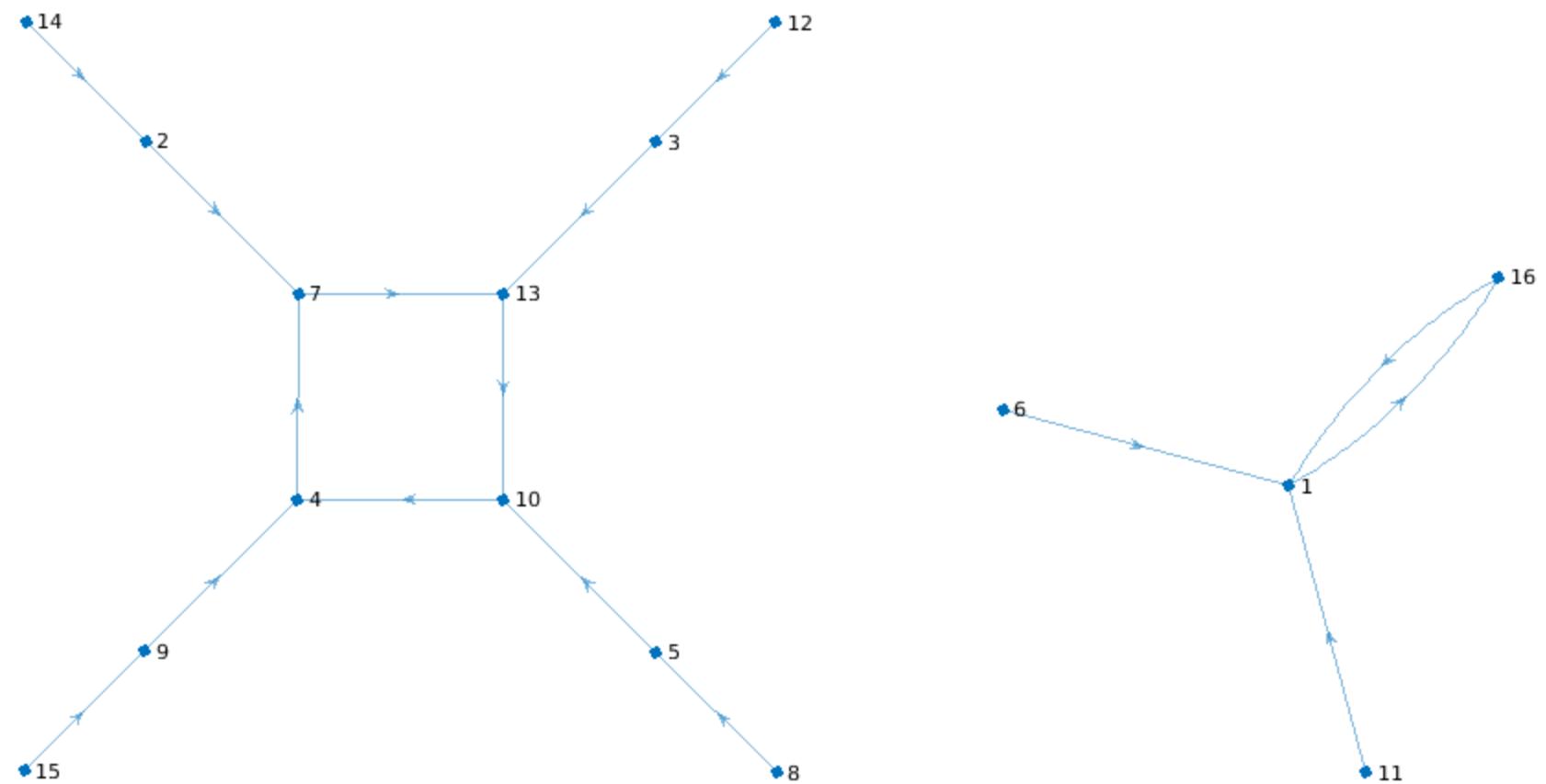


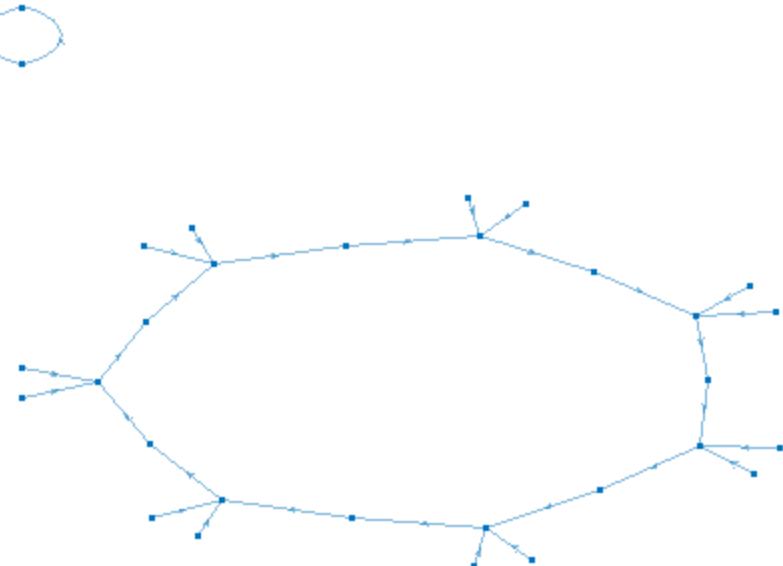
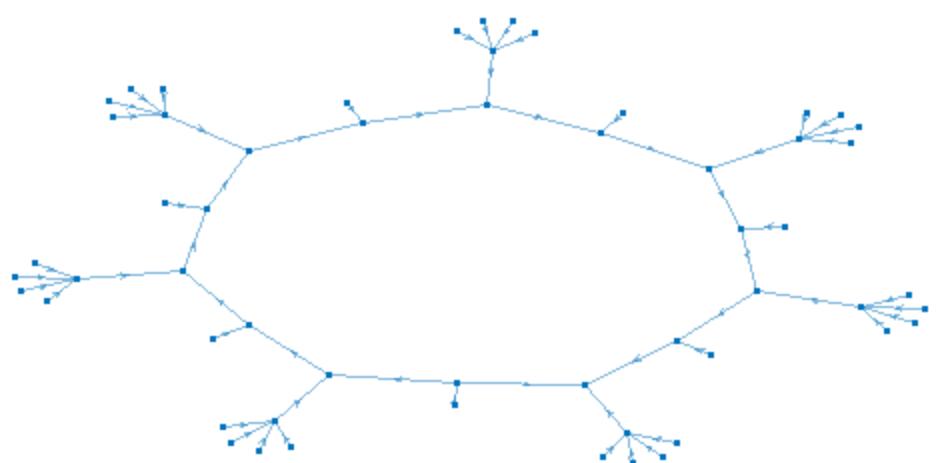
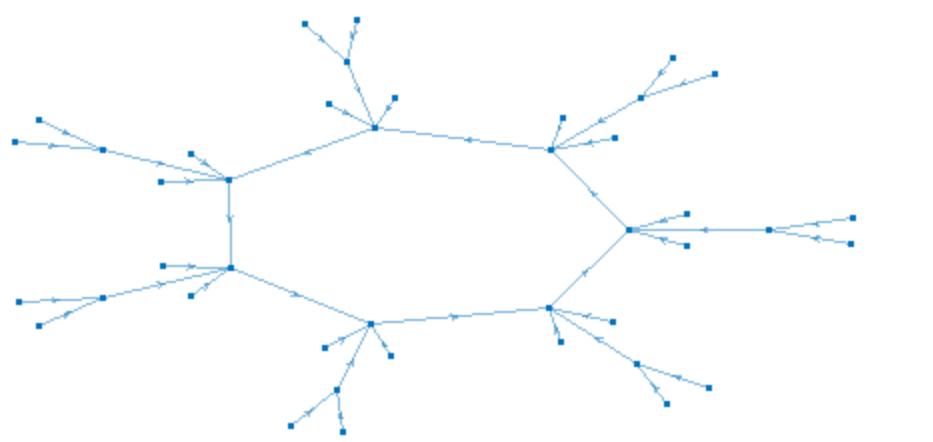


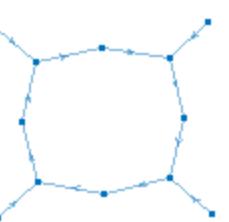
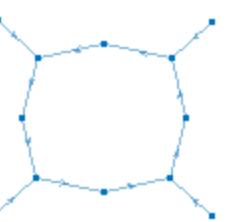
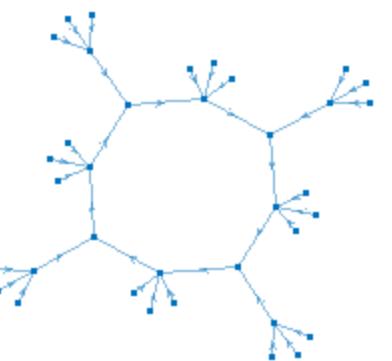
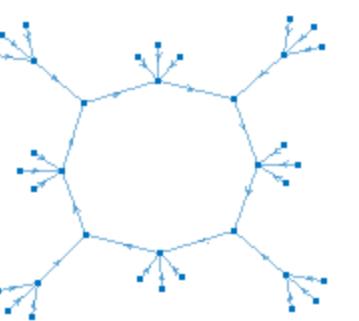
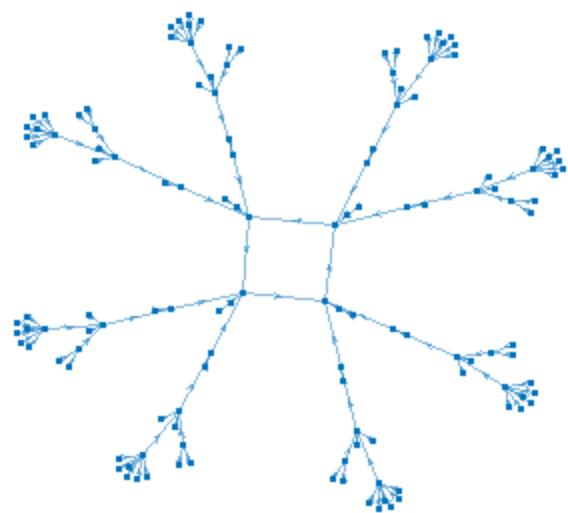
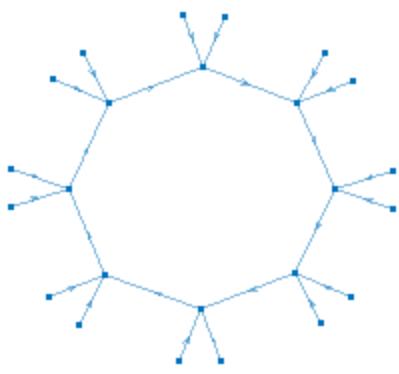


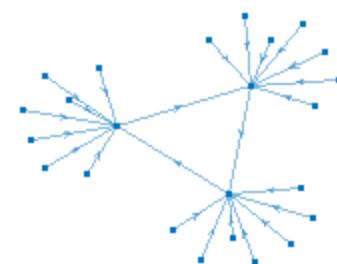
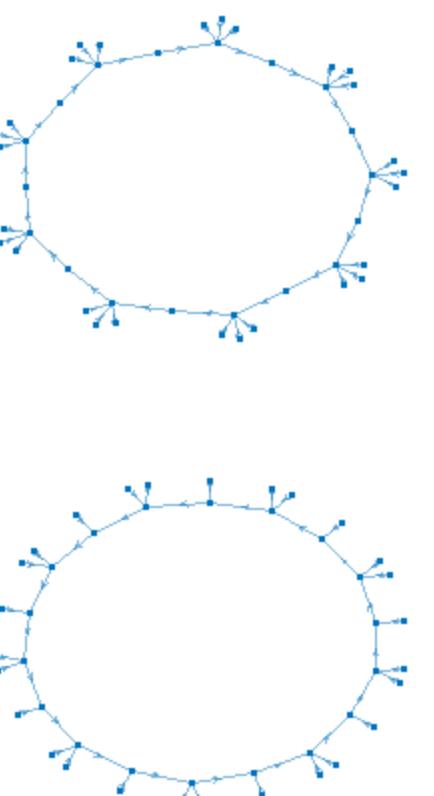
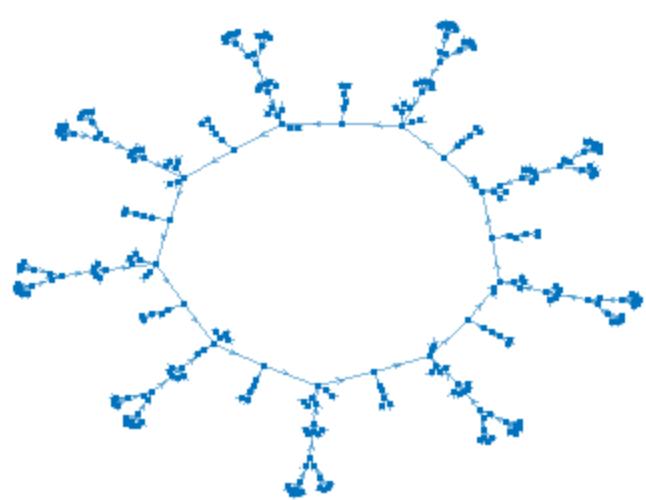
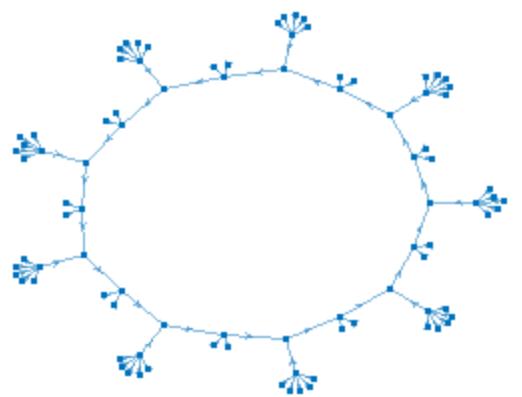
11

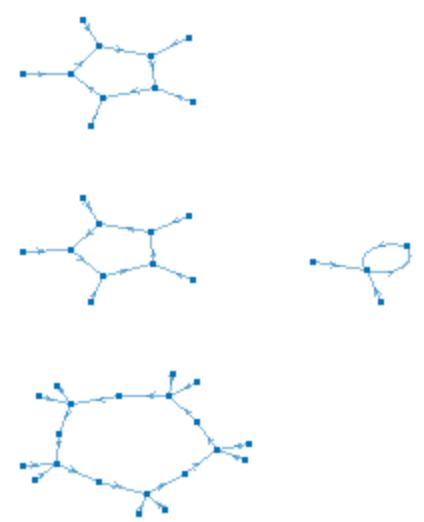
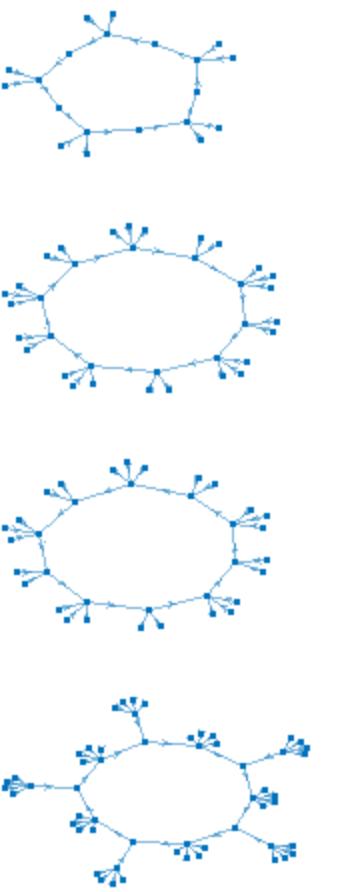
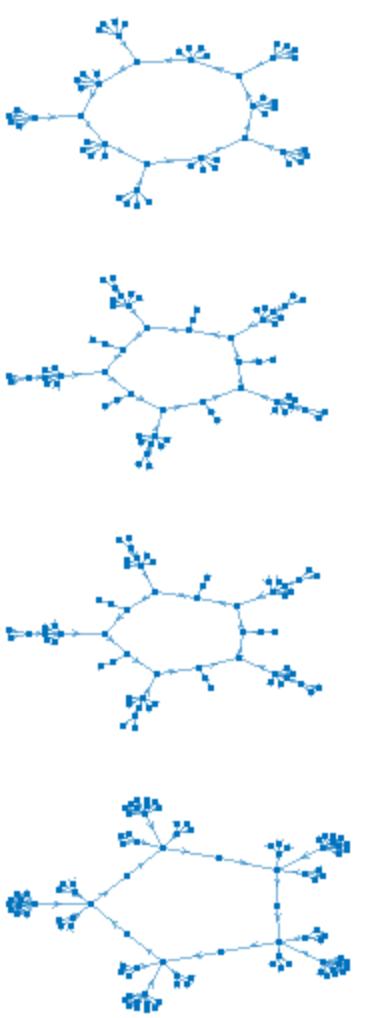
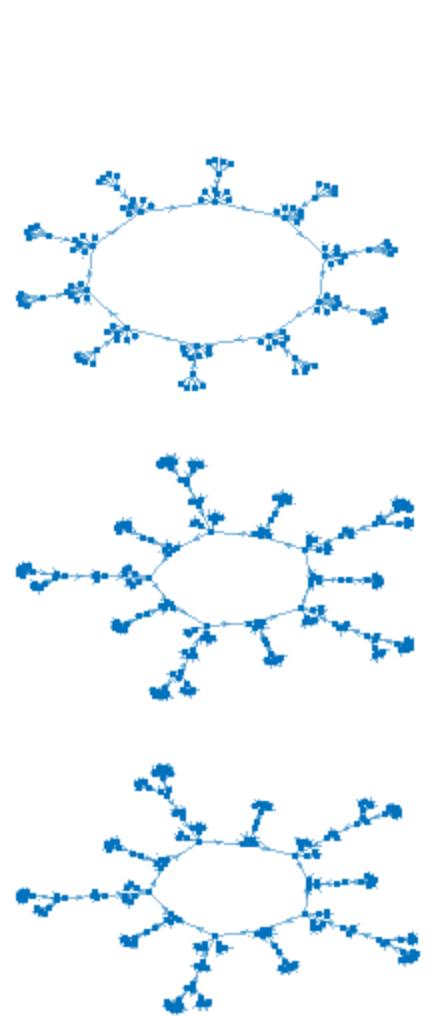


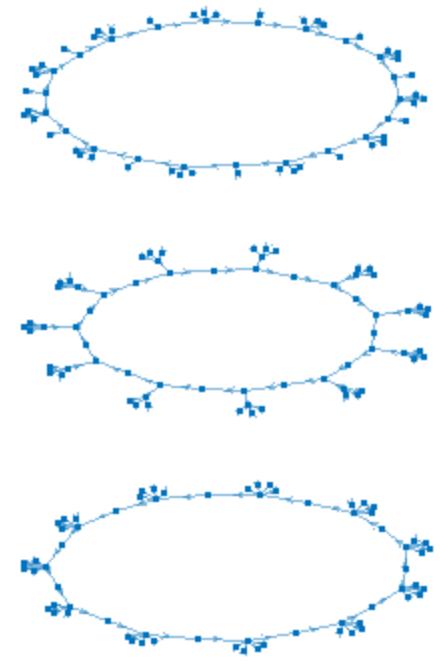
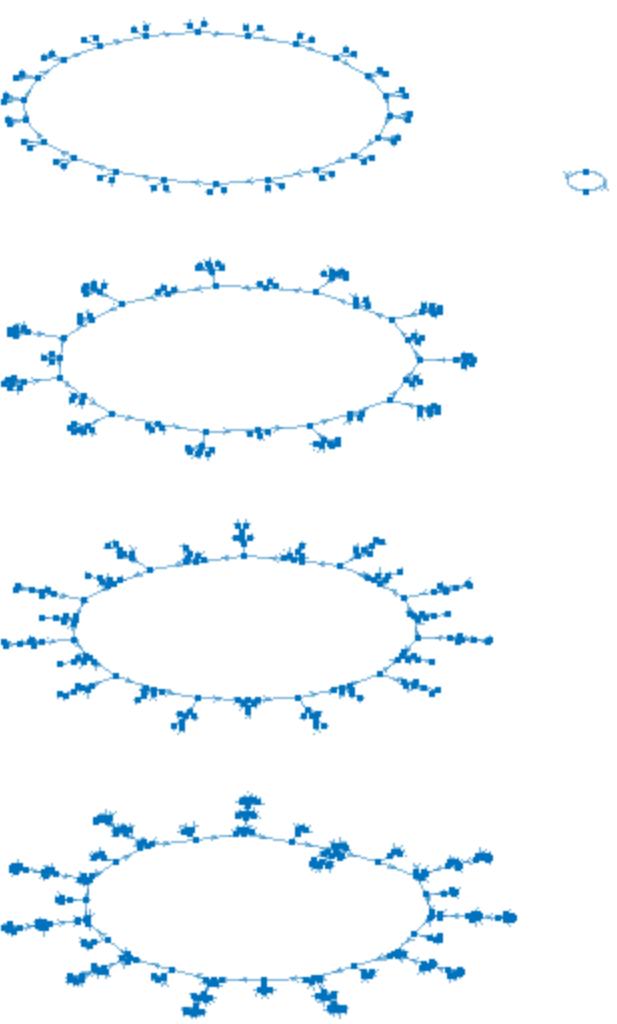
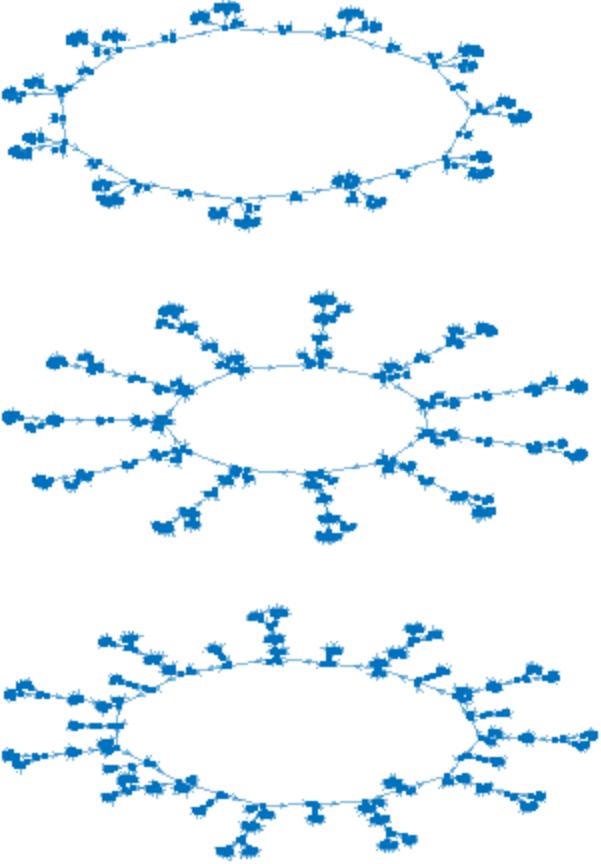


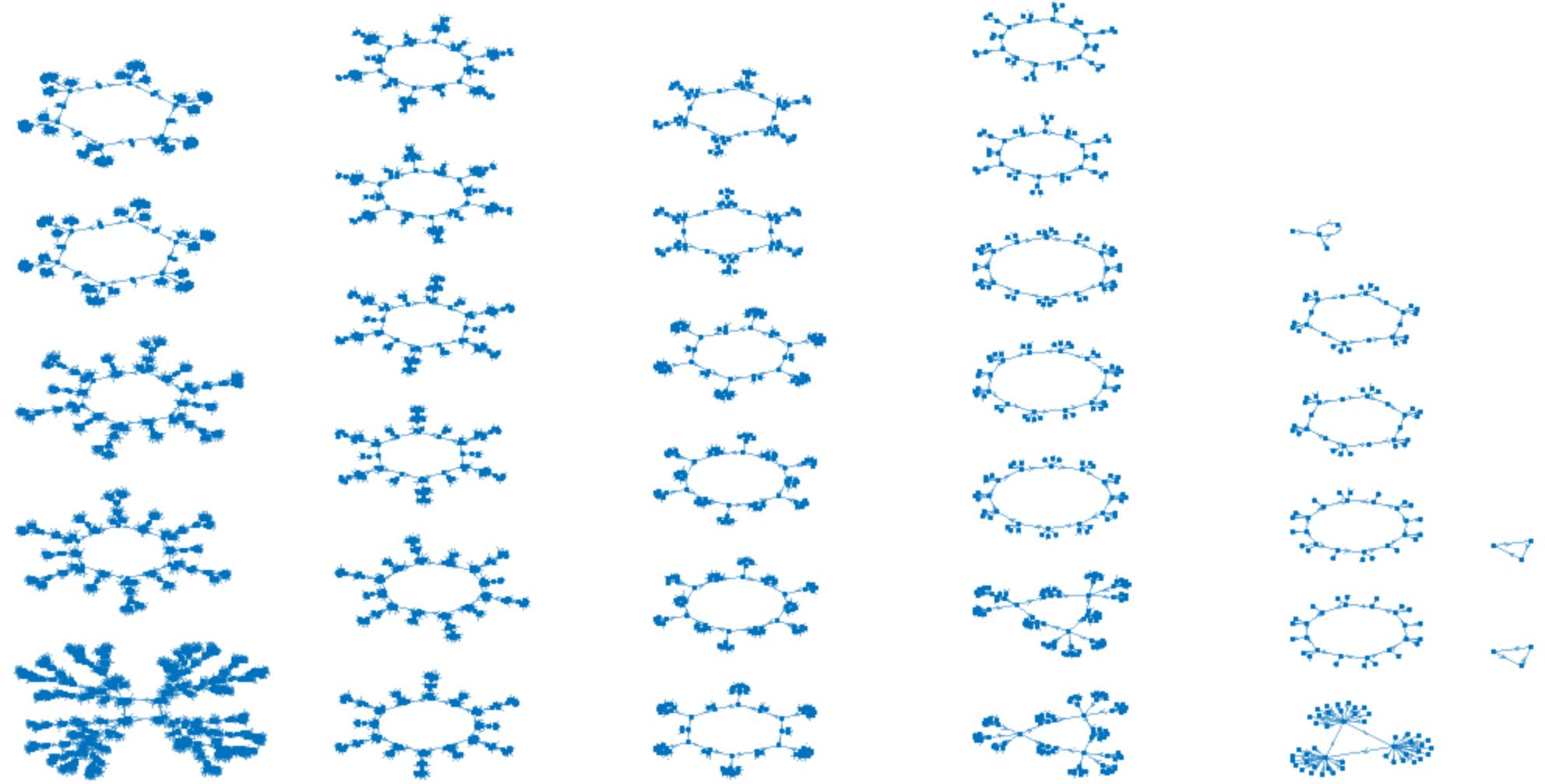


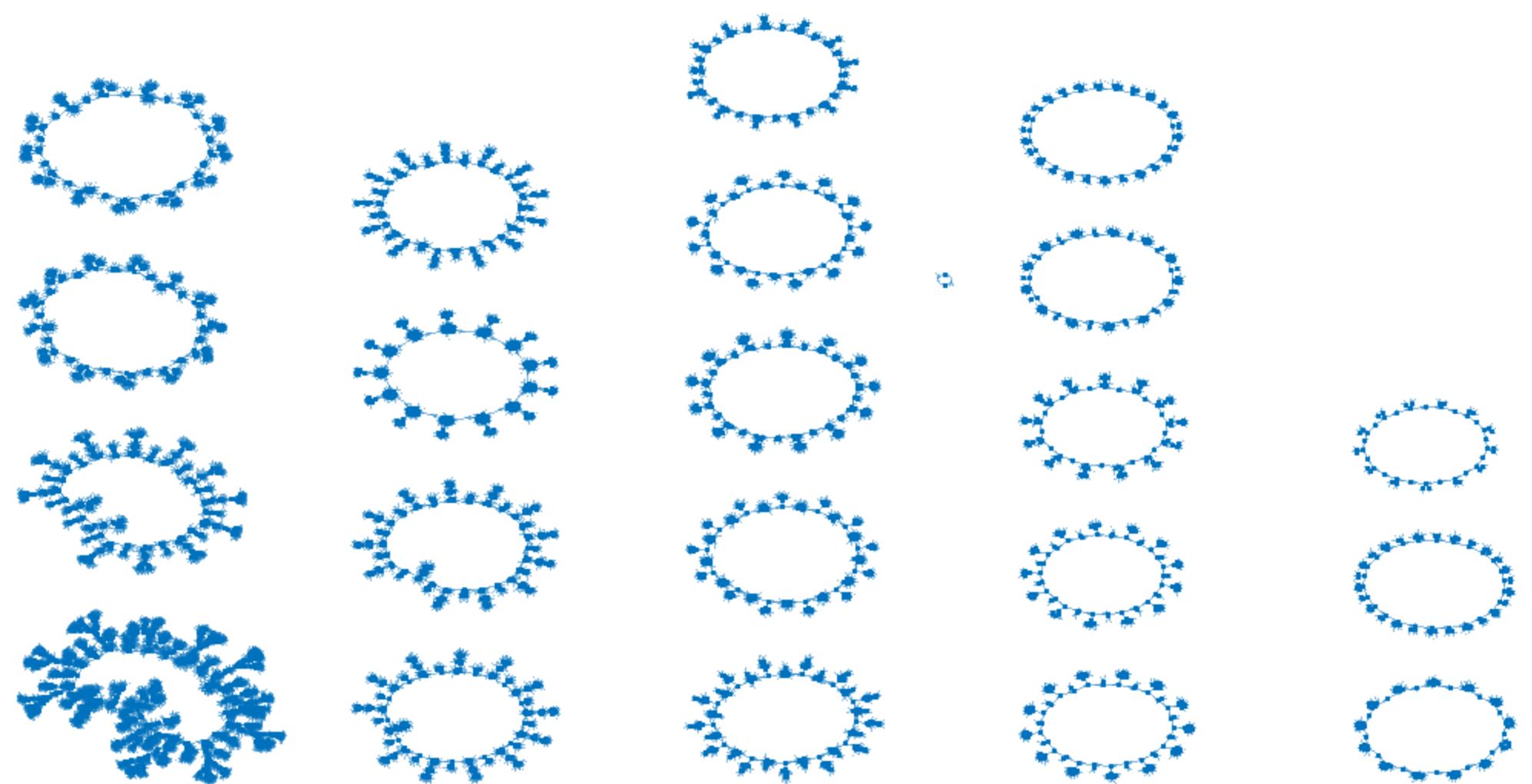




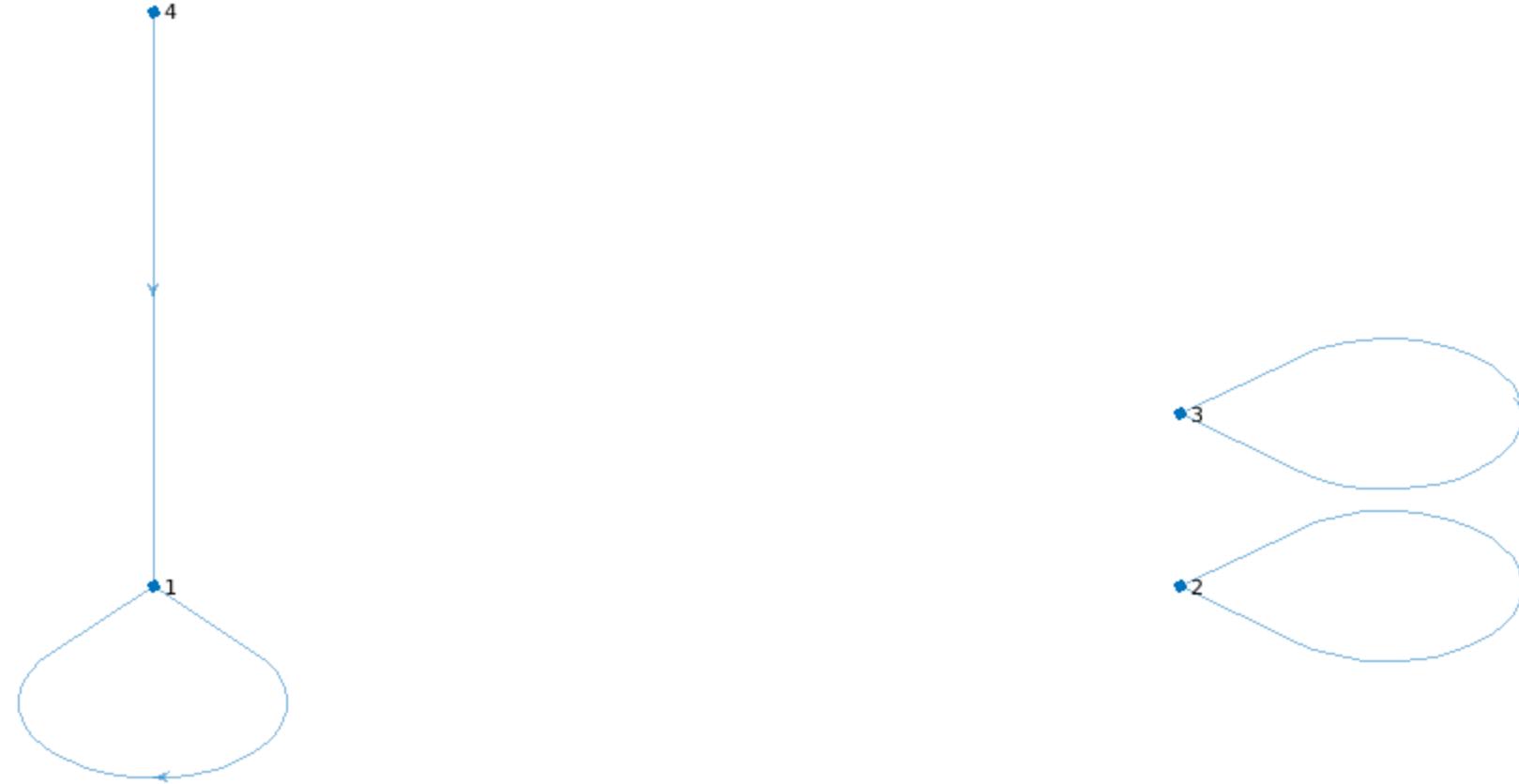


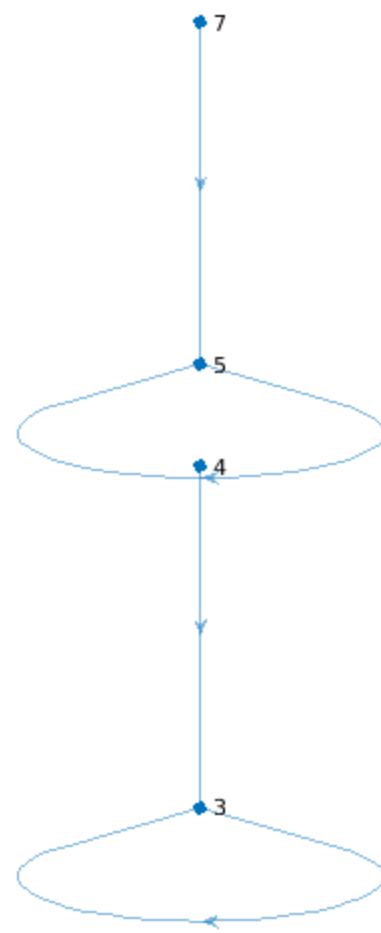
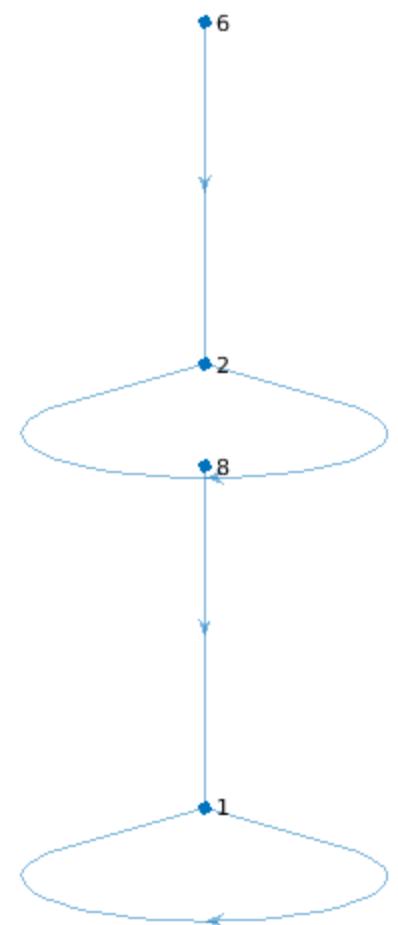


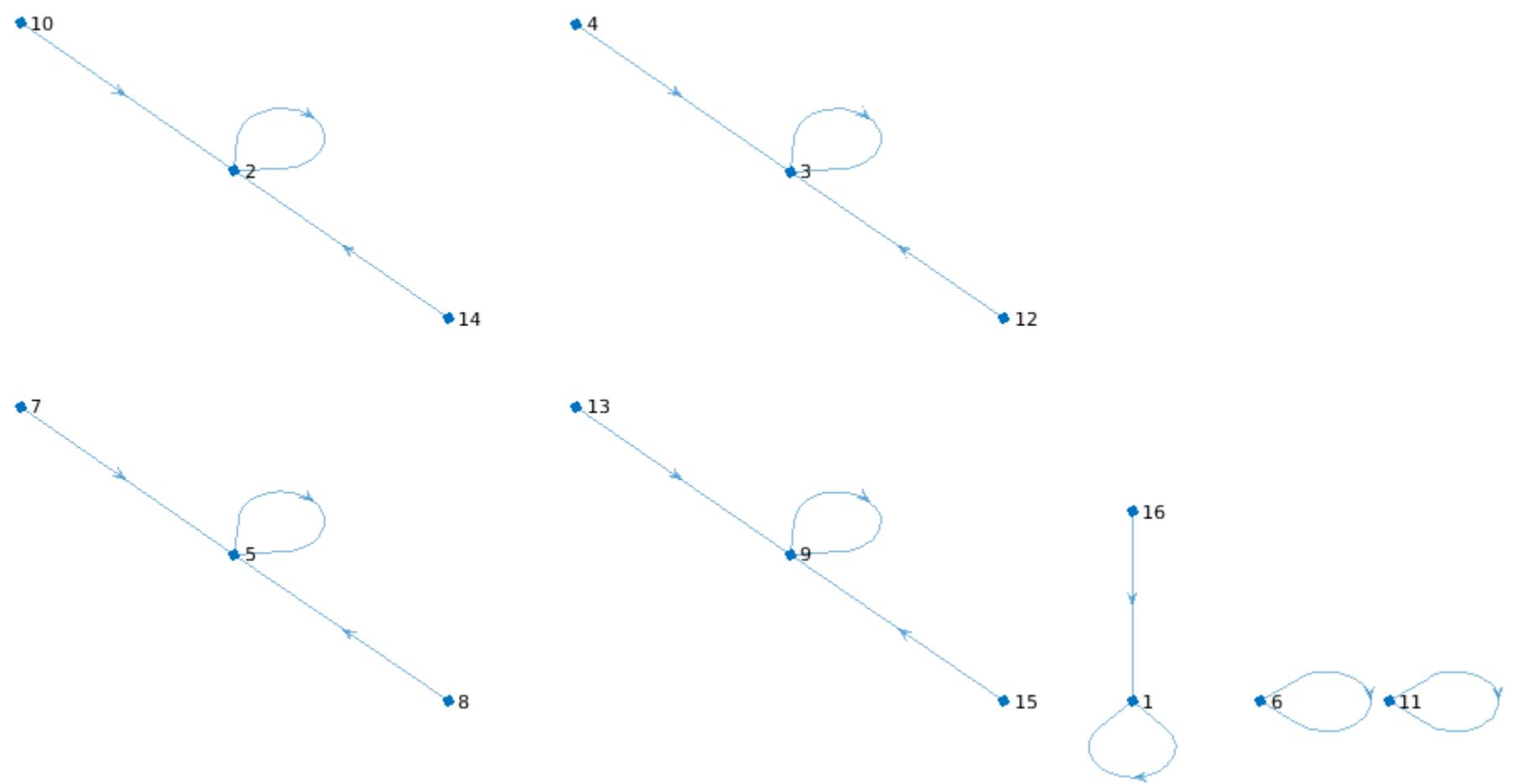


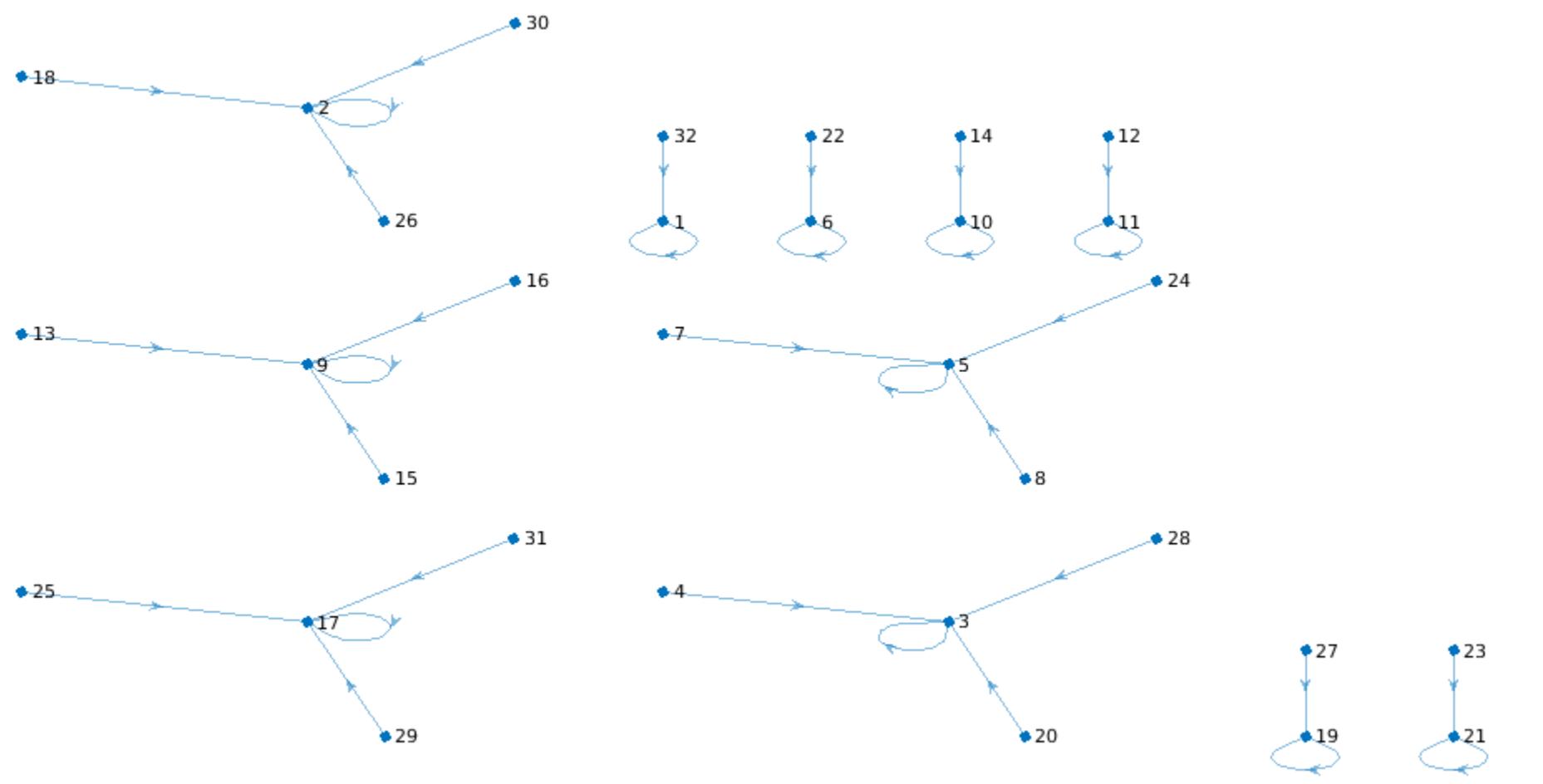


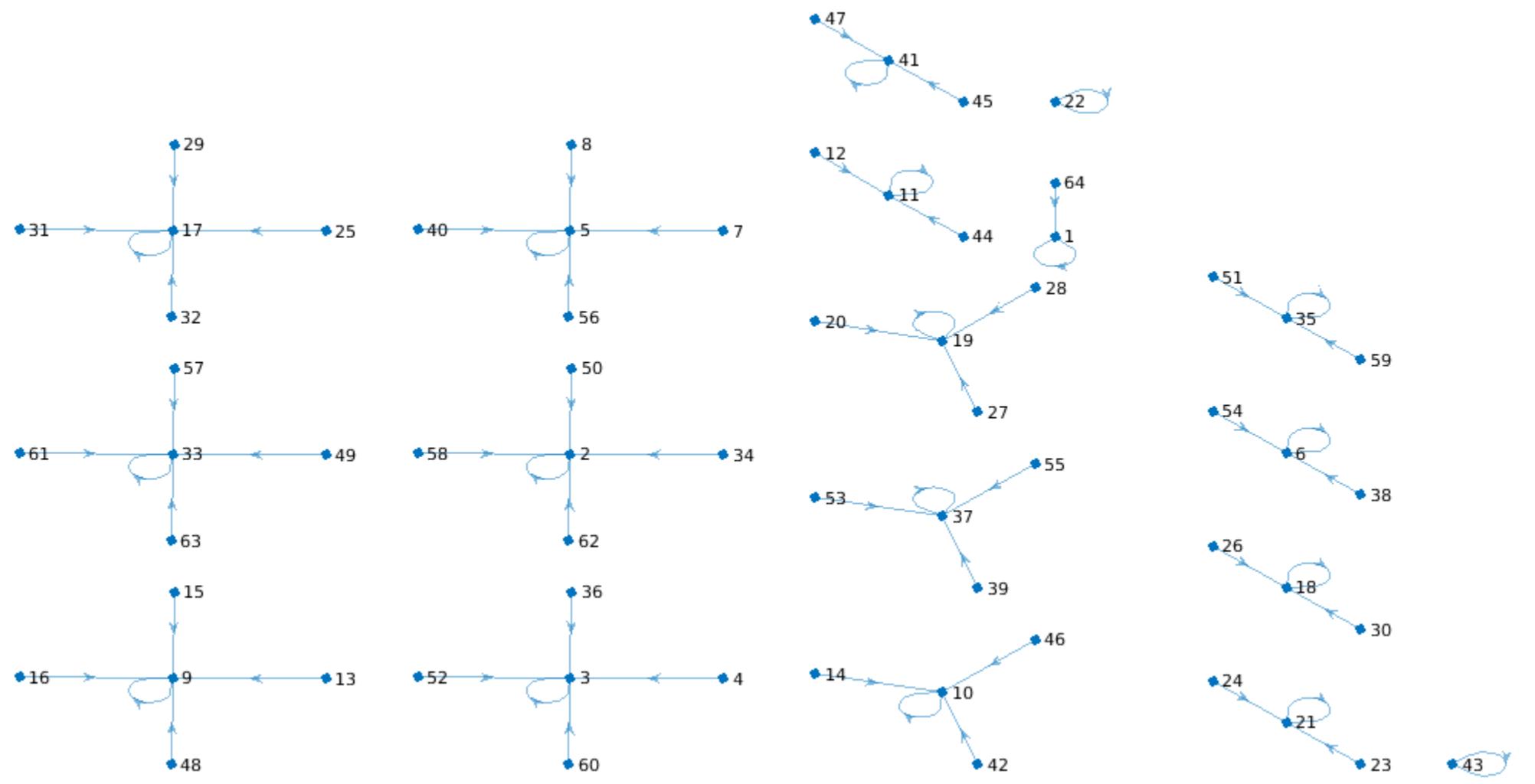
12

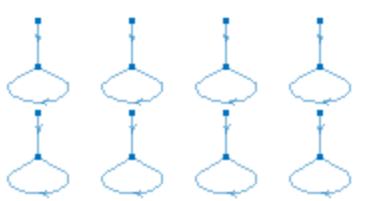
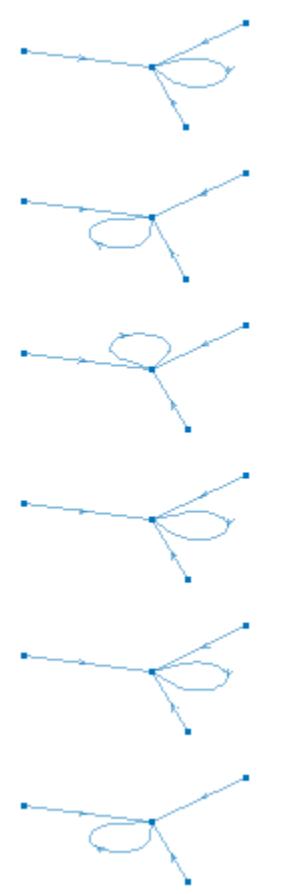
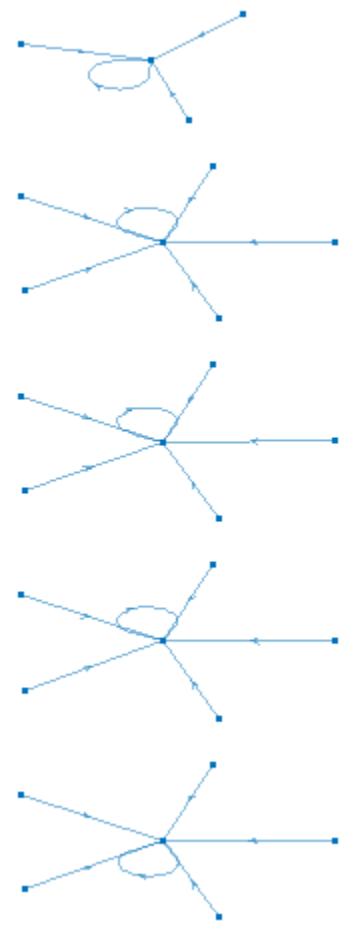
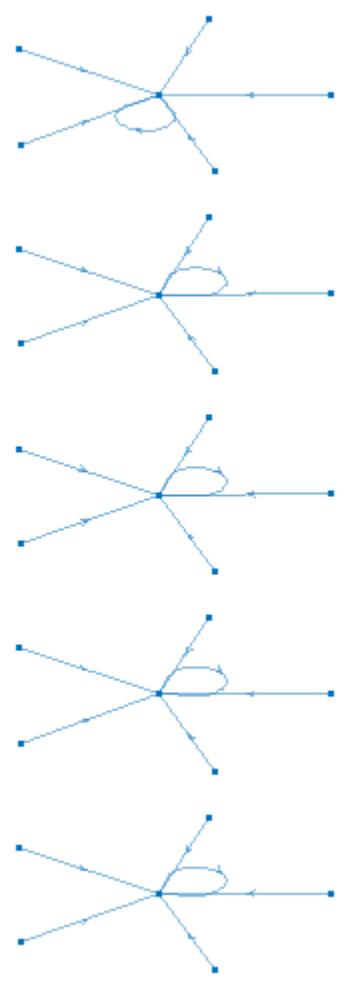
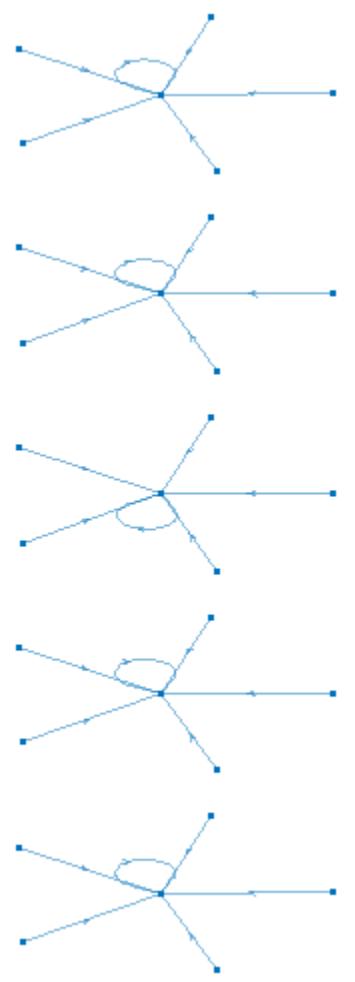


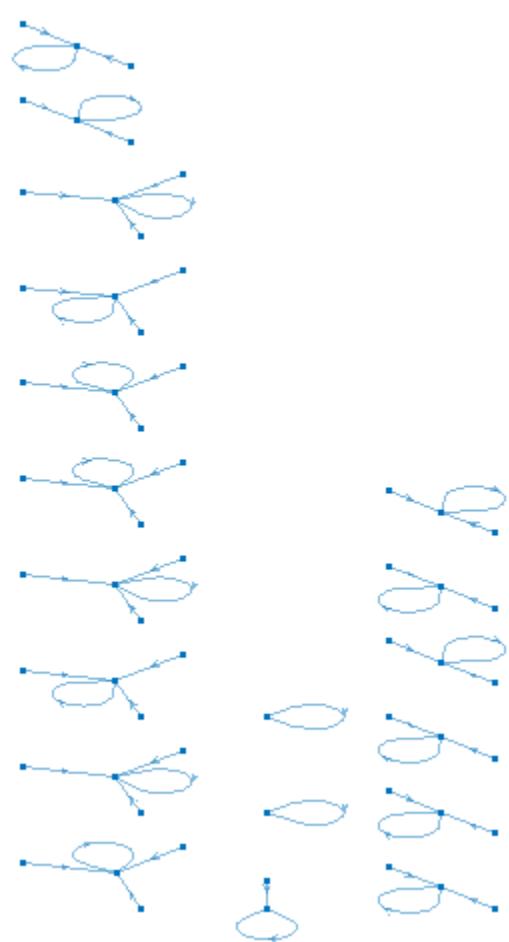
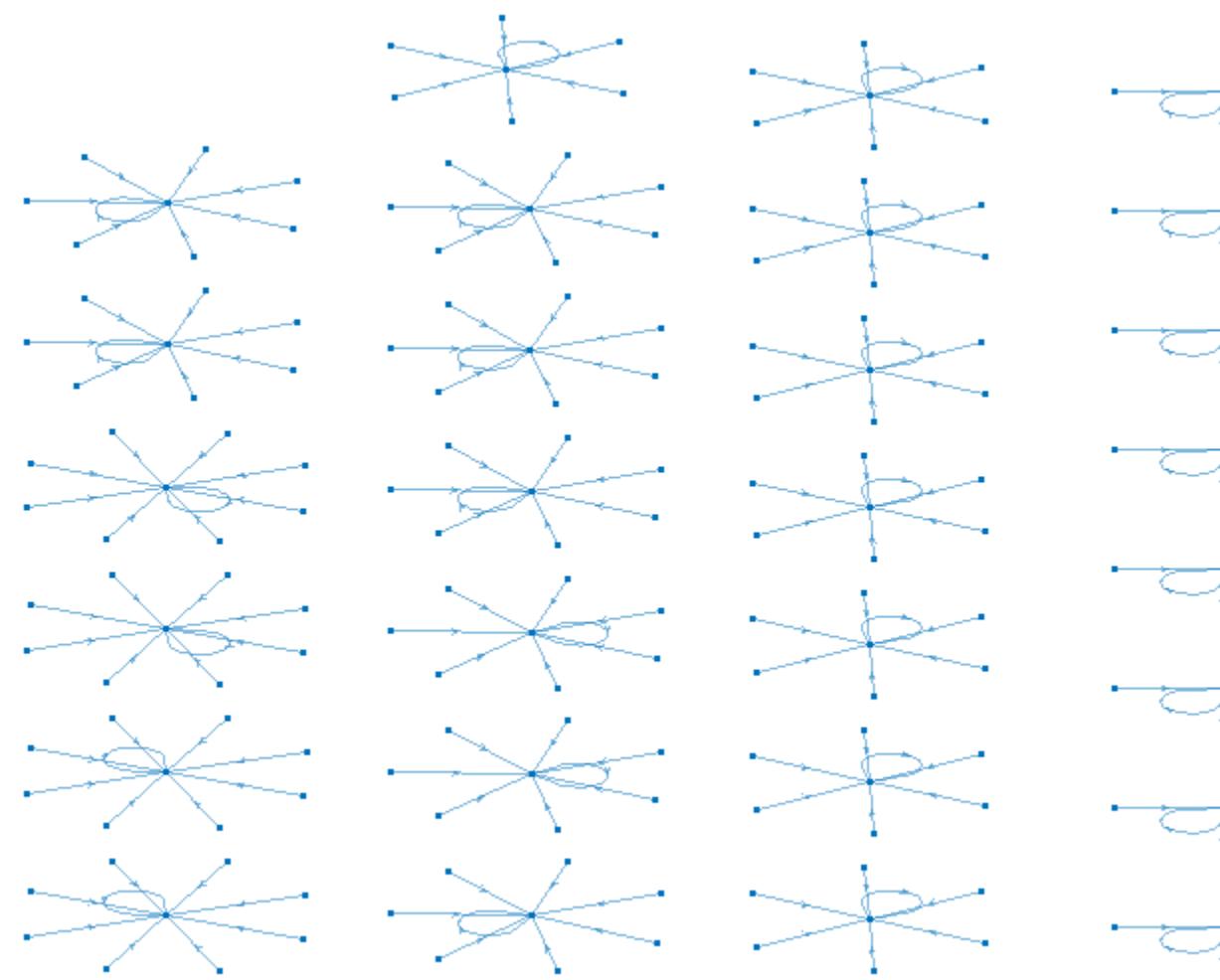


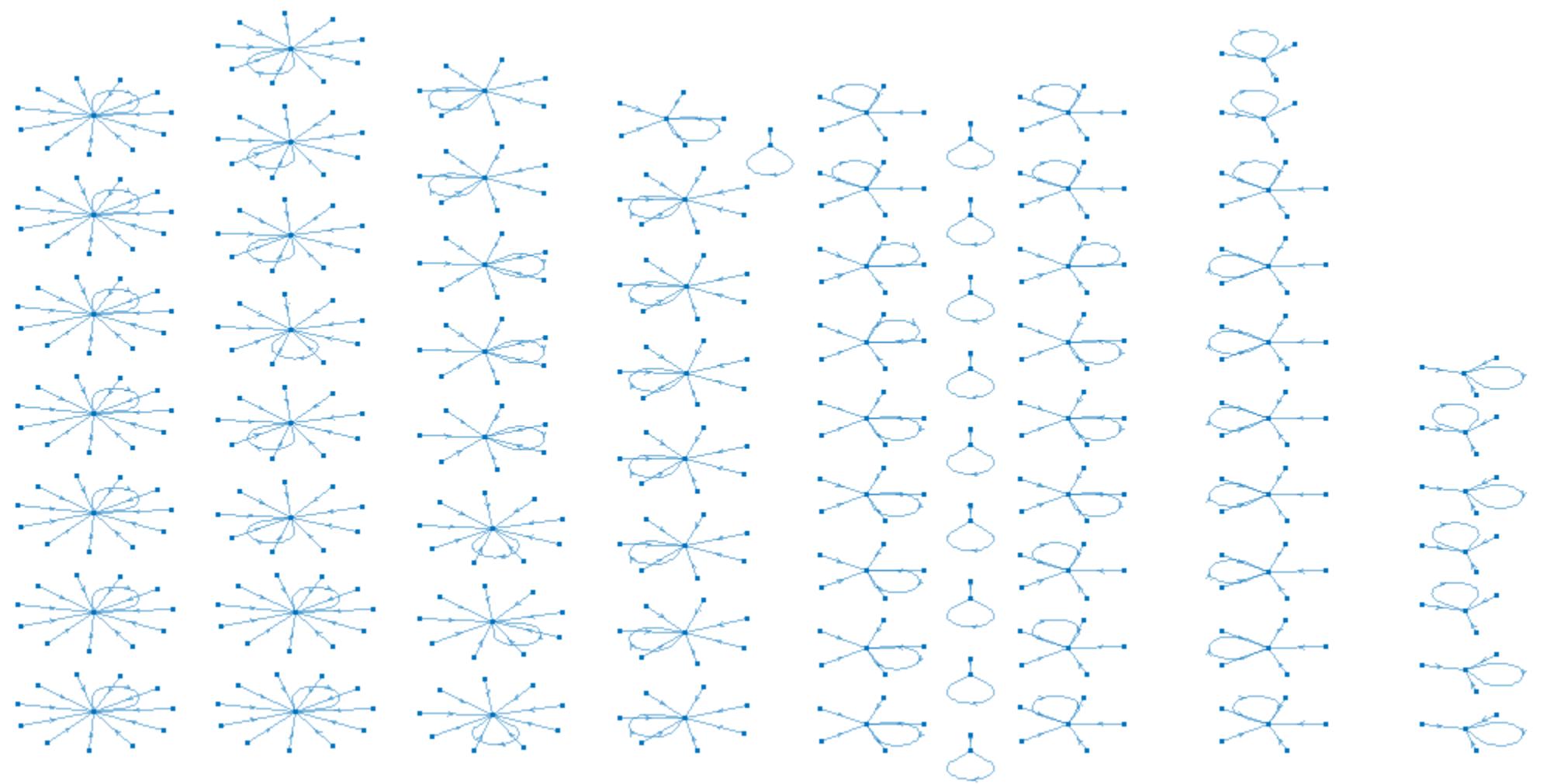


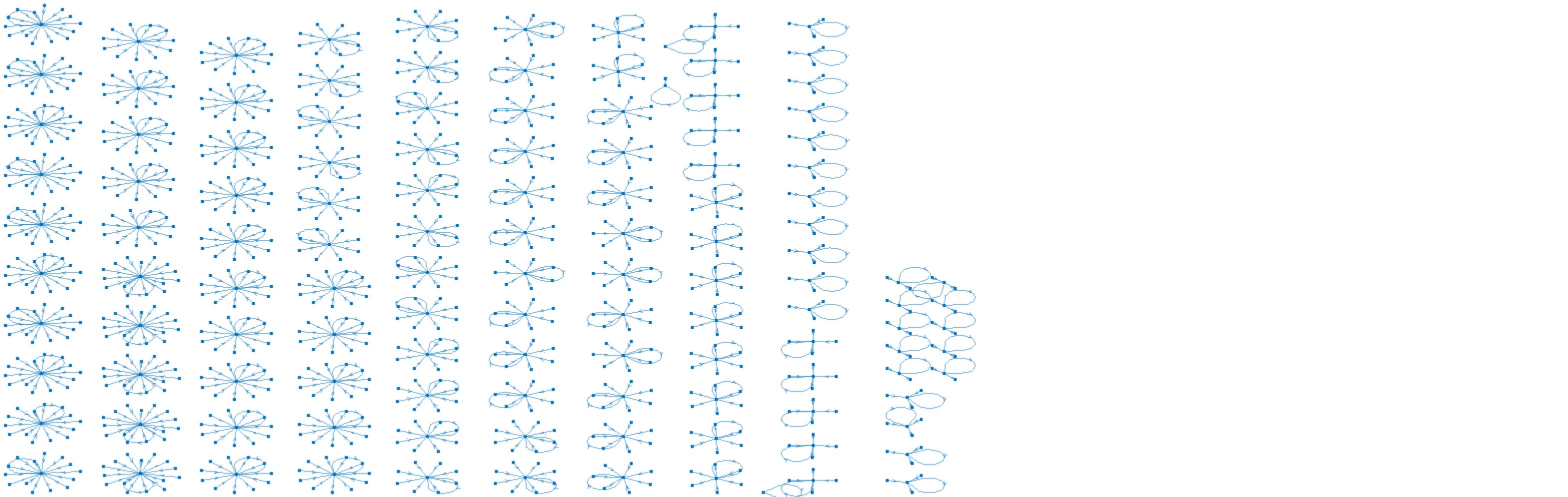


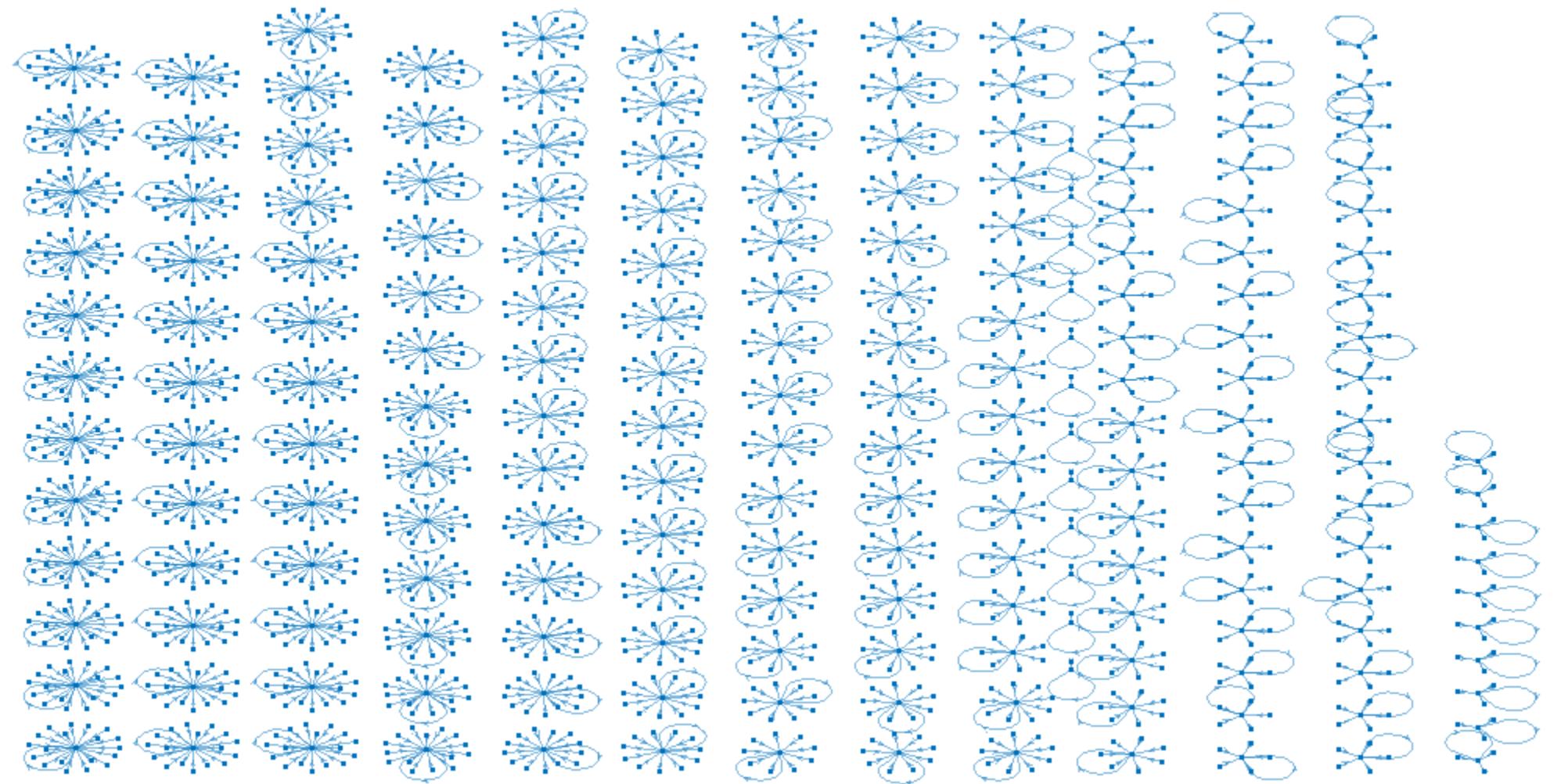


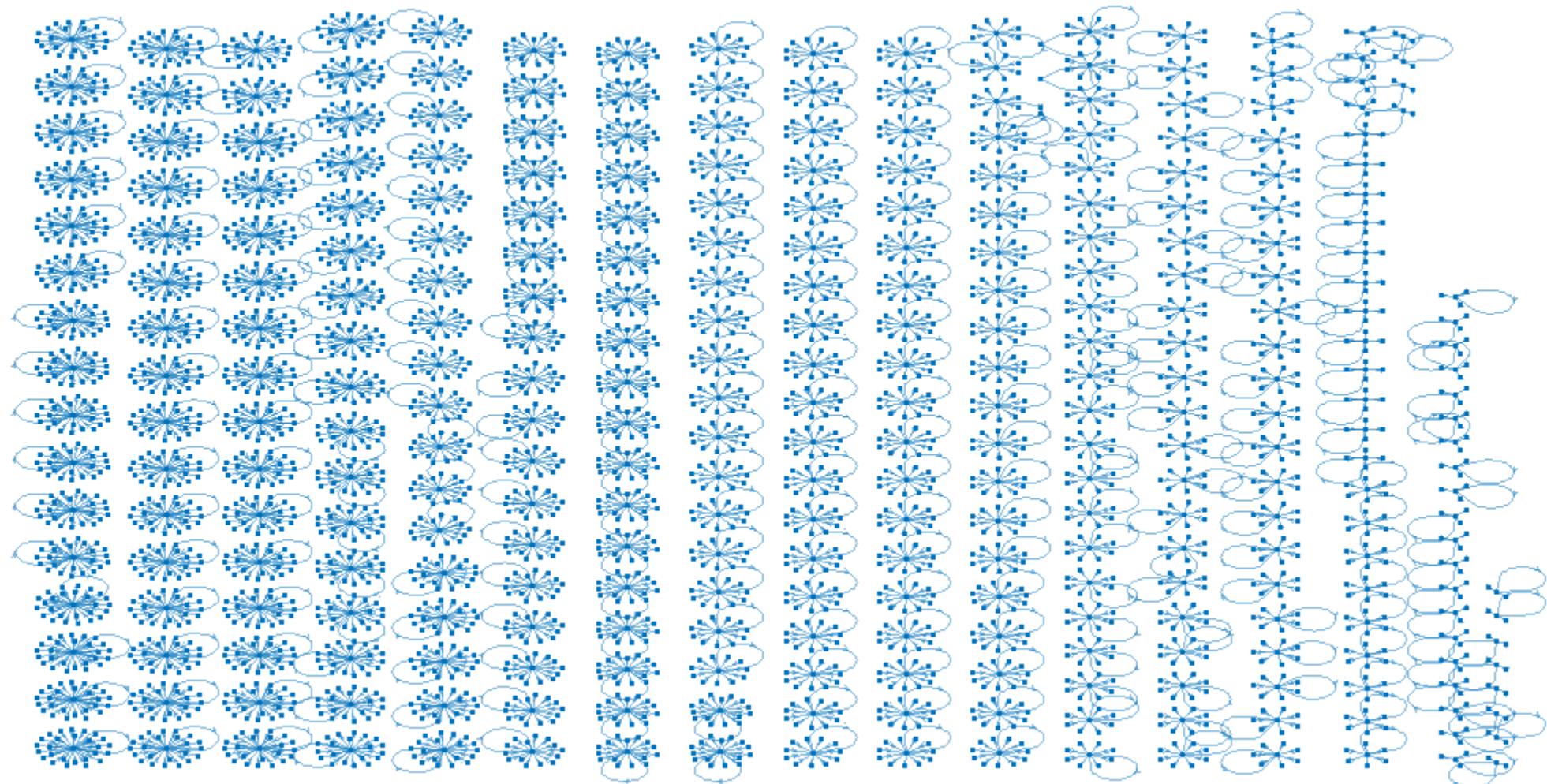


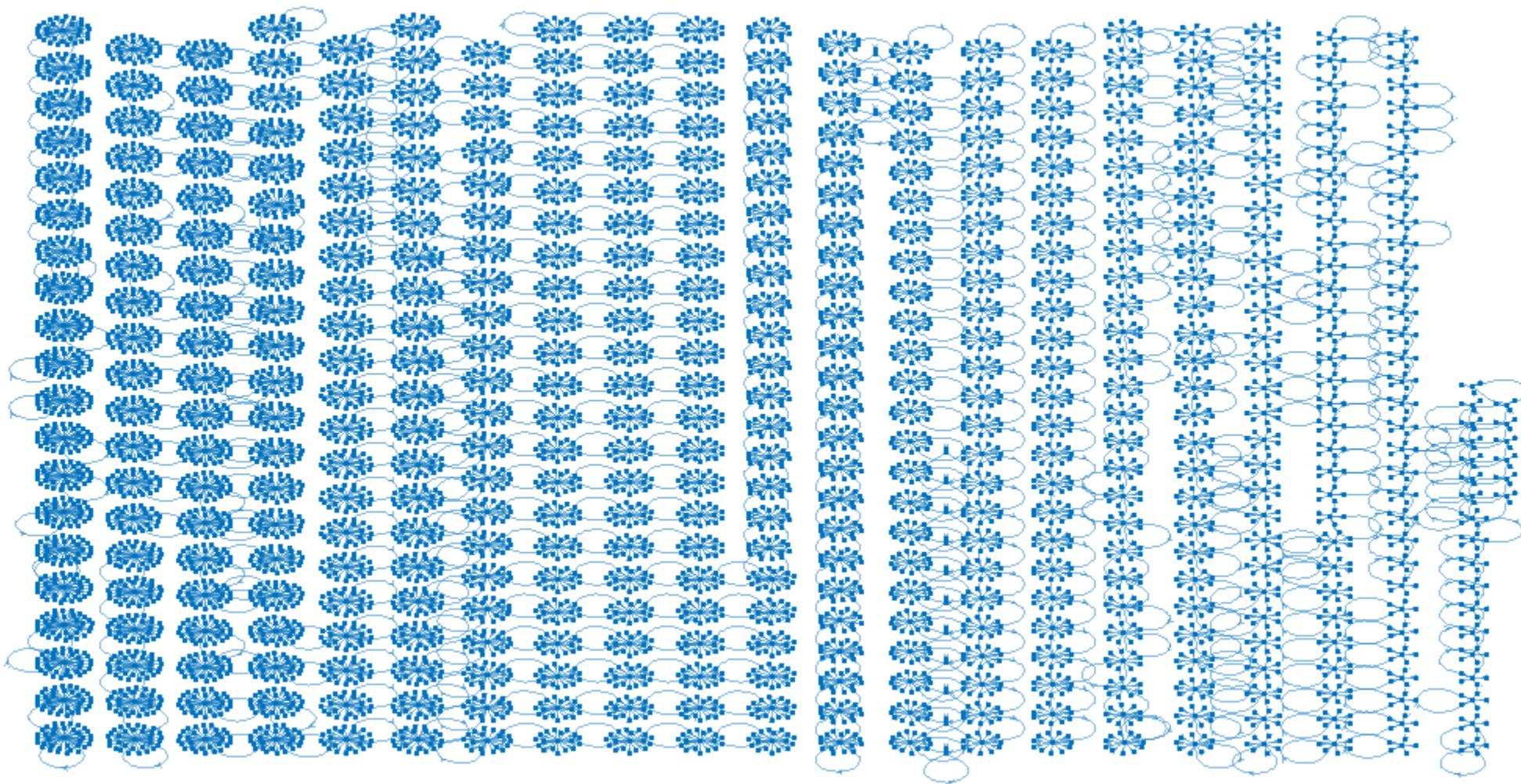






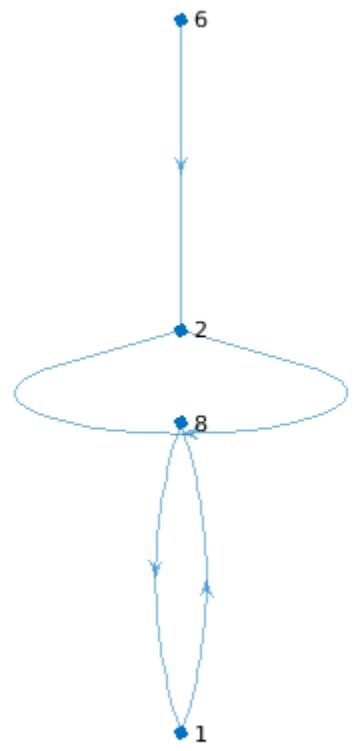


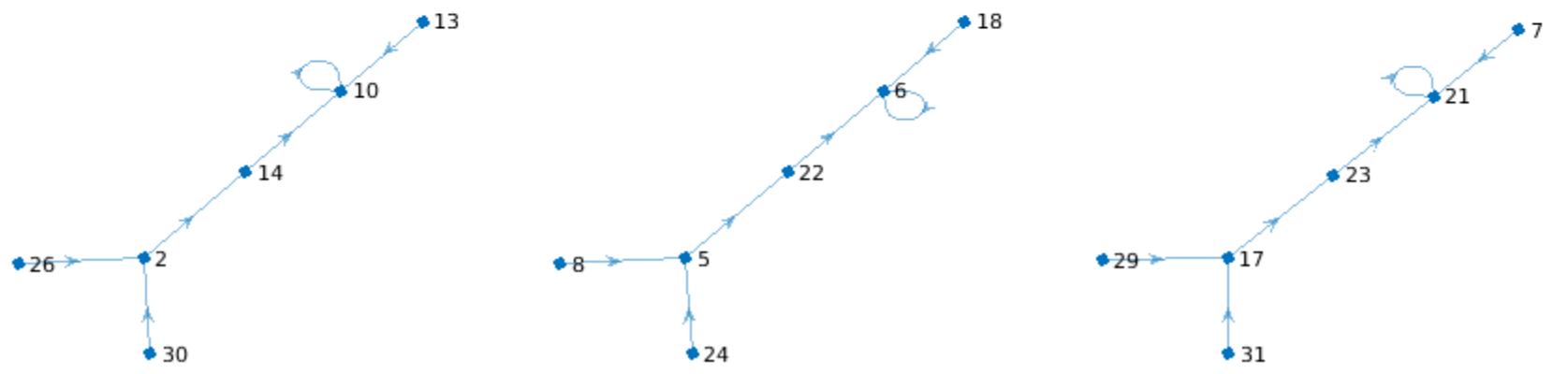
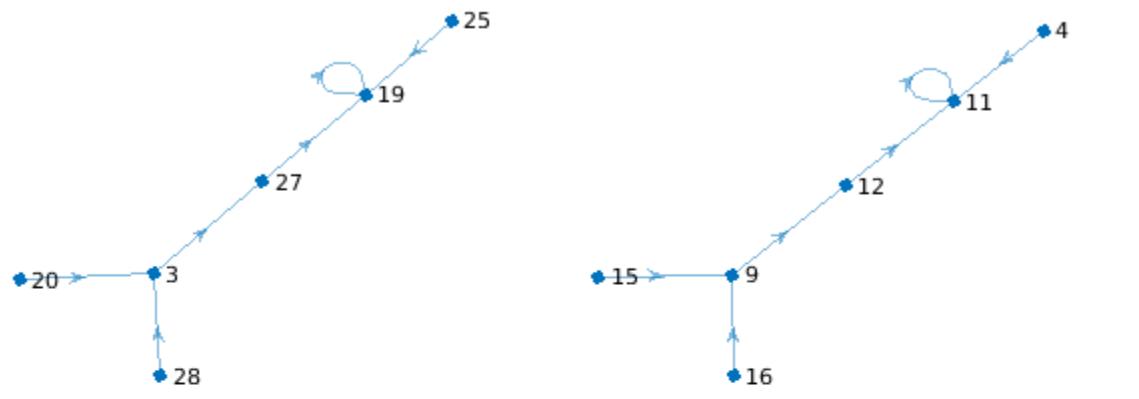
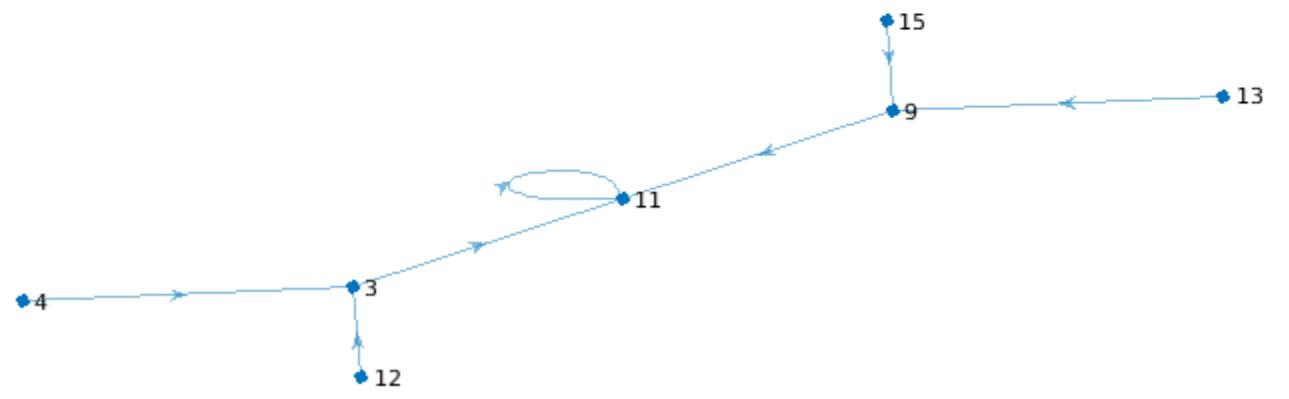
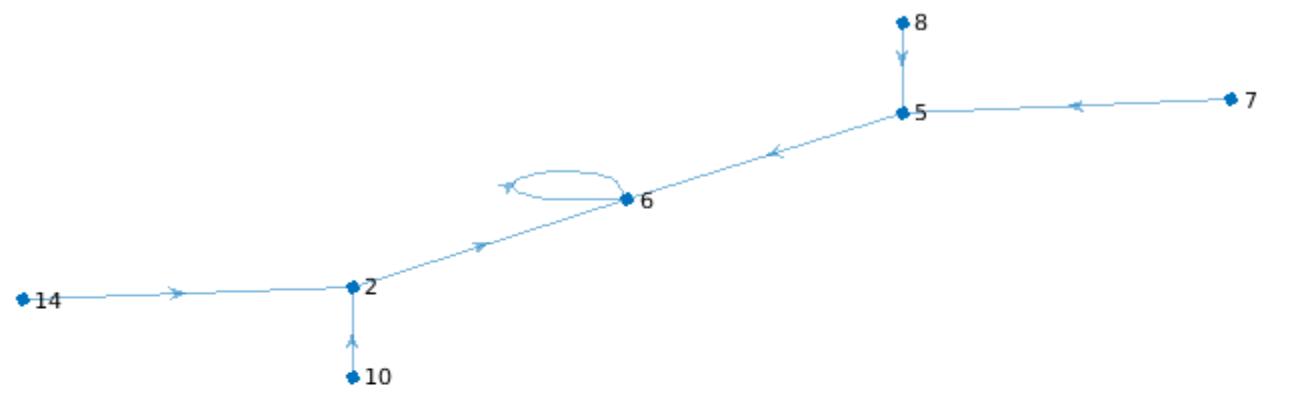


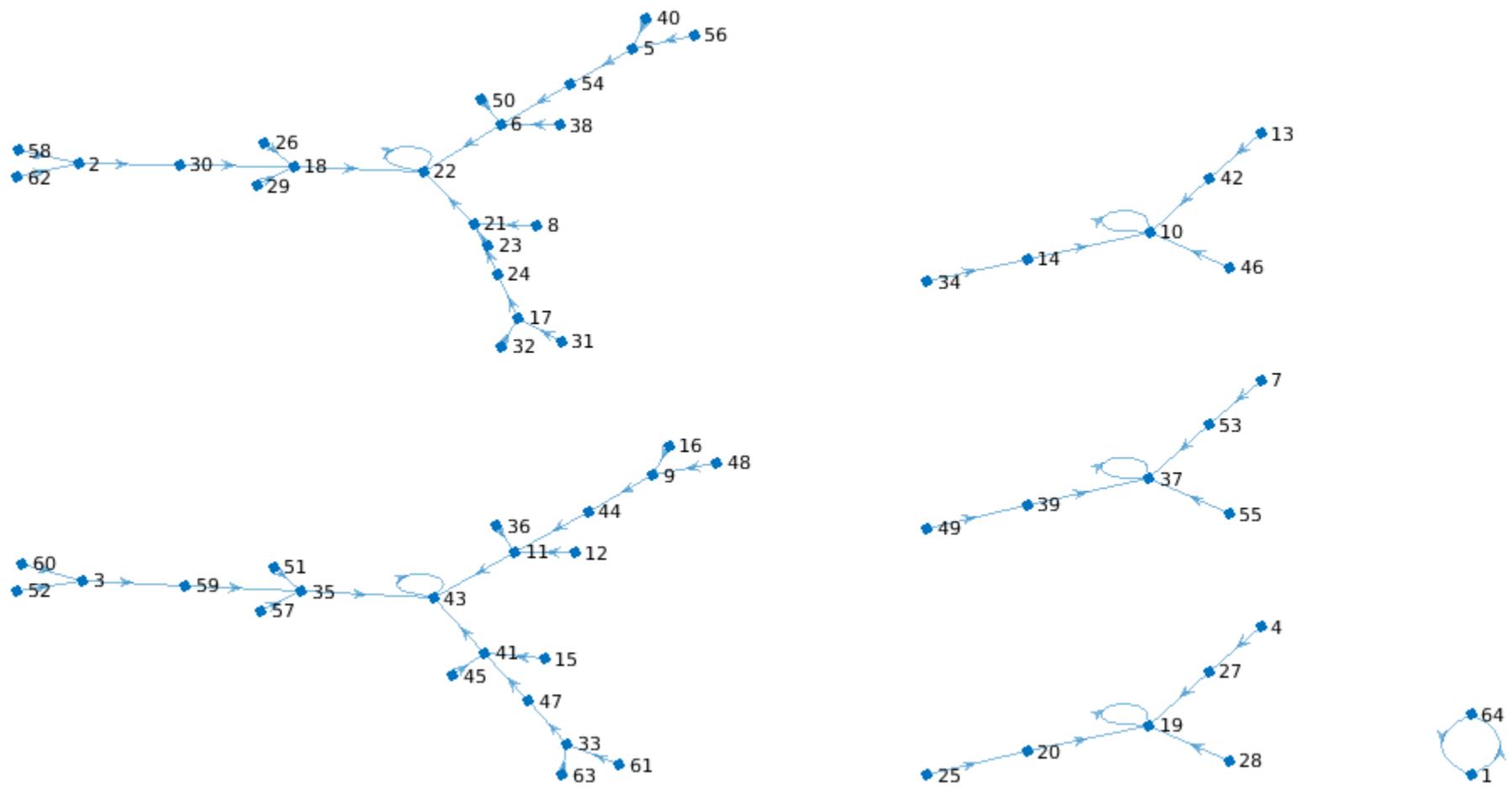


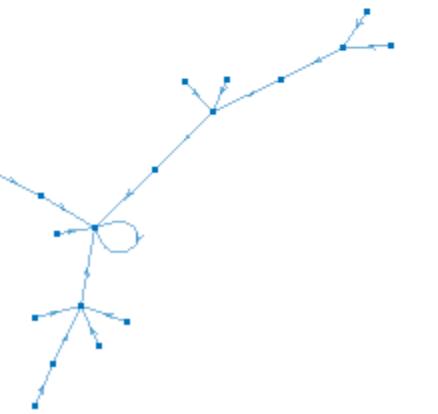
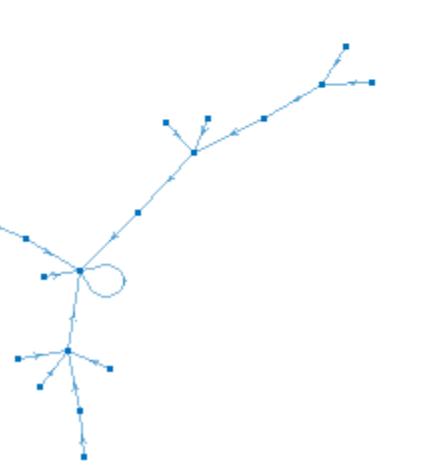
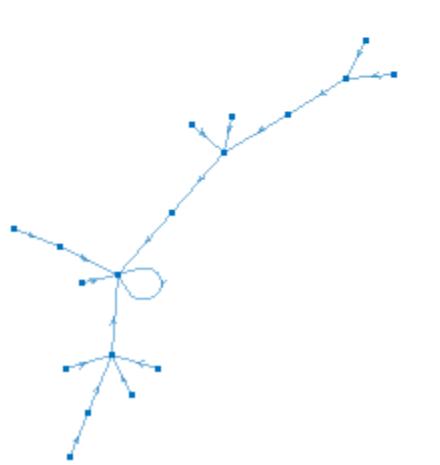
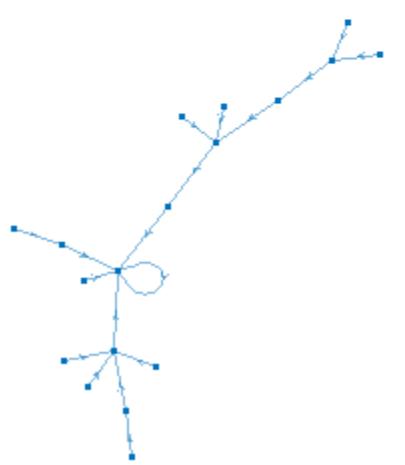
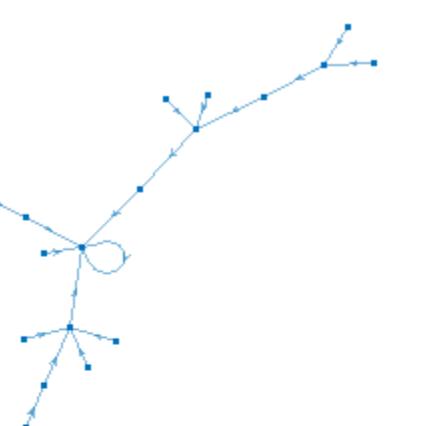
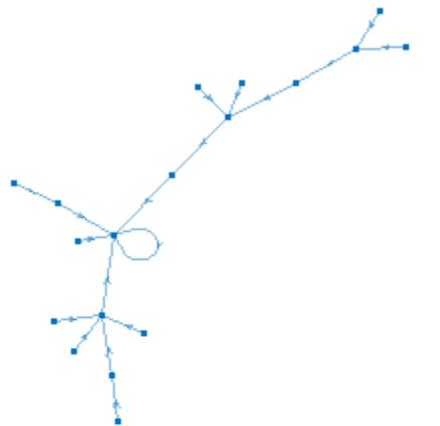
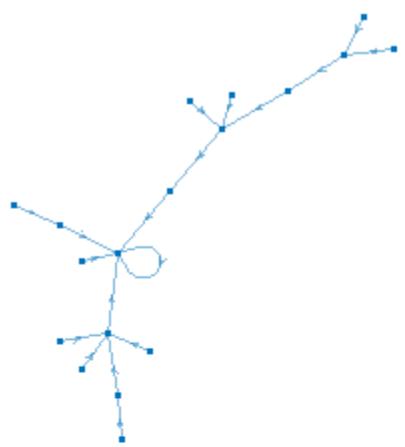
13

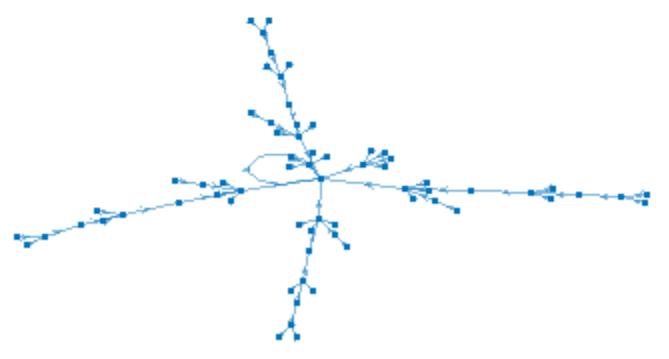
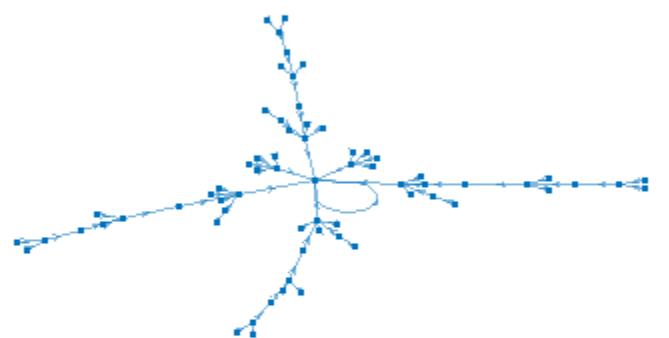
a

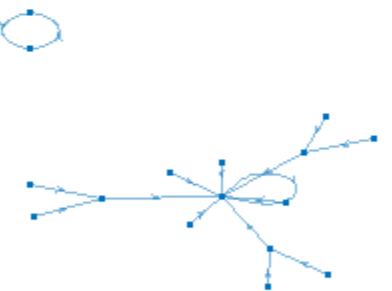
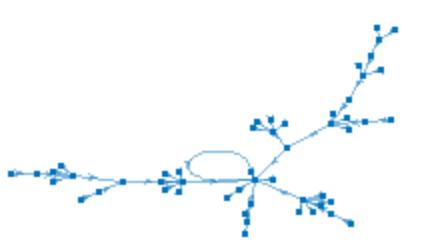
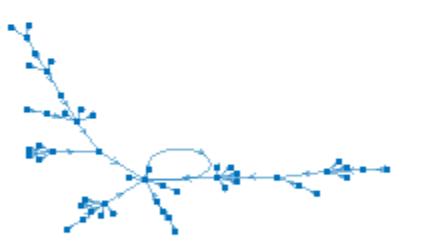
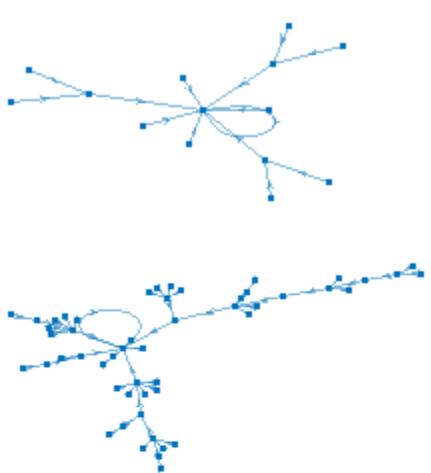
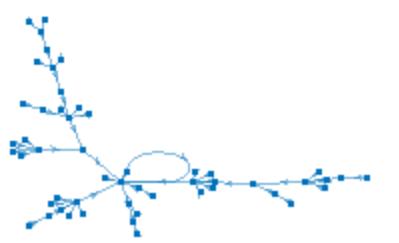
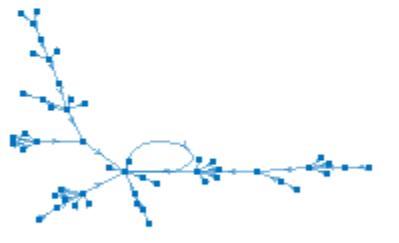
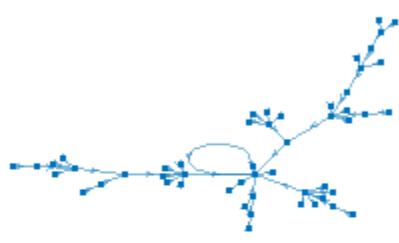
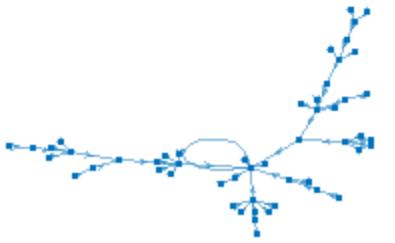
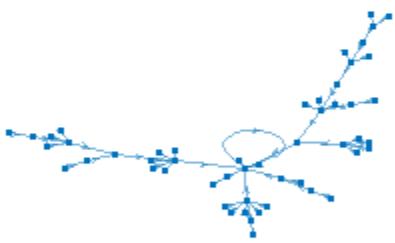
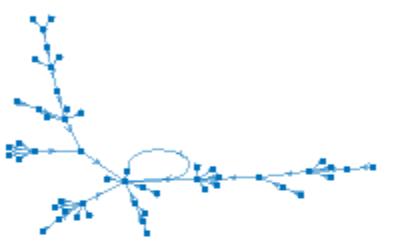


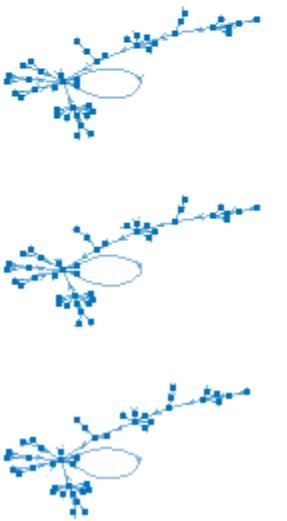
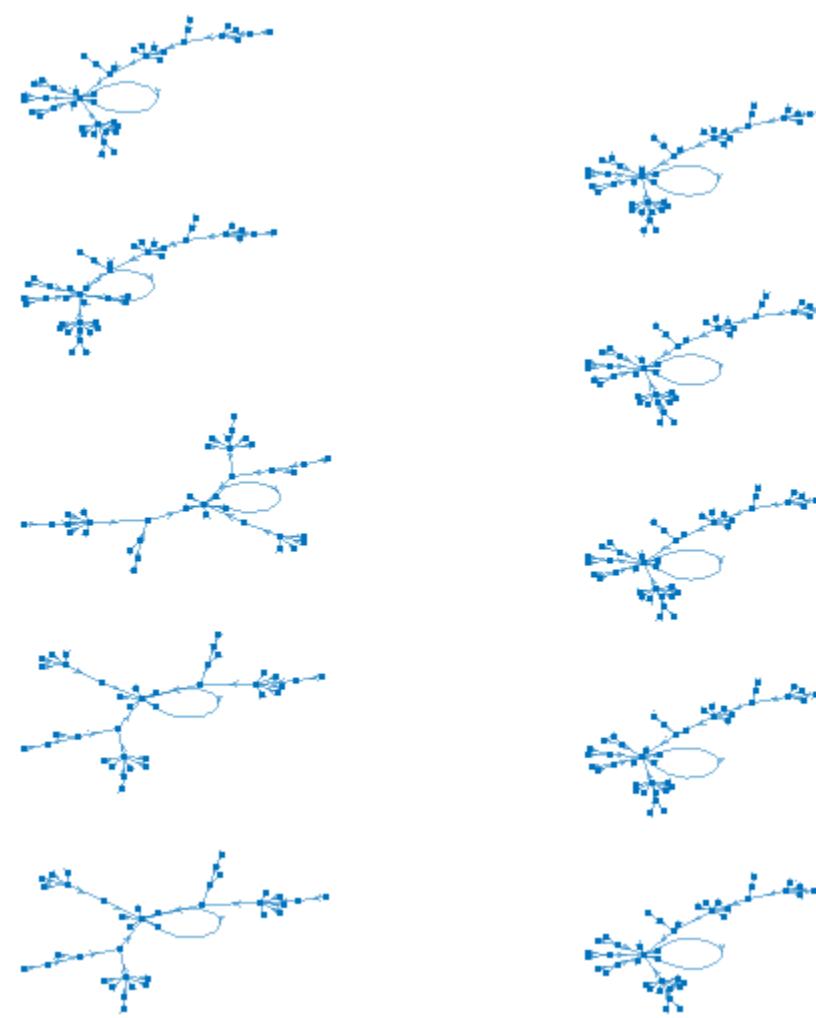
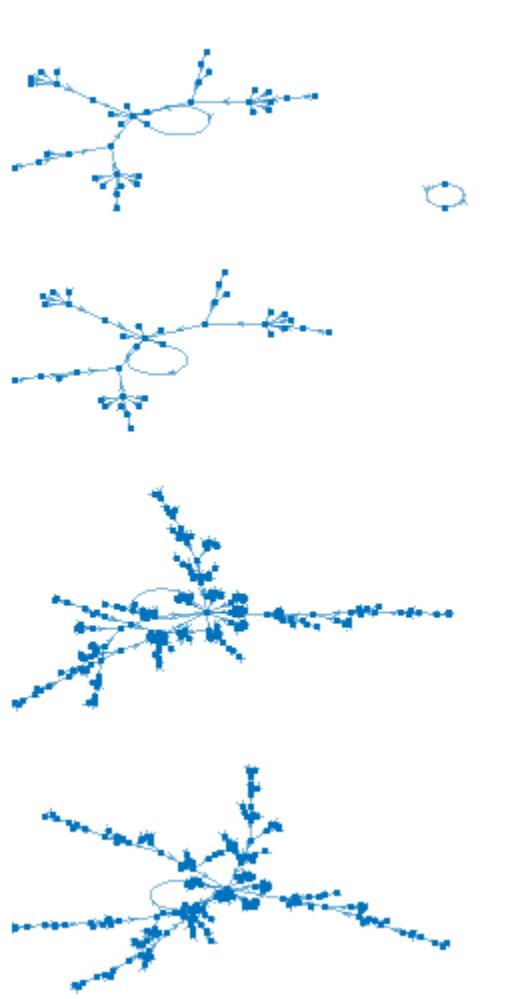


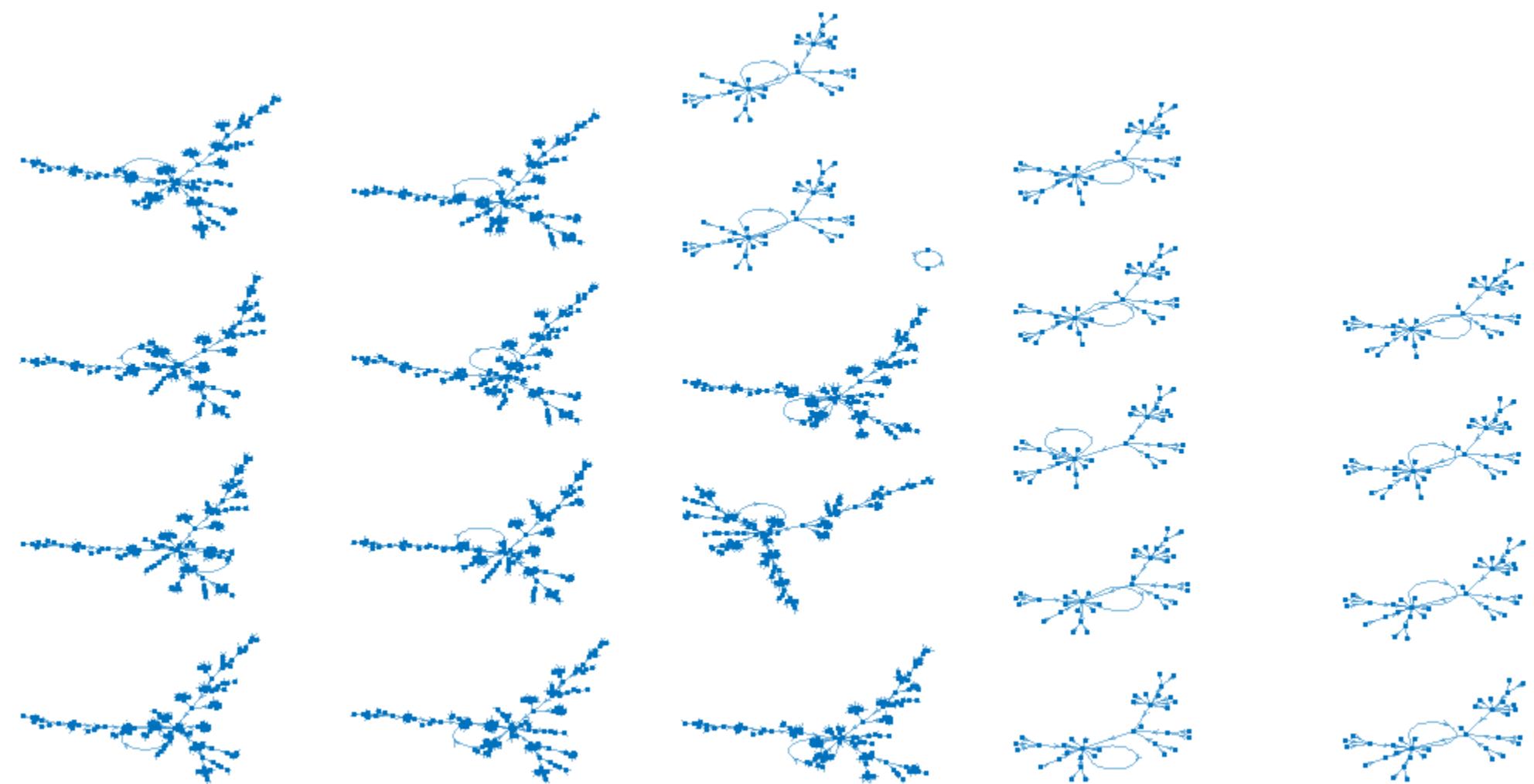


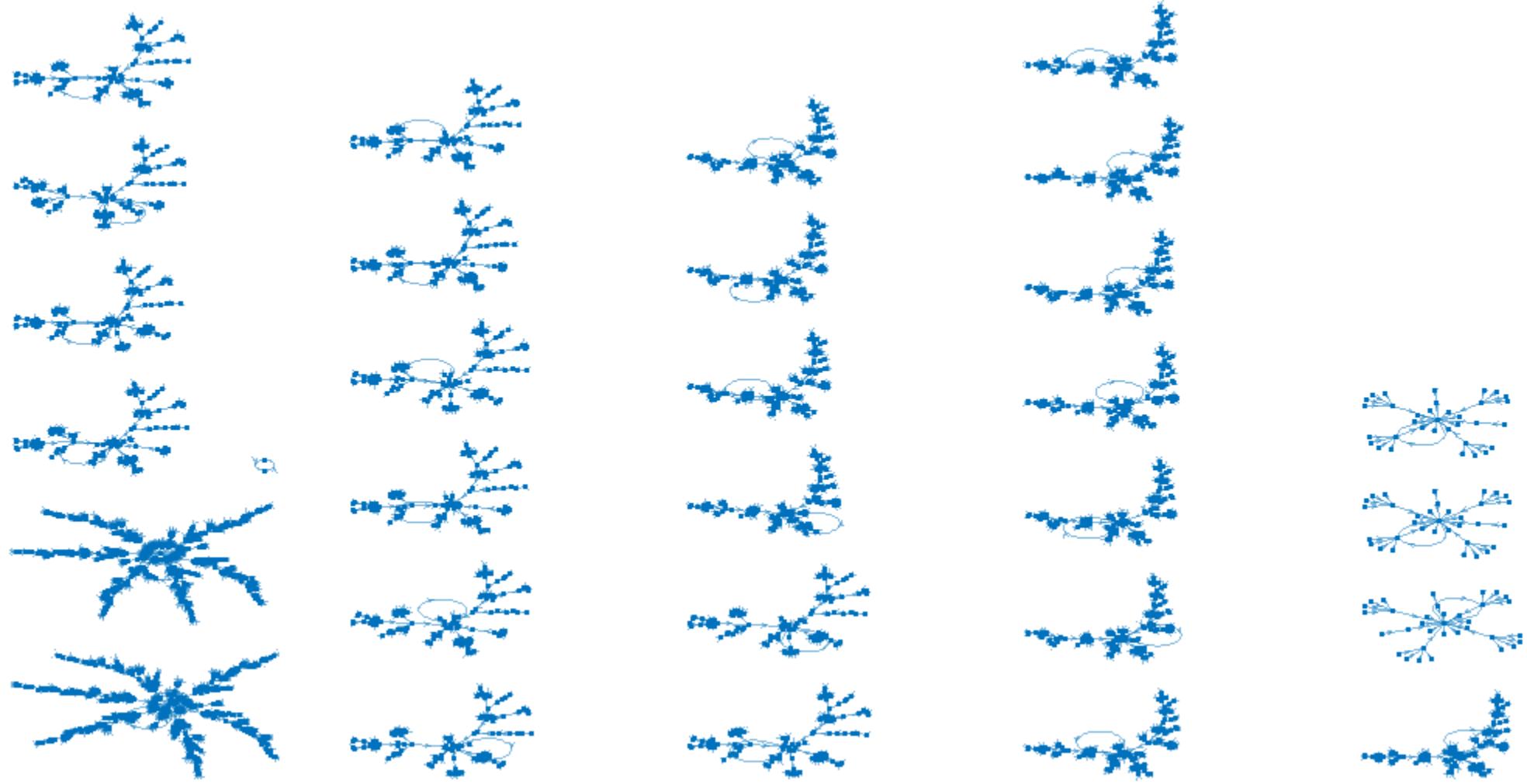


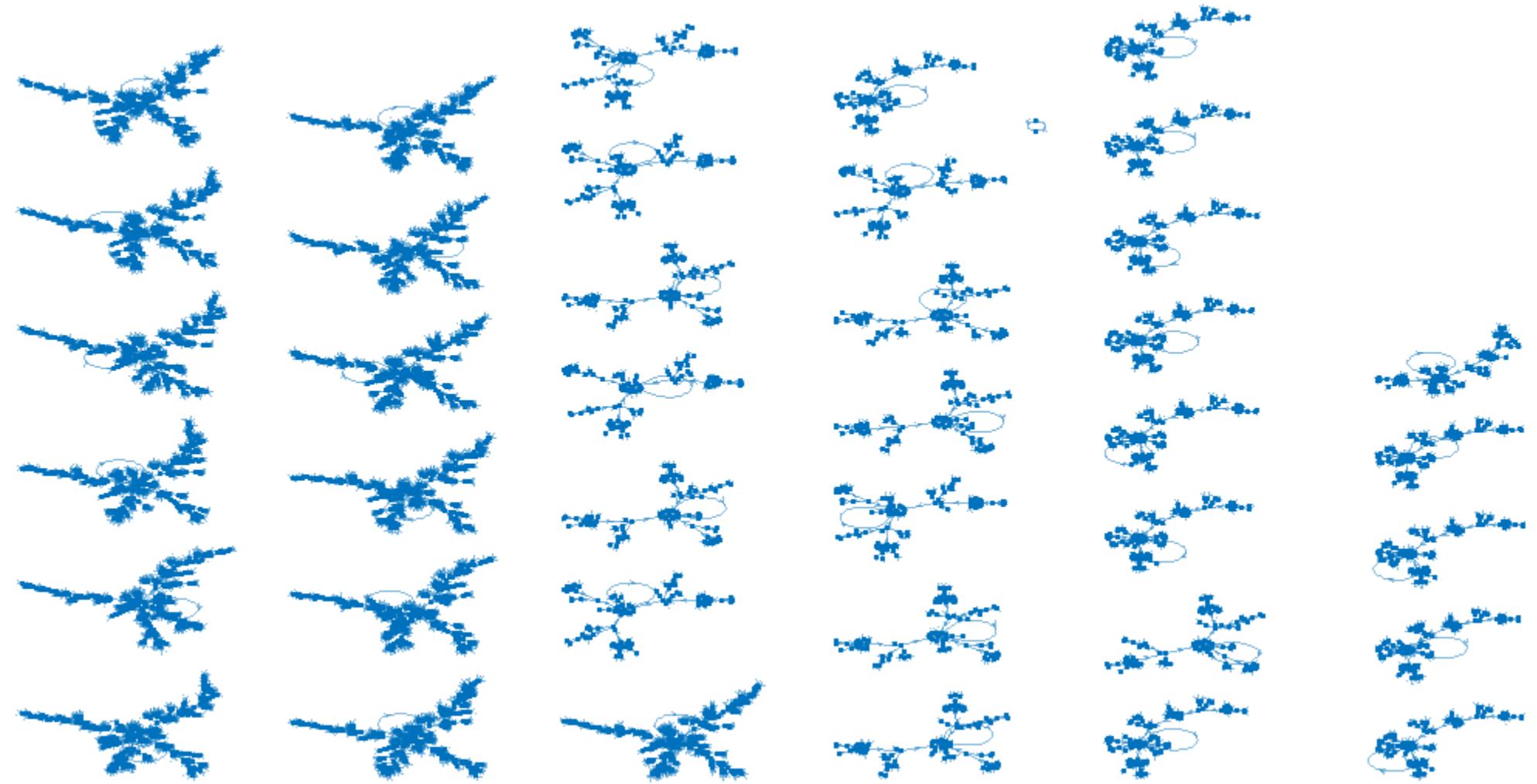


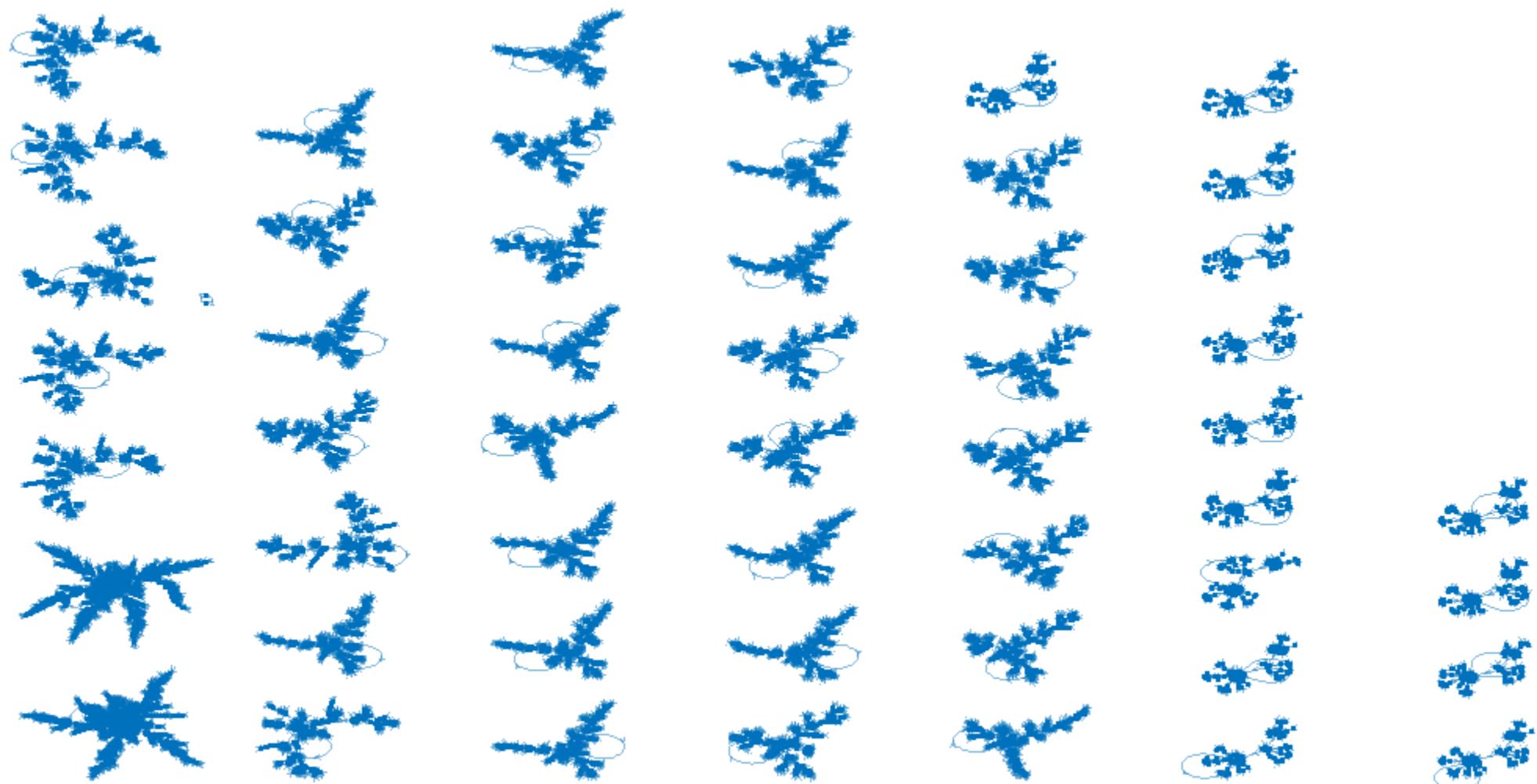


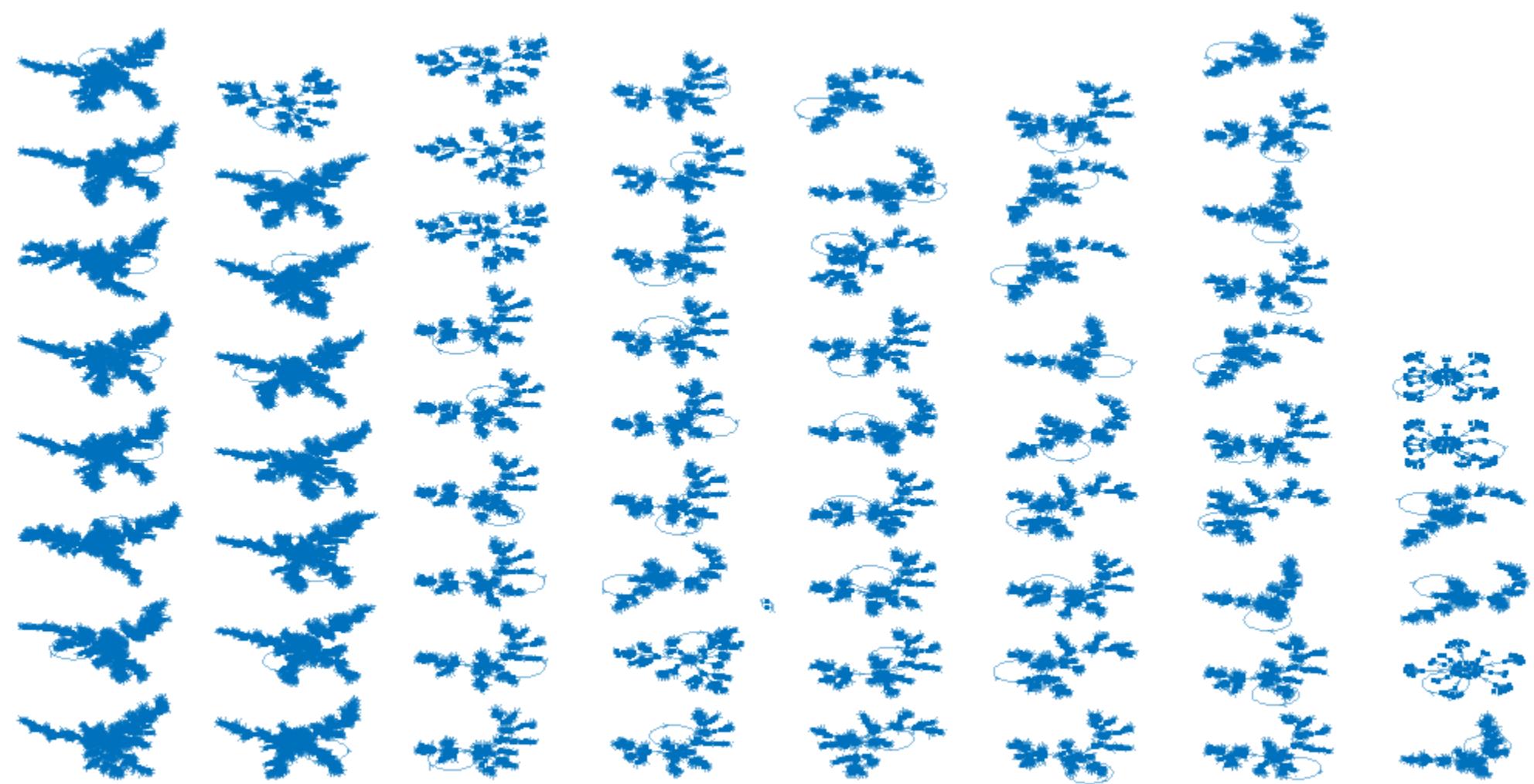


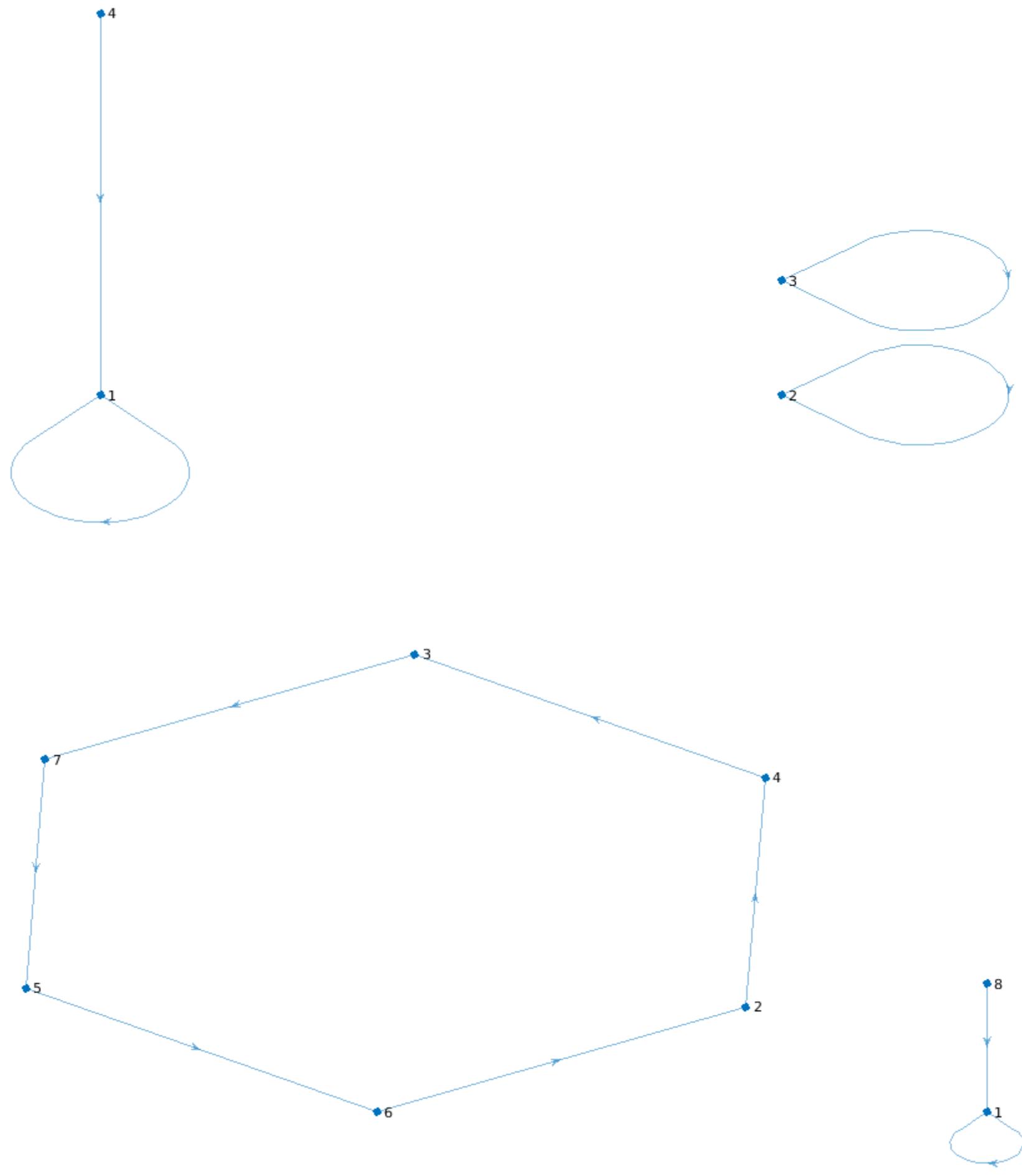


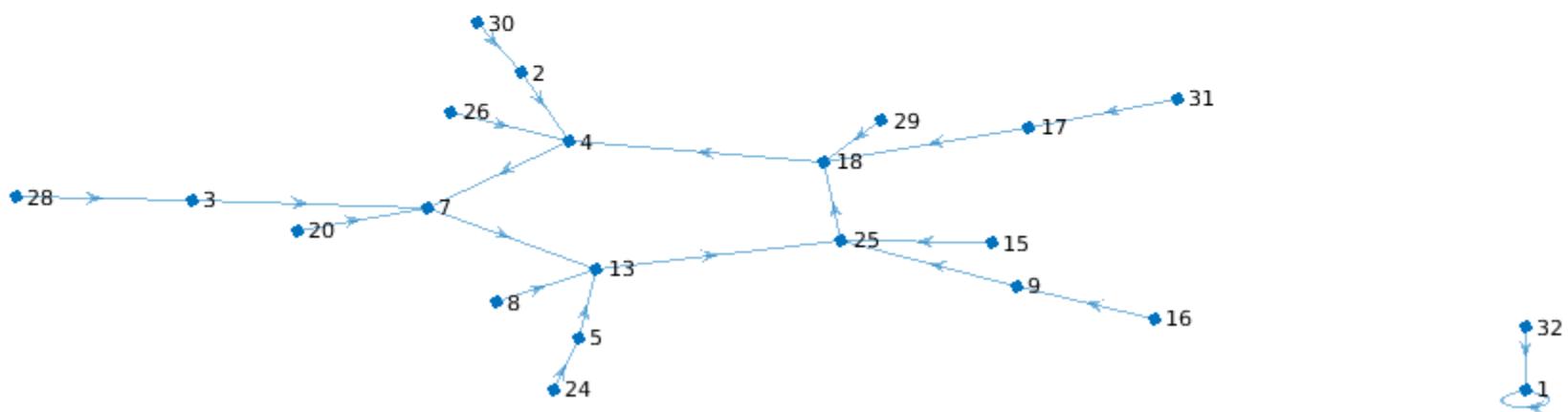
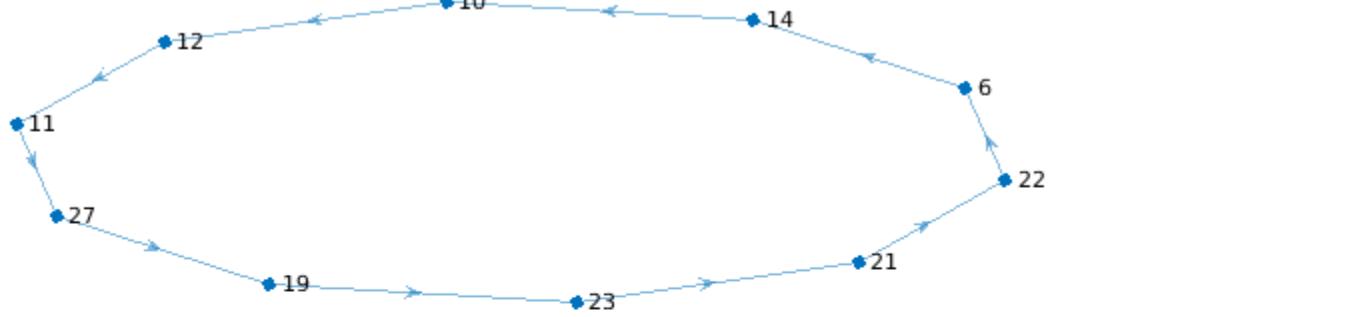
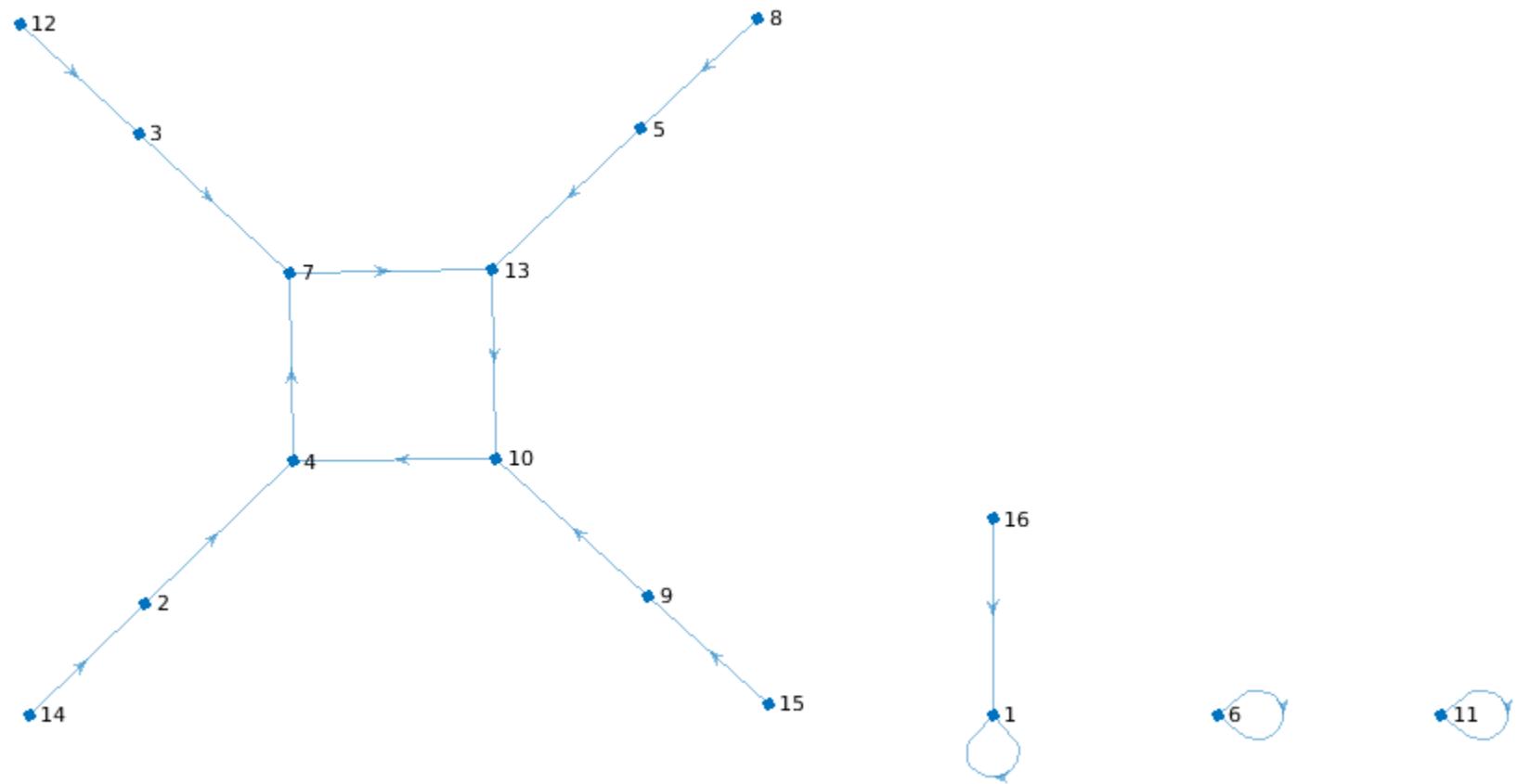


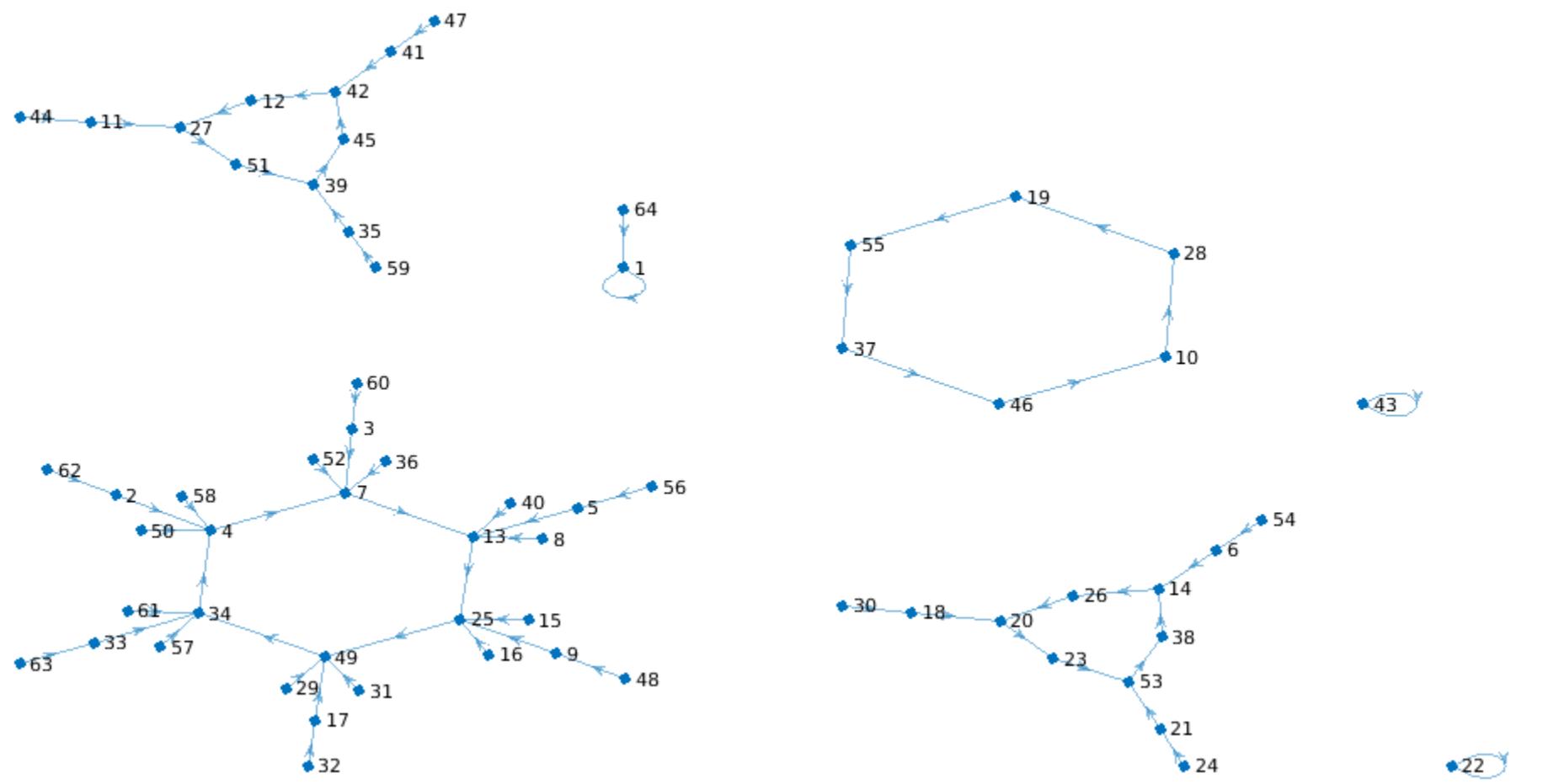


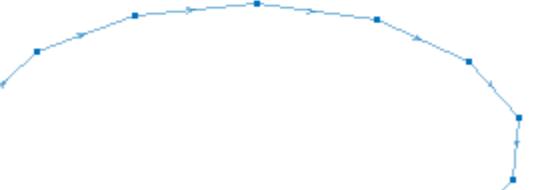
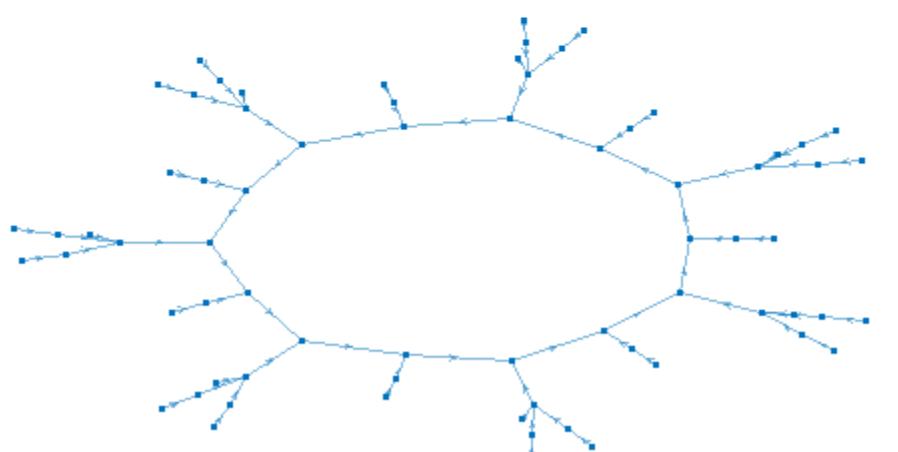
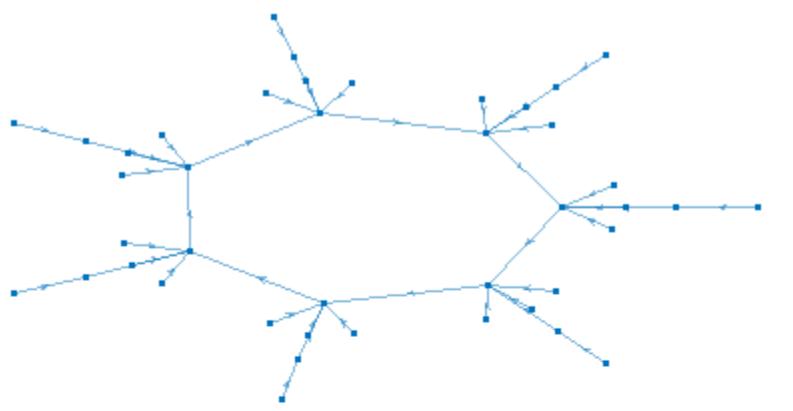


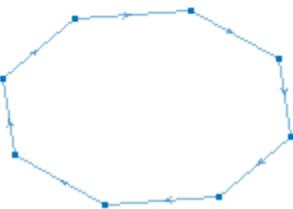
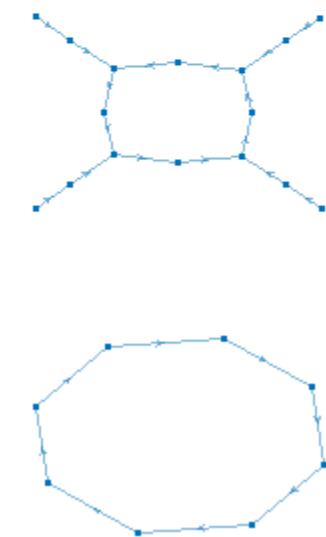
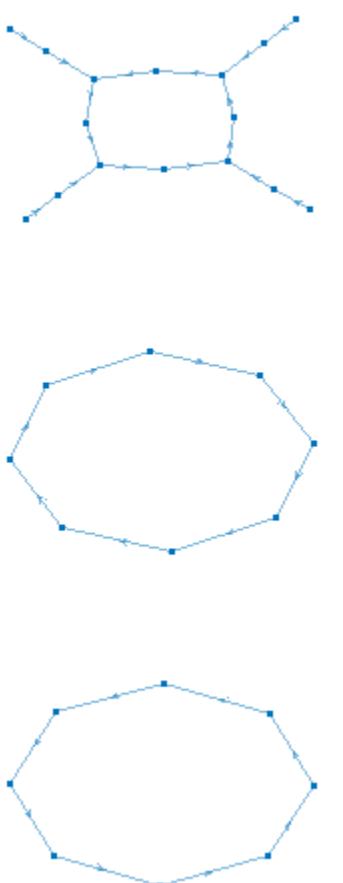
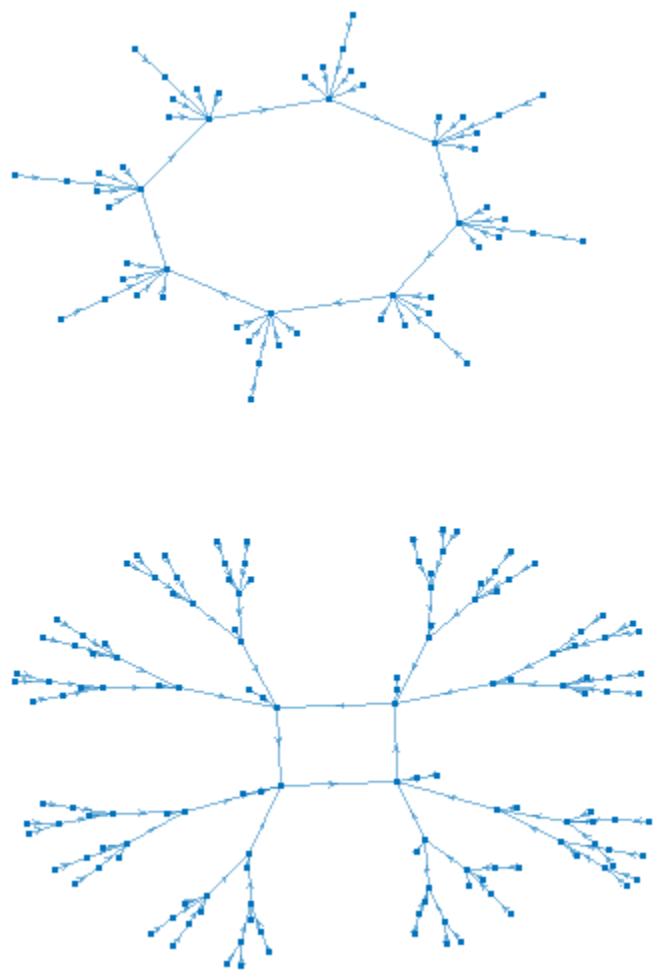


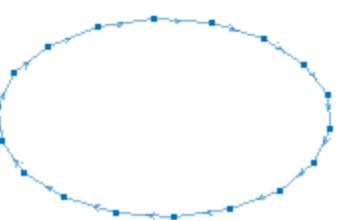
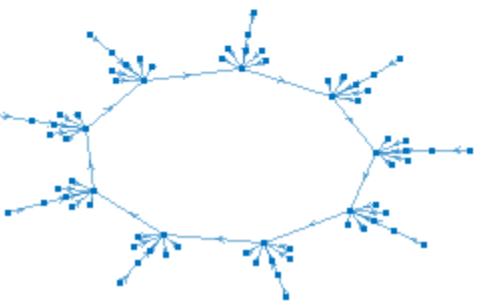
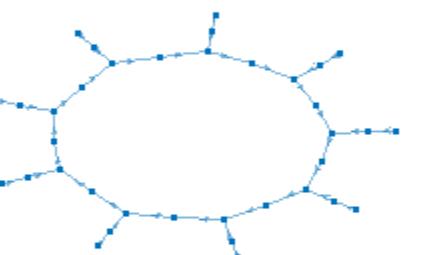
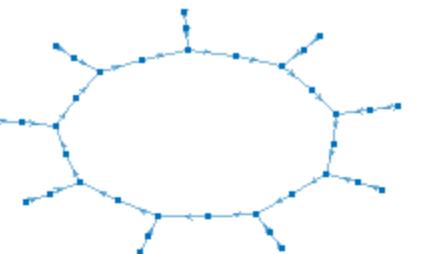
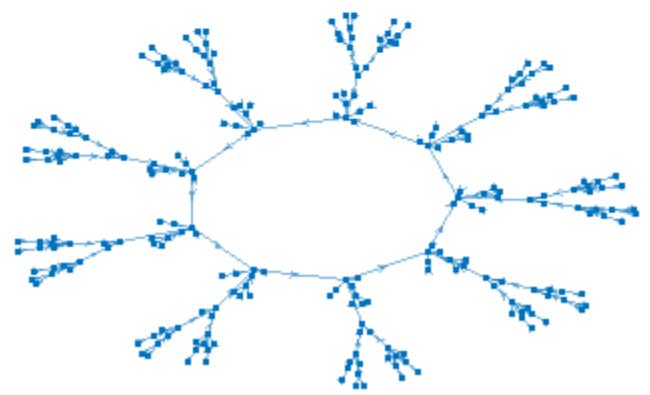
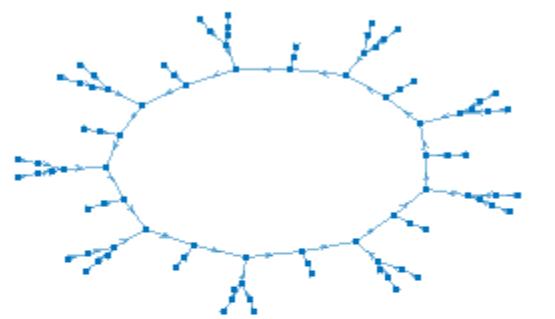


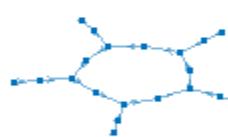
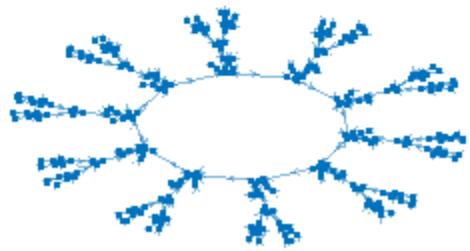
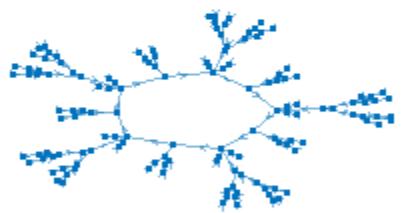
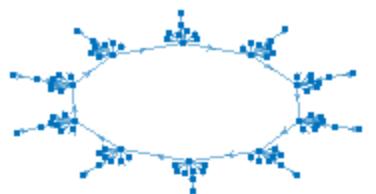


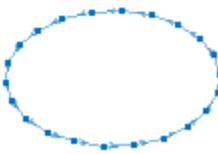
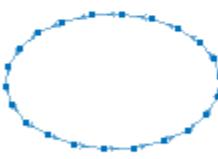
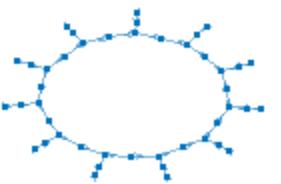
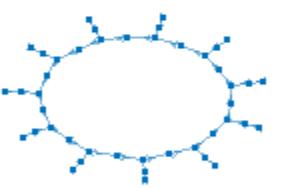
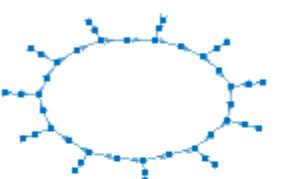
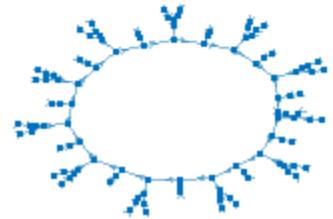
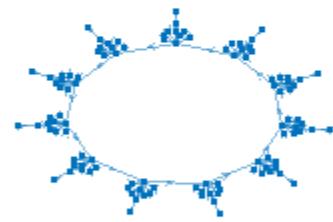
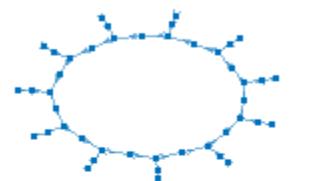
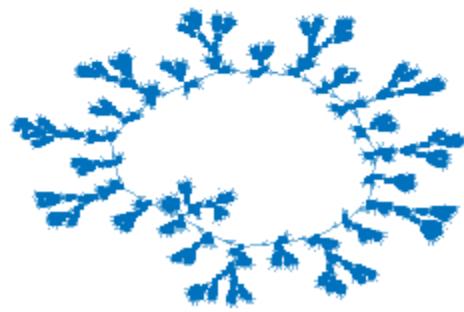
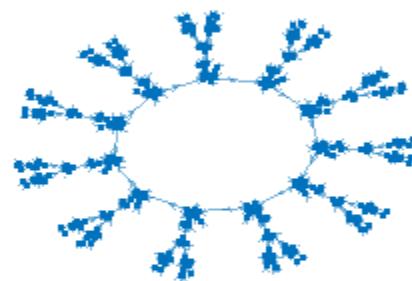
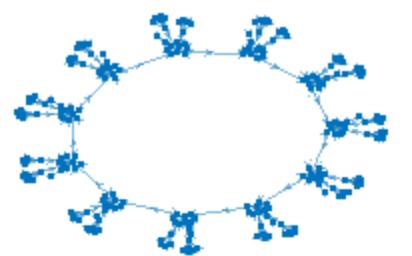


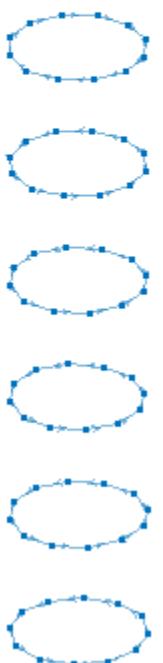
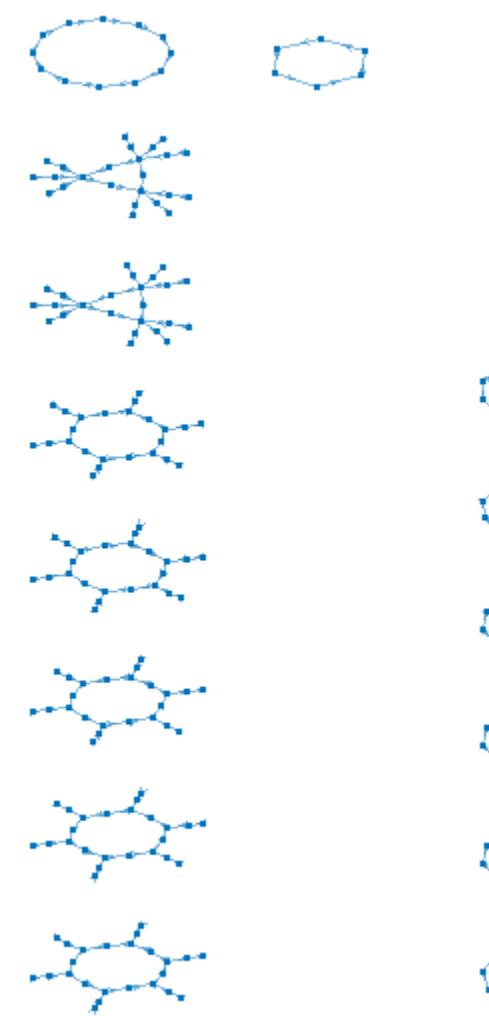
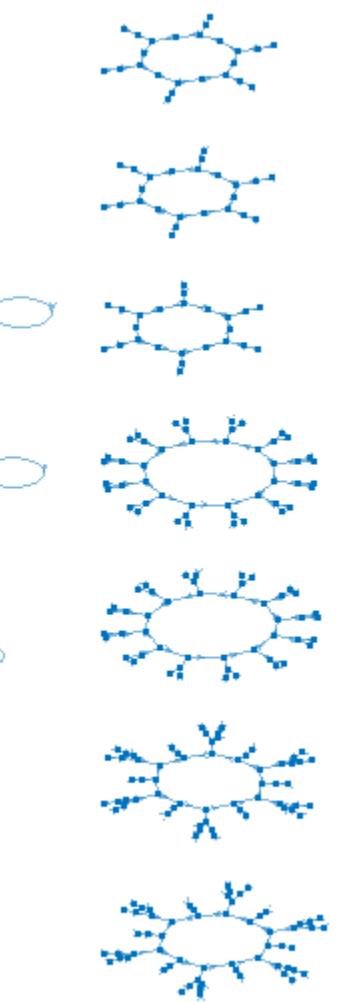
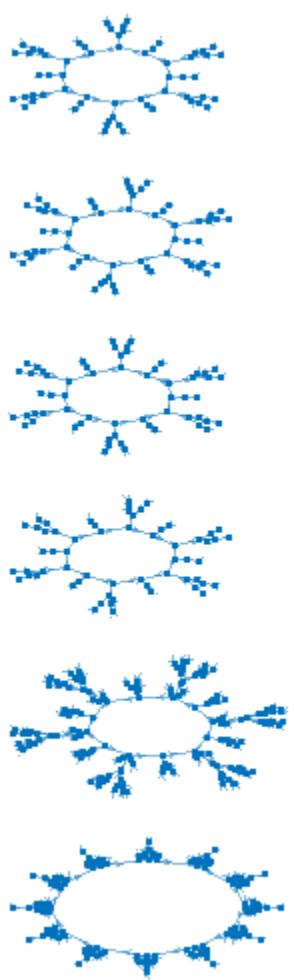
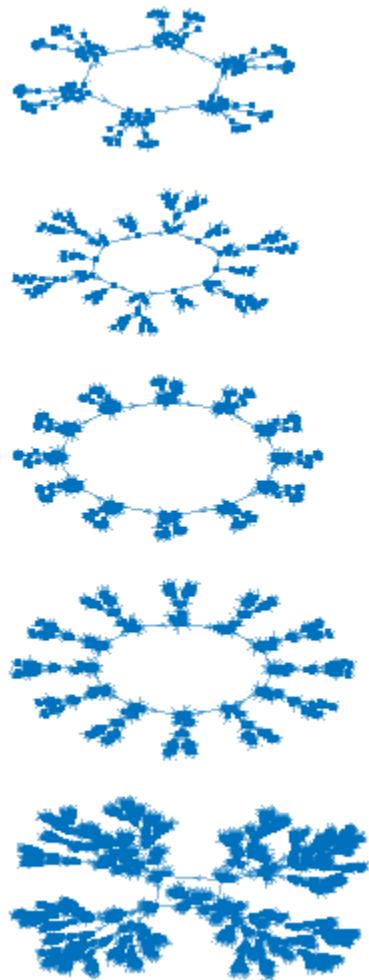


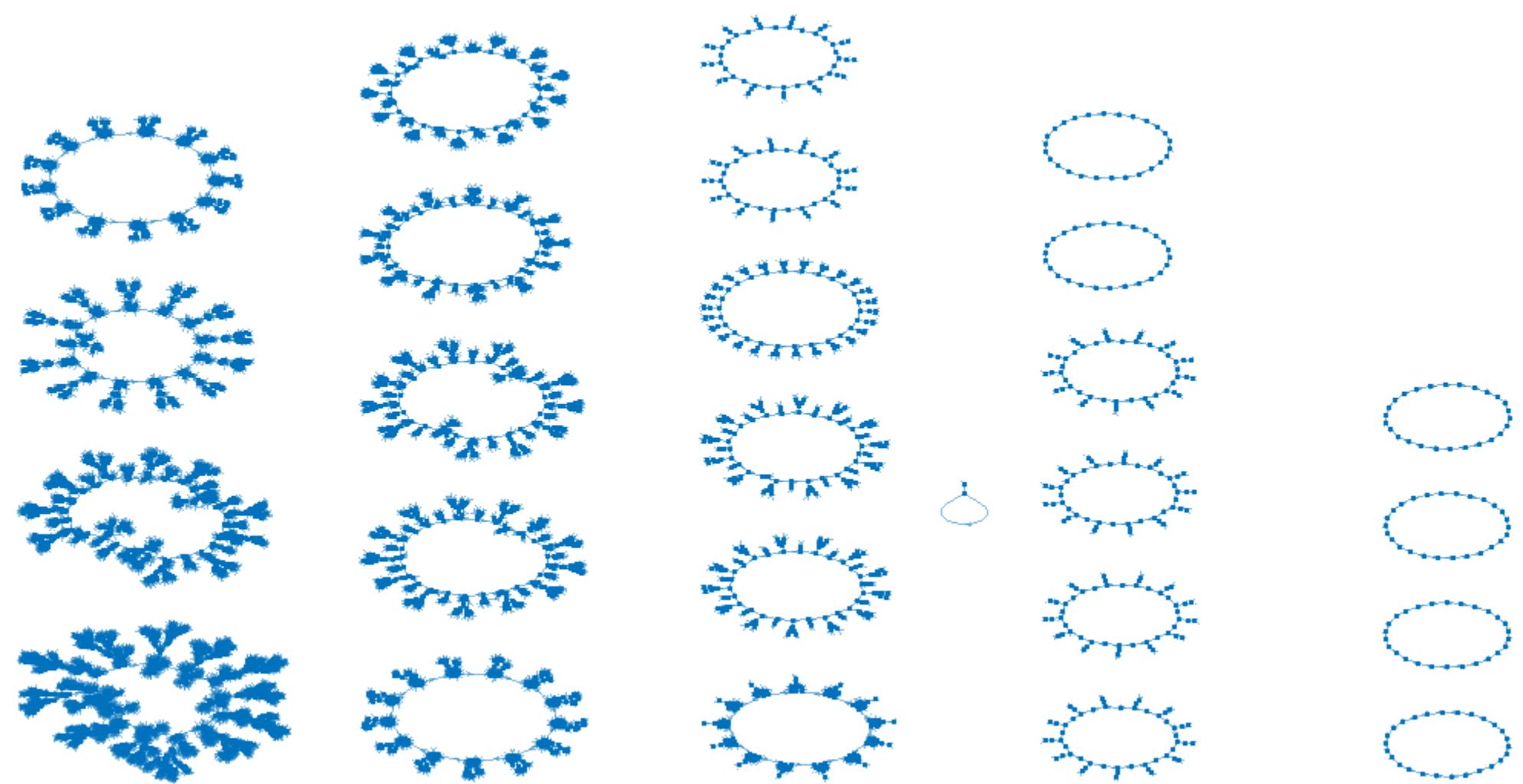


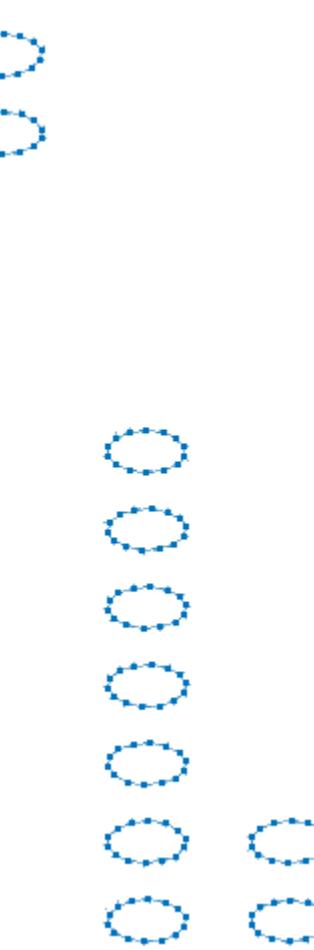
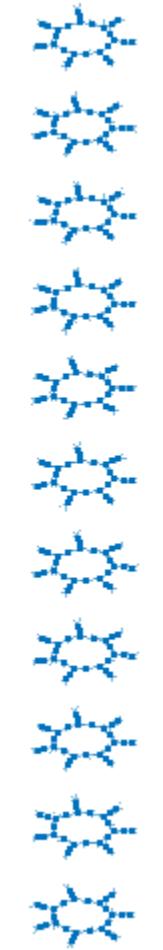
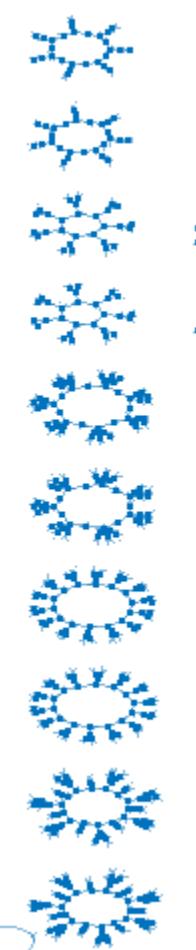
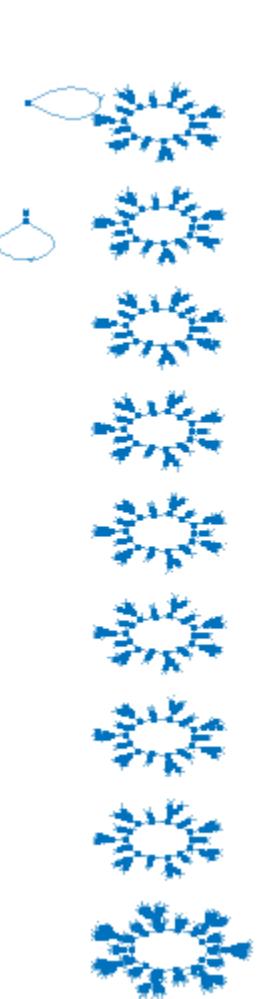
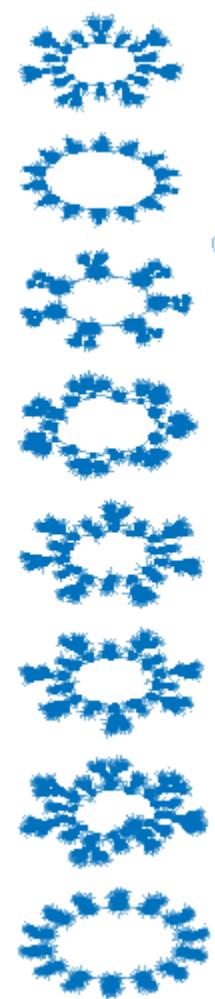
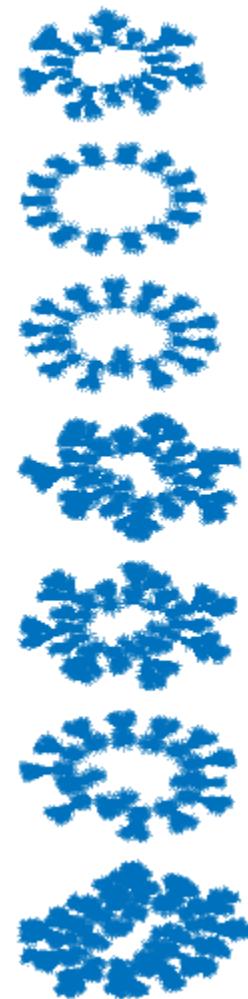


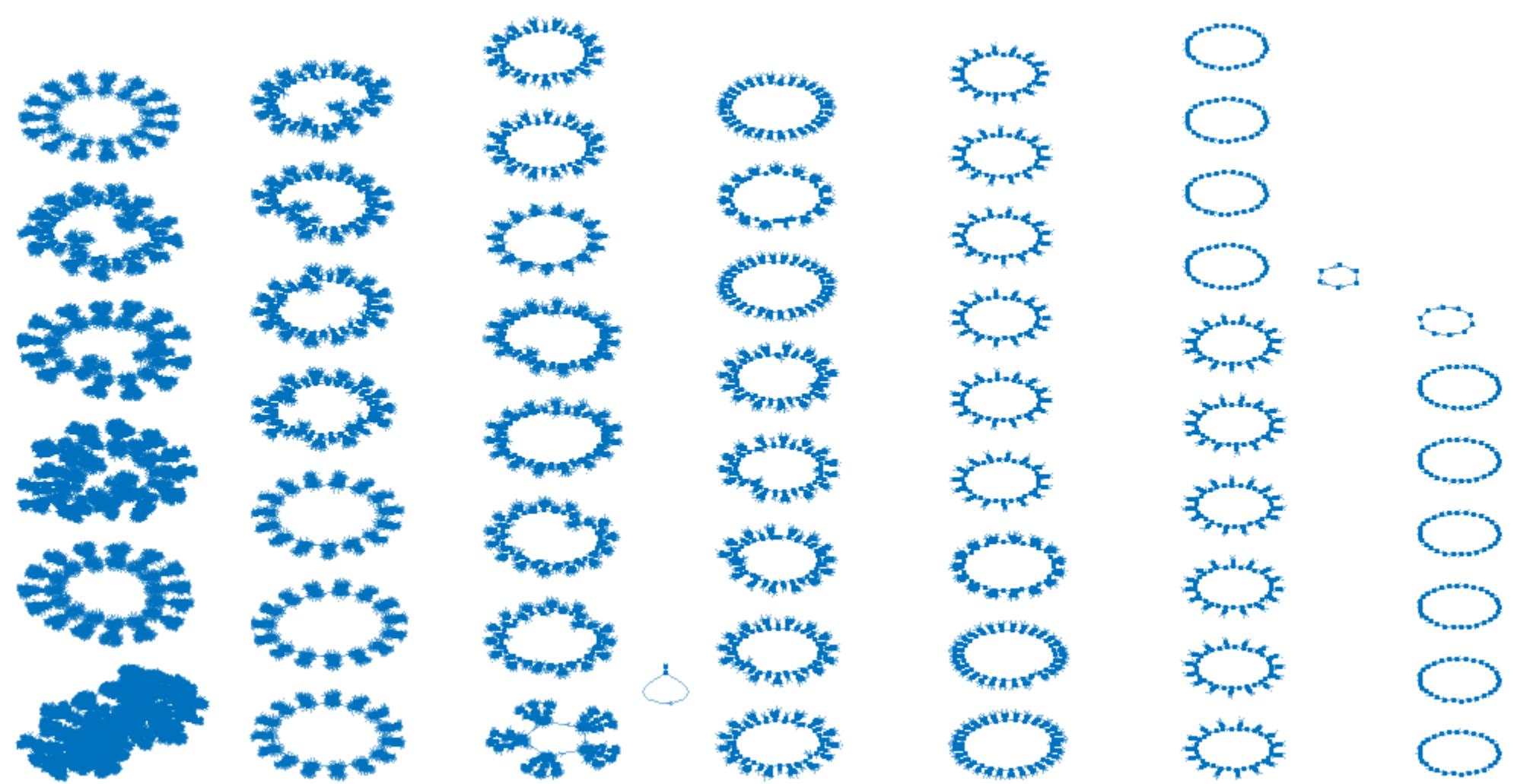




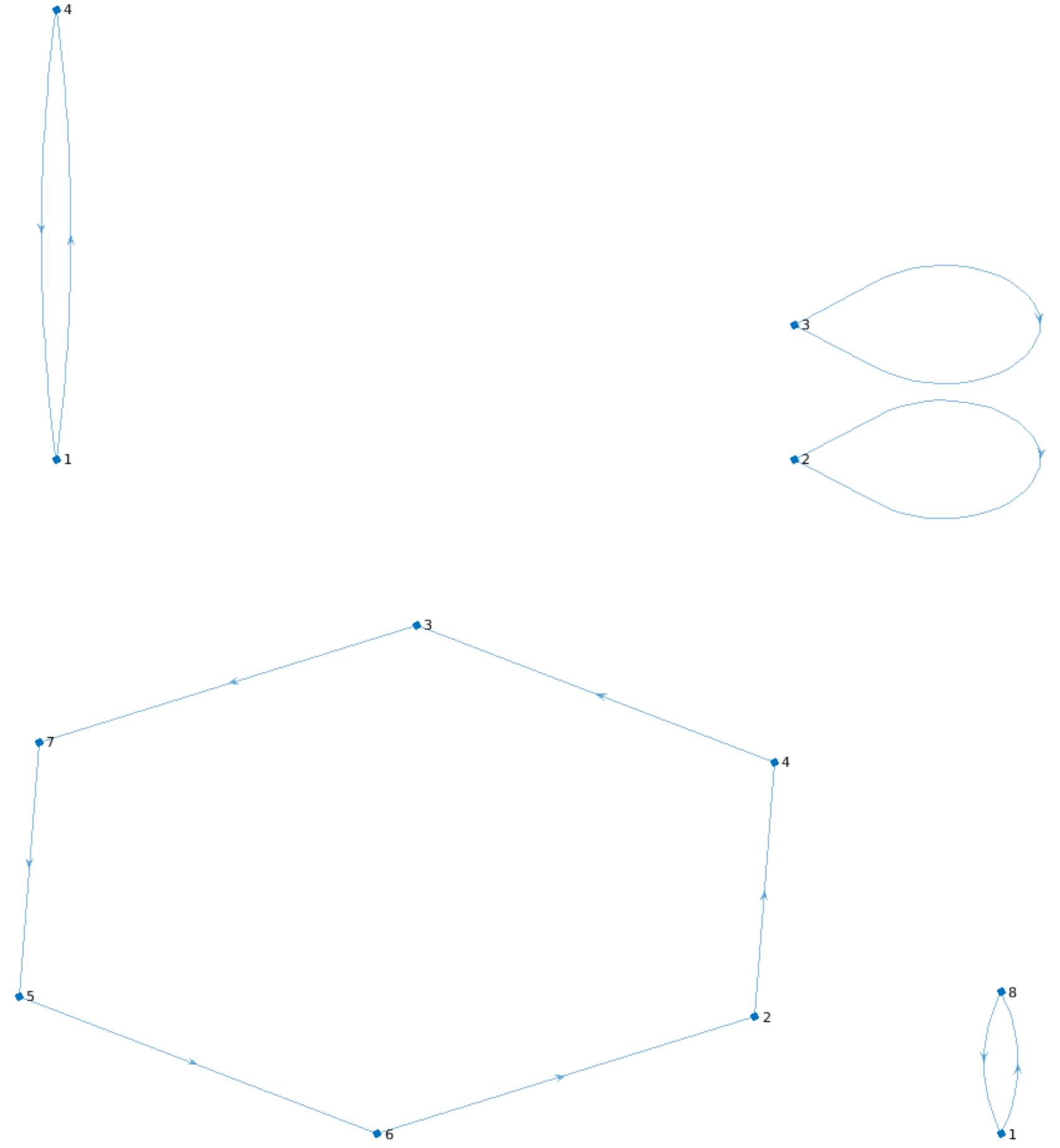


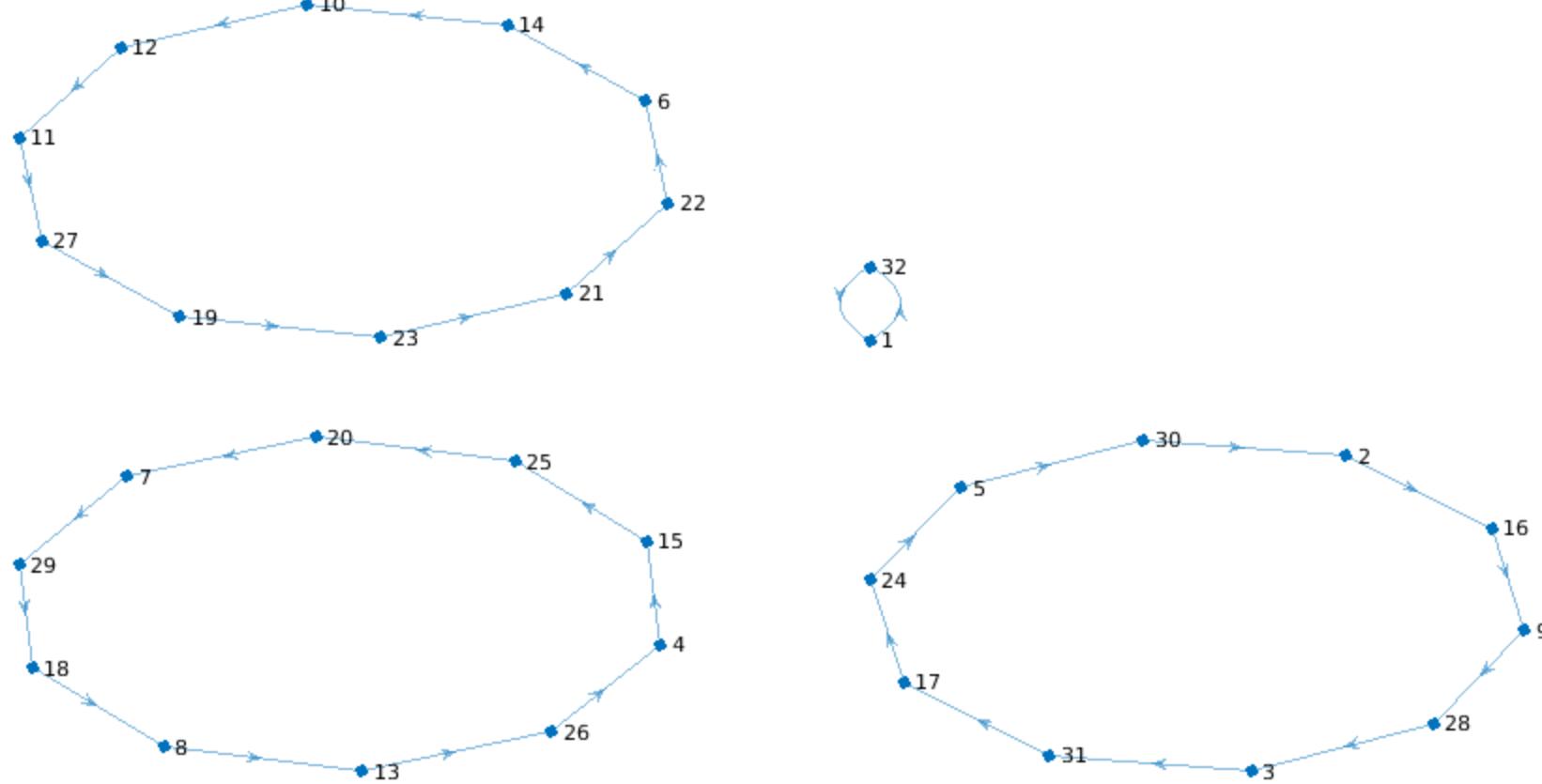
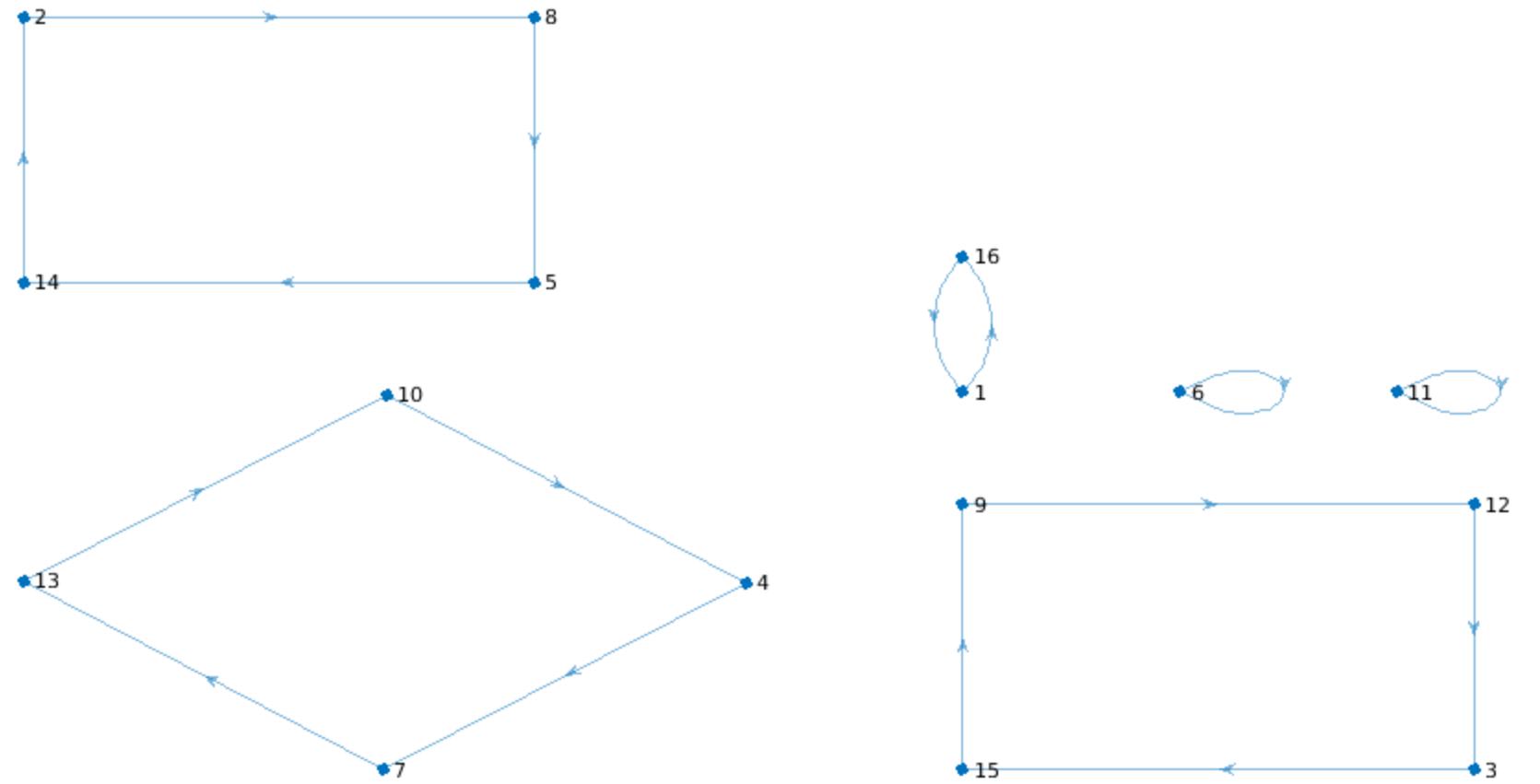


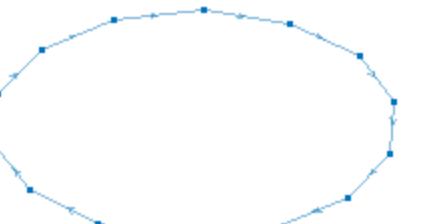
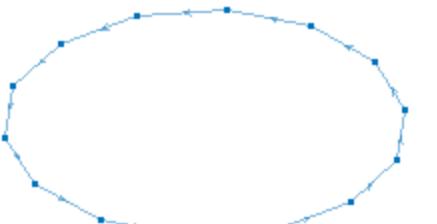
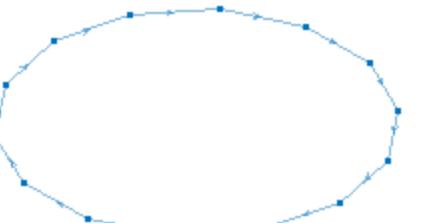
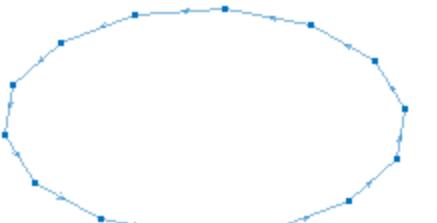
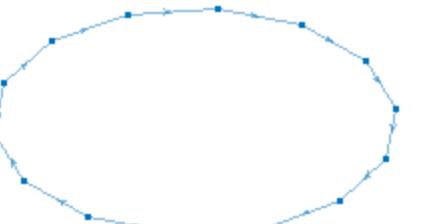
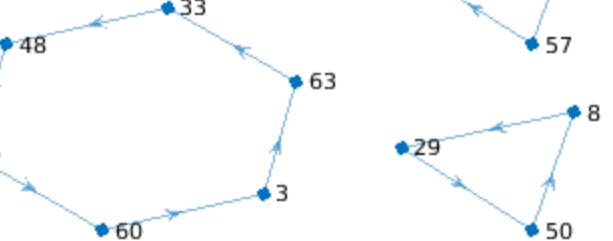
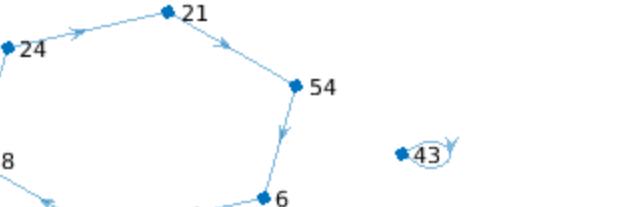
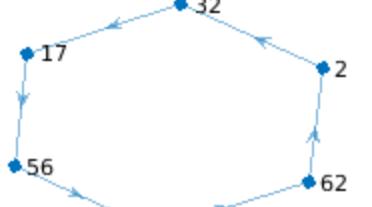
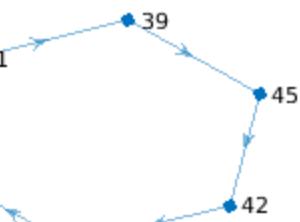
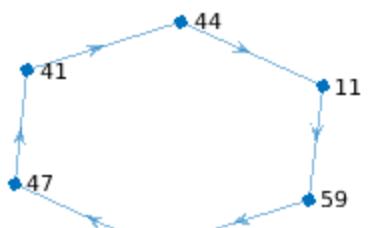


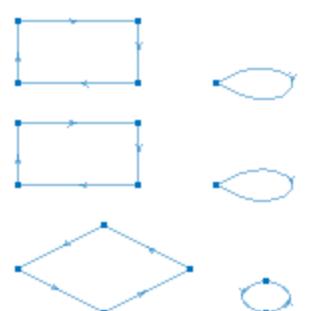
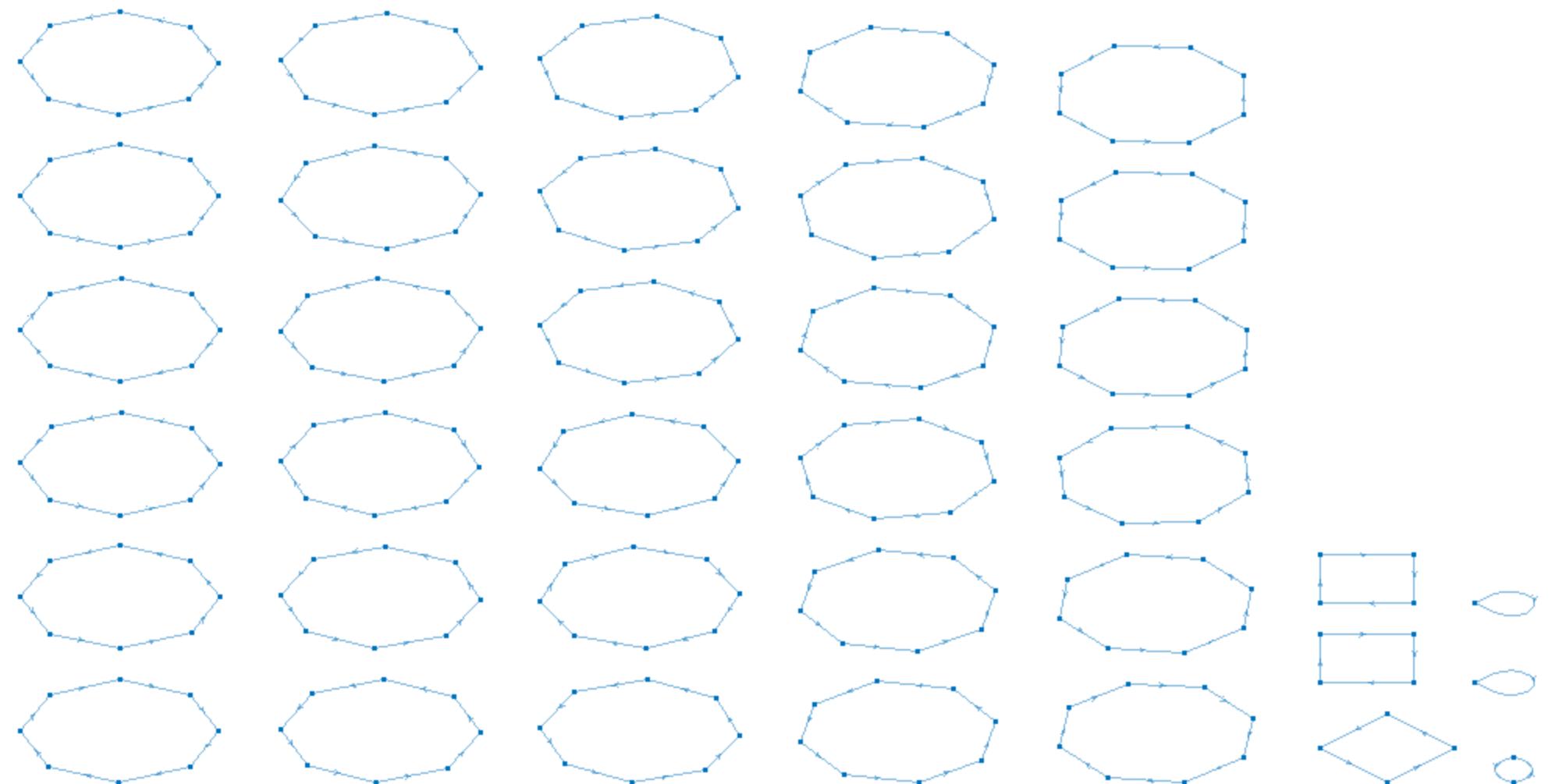


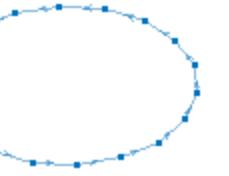
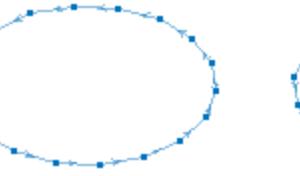
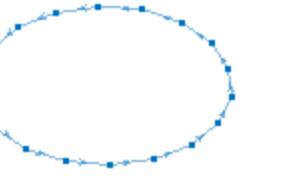
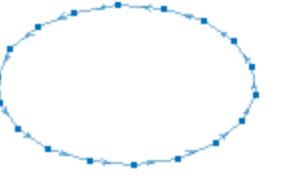
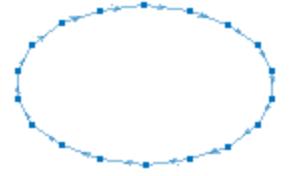
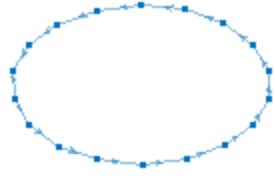
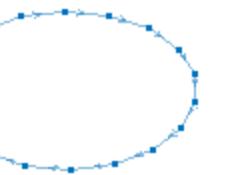
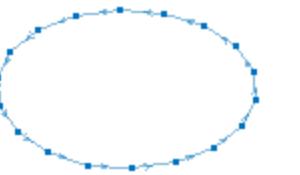
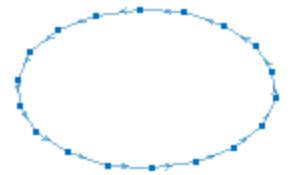
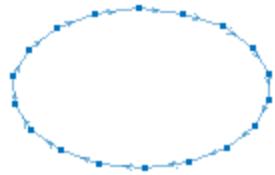
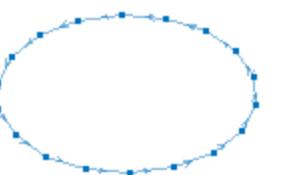
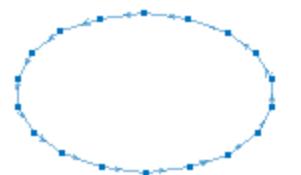
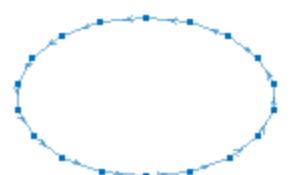
15

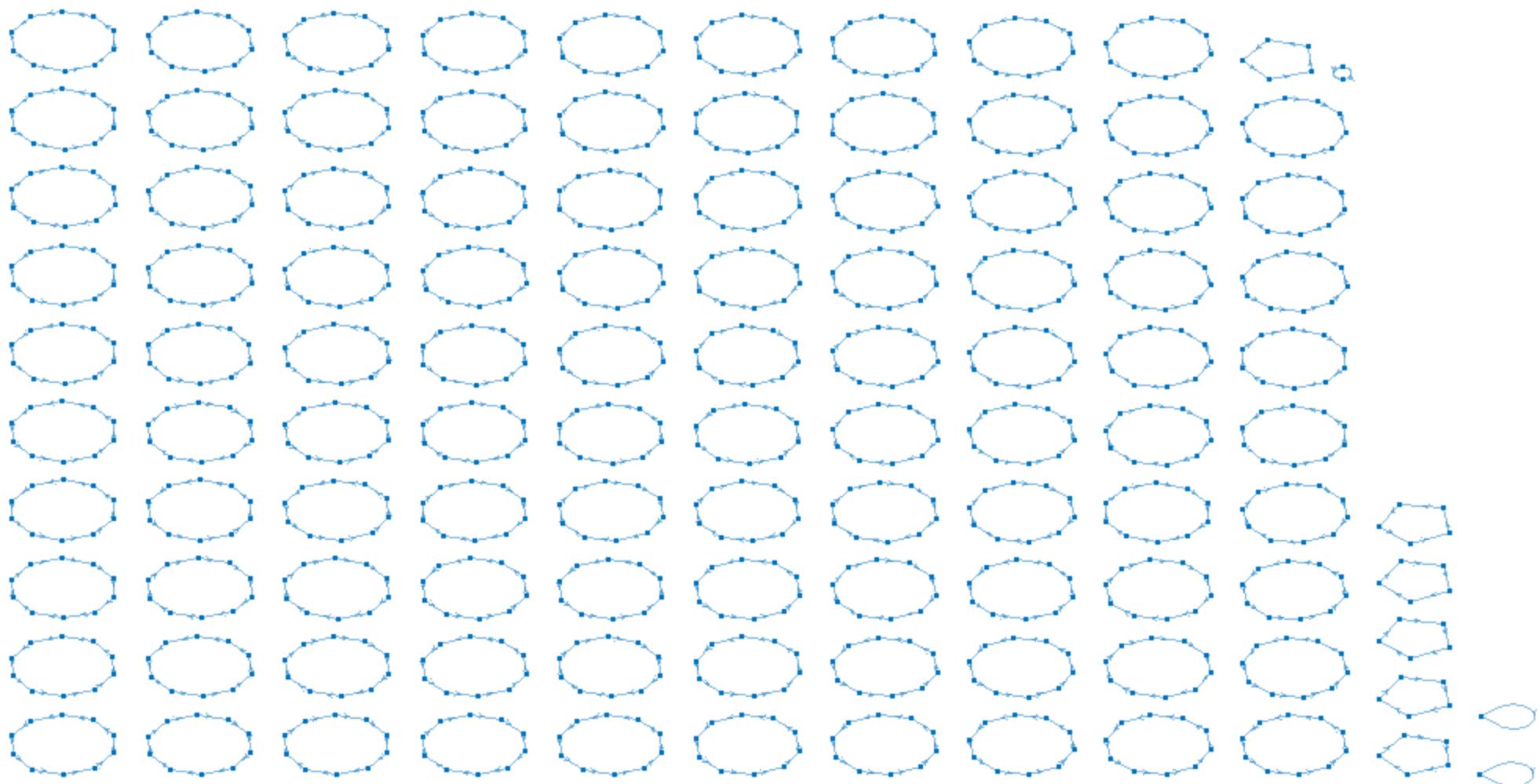


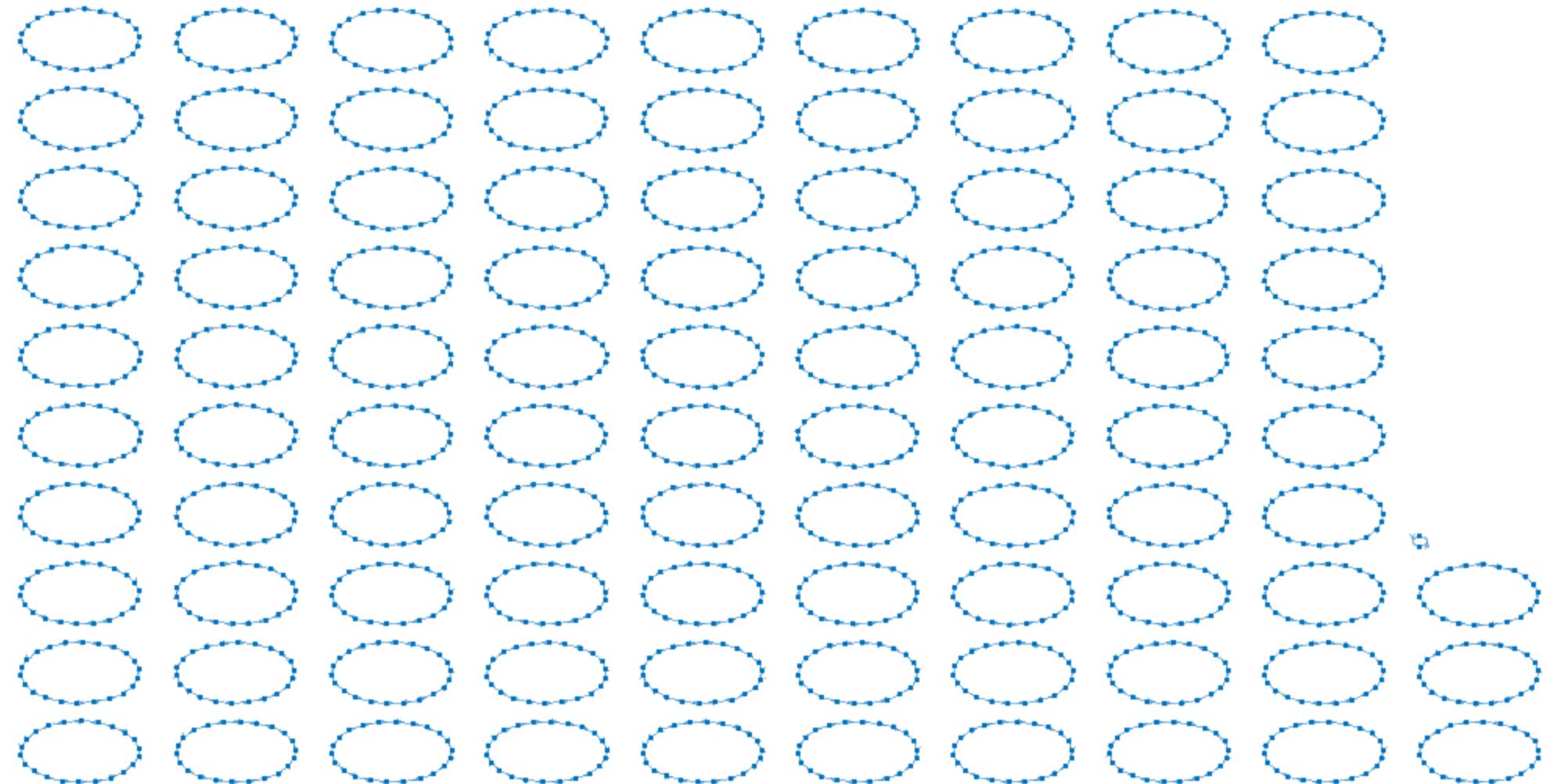


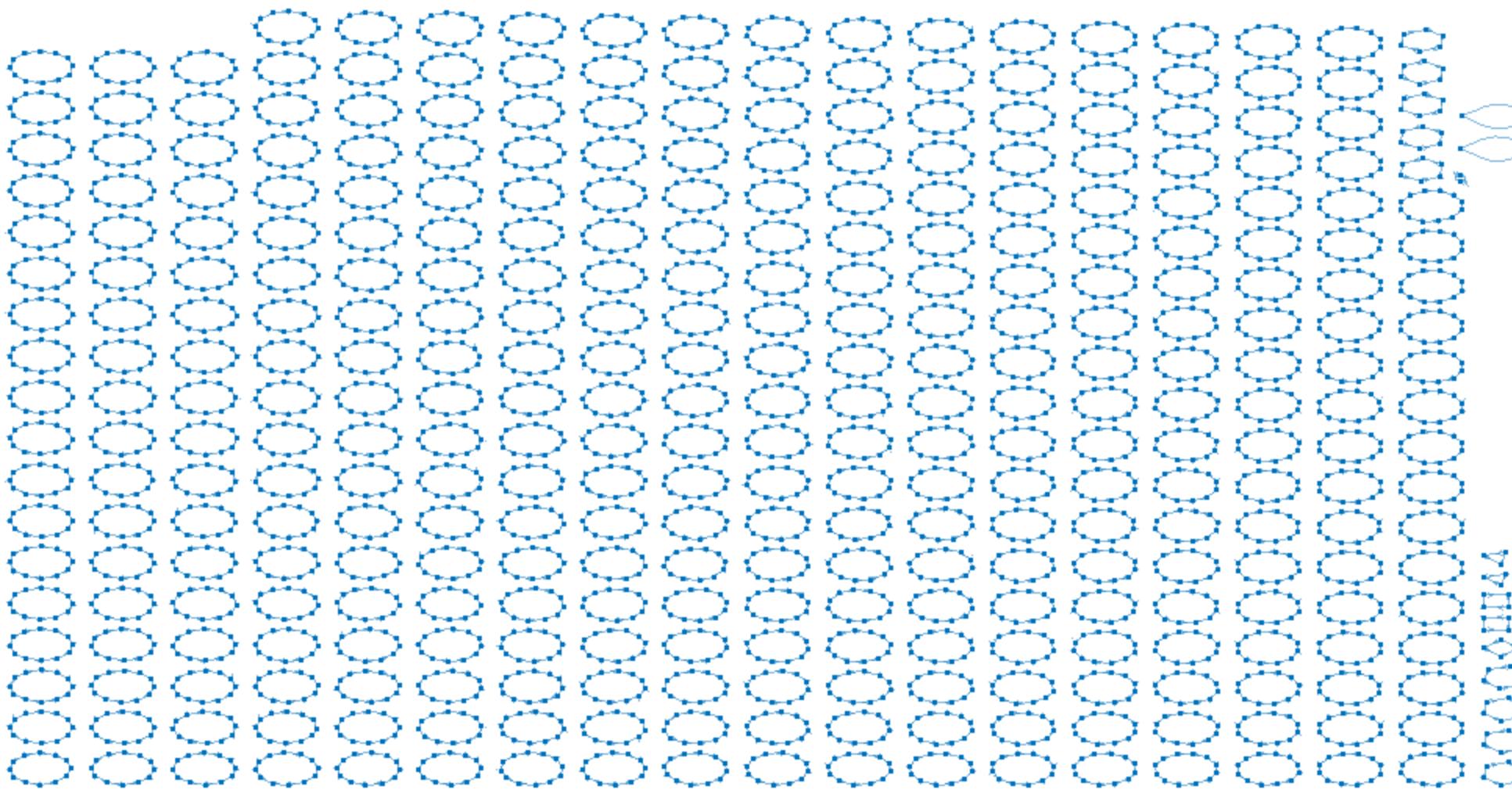




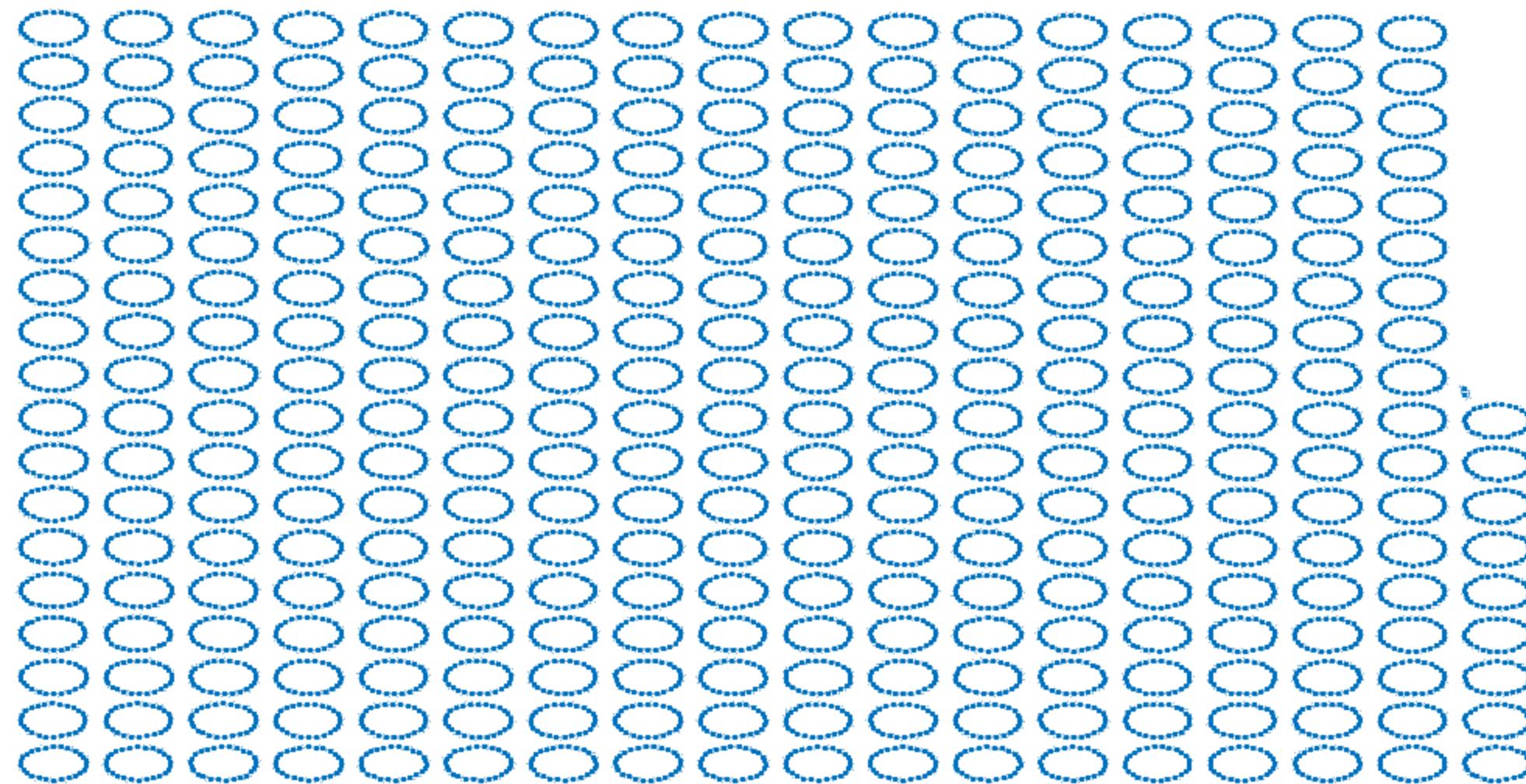


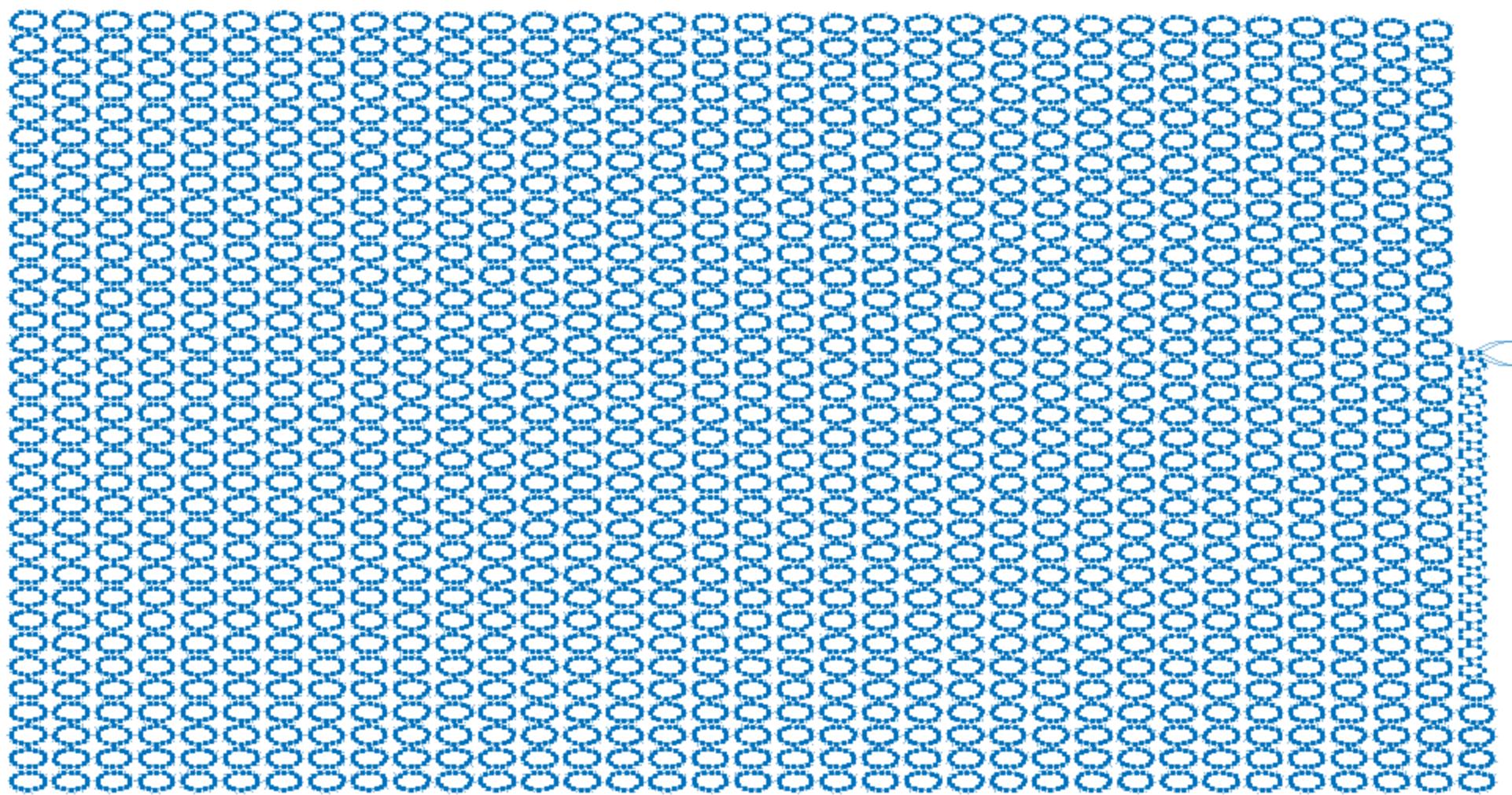


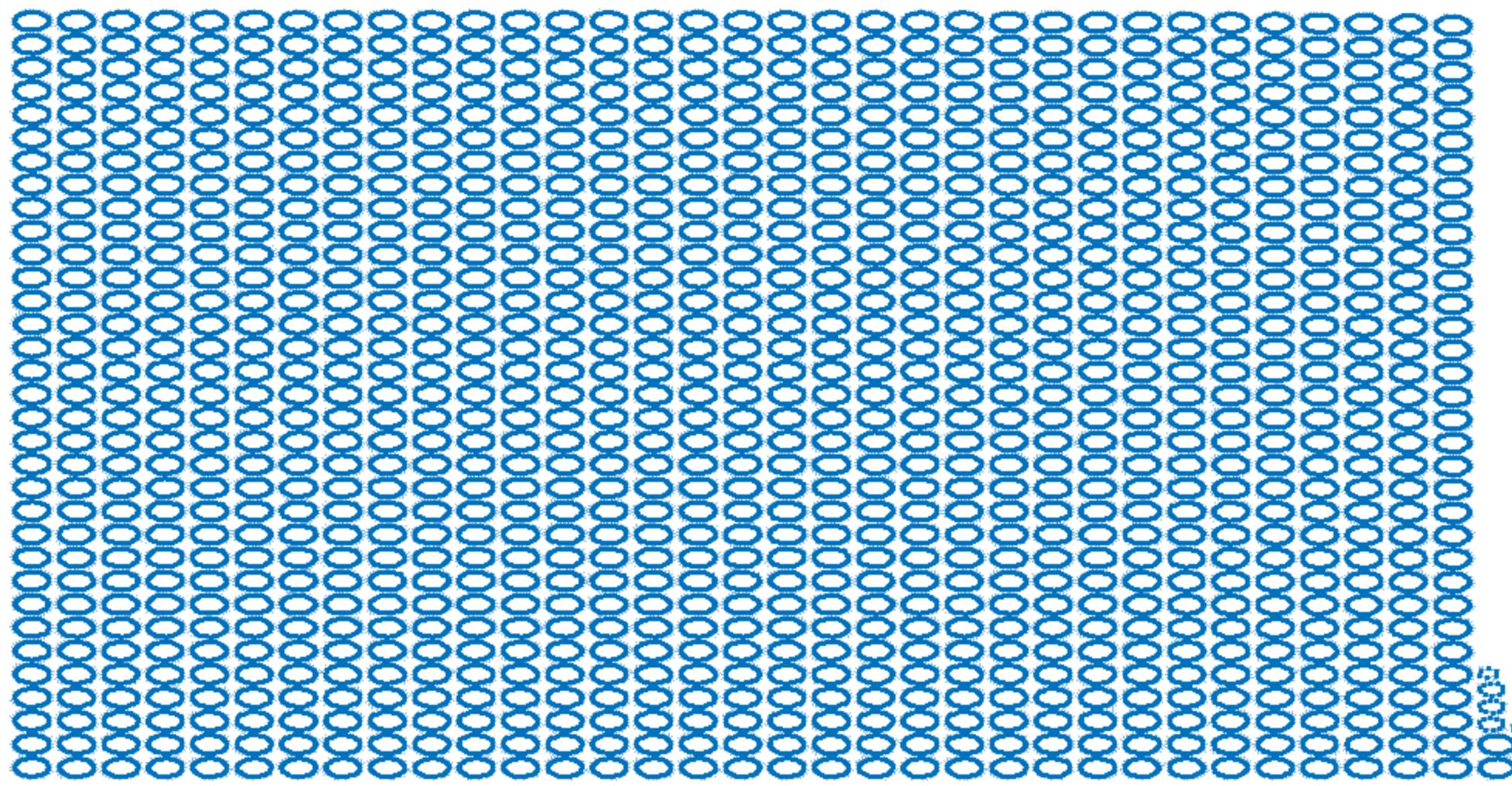




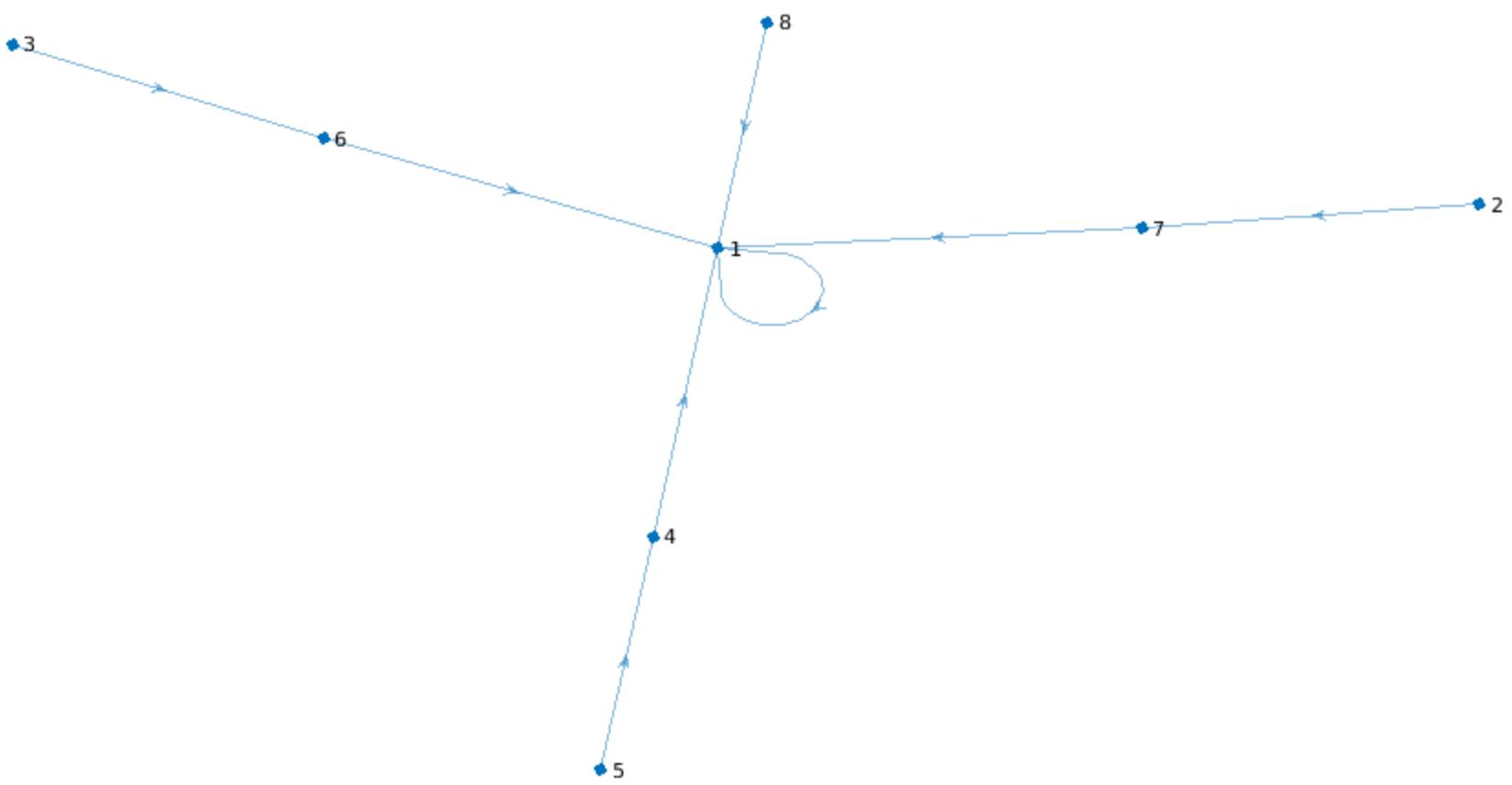
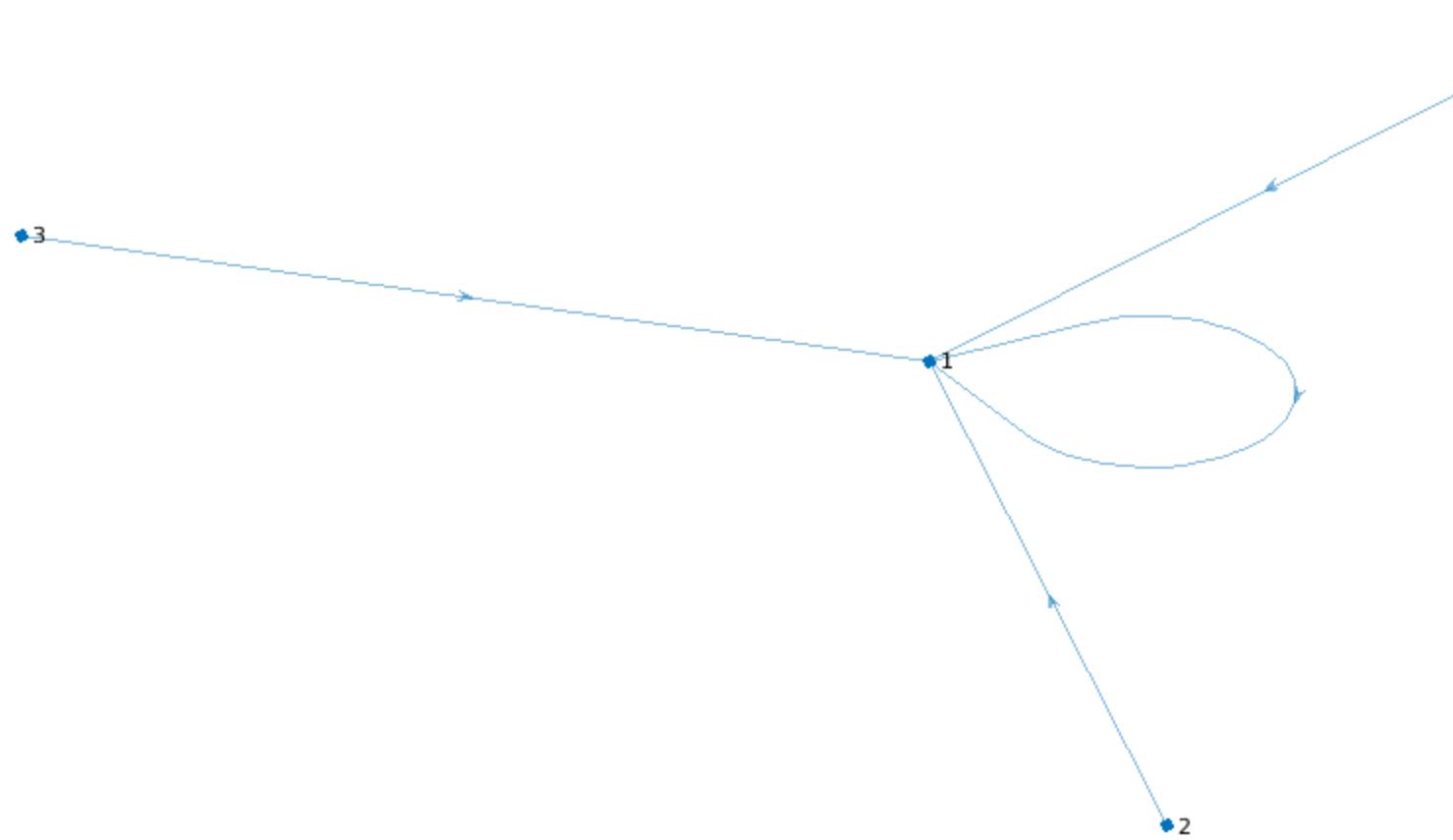
III44

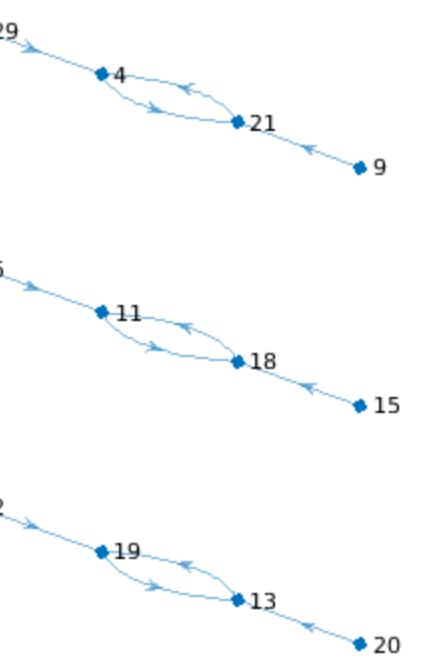
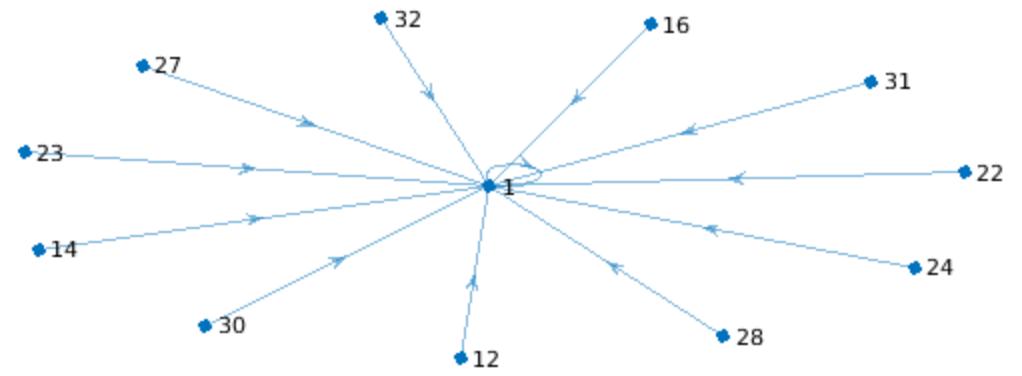
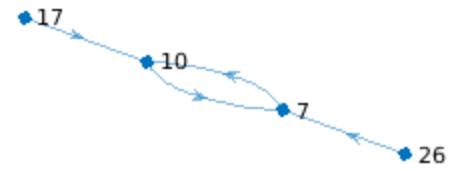
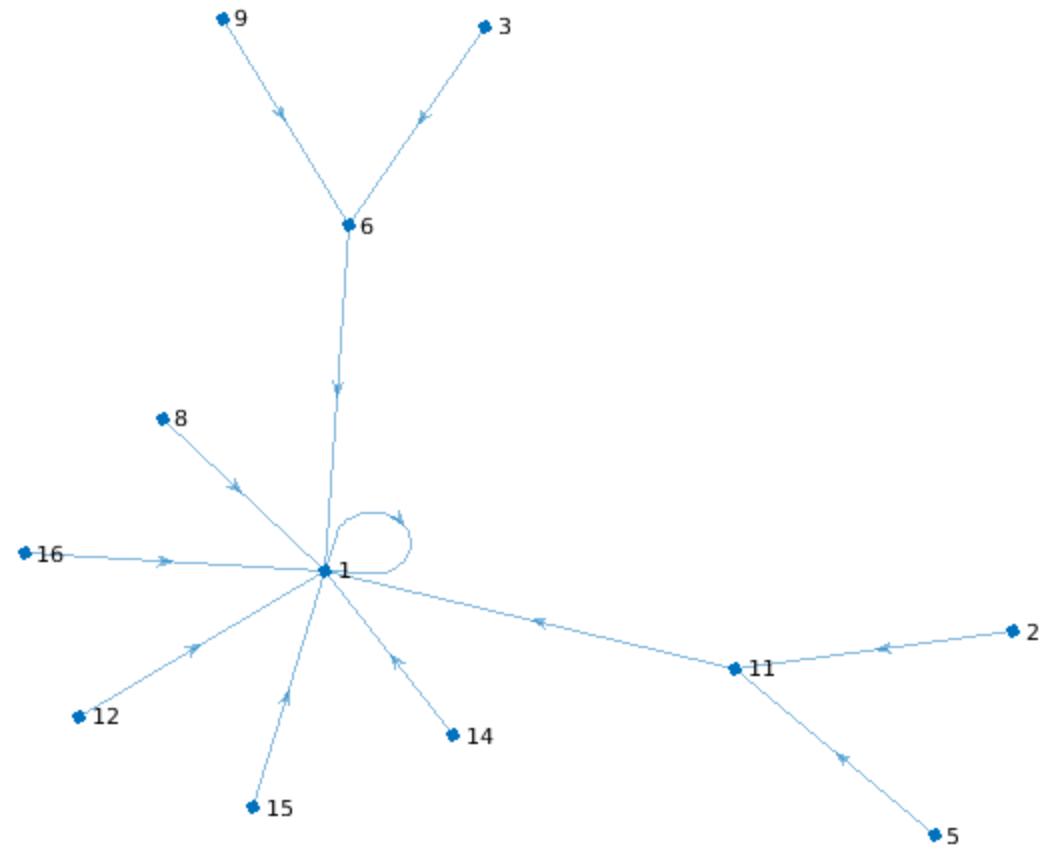


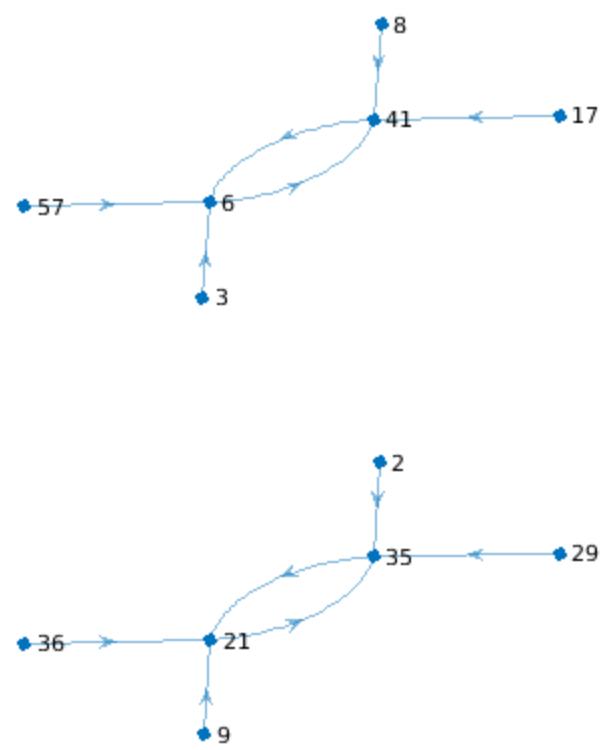
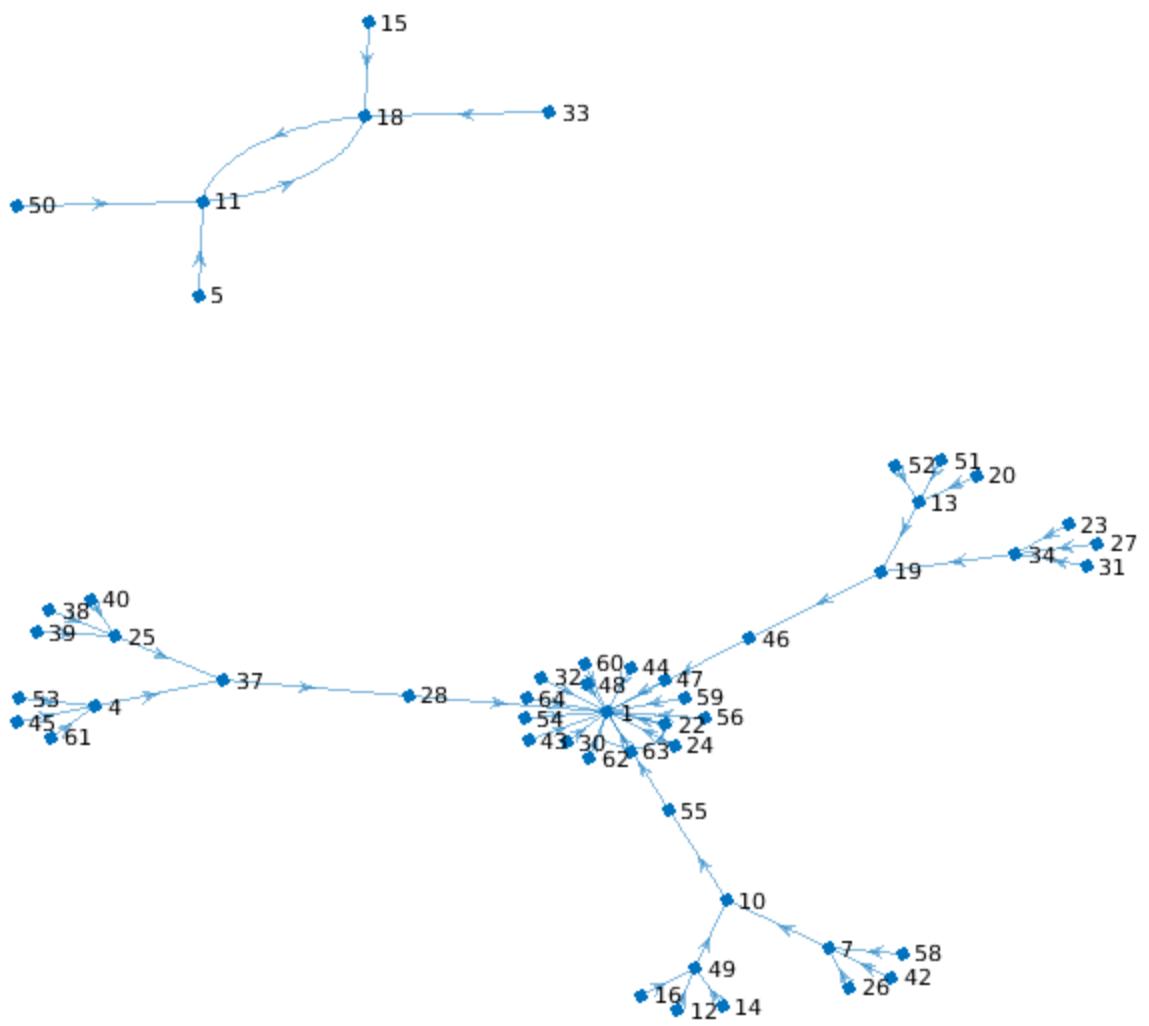


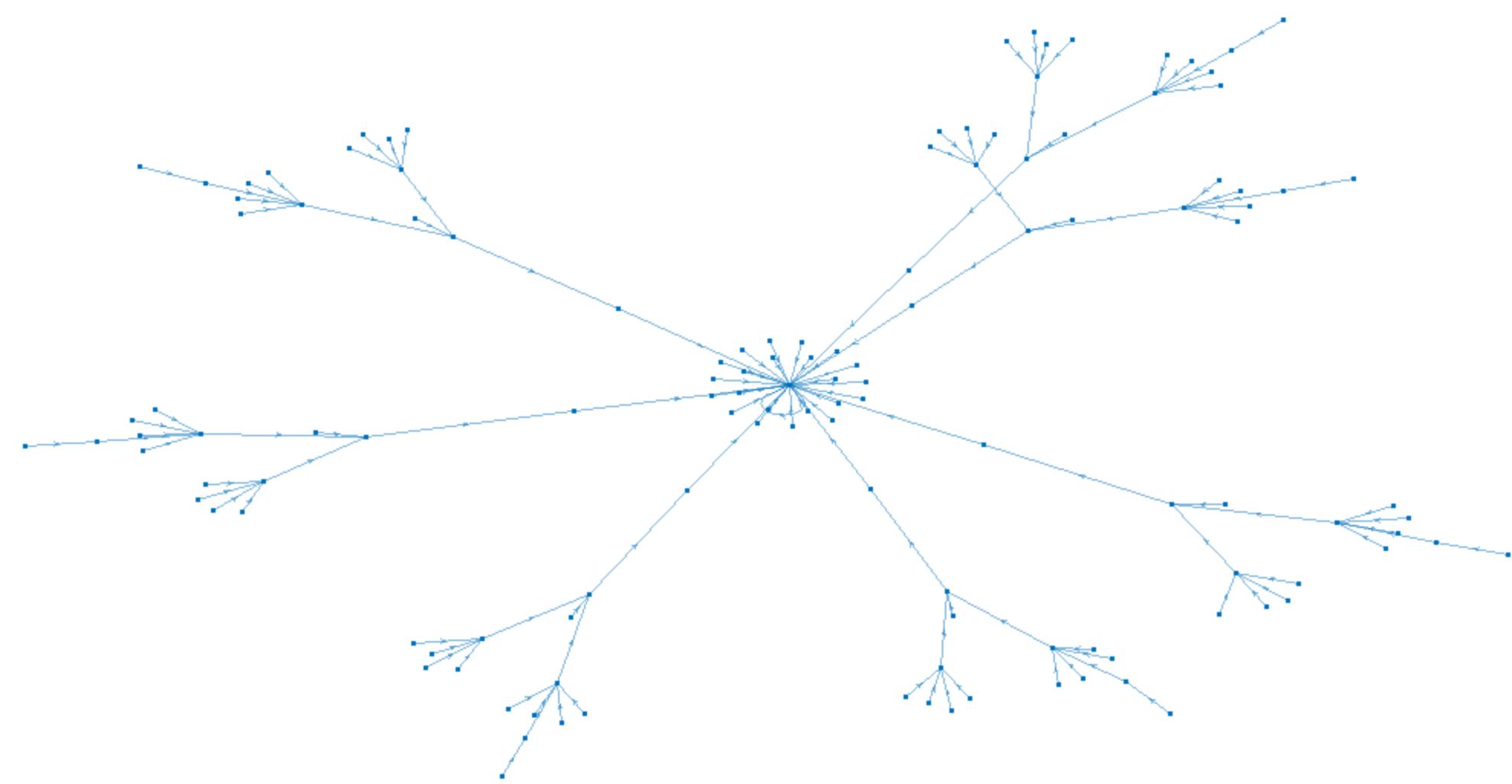


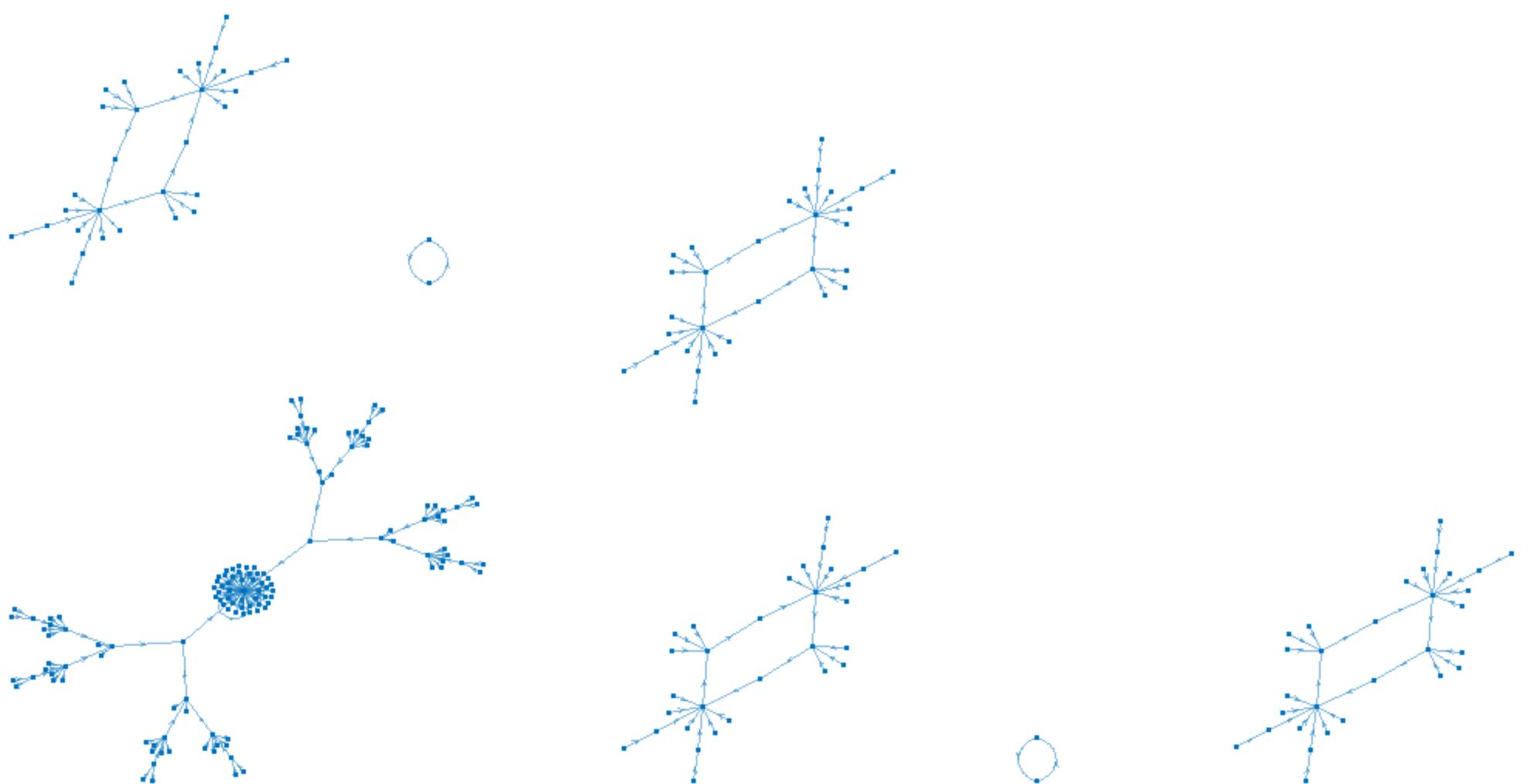
18

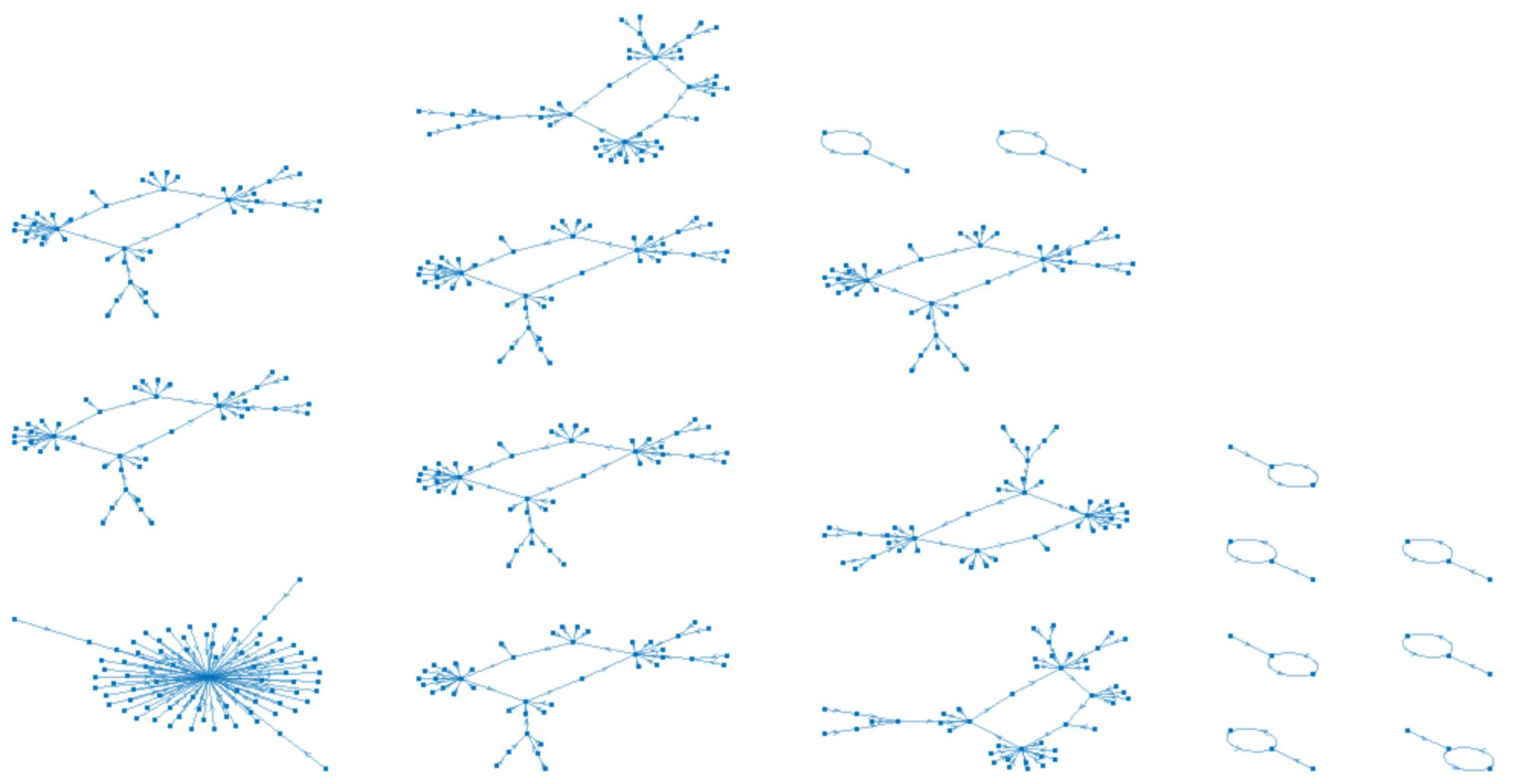


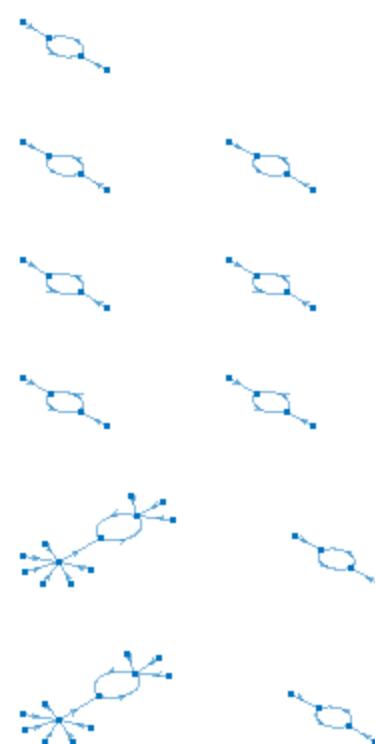
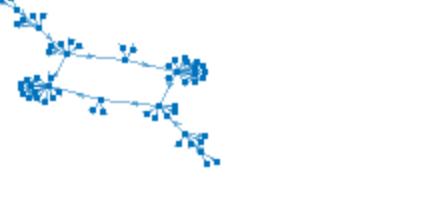
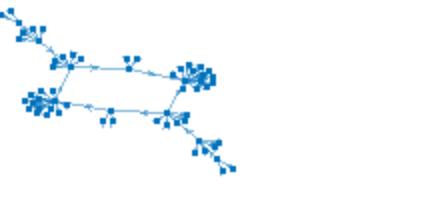
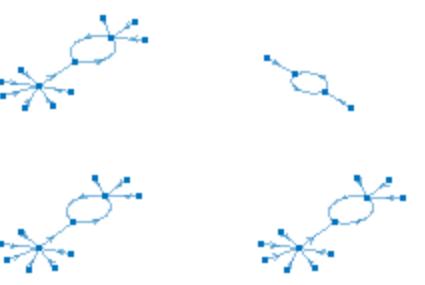
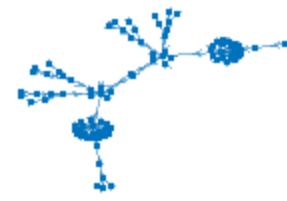
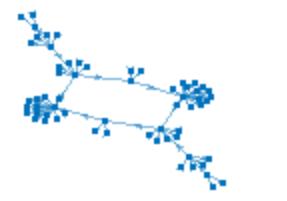
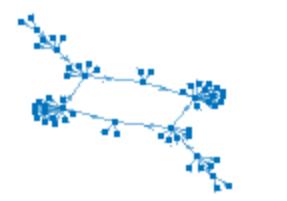
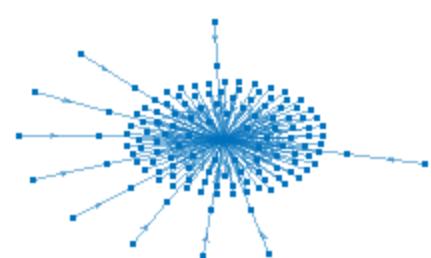
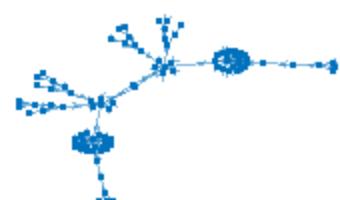
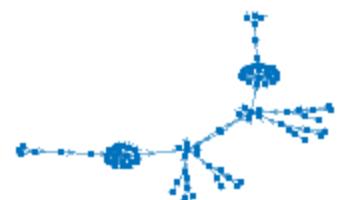
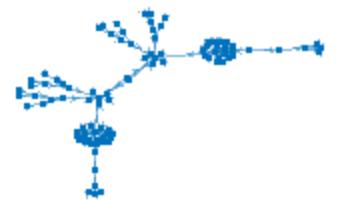


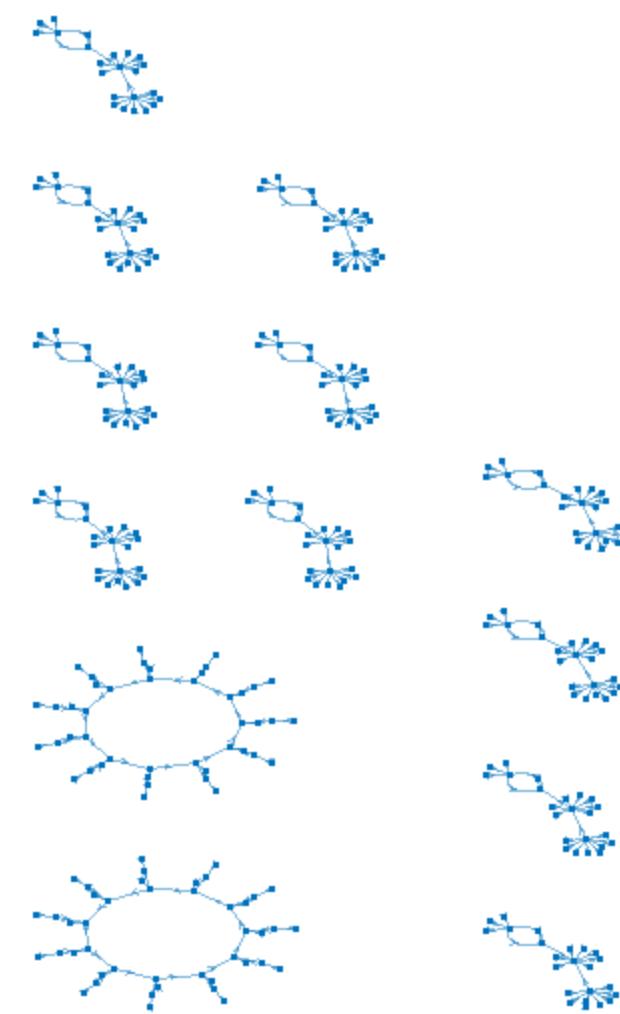
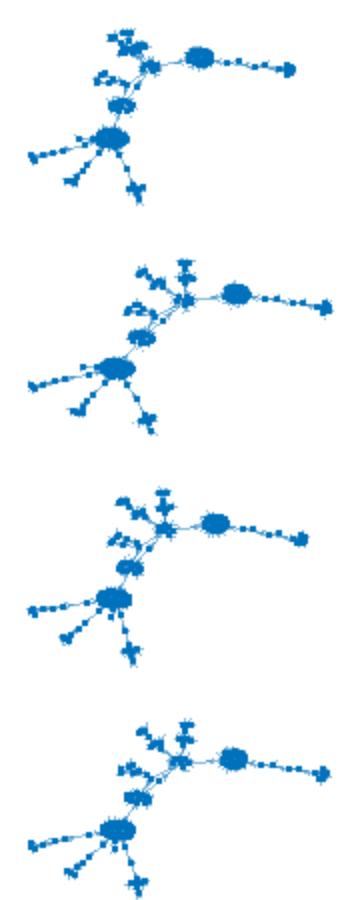
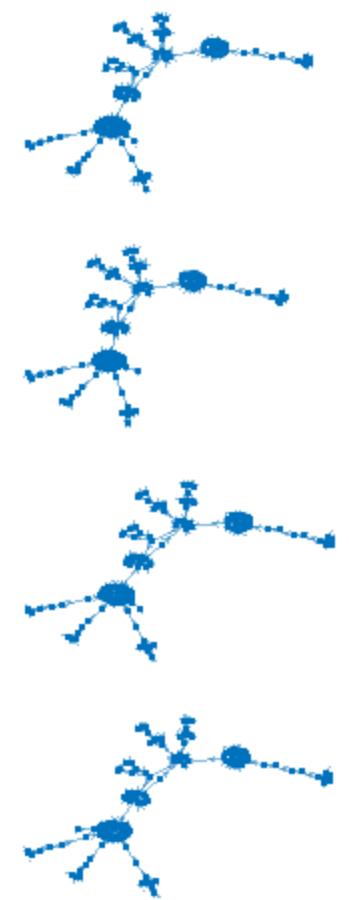
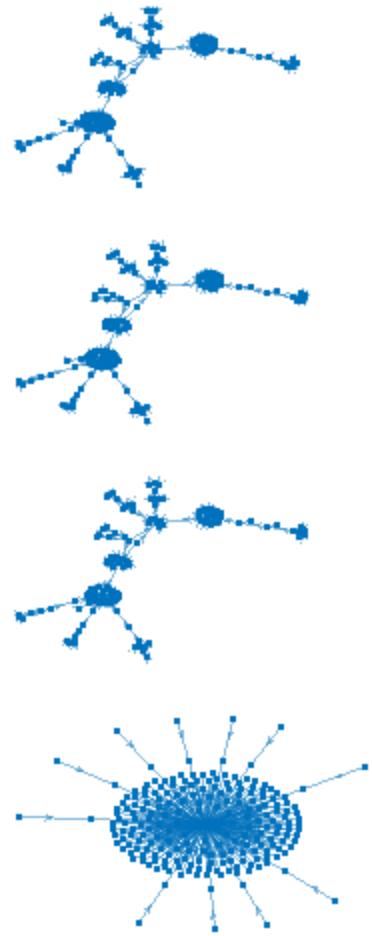


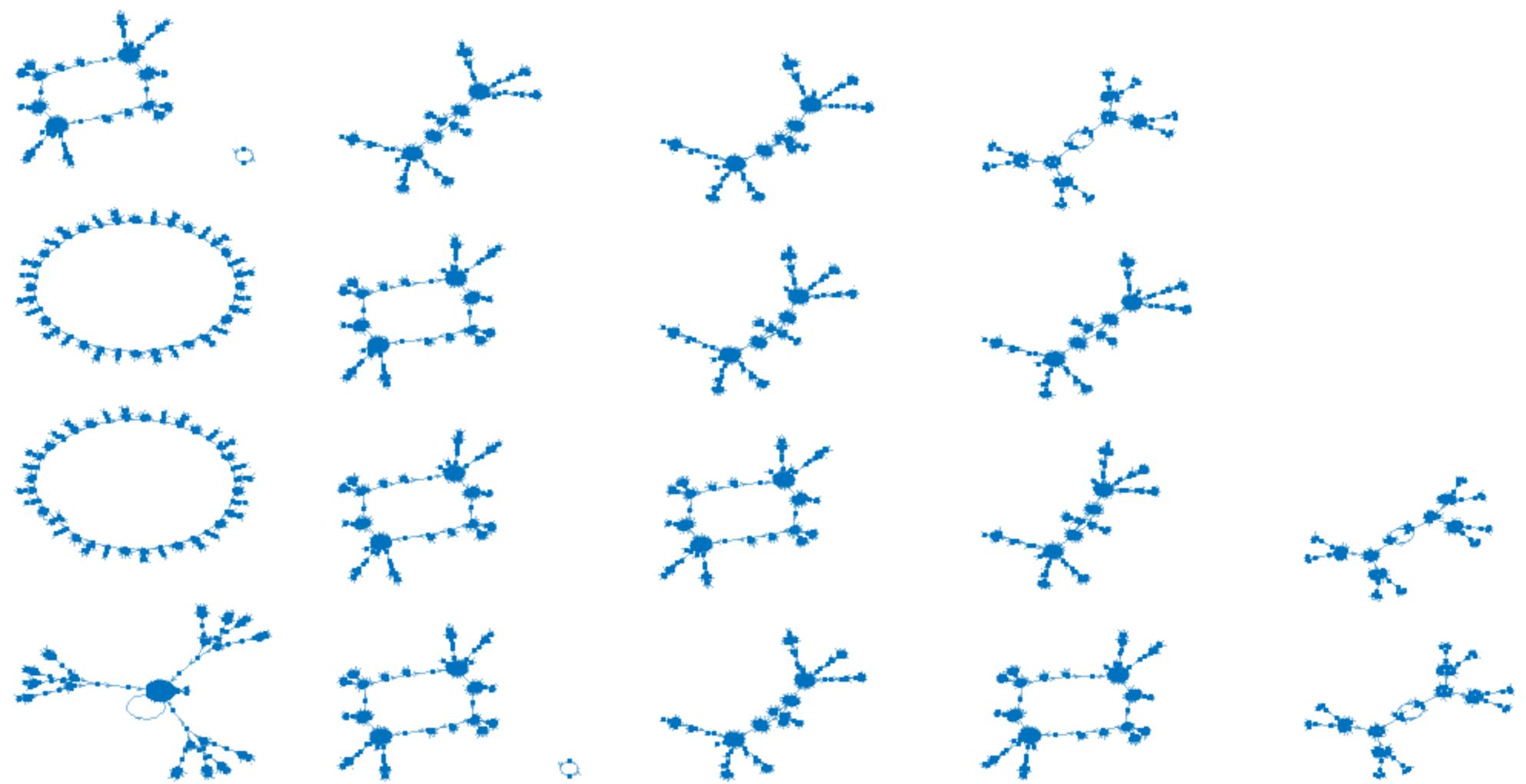


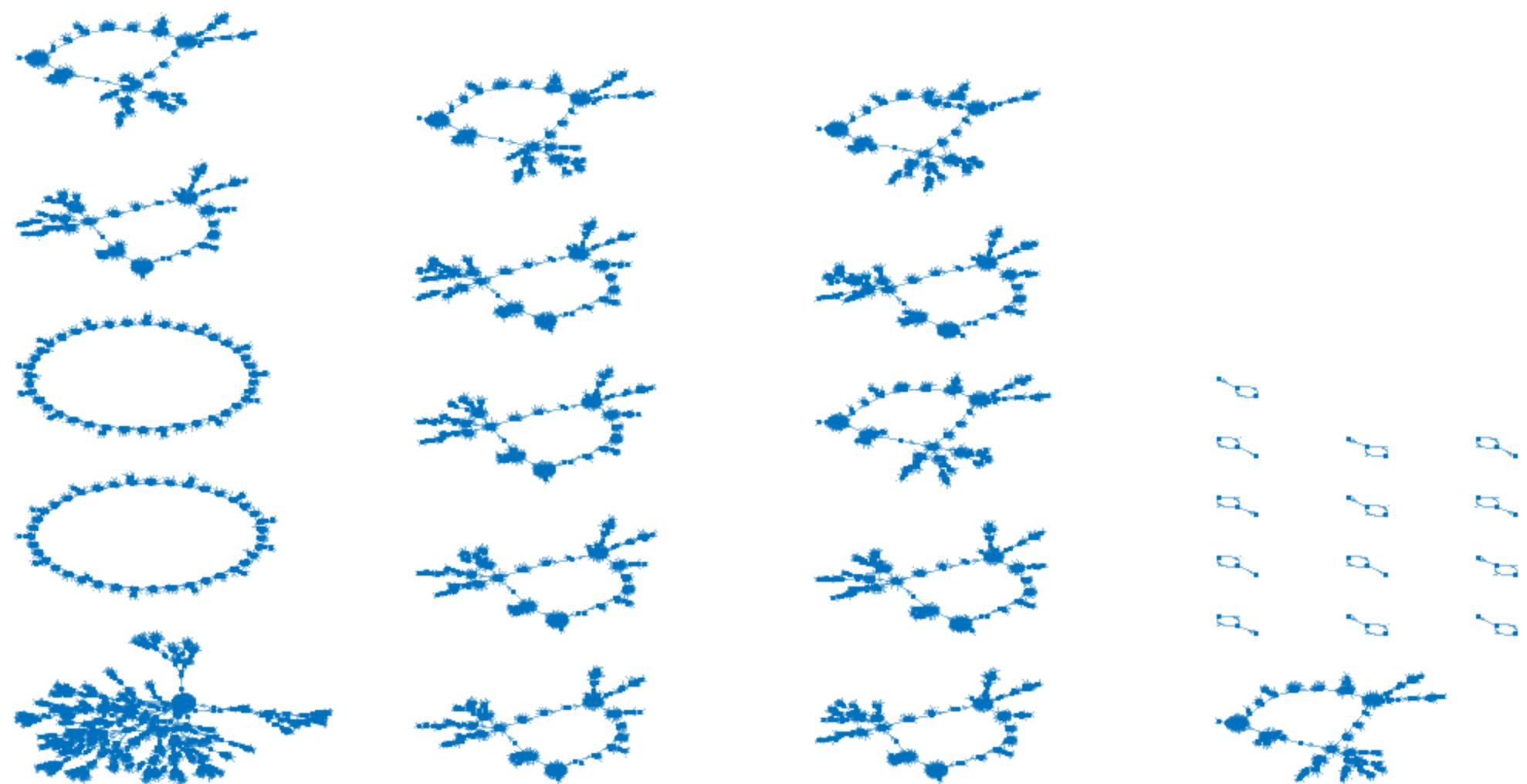




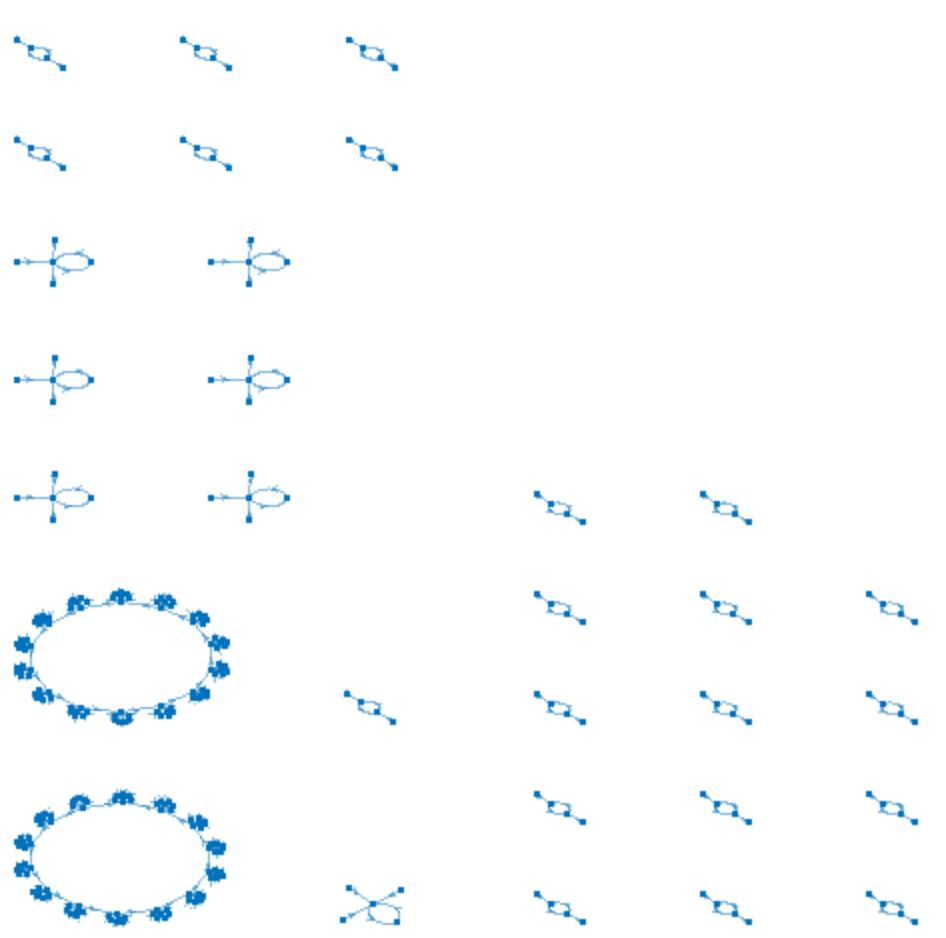
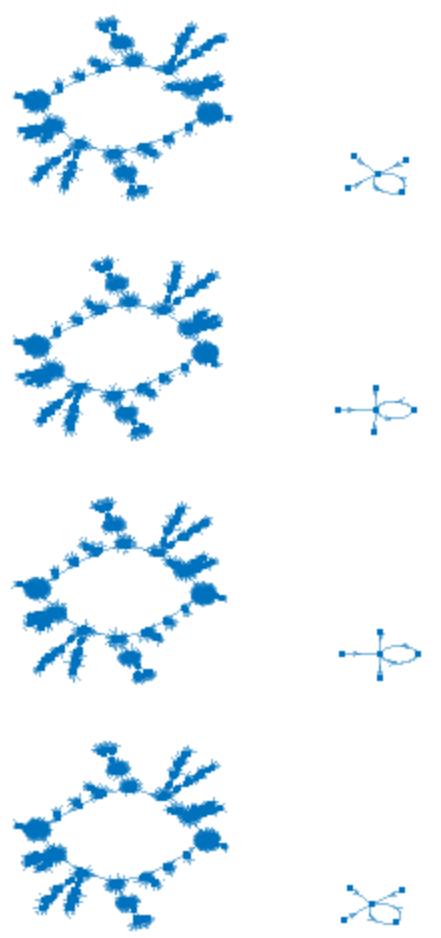
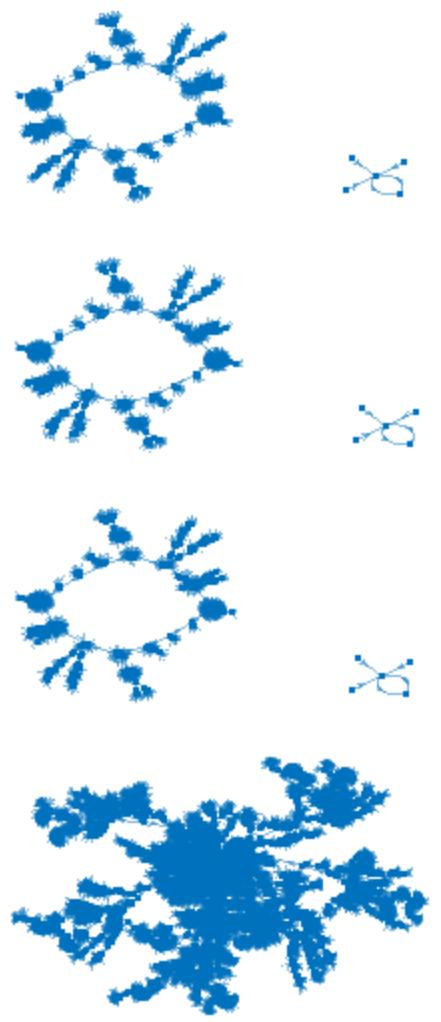


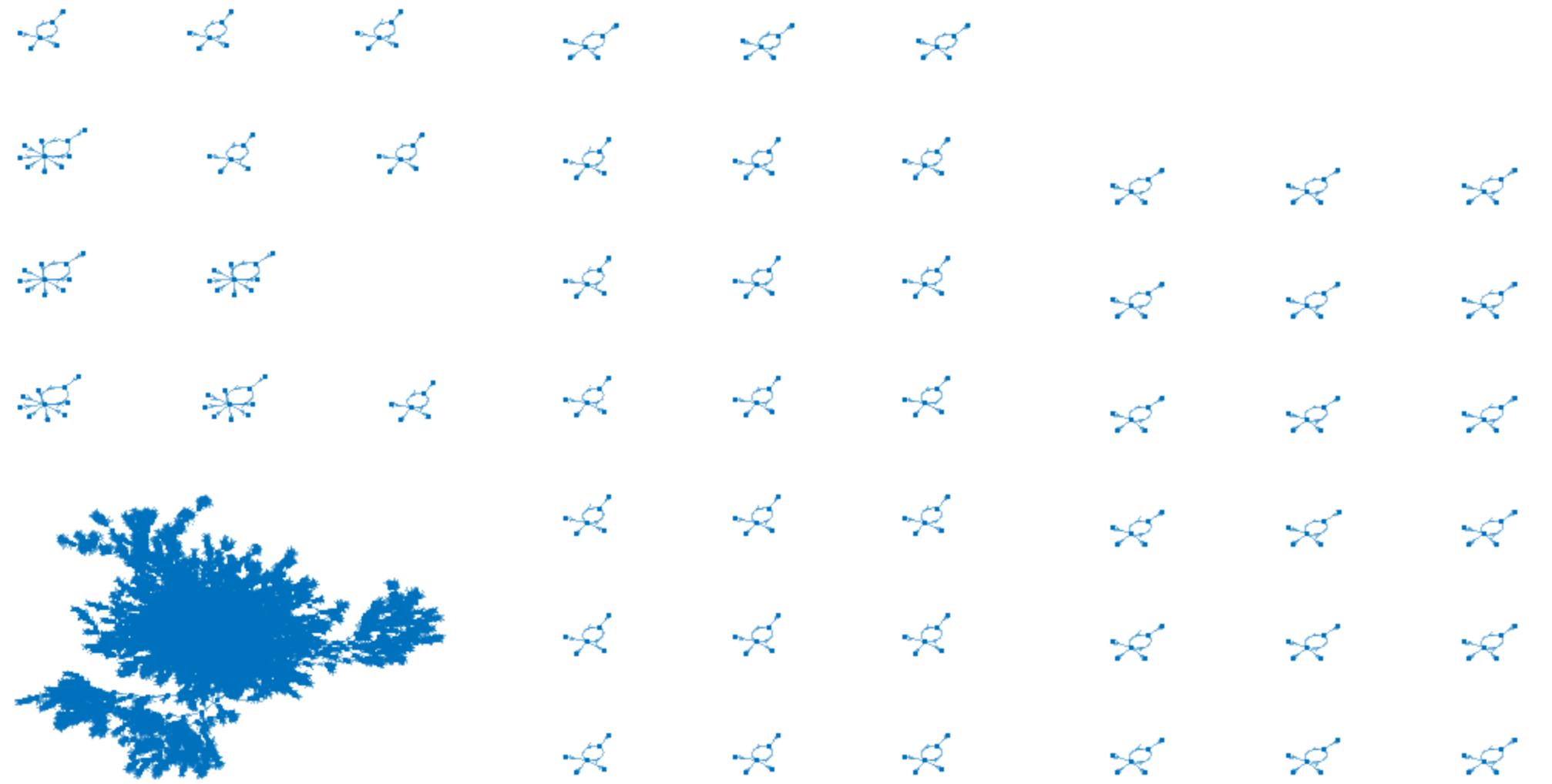


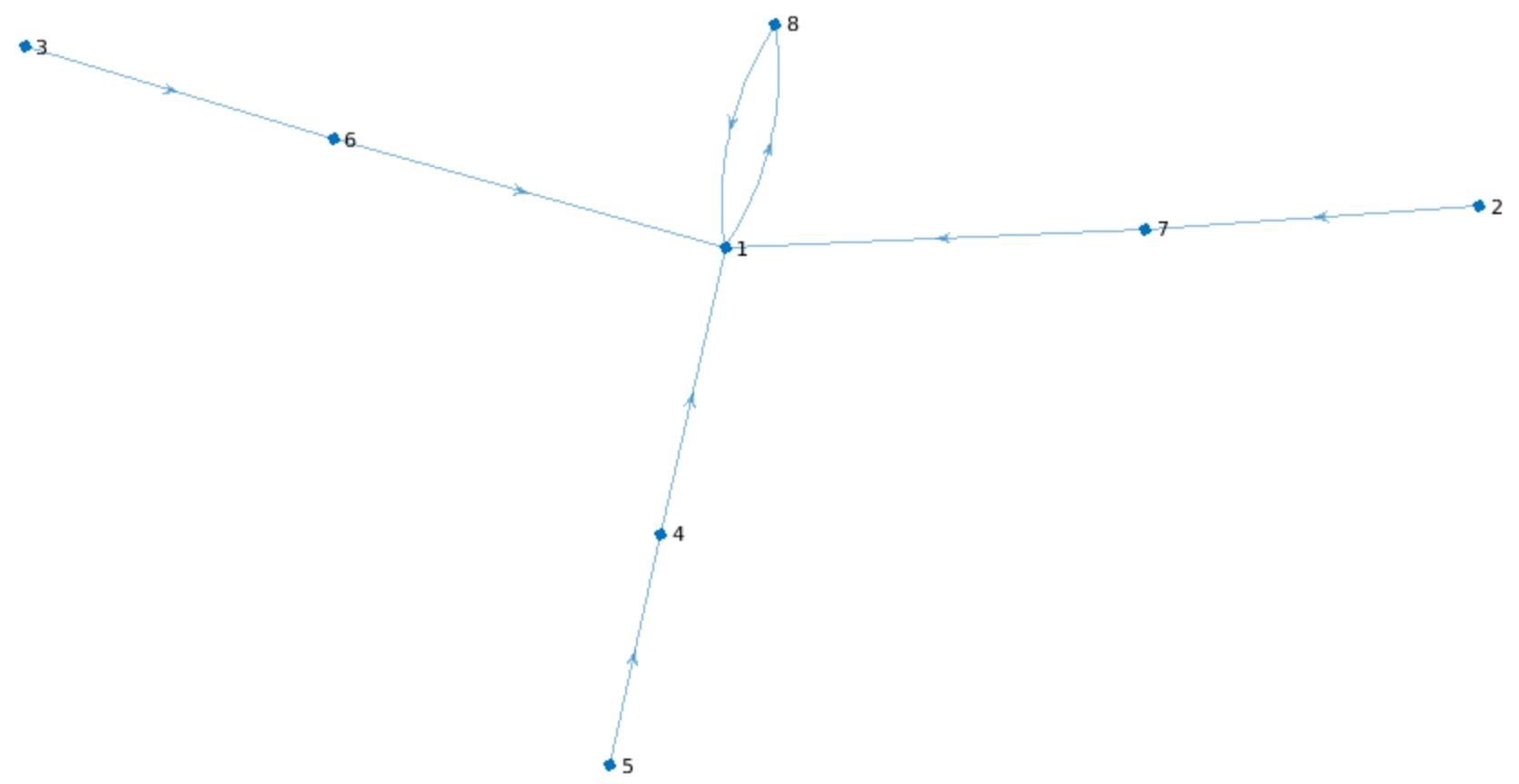
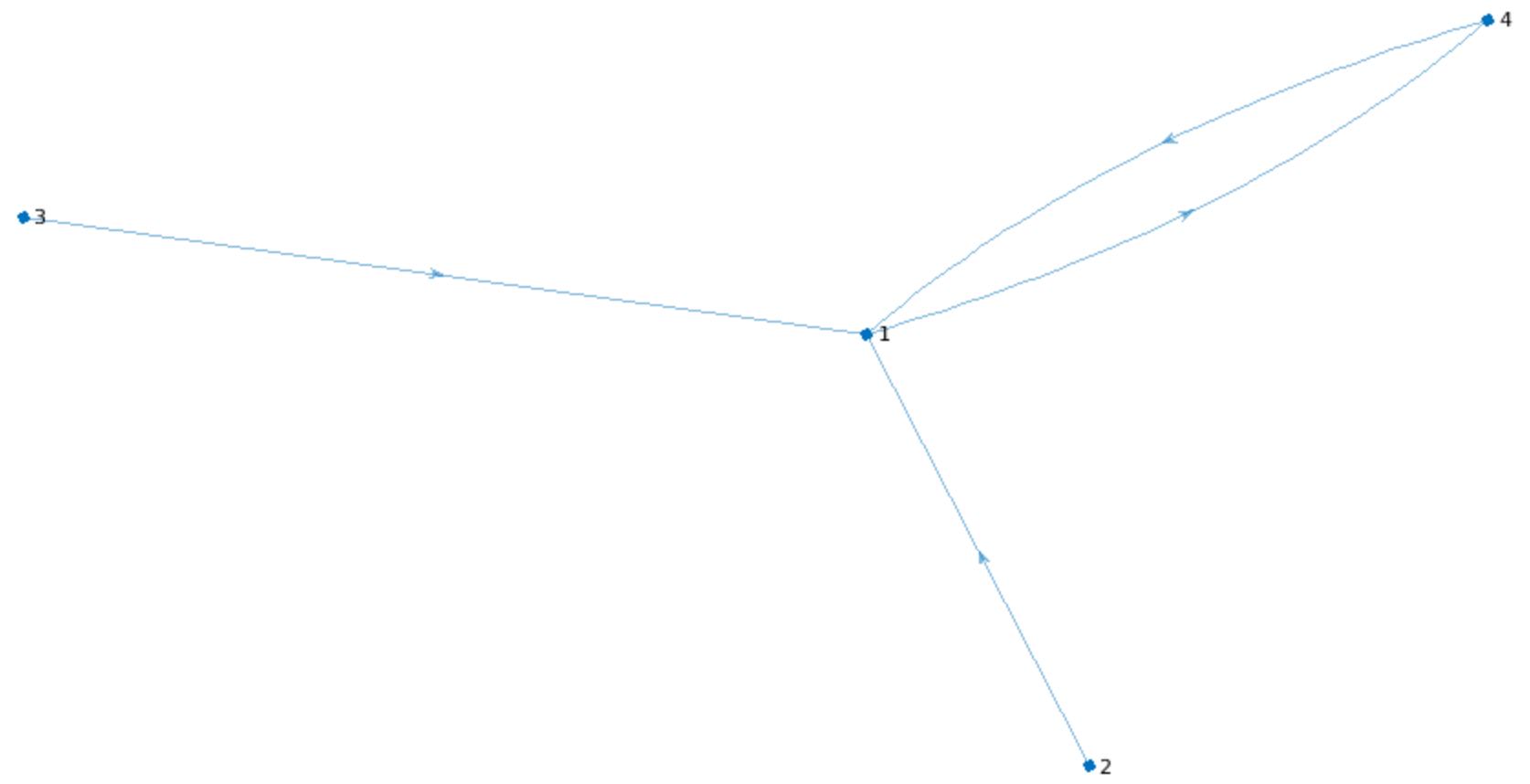


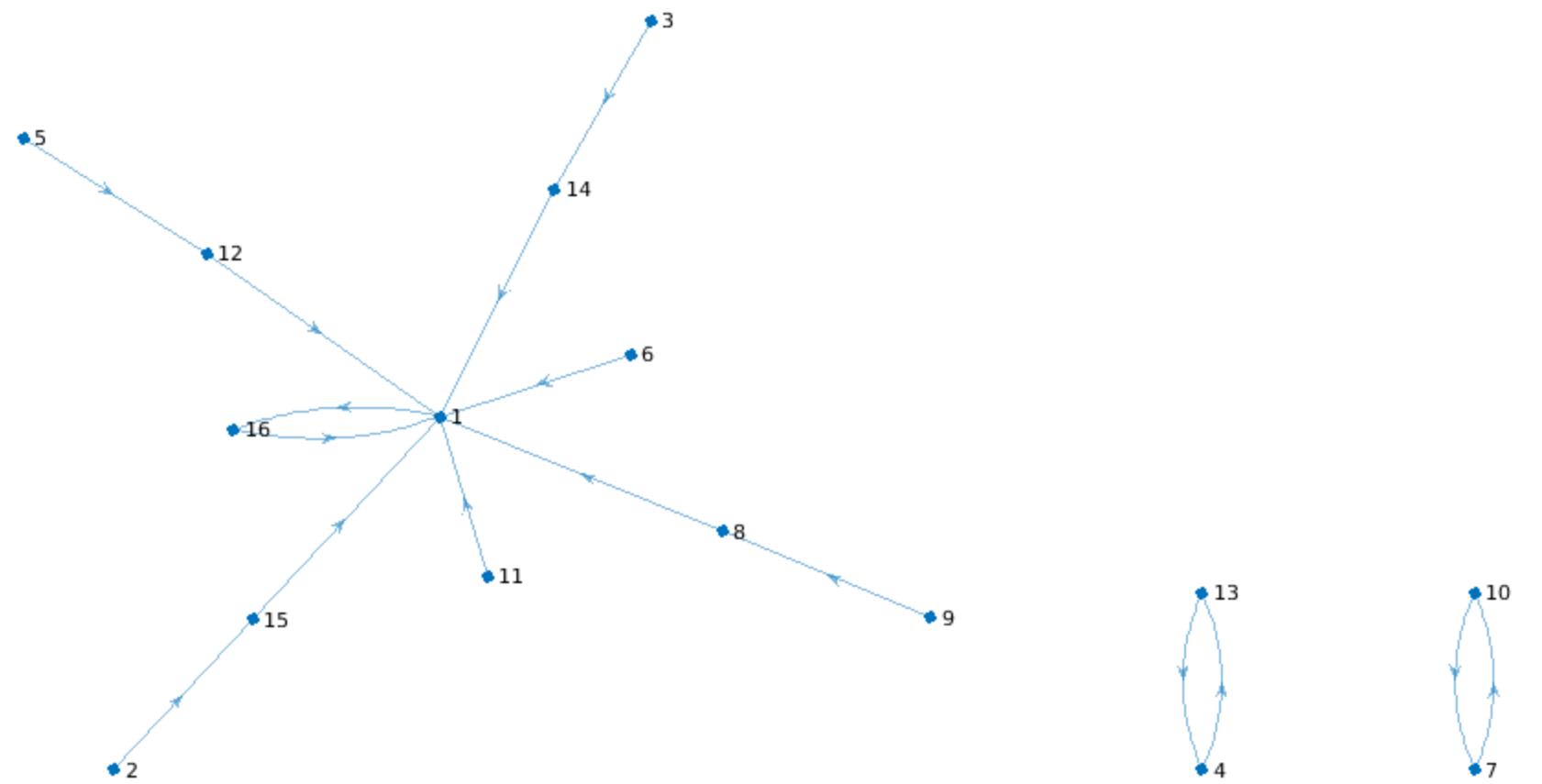


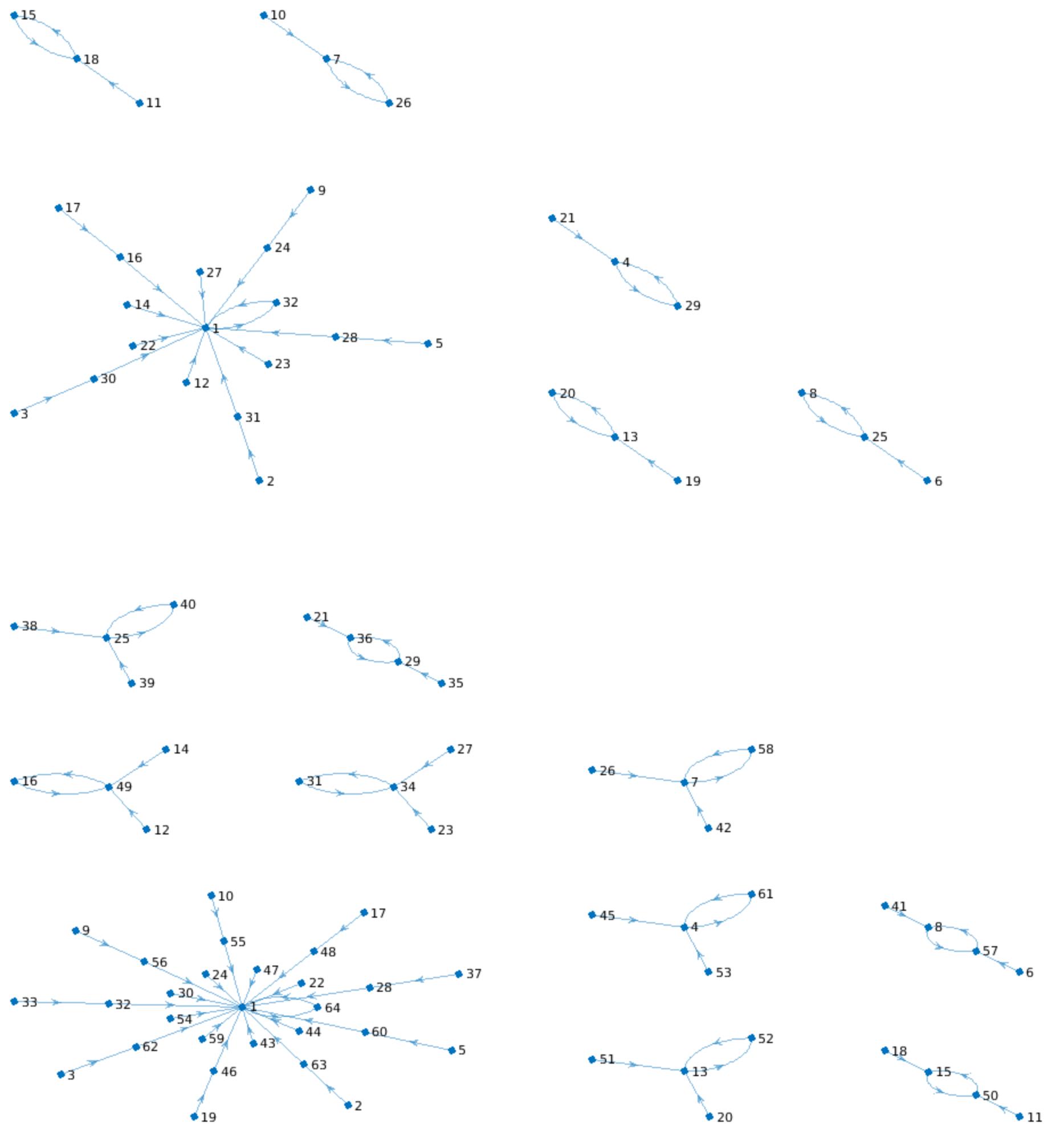
$\rho \rho \rho \rho$
 $\rho \rho \rho \rho$
 $\rho \rho \rho \rho$

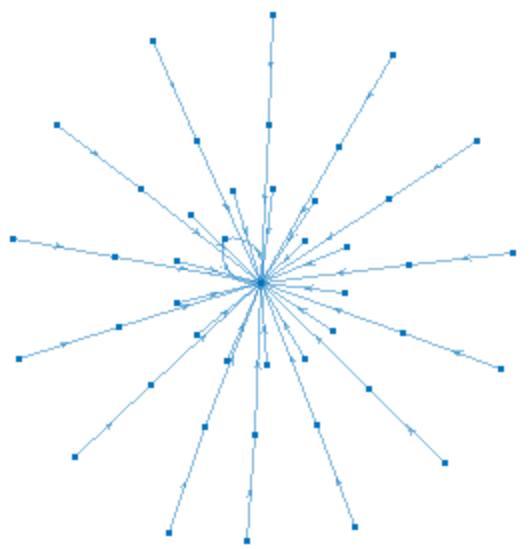
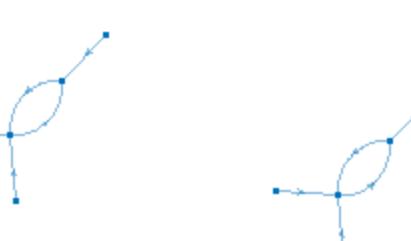
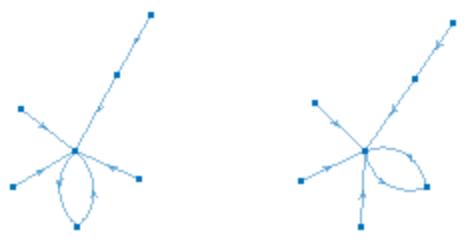


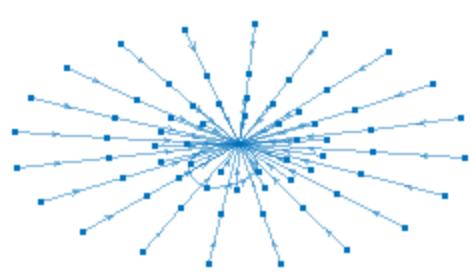


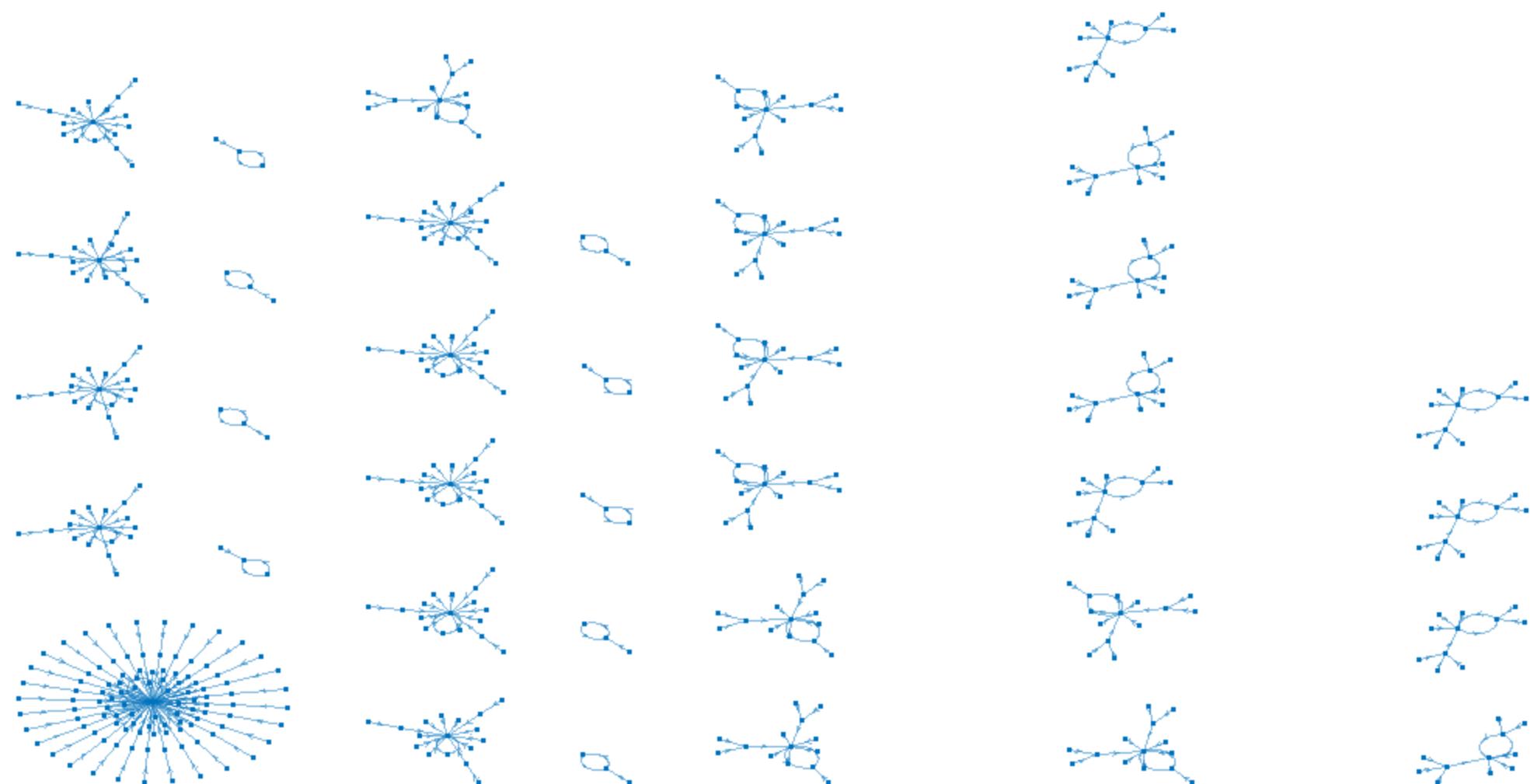


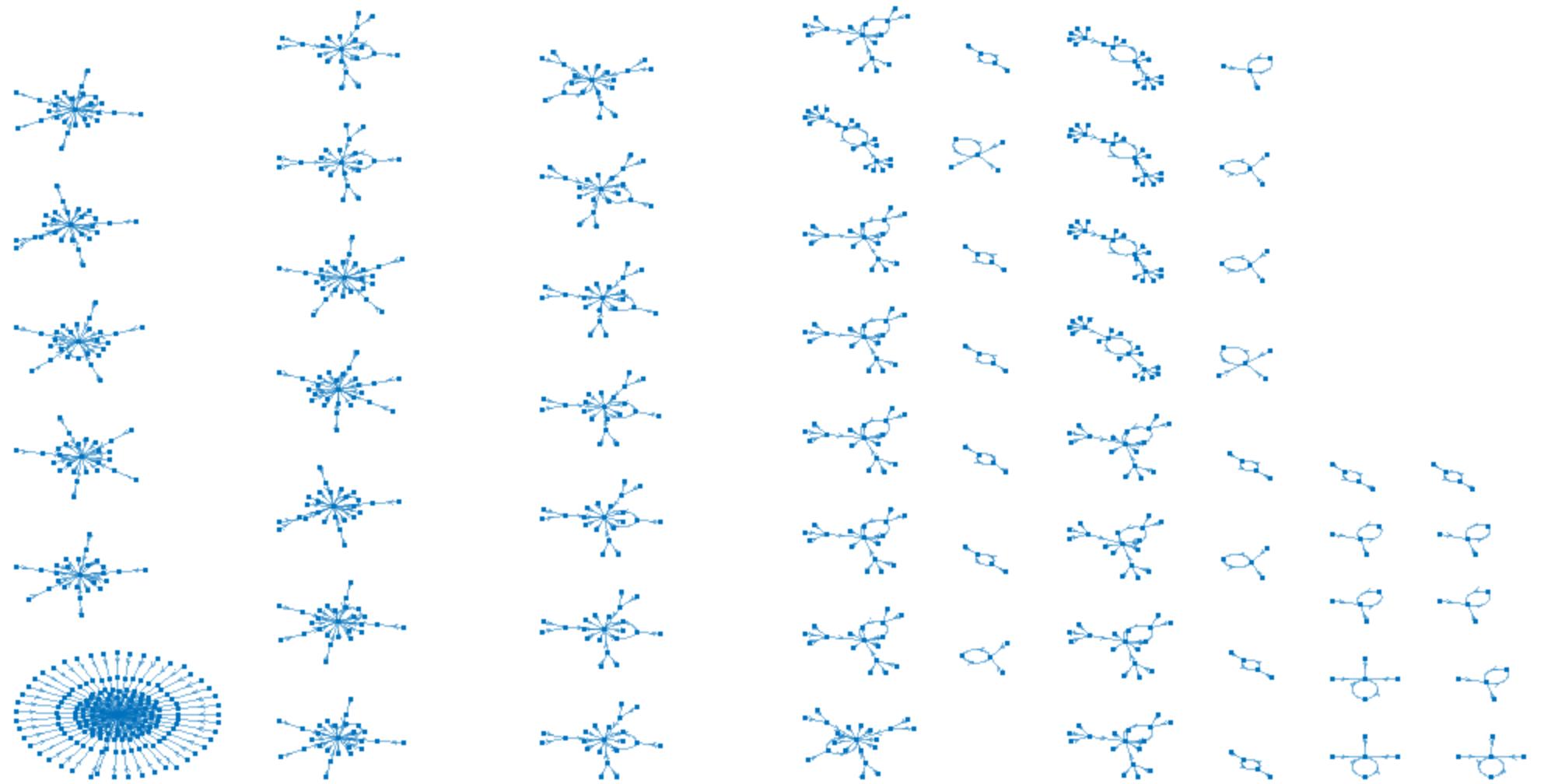


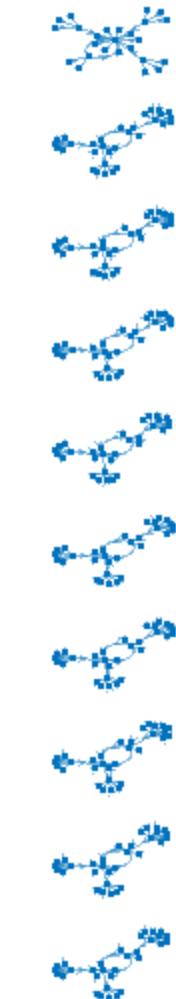
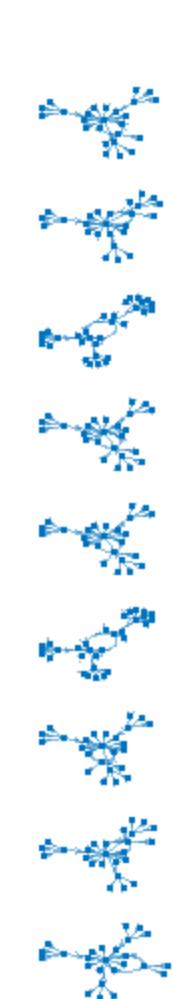
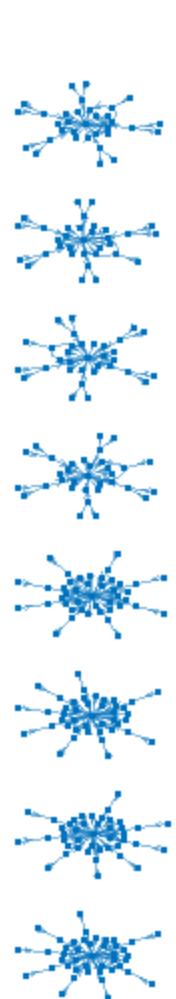
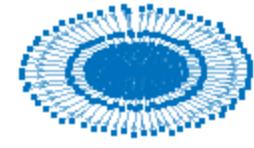
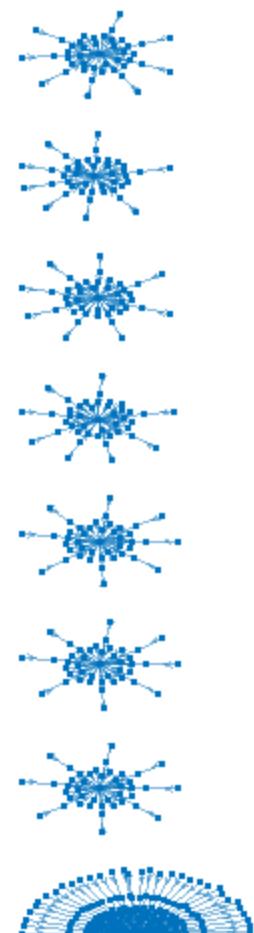




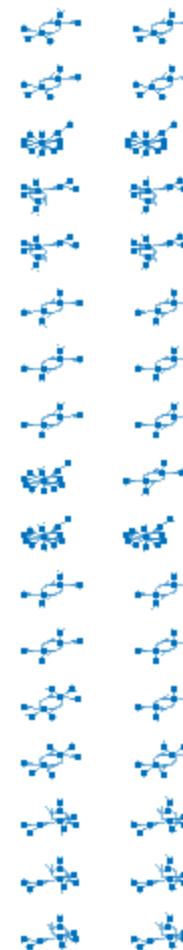
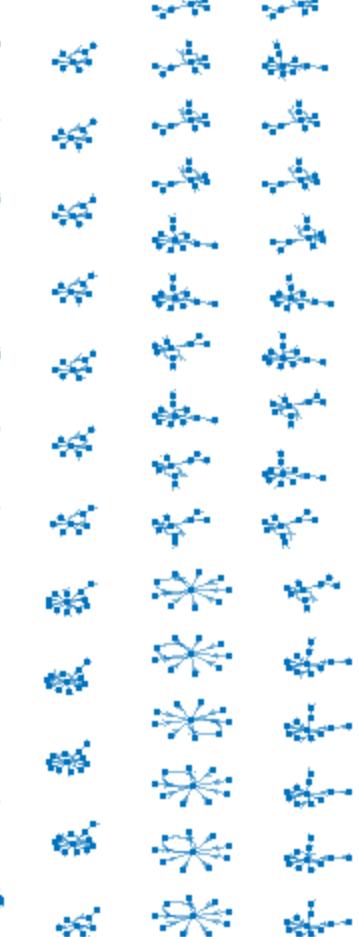
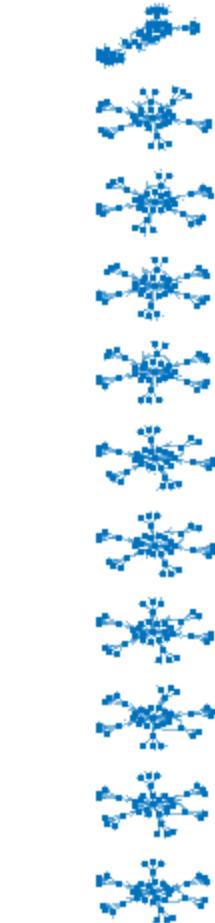
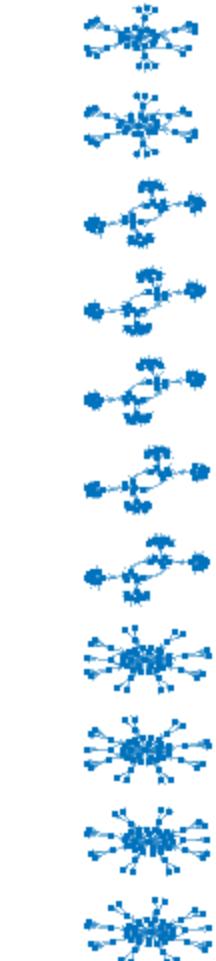
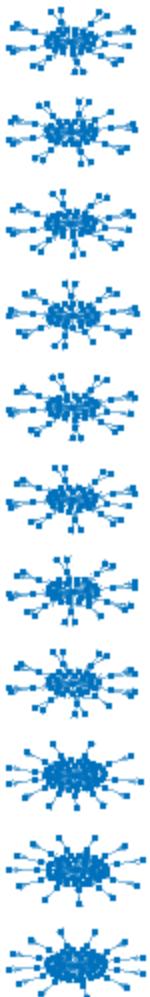
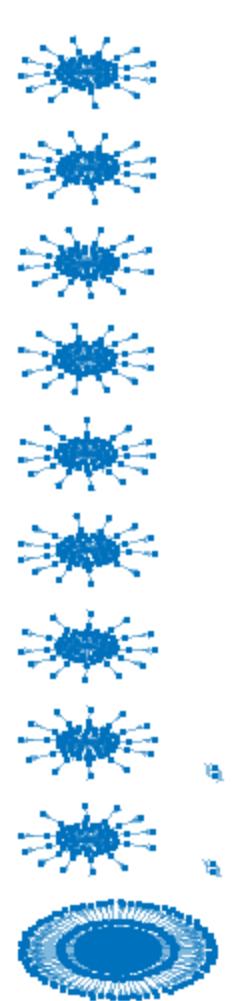








1. 2. 3. 4. 5. 6. 7. 8. 9. 10.



好

好

好

好

好

好

好

好

好

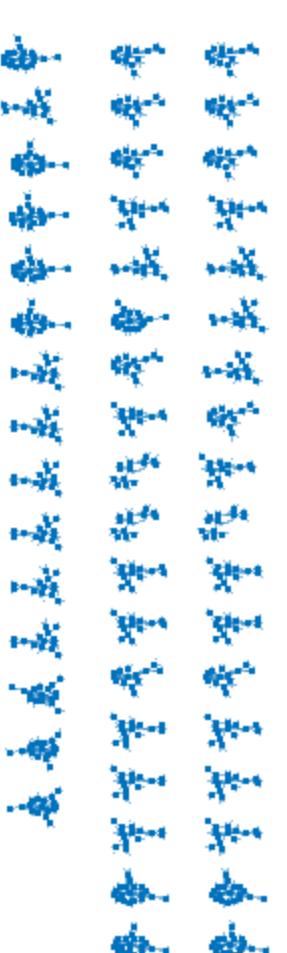
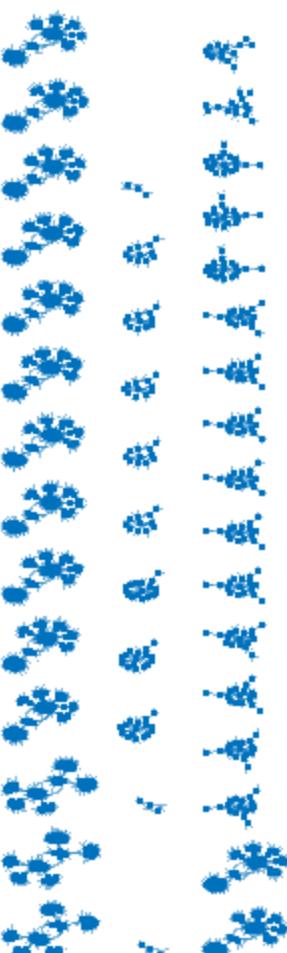
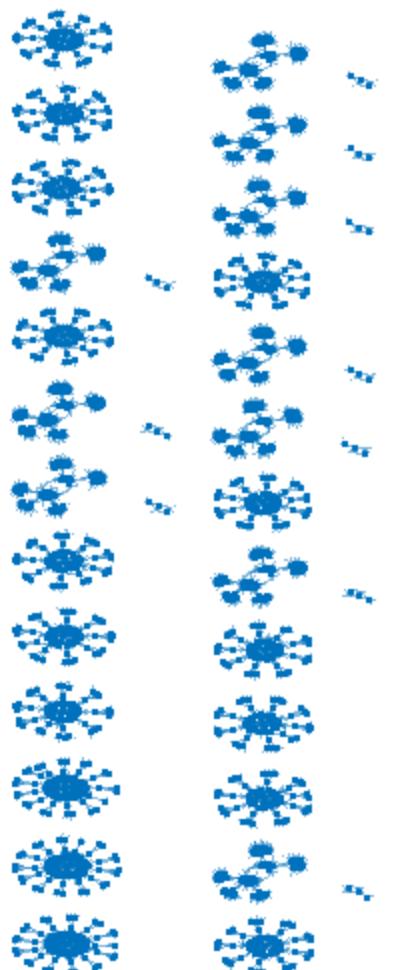
好

好

好

好

好

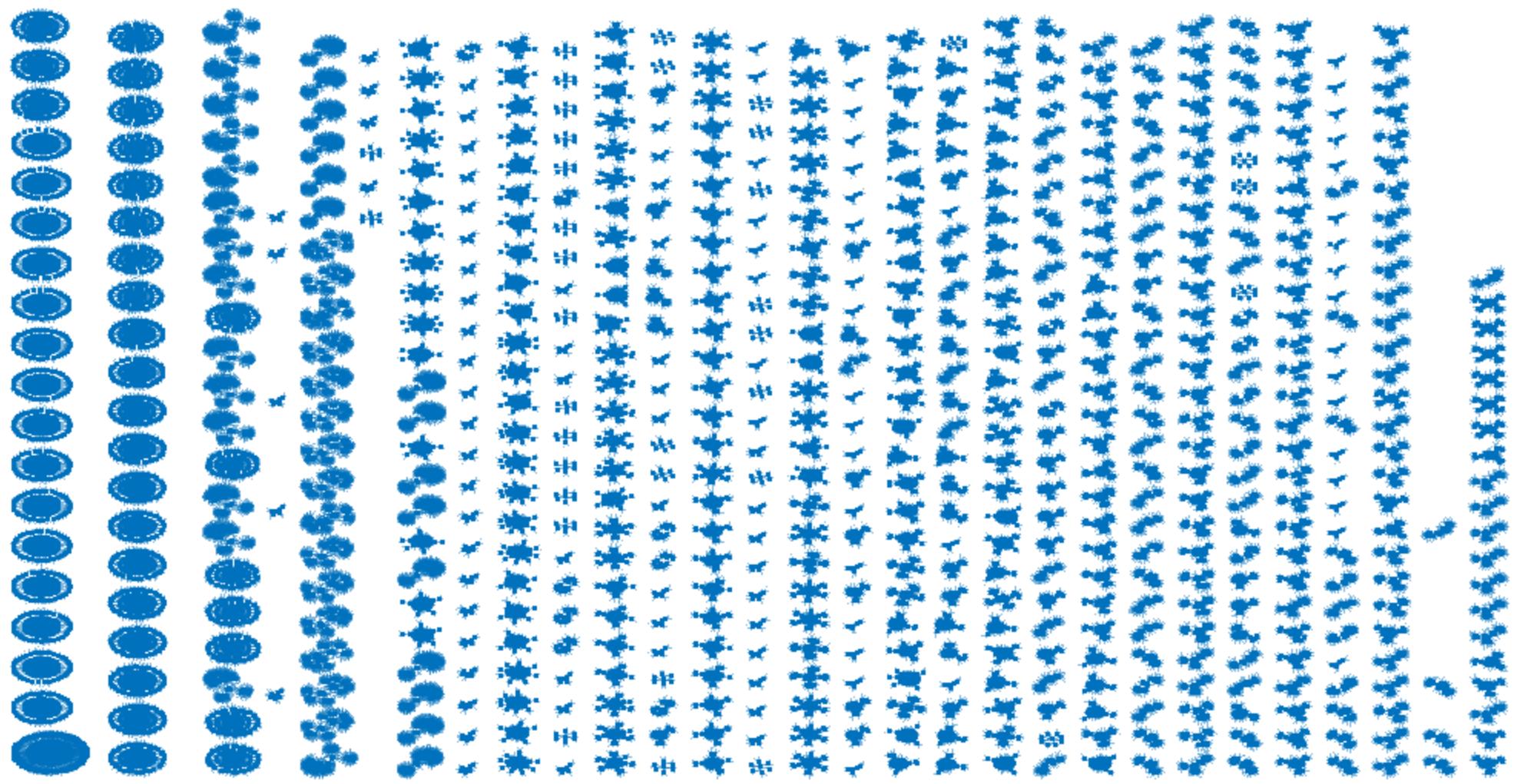


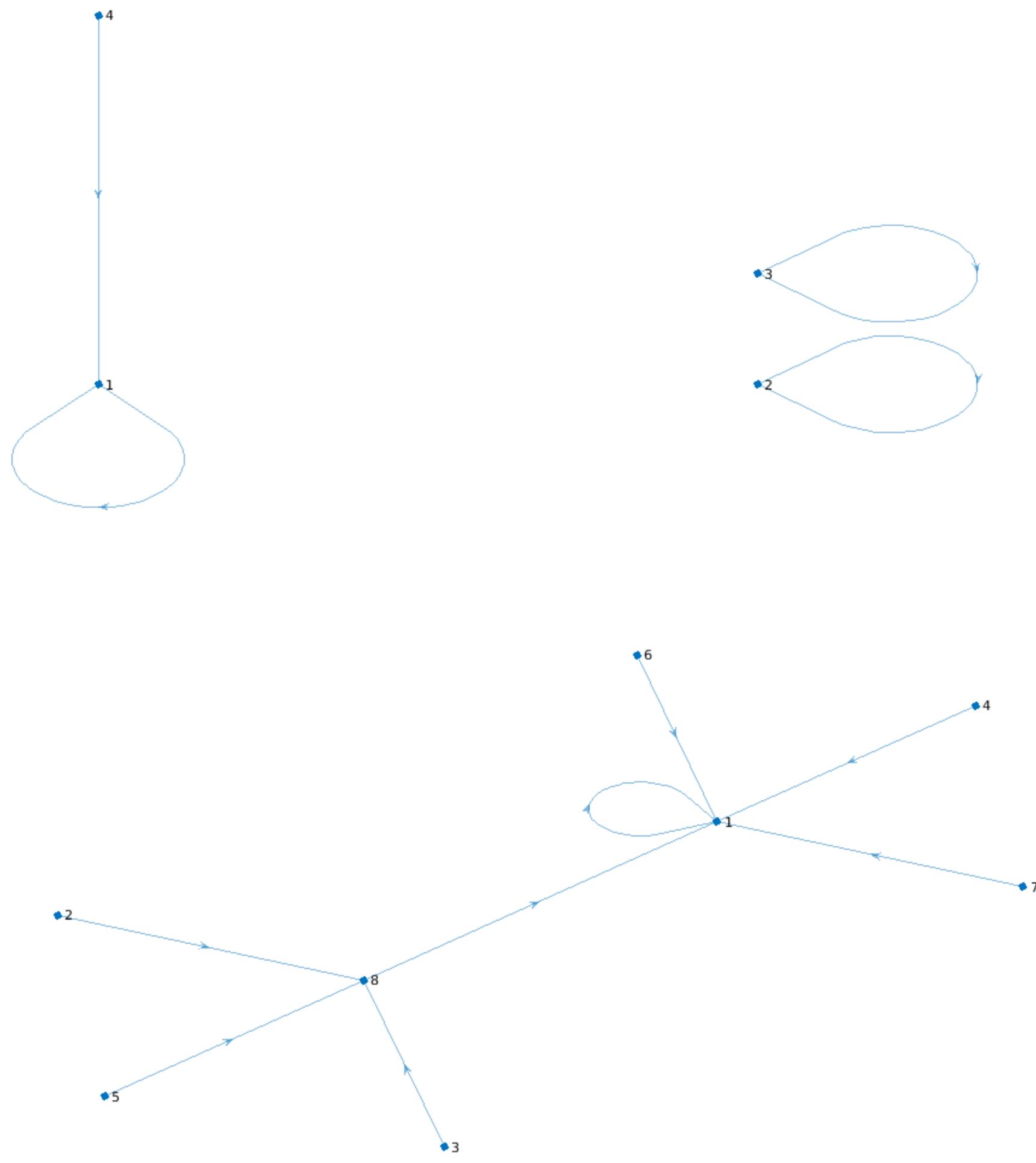
「我就是我，是颜色不一样的烟火。
天空没有鸟飞过，但我已飞过。
我就是我，是颜色不一样的烟火。
天空没有鸟飞过，但我已飞过。」

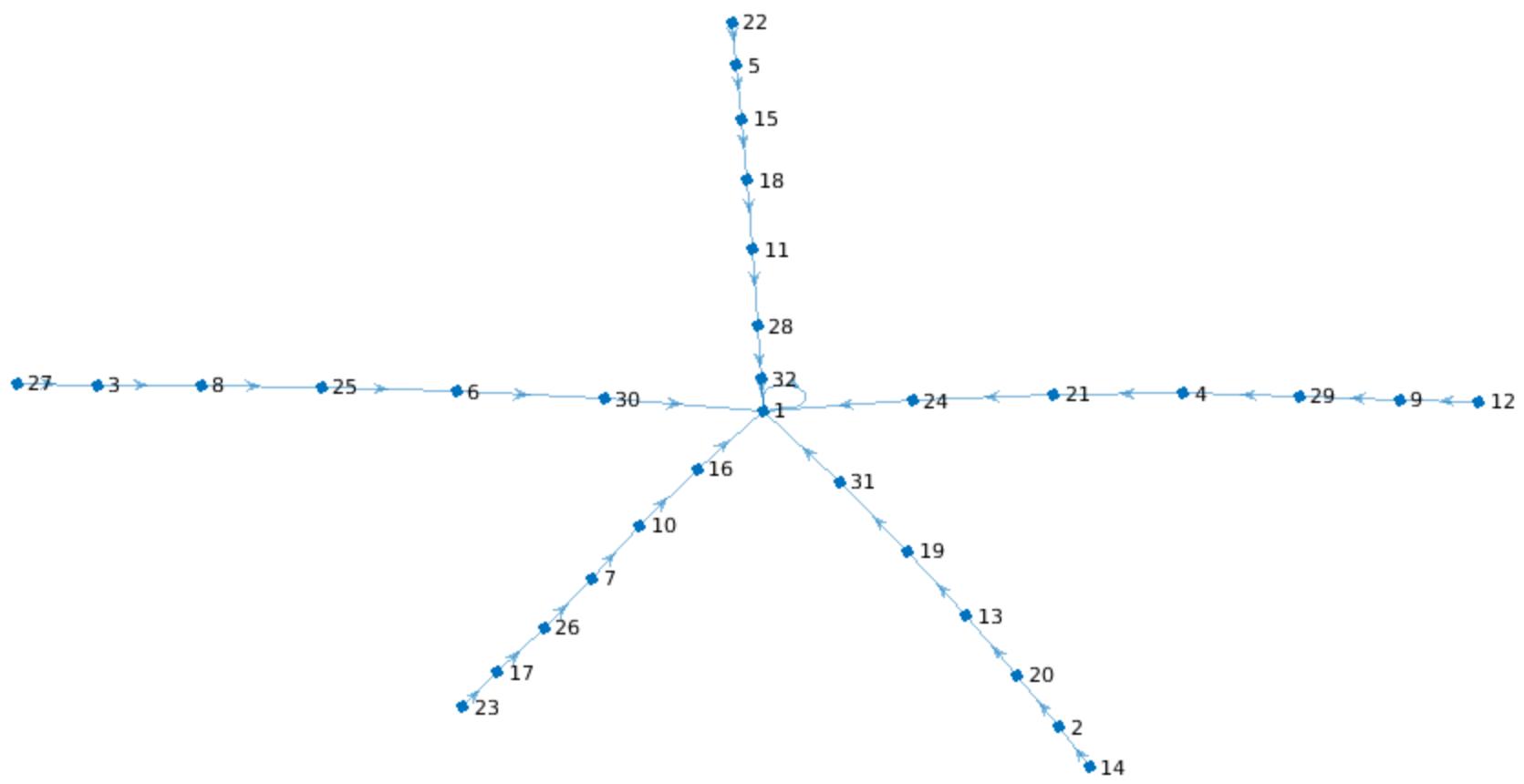
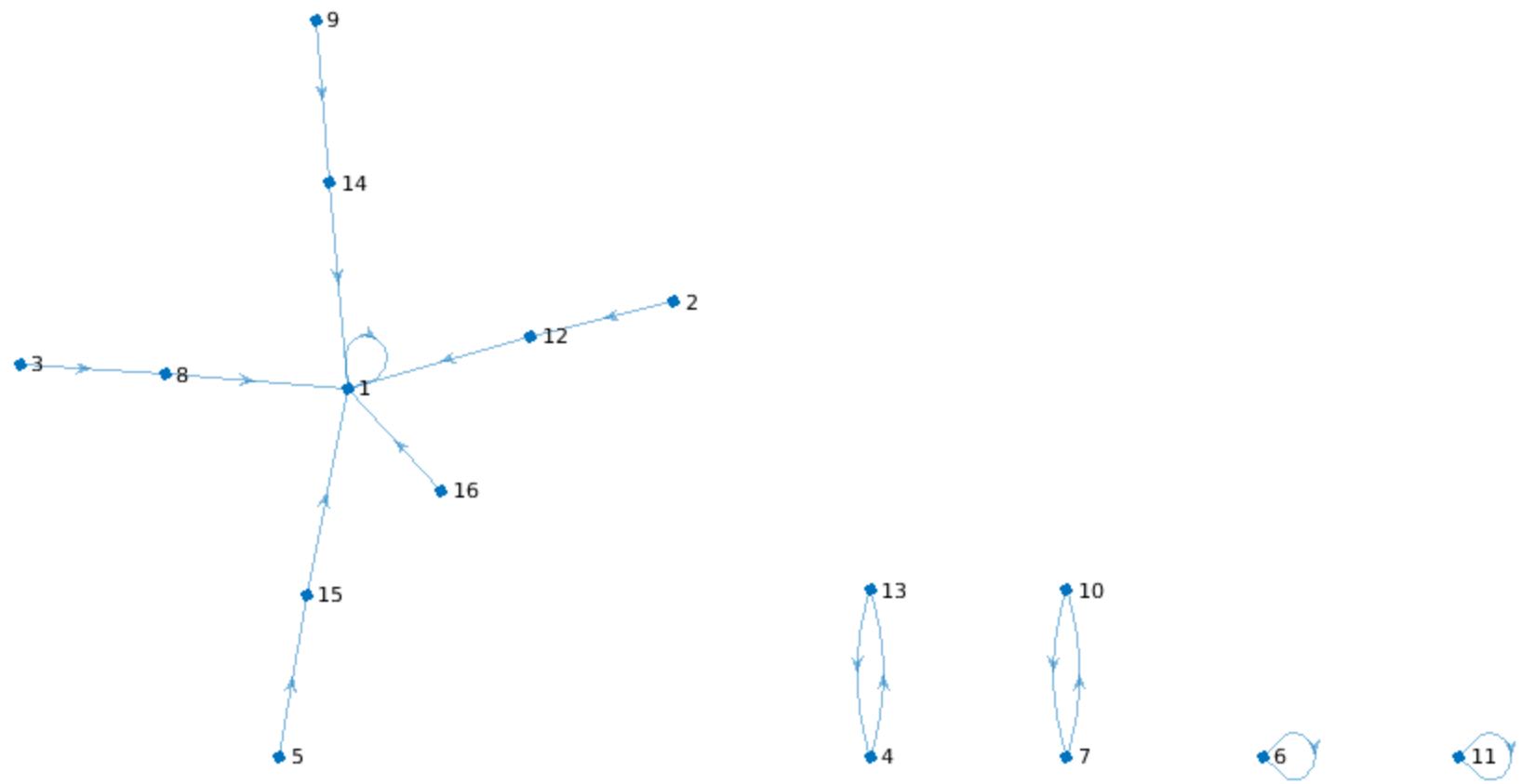
——《颜色不一样的烟火》
王菲

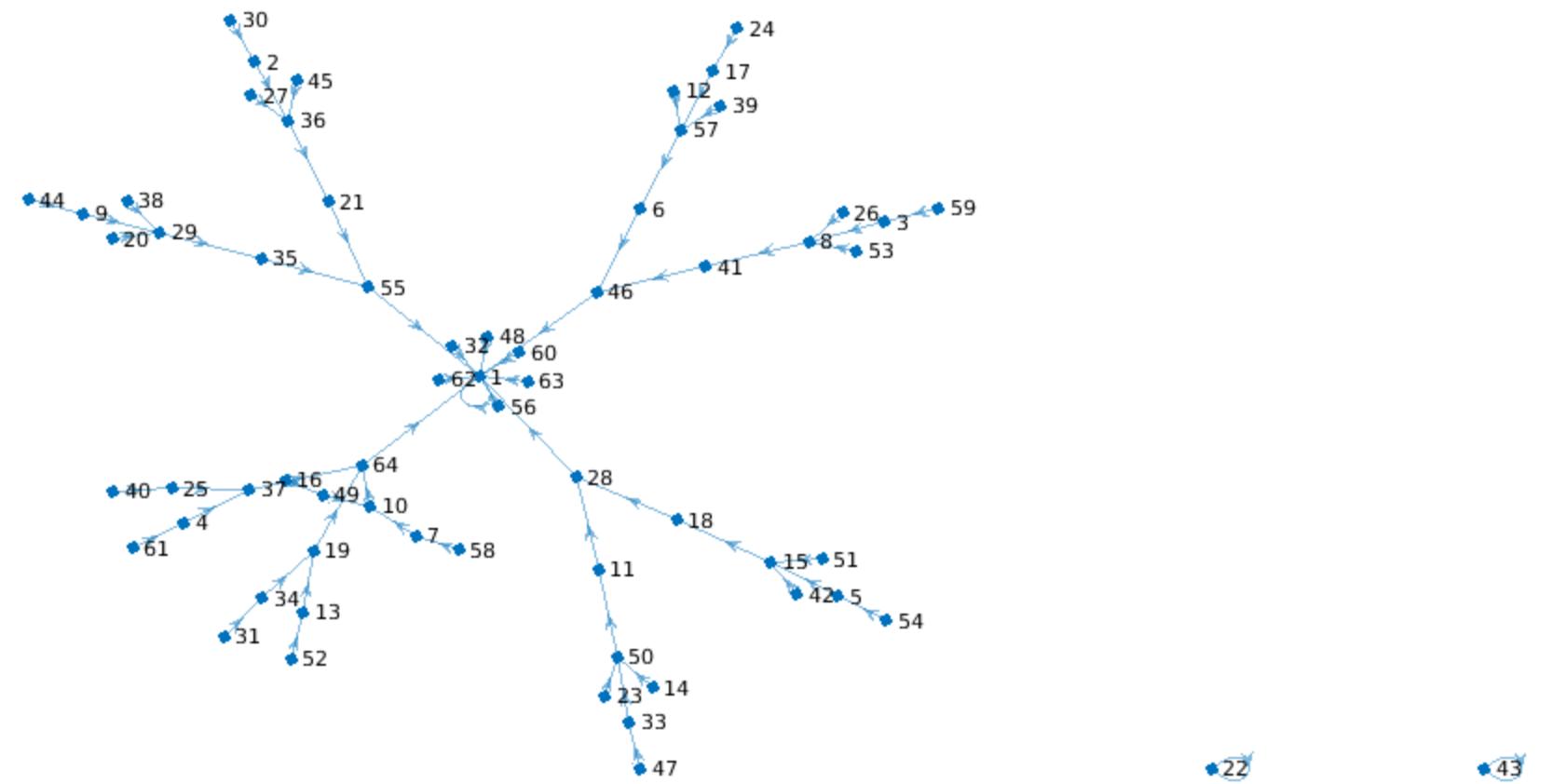


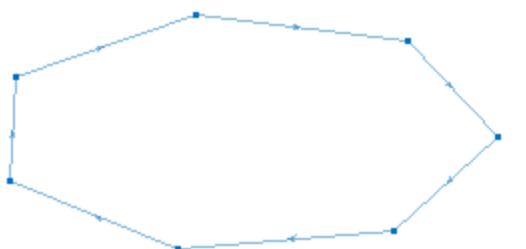
/ / / / / / / / / /

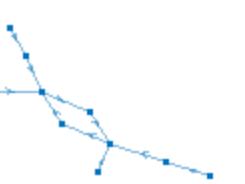
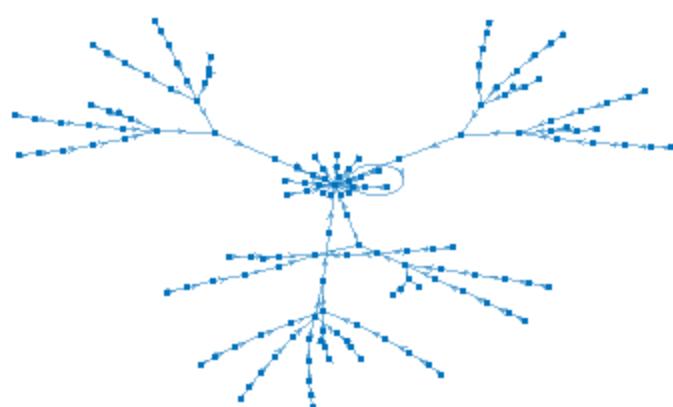


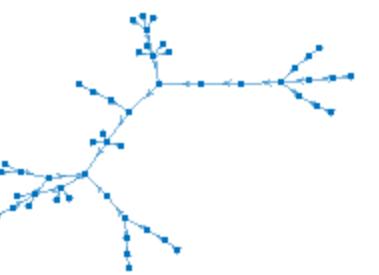
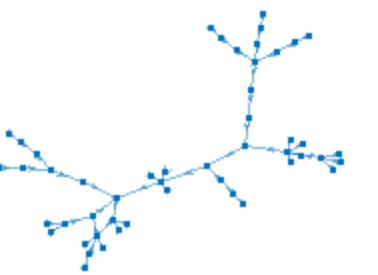
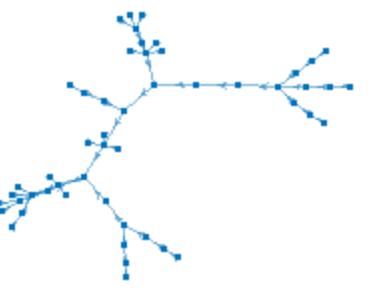
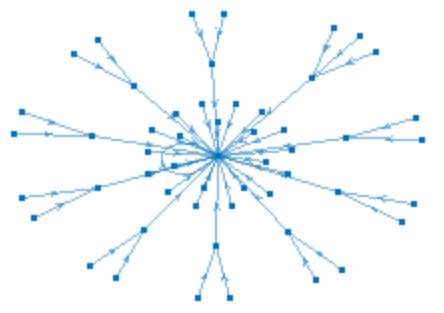
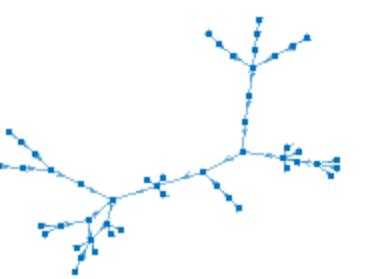
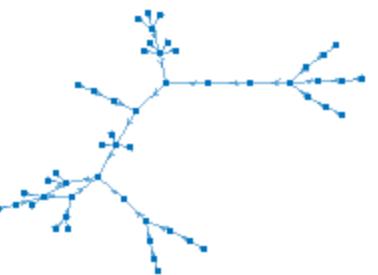
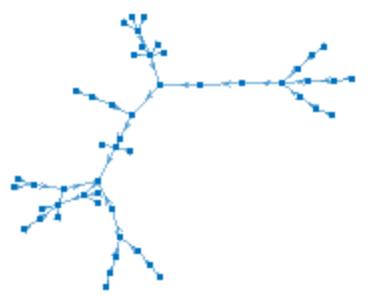
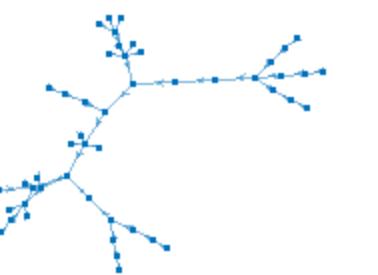
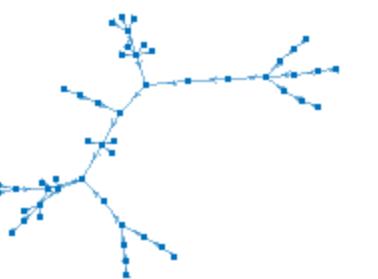
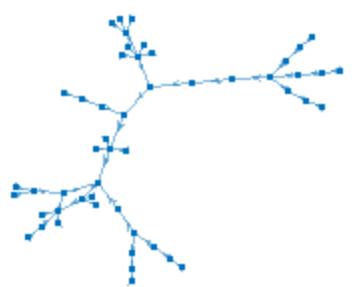


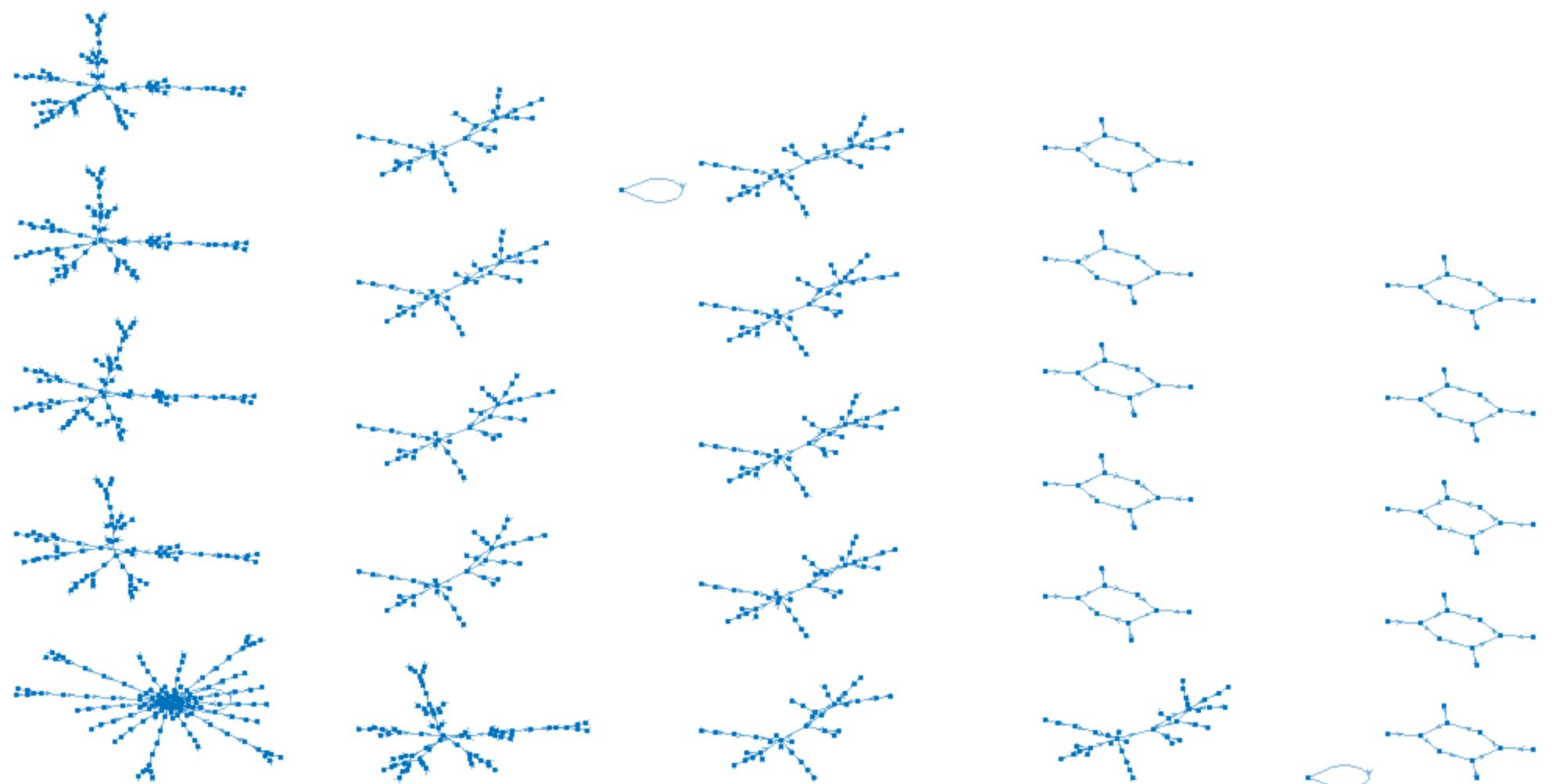


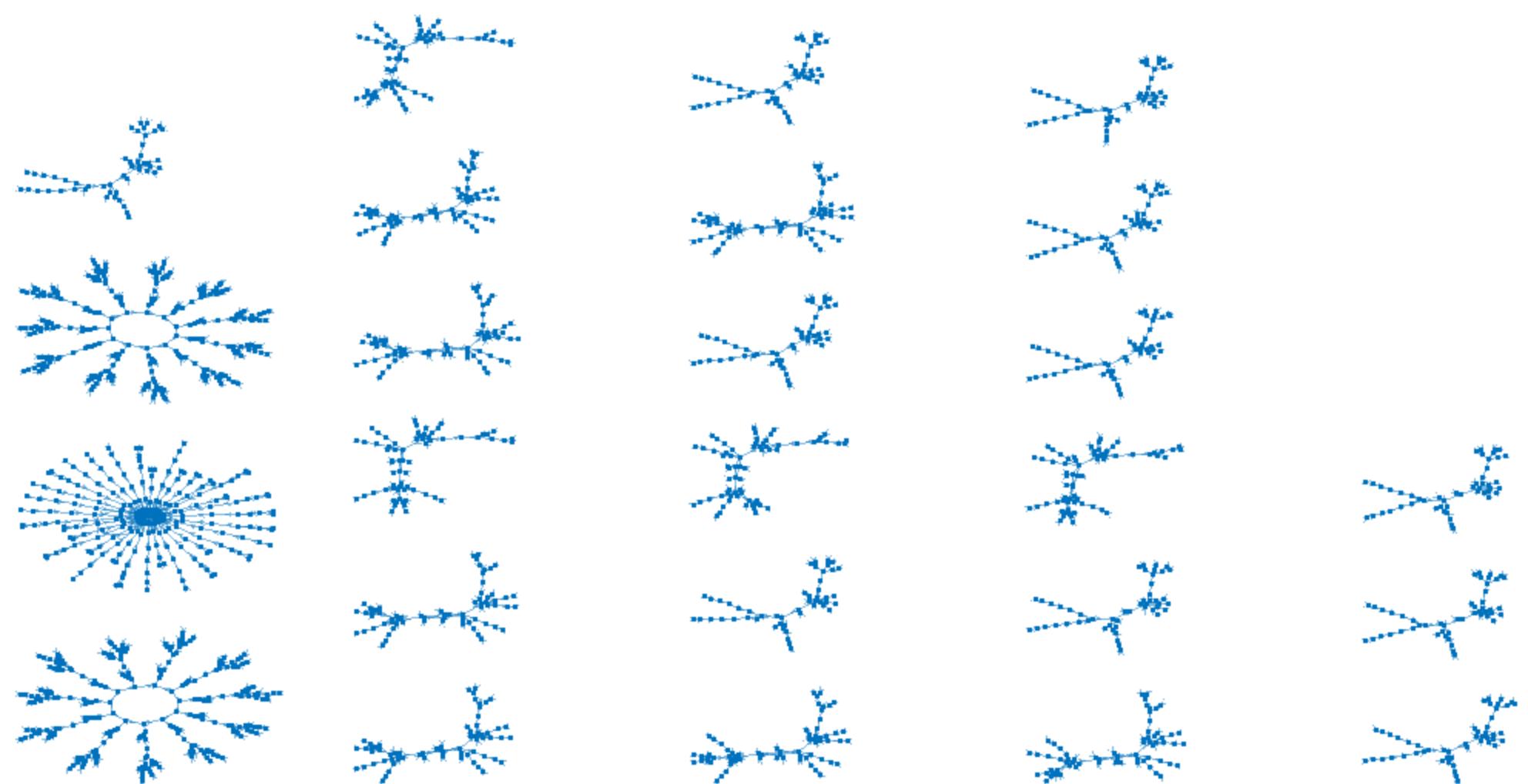


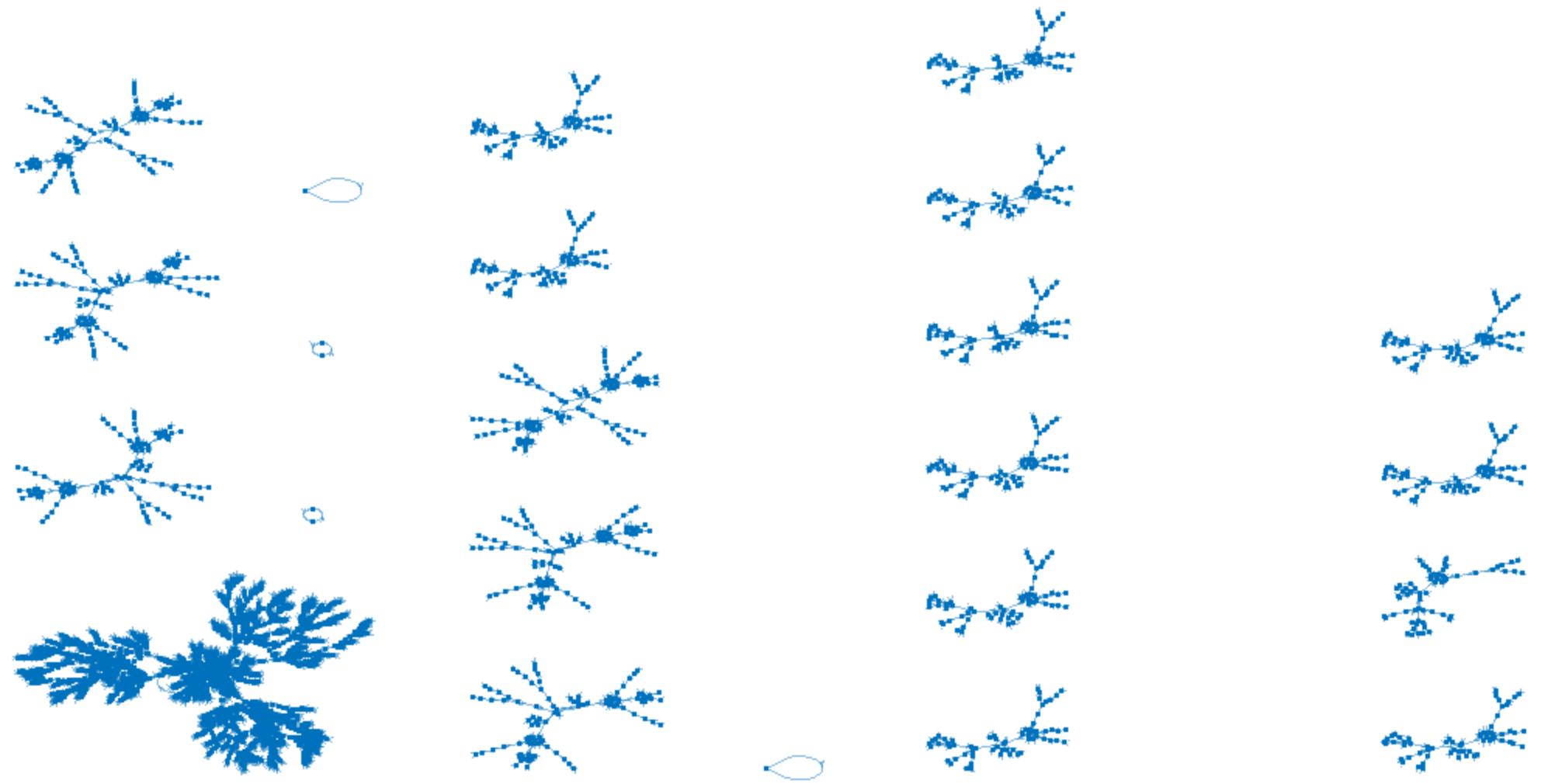


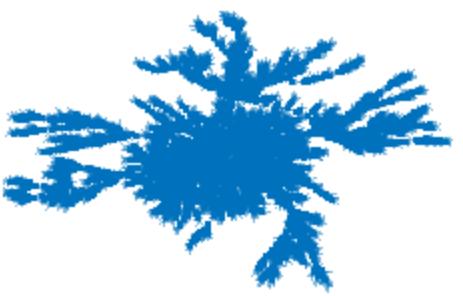
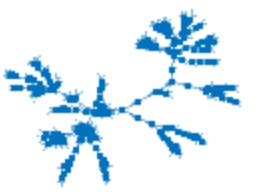


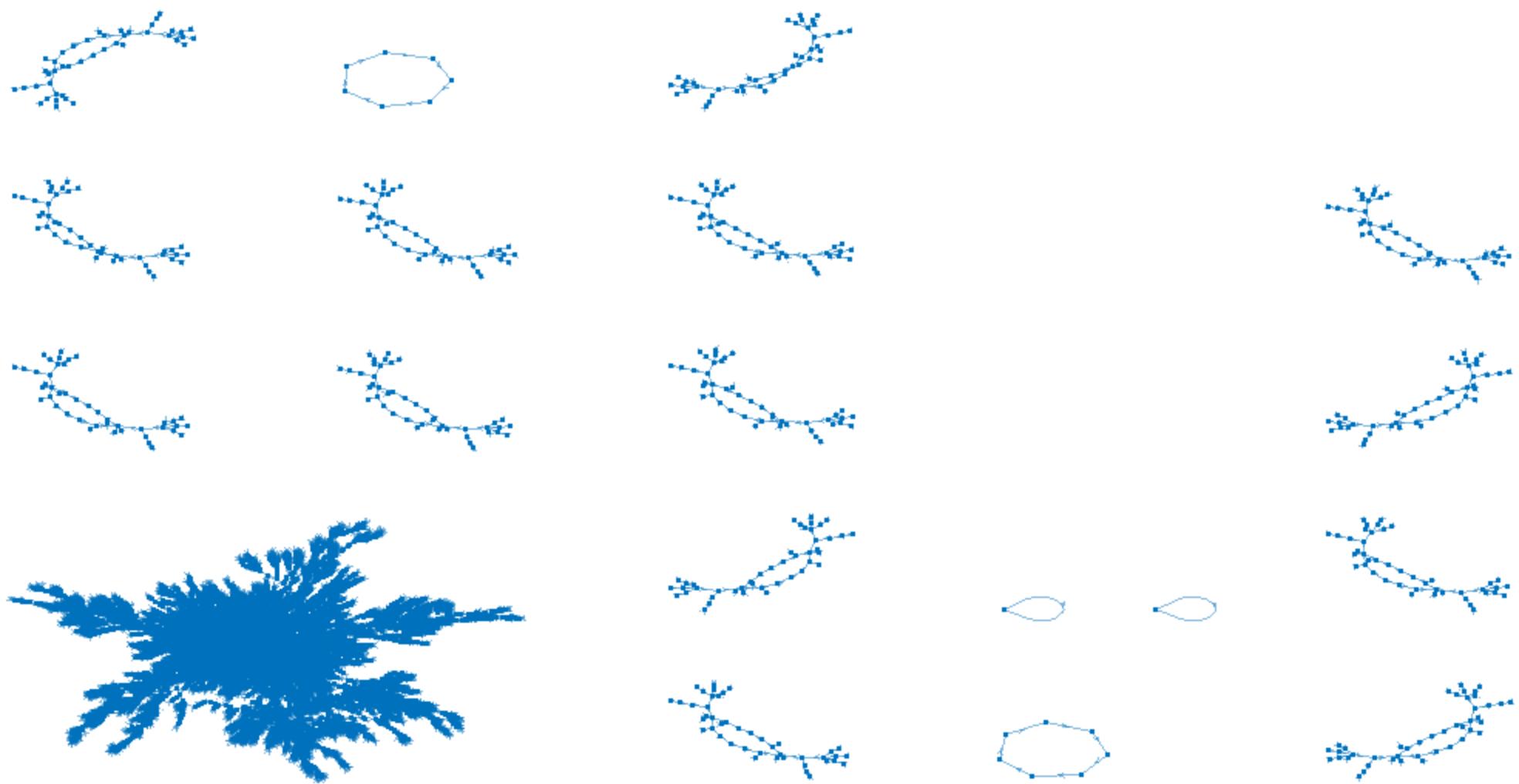


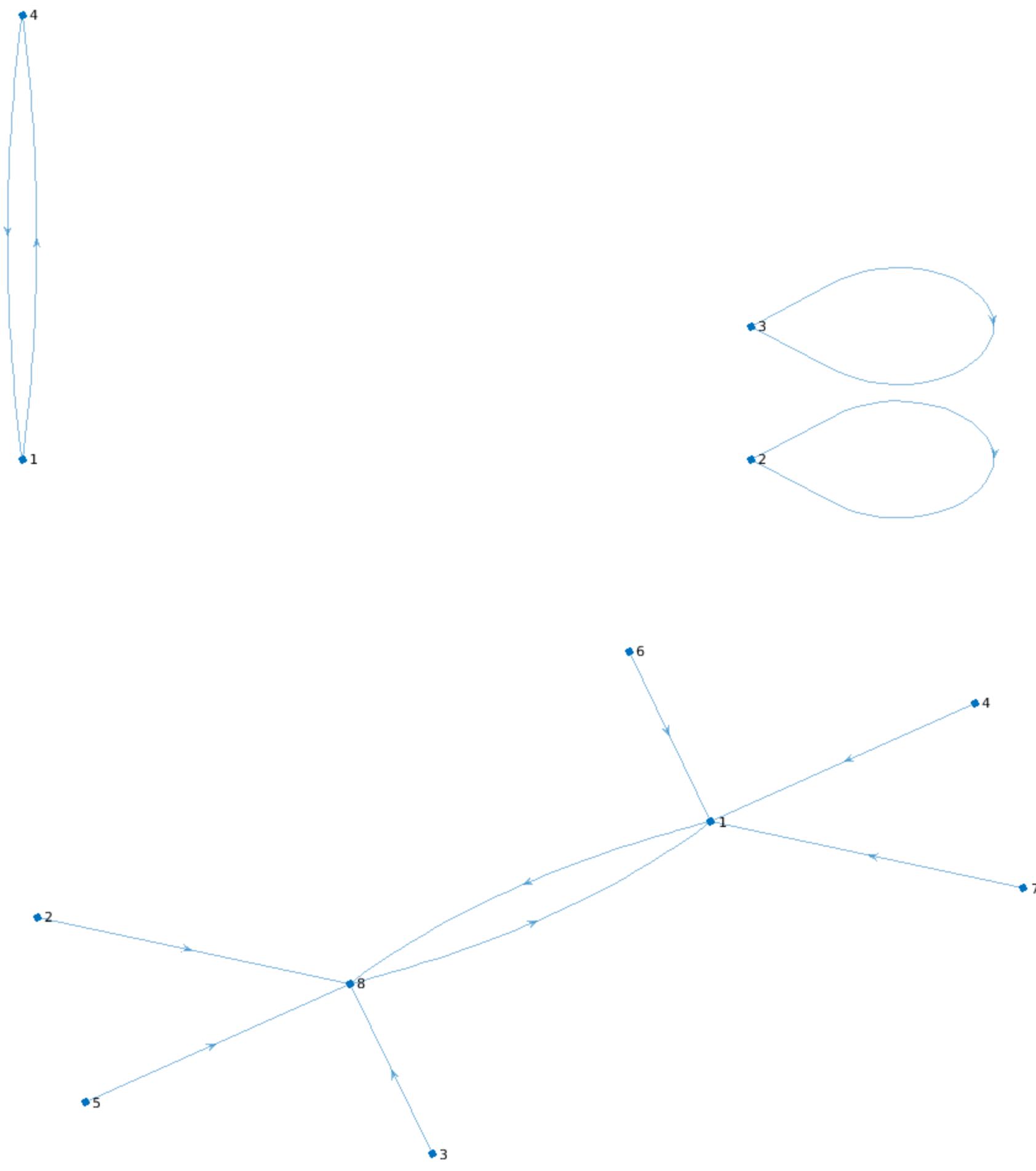


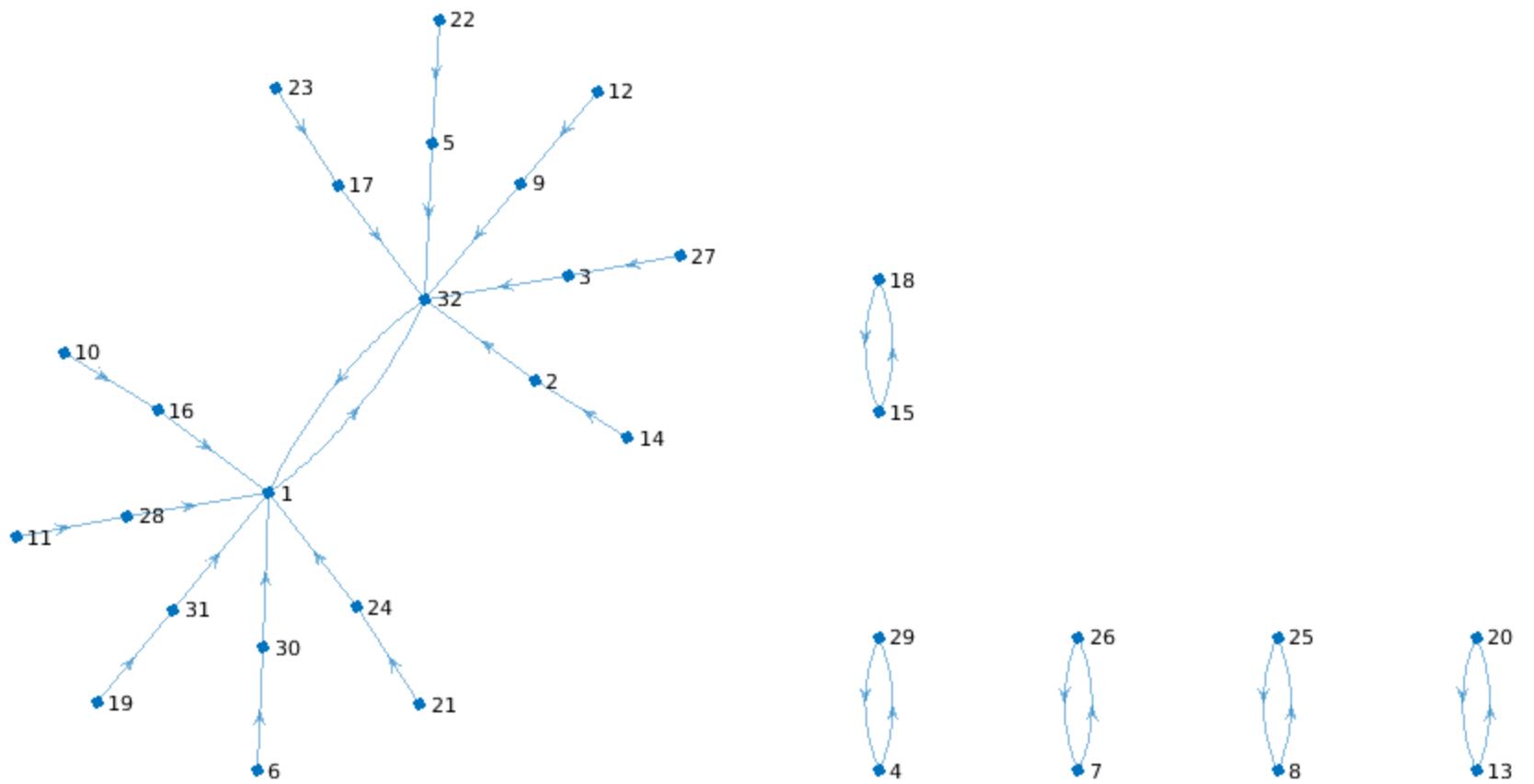
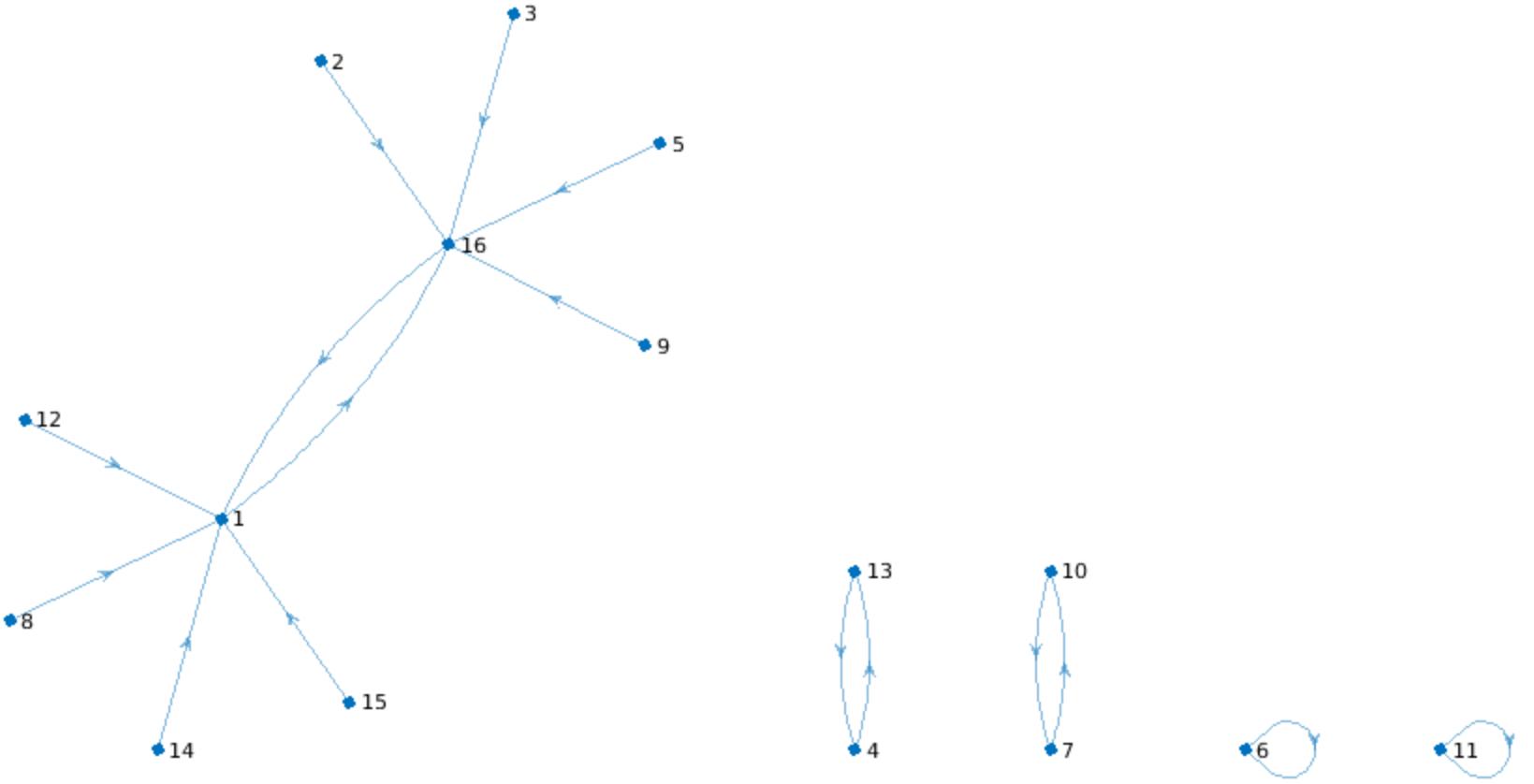


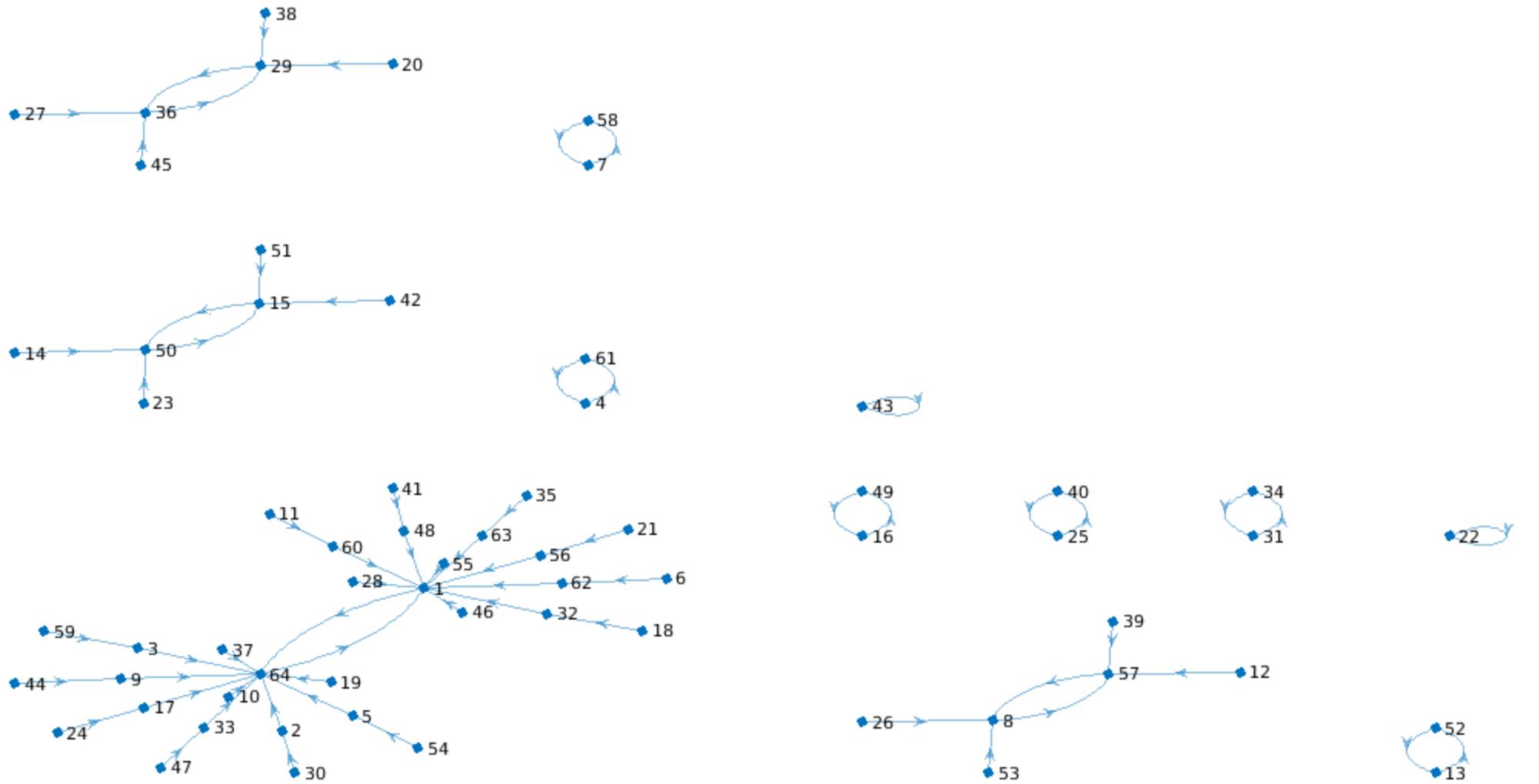


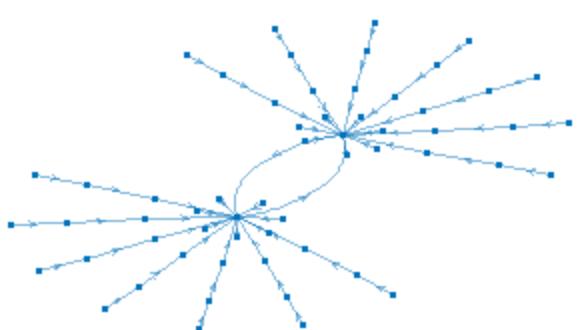


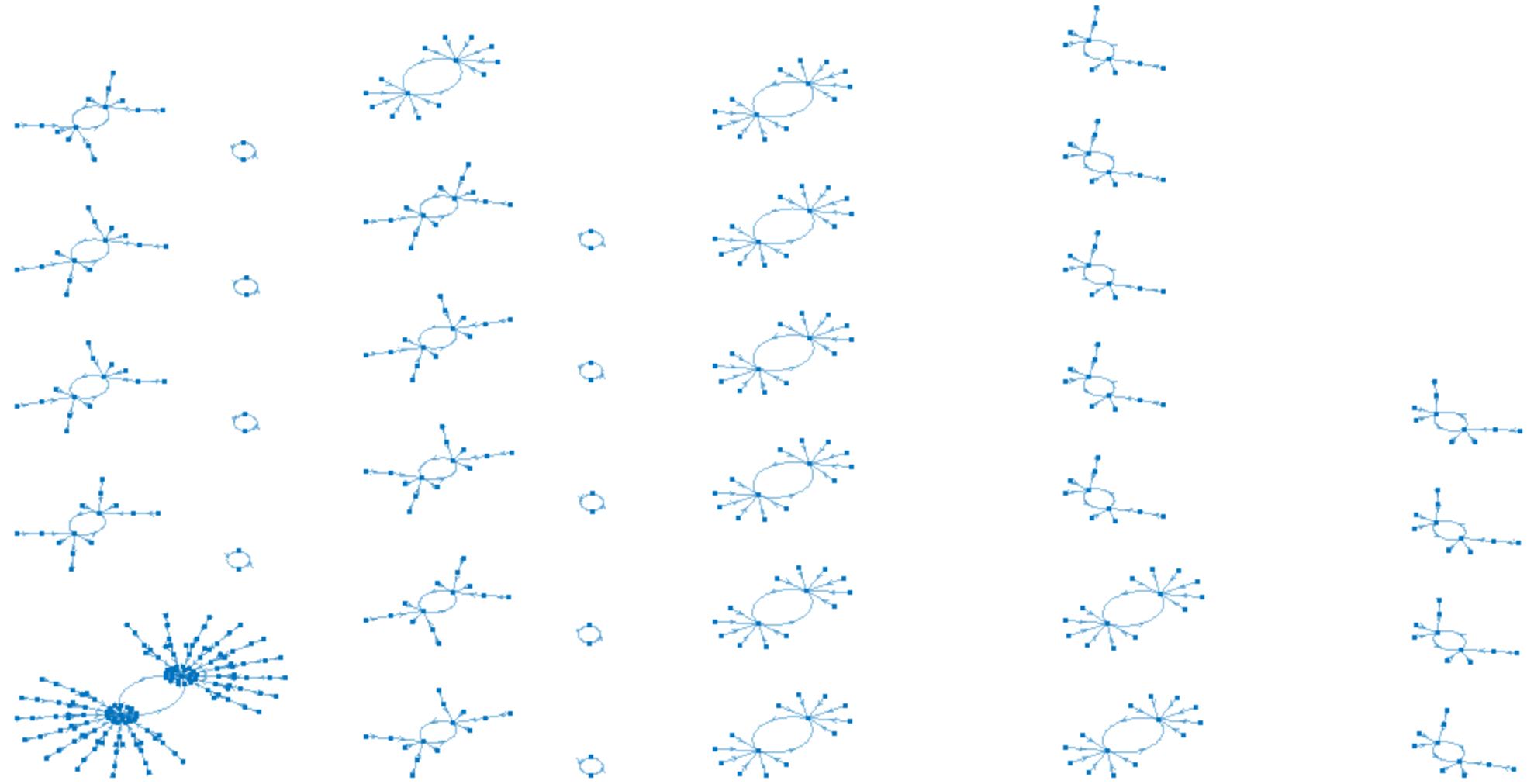


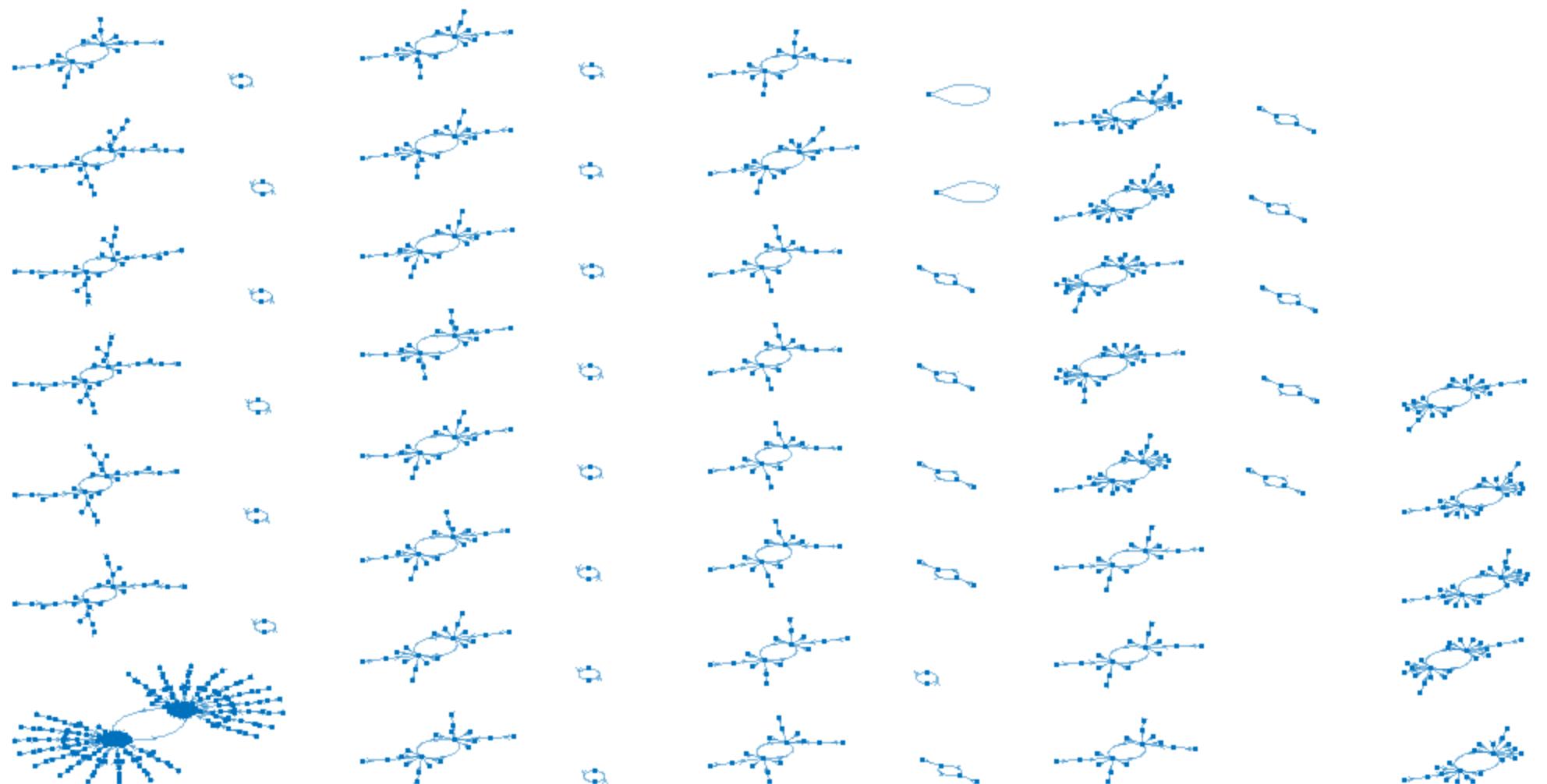


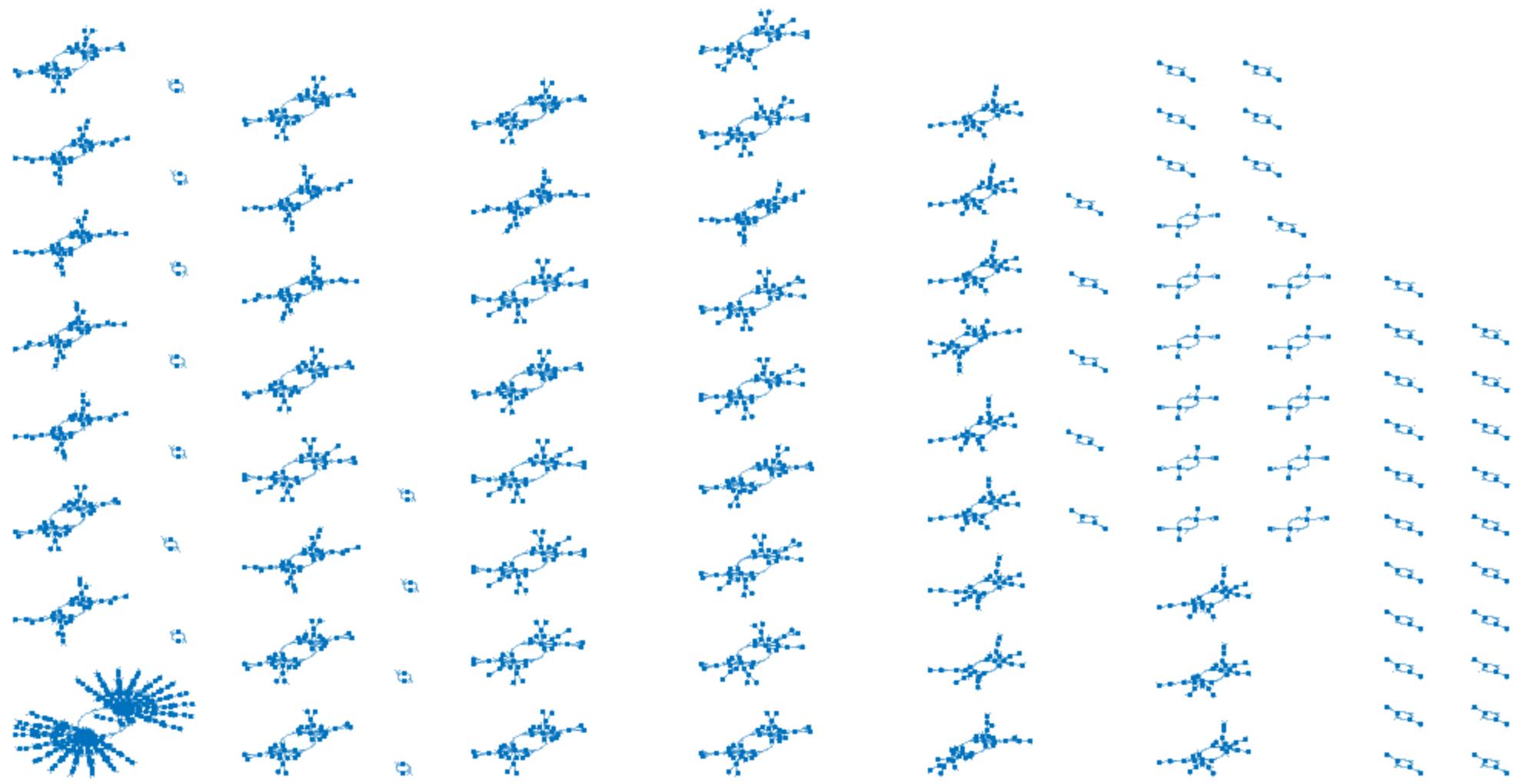


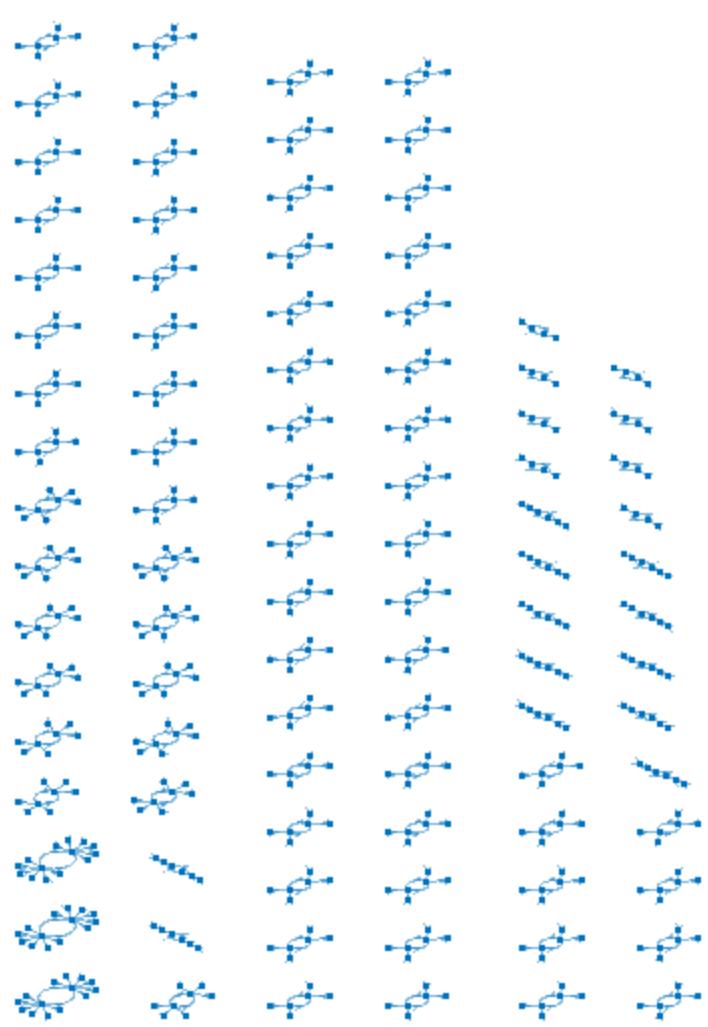
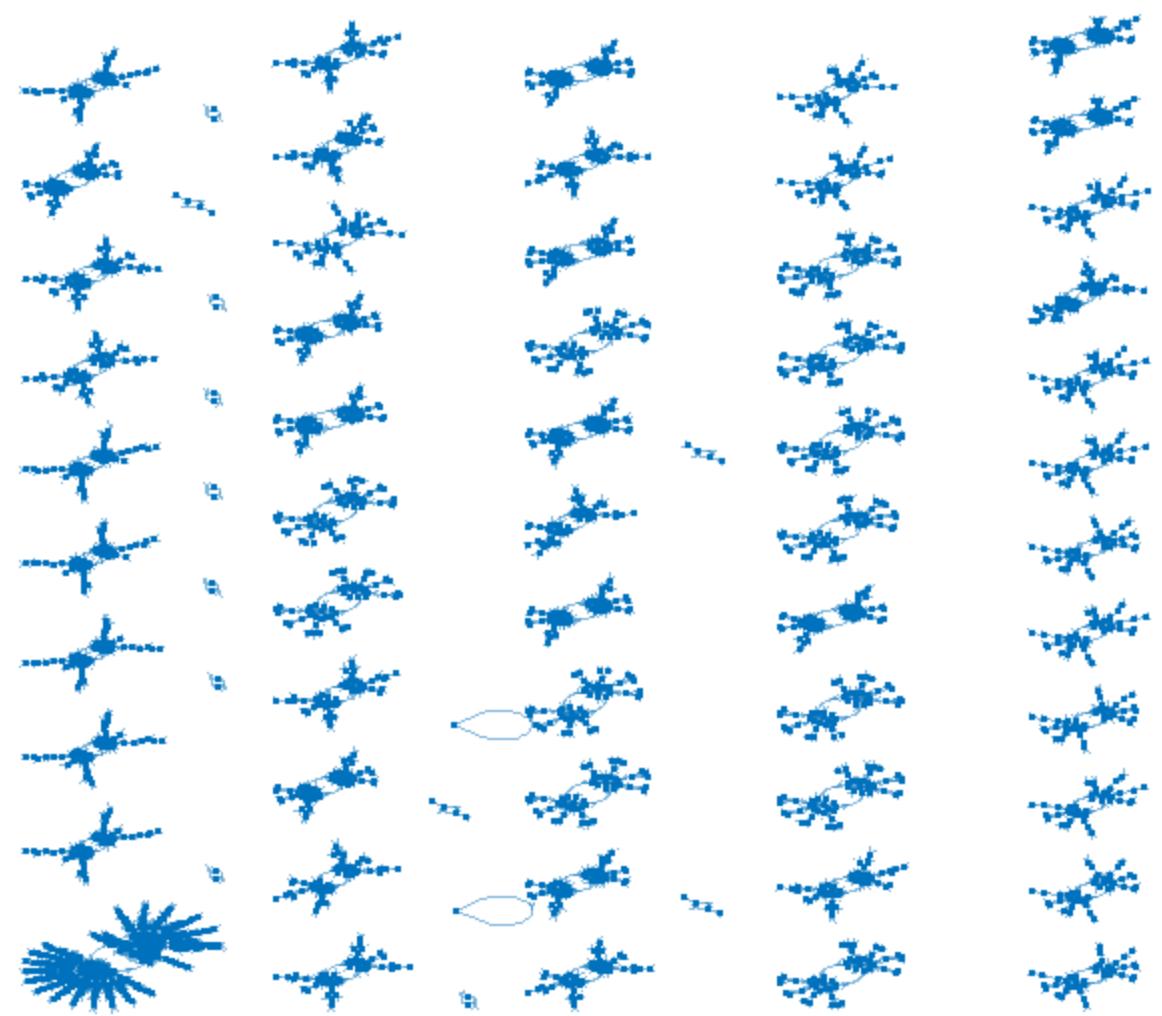


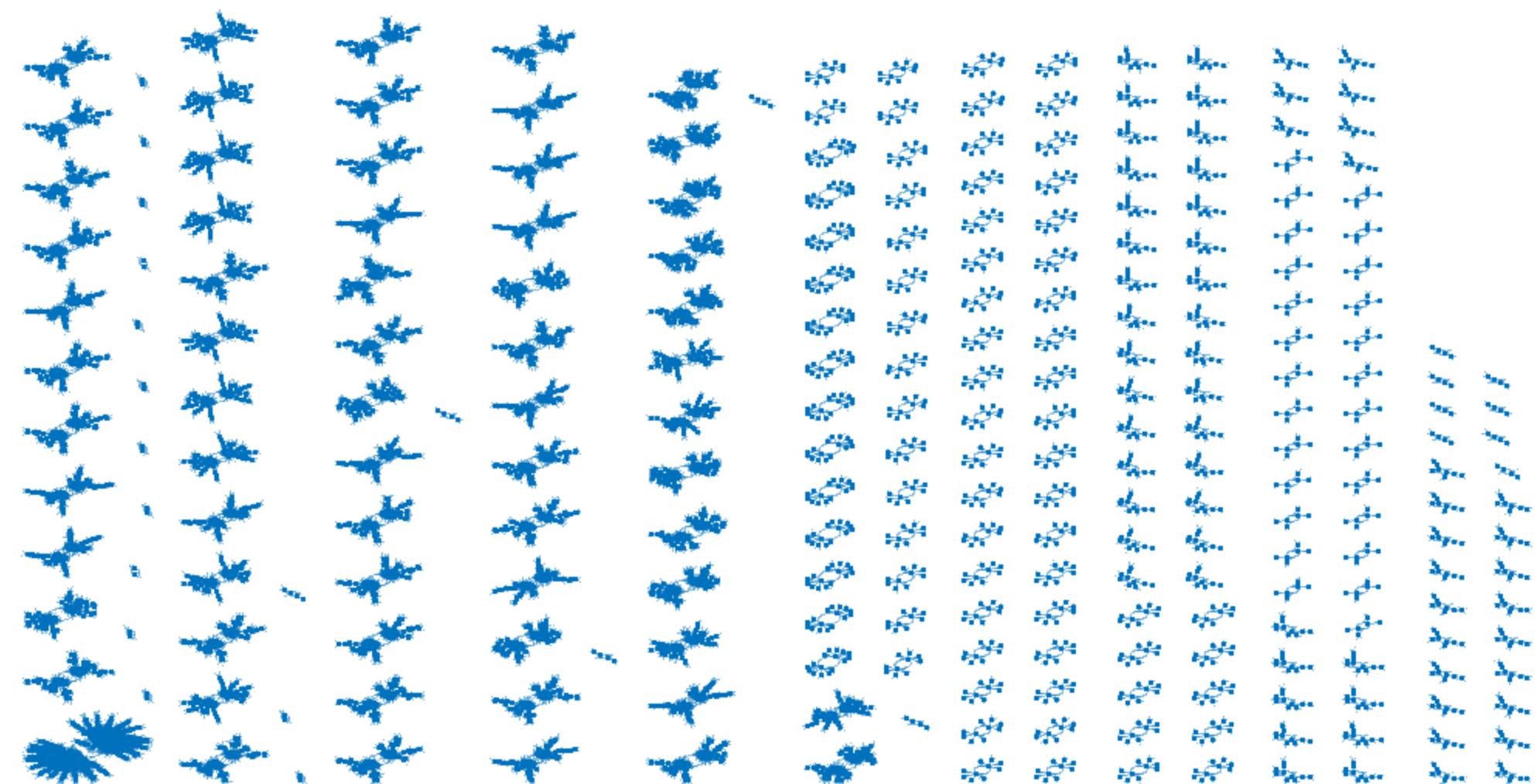


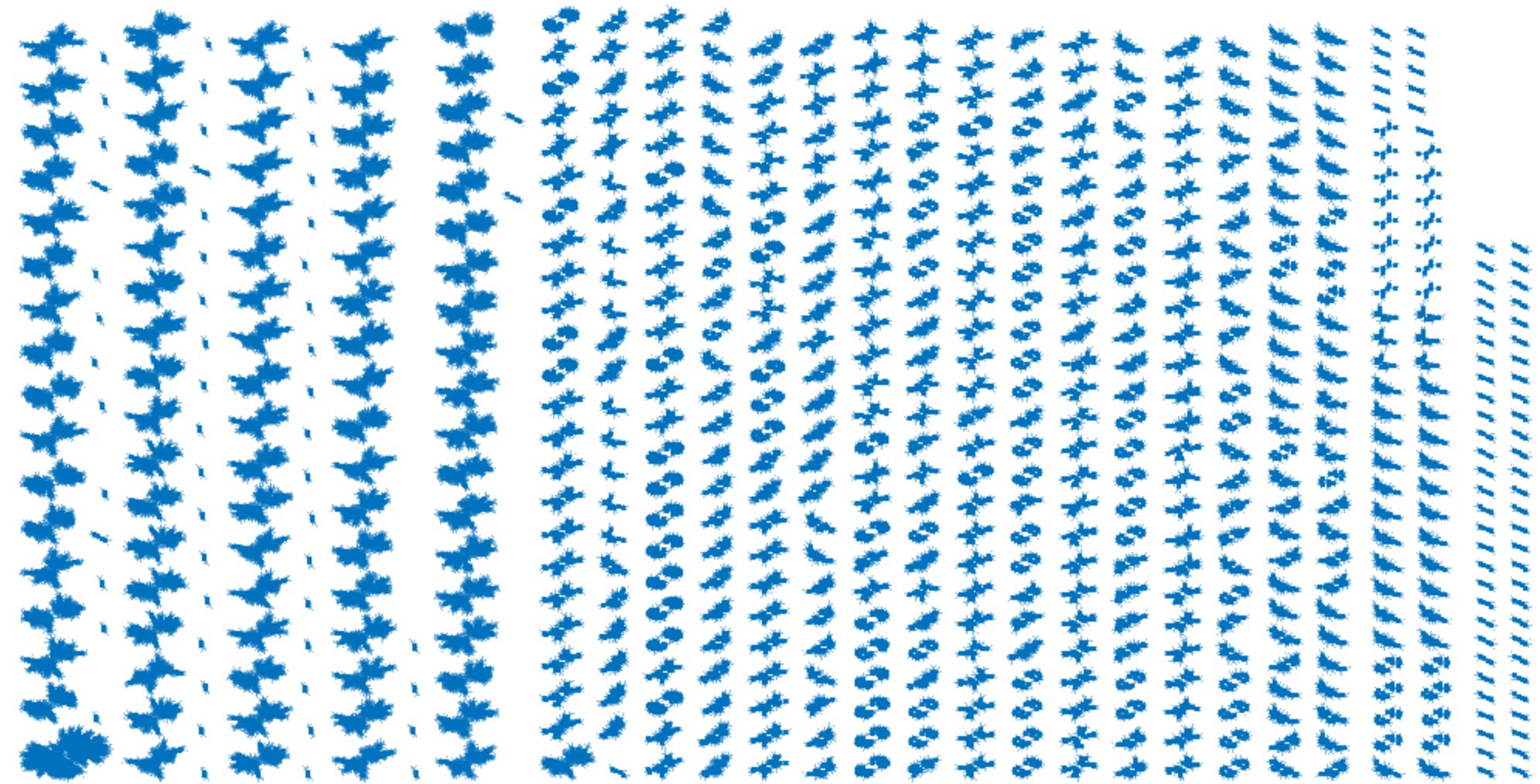


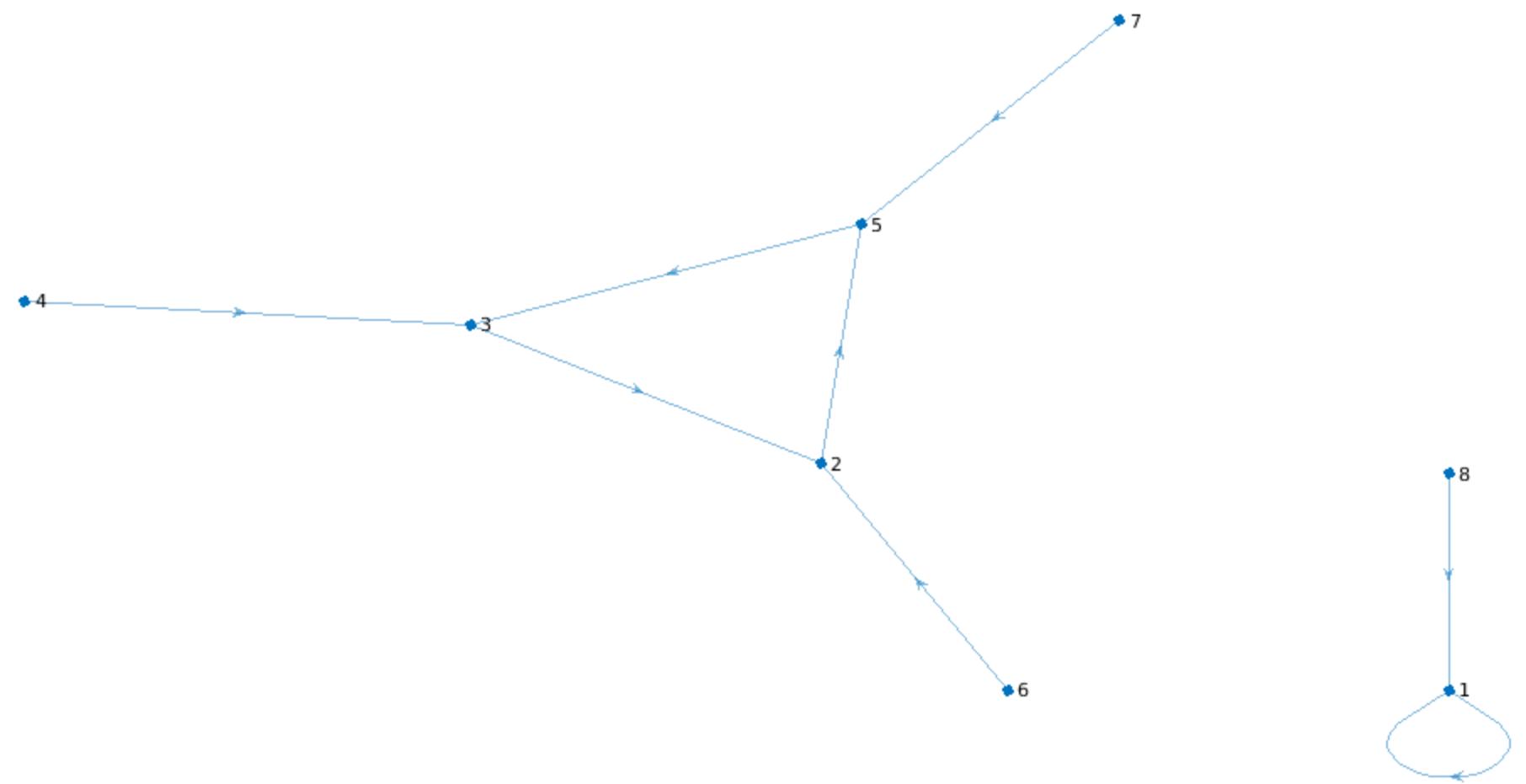
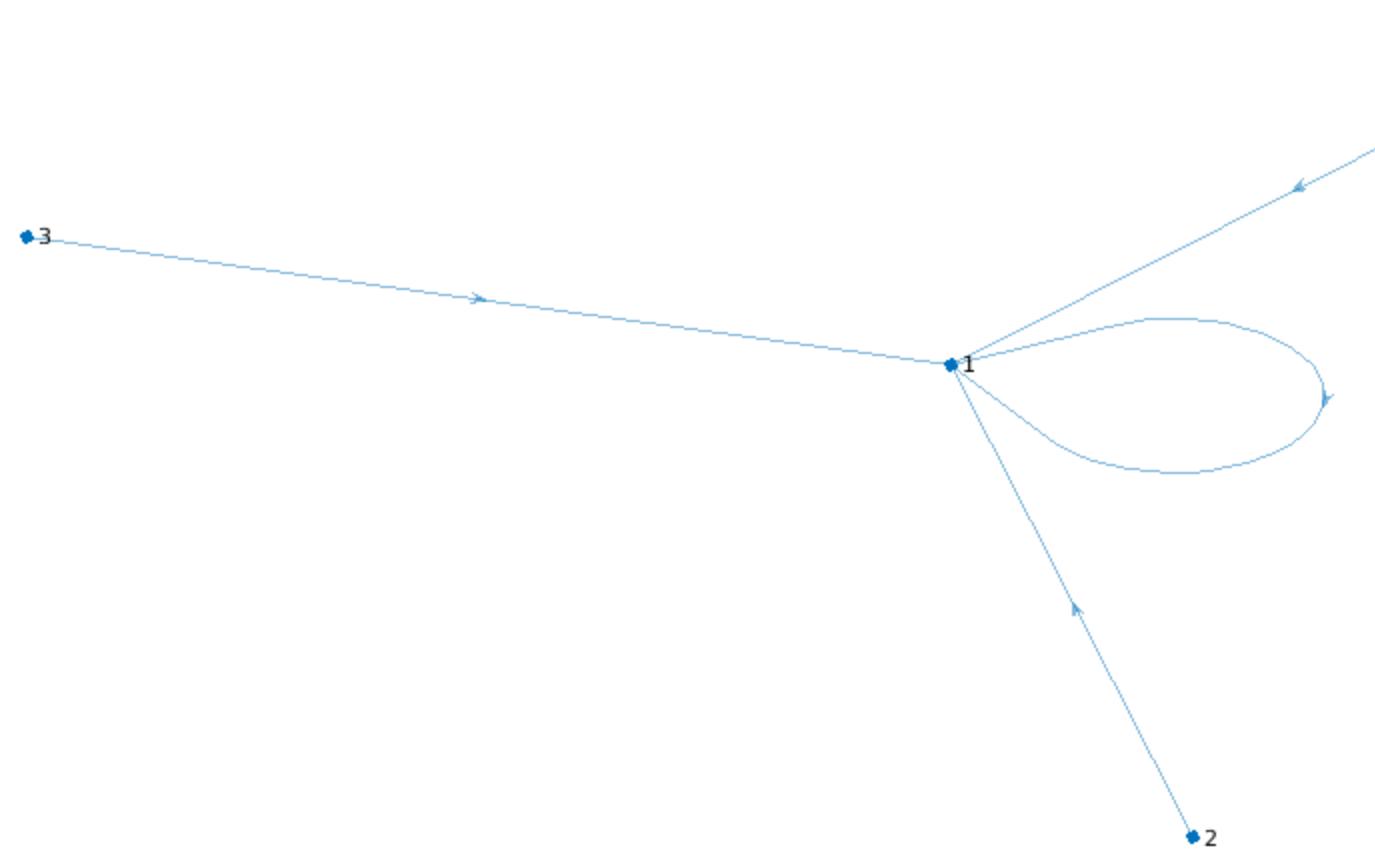


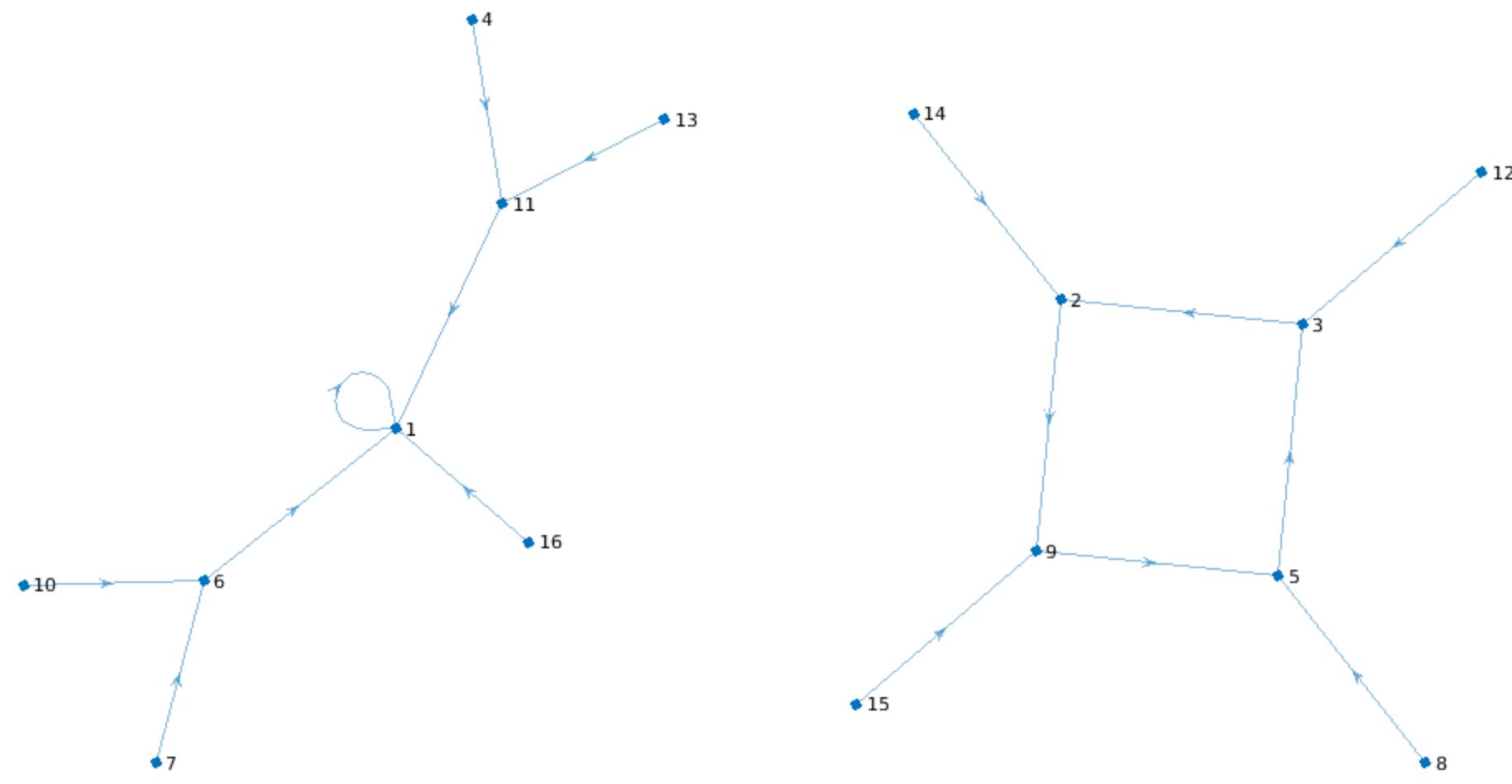


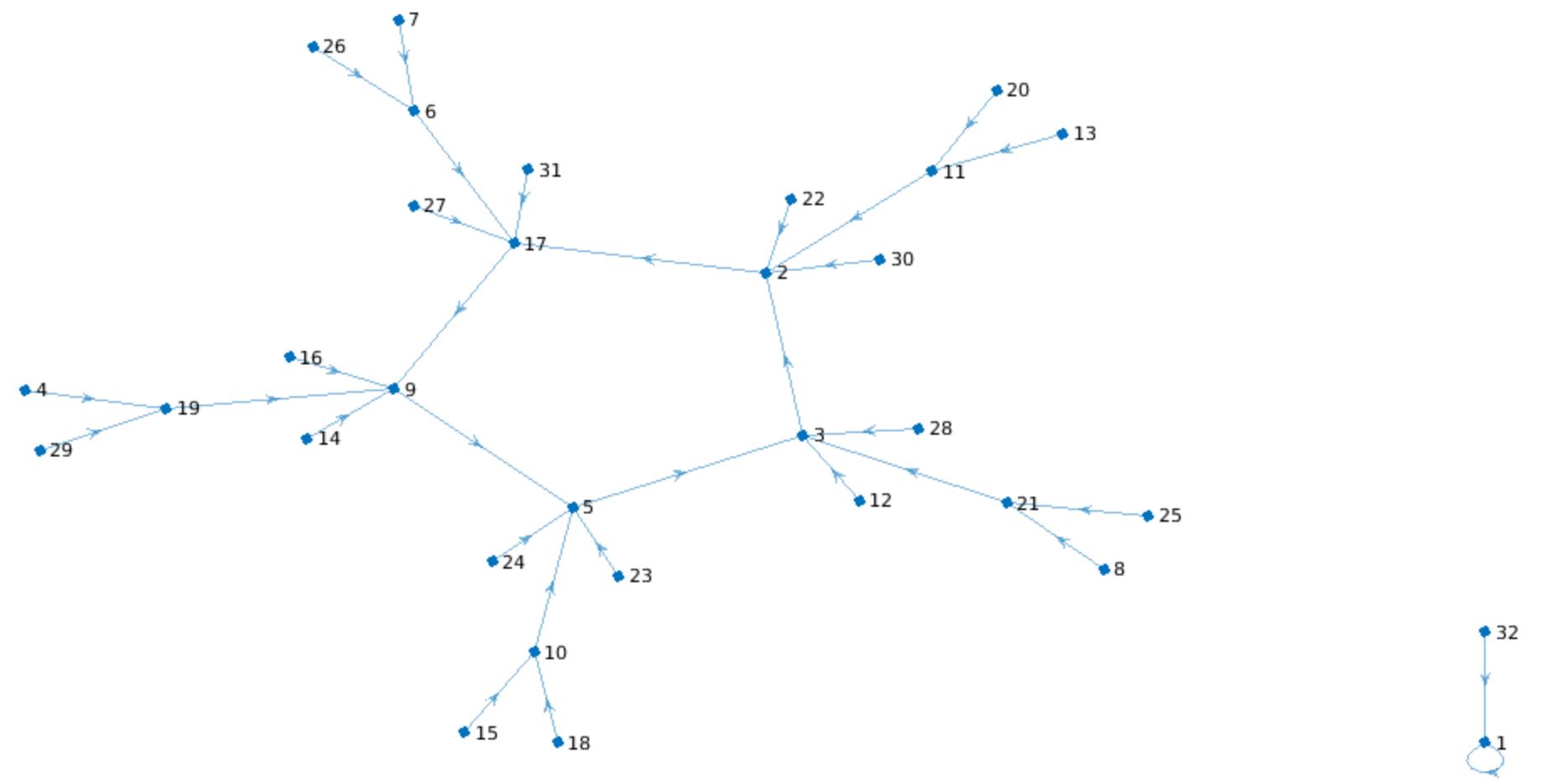


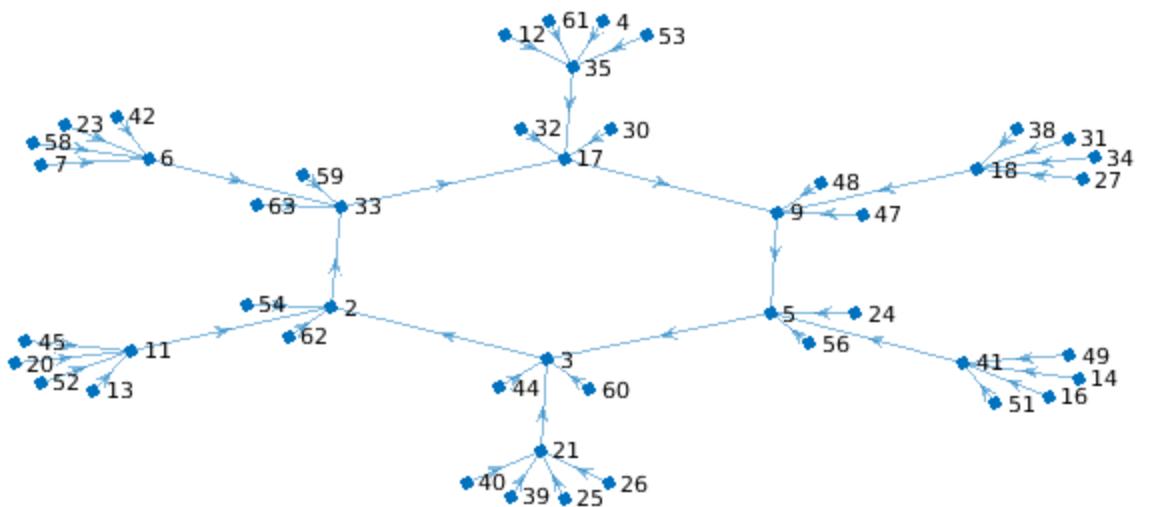
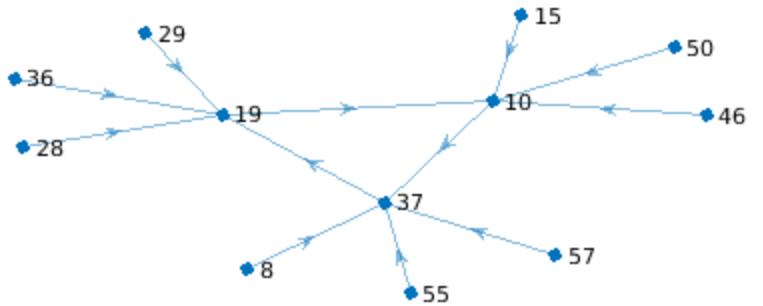


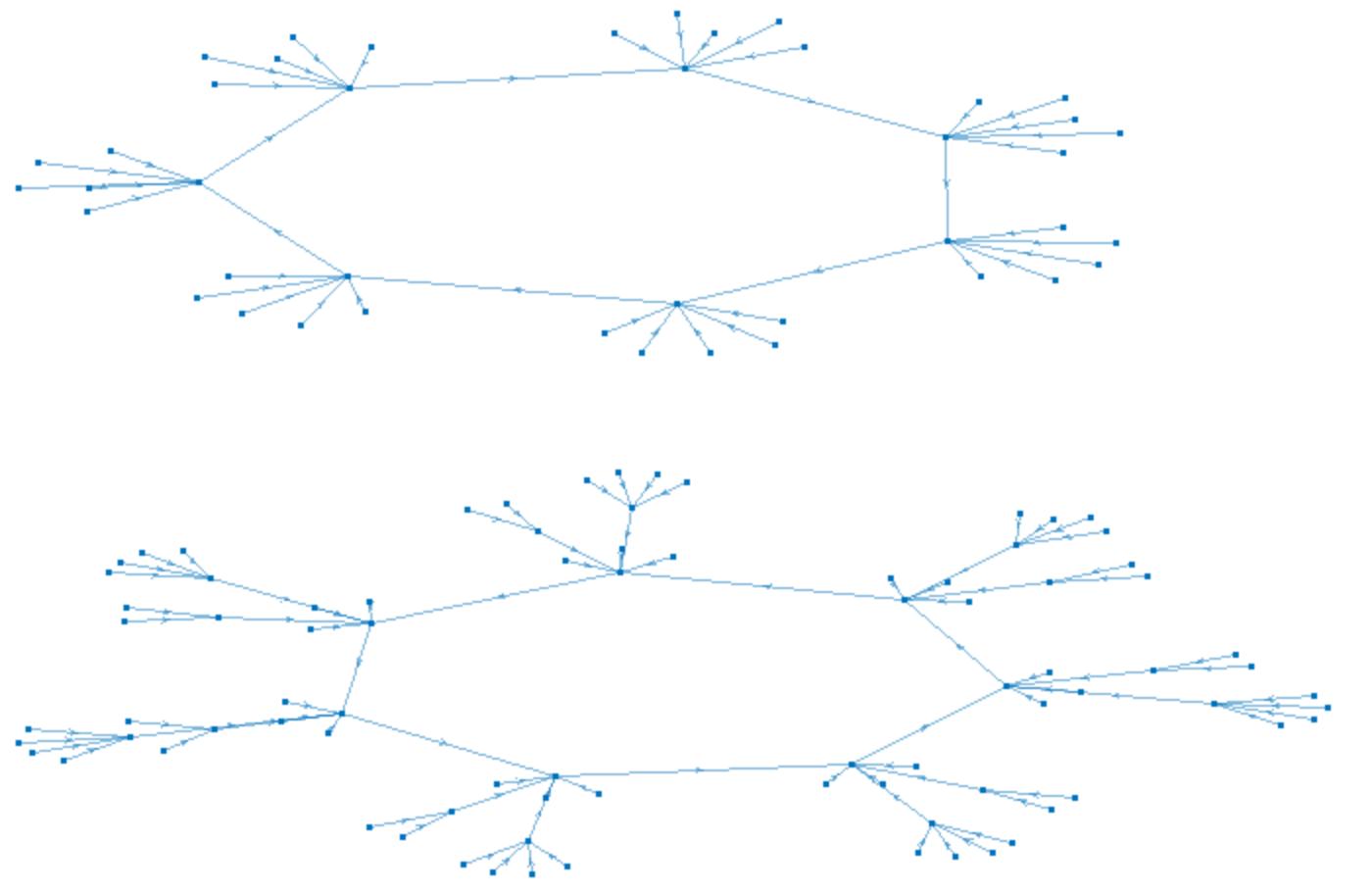


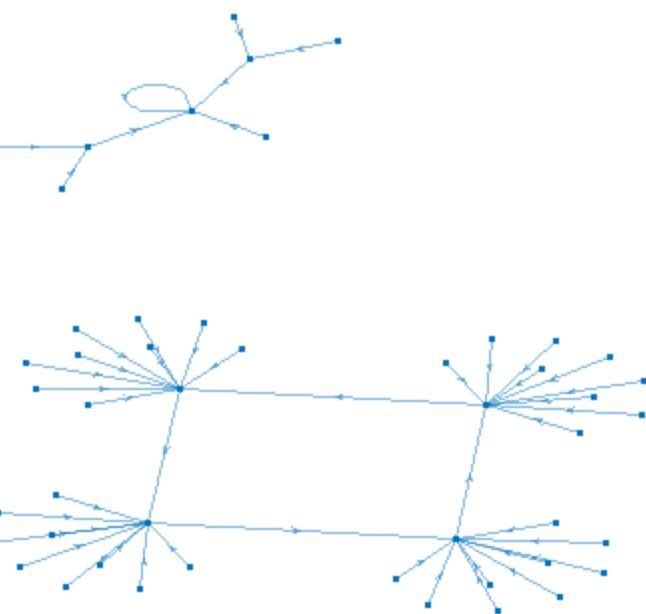
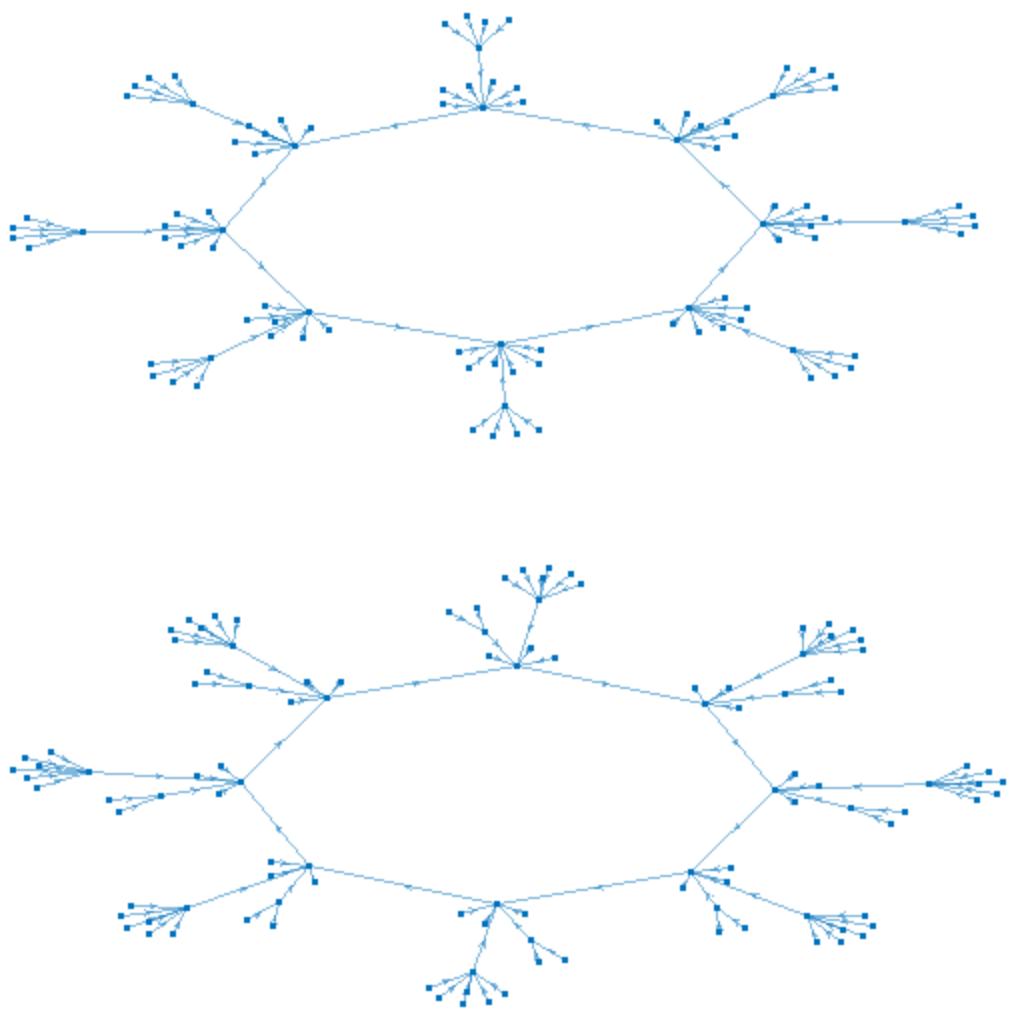


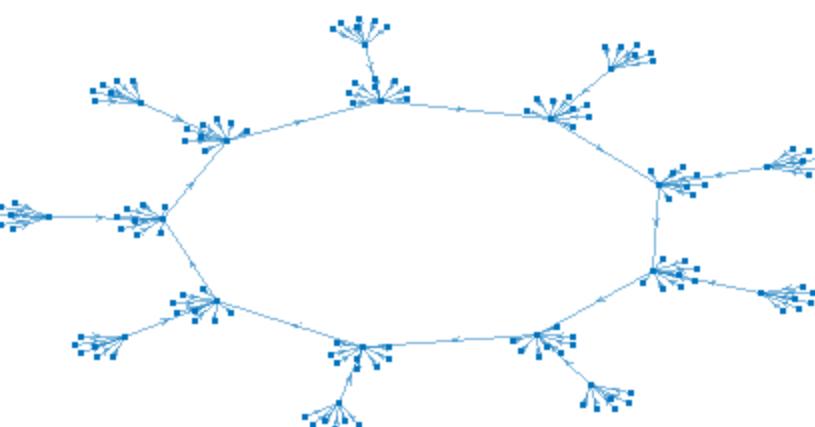
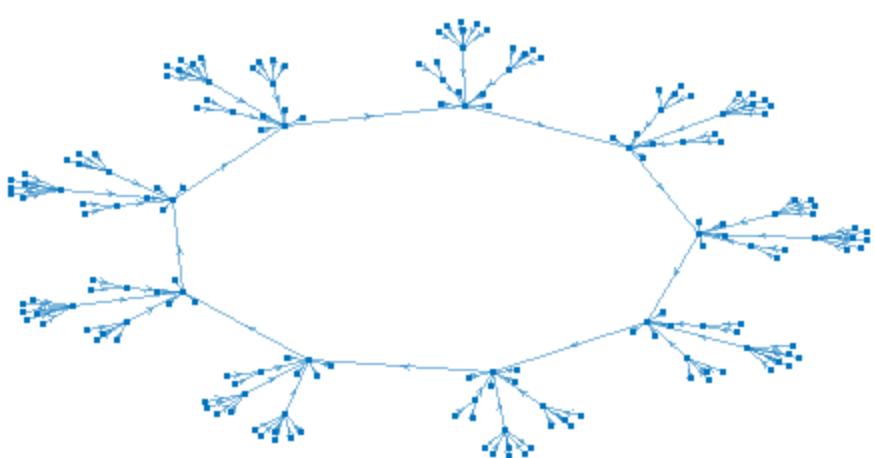
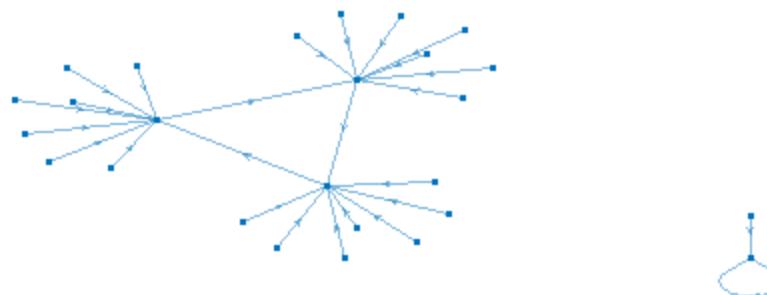
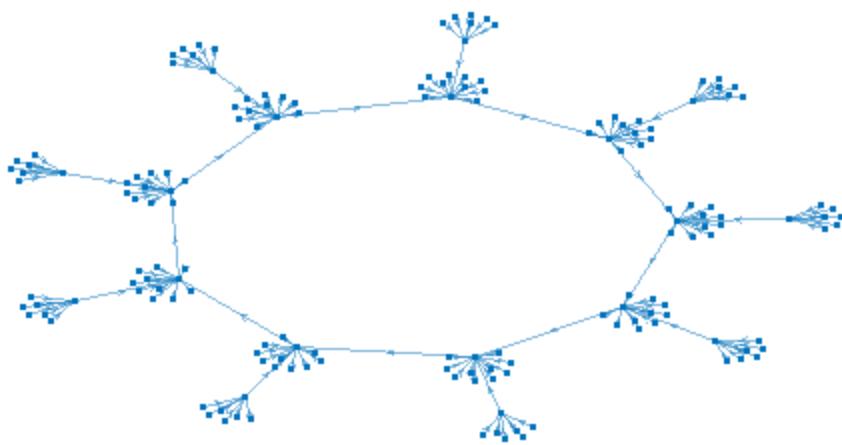


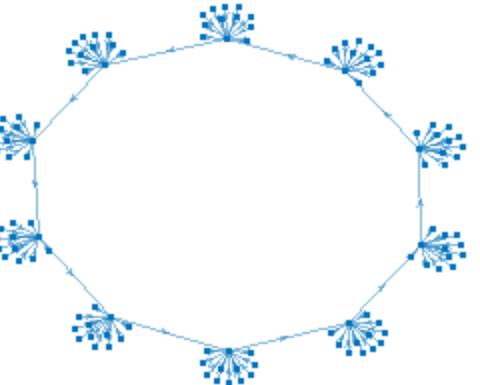
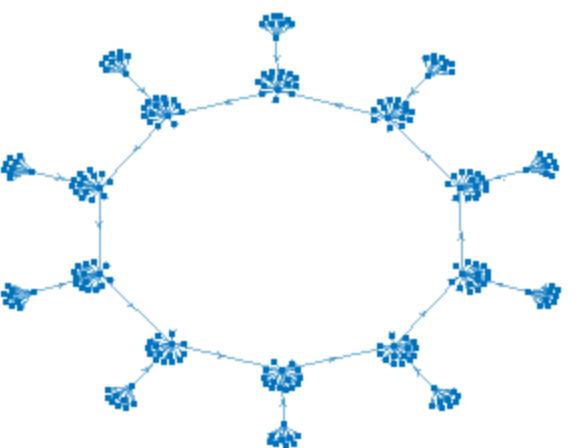
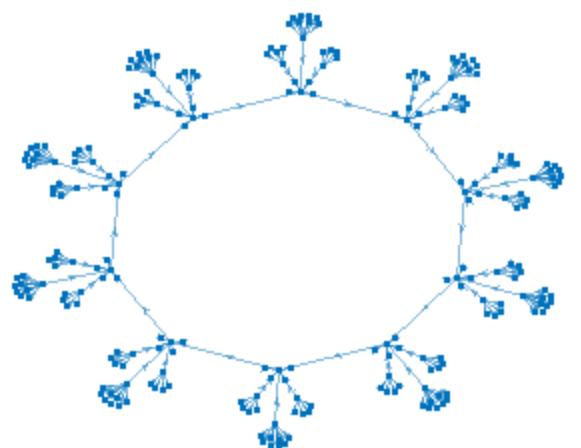
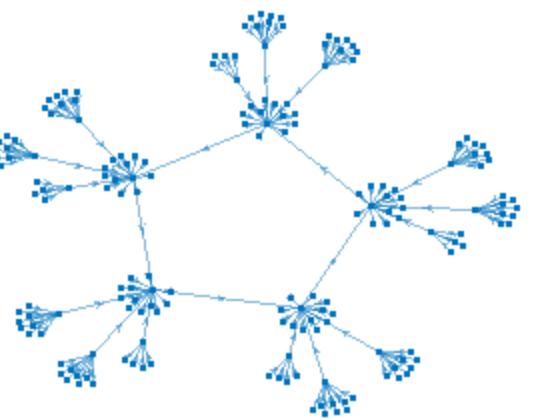
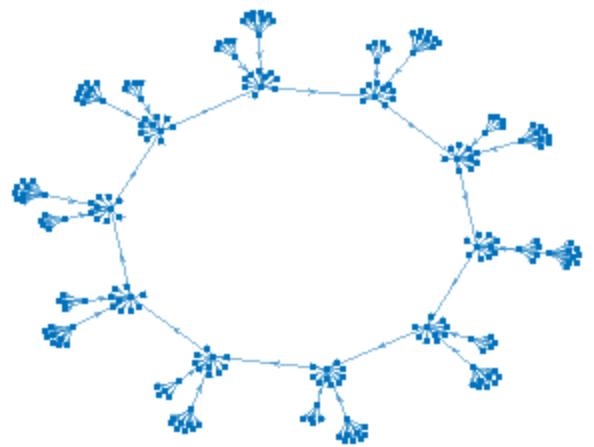


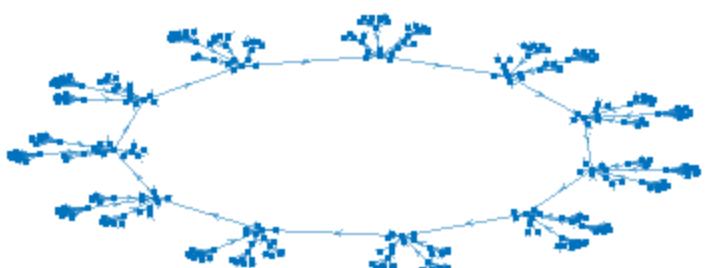
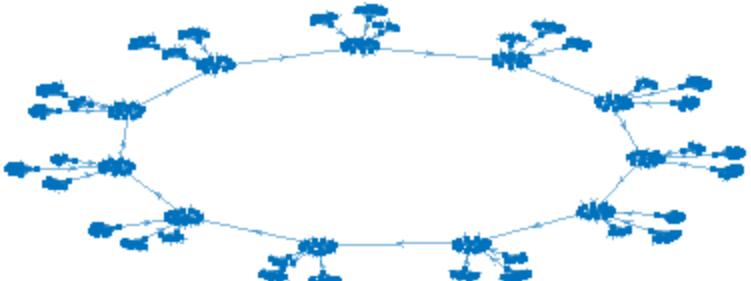
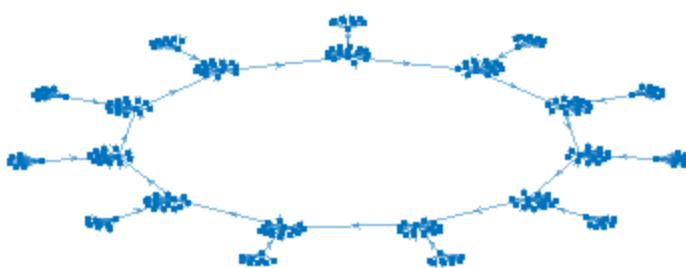
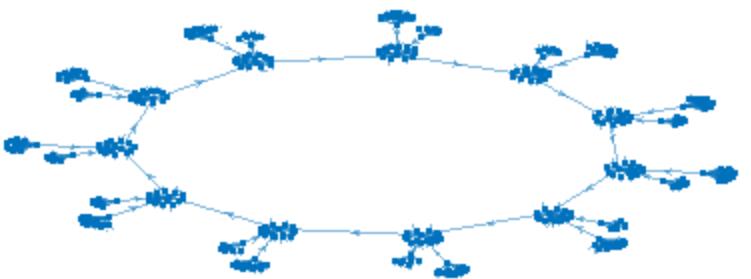
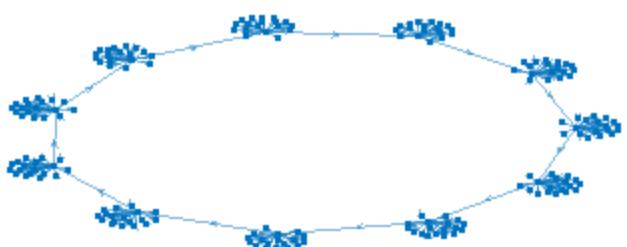
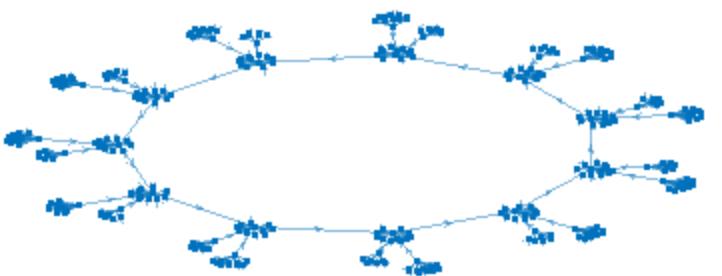


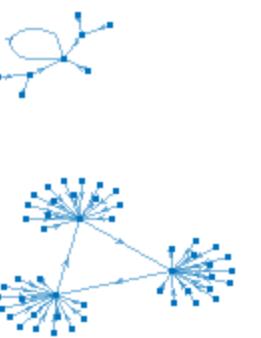
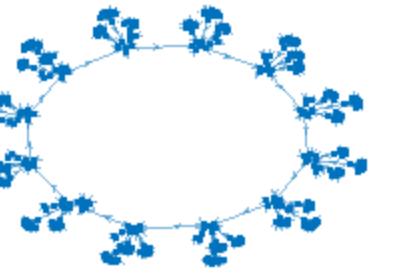
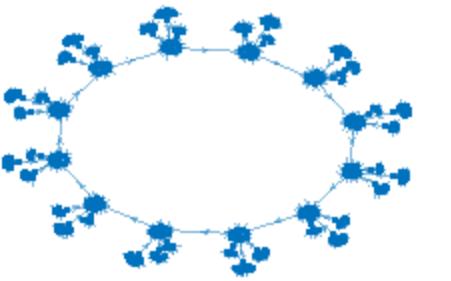
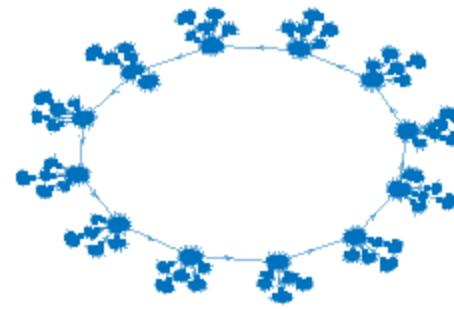
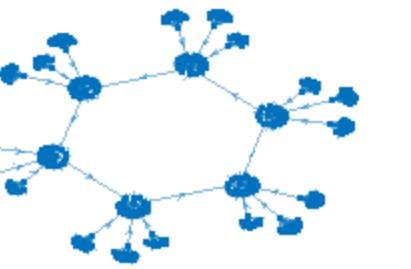
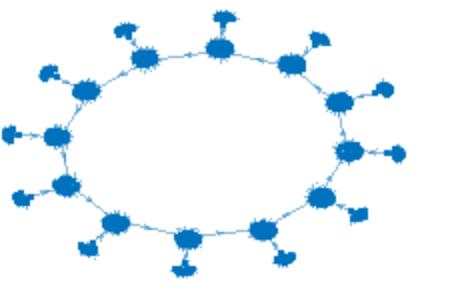
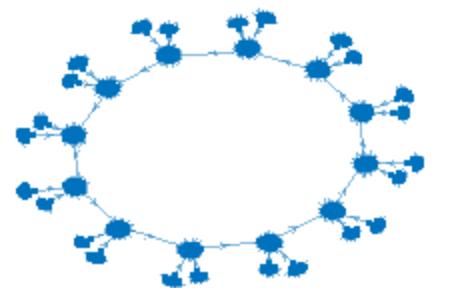
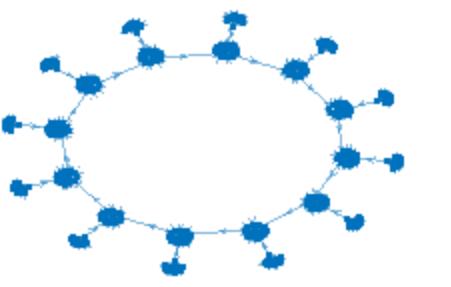
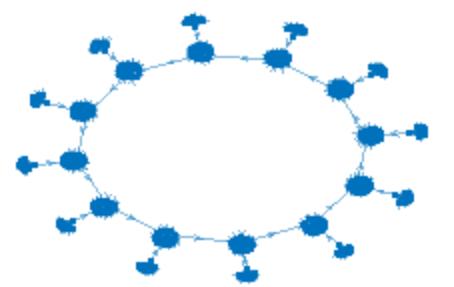


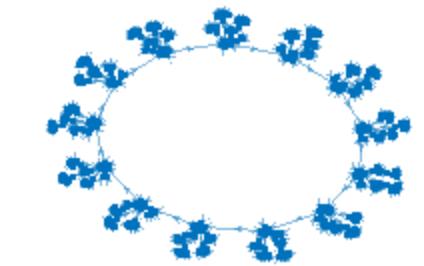
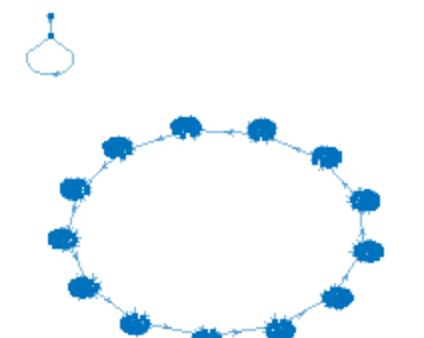
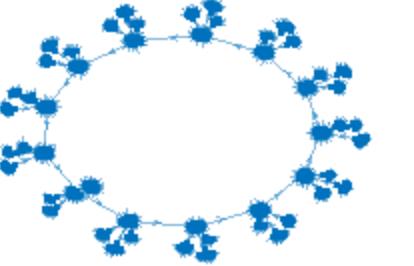
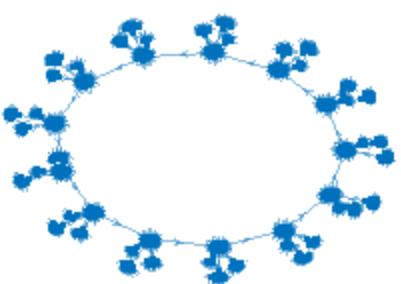
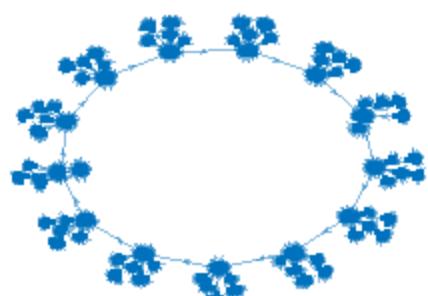
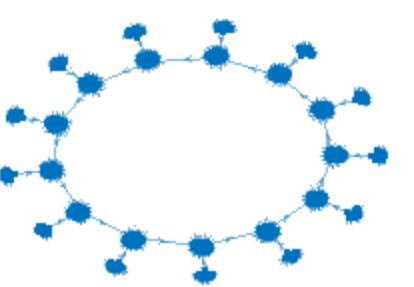
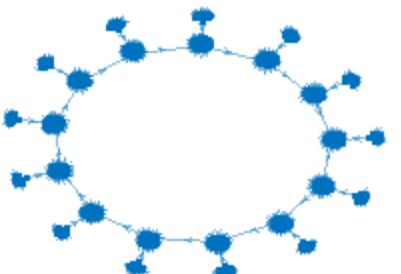
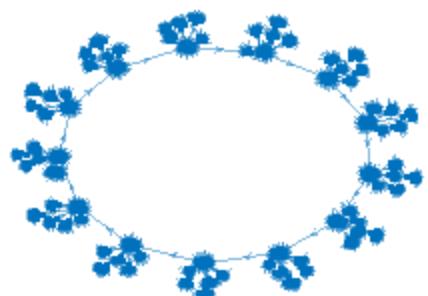
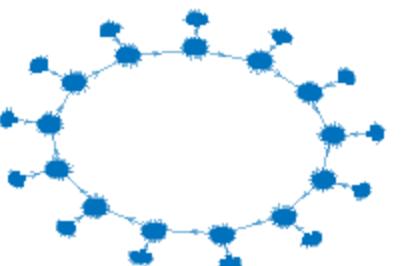
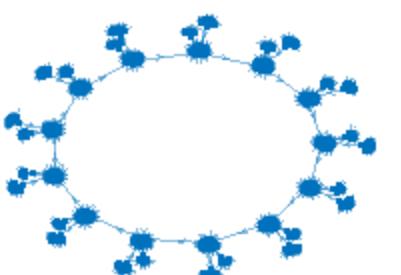
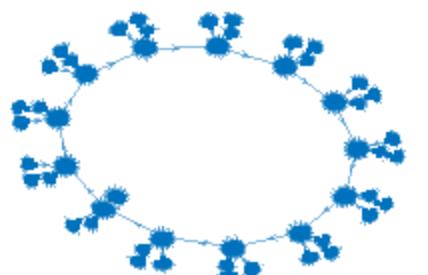


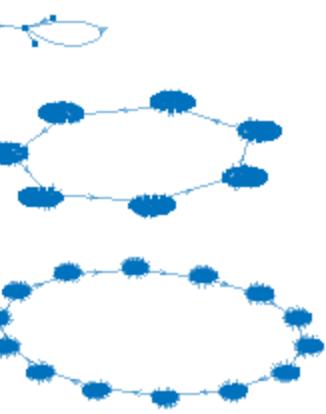
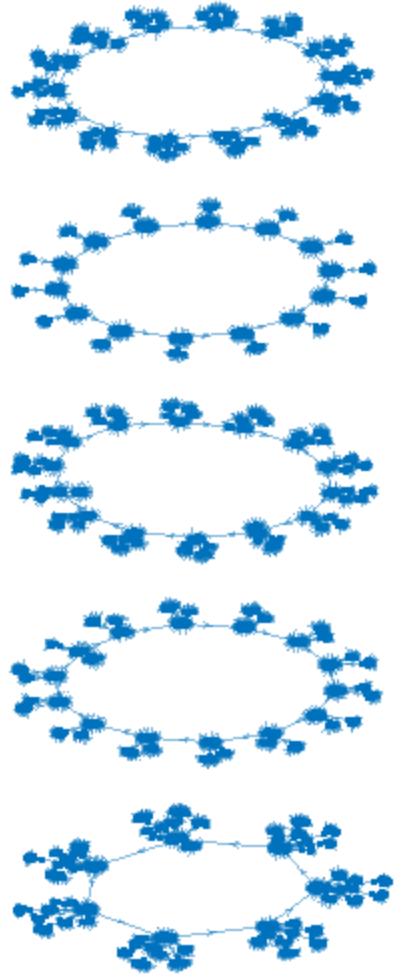
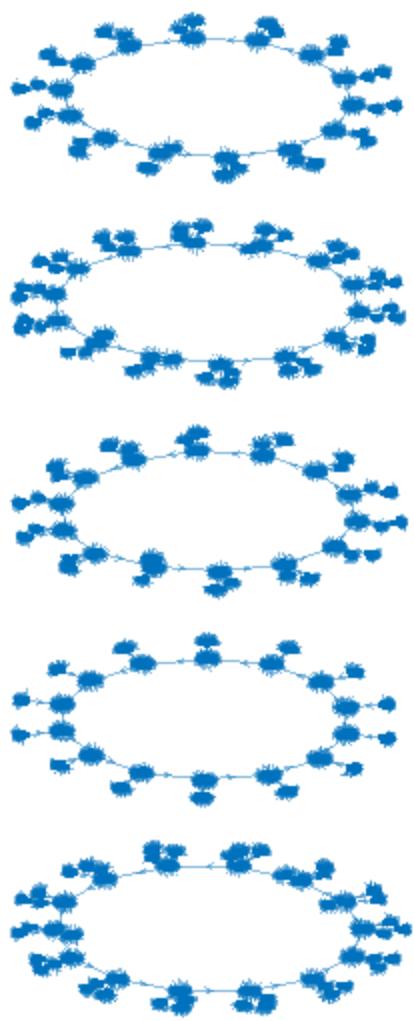
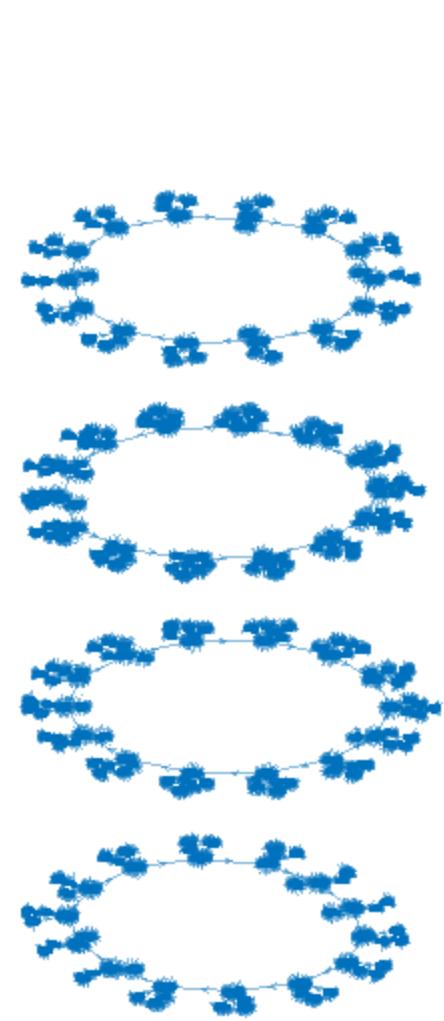


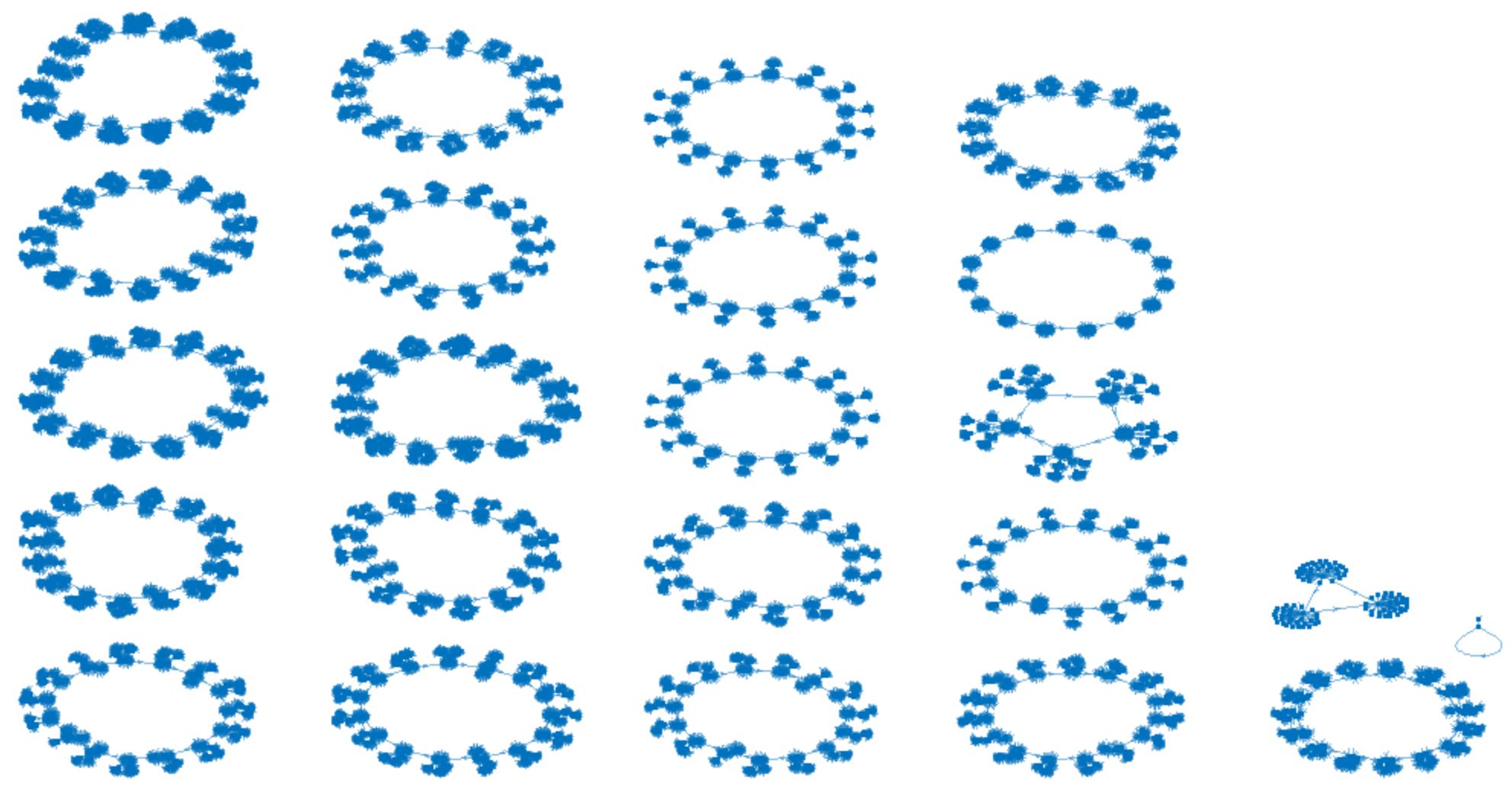




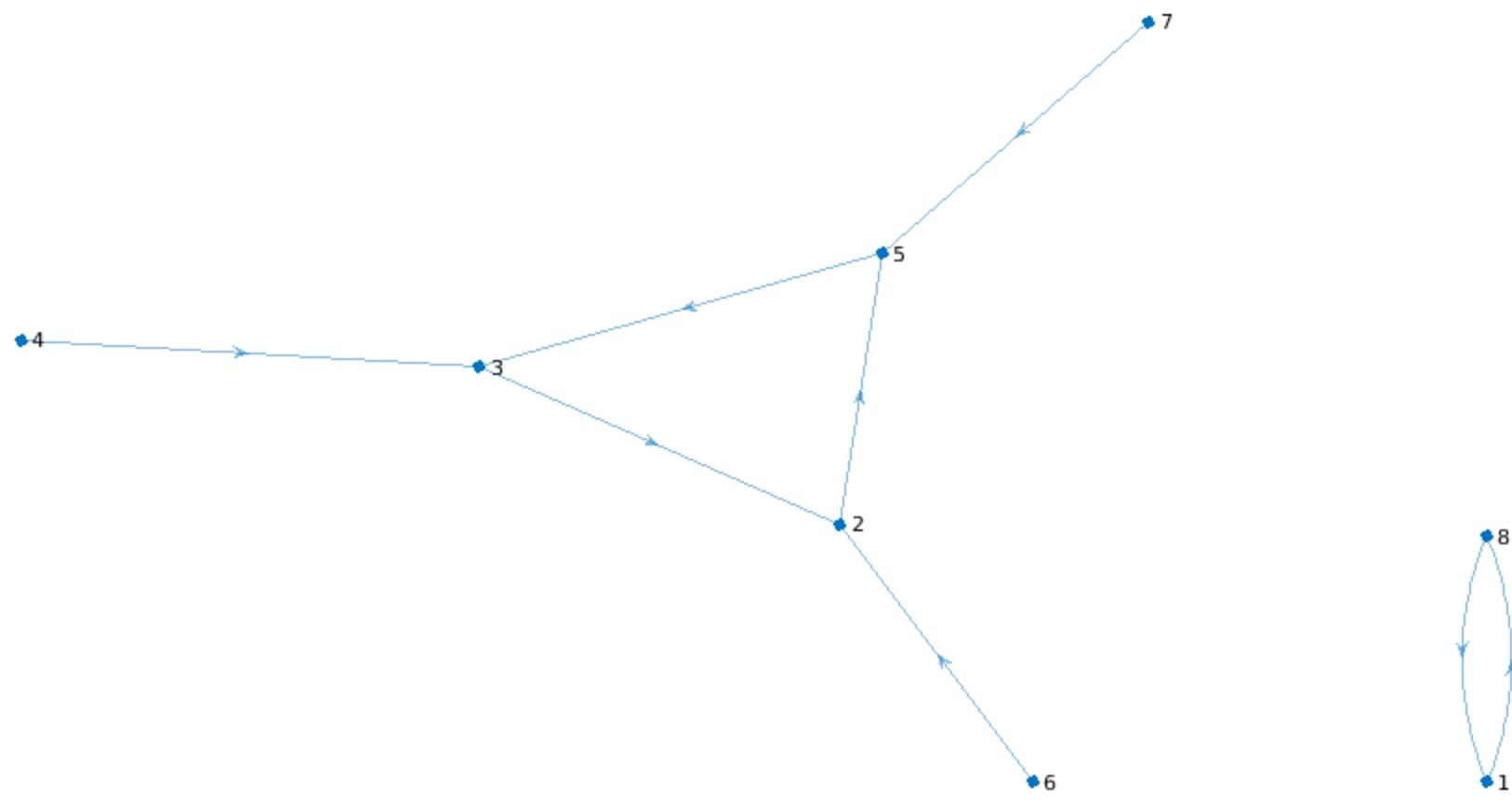
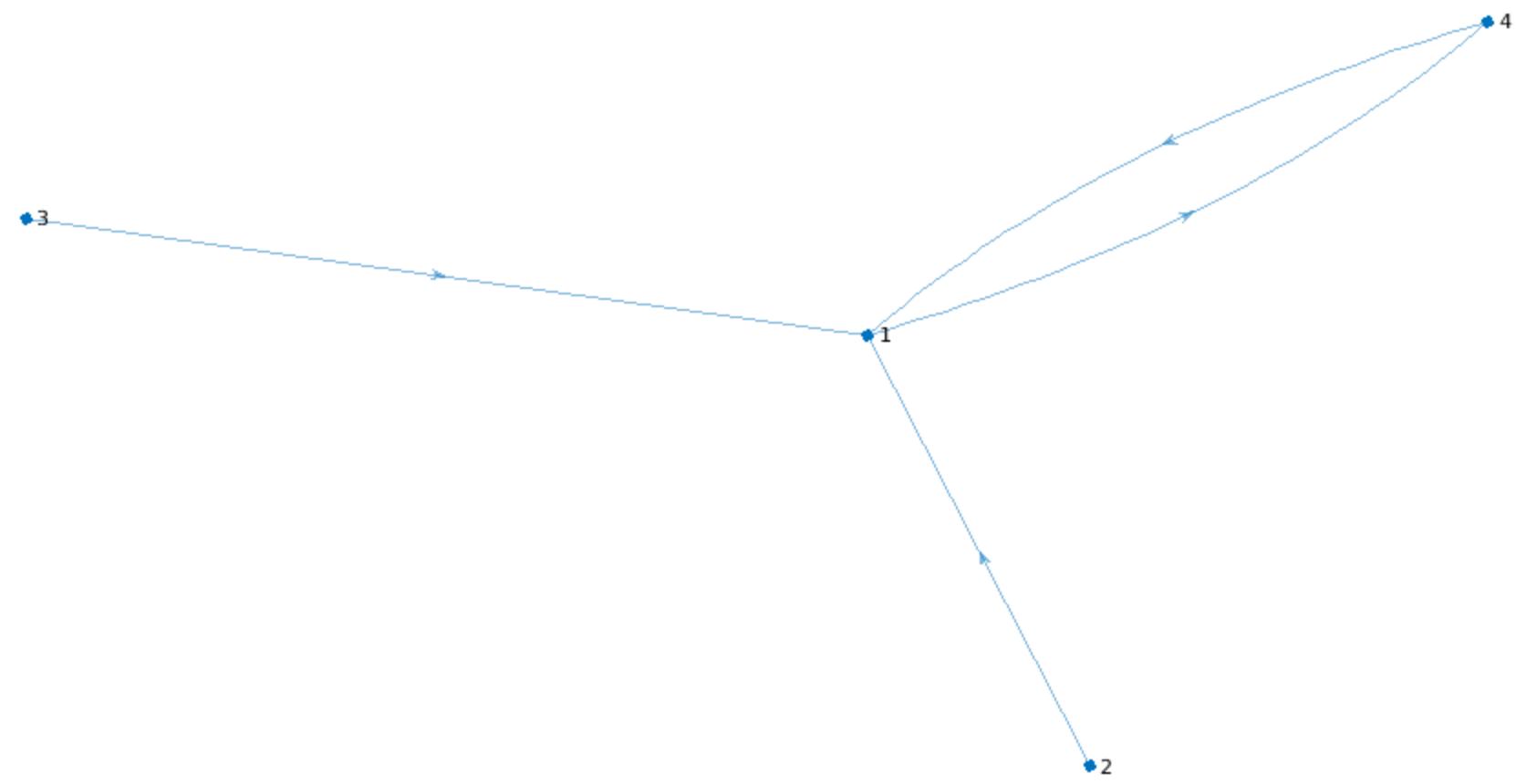


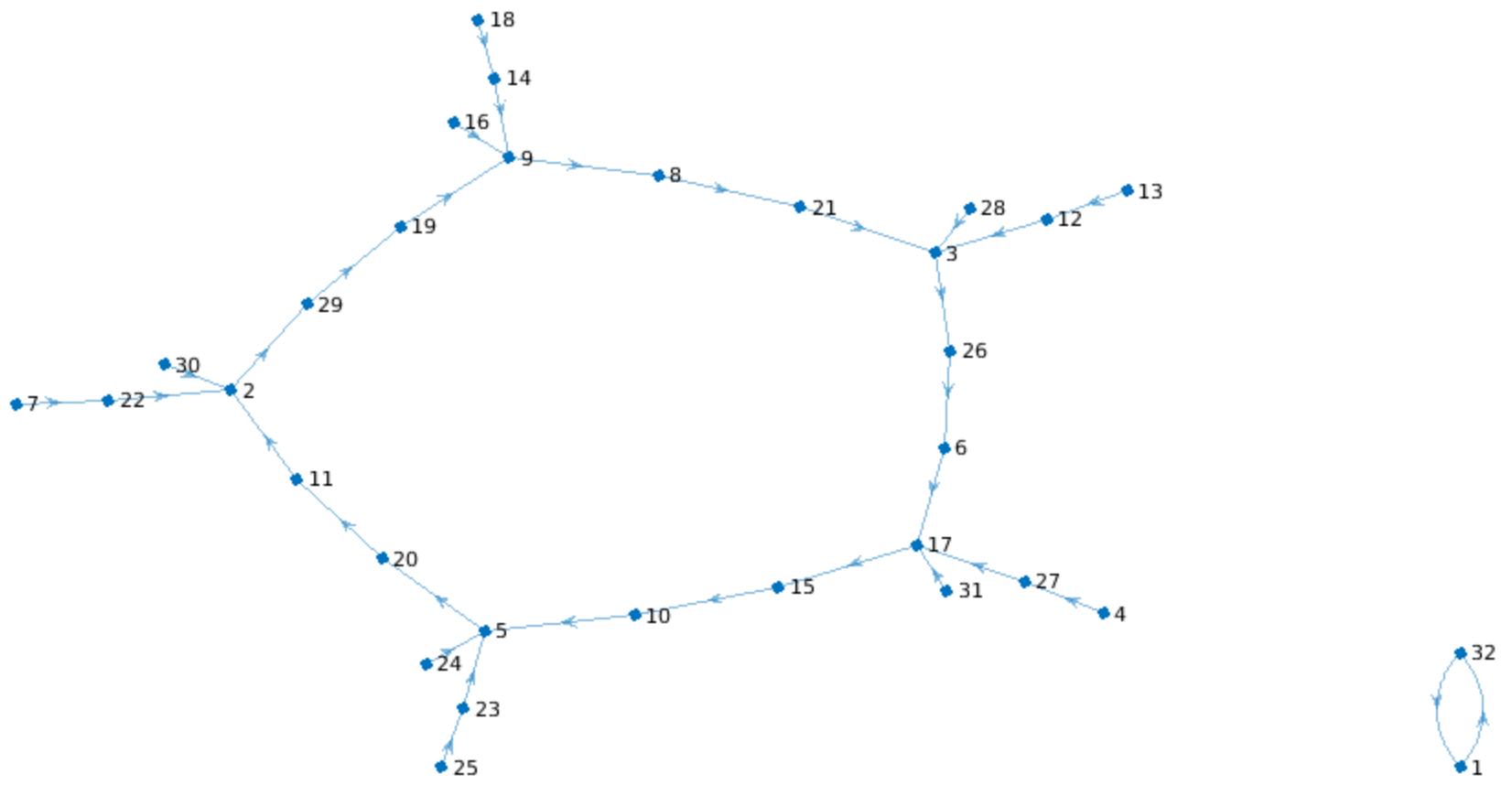


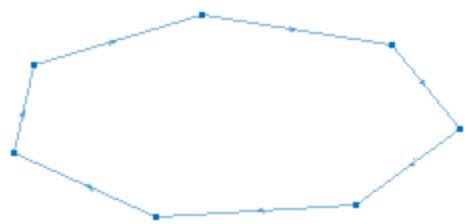
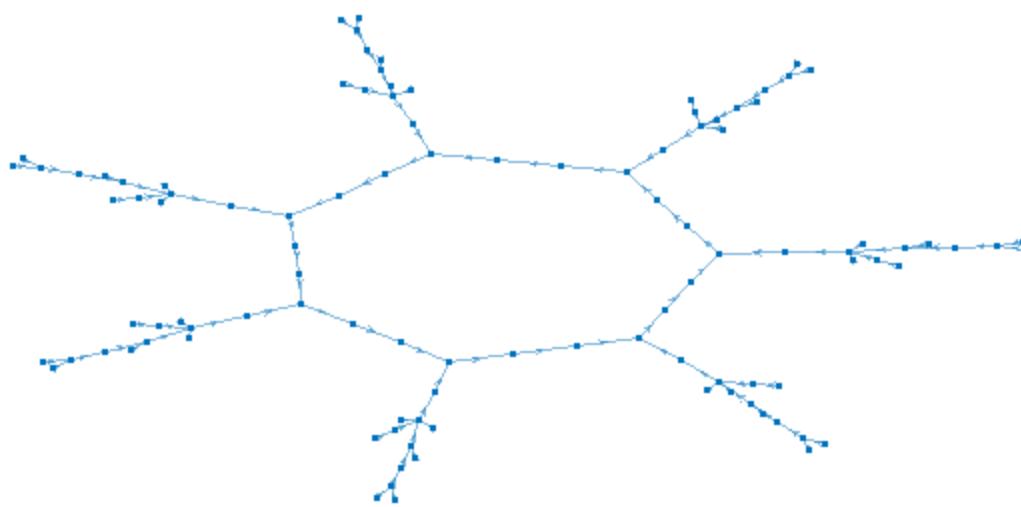
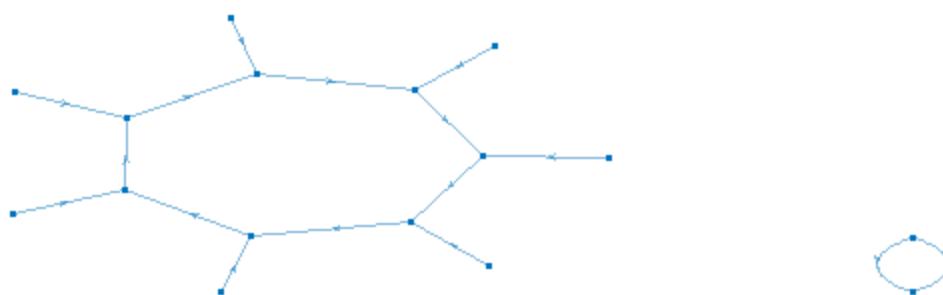
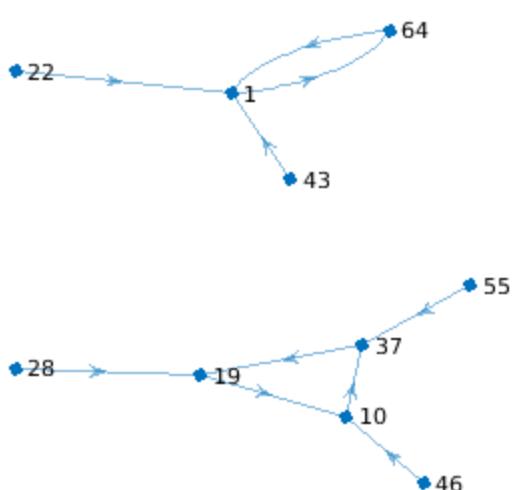
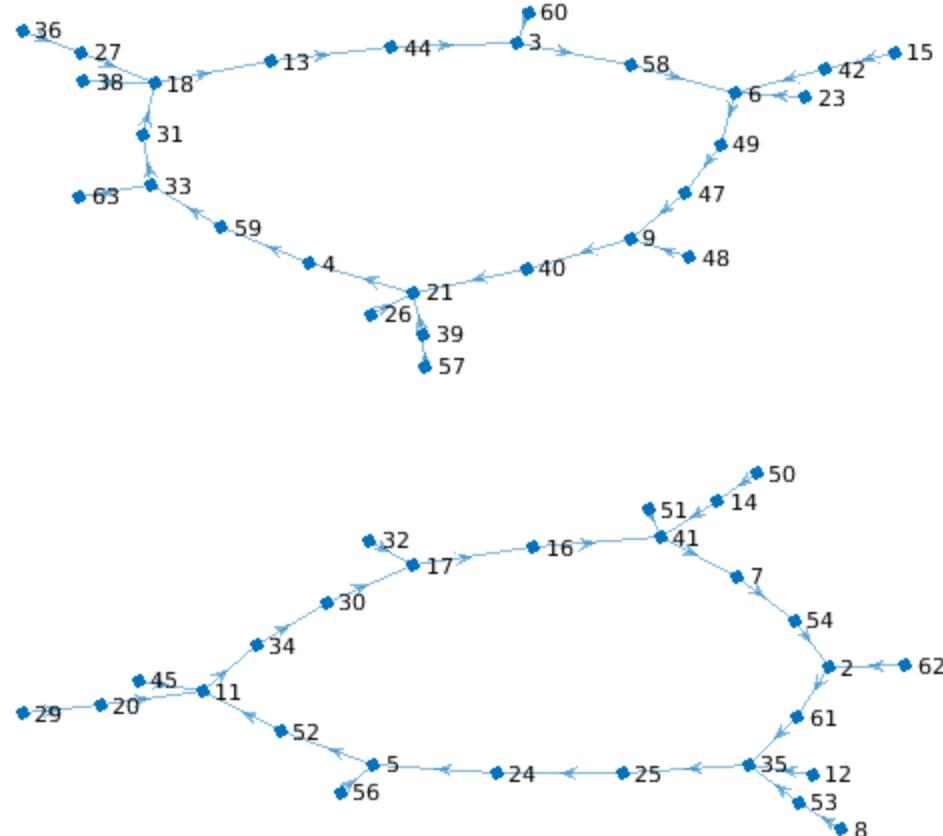


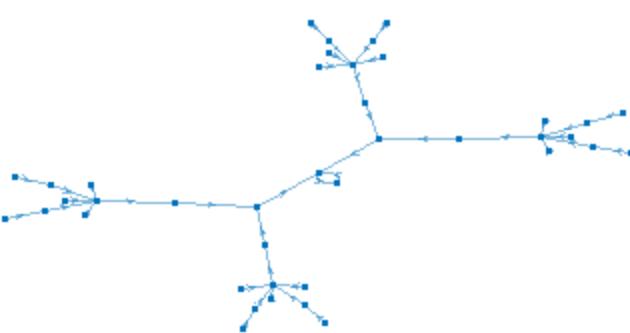
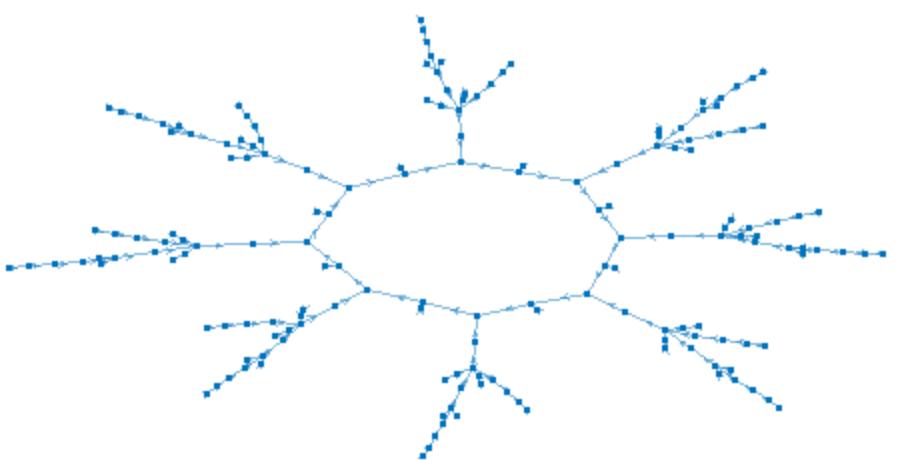
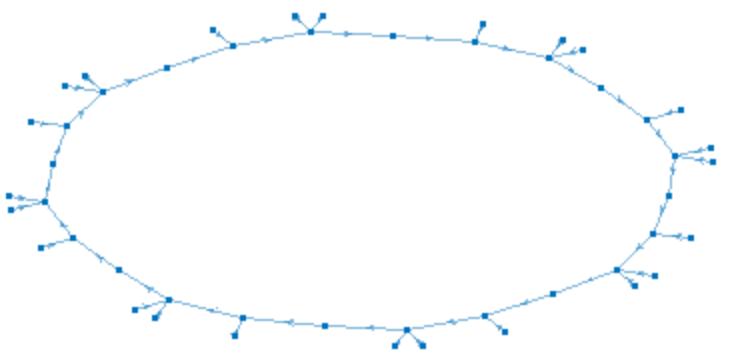


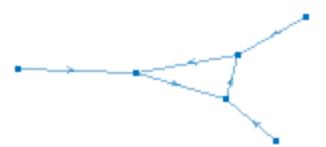
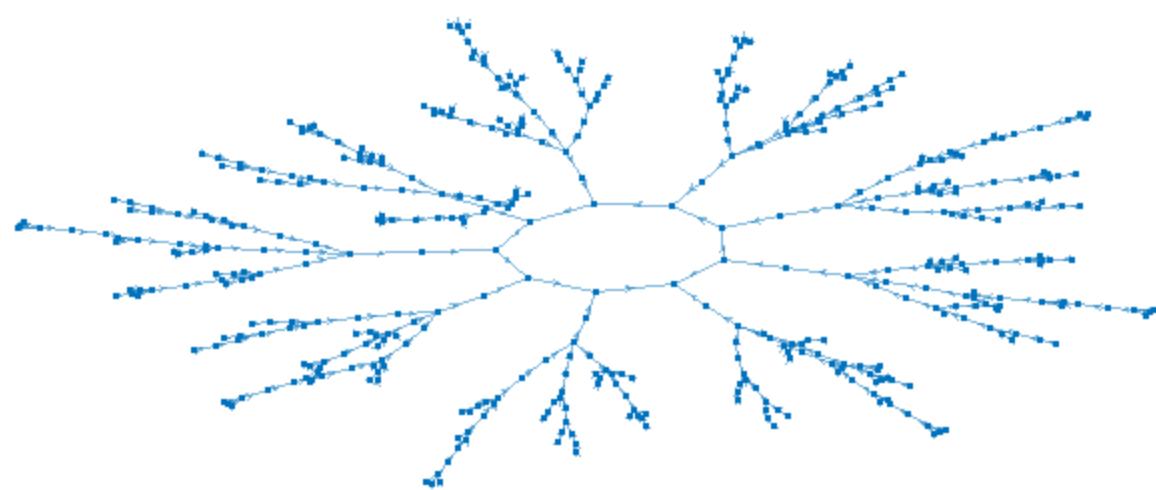
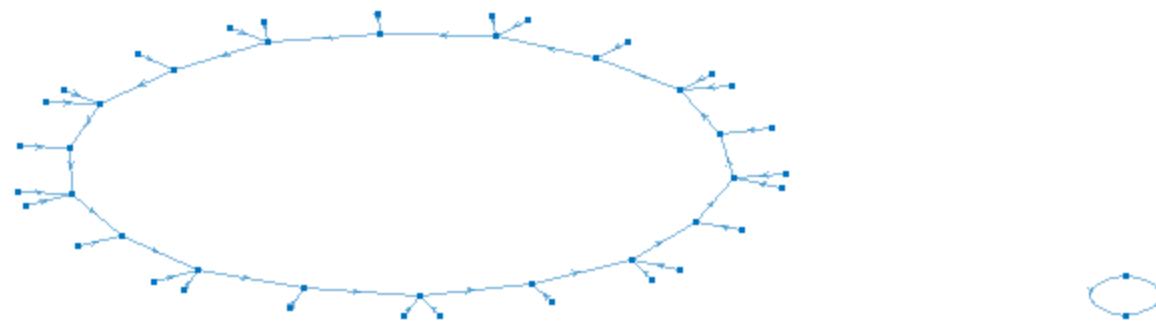
25

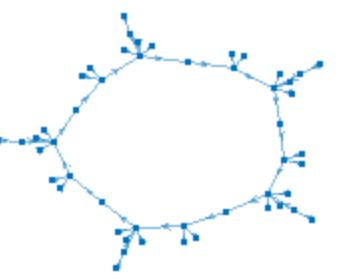
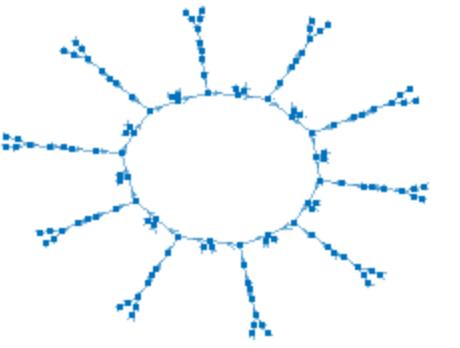
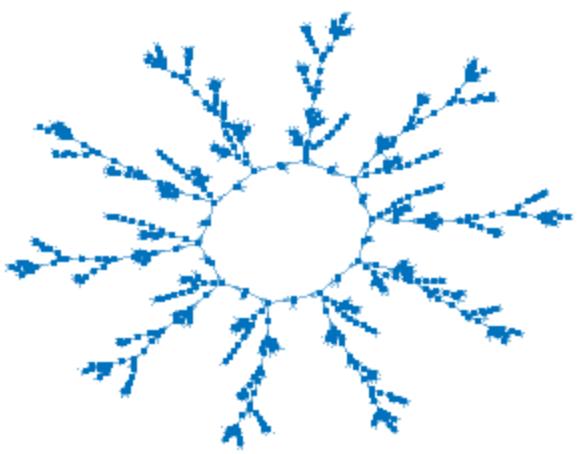
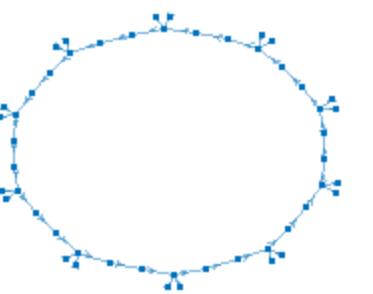
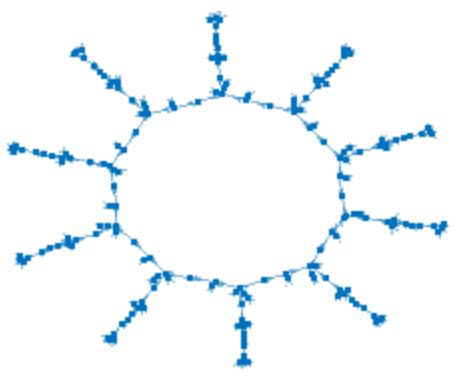


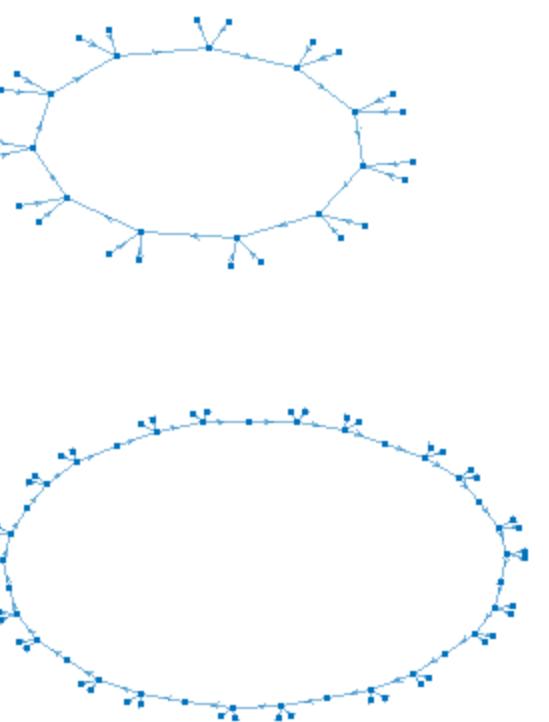
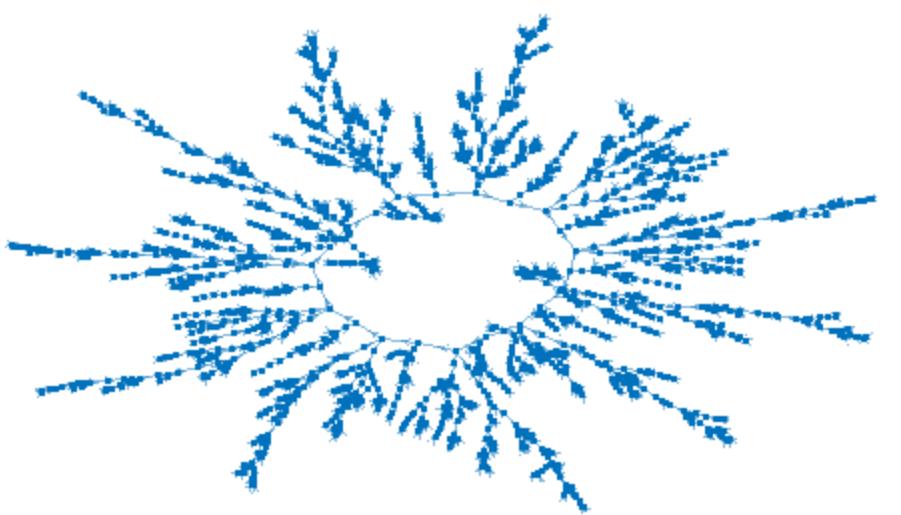
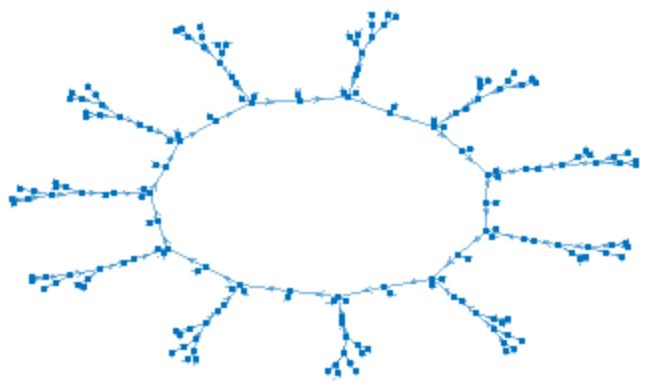


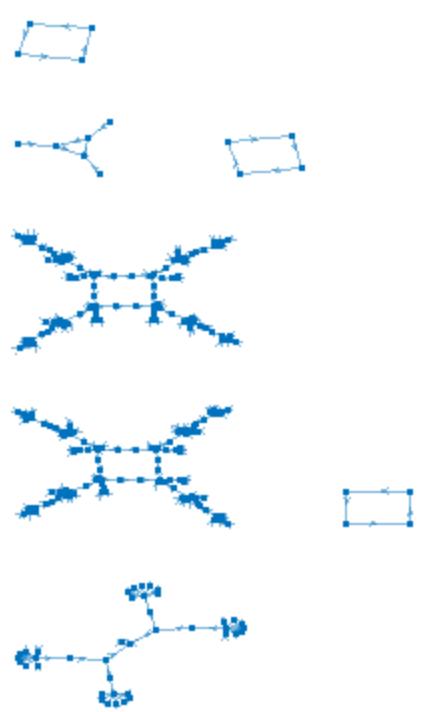
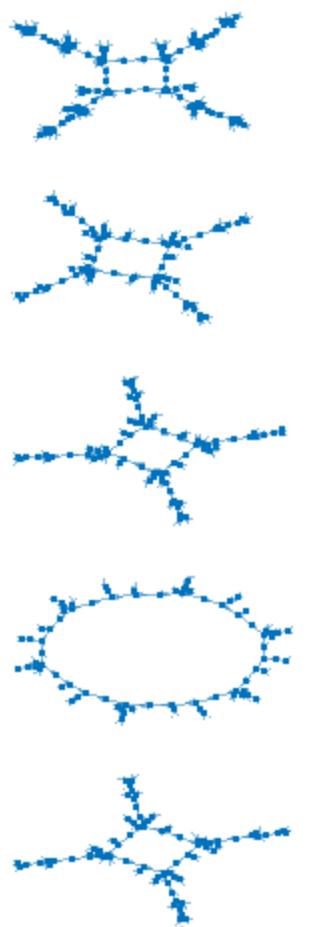
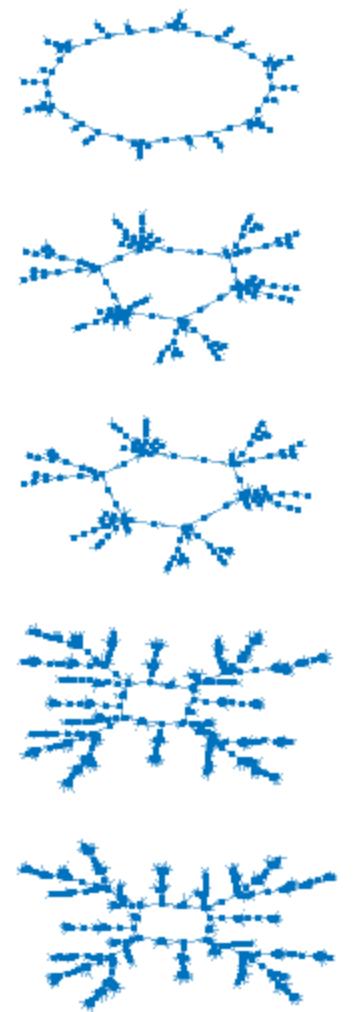
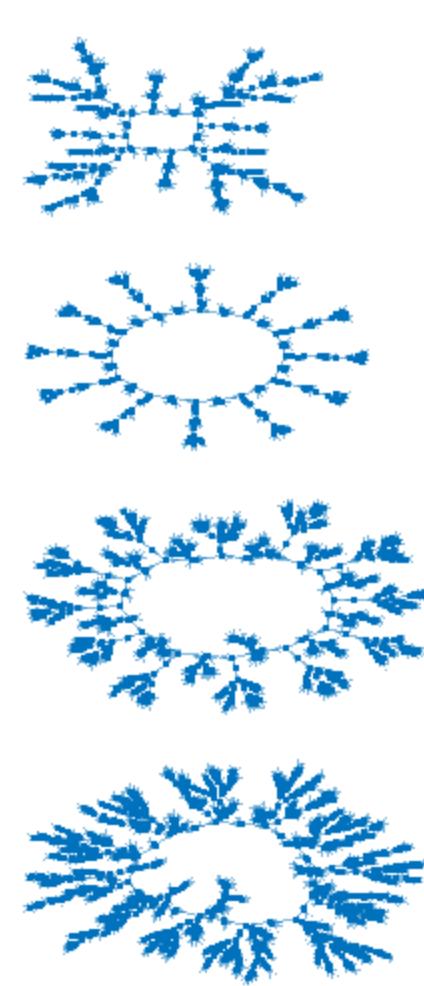


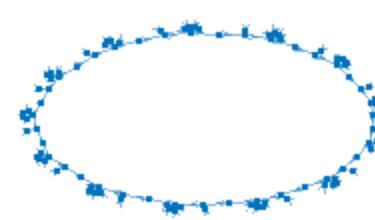
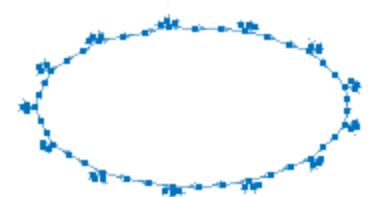
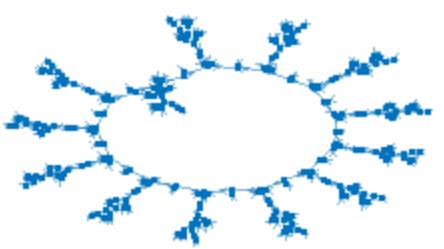
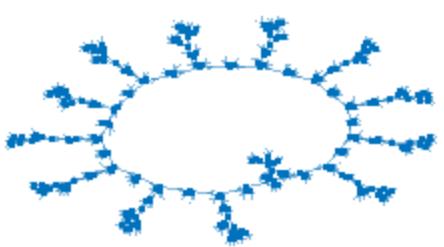
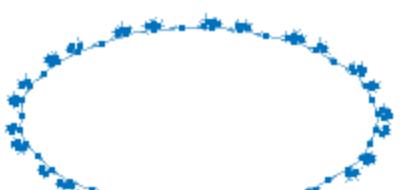
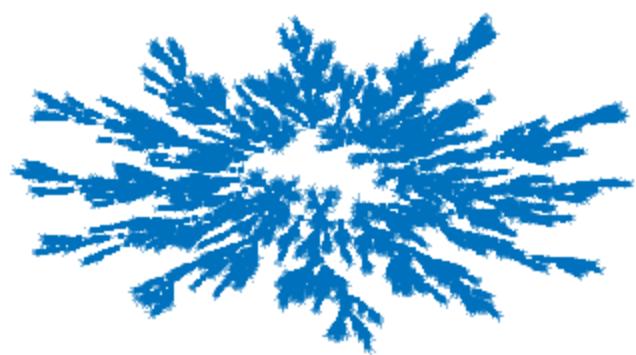
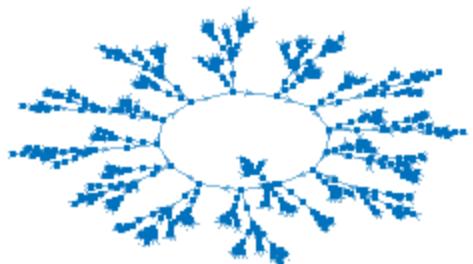
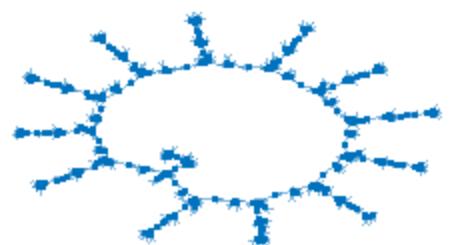


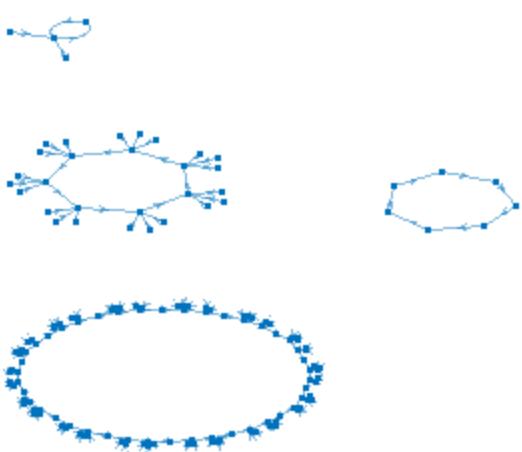
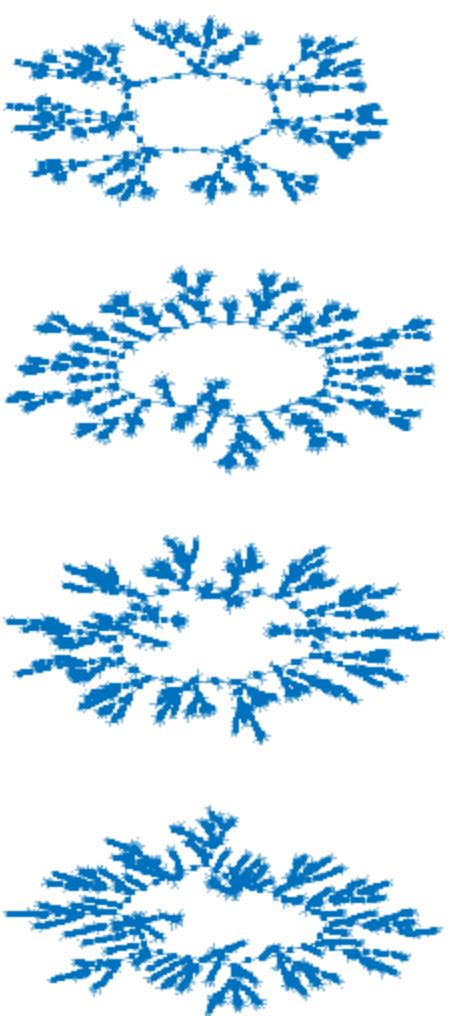
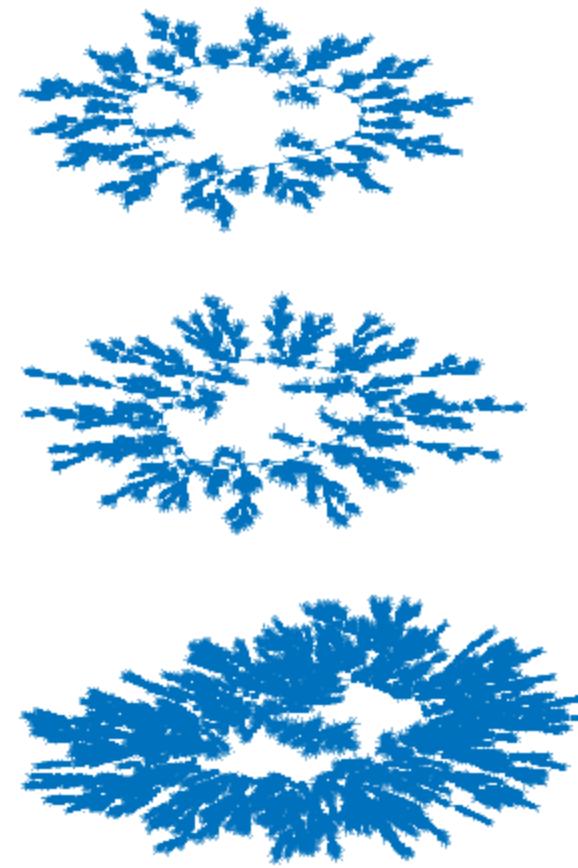


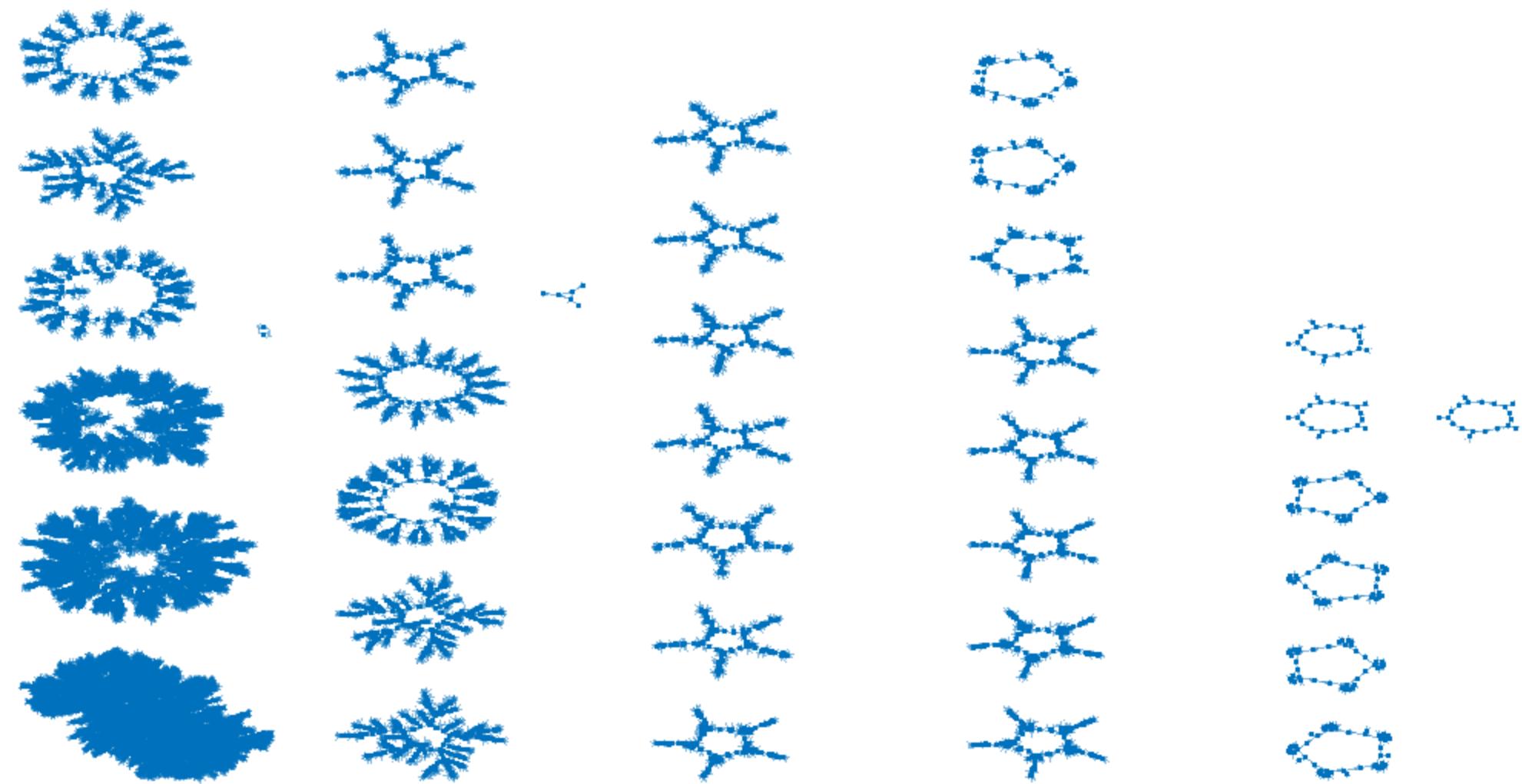


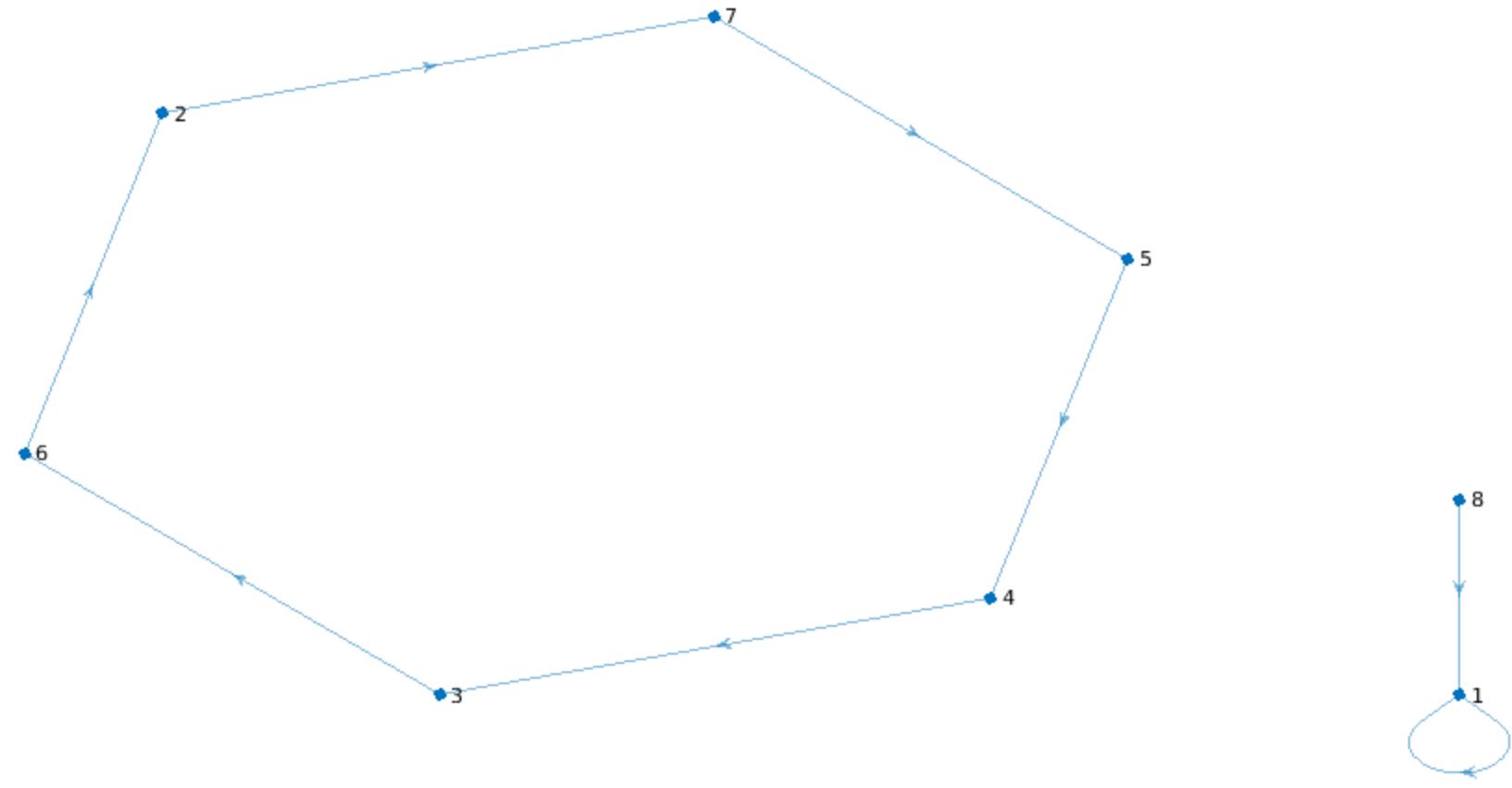
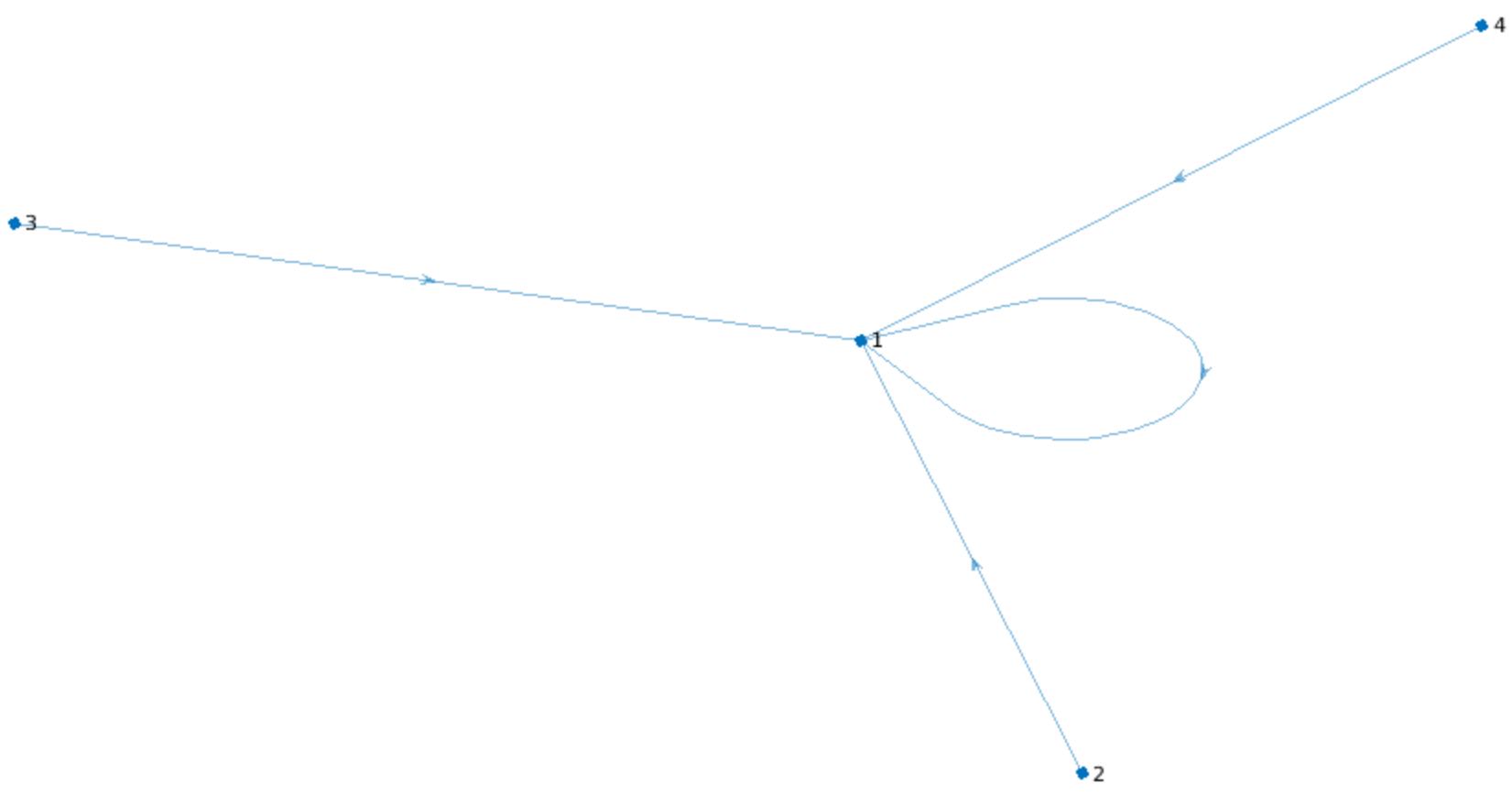


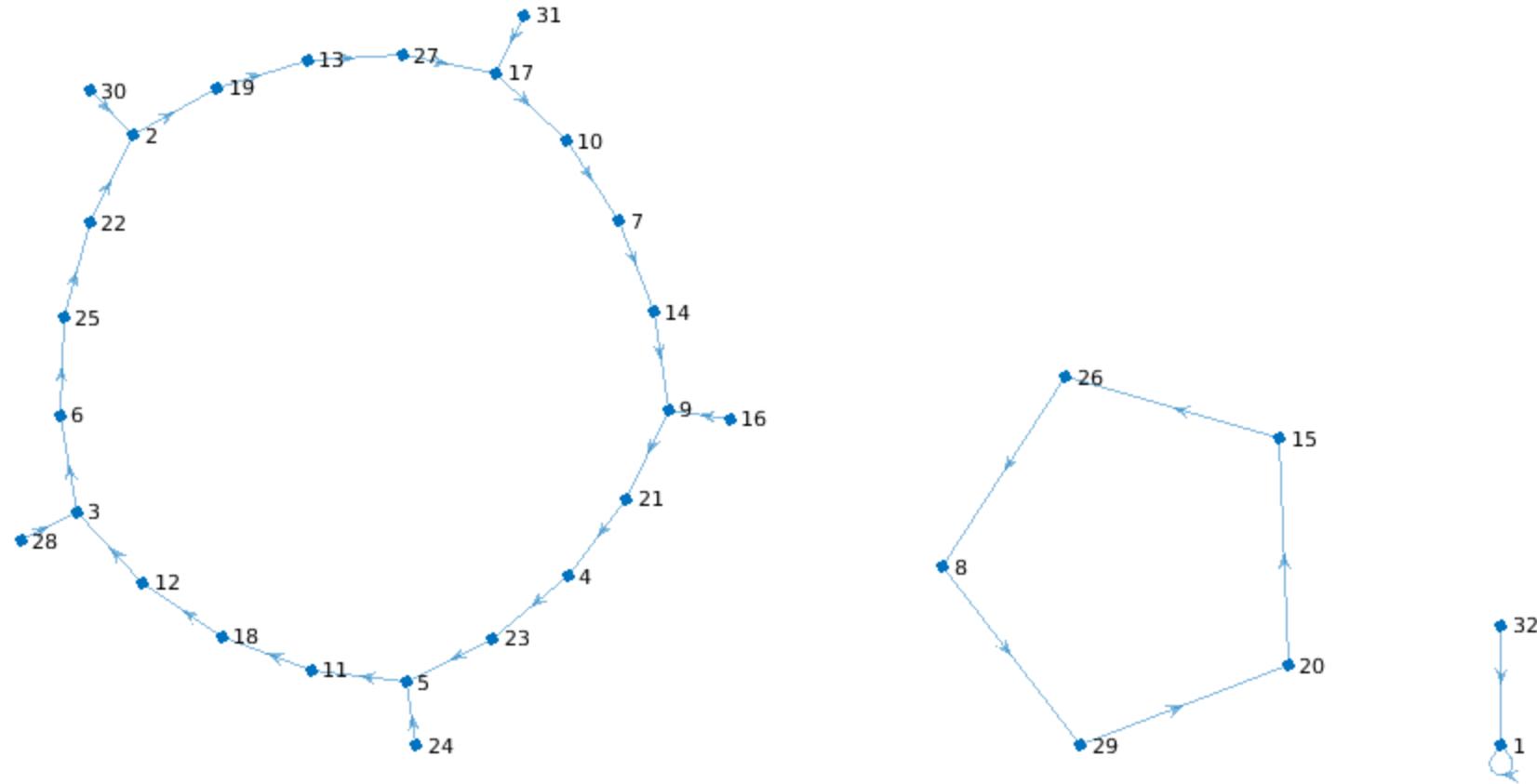
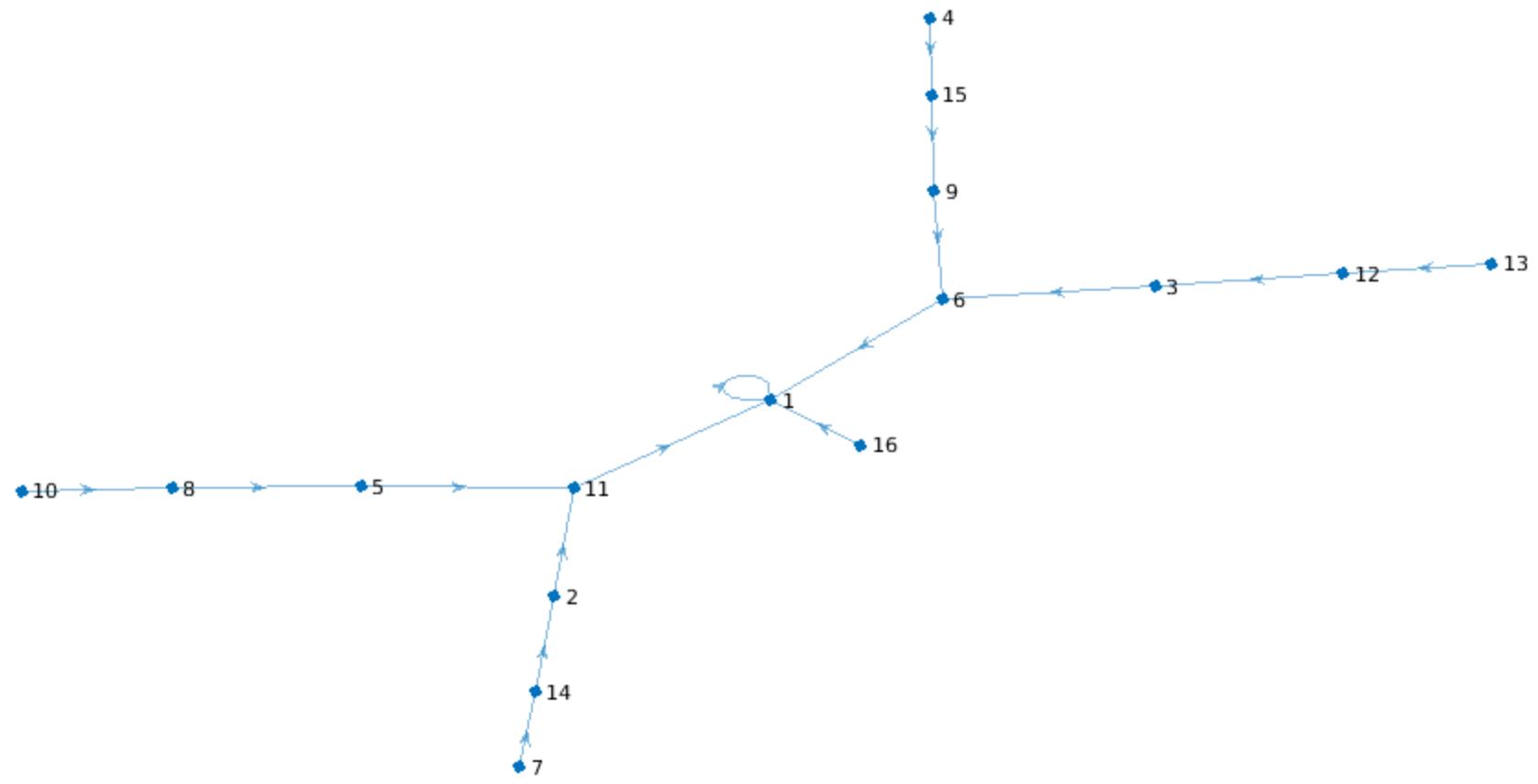


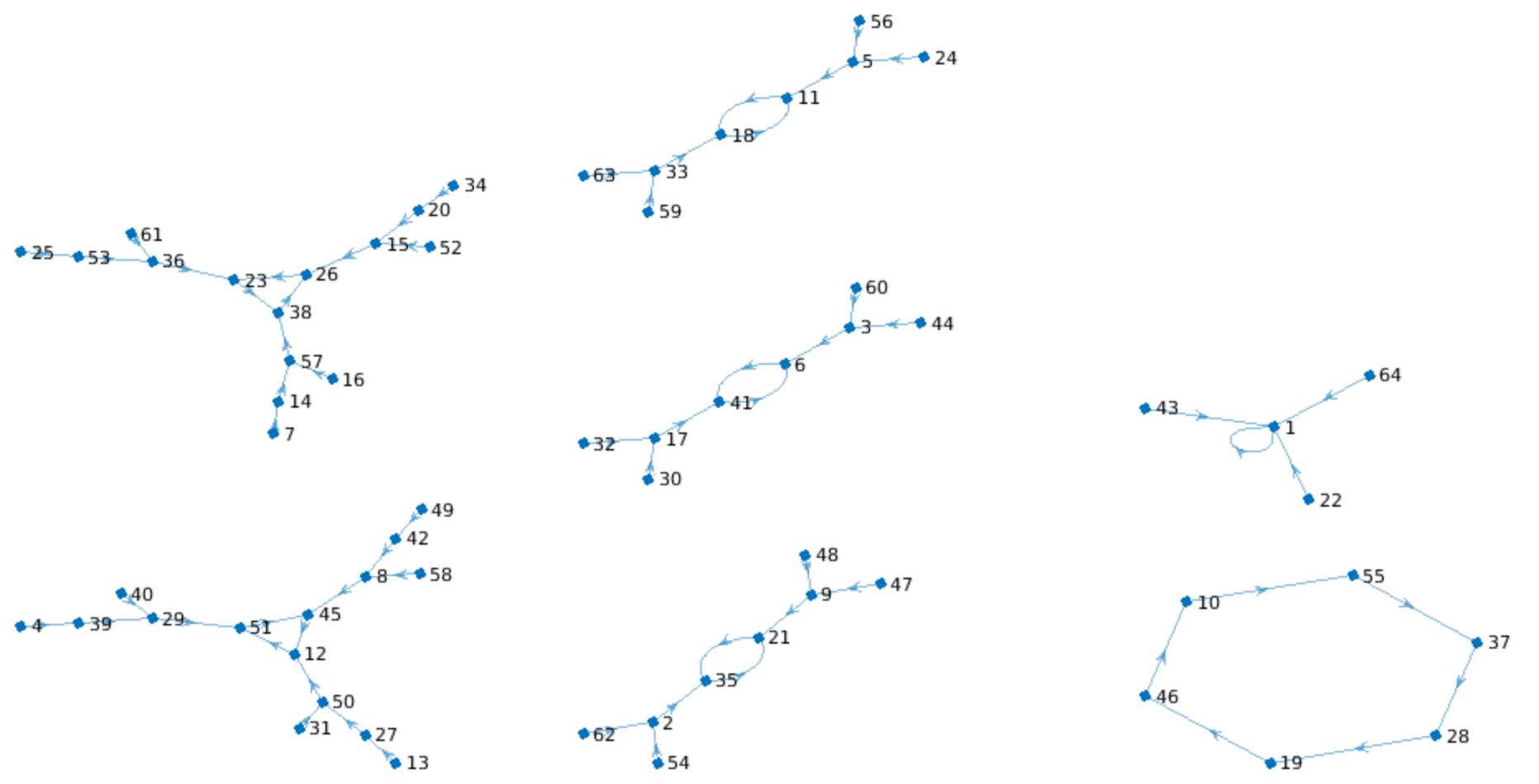


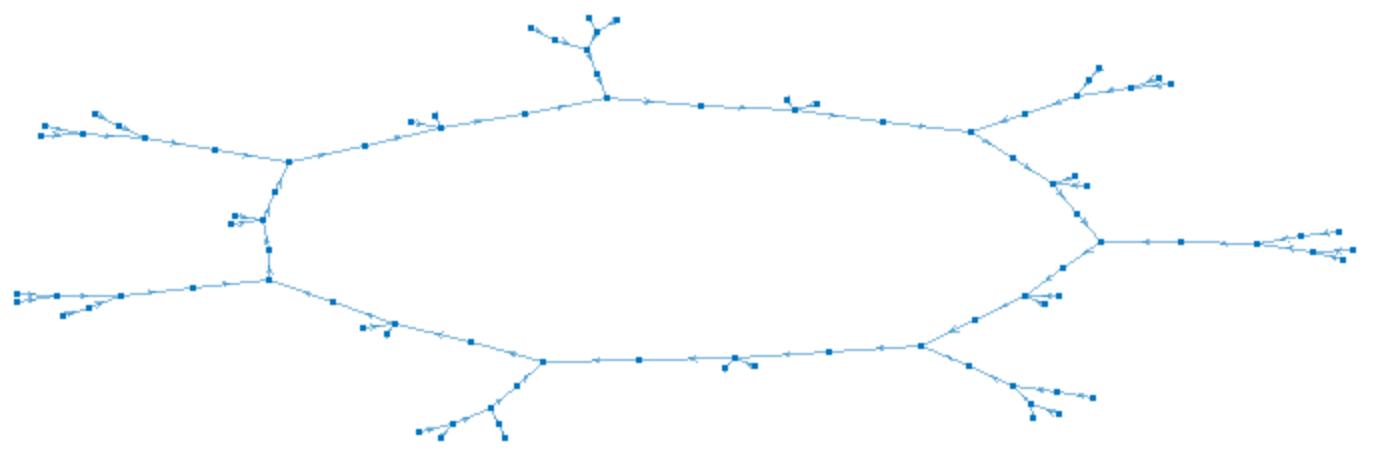
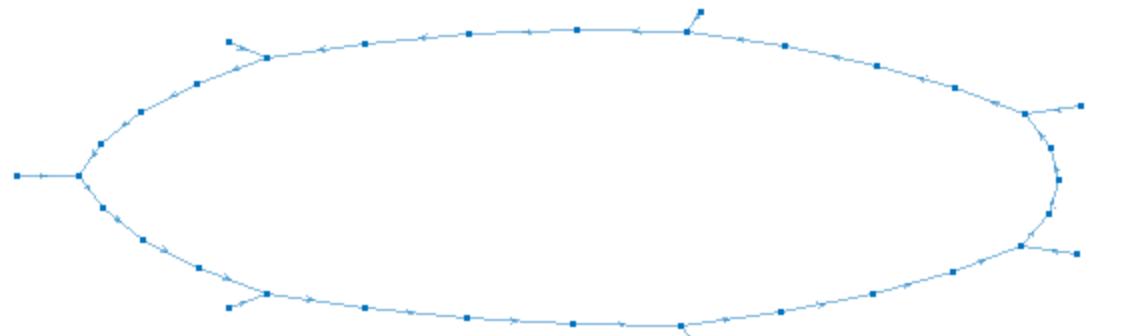


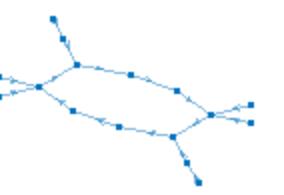
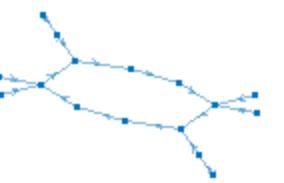
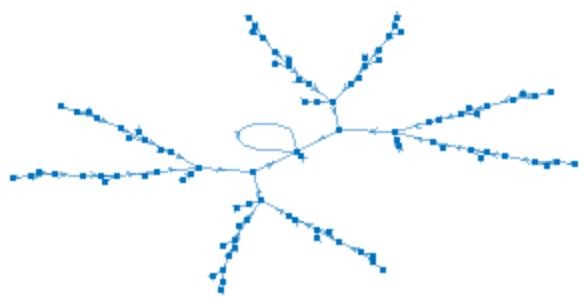
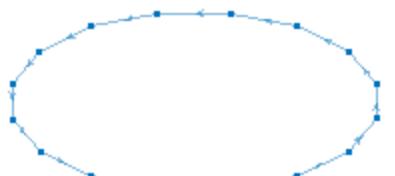


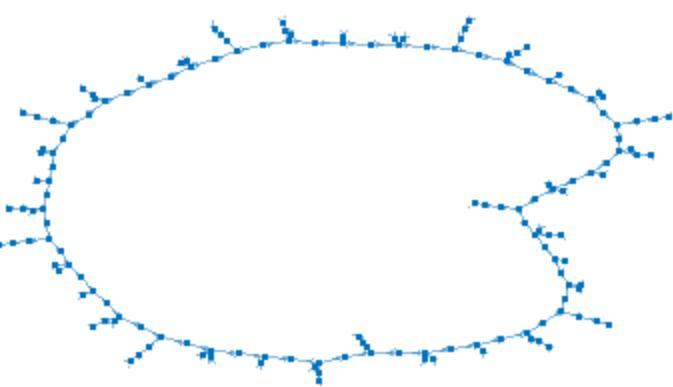
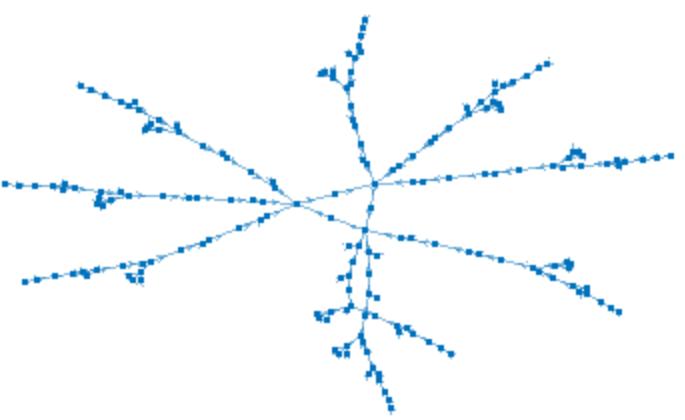
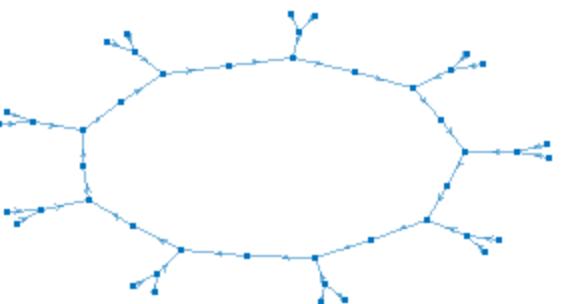
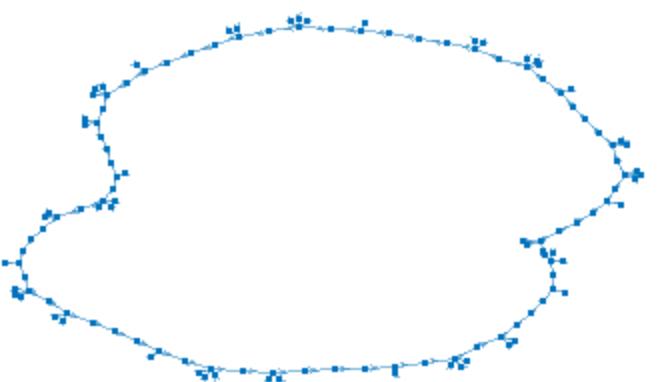


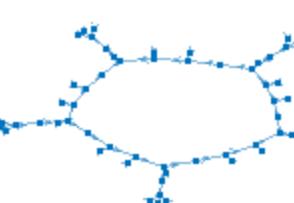
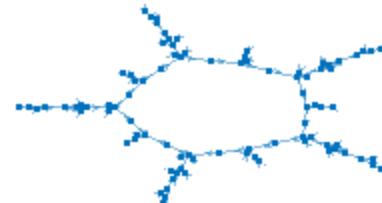
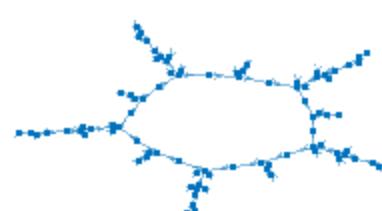
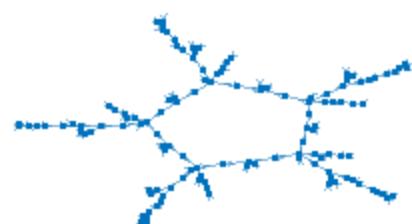
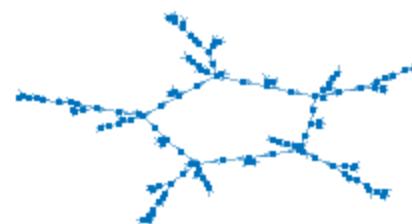
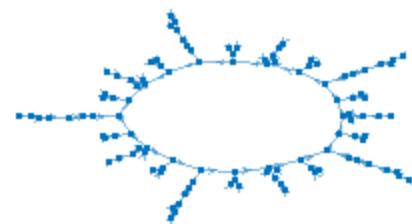
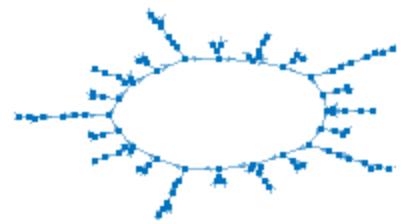


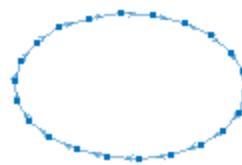
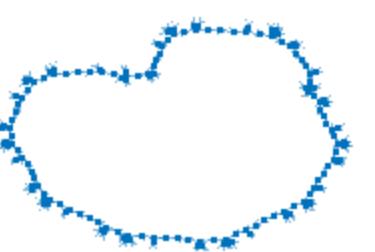
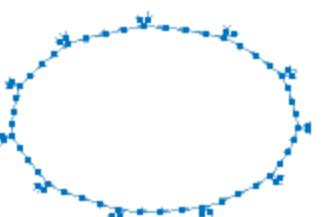
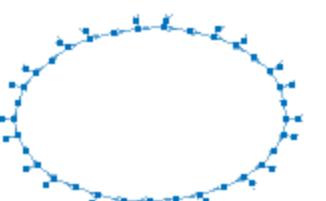
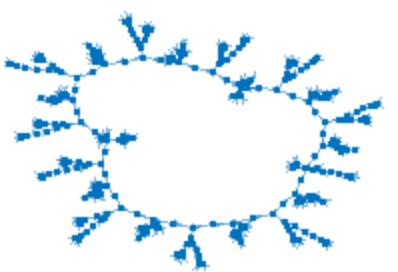
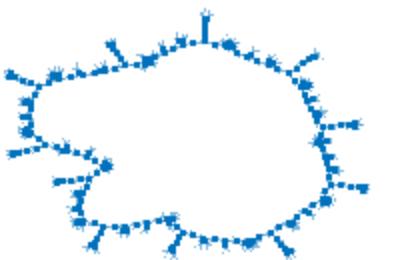
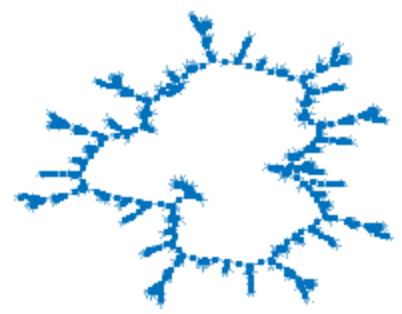
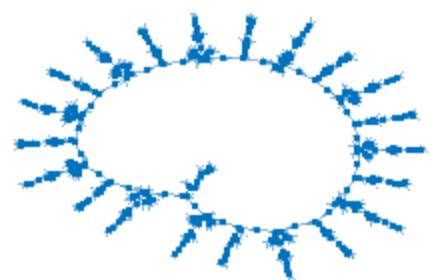
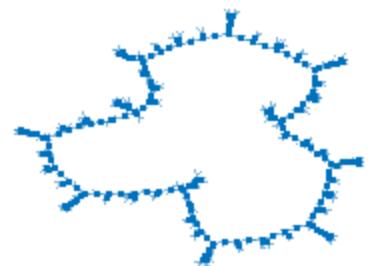


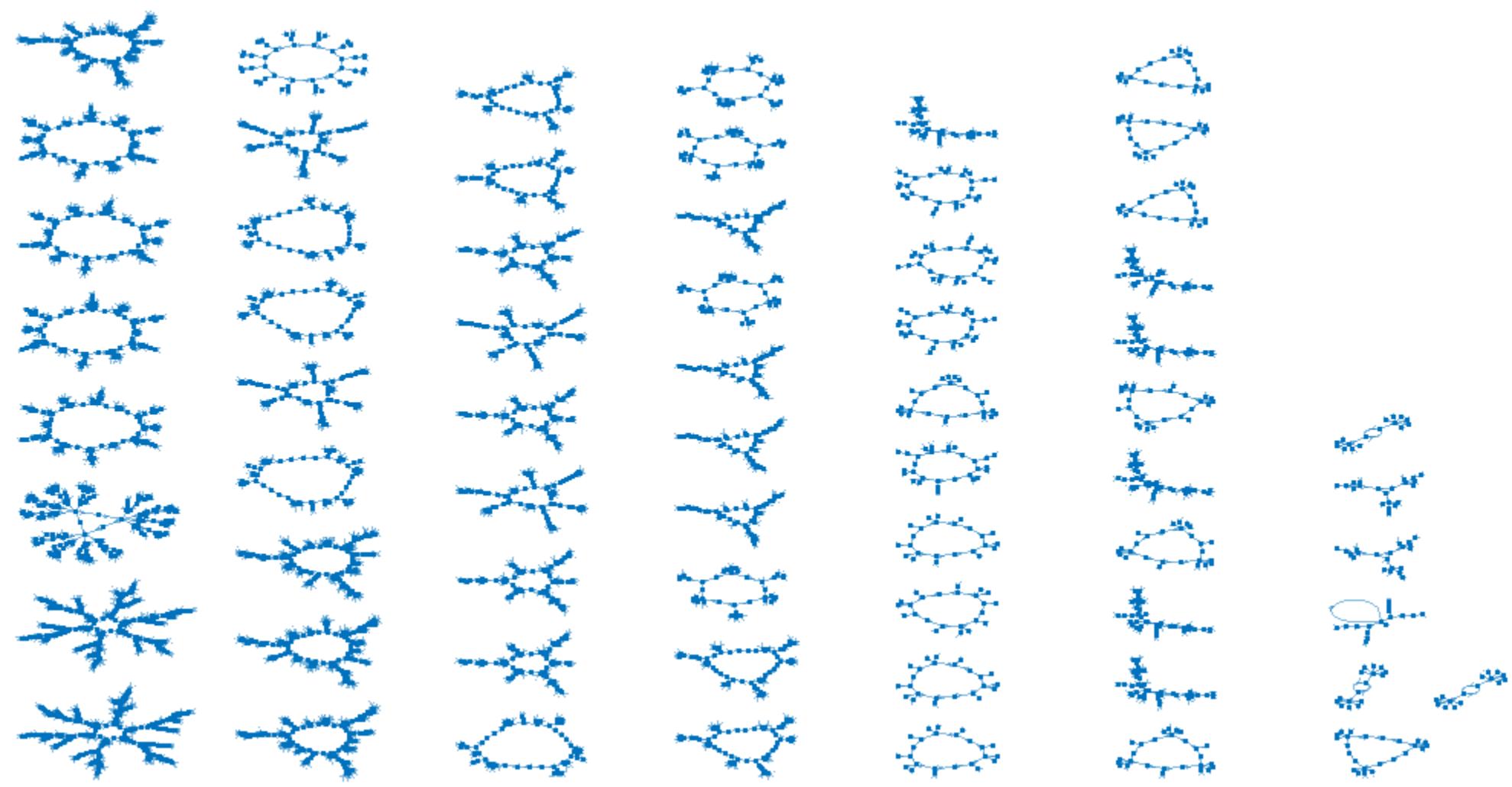


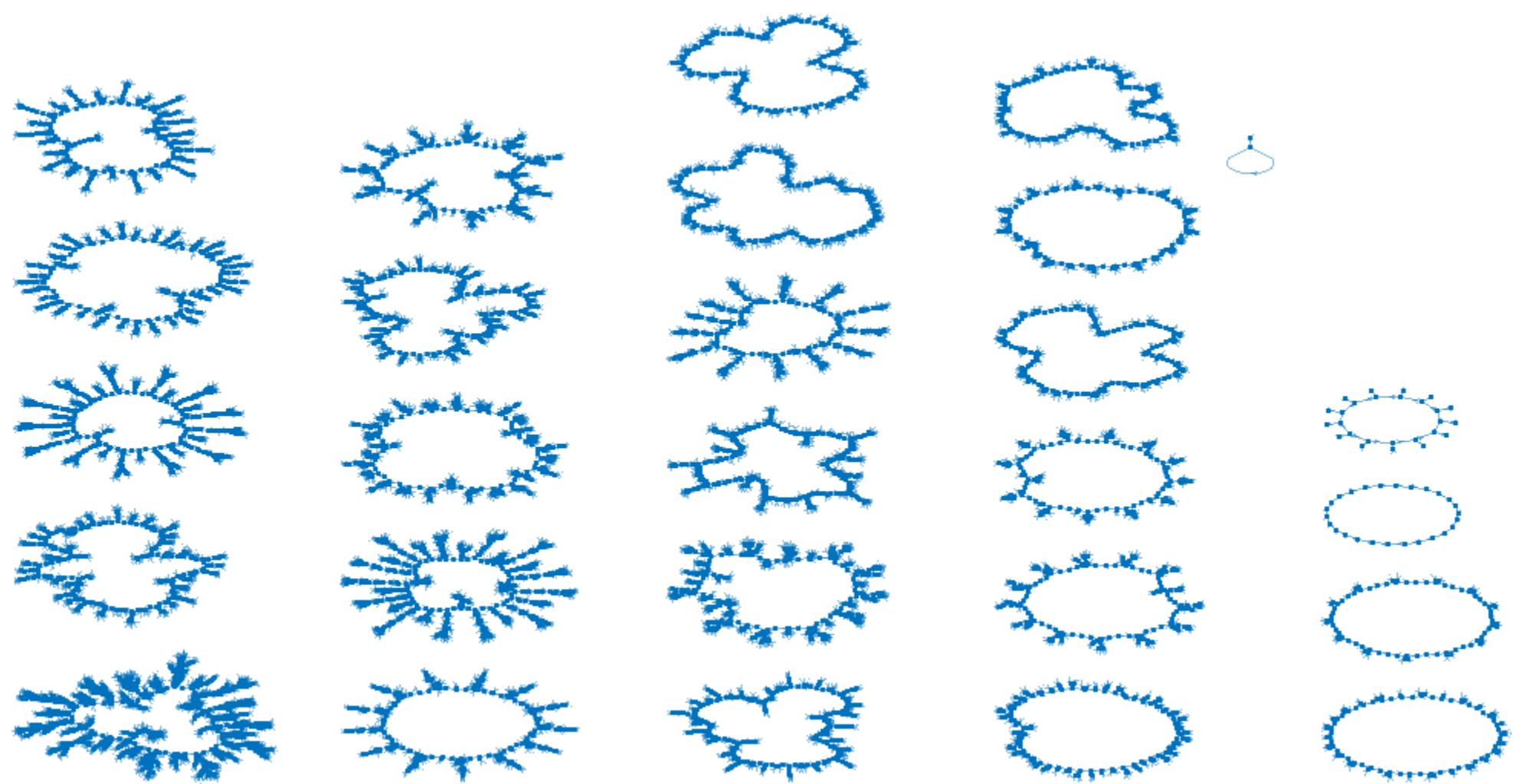


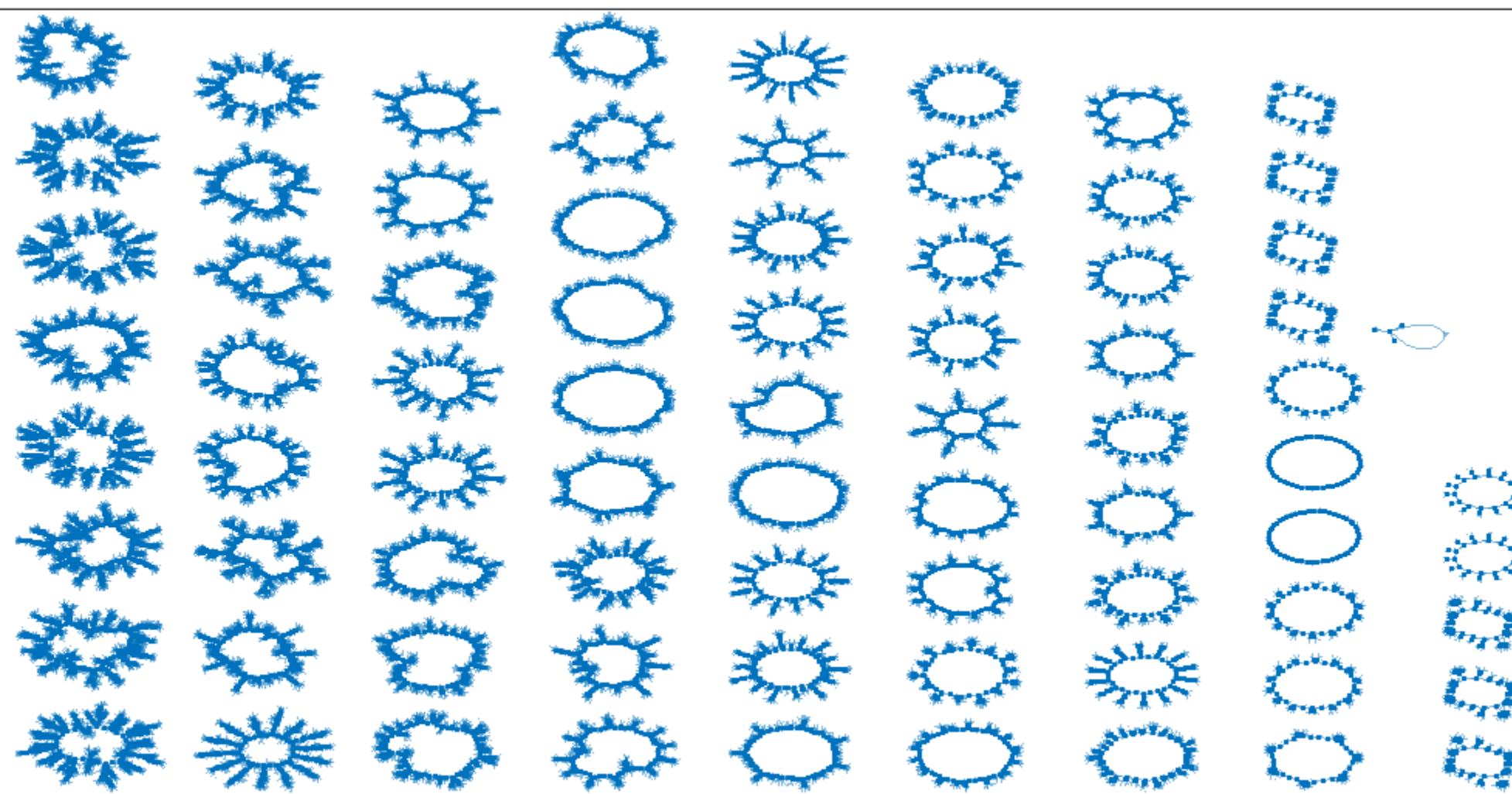


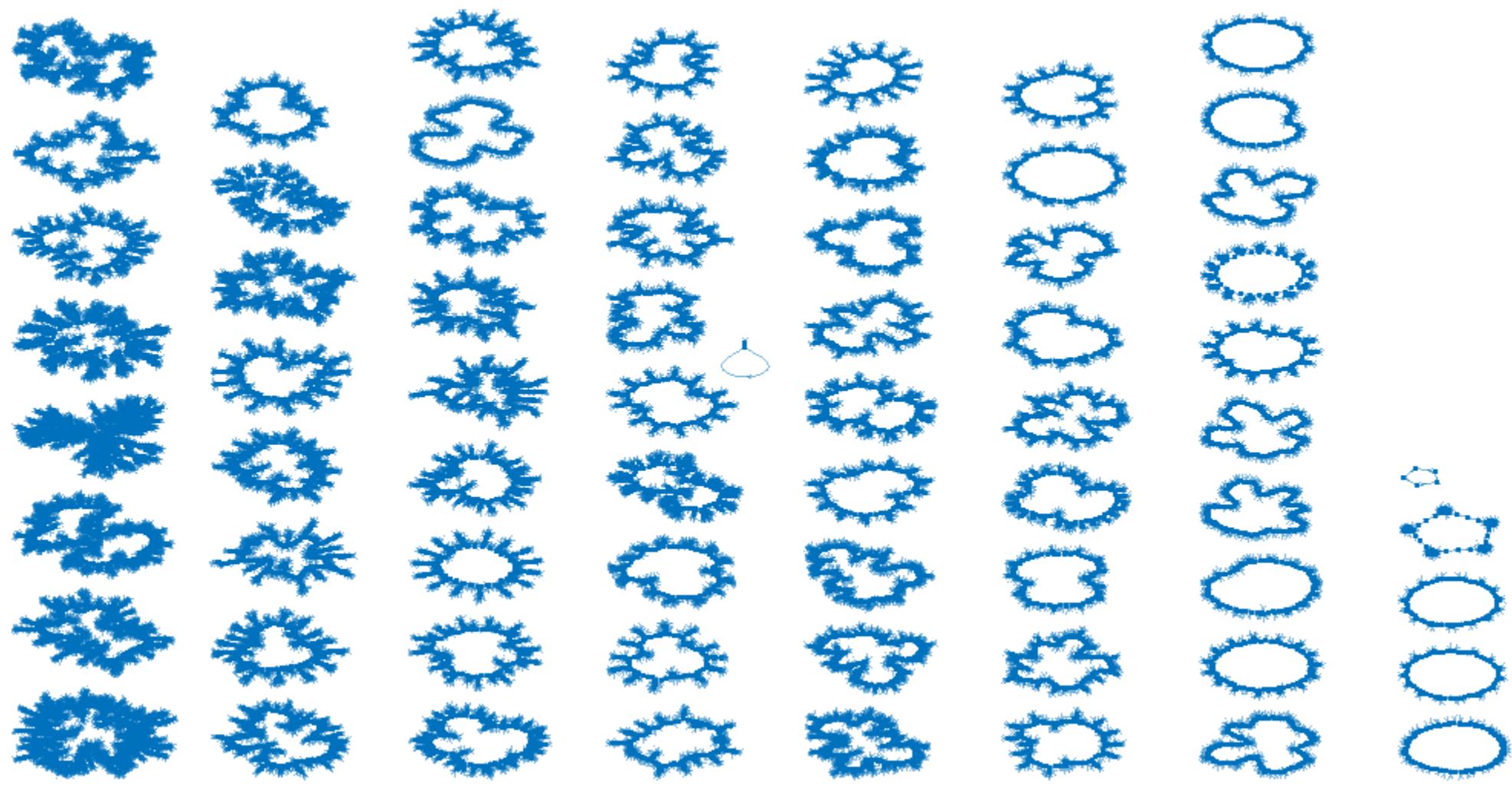


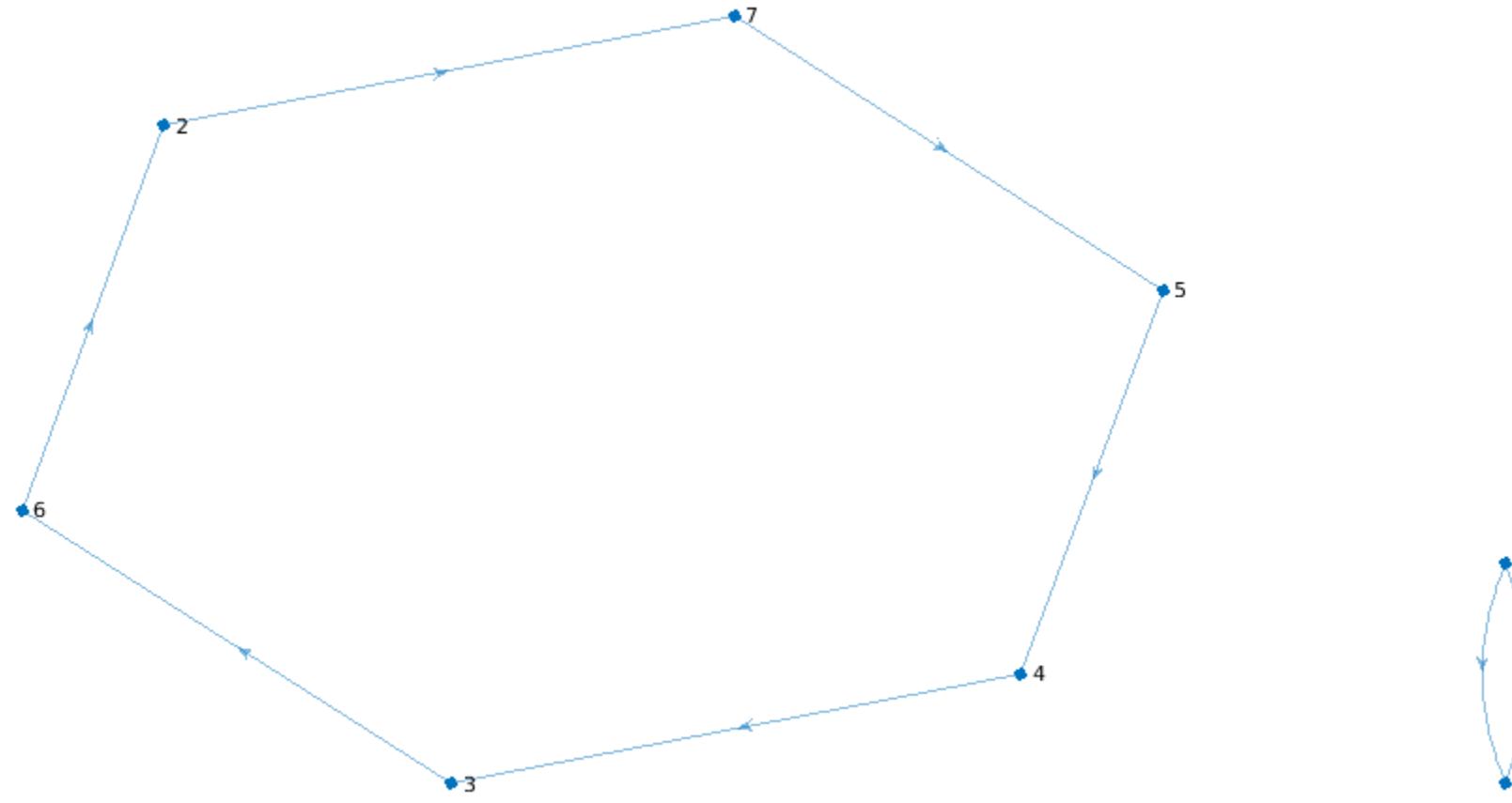
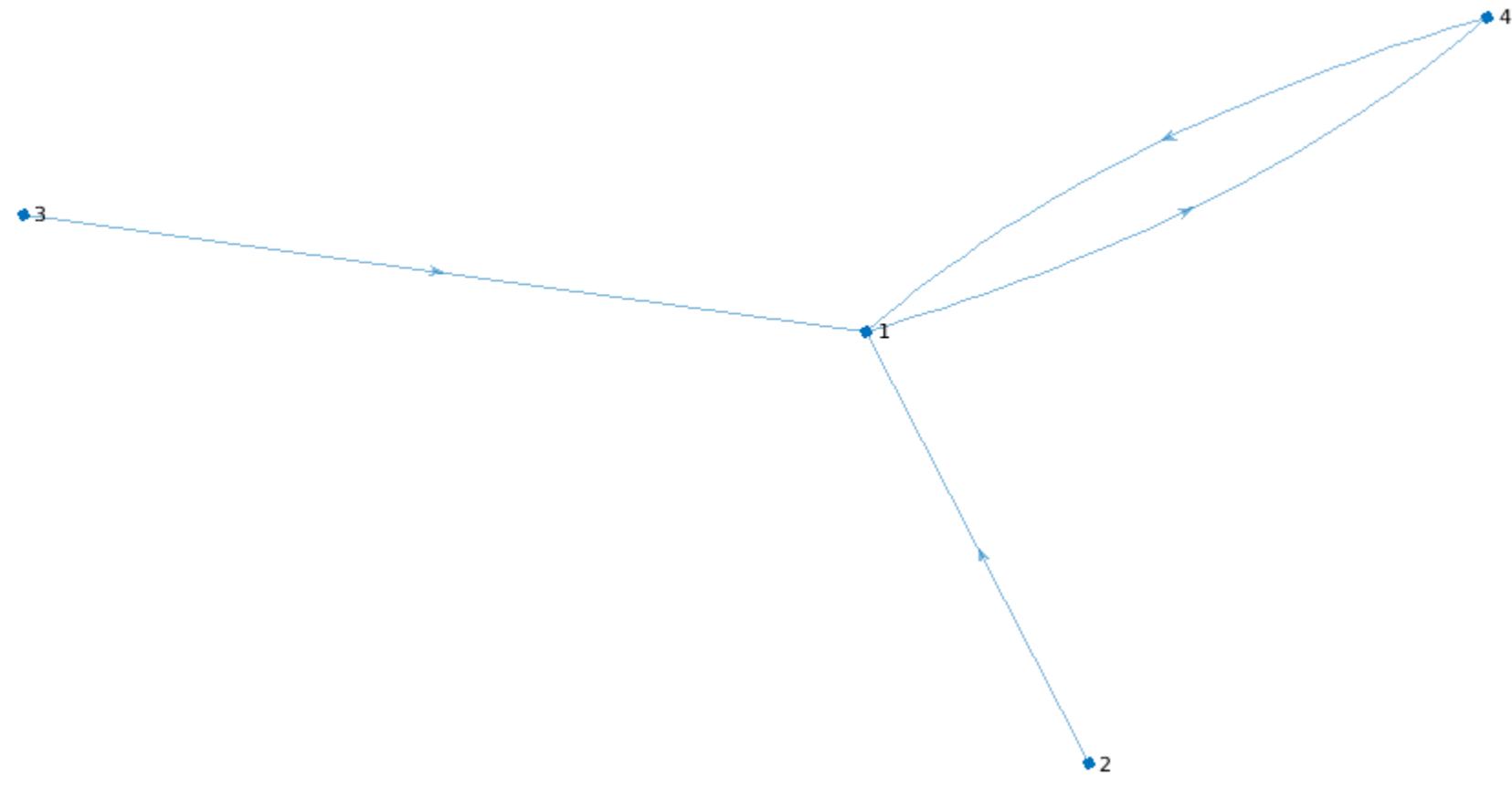


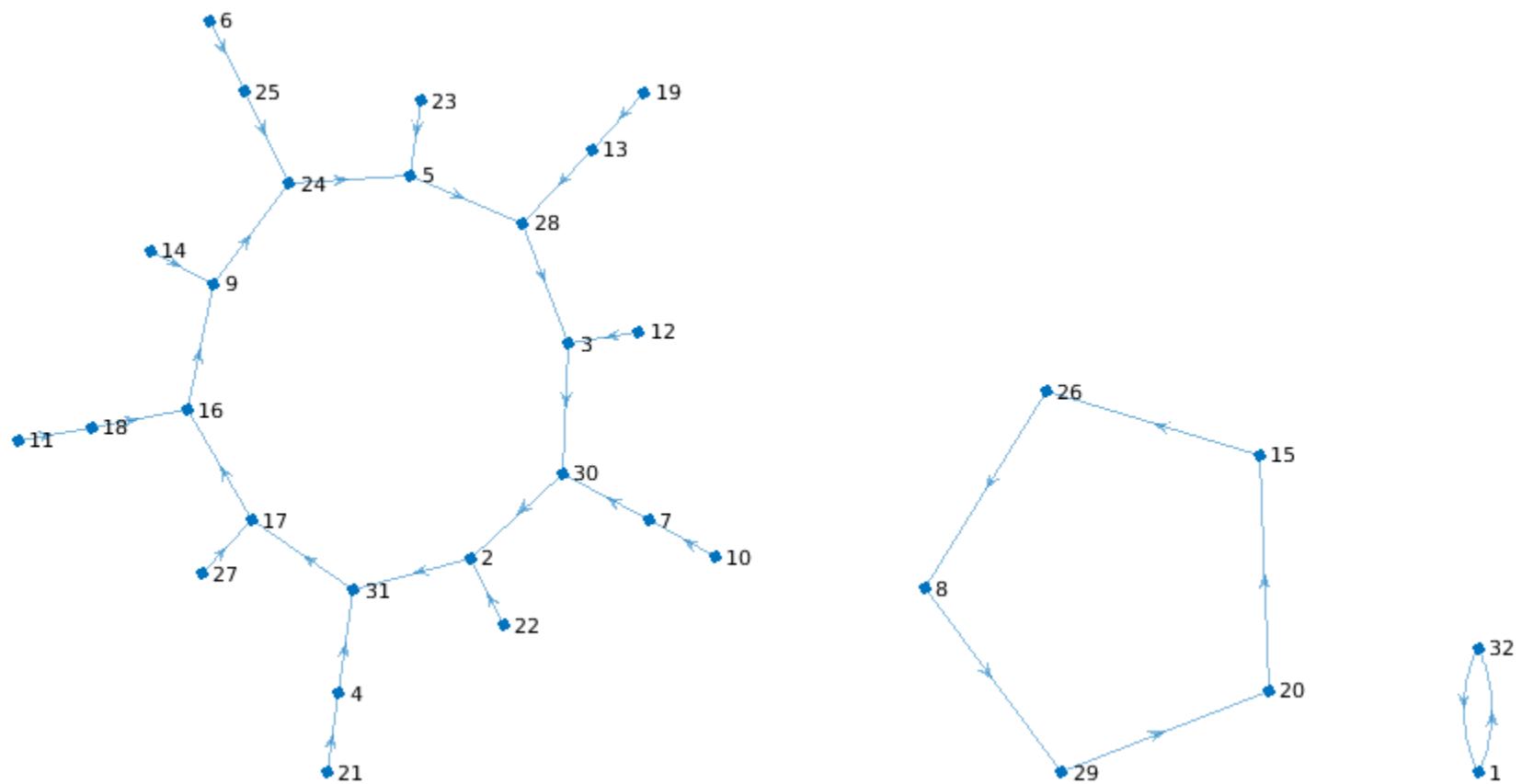


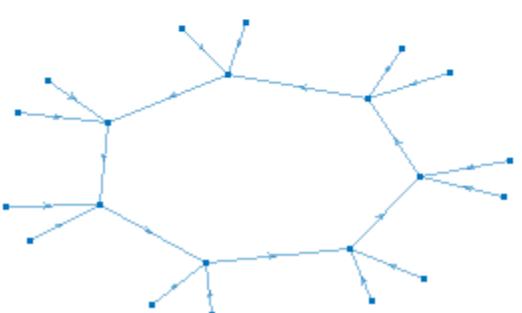
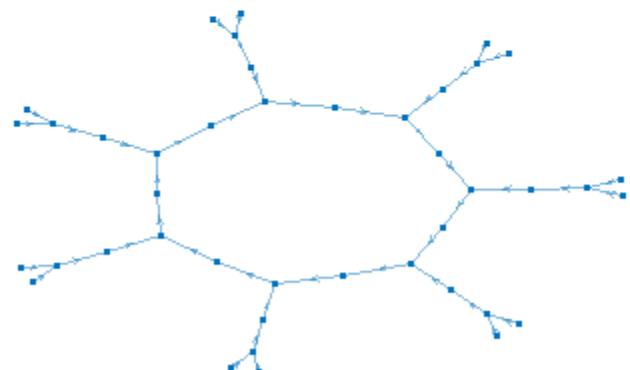
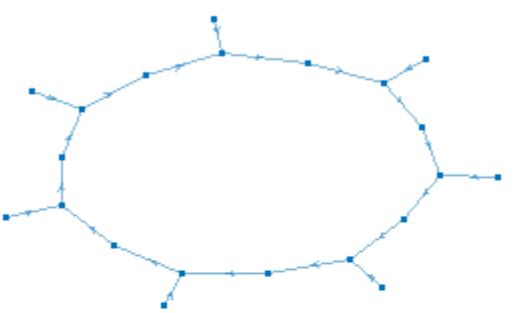
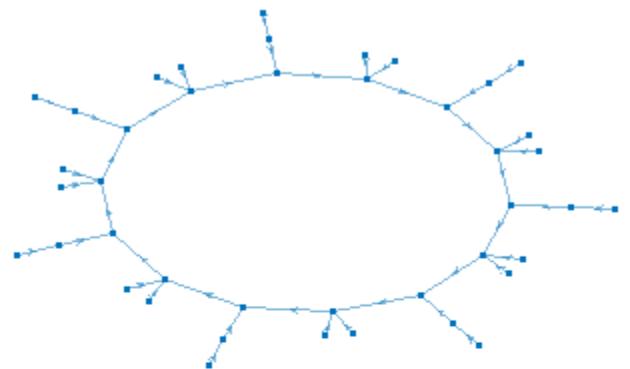
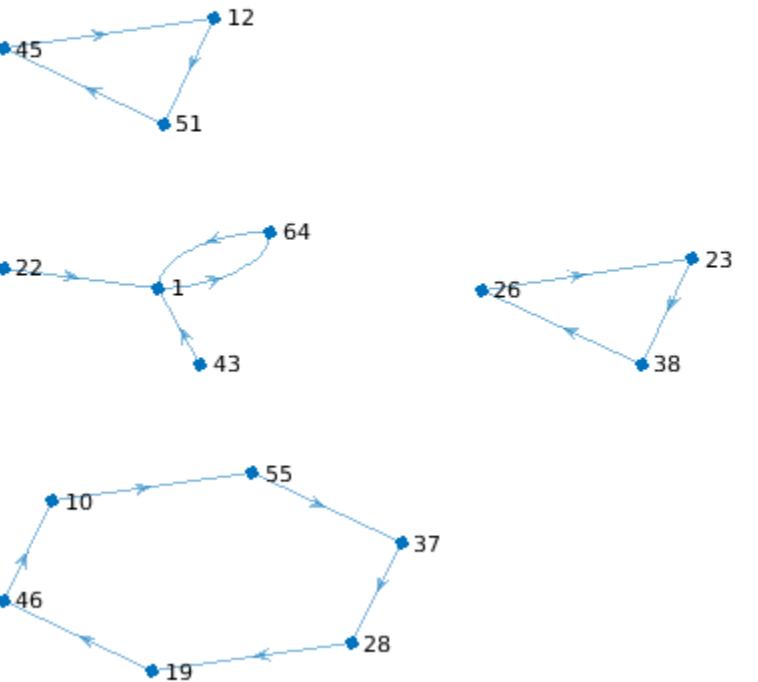
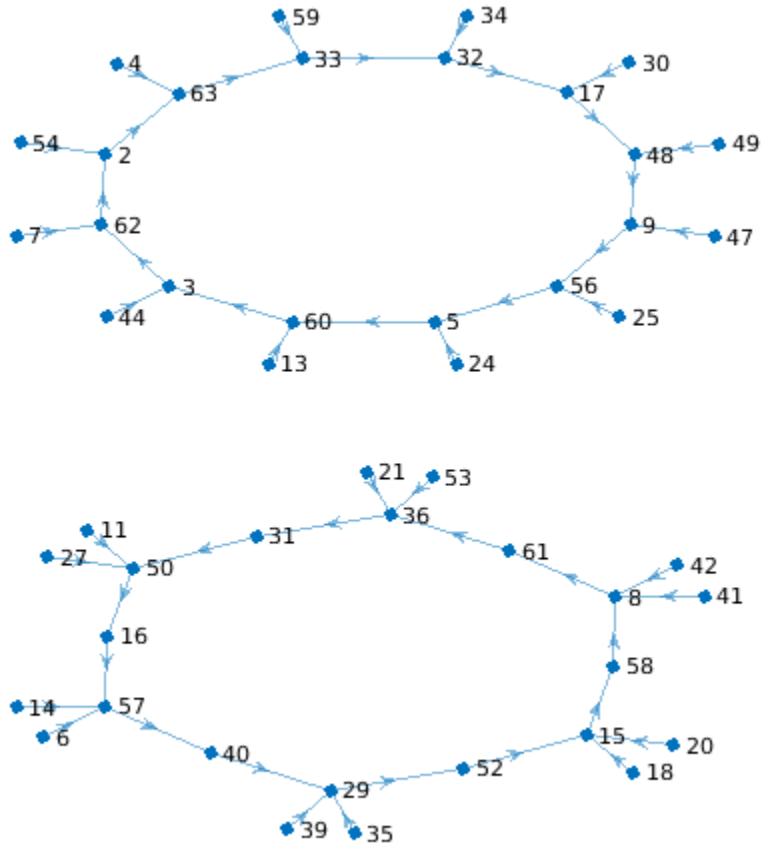


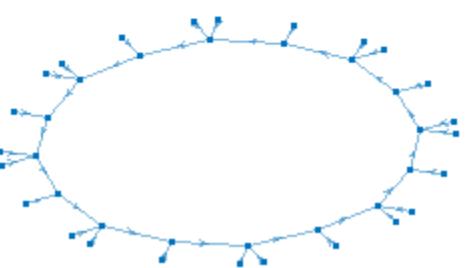
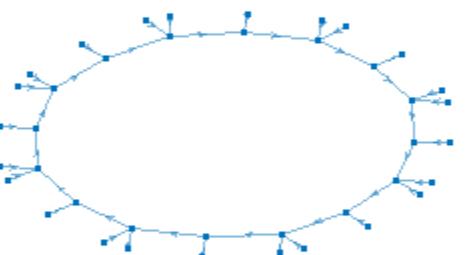
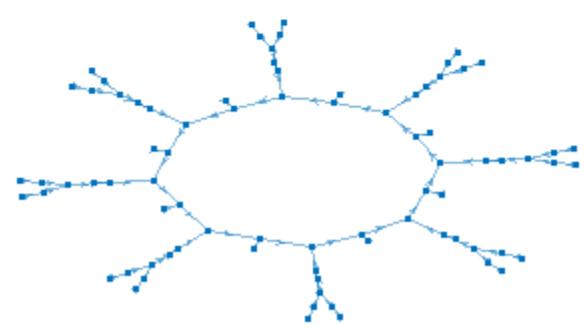
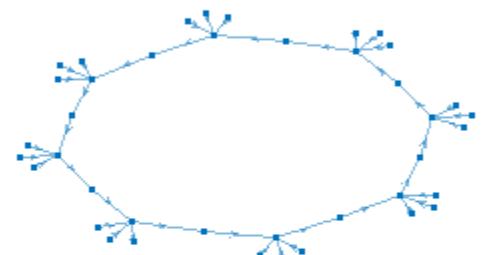
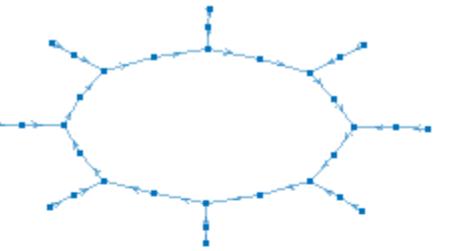


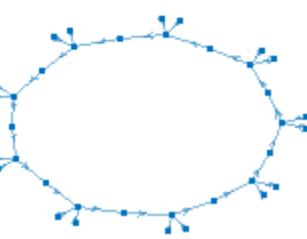
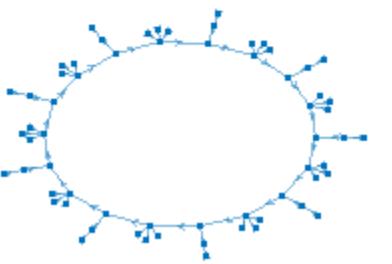
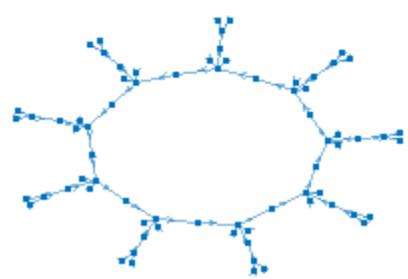
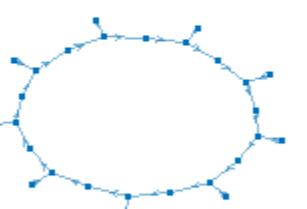
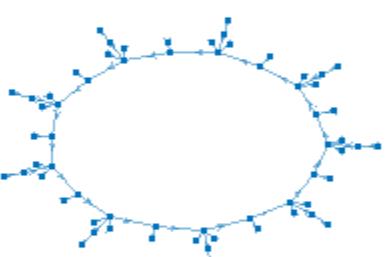
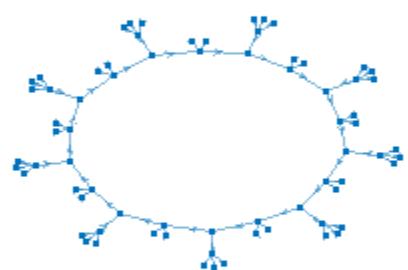
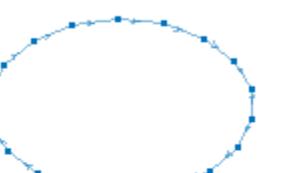
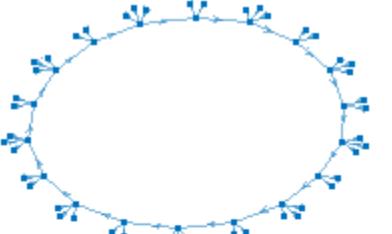
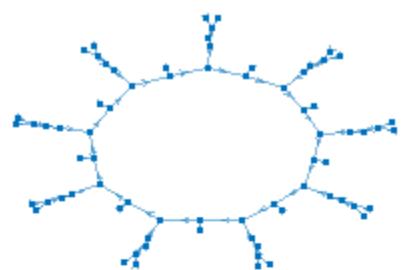


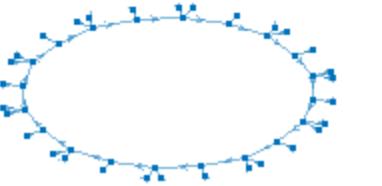
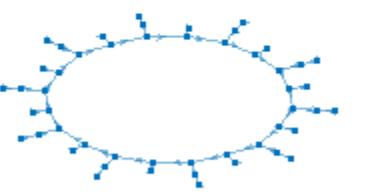
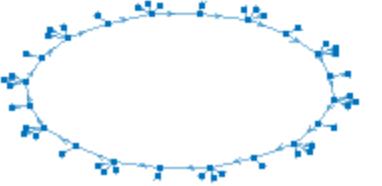
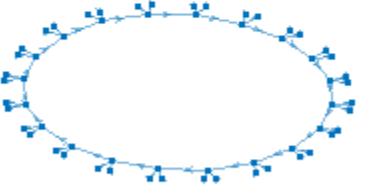
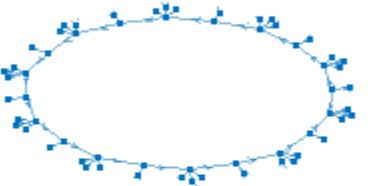
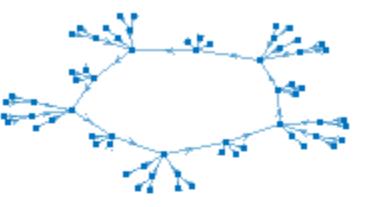
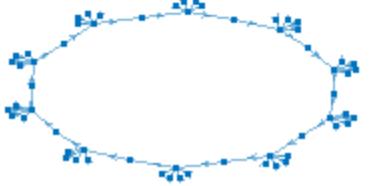
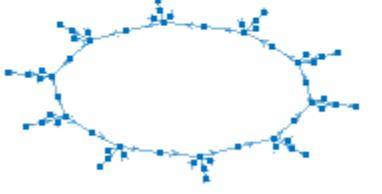
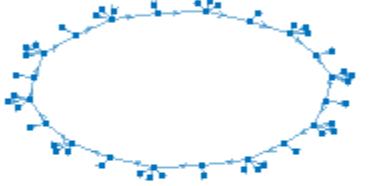
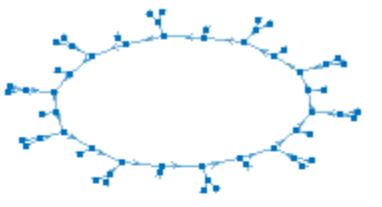
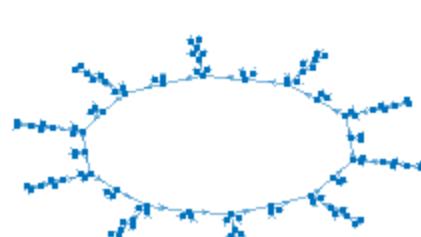
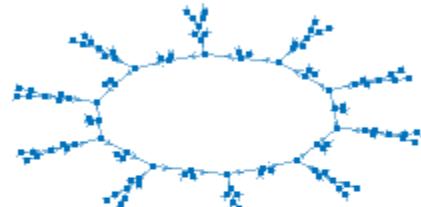
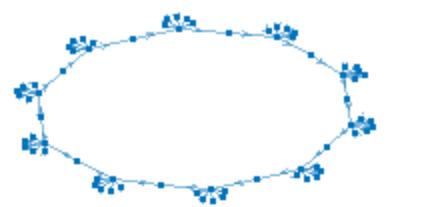


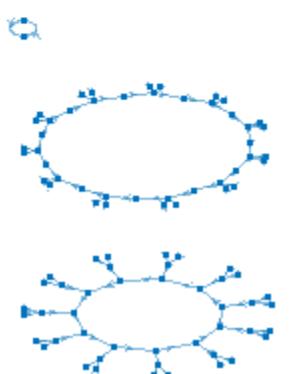
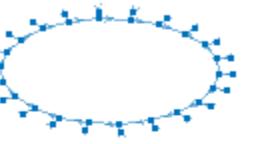
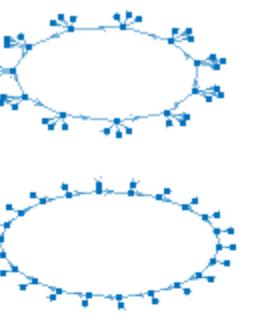
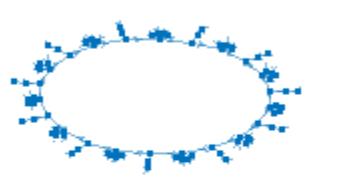
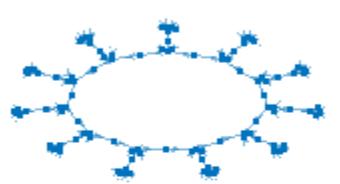
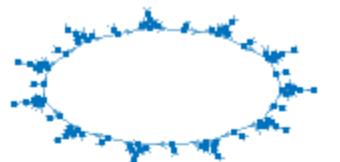
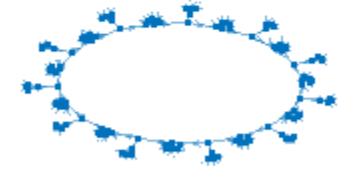
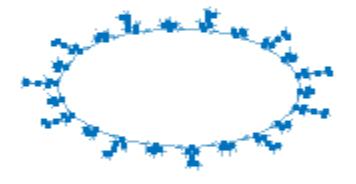
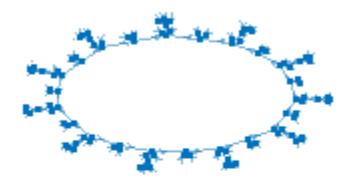
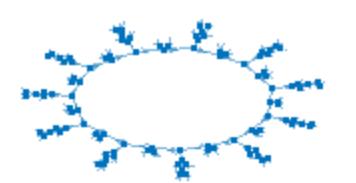


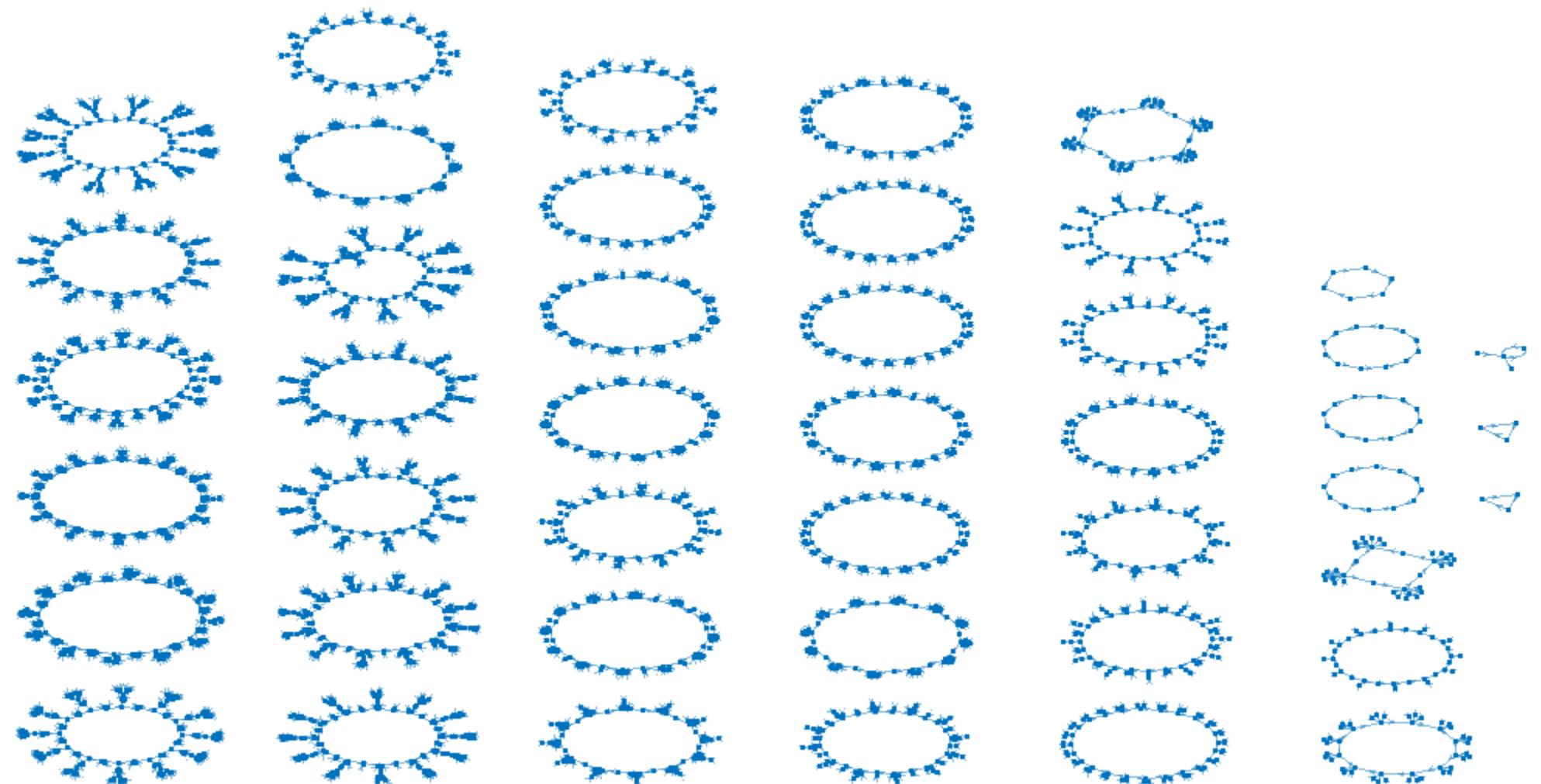


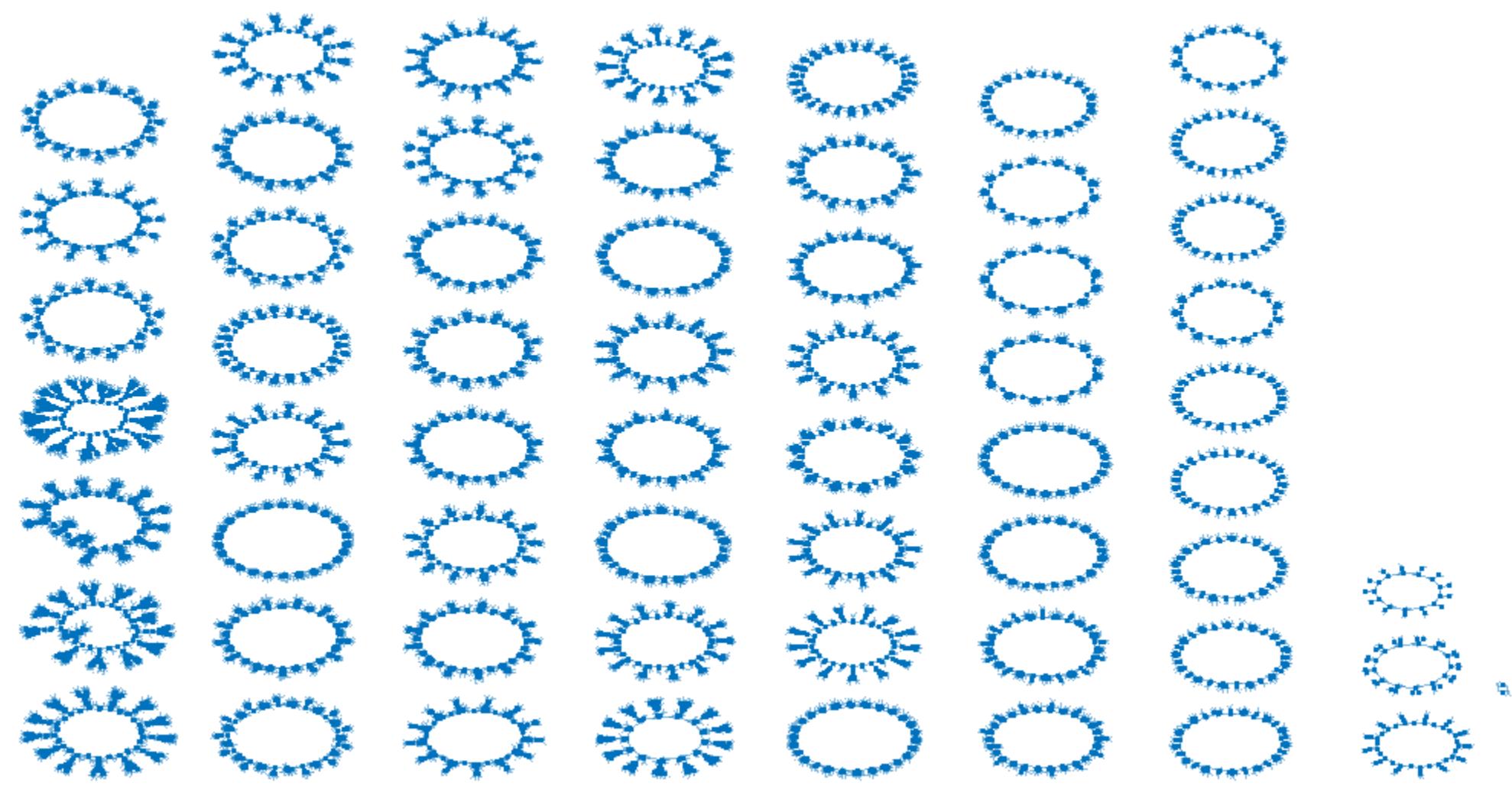


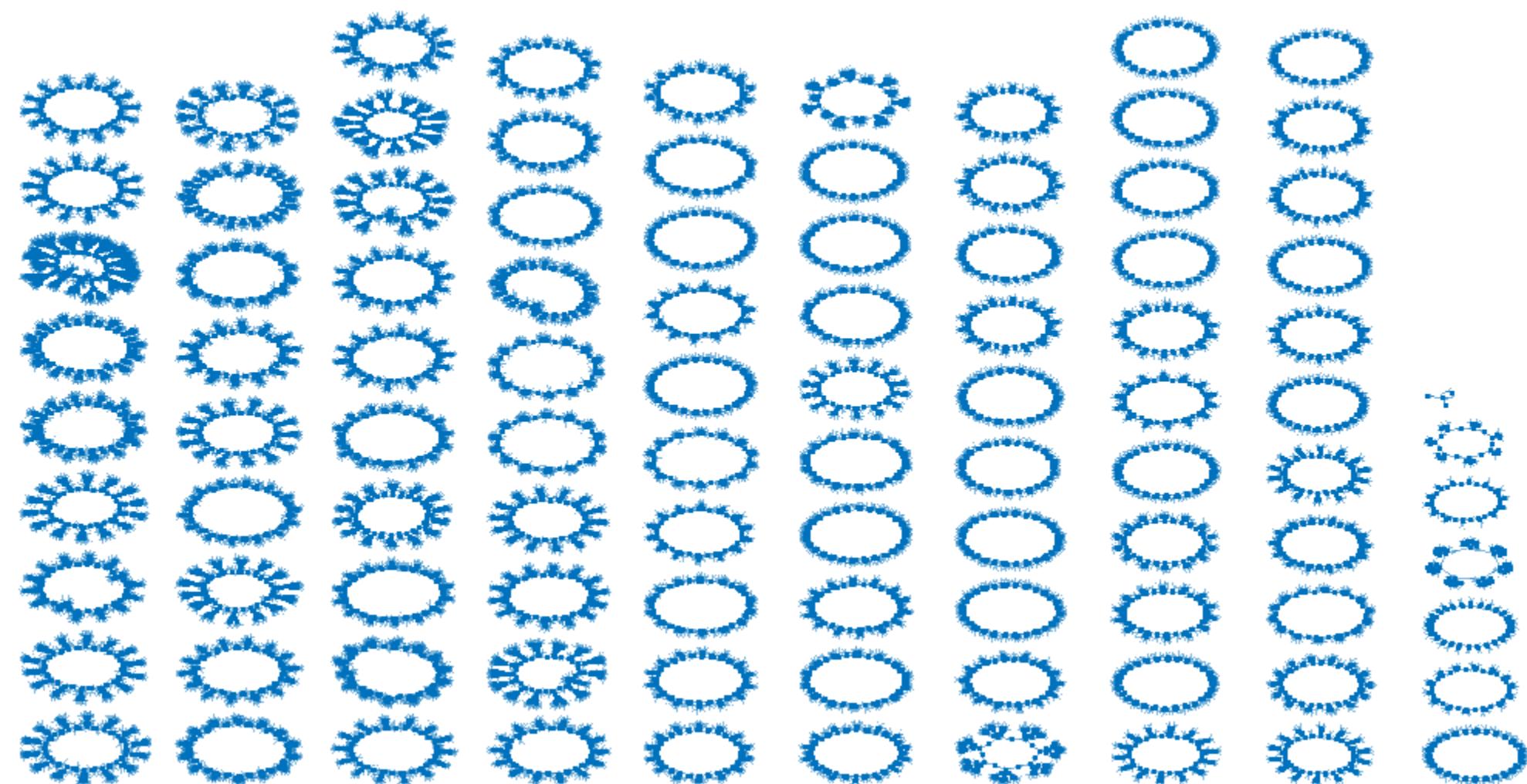


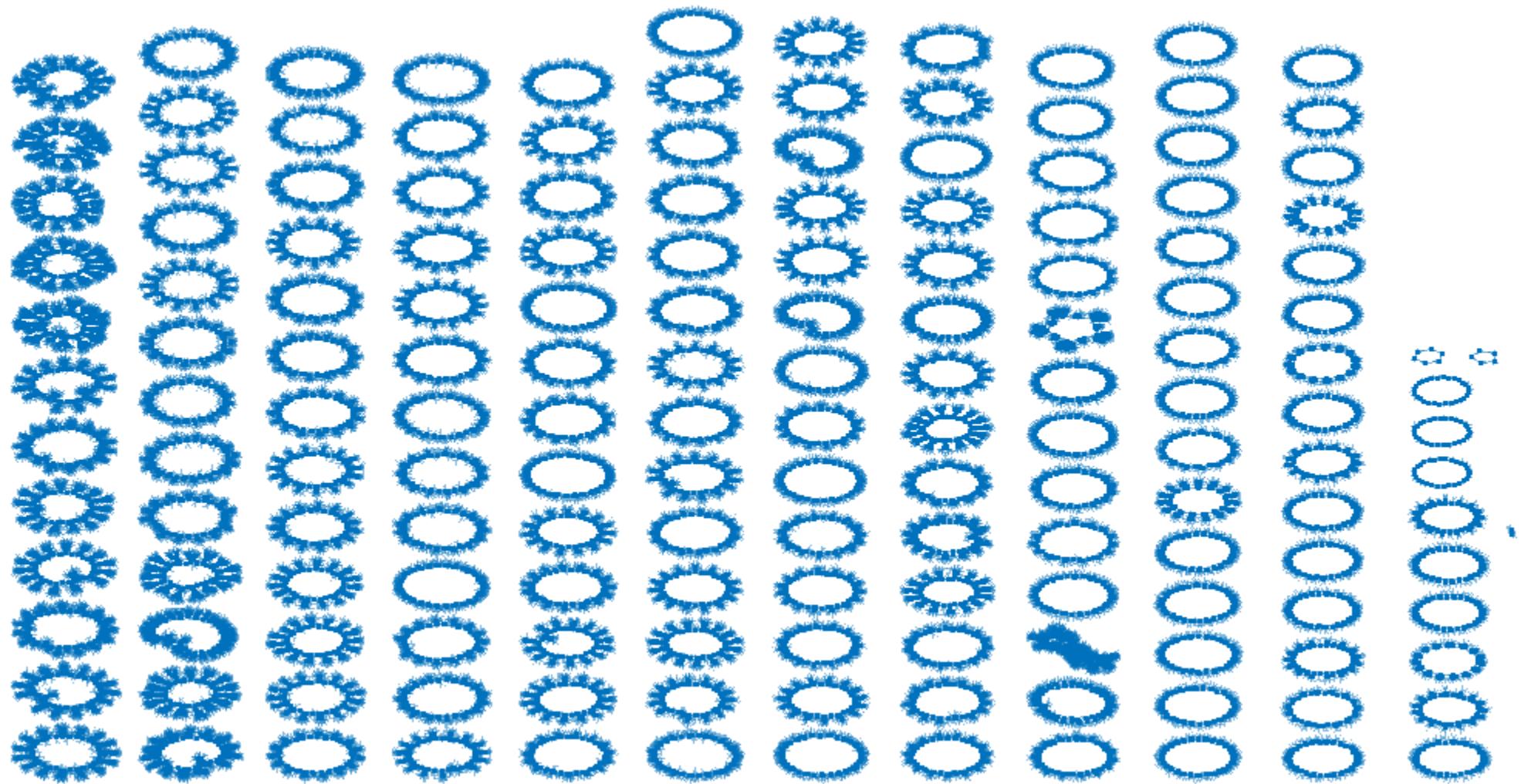


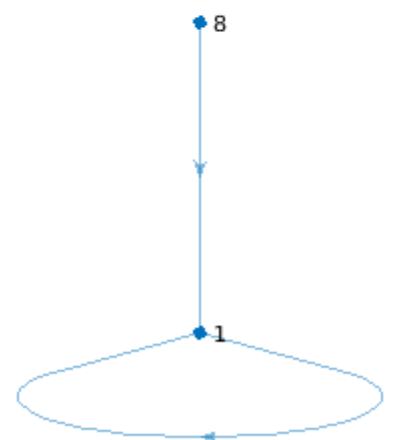
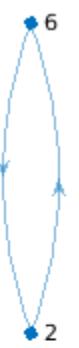
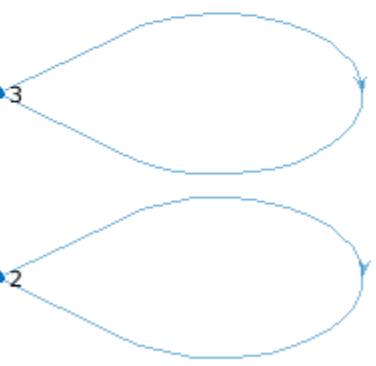
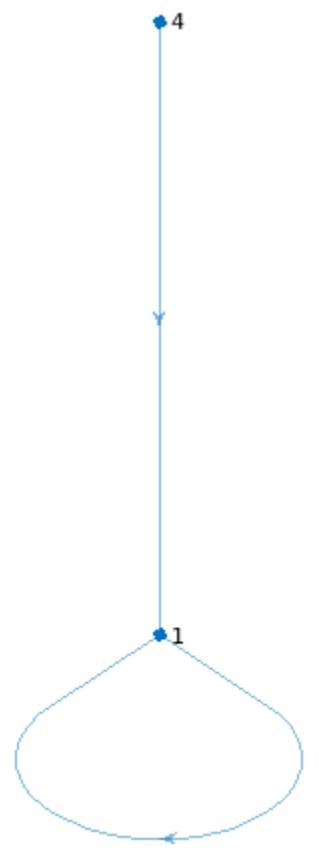


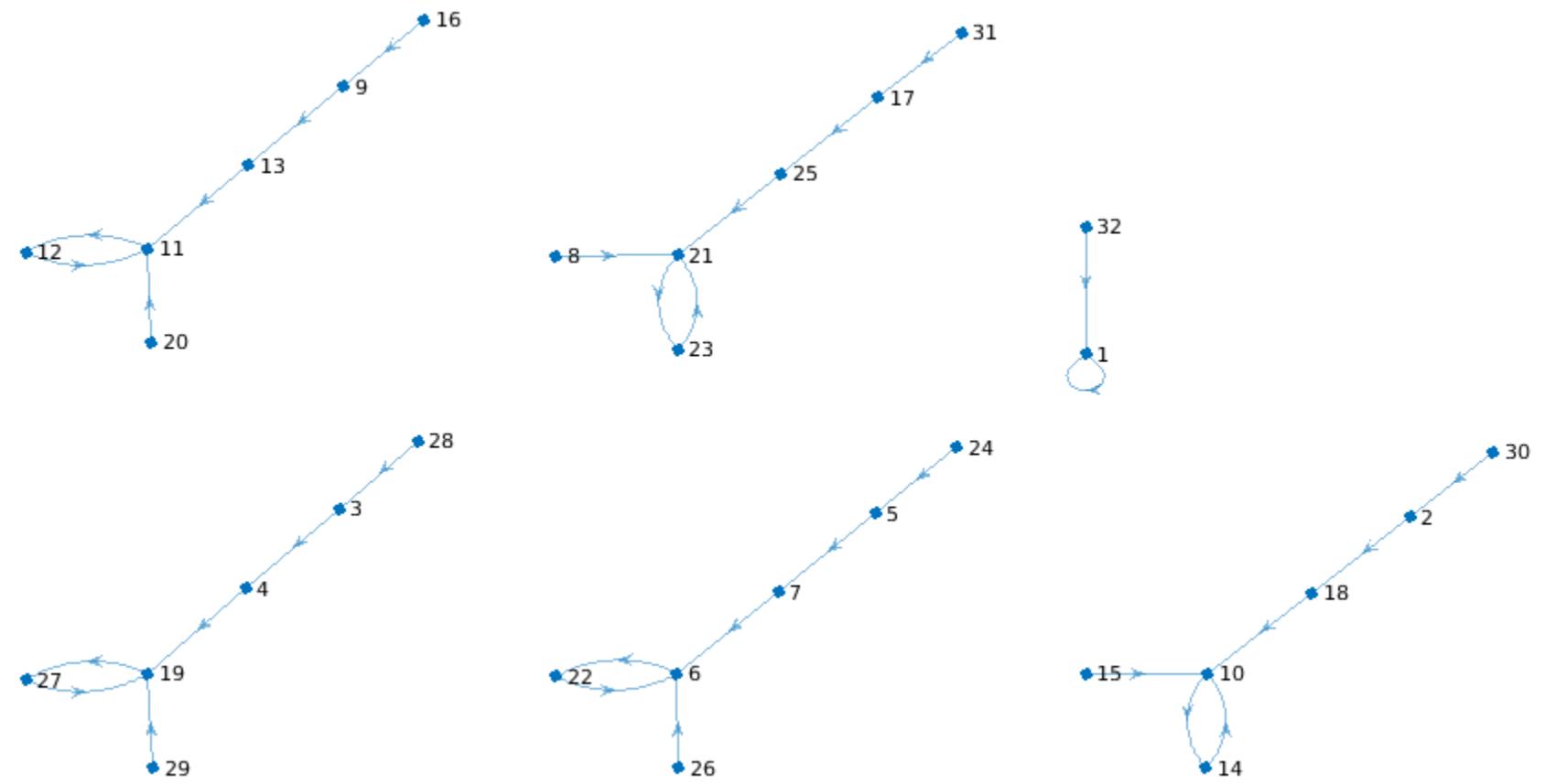
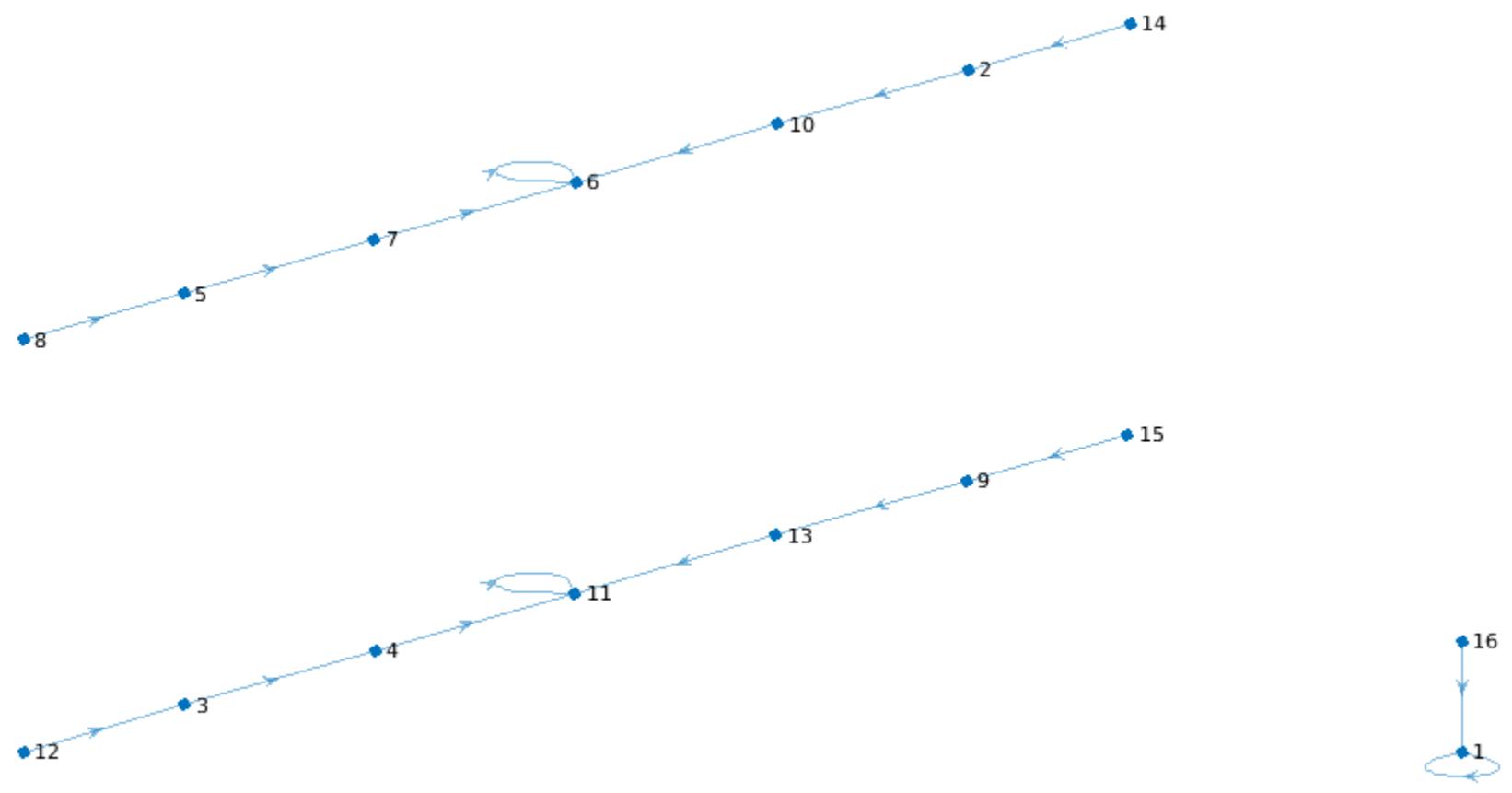


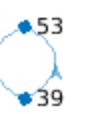
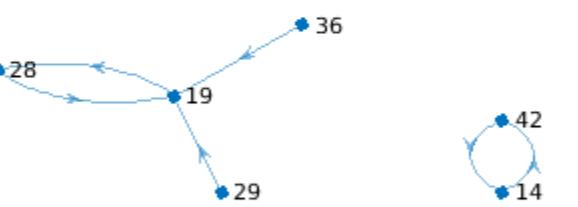
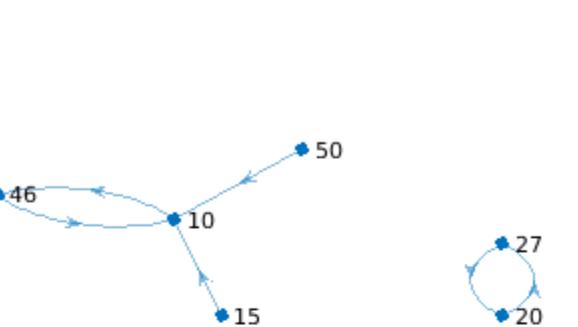
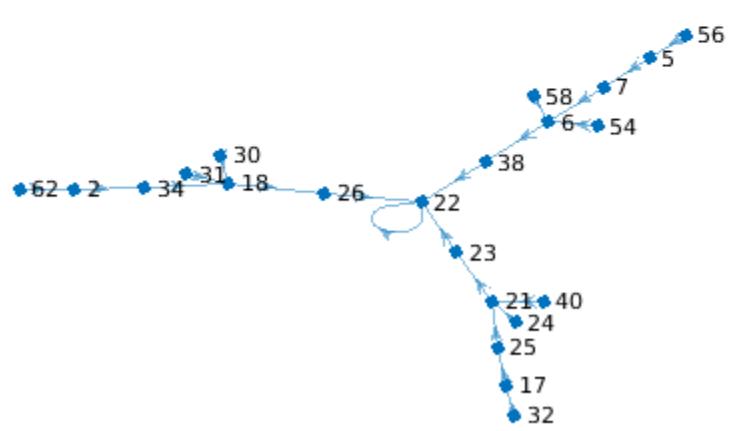
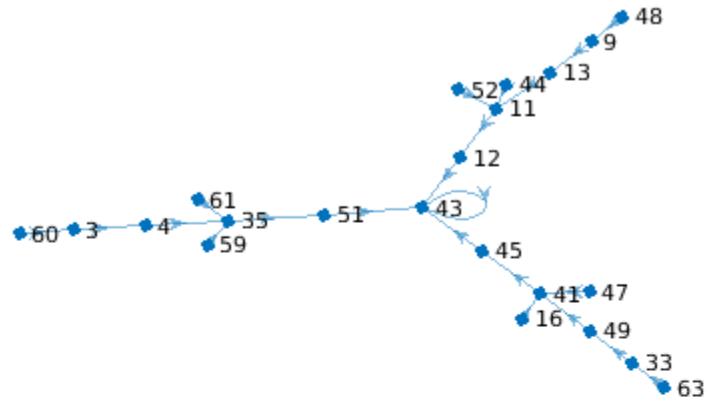


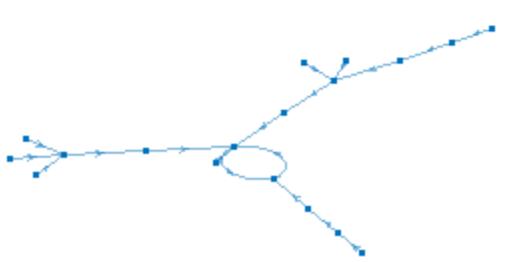
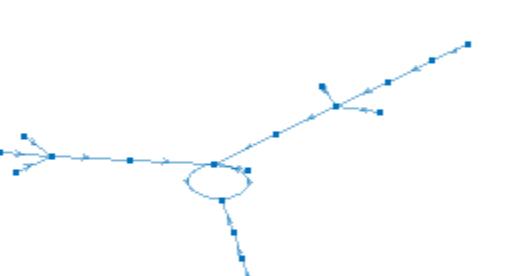
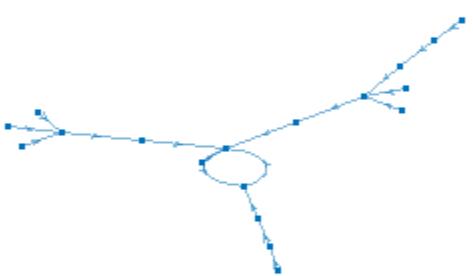
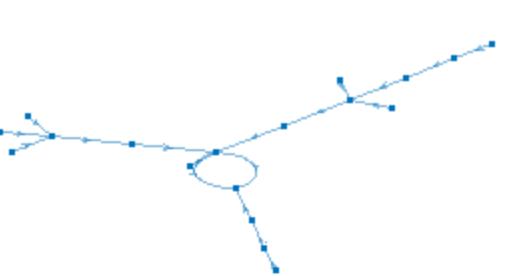
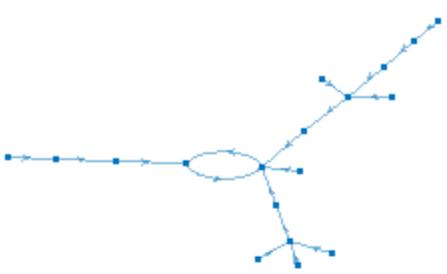


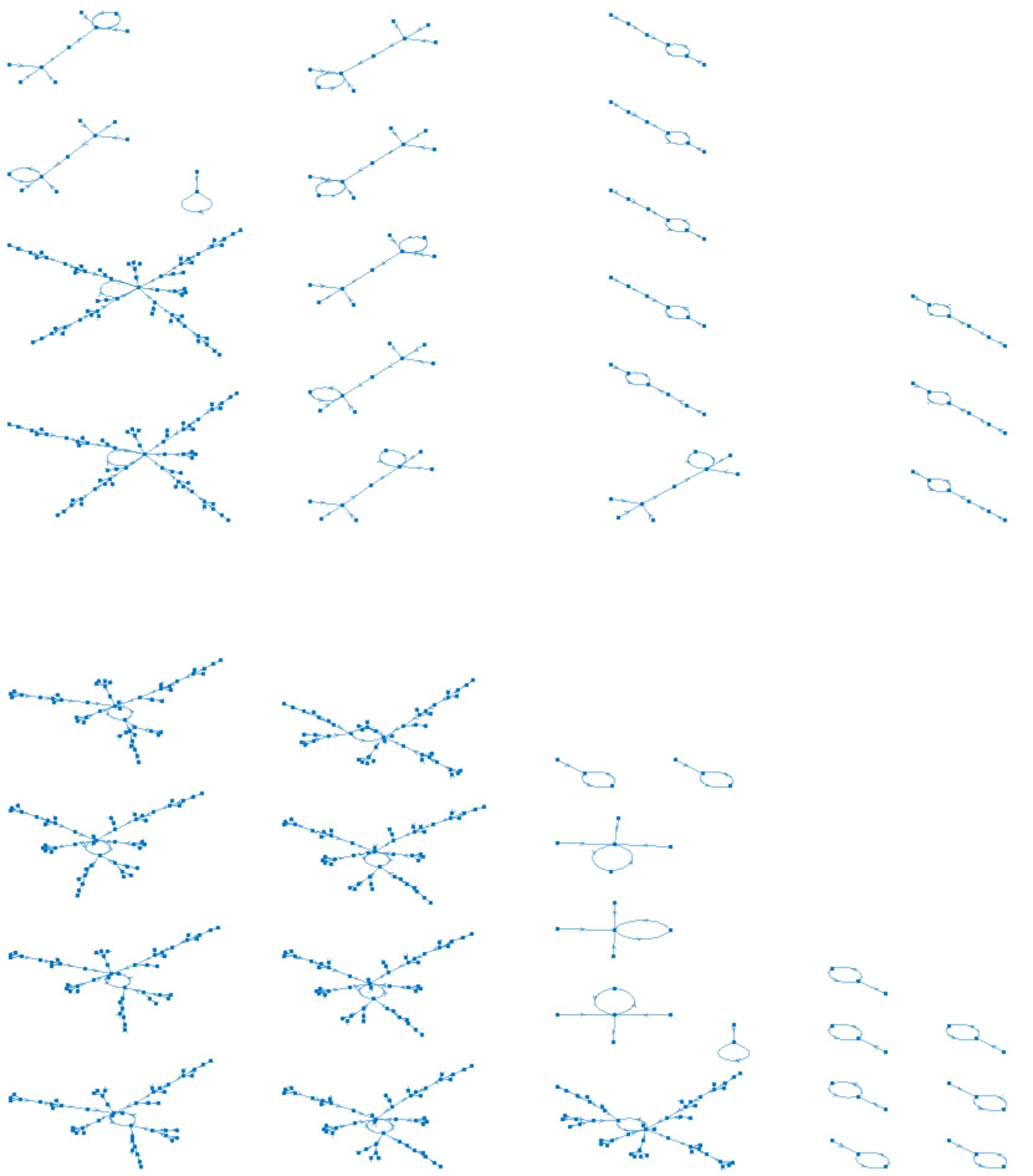


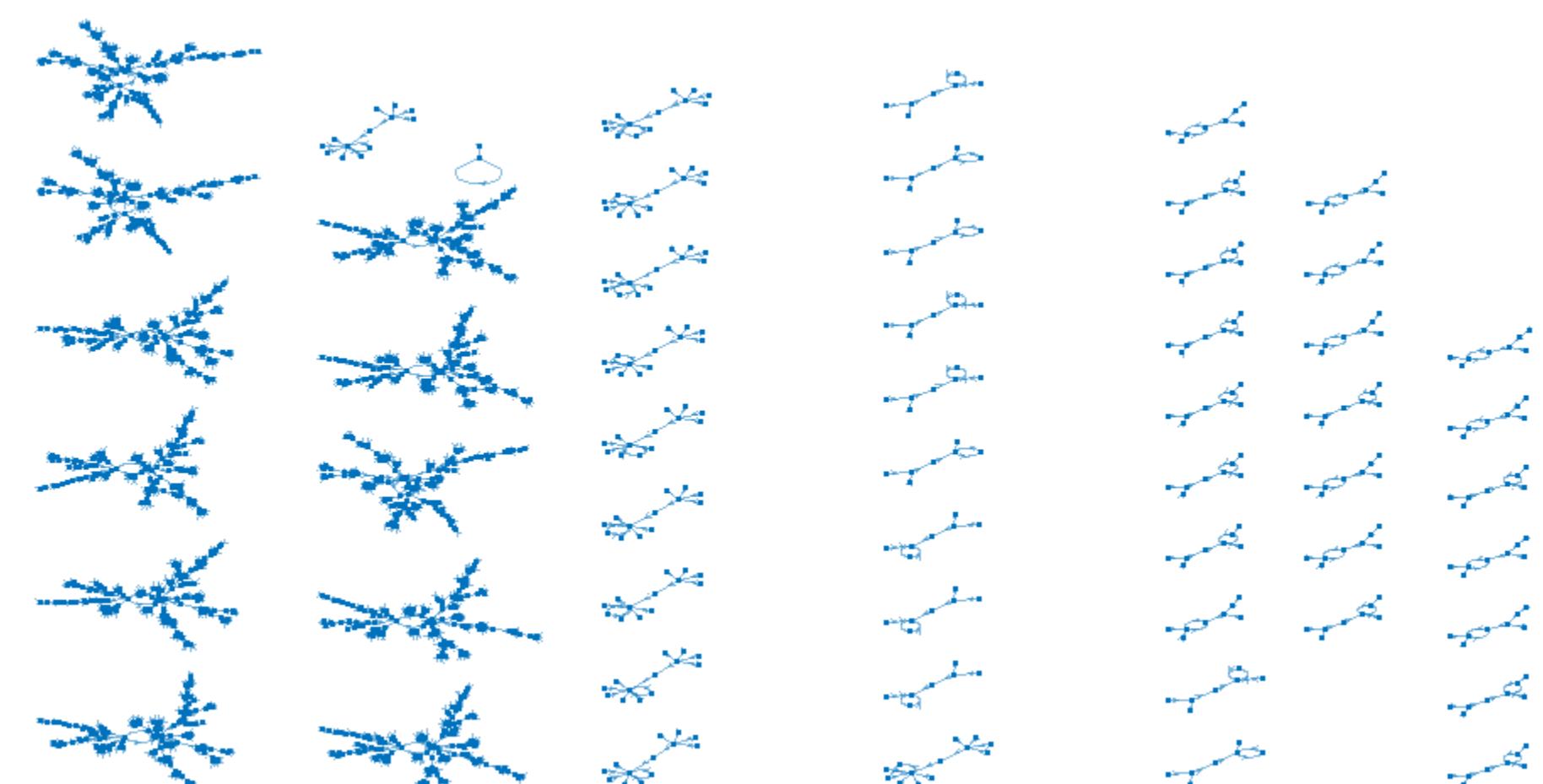
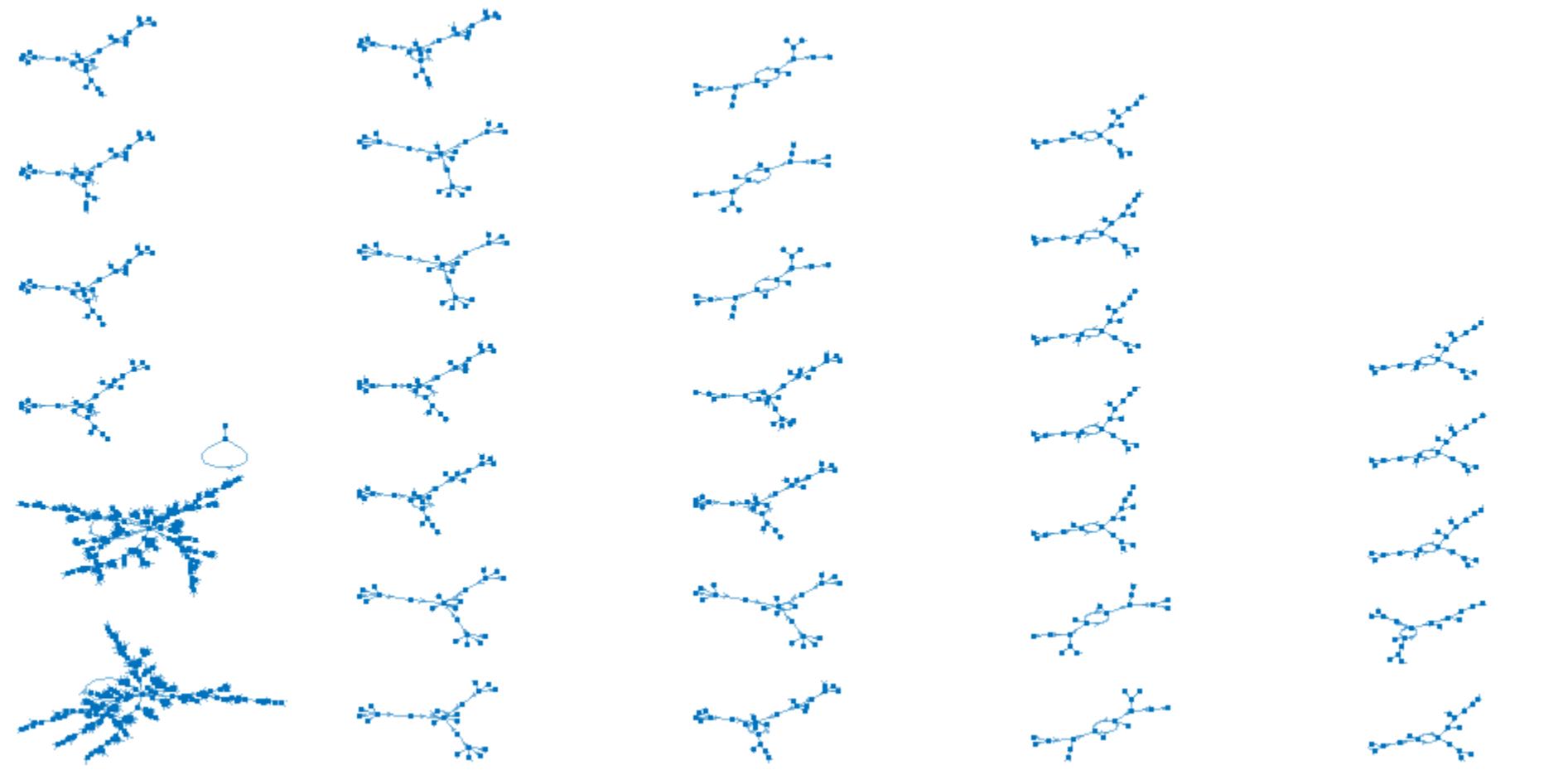


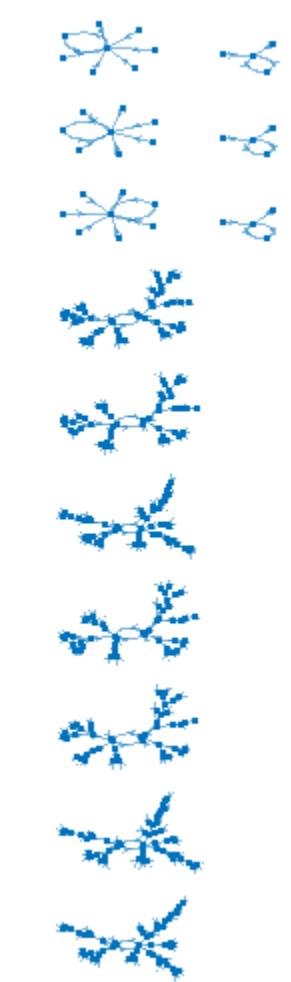
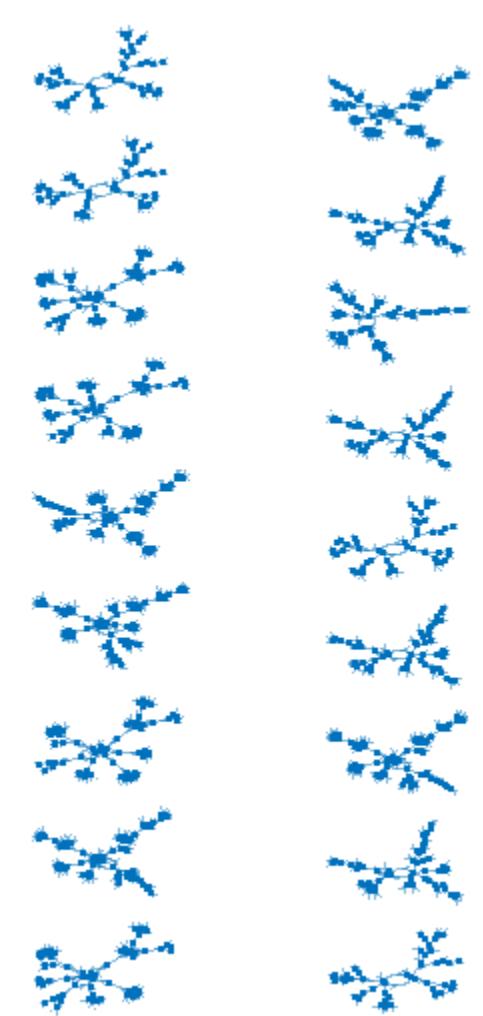
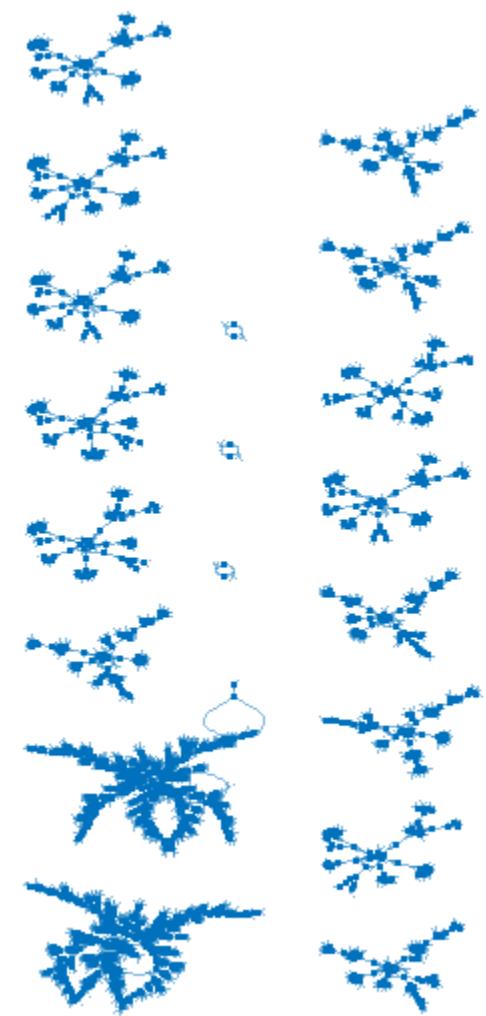












4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

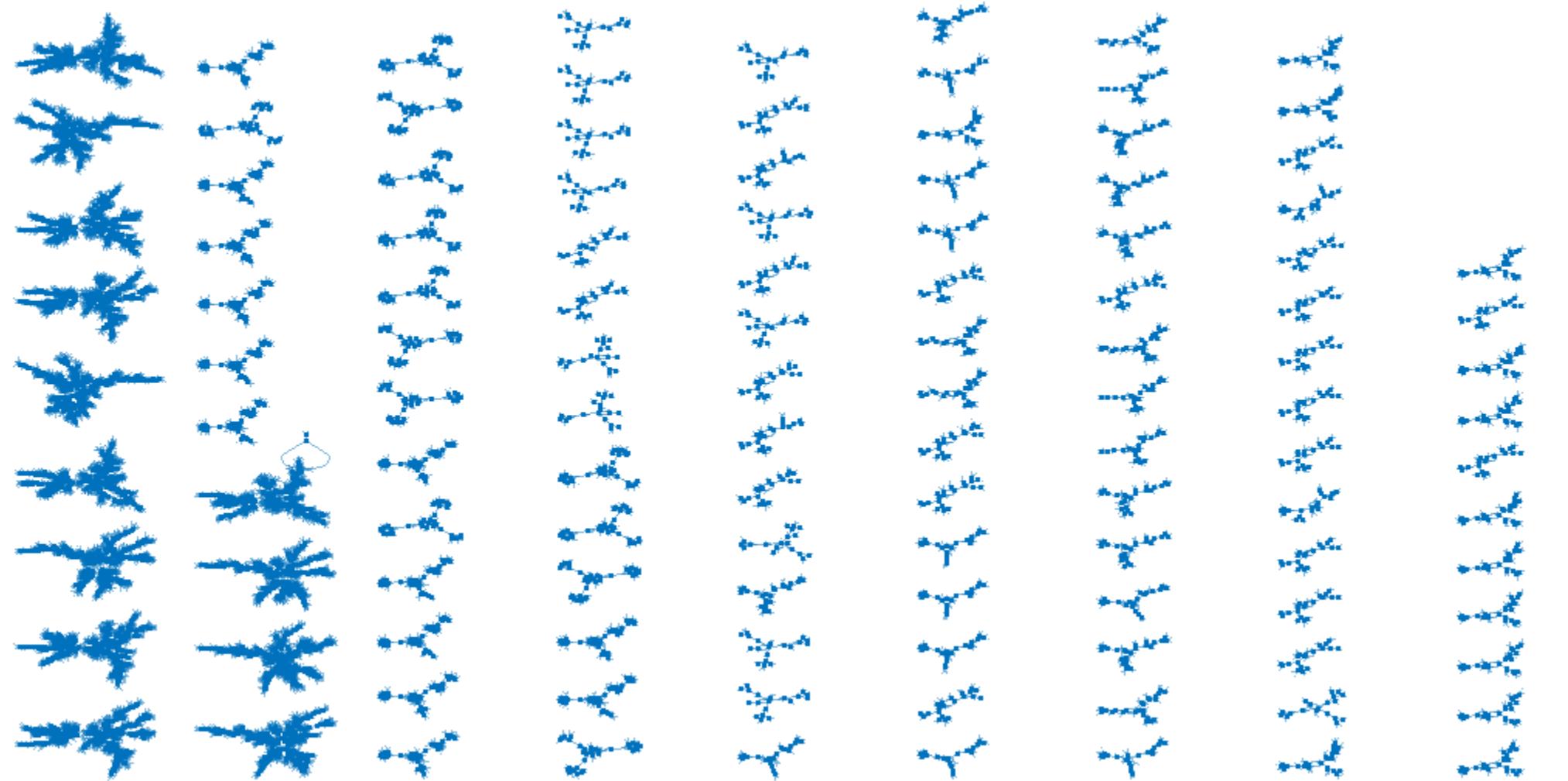
39

40

41

42

43



1986年1月1日
新嘉坡

新嘉坡植物园

新嘉坡植物园

新嘉坡植物园

新嘉坡植物园

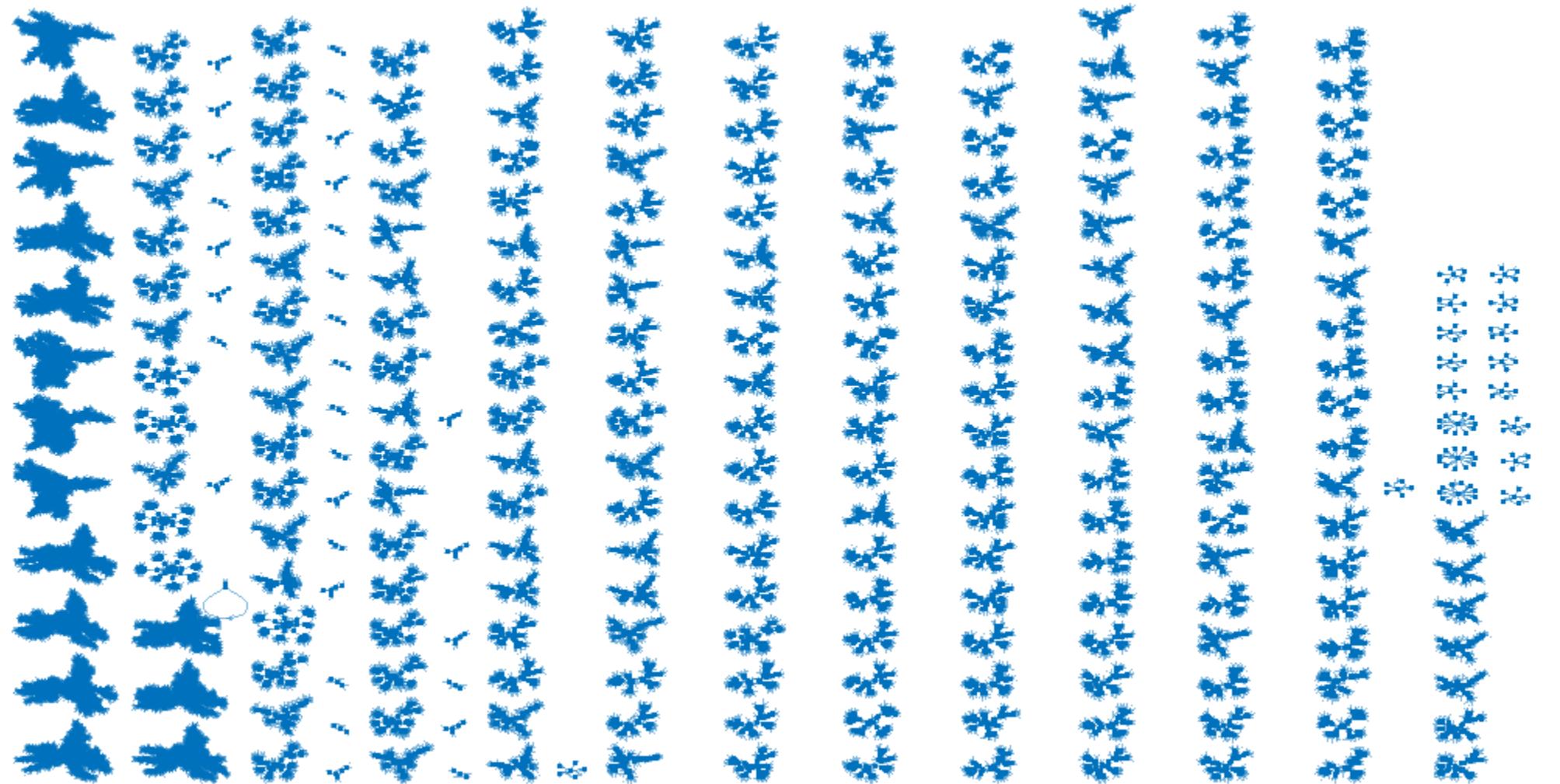
新嘉坡植物园

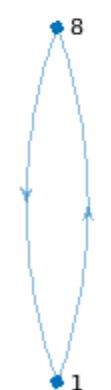
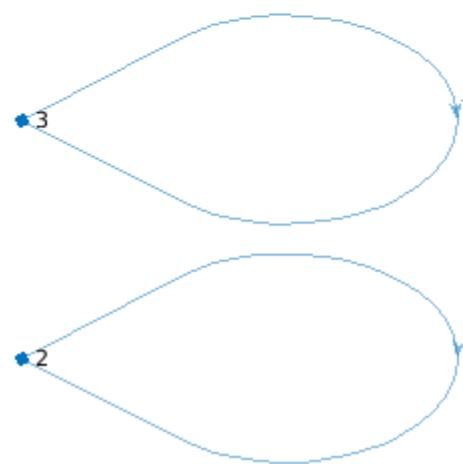
新嘉坡植物园

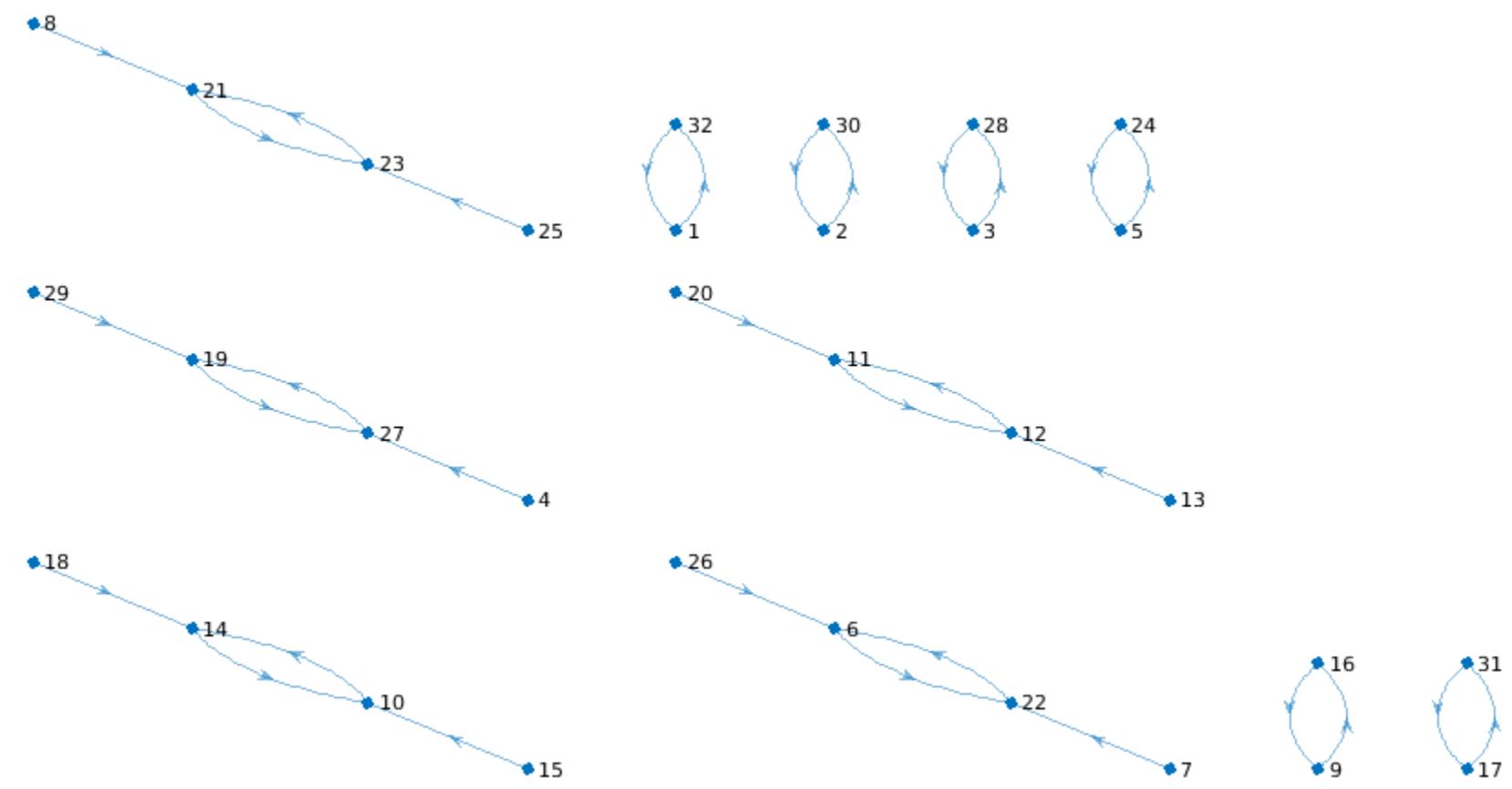
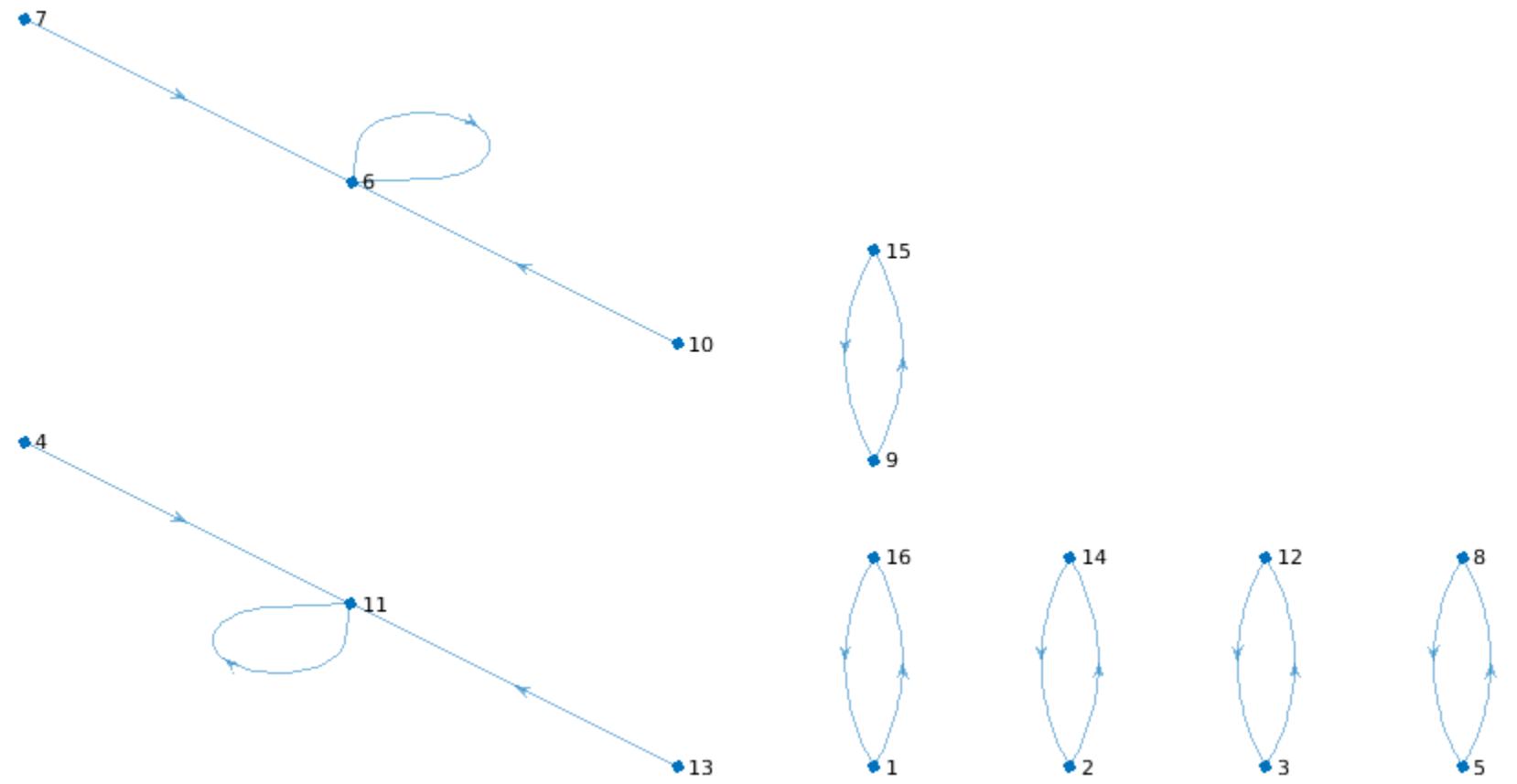
新嘉坡植物园

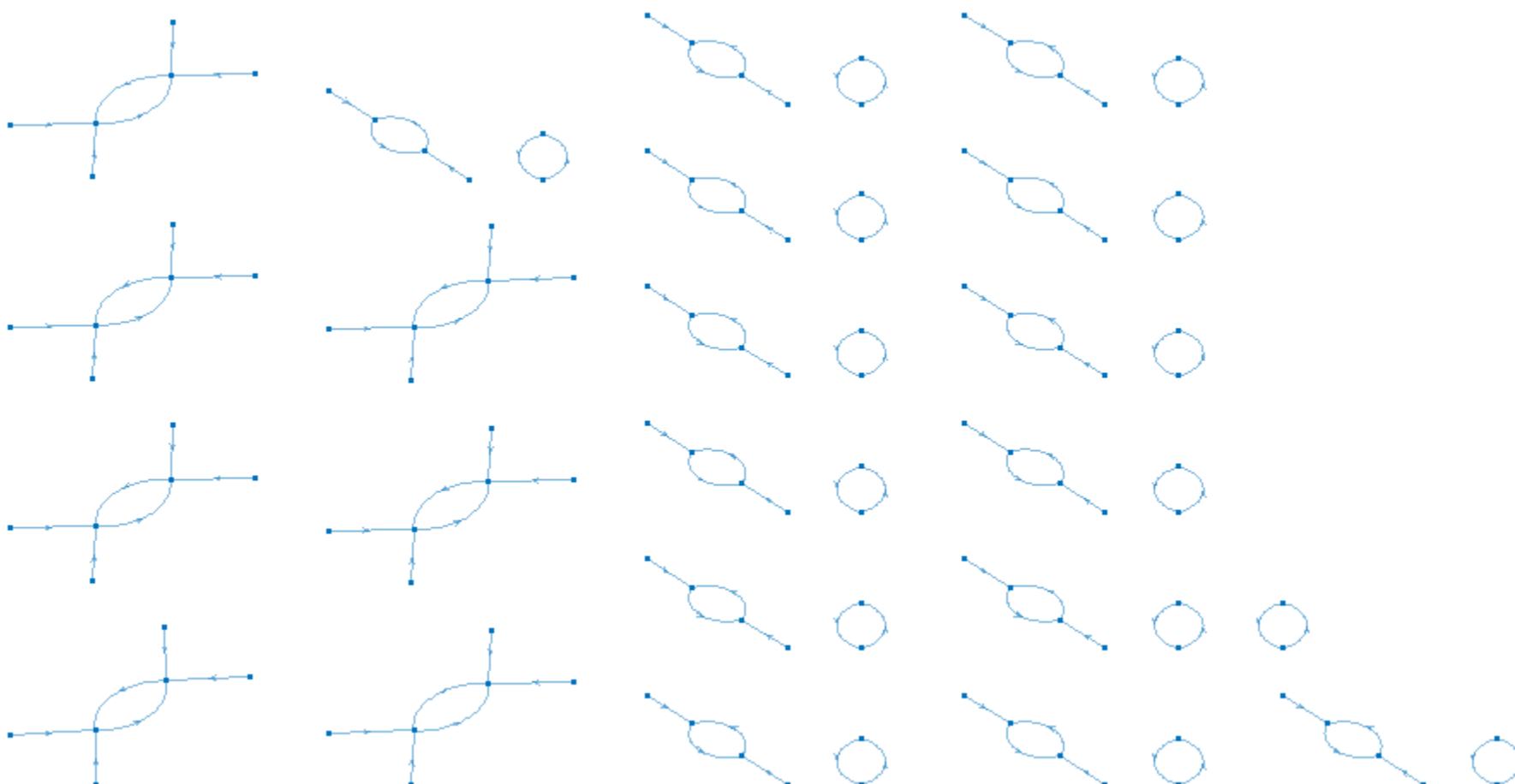
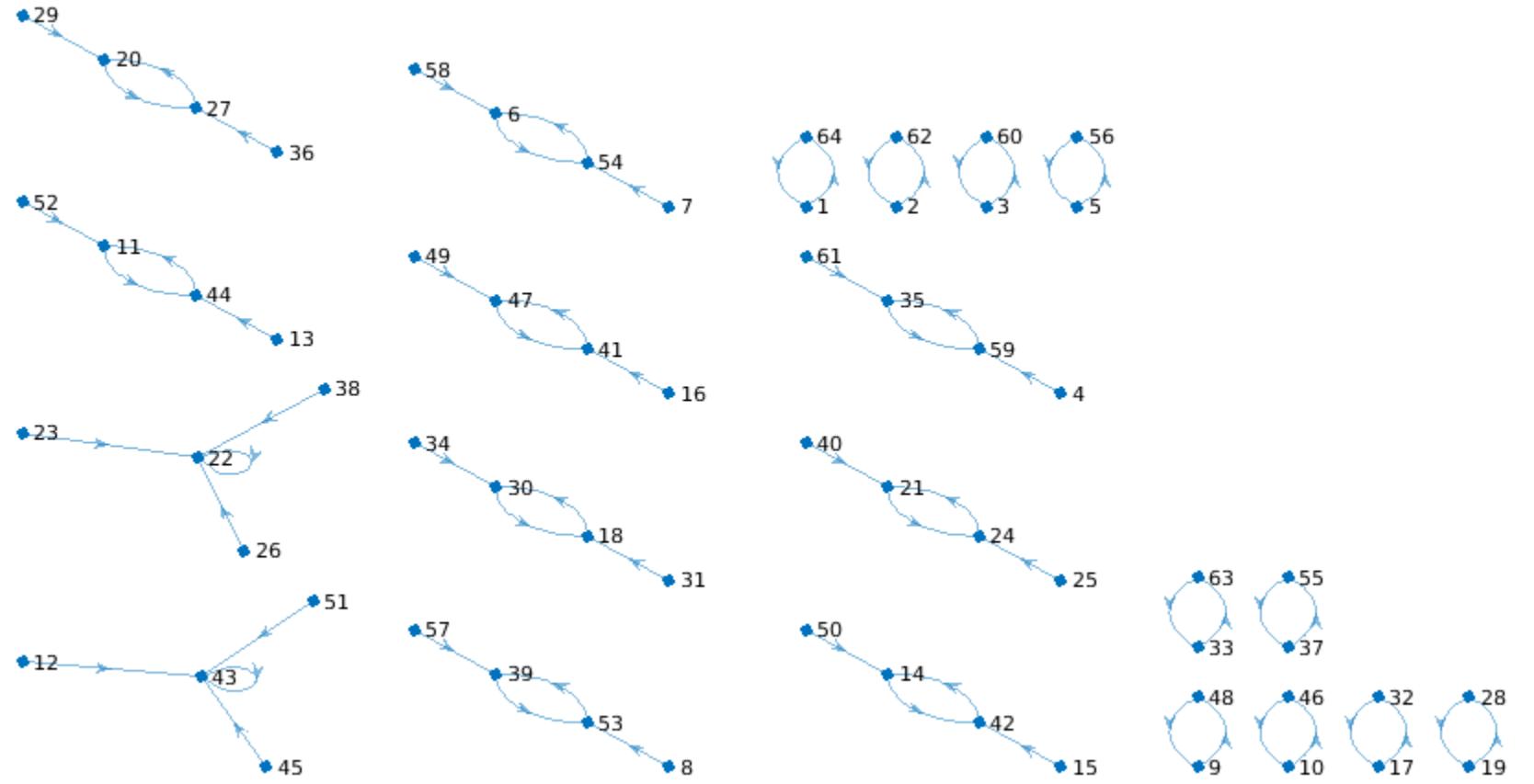
新嘉坡植物园

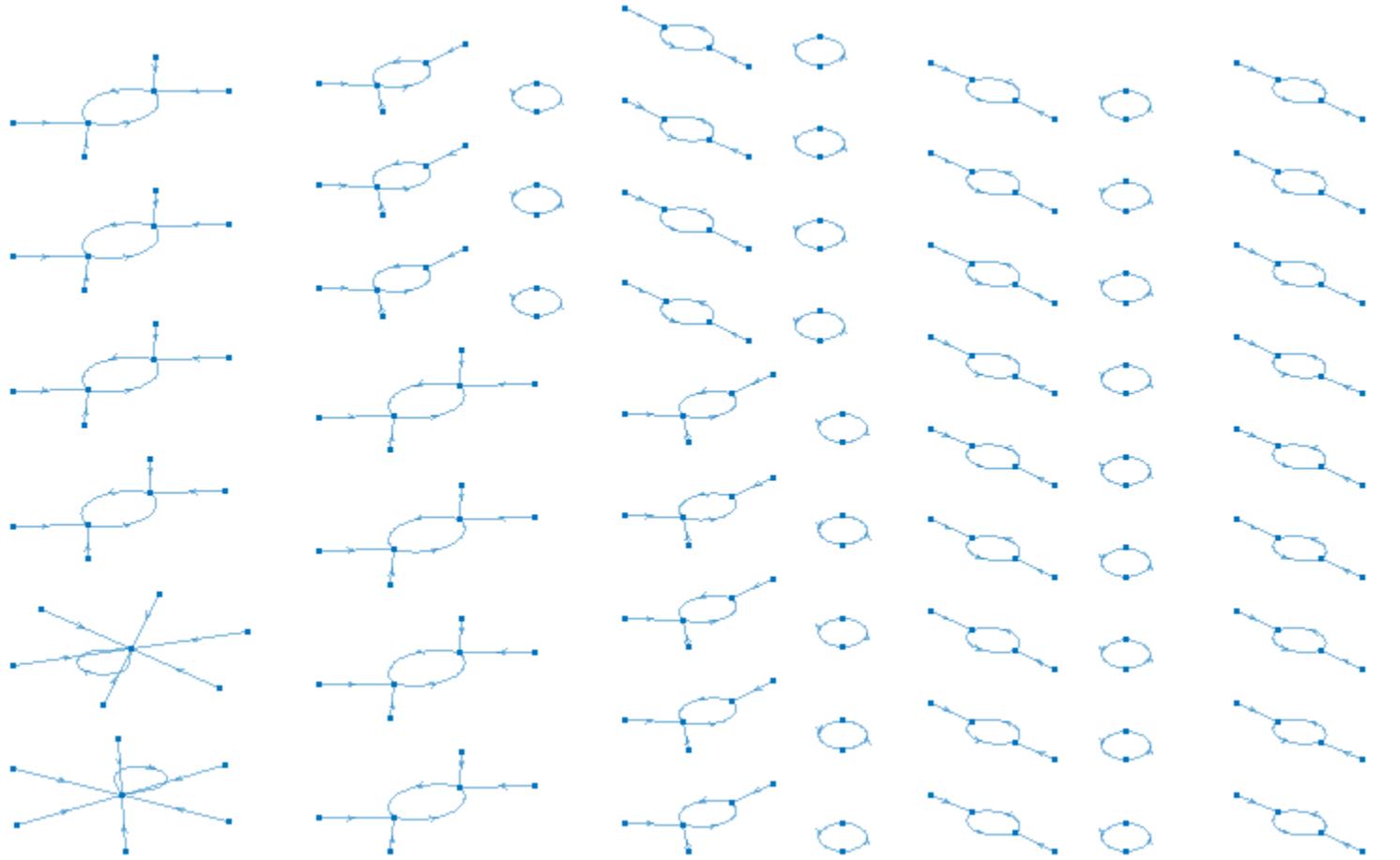
新嘉坡植物园

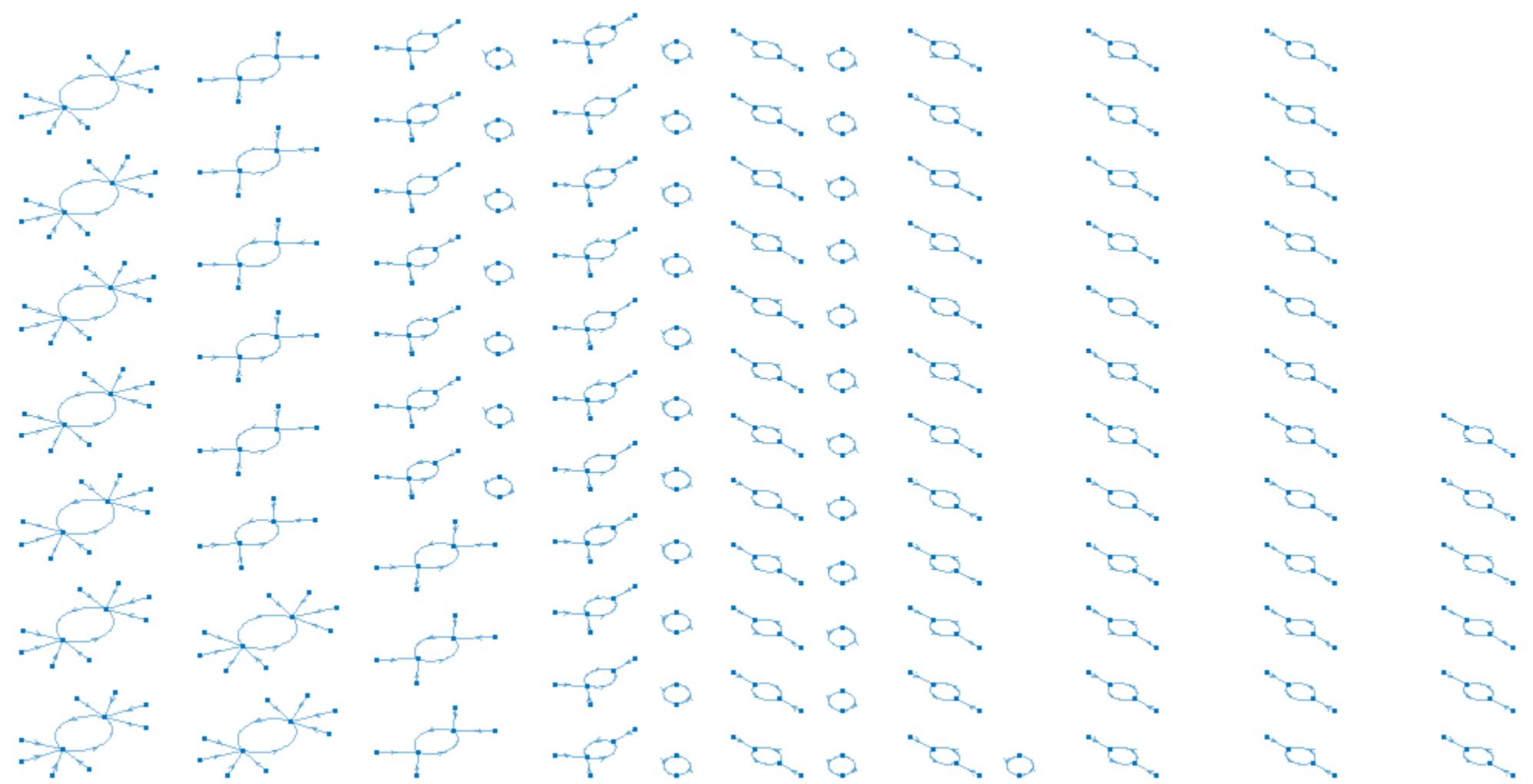


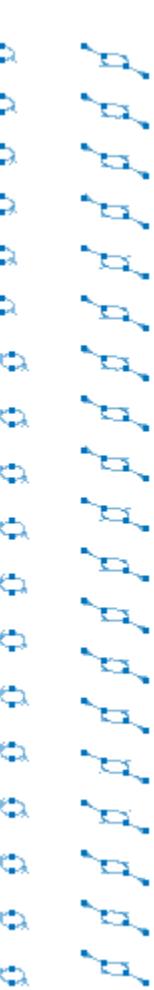
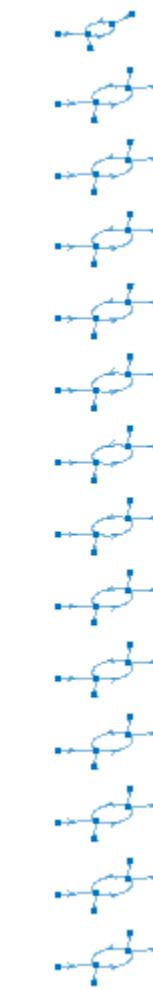
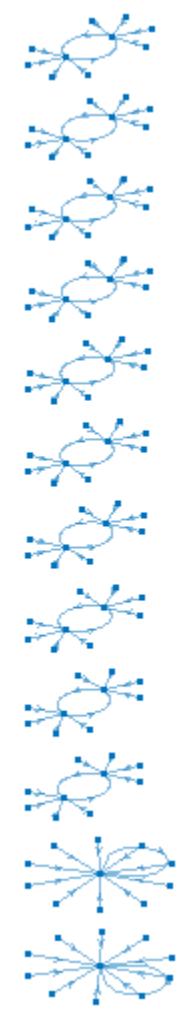












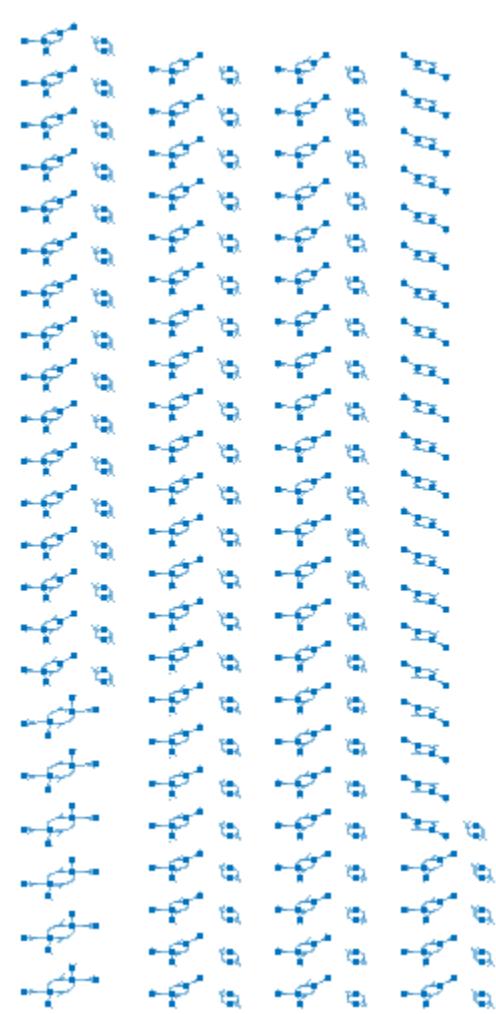
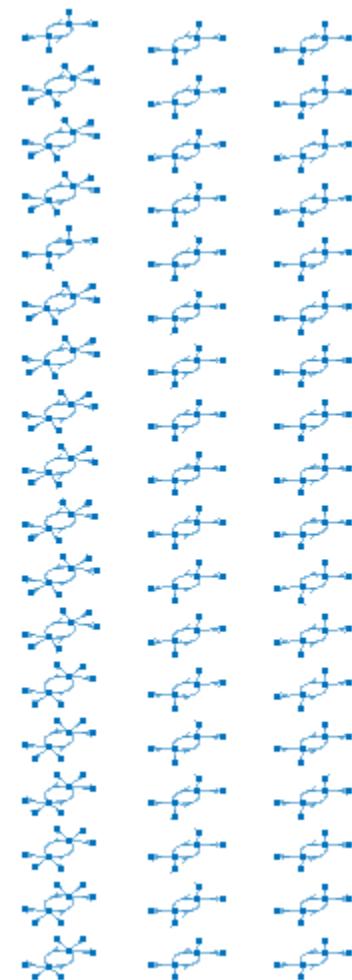
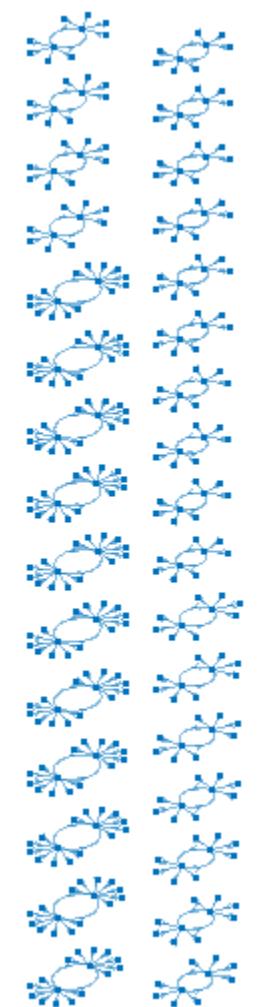
p' p' p' p' p' p' p' p' p' p'

p' p' p' p' p' p' p' p' p' p'

p' p' p' p' p' p' p' p' p' p'

p' p' p' p' p' p' p' p' p' p'

p' p' p' p' p' p' p' p' p'



$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

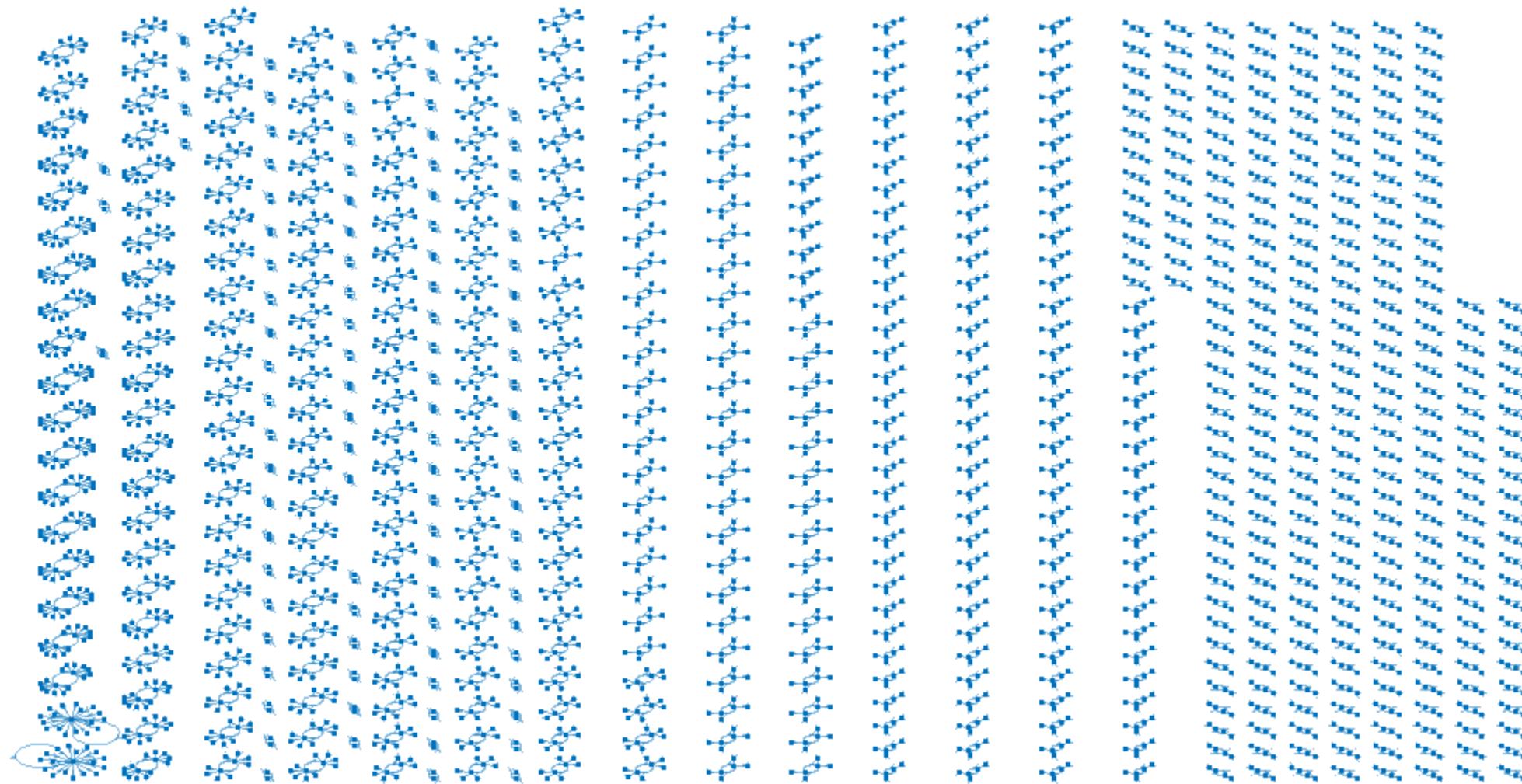
$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

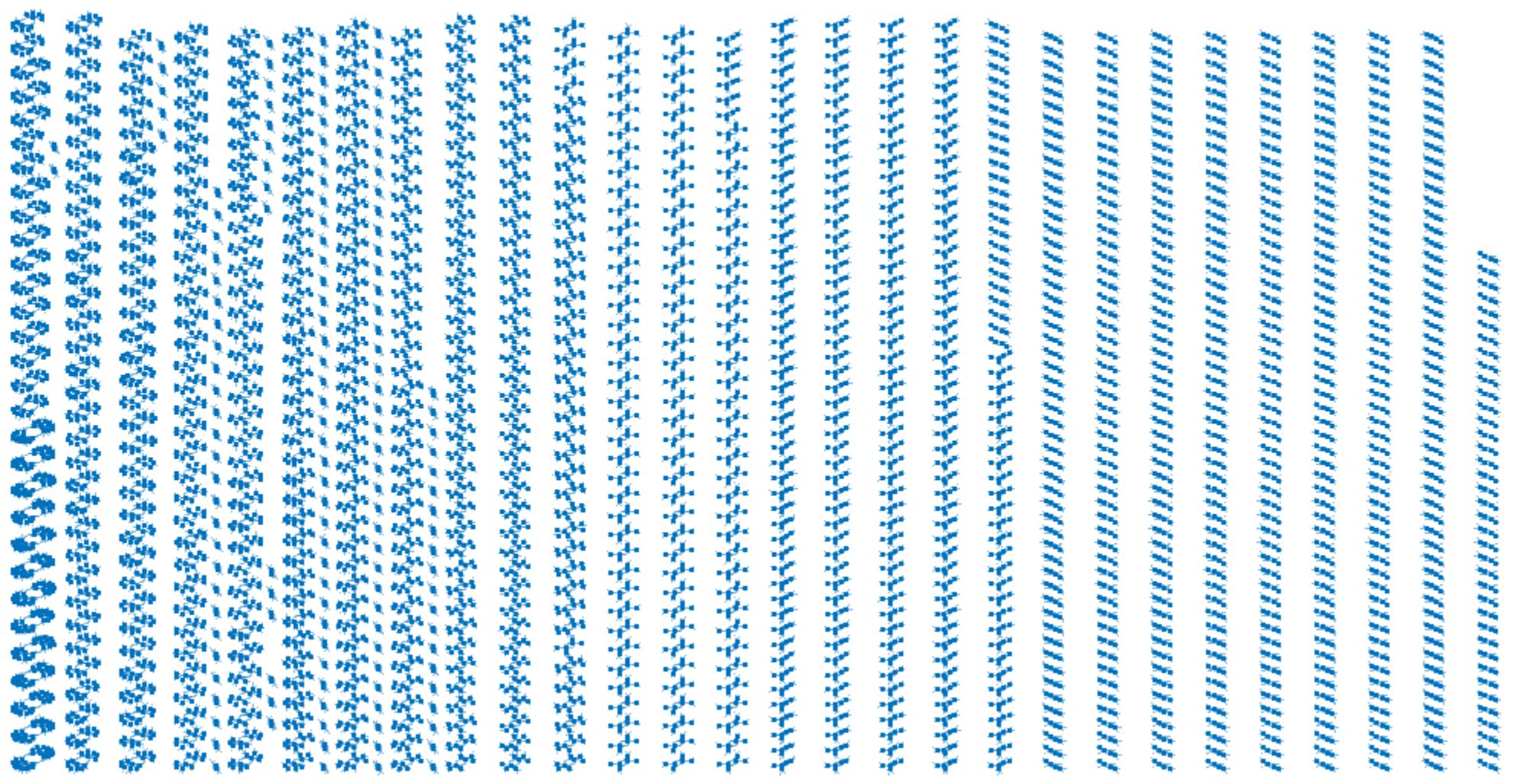
$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

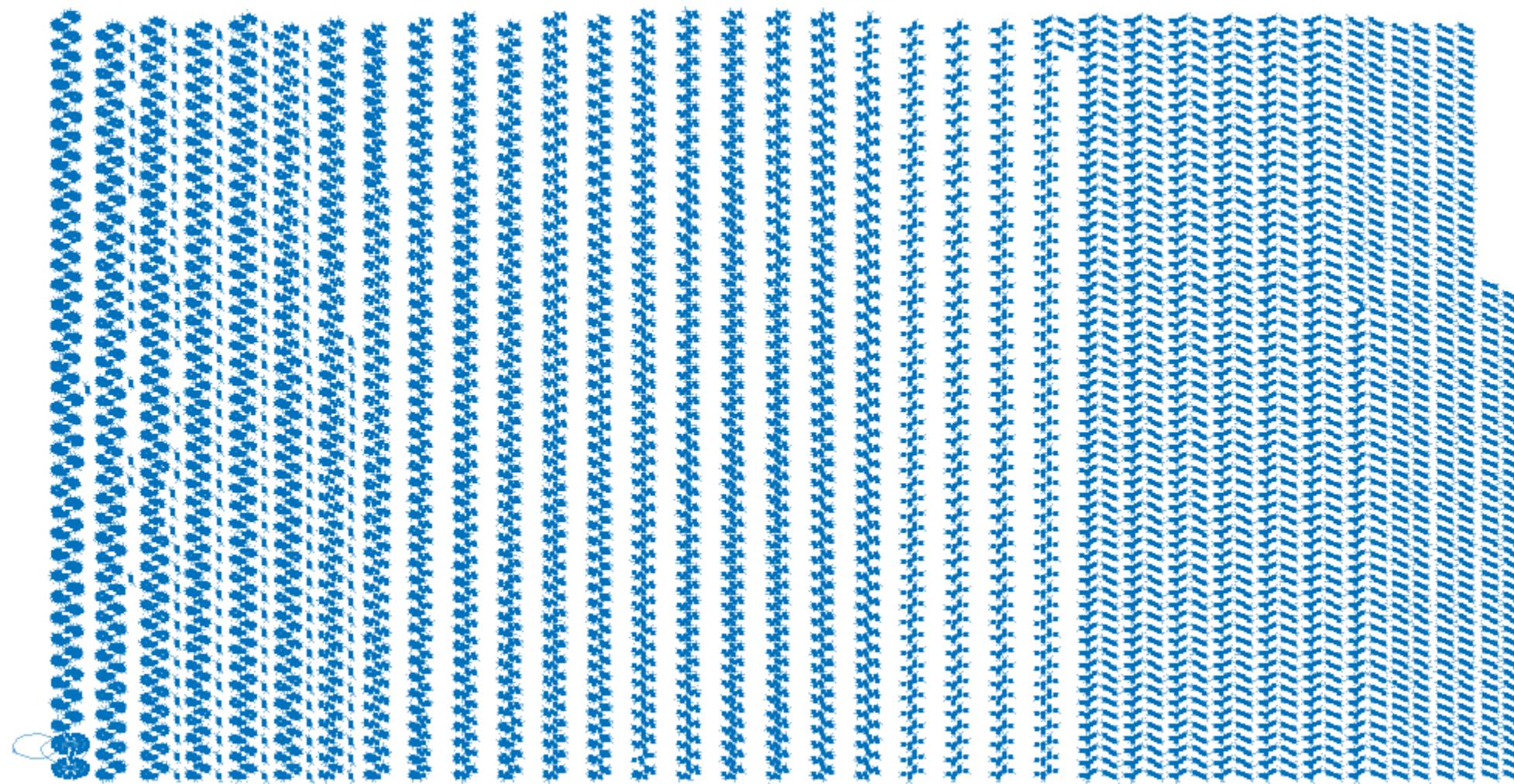
$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

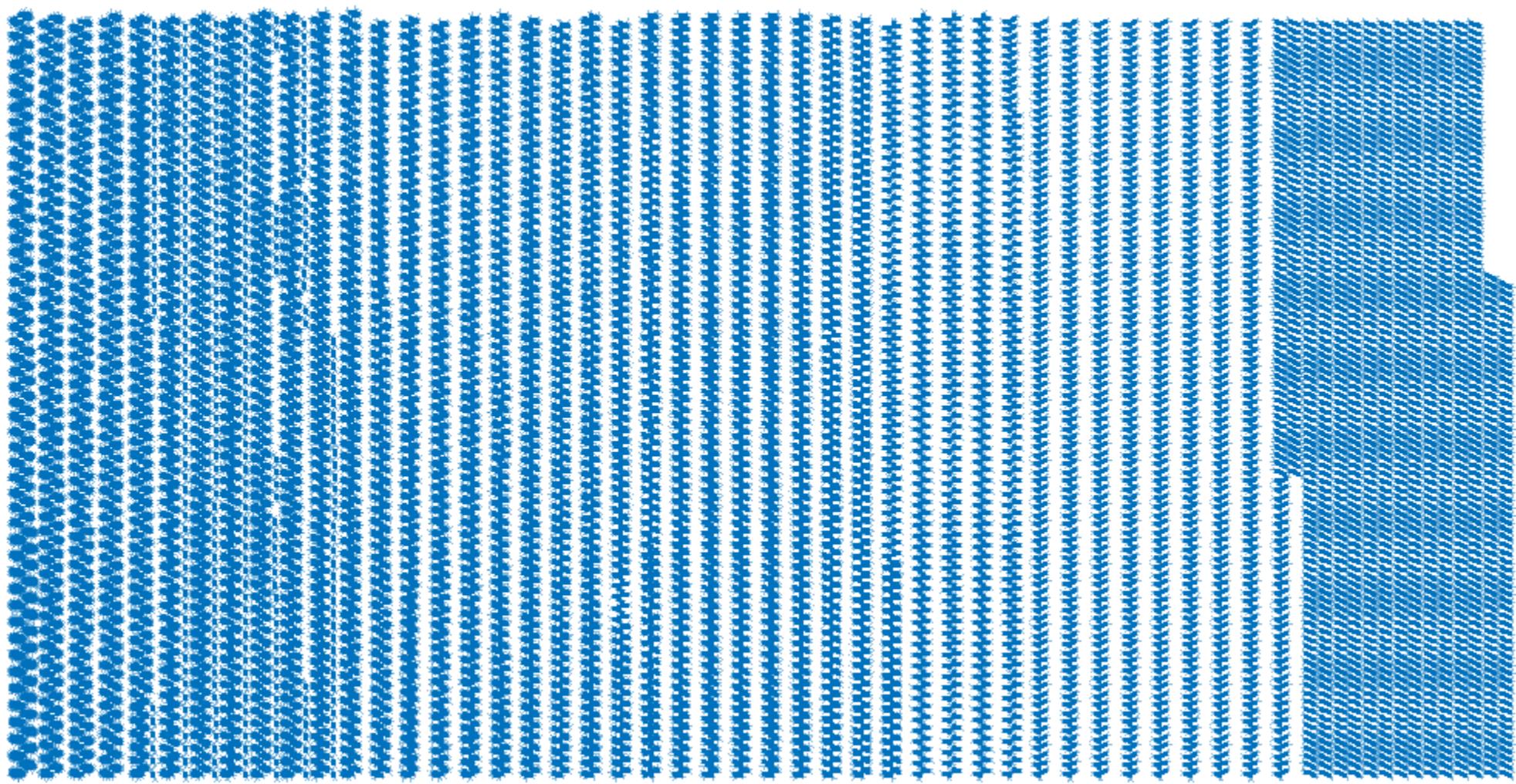
$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

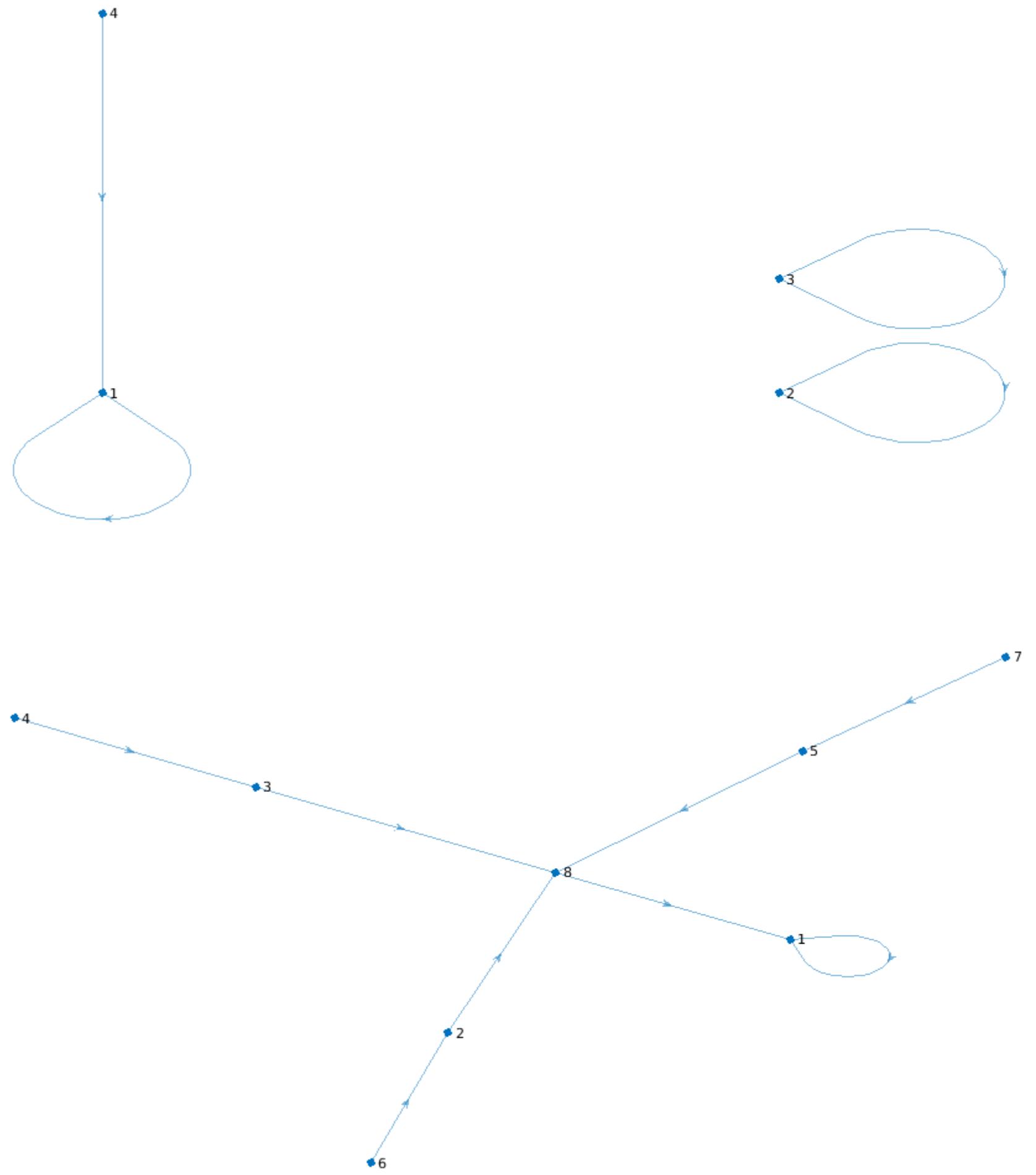
$\beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta' \beta'$

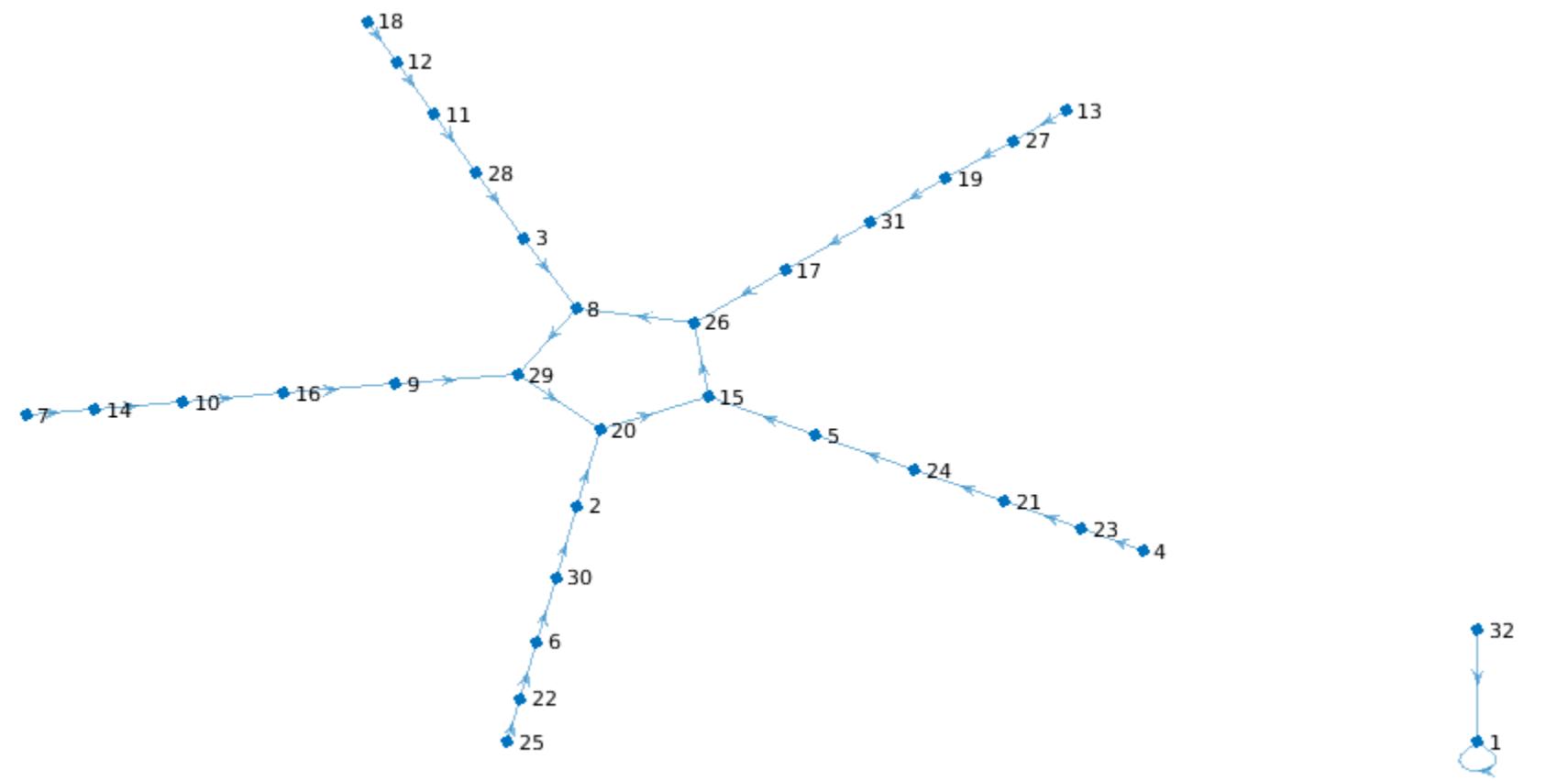
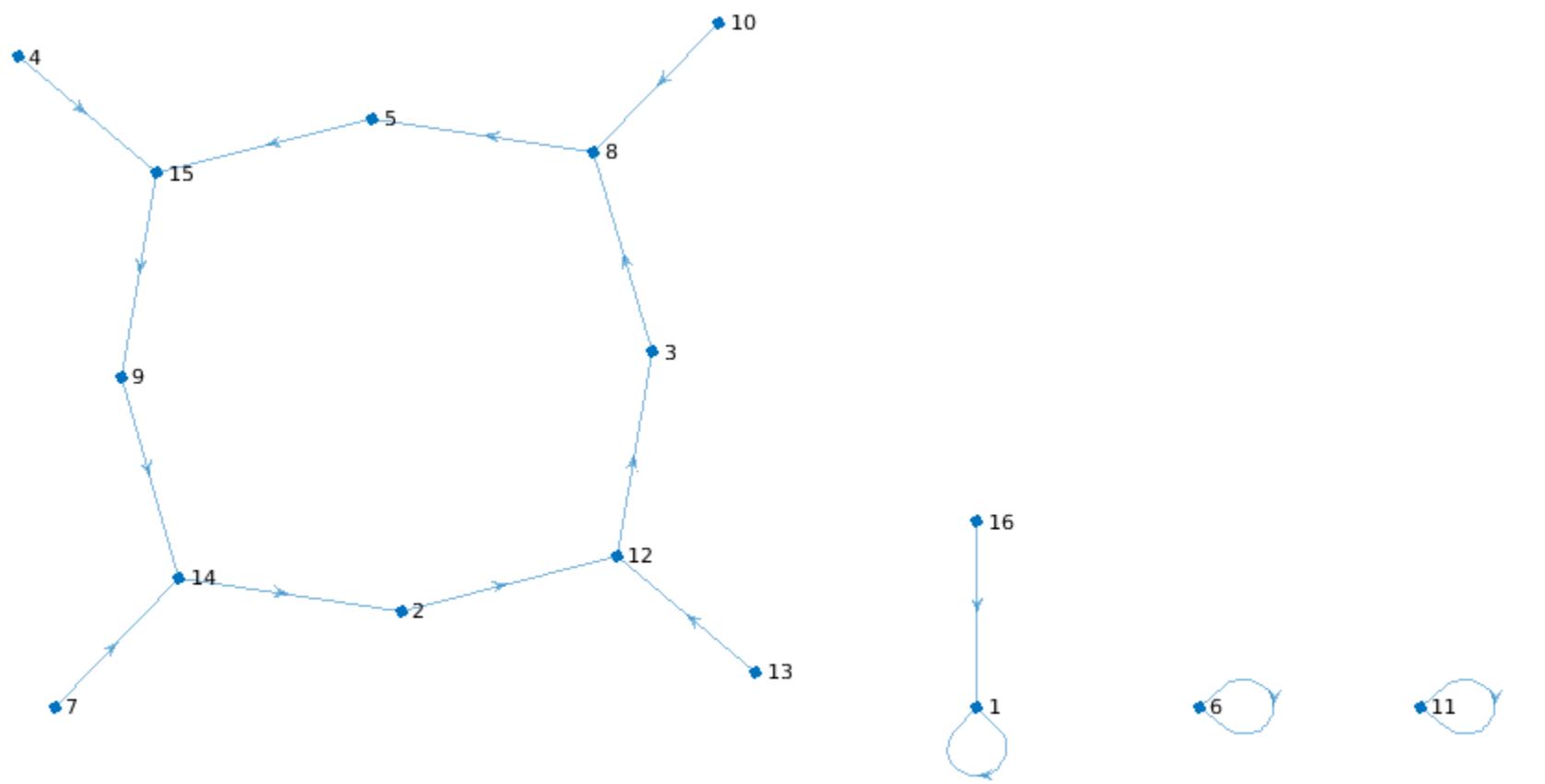


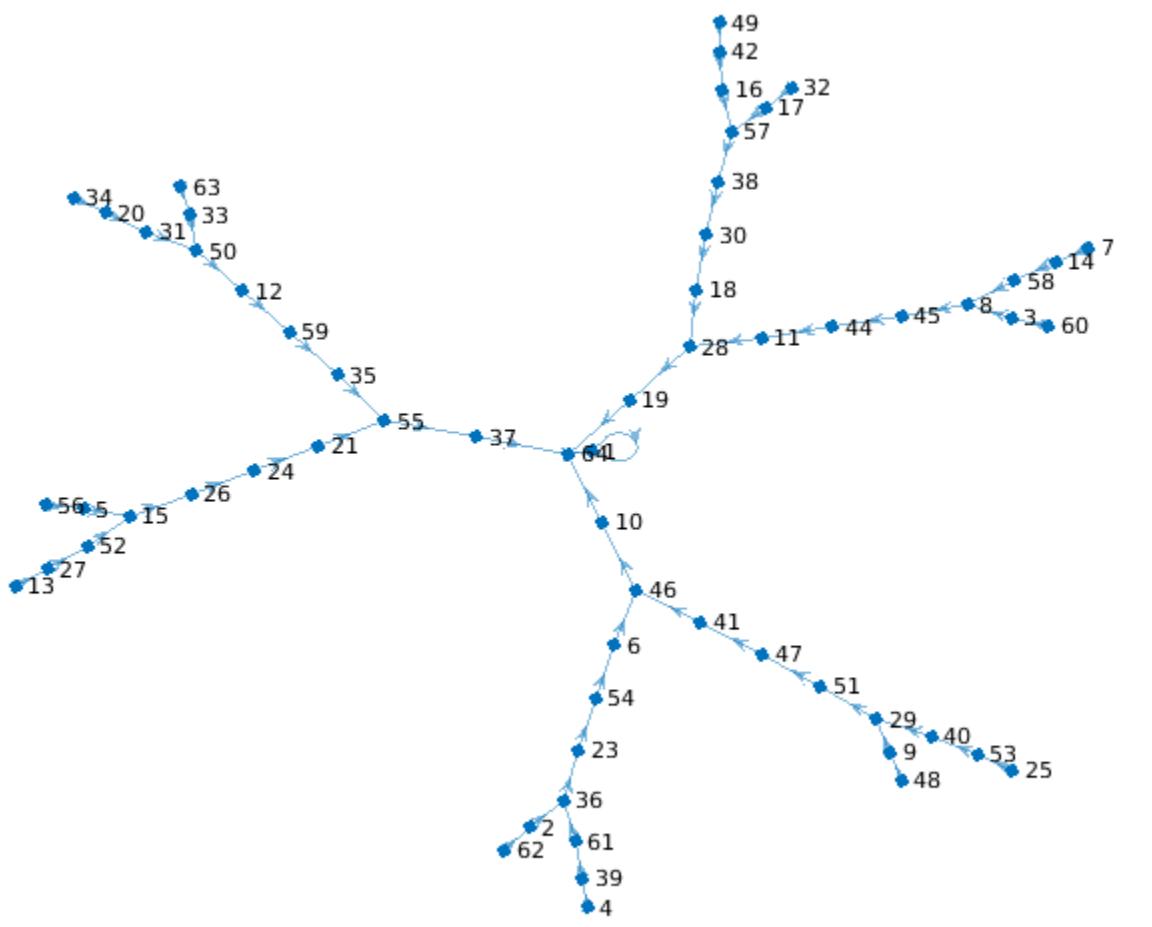


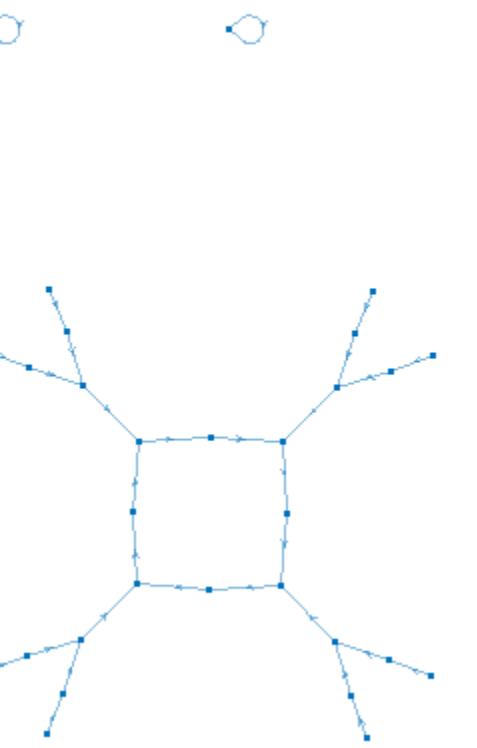
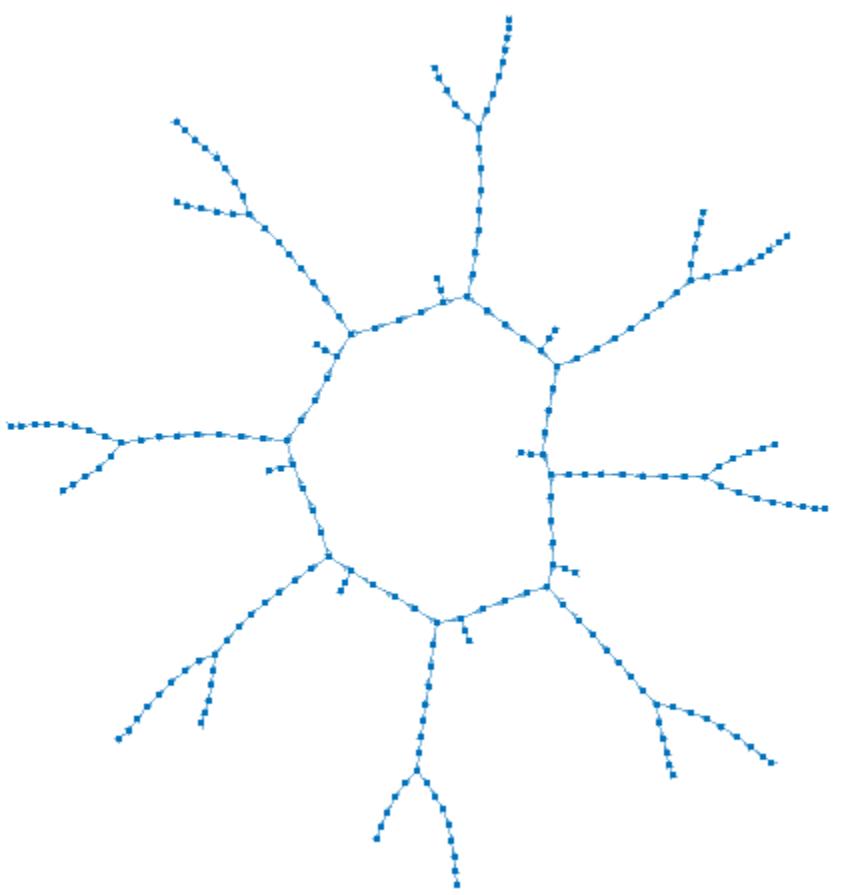
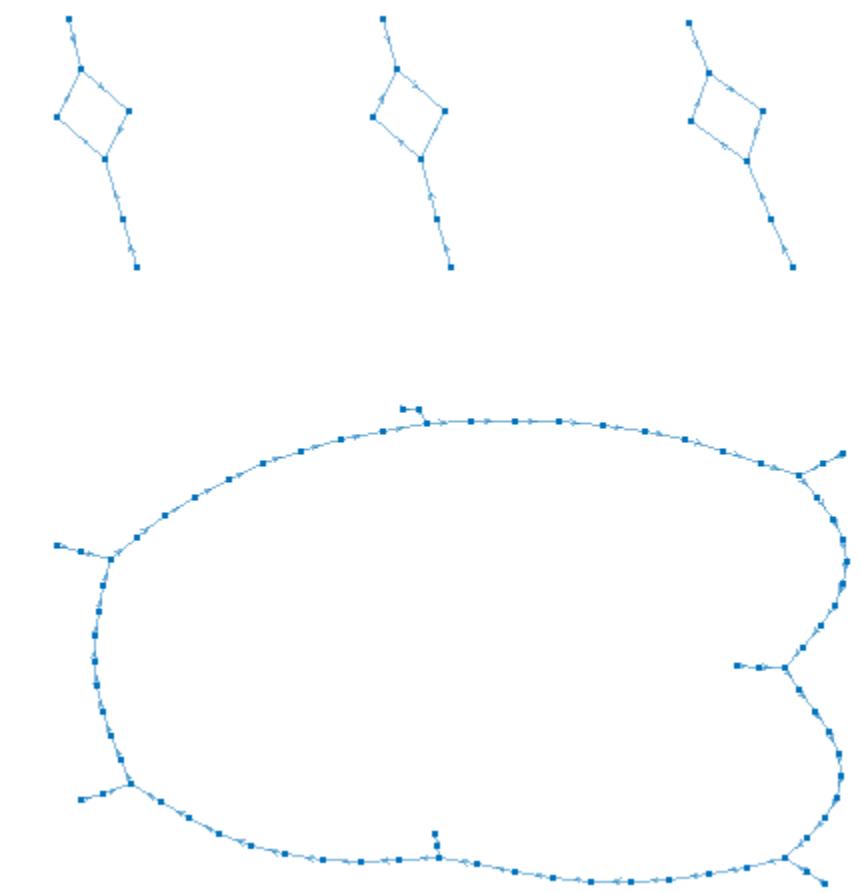




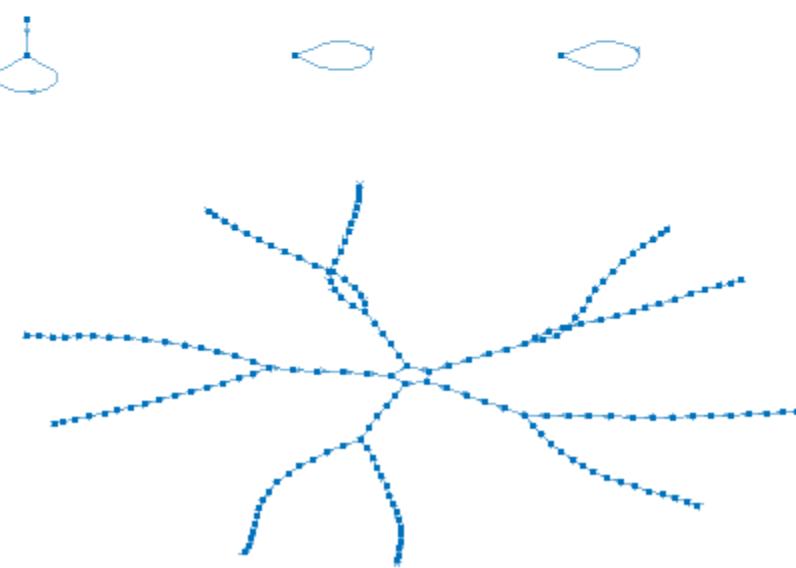
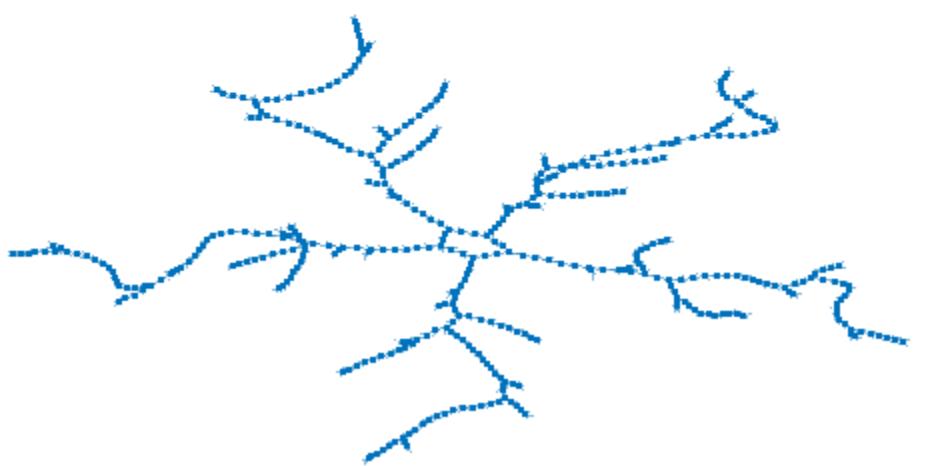
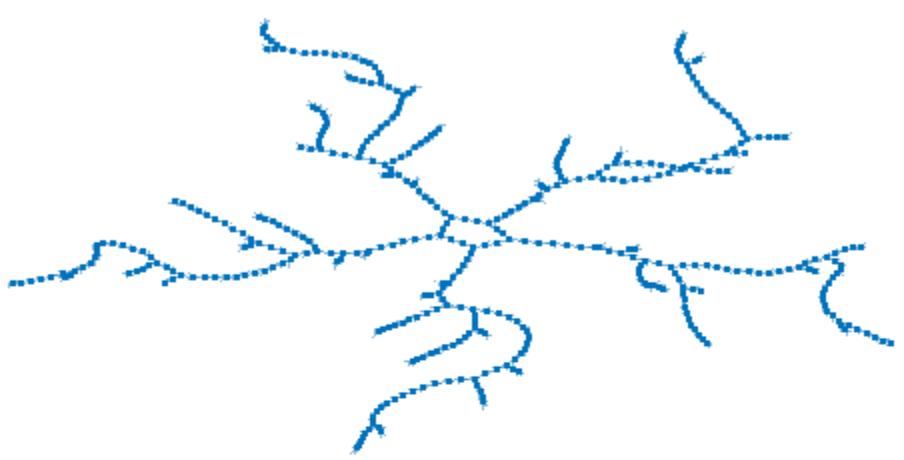
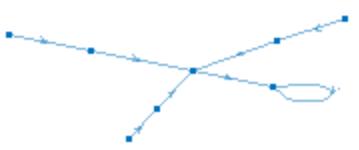
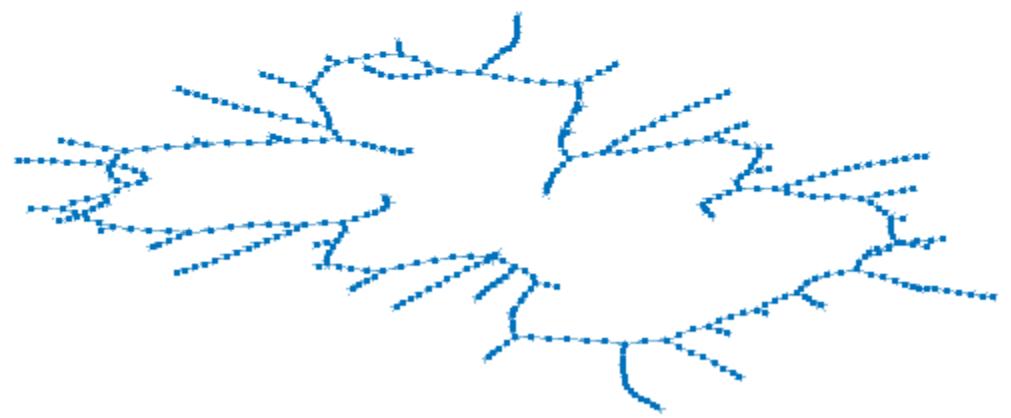
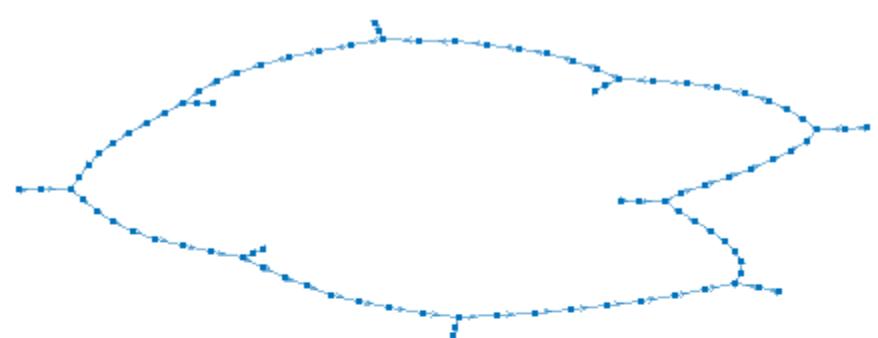


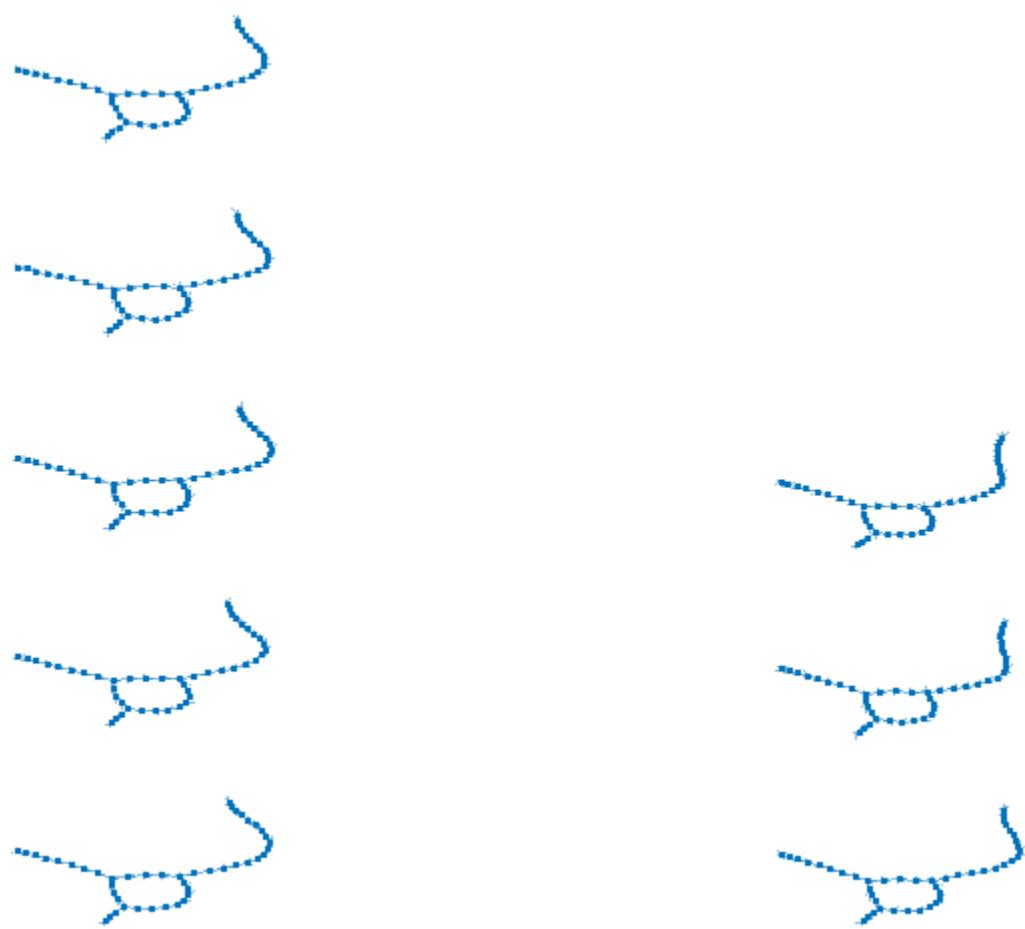
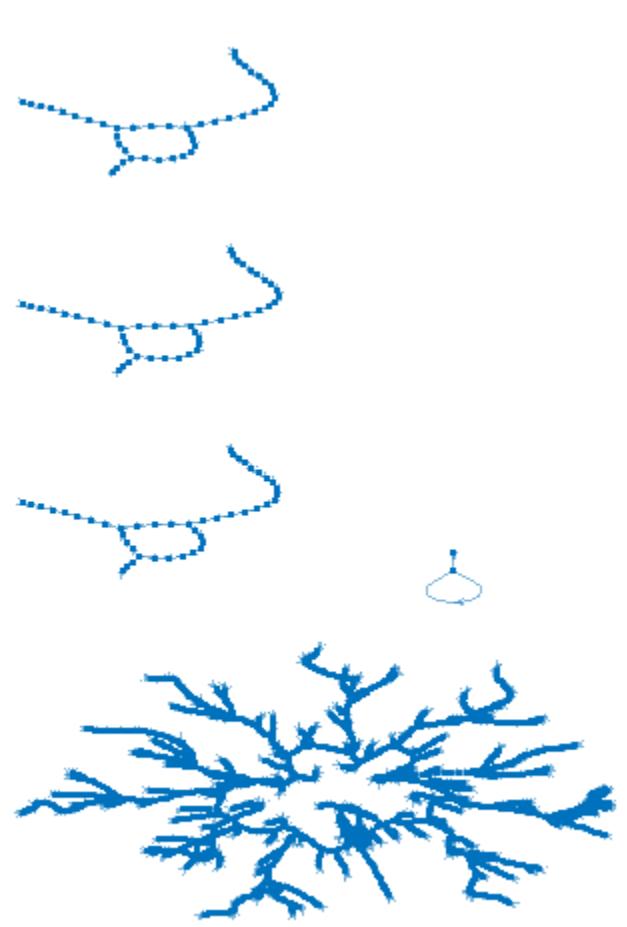


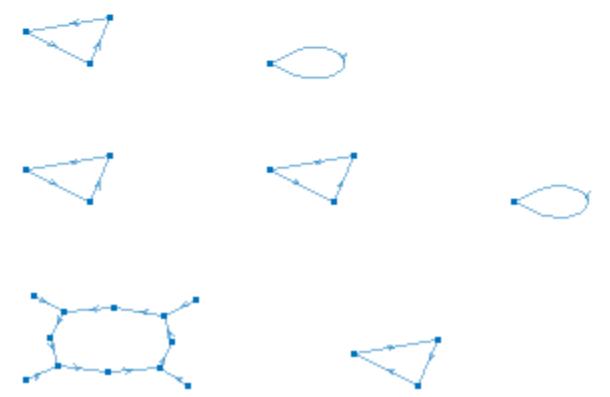
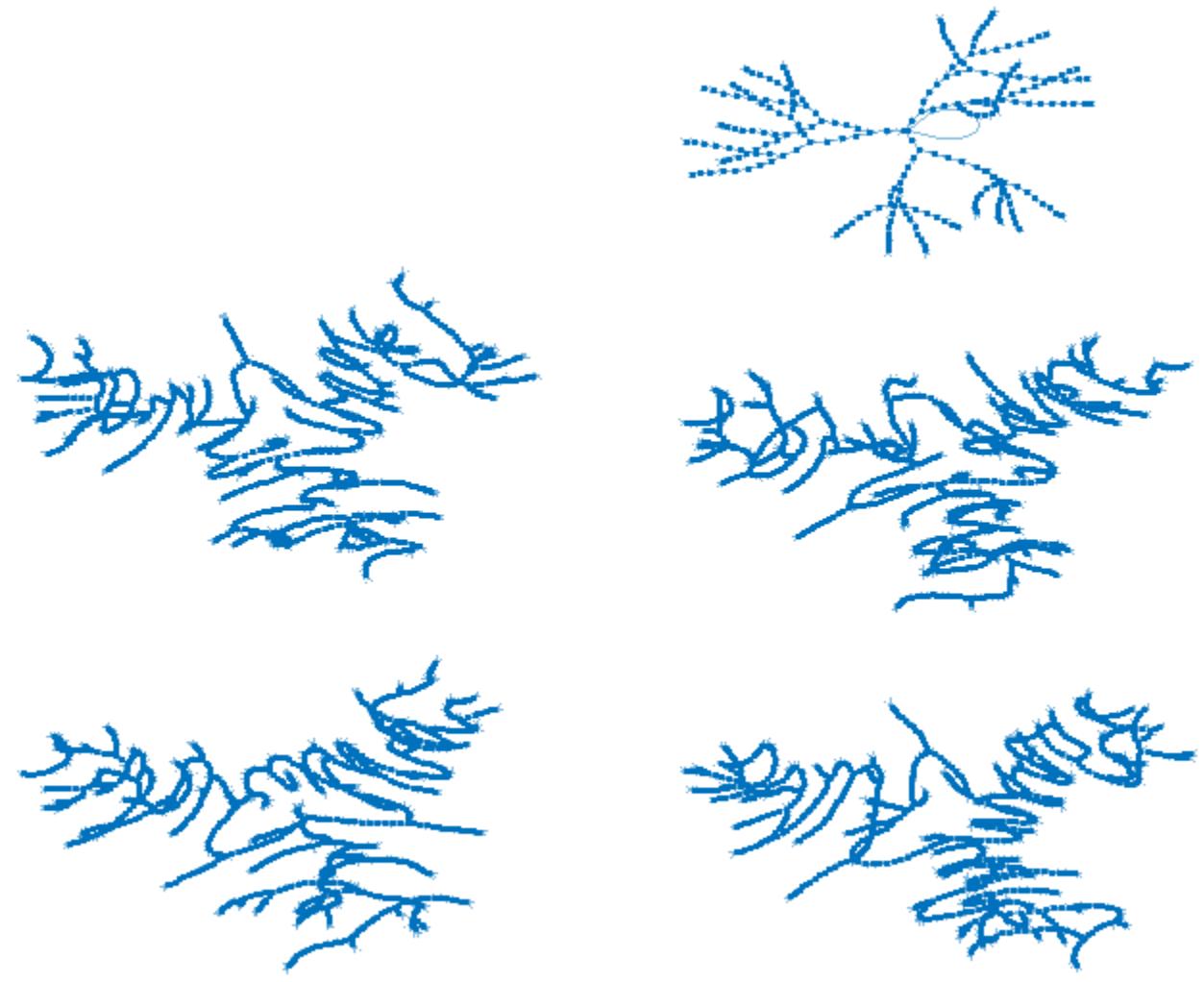


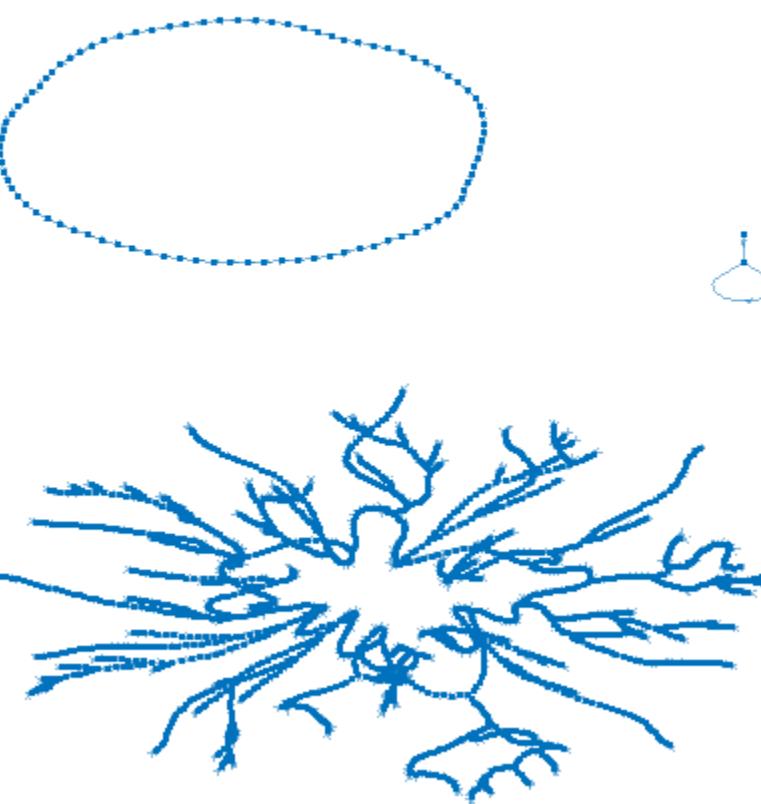
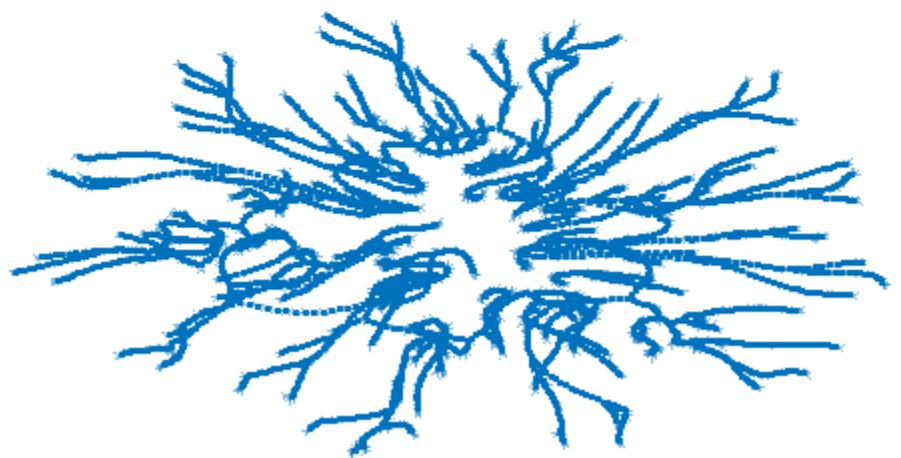
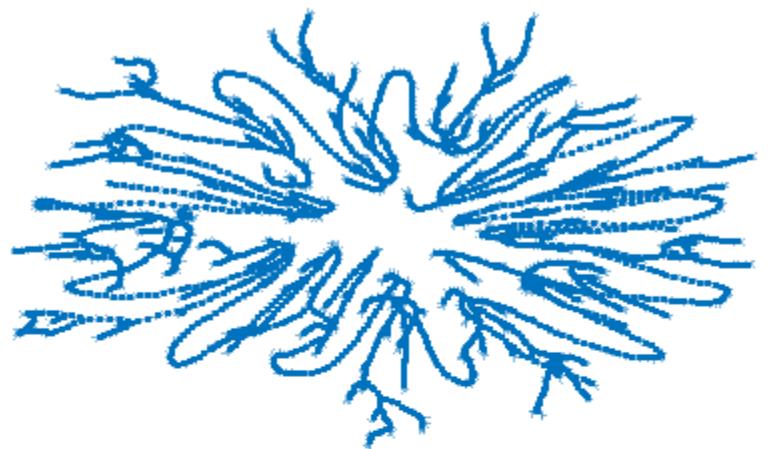


1



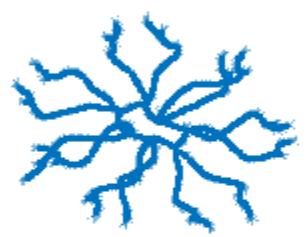




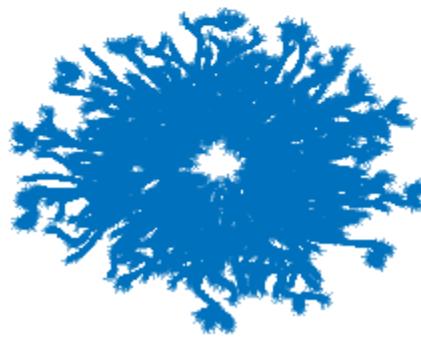




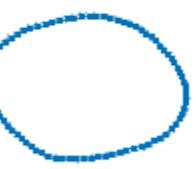
o

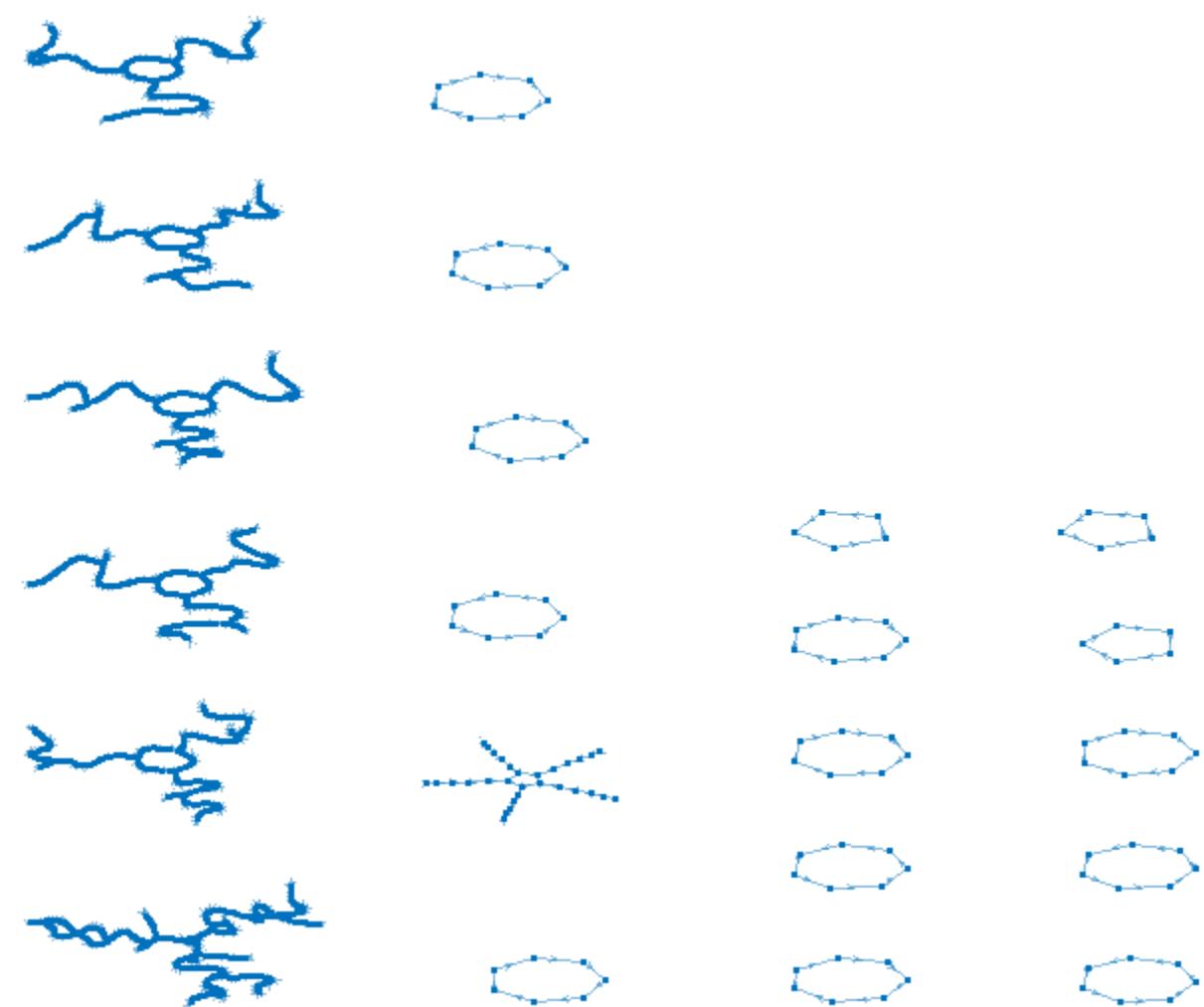
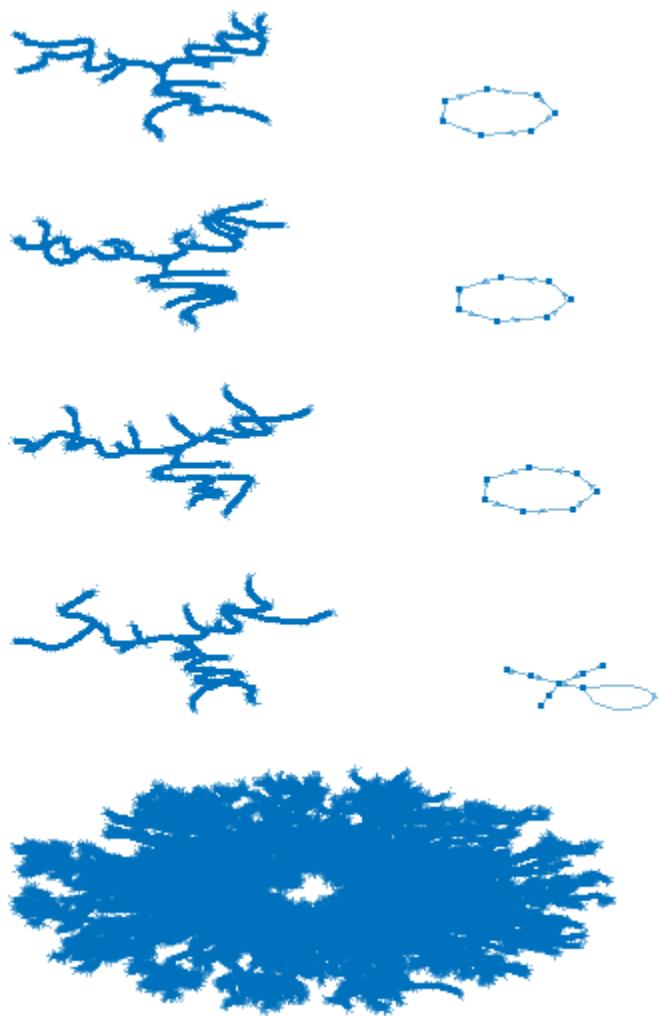


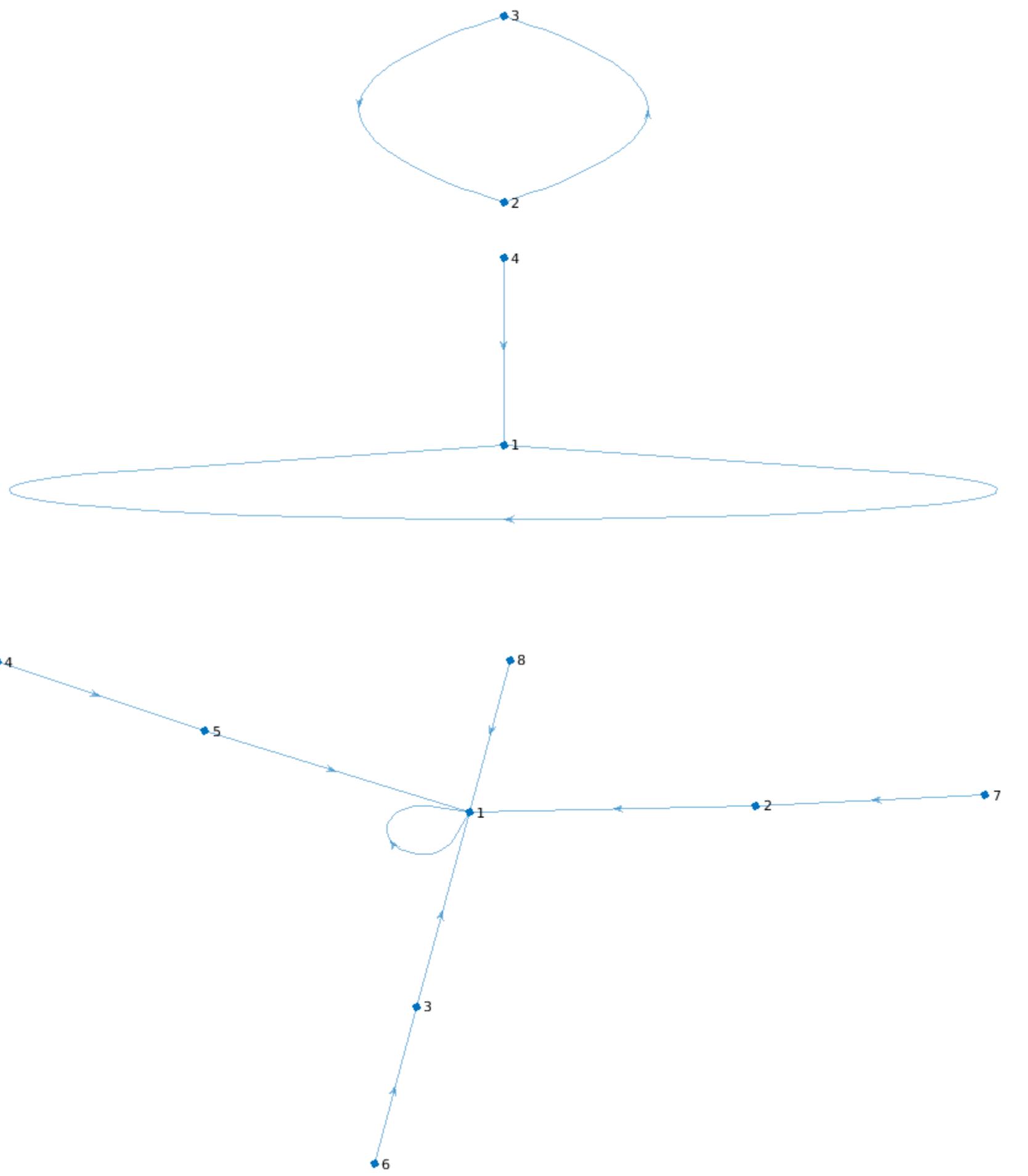
o

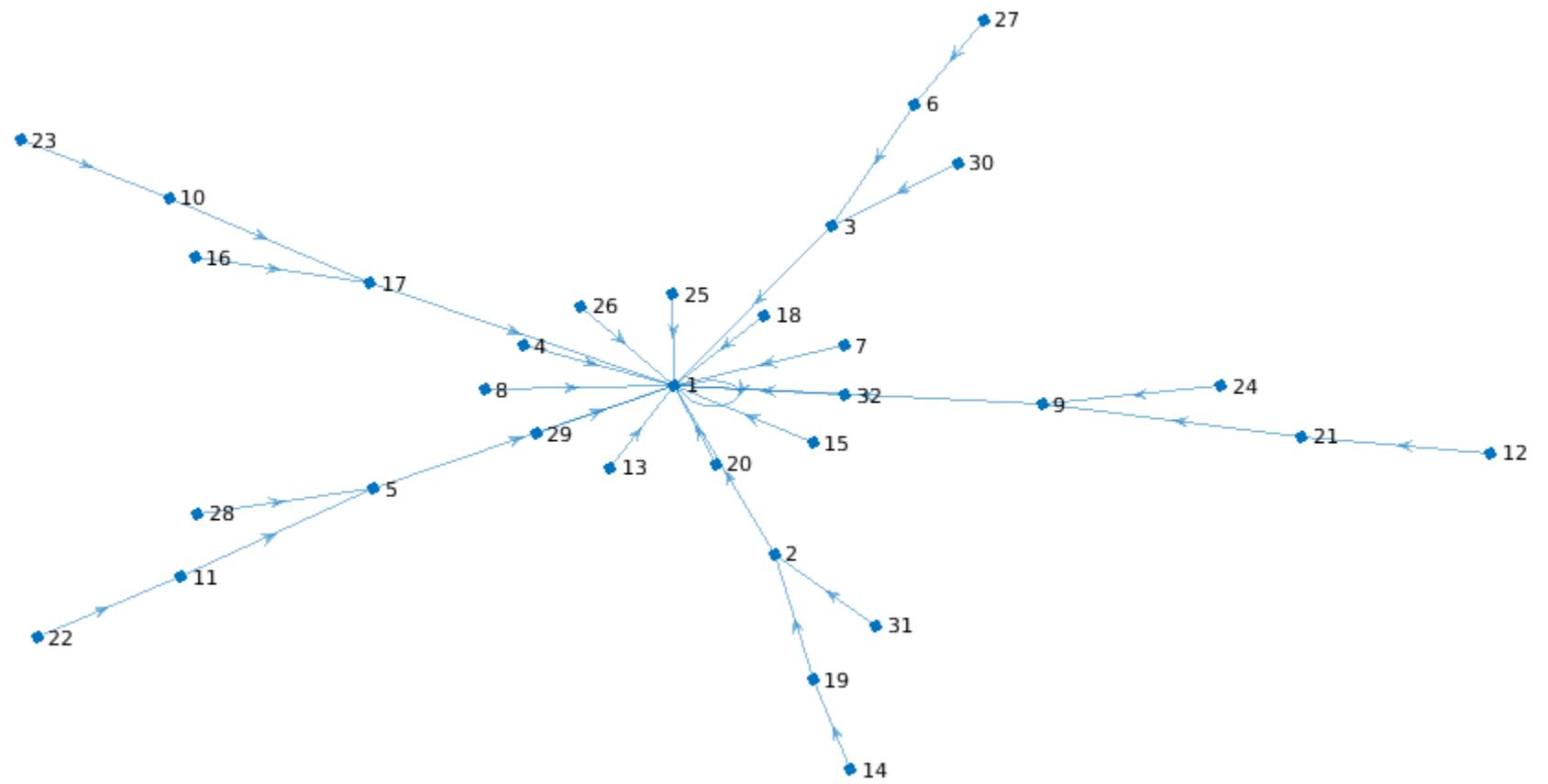
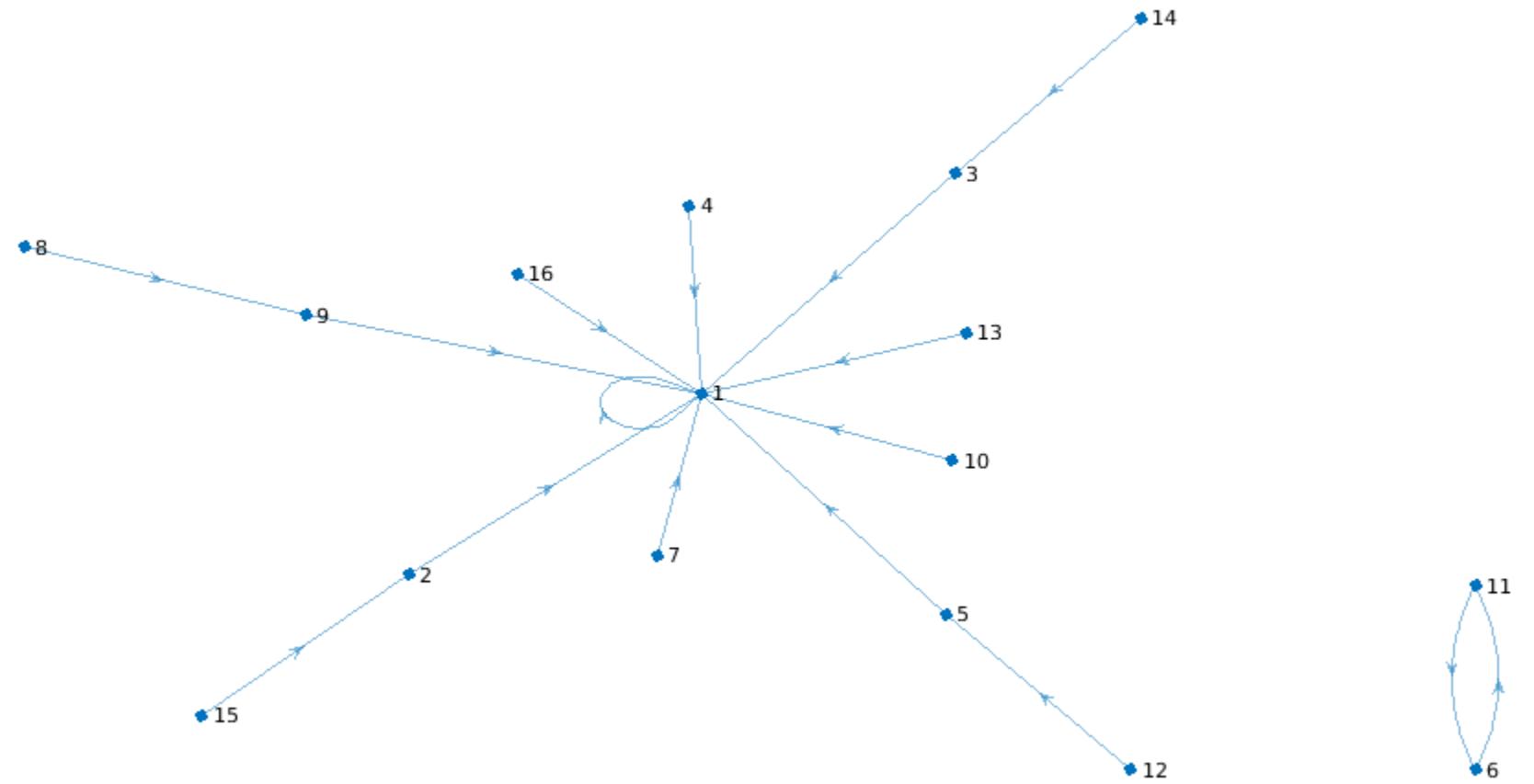


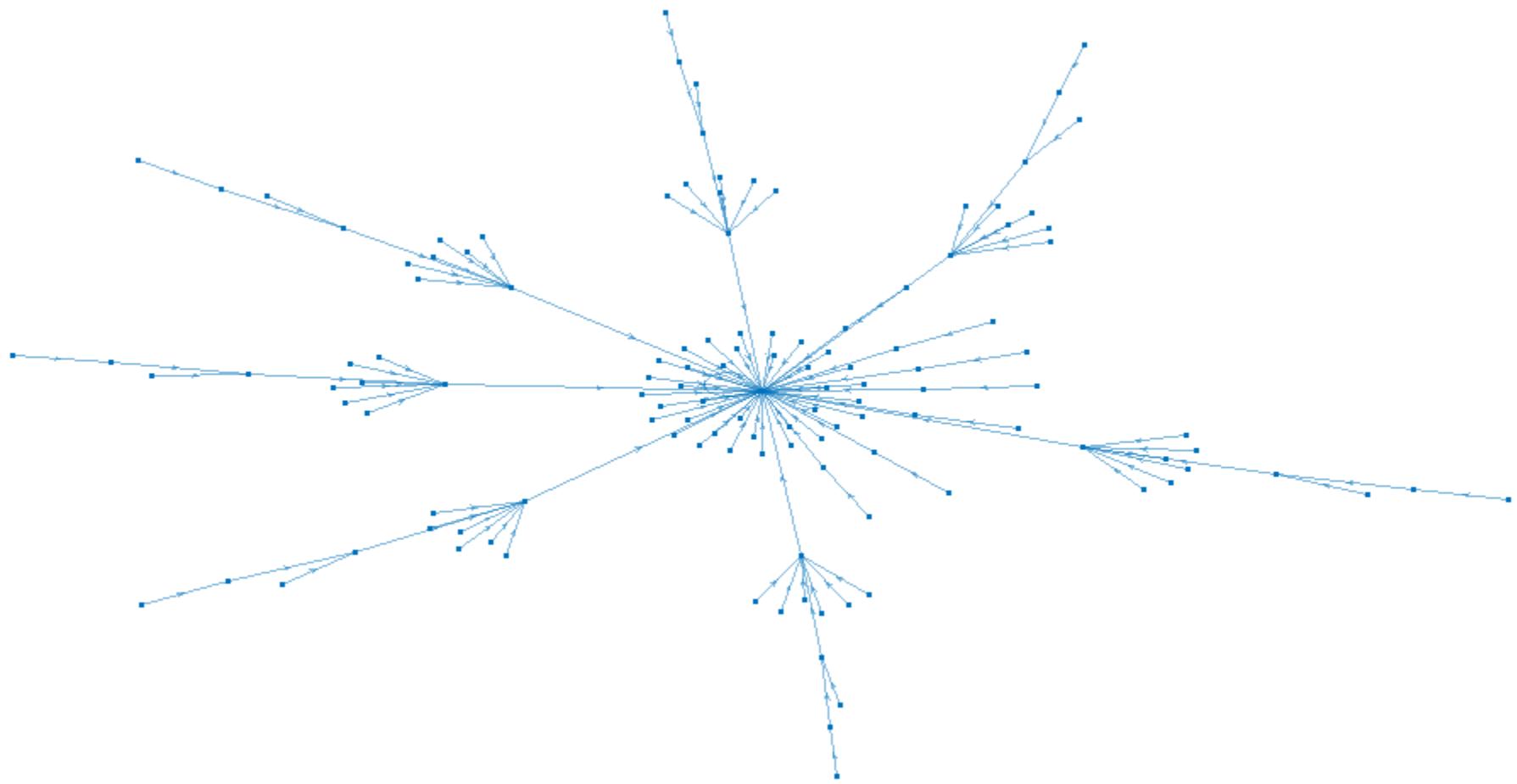
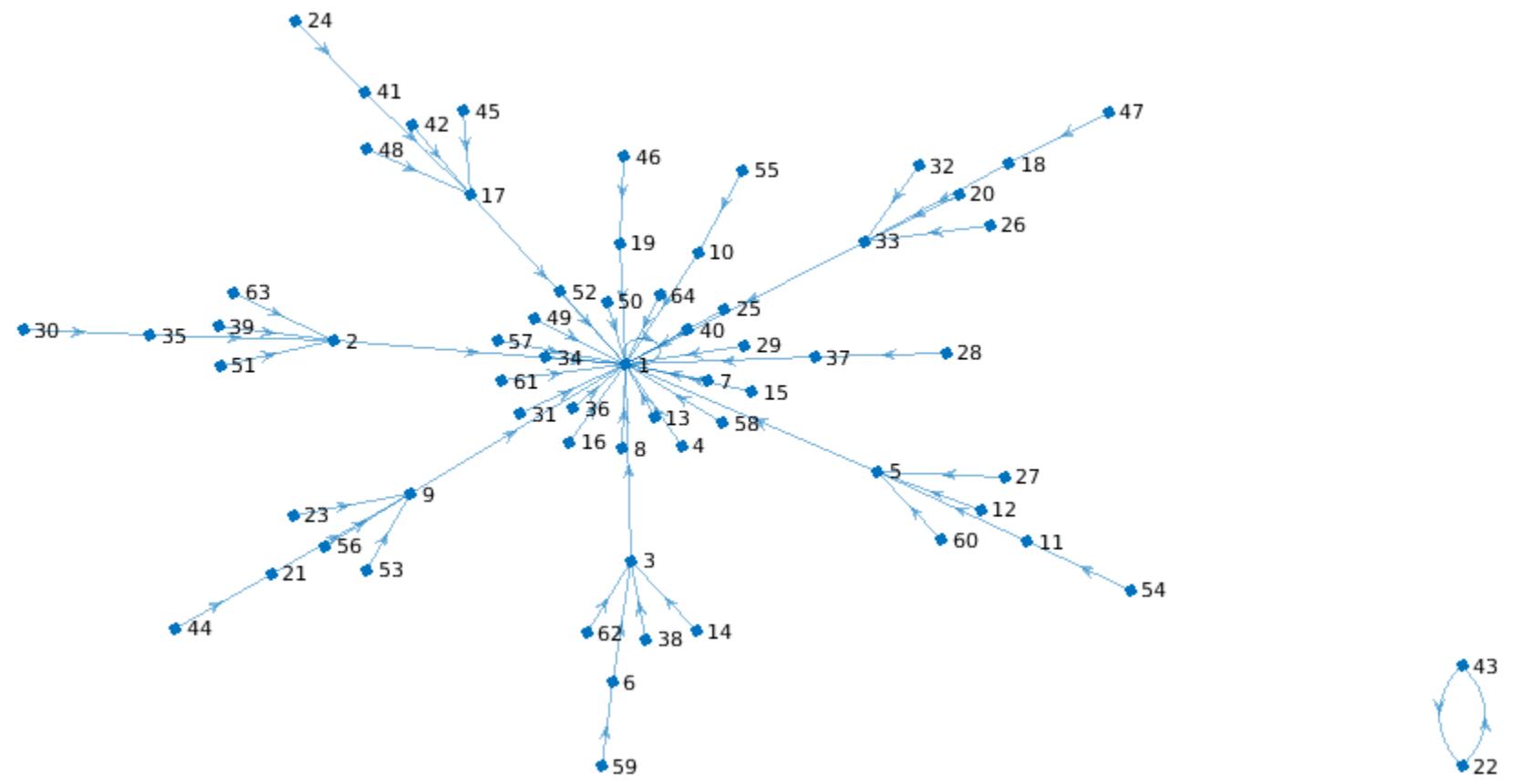
o

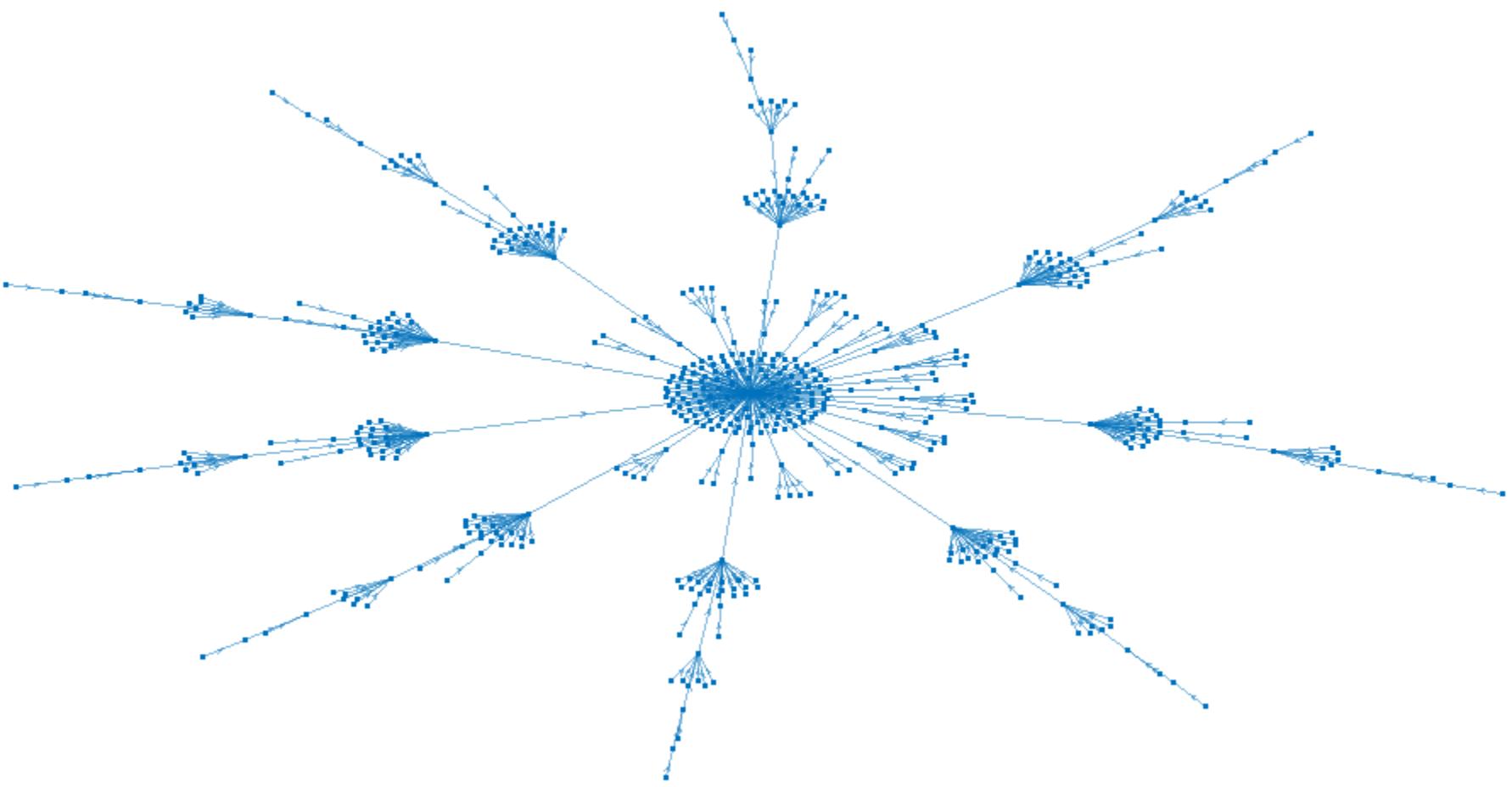
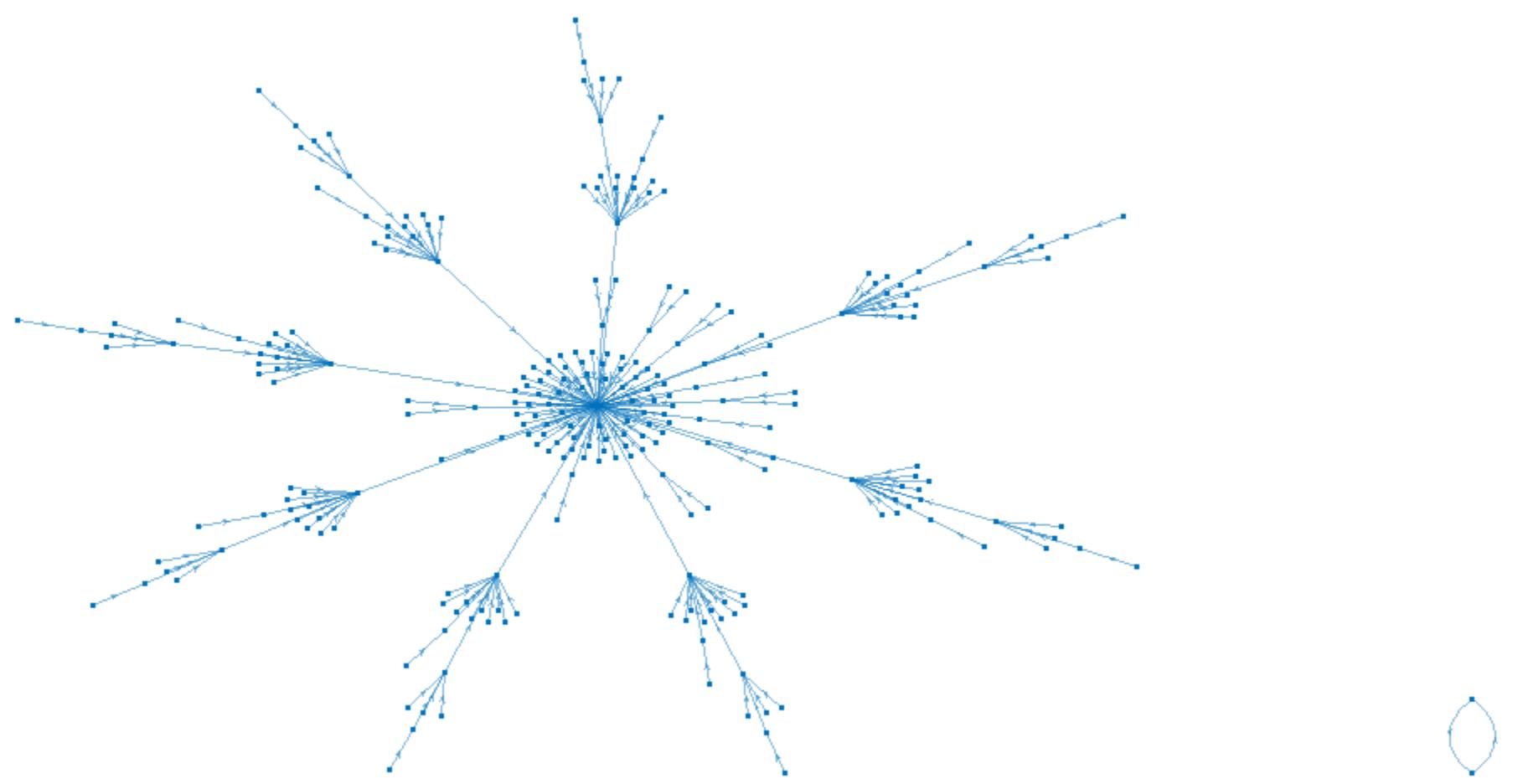


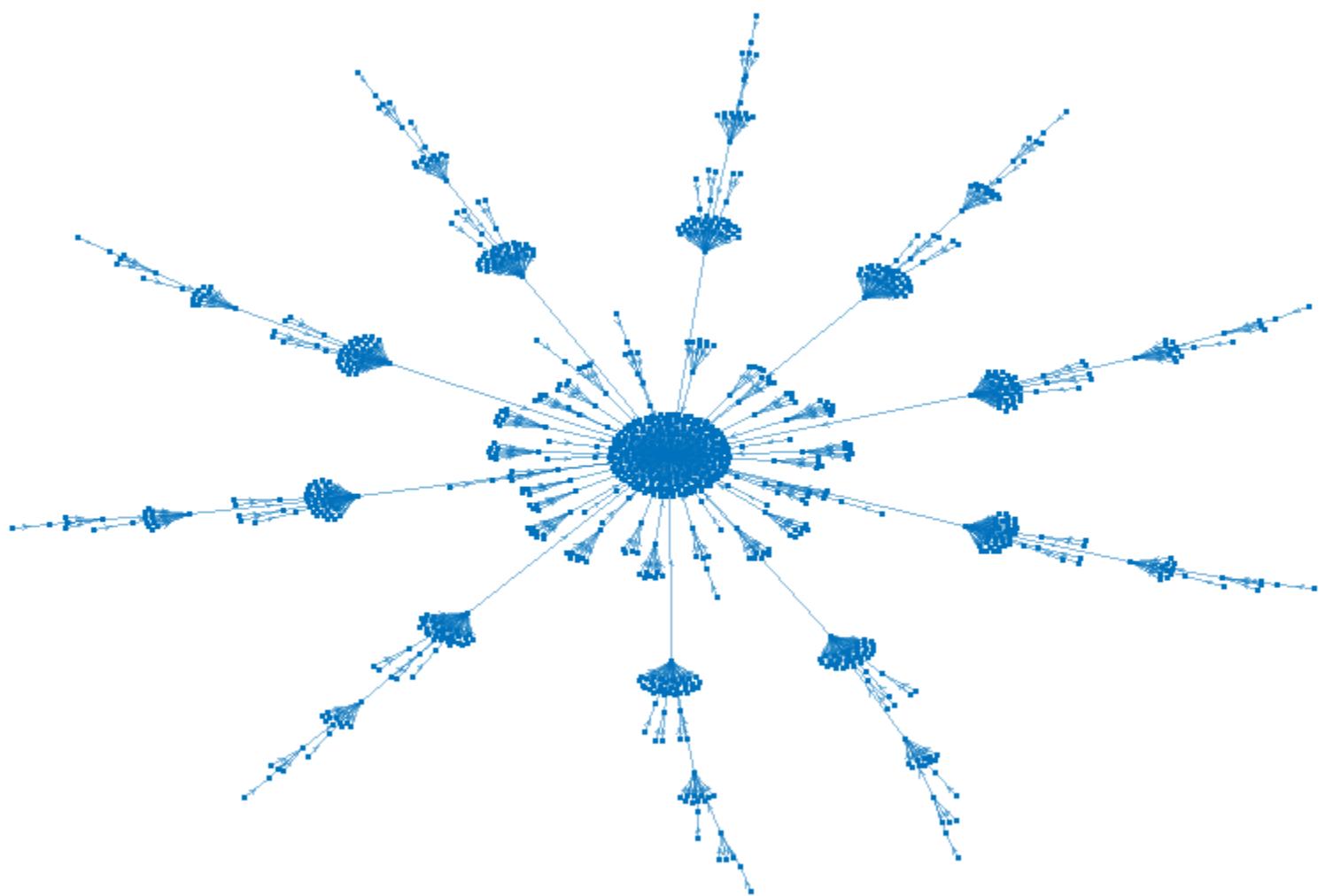


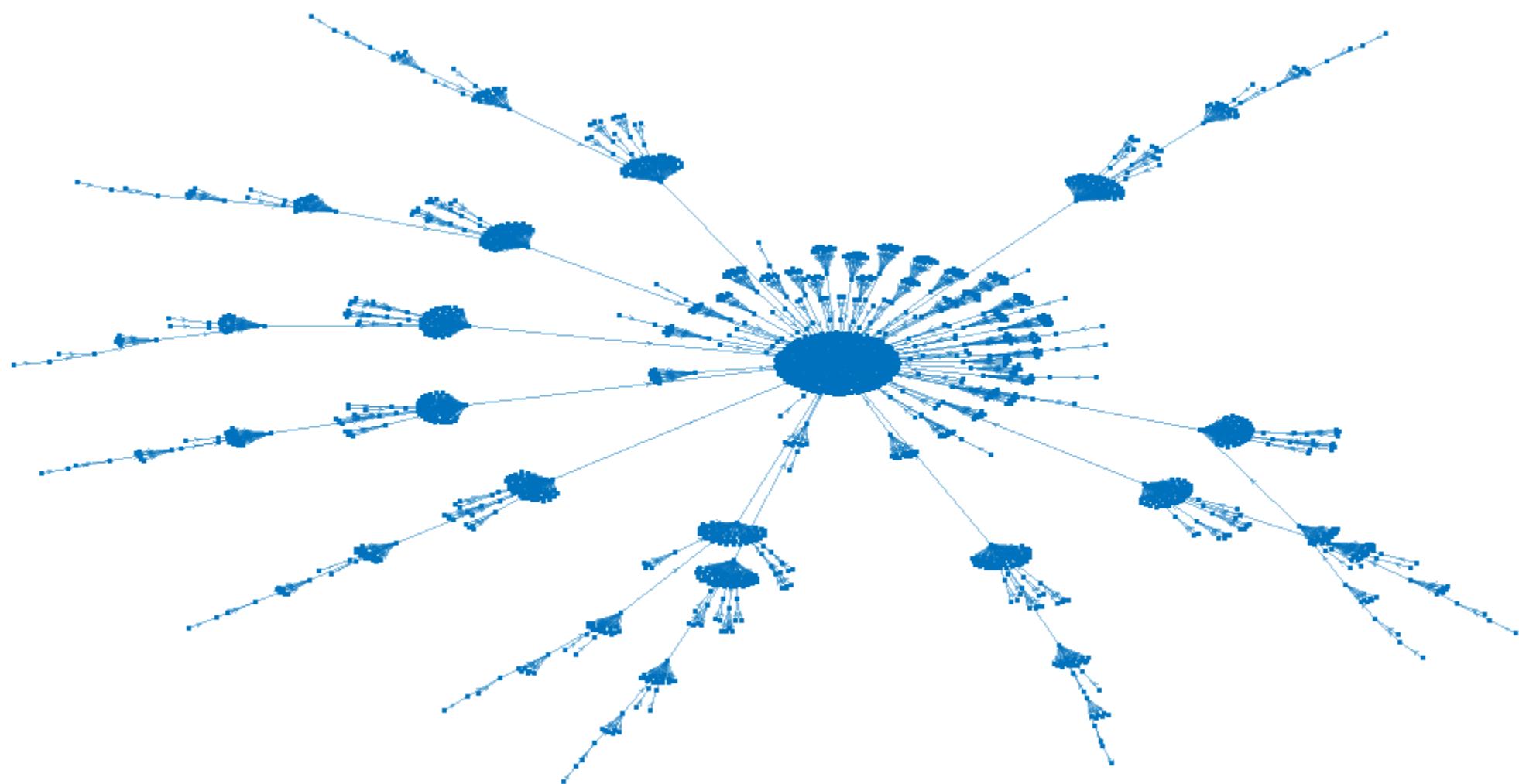


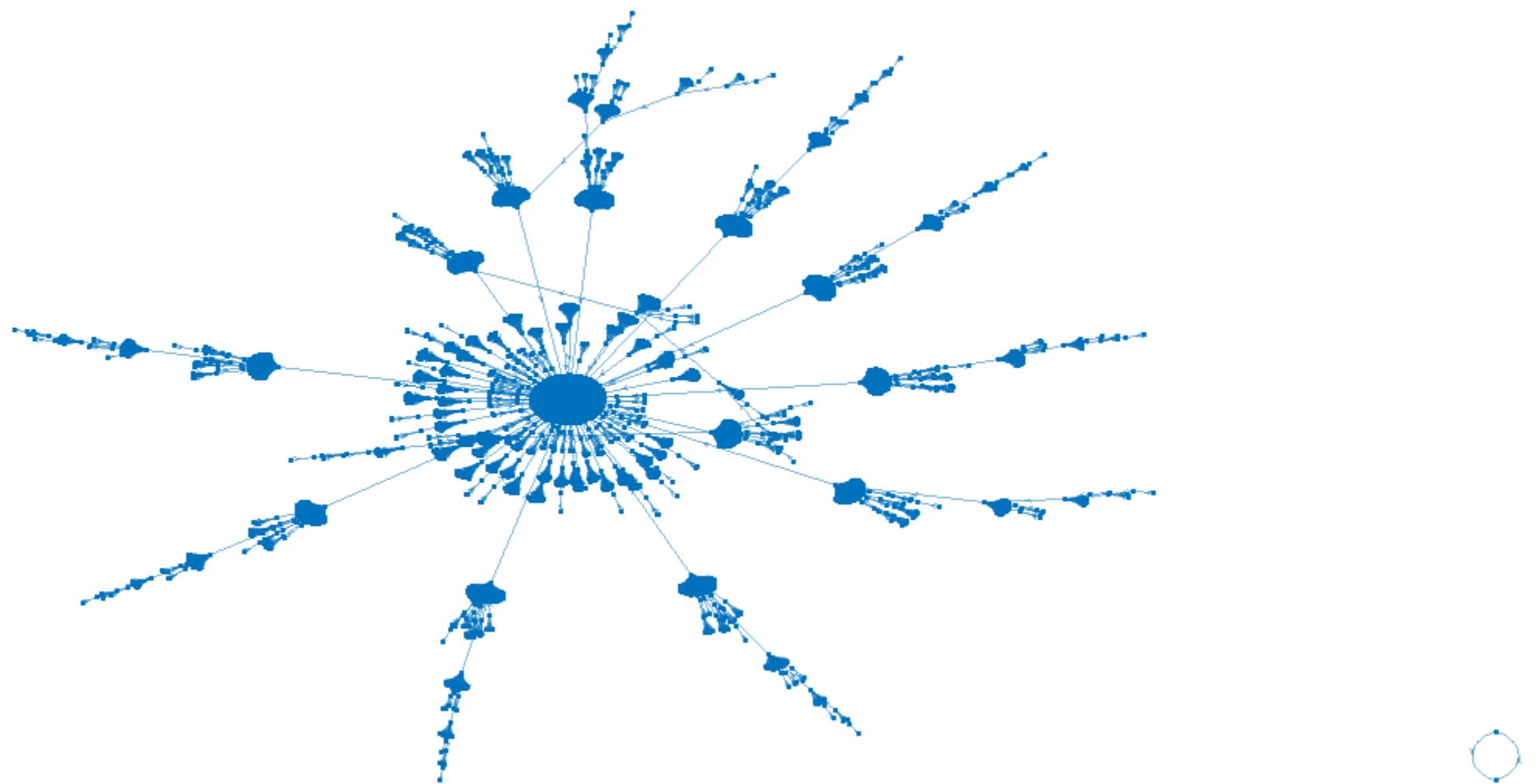


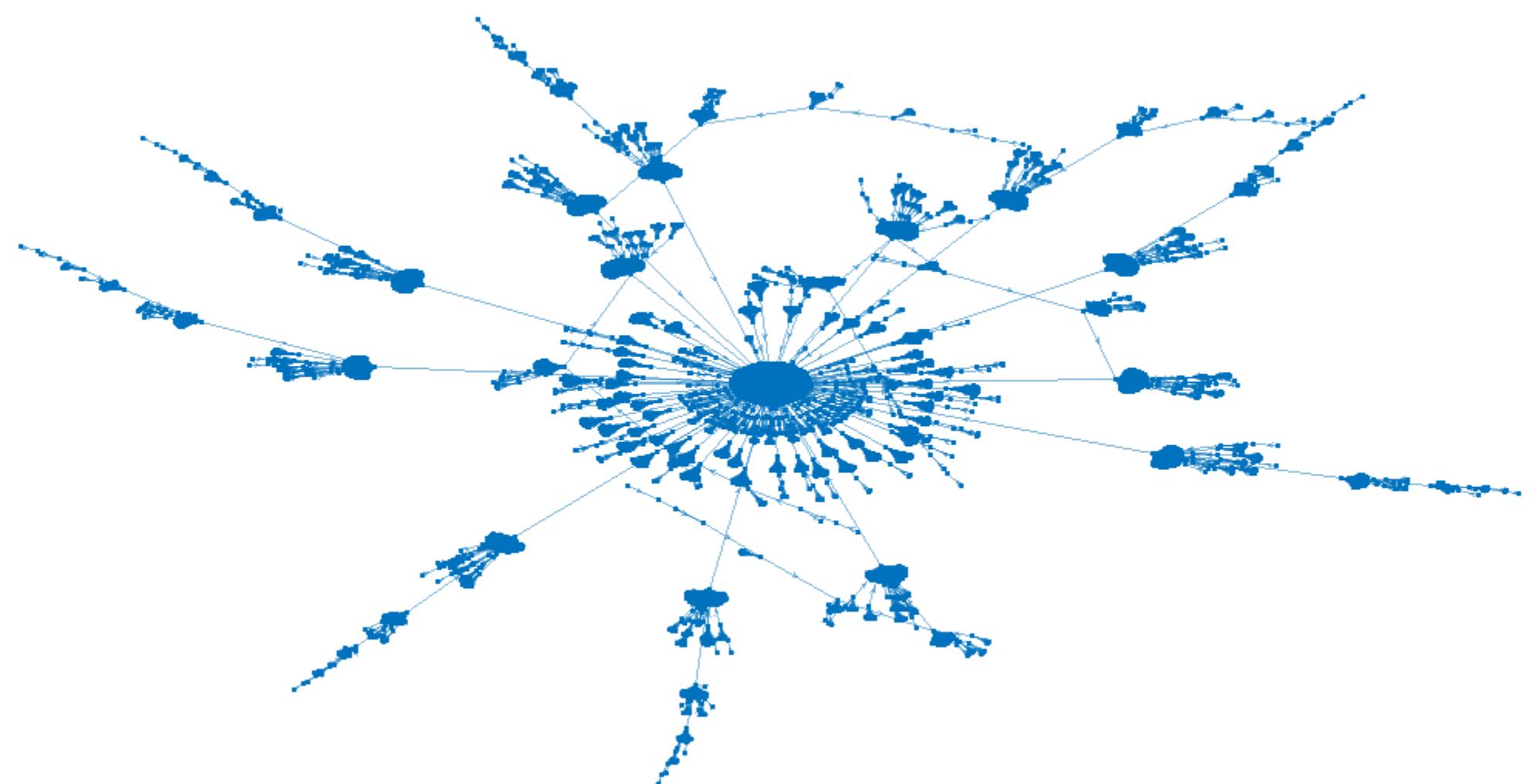


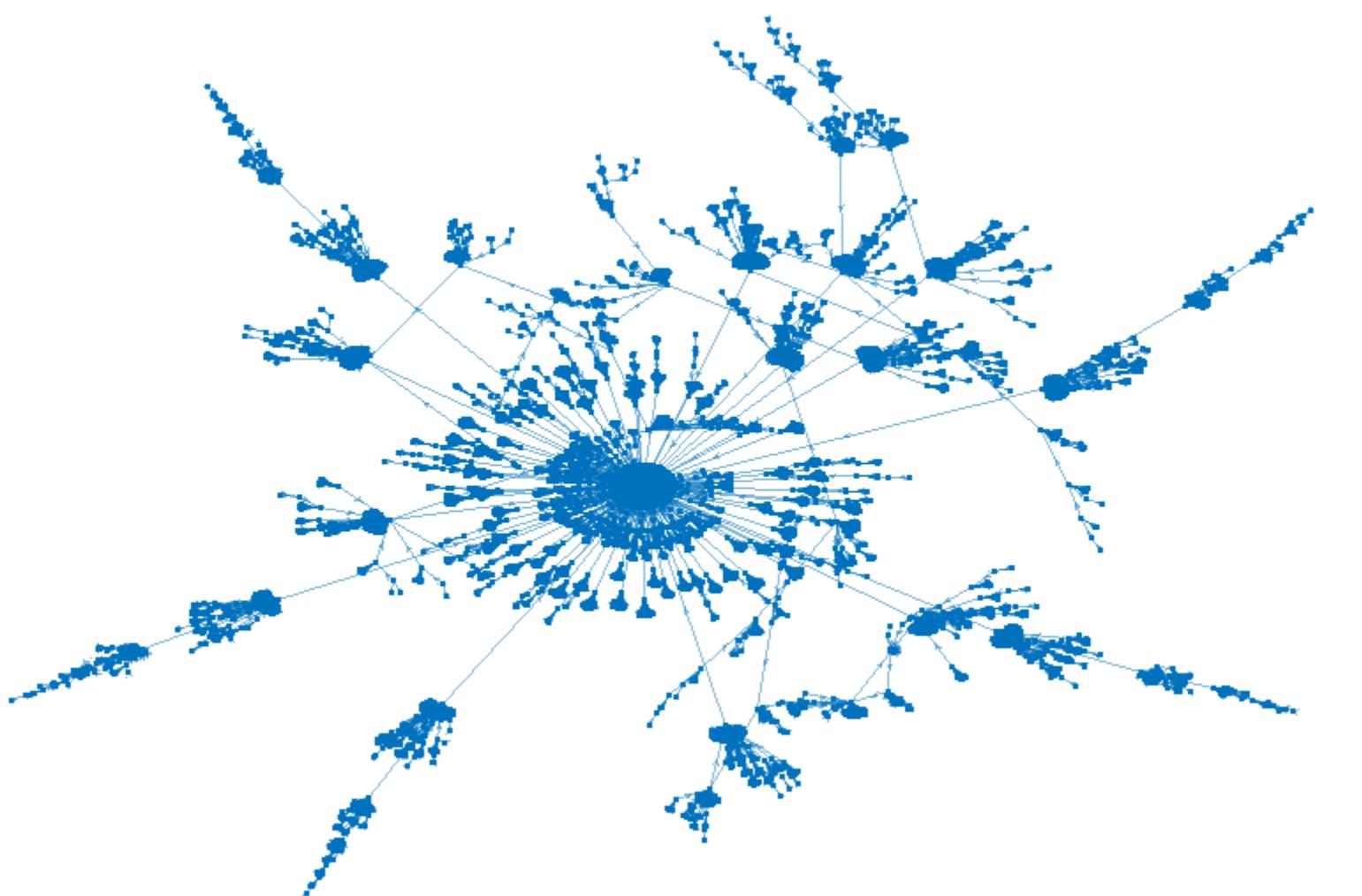




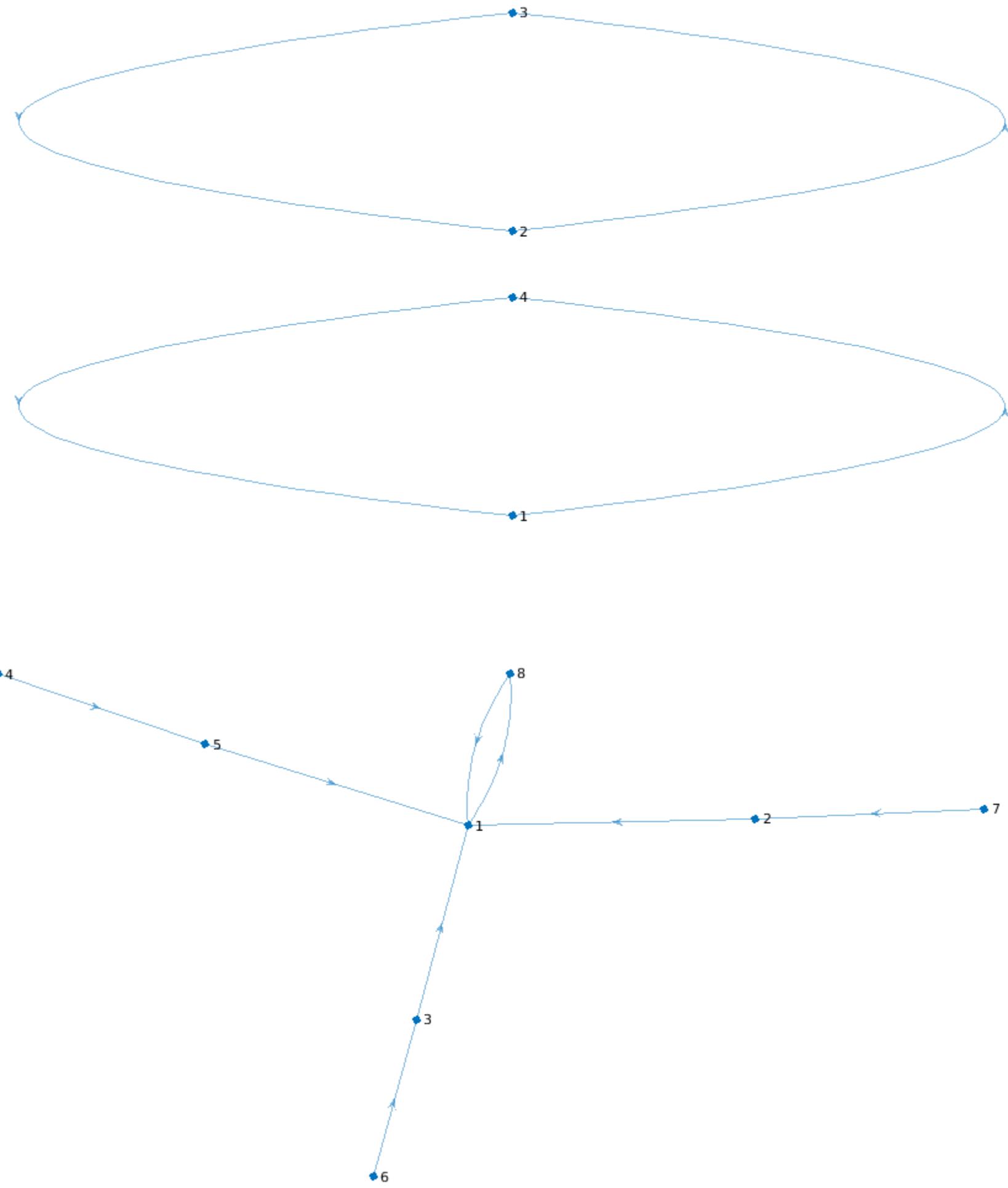


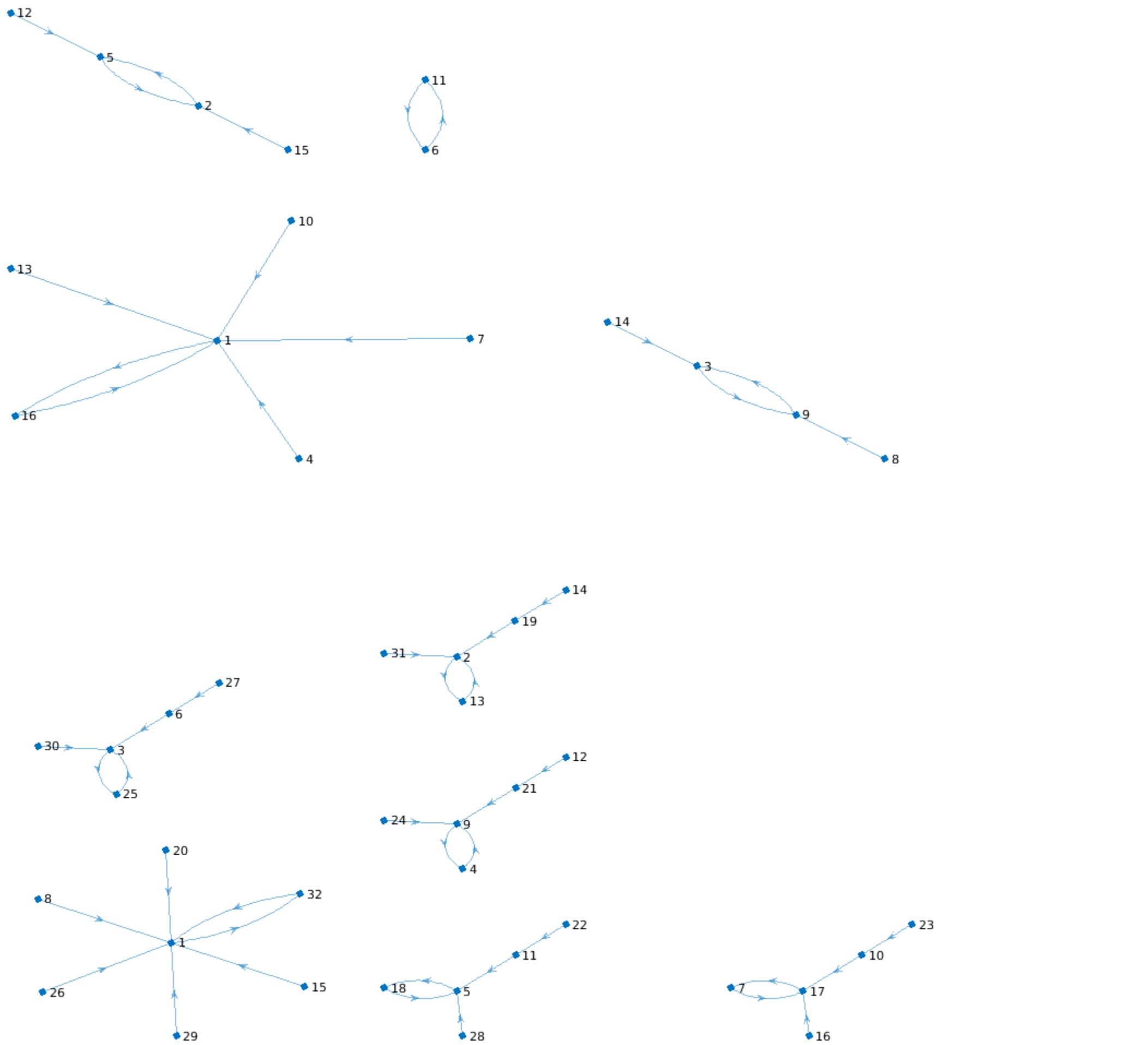


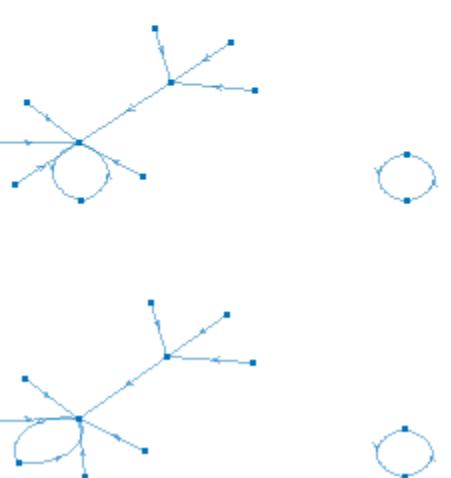
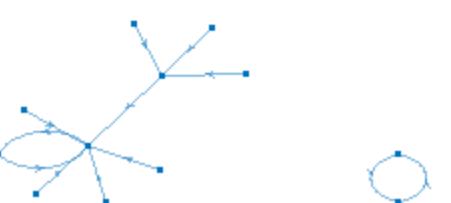
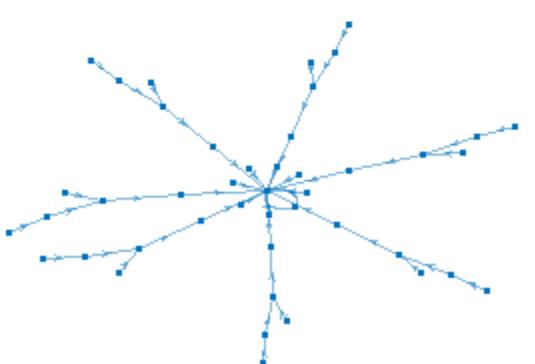
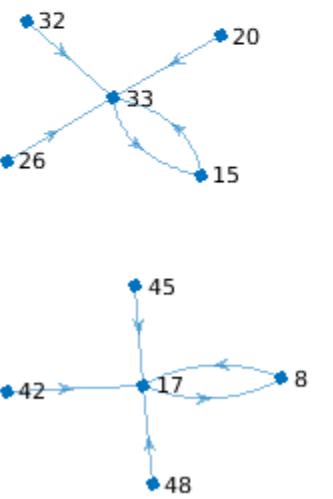
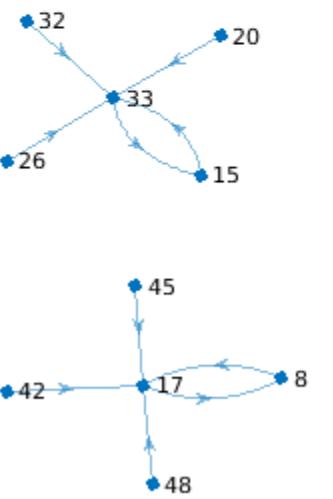
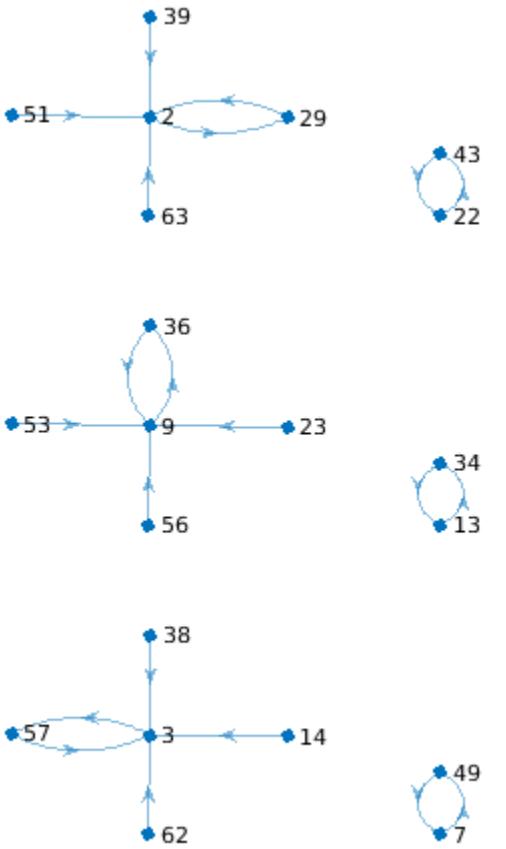
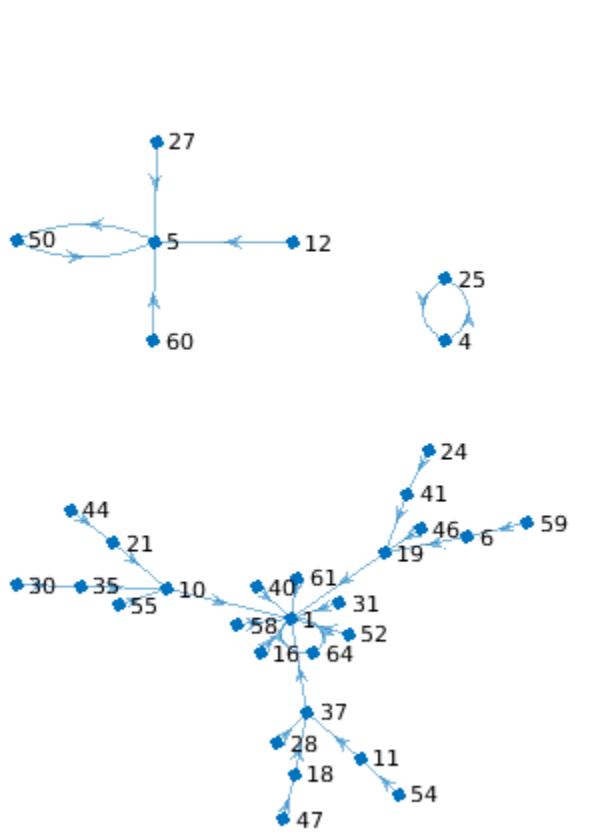


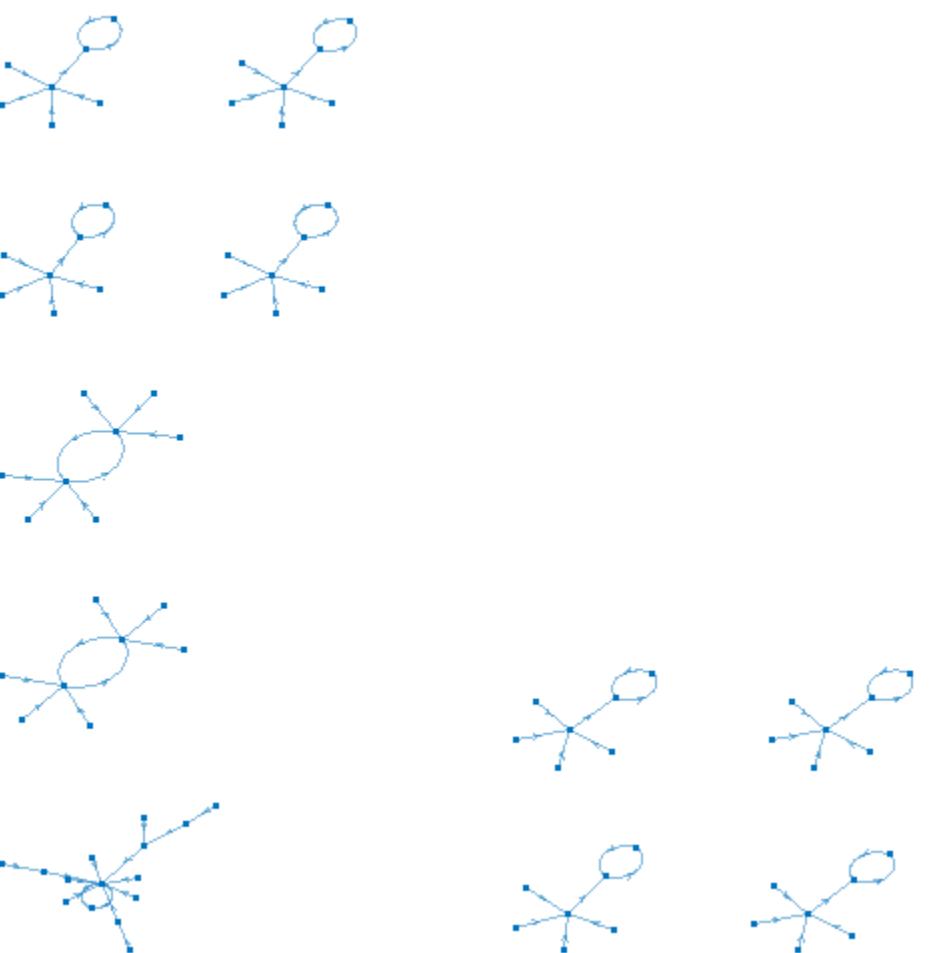
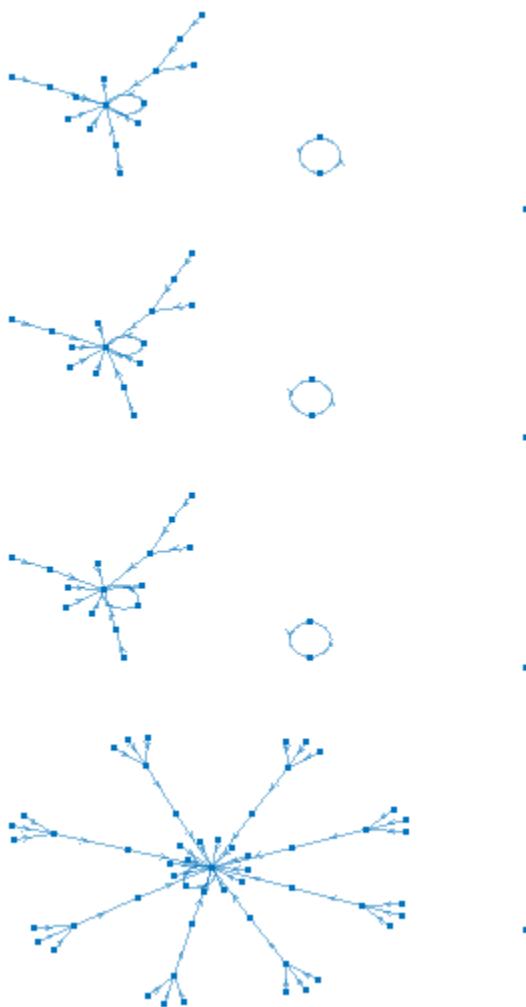


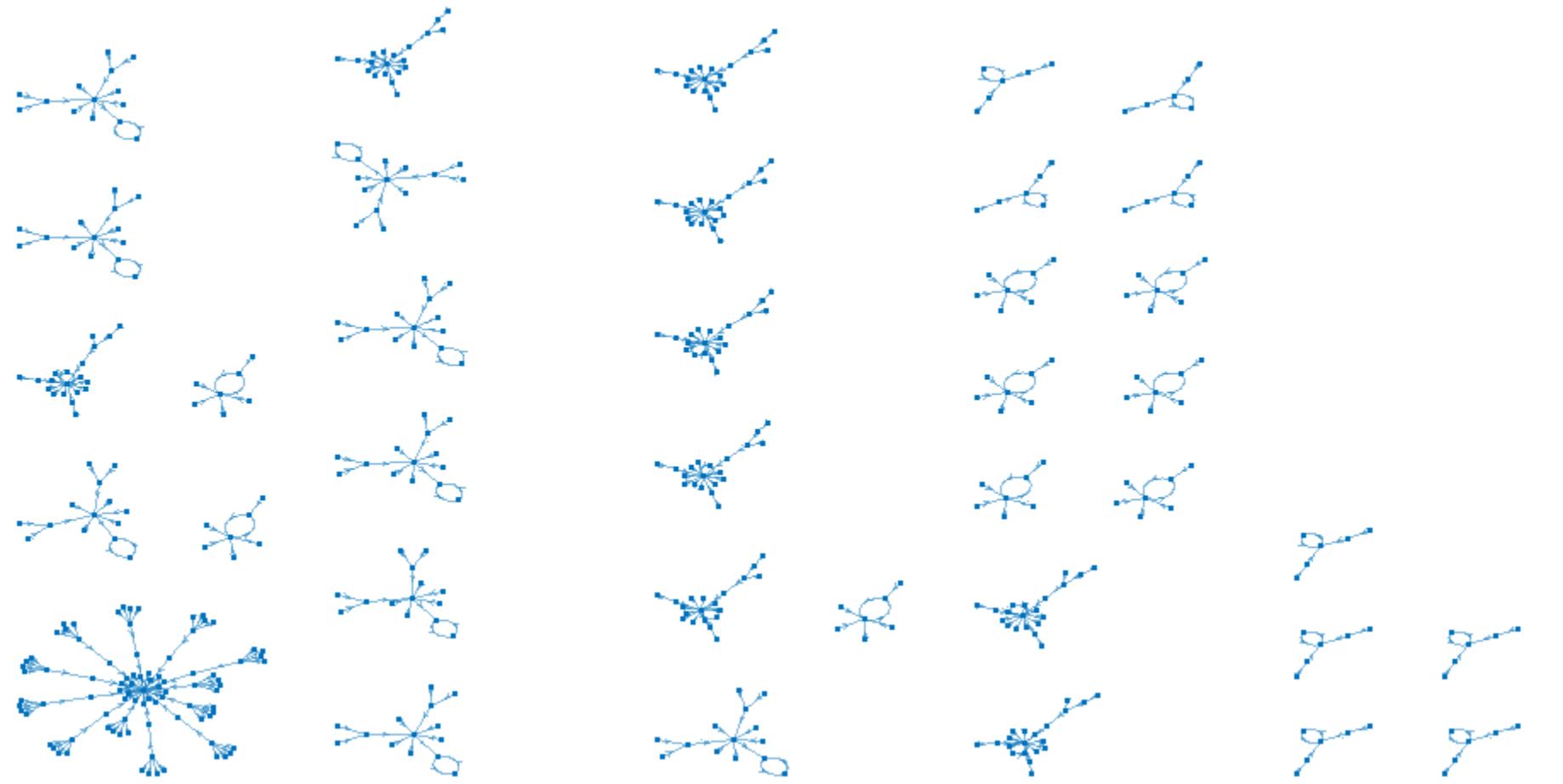


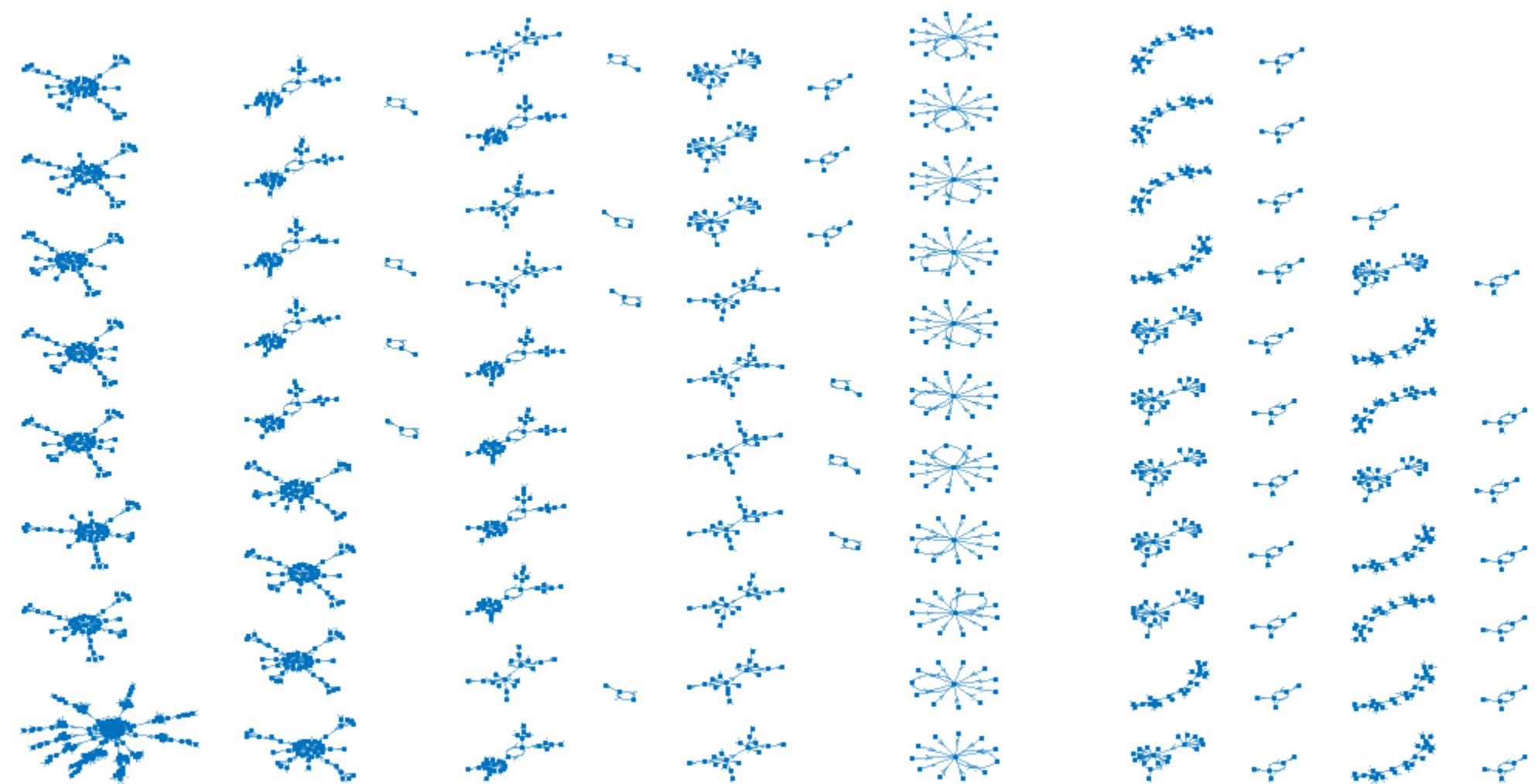


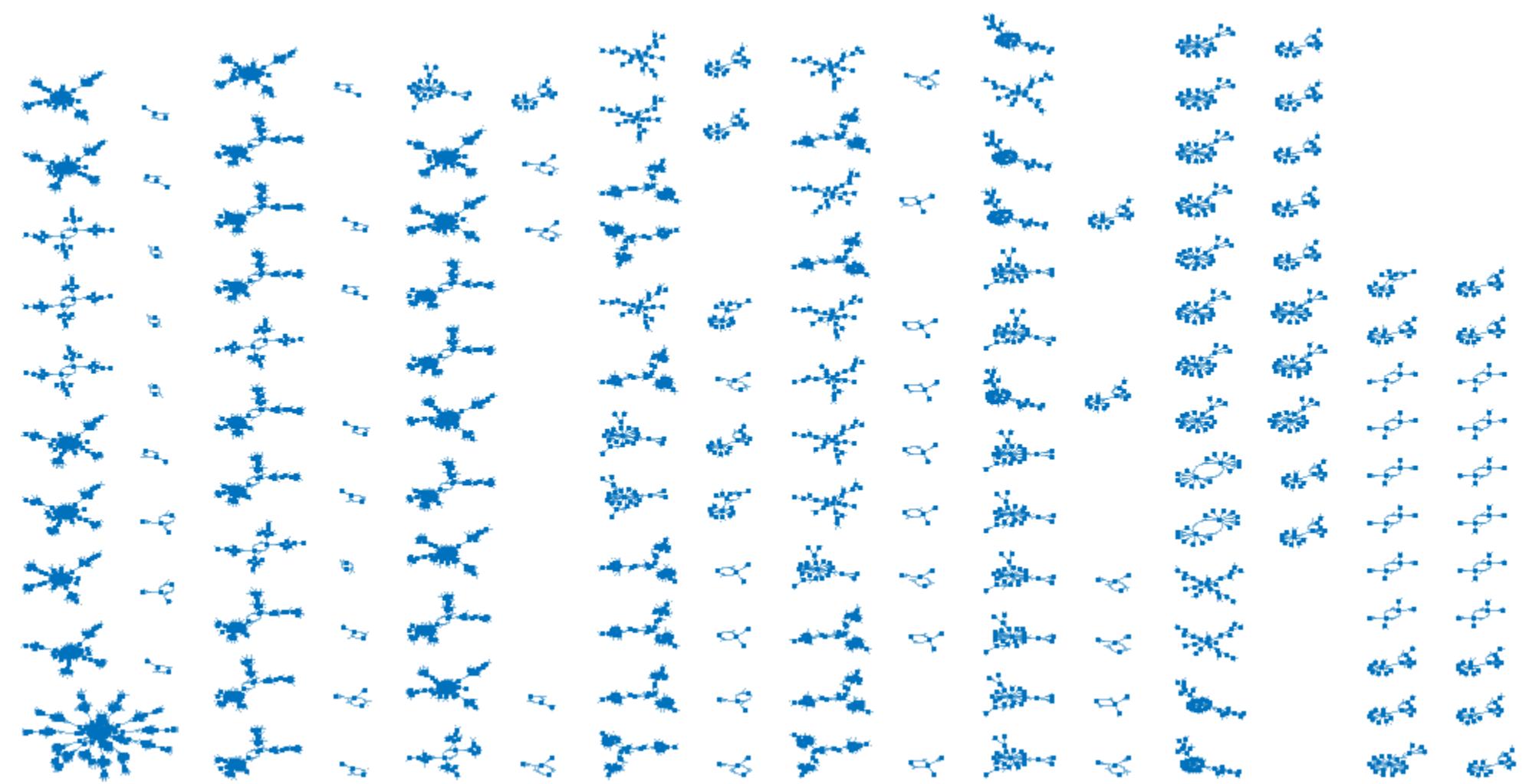












卷之三

卷之三

卷之三

卷之三

卷之三

卷之三

卷之三

卷之三

卷之三

其後人復有以爲子者，則是子之子也。故曰：「子之子」。

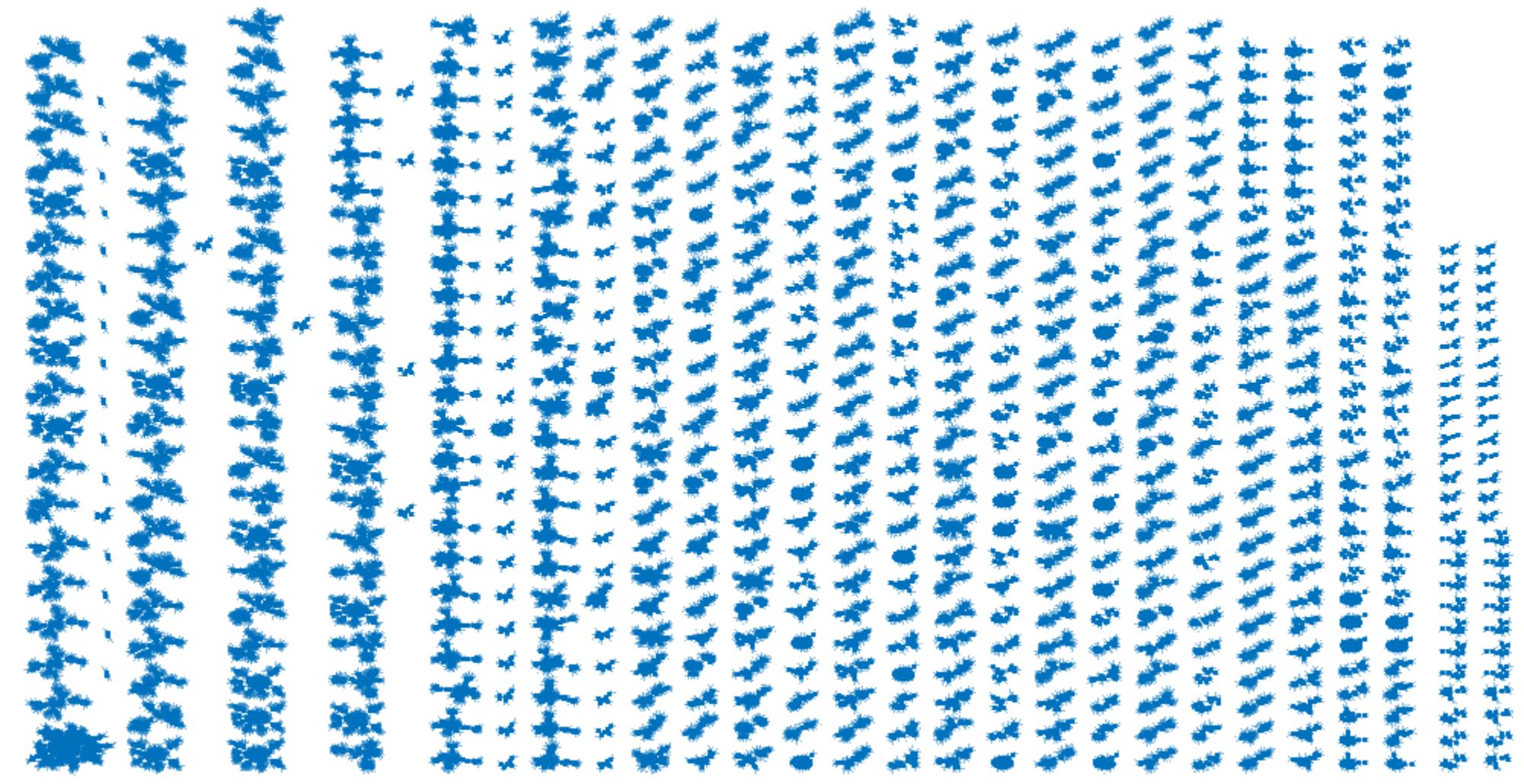
The image consists of a 10x10 grid of small, dark blue, irregular shapes. Each shape is roughly circular or oval with a textured, spiky or granular surface. They are arranged in a staggered, non-uniform pattern across the frame. The overall effect is like a microscopic view of a material's surface or a collection of small, randomly oriented particles.

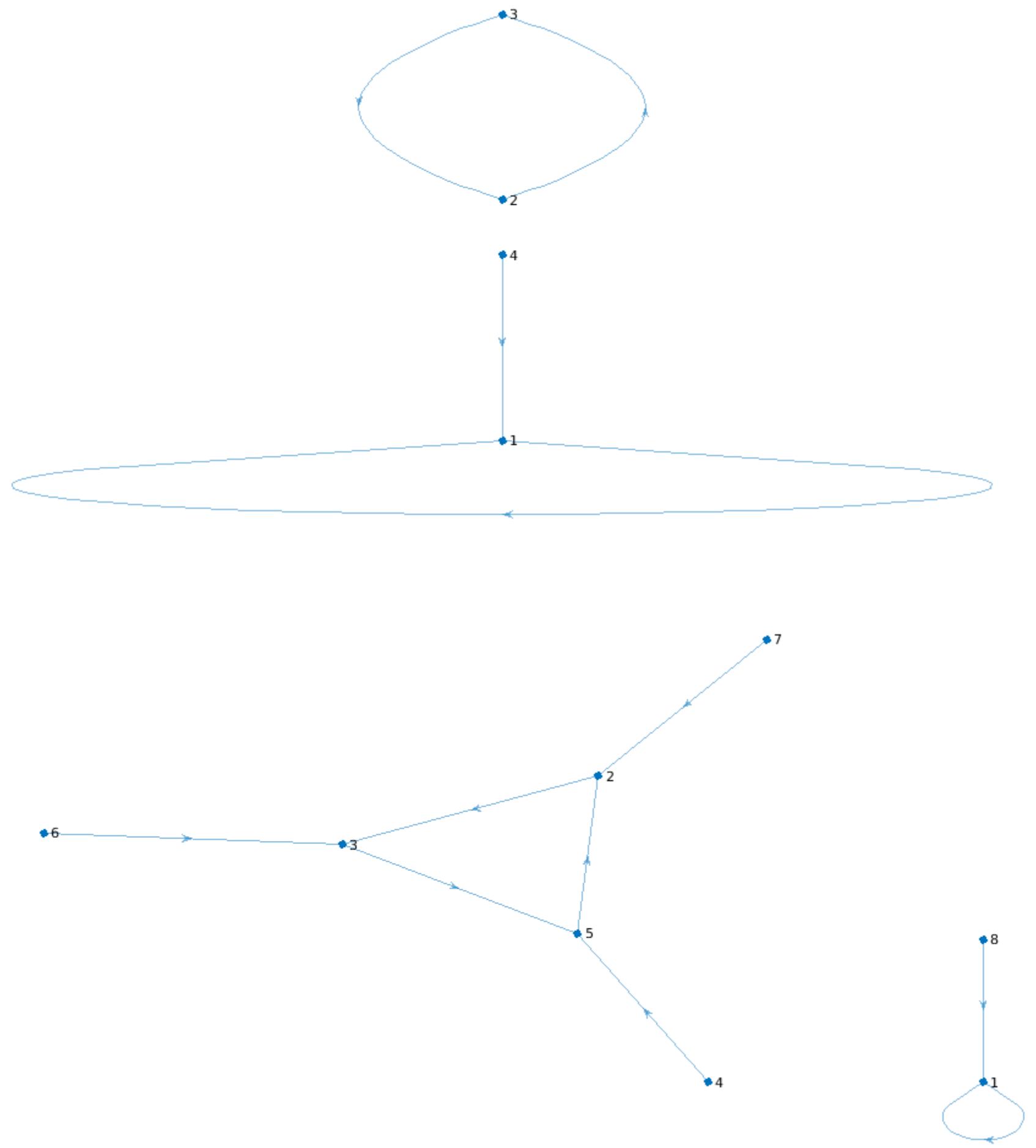
卷之三

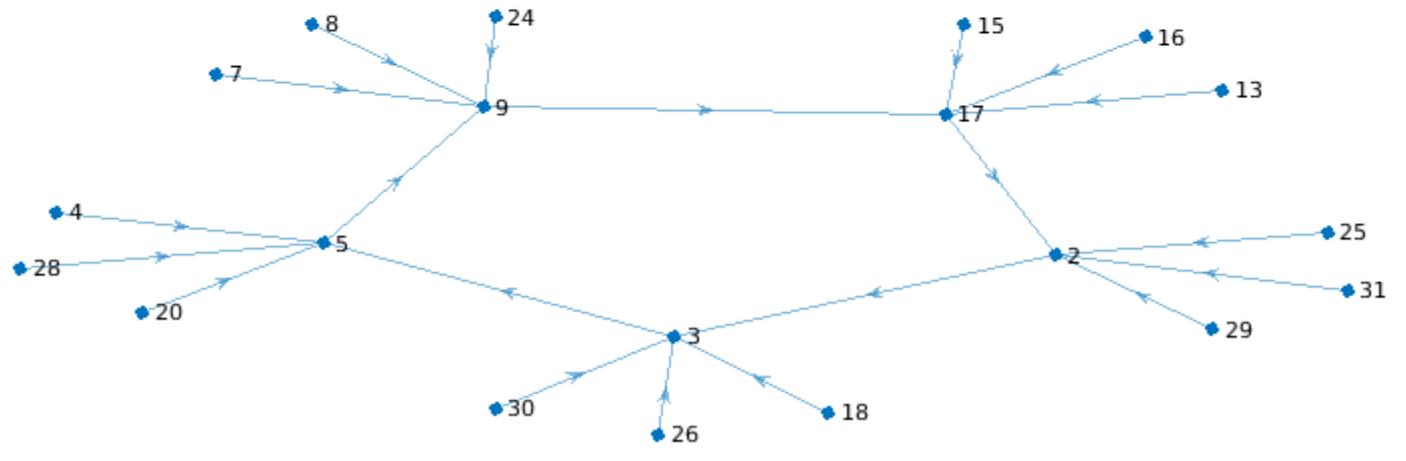
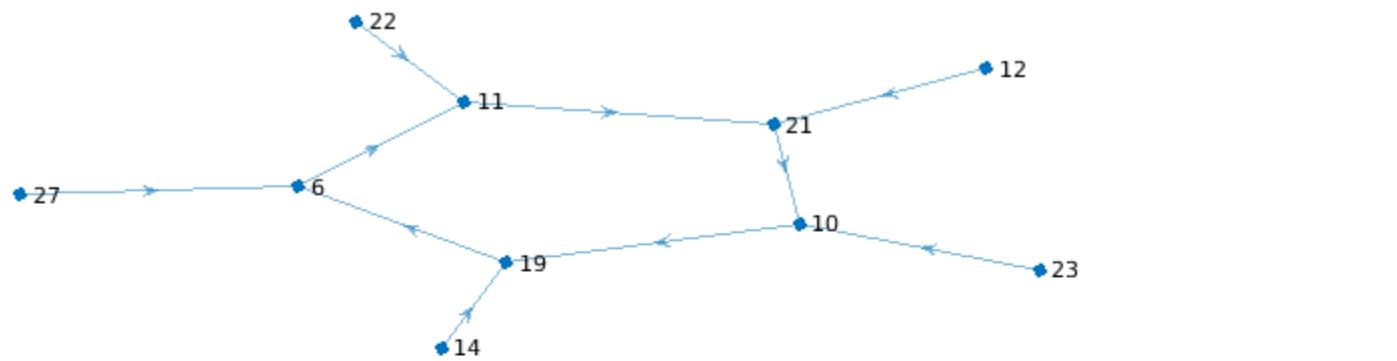
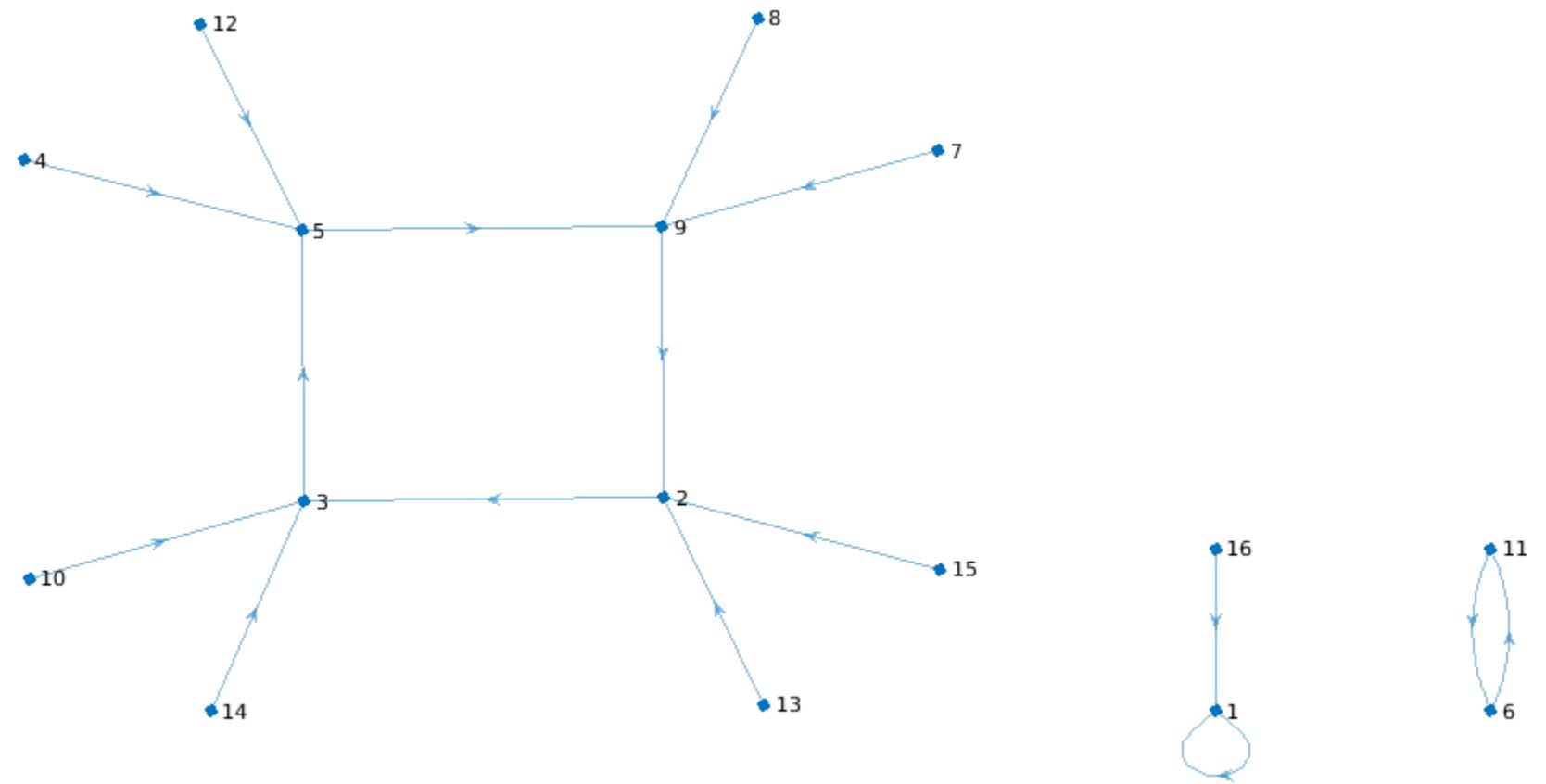
卷之三

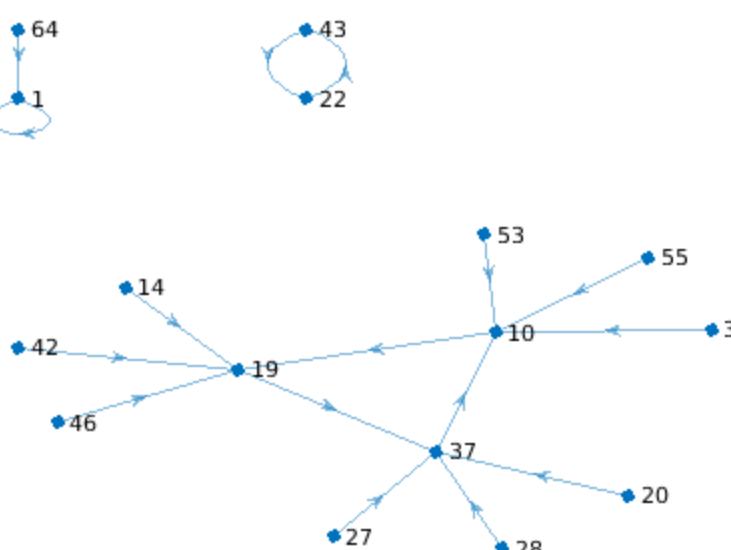
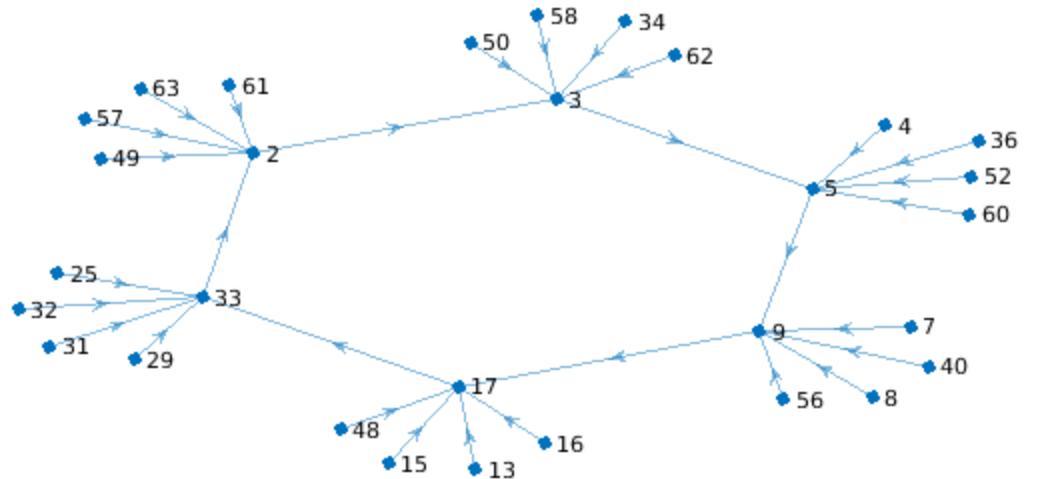
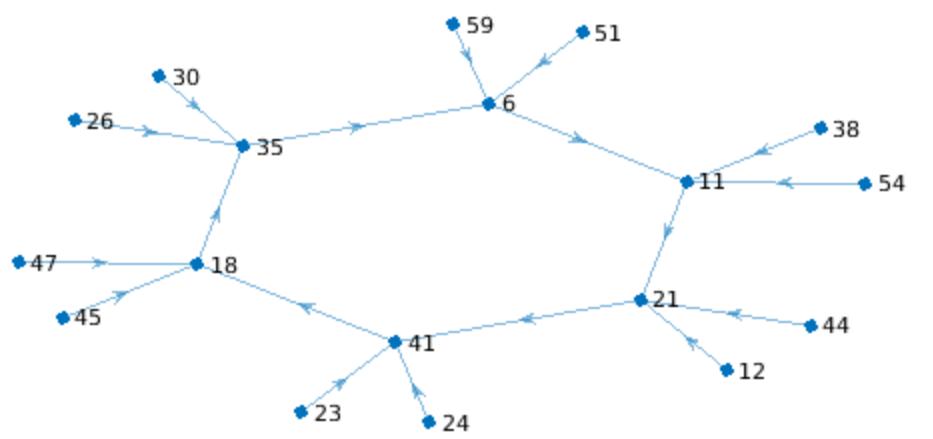
卷之三

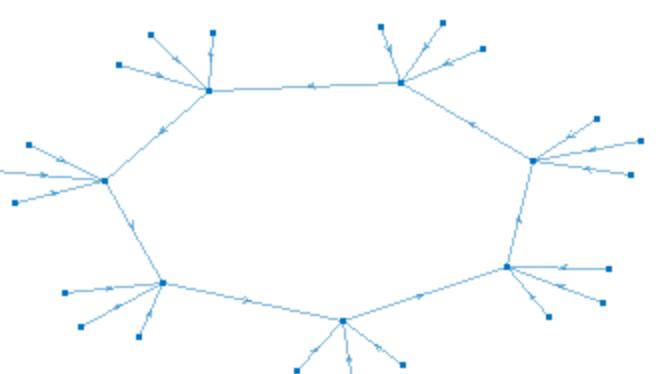
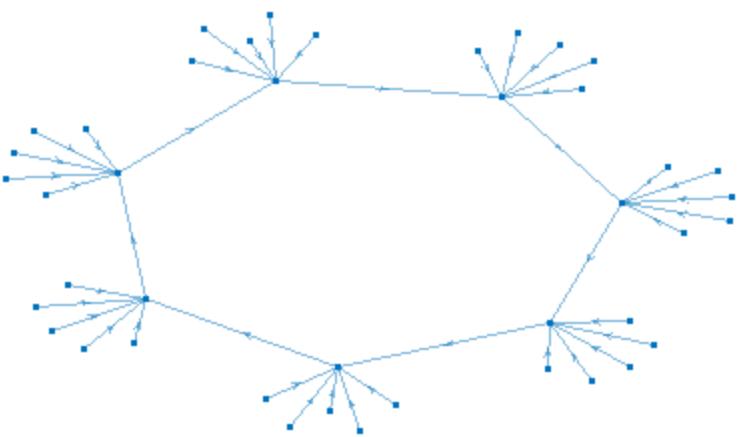
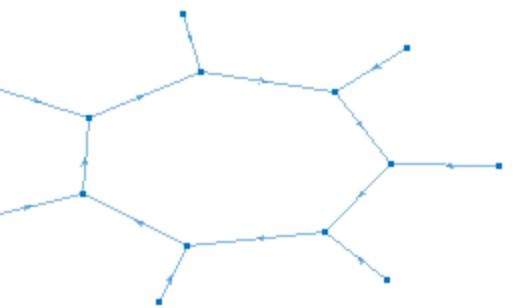
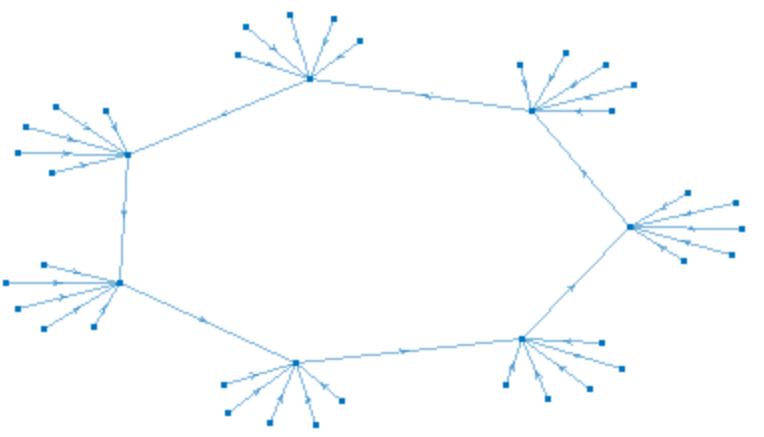
卷之三

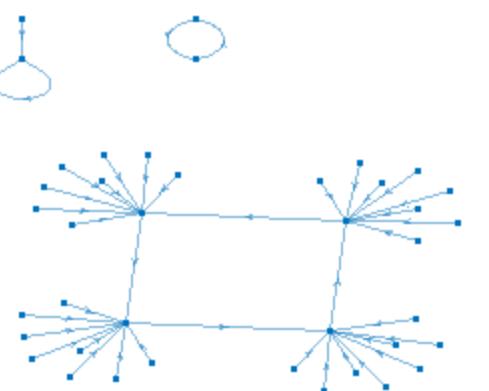
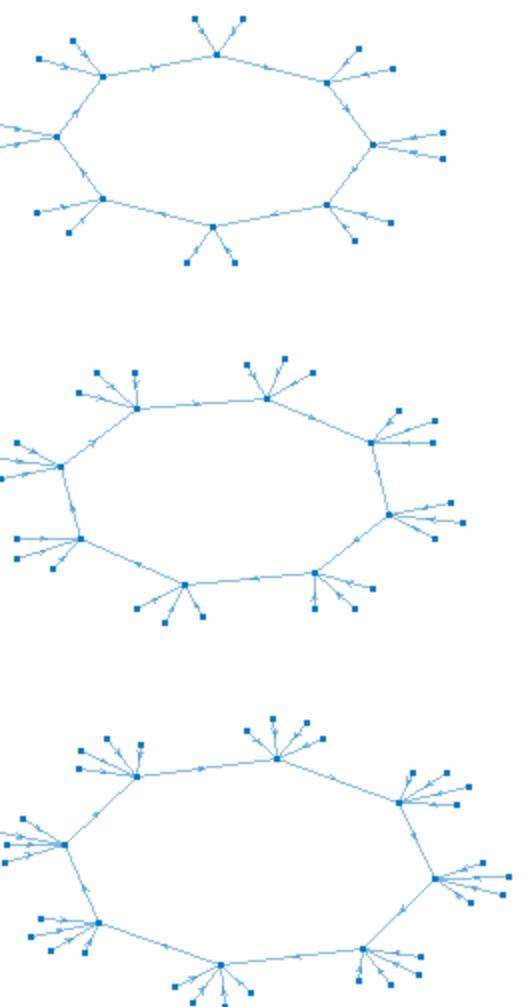
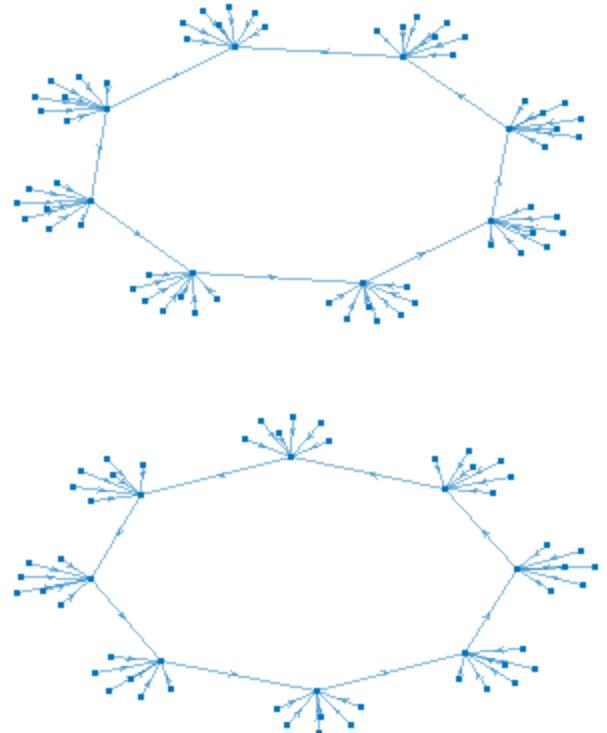


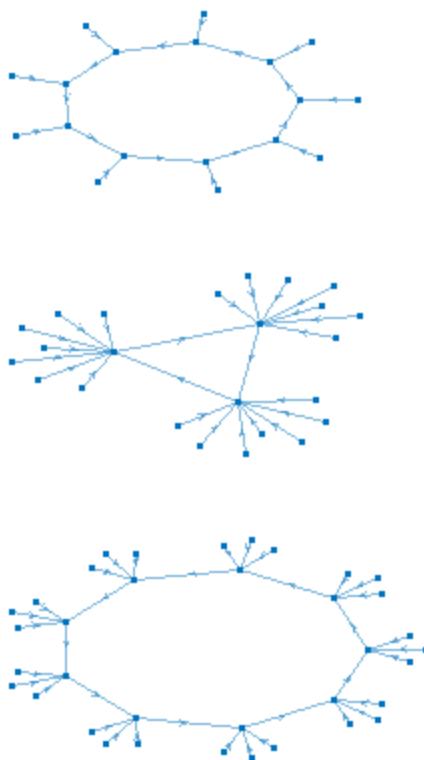
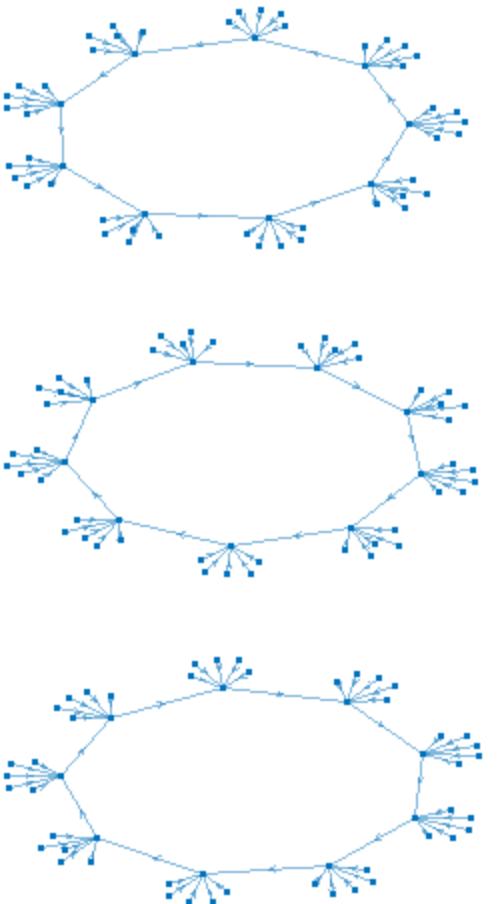
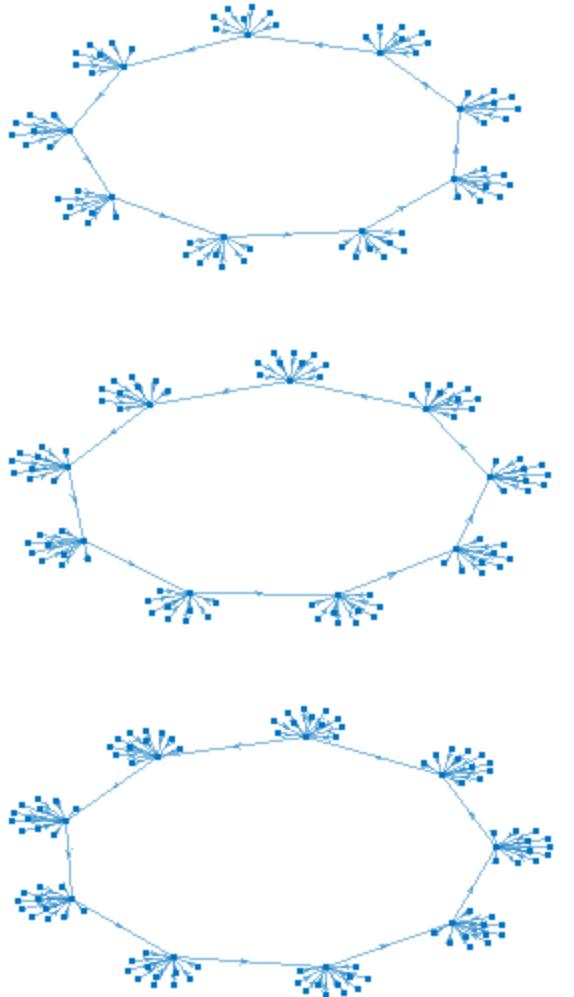


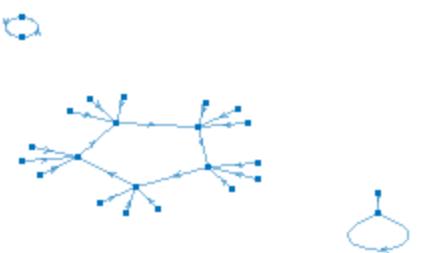
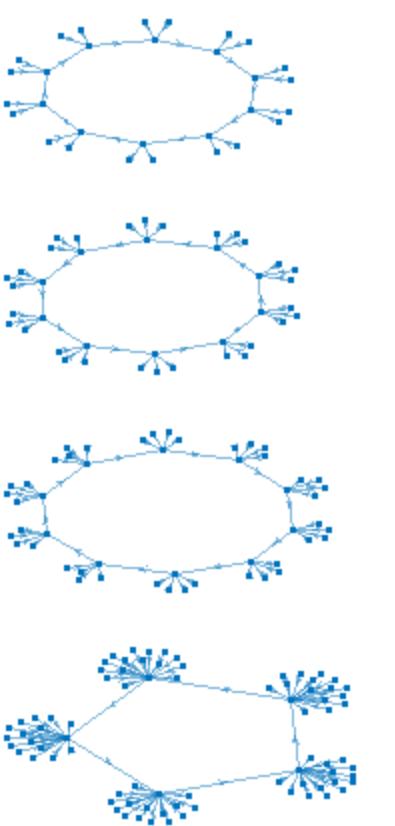
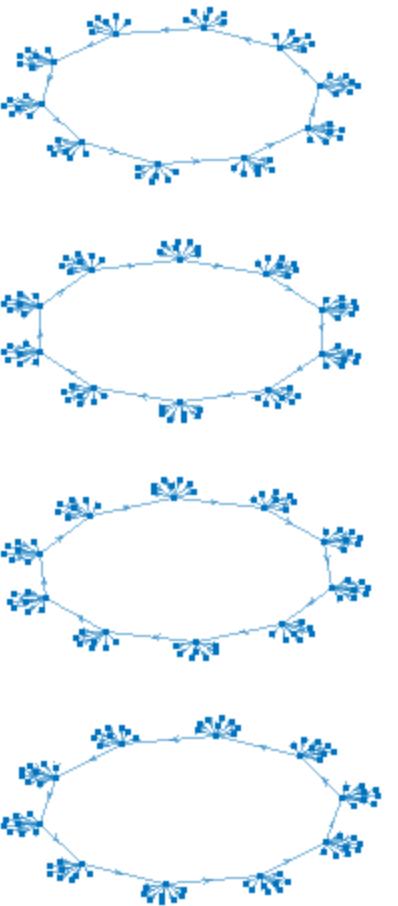
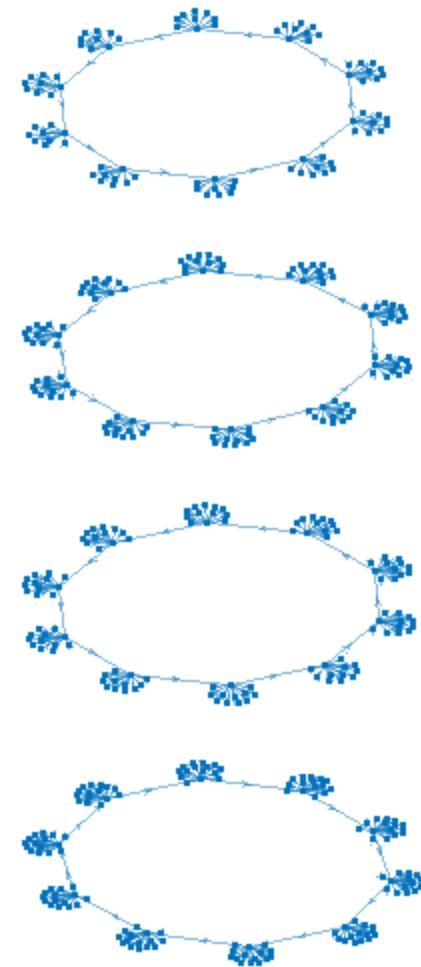


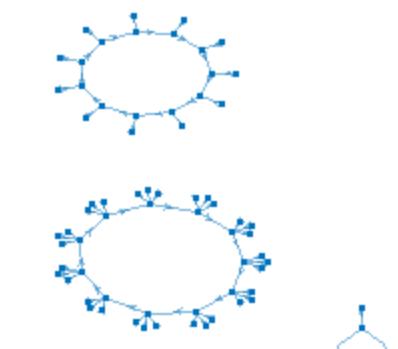
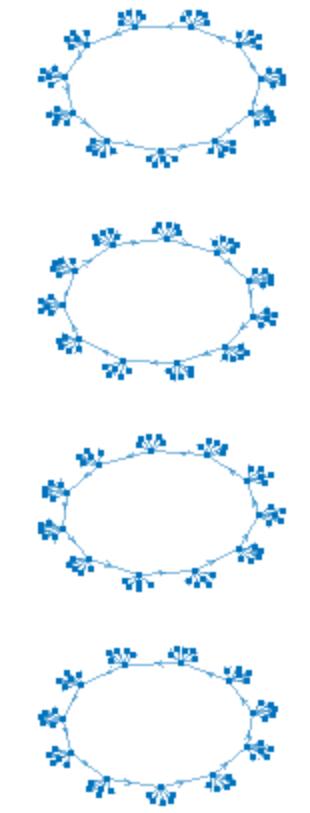
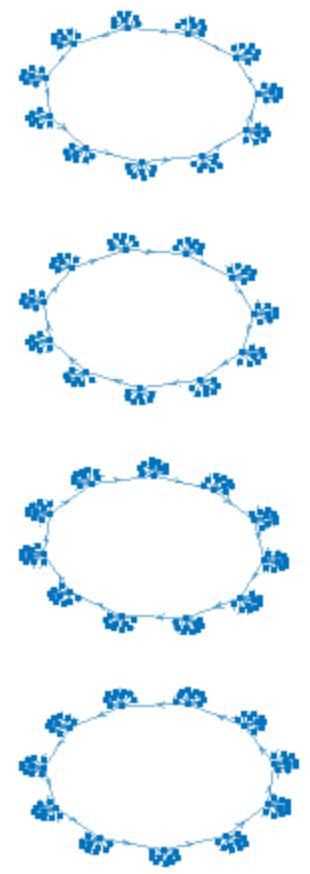
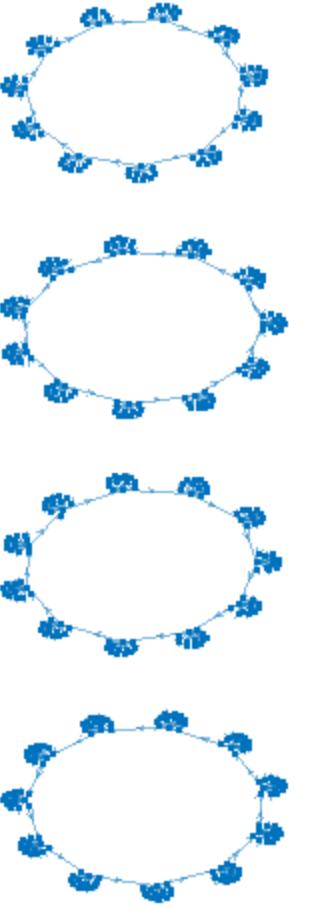
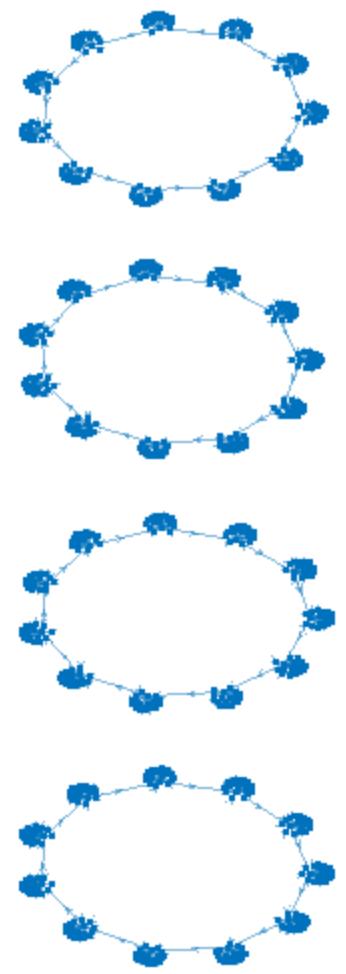




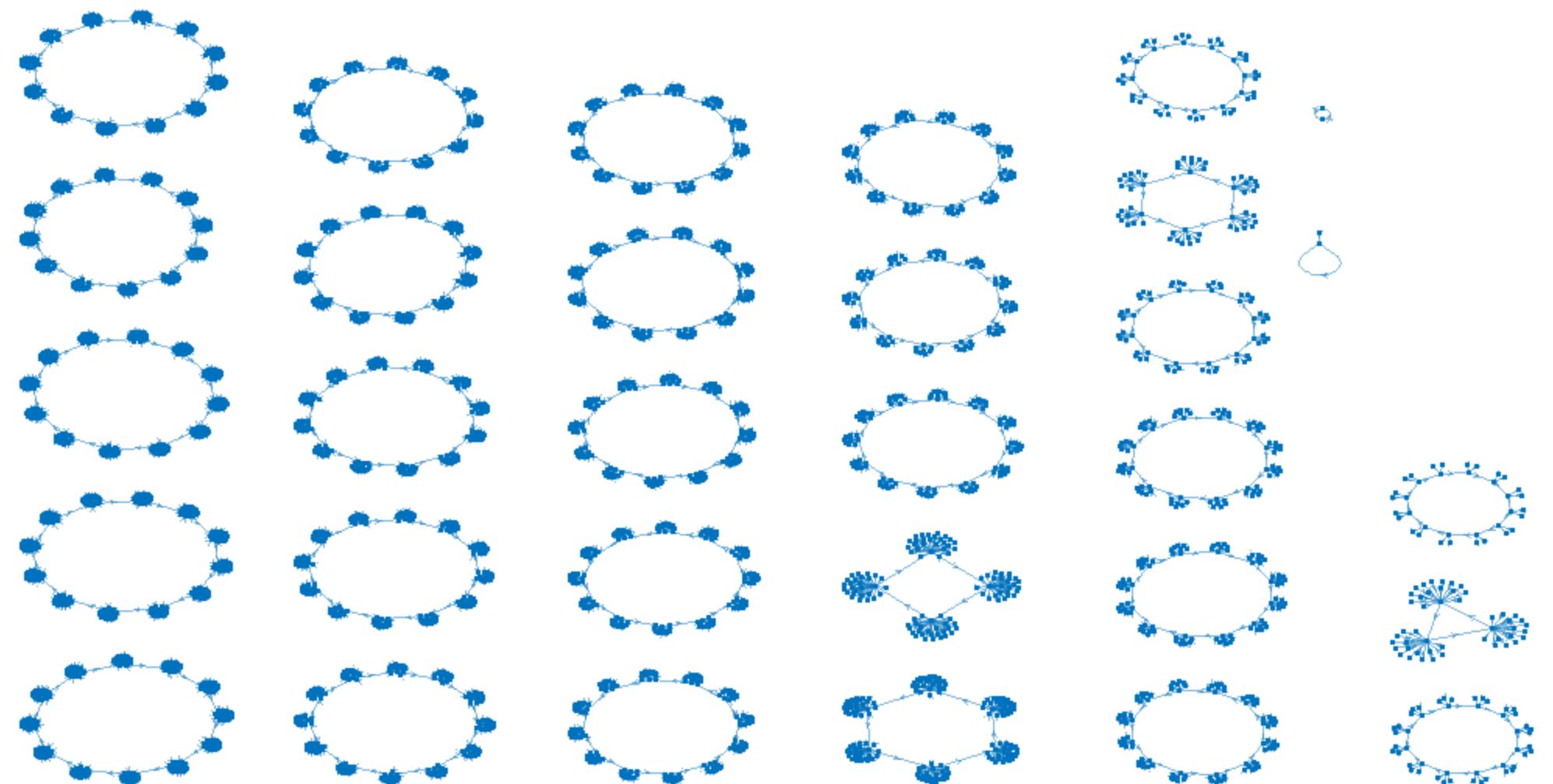


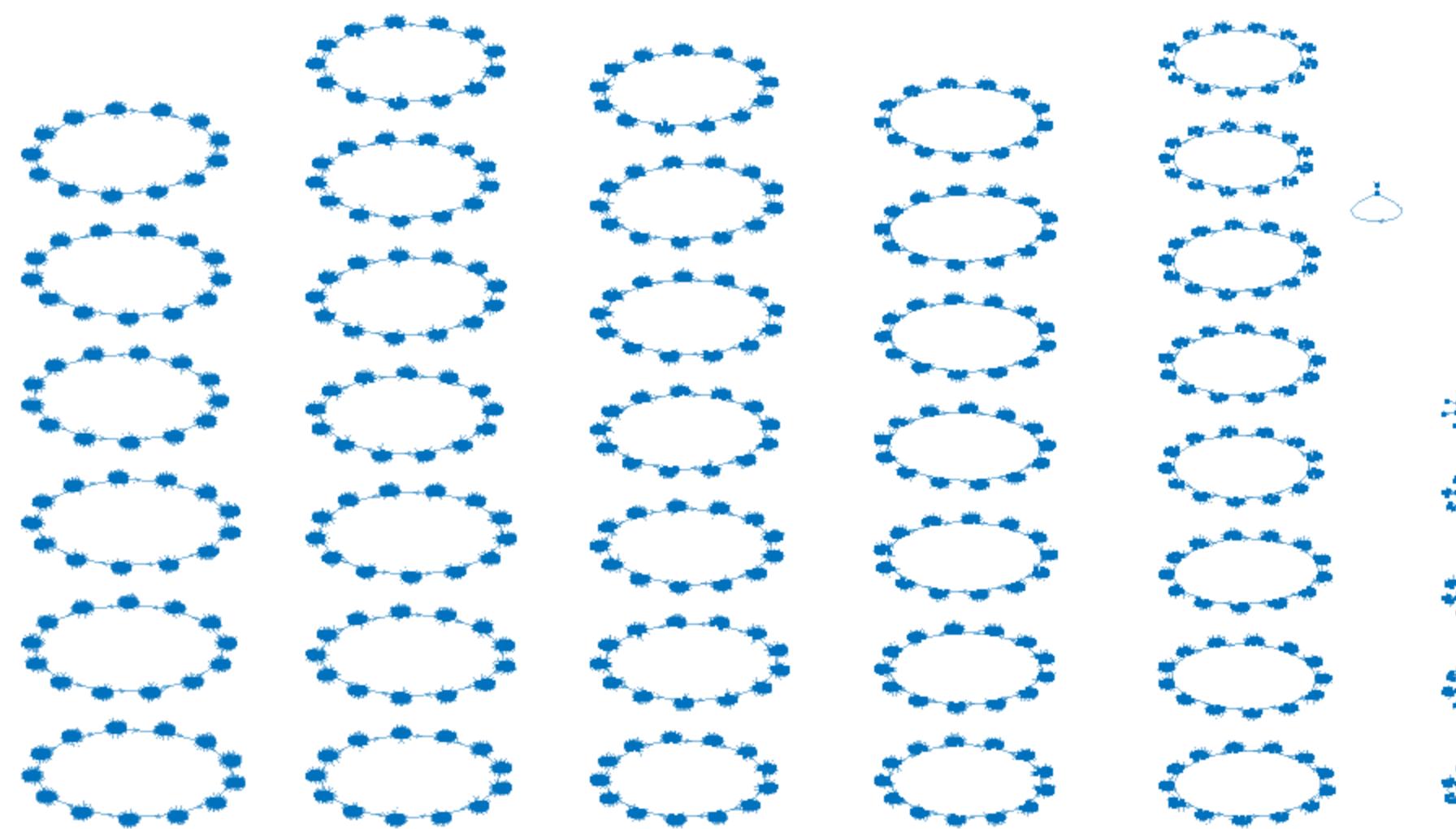


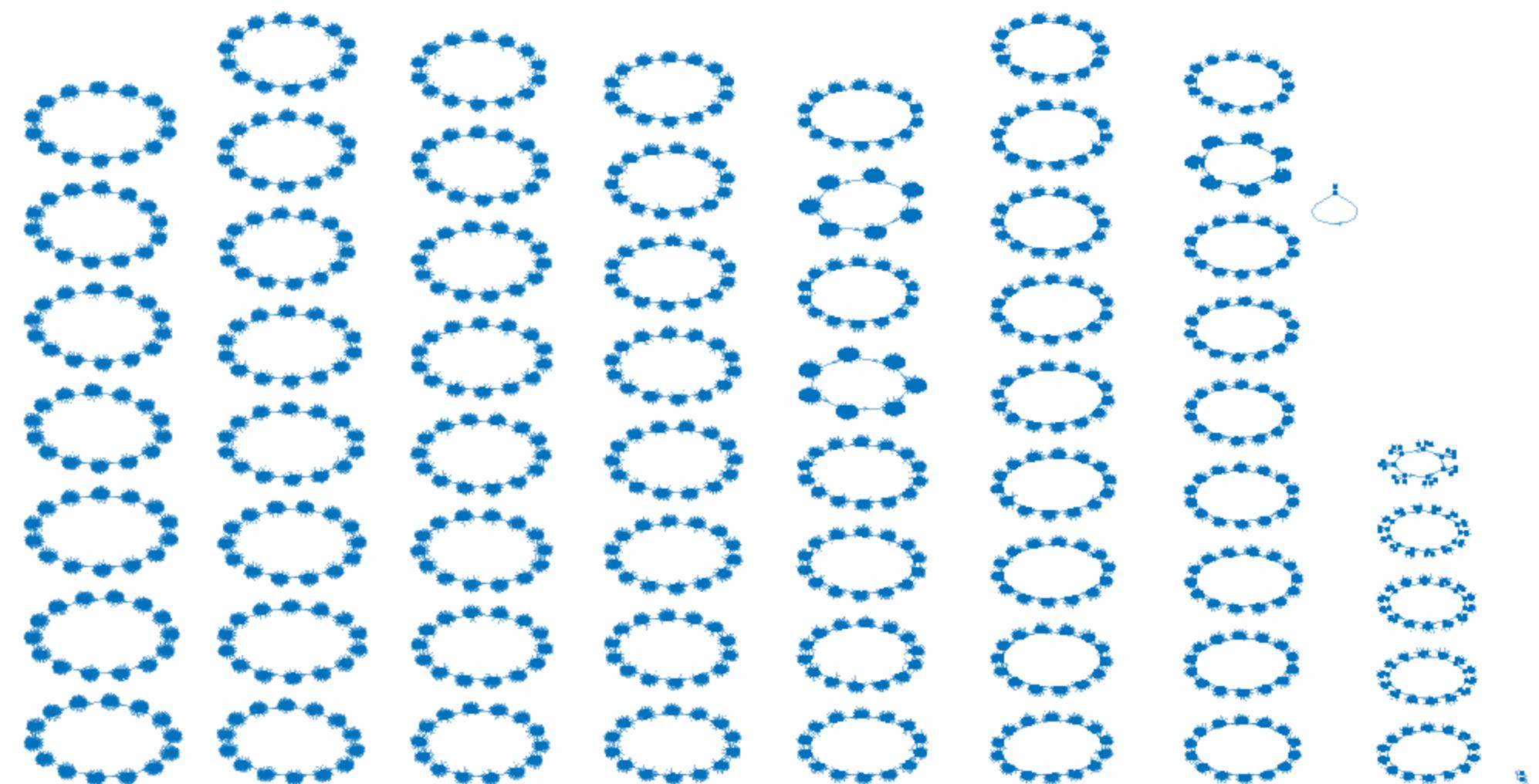


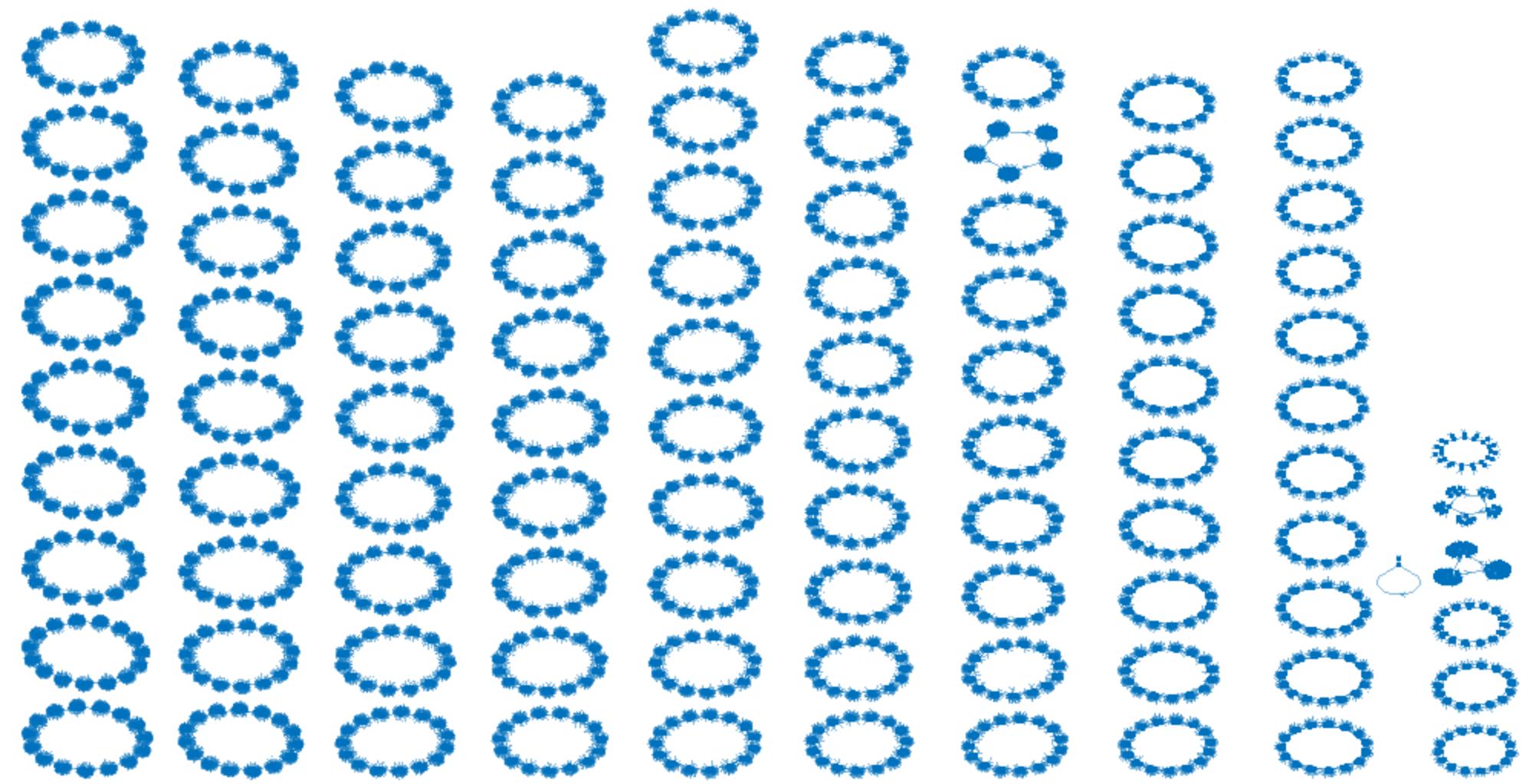


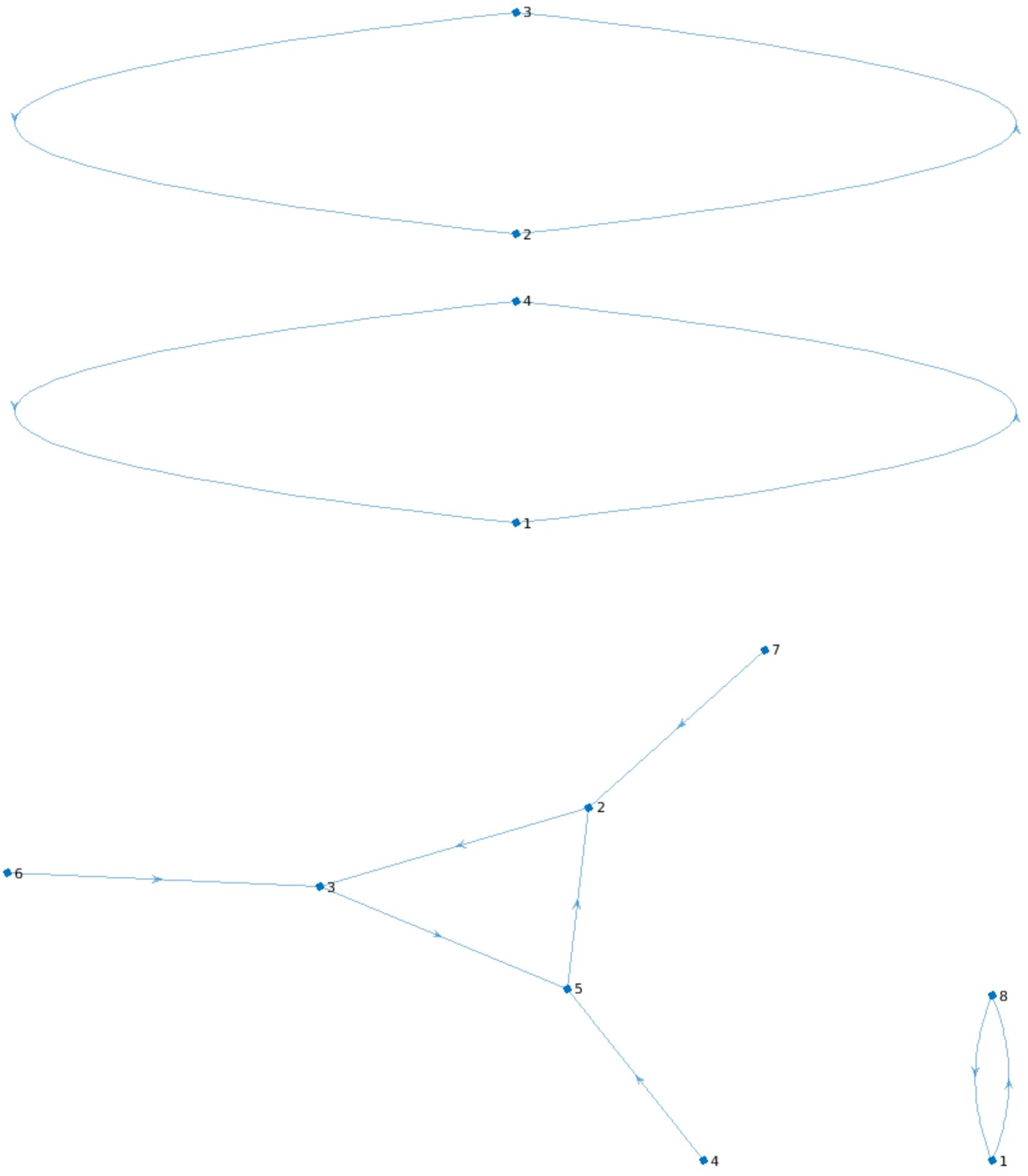
1

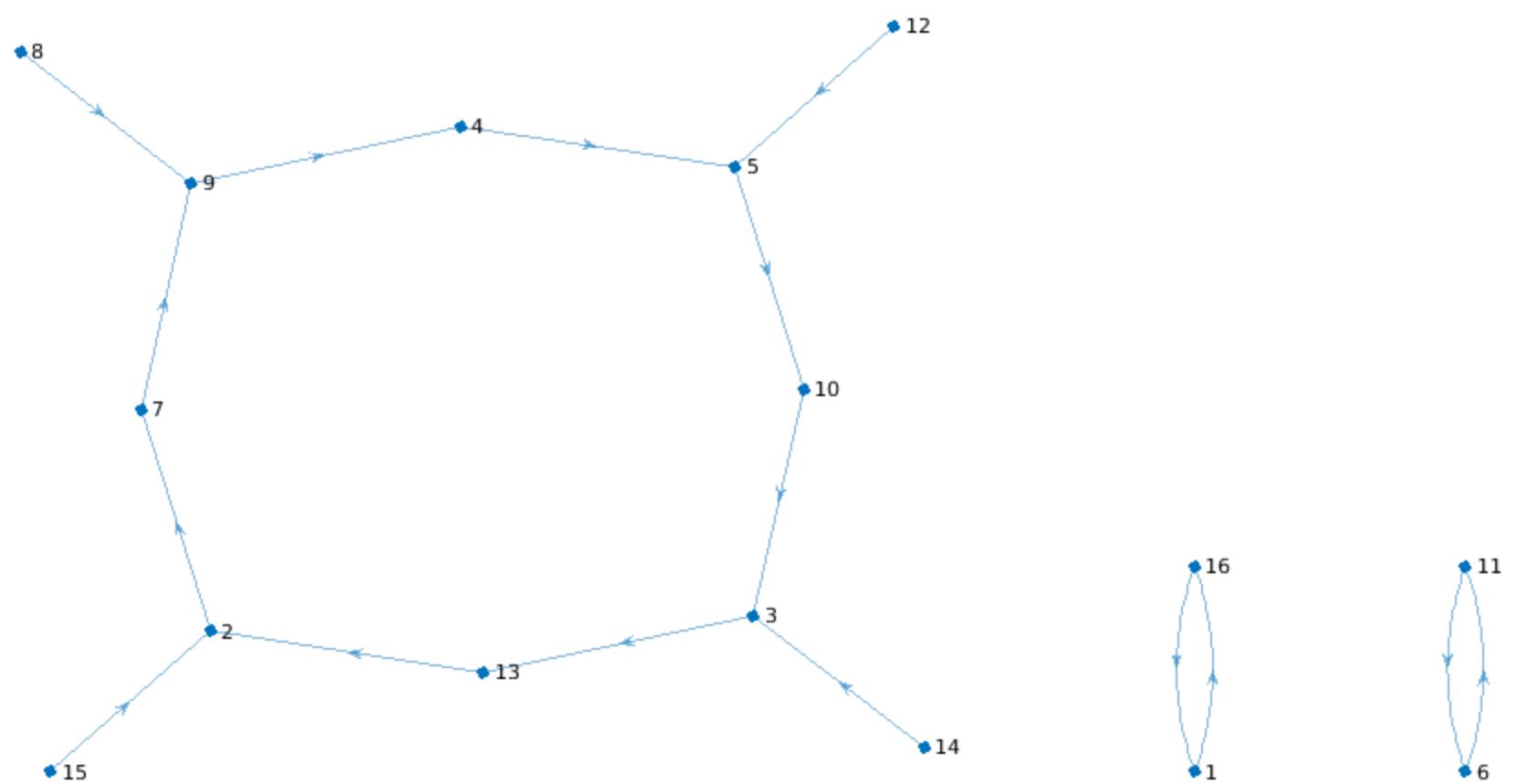


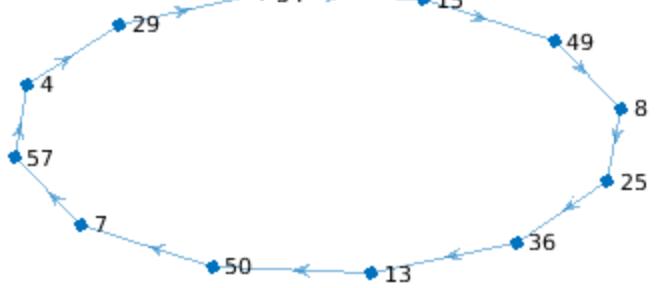
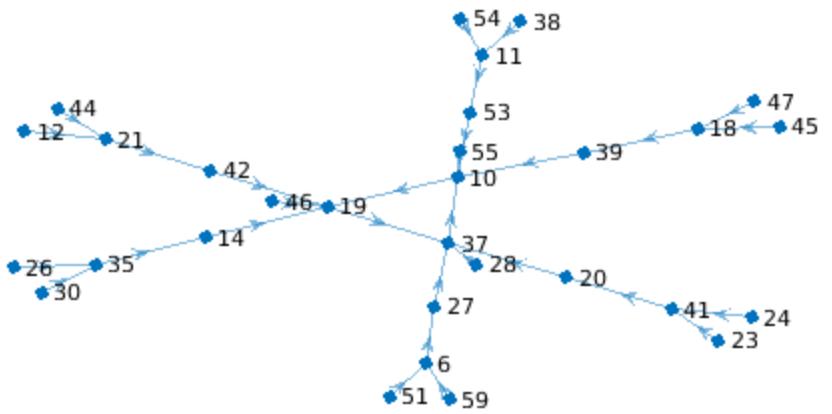
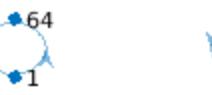
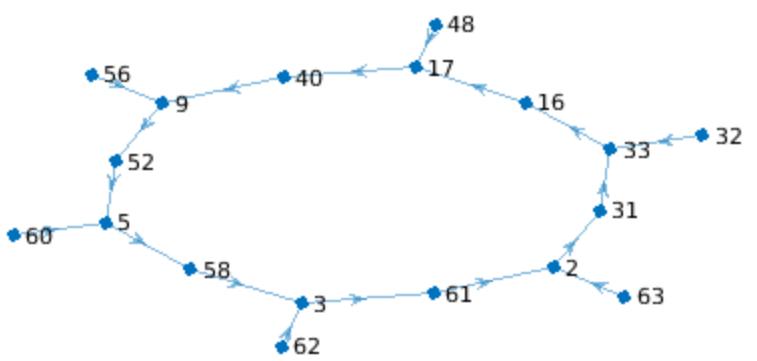
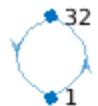
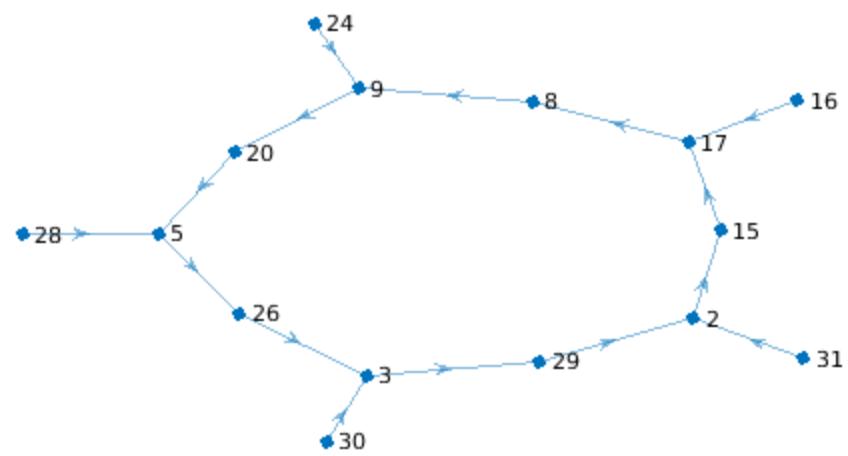
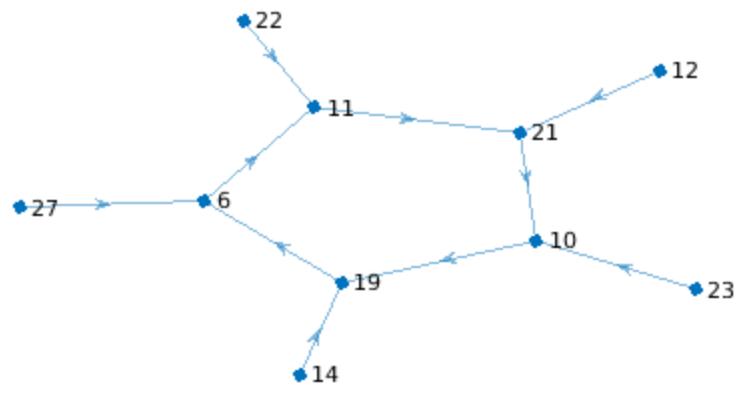


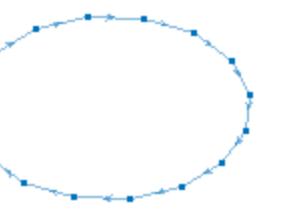
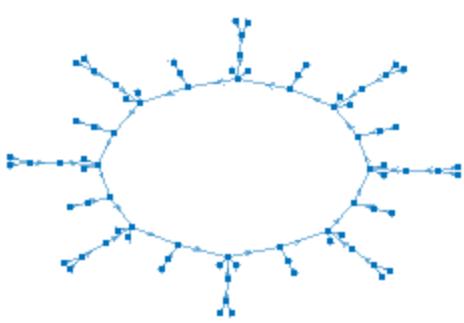
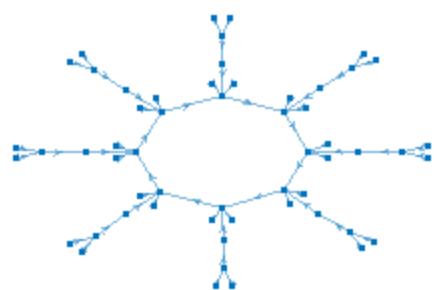
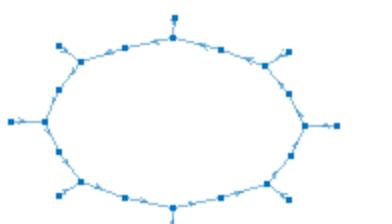
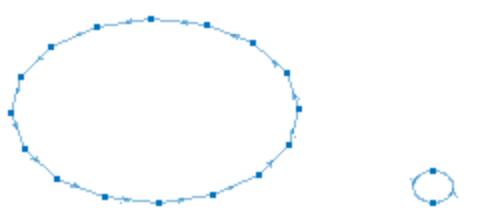
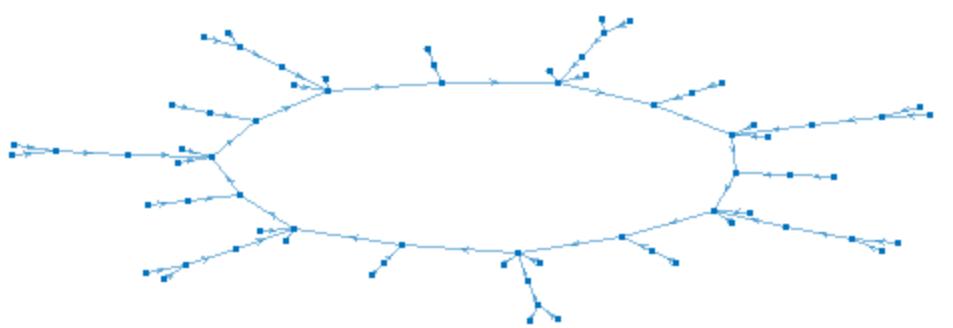
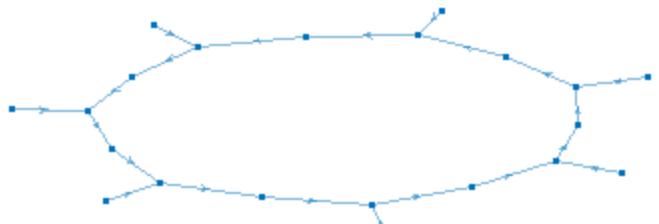


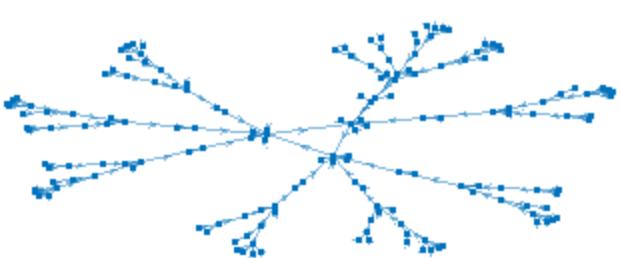
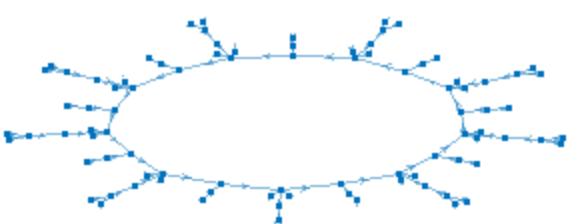
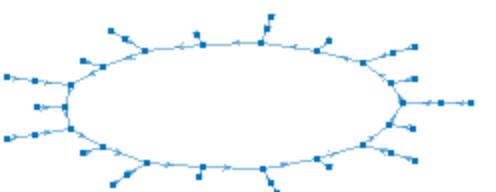
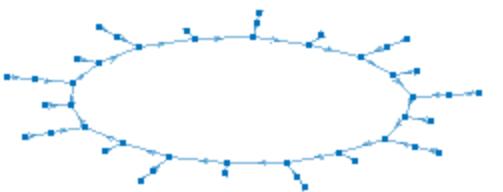




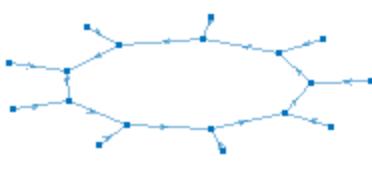


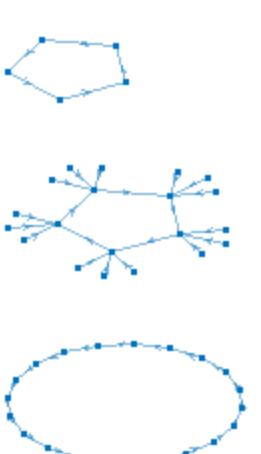
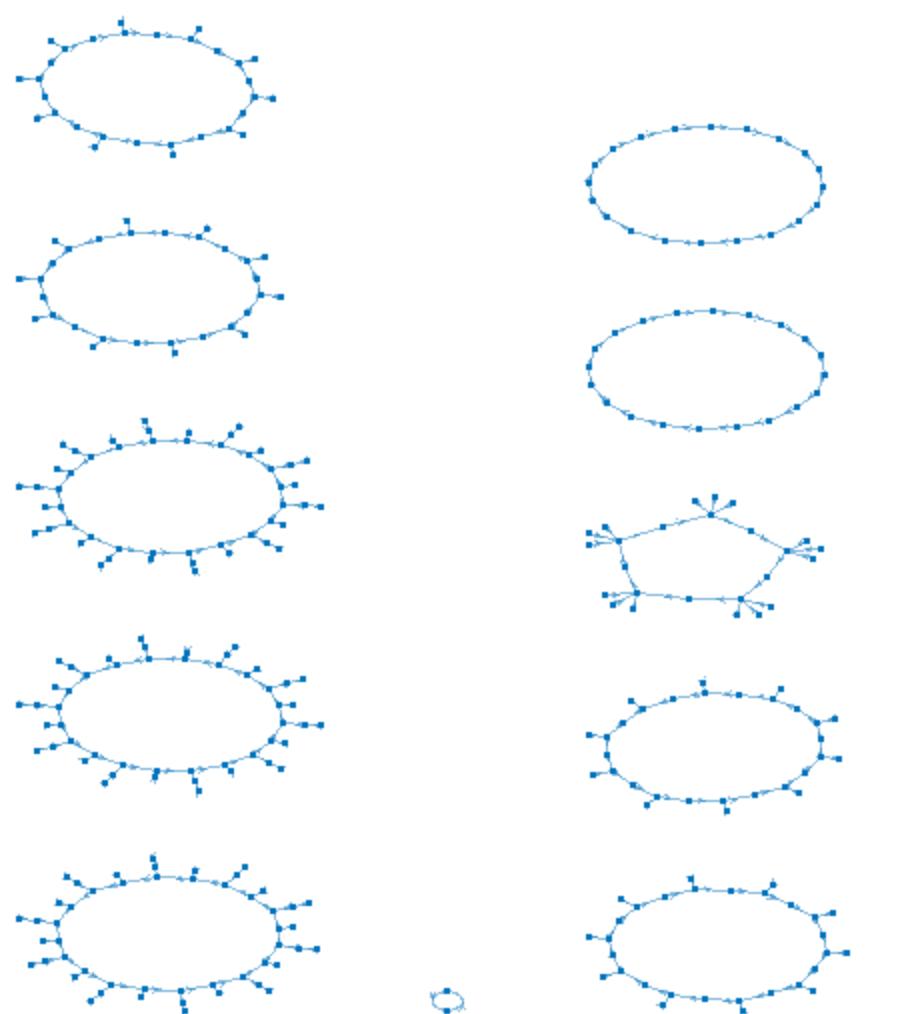
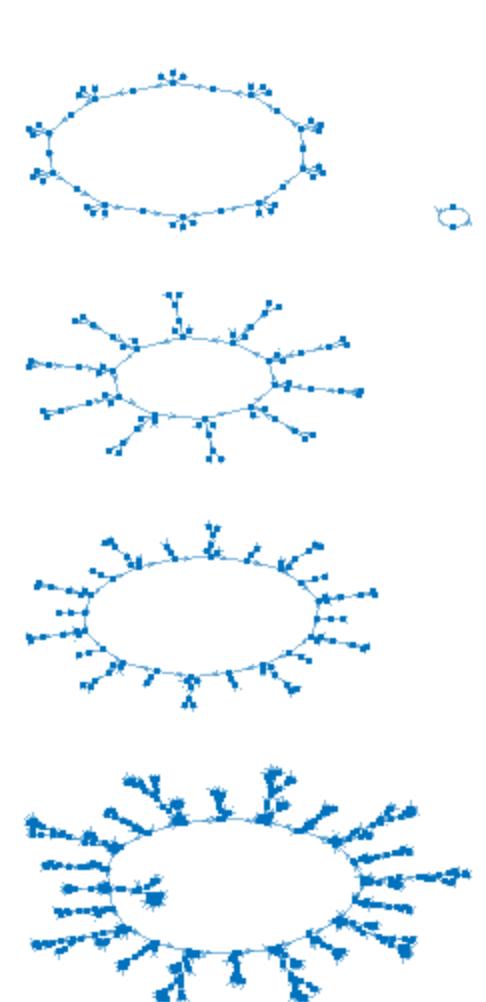


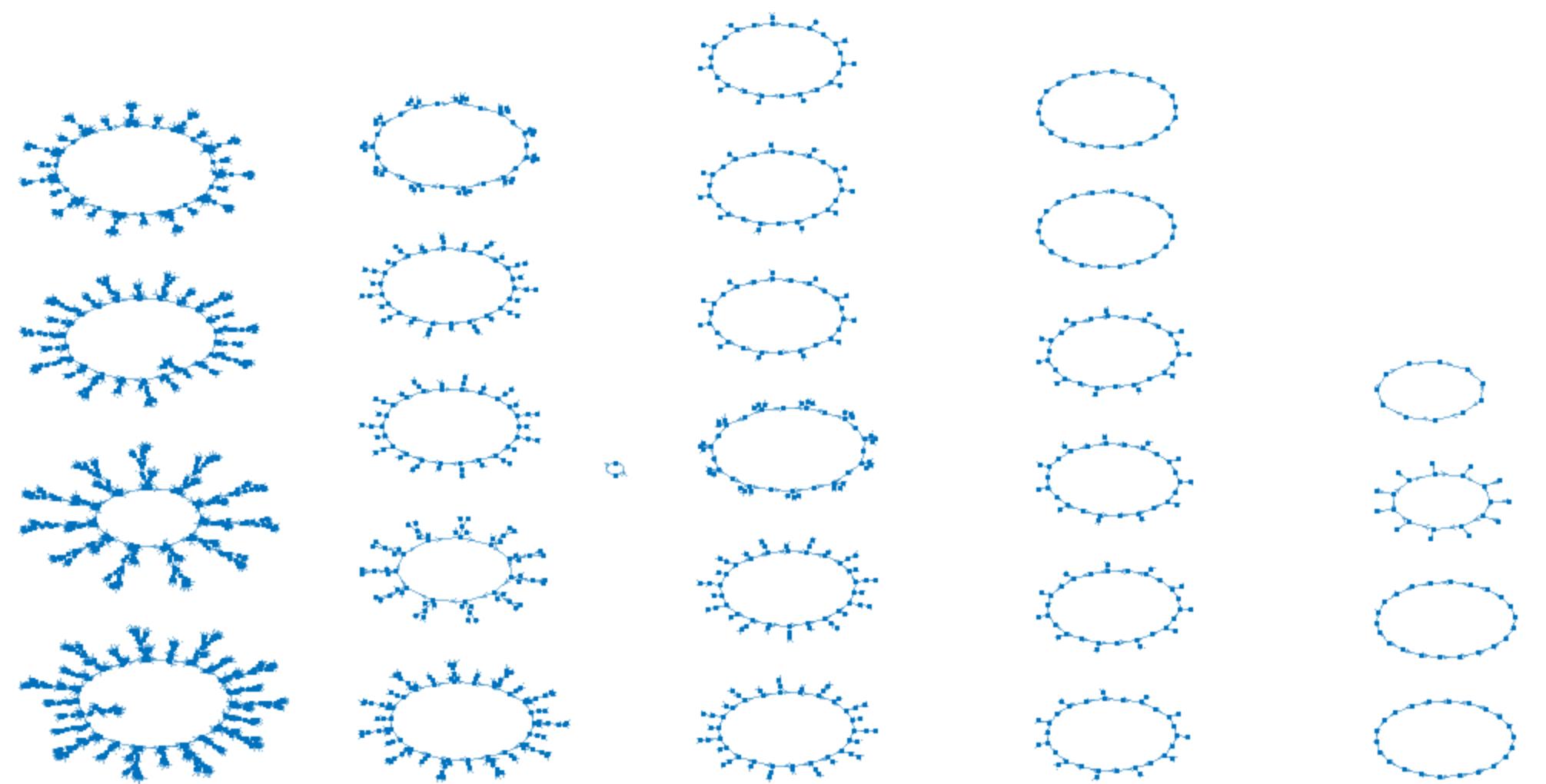


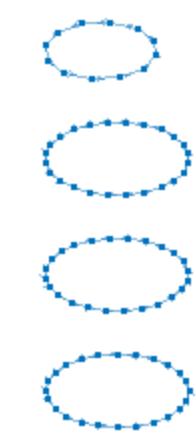
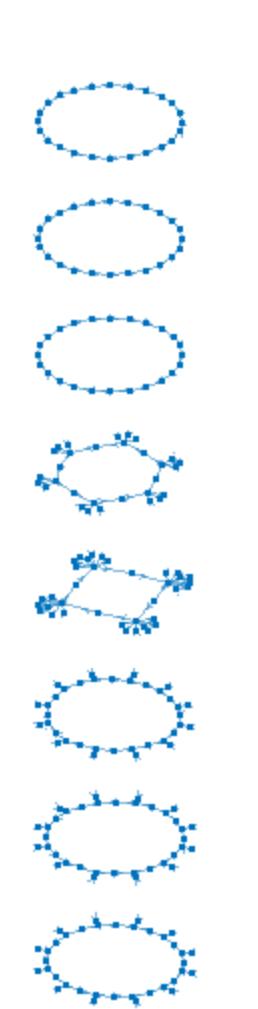
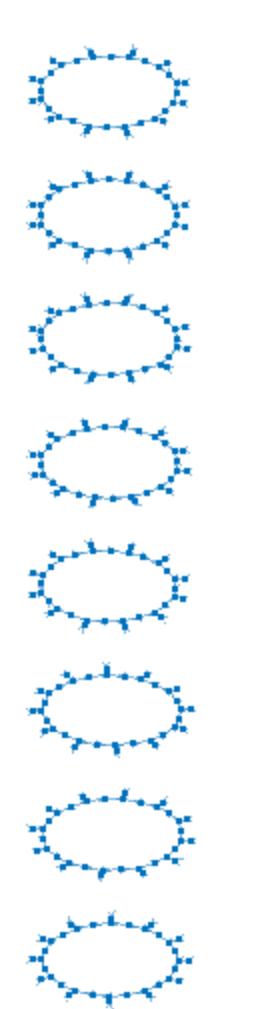
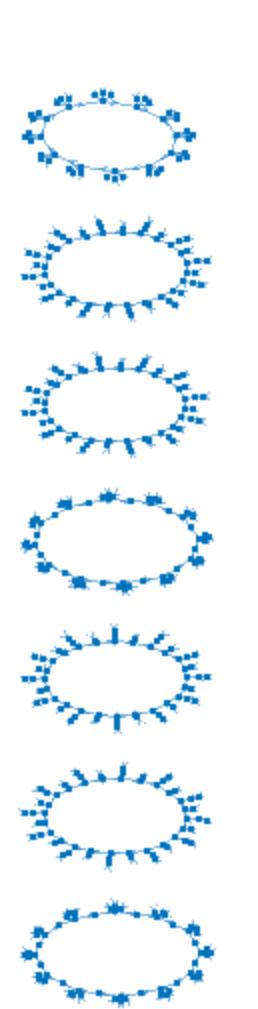
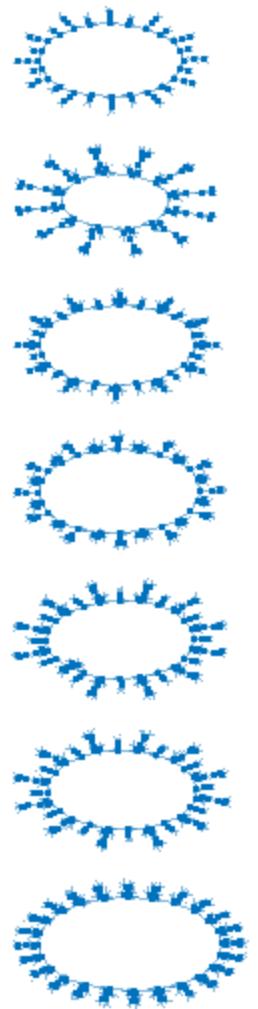
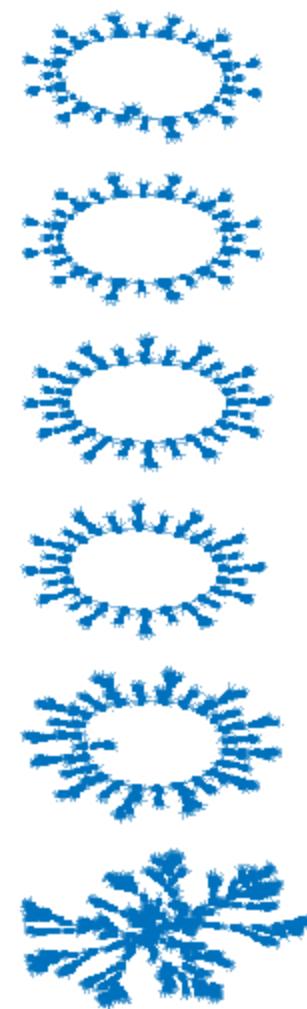


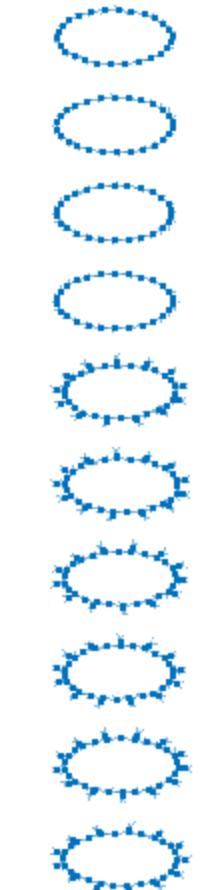
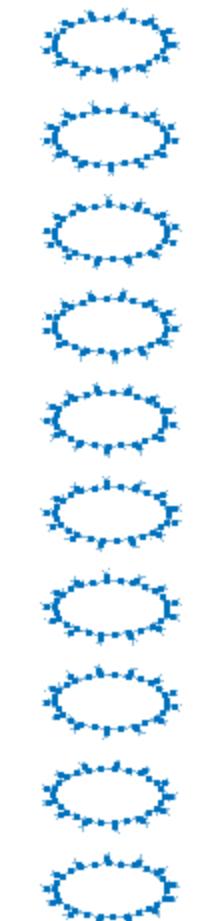
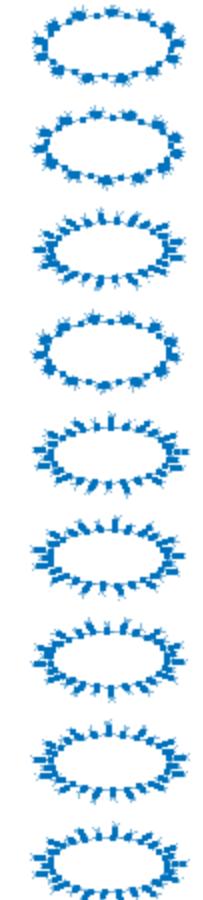
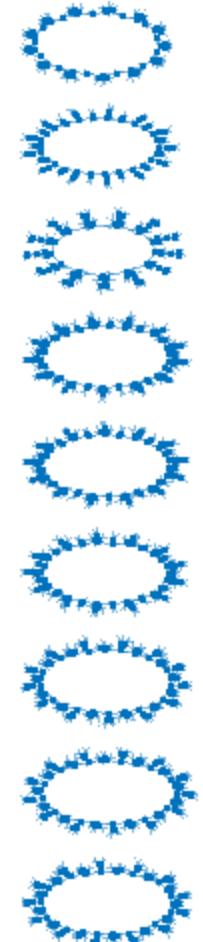
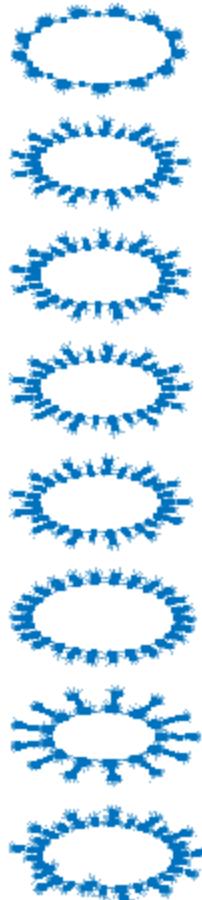
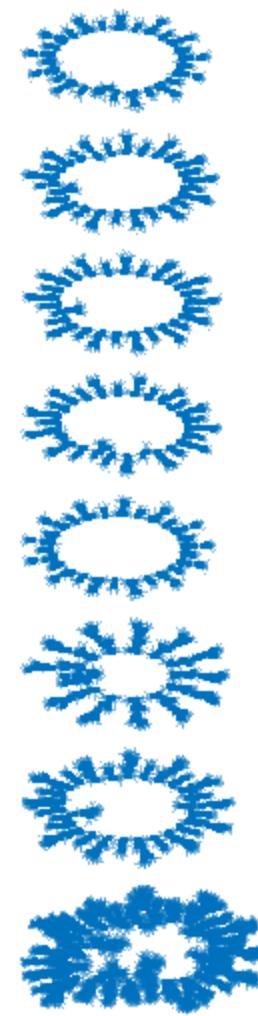
o

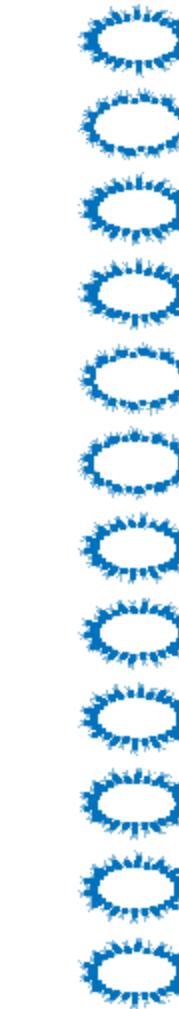
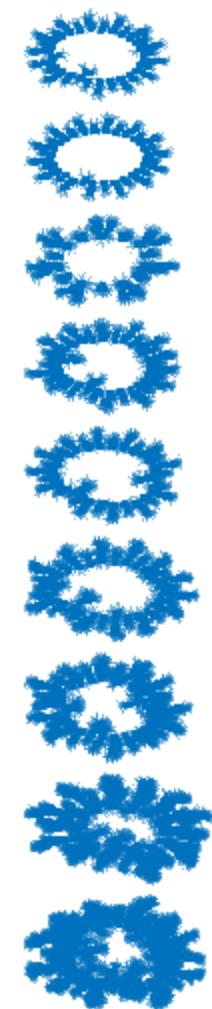


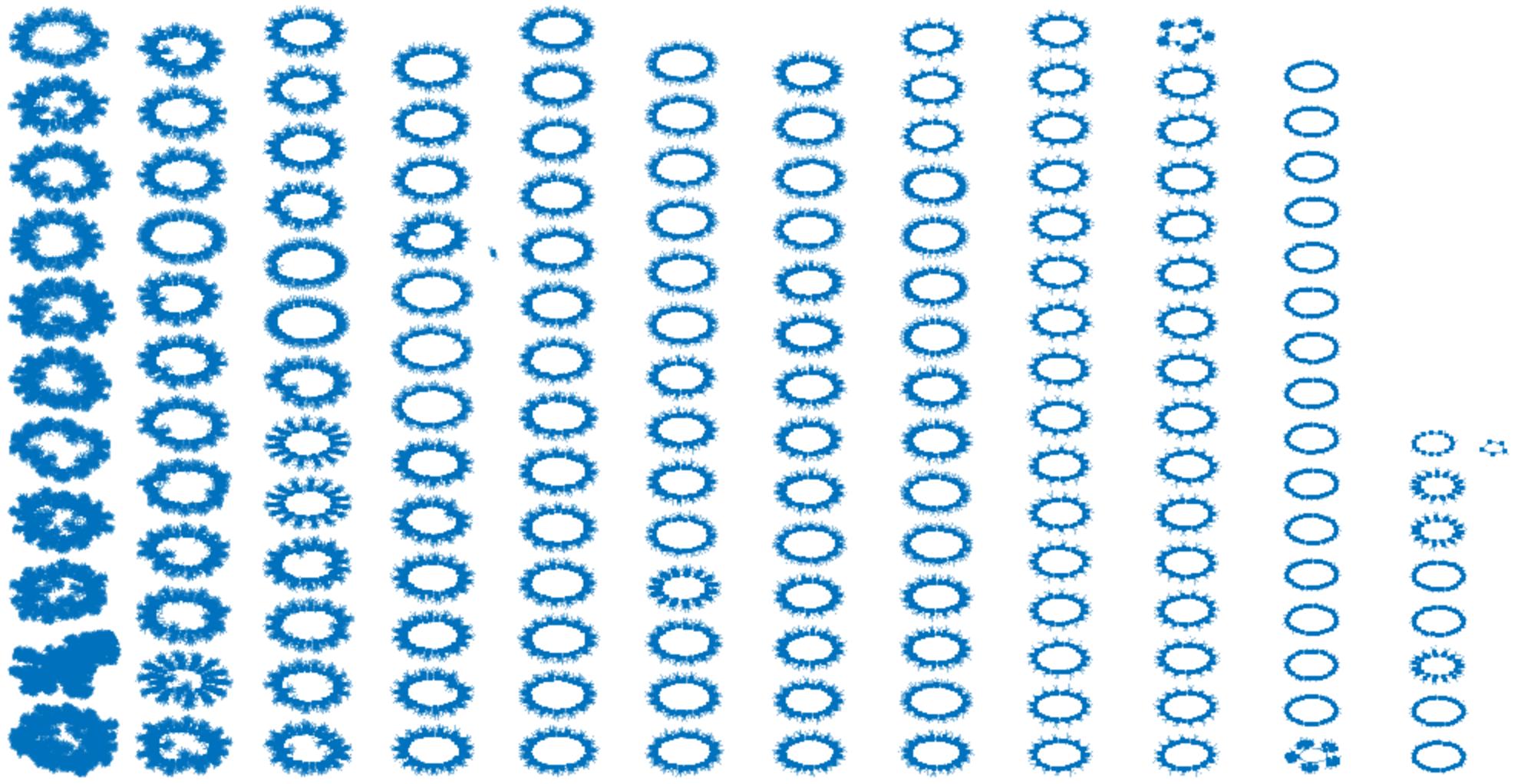


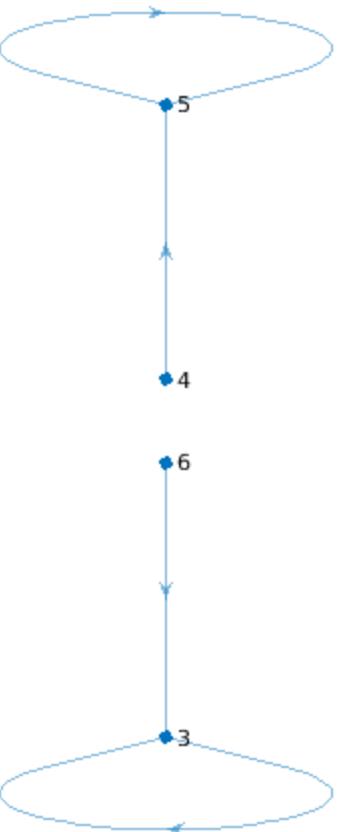
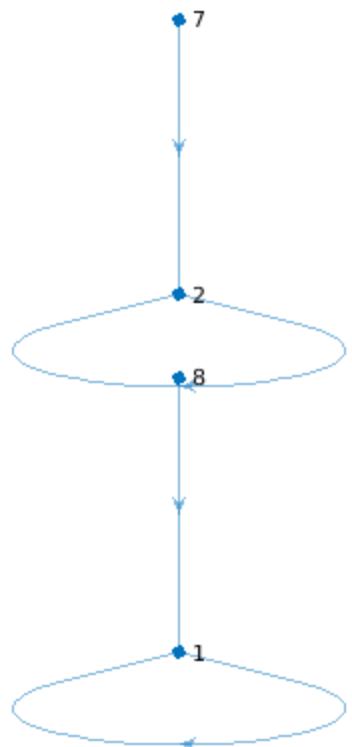
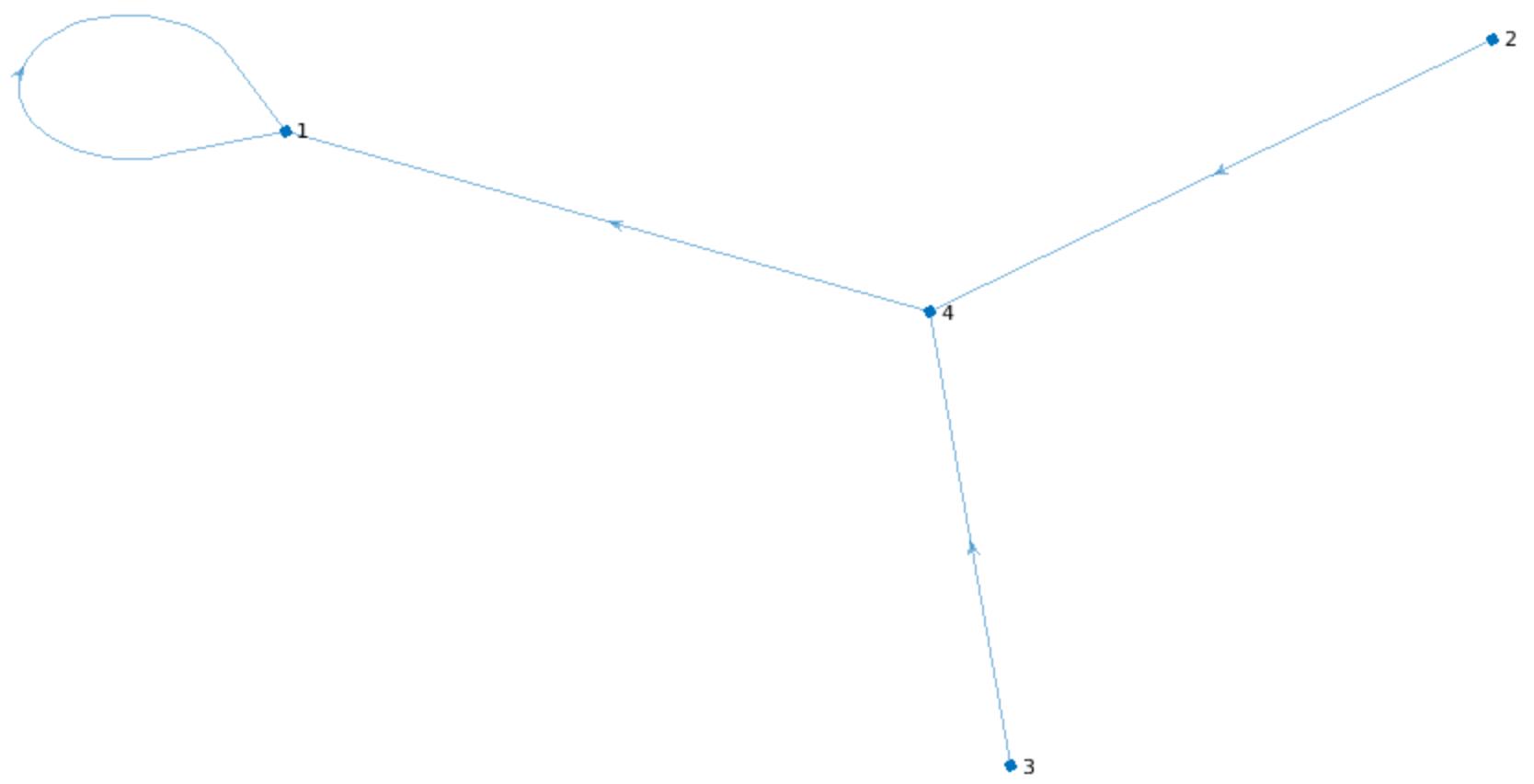


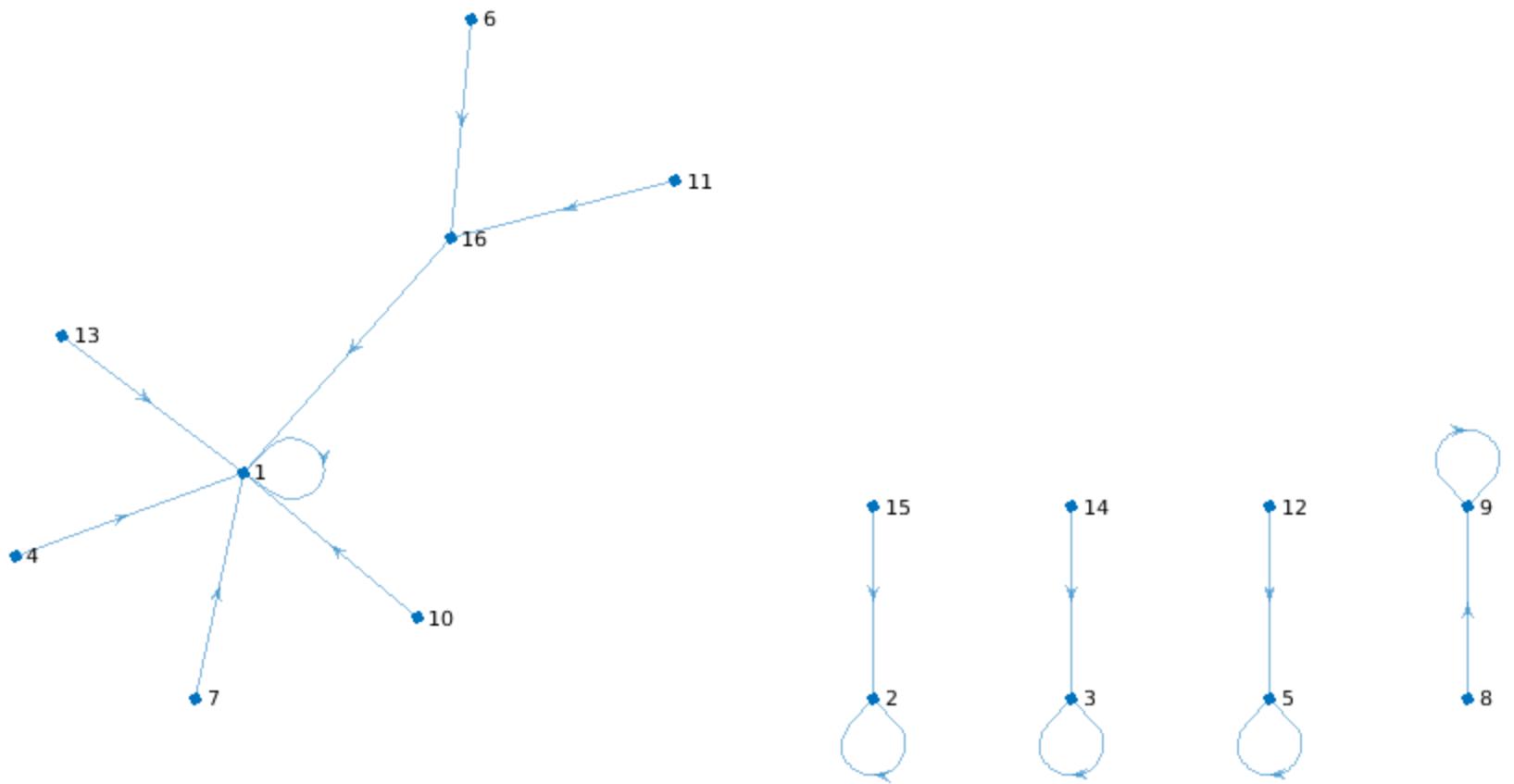


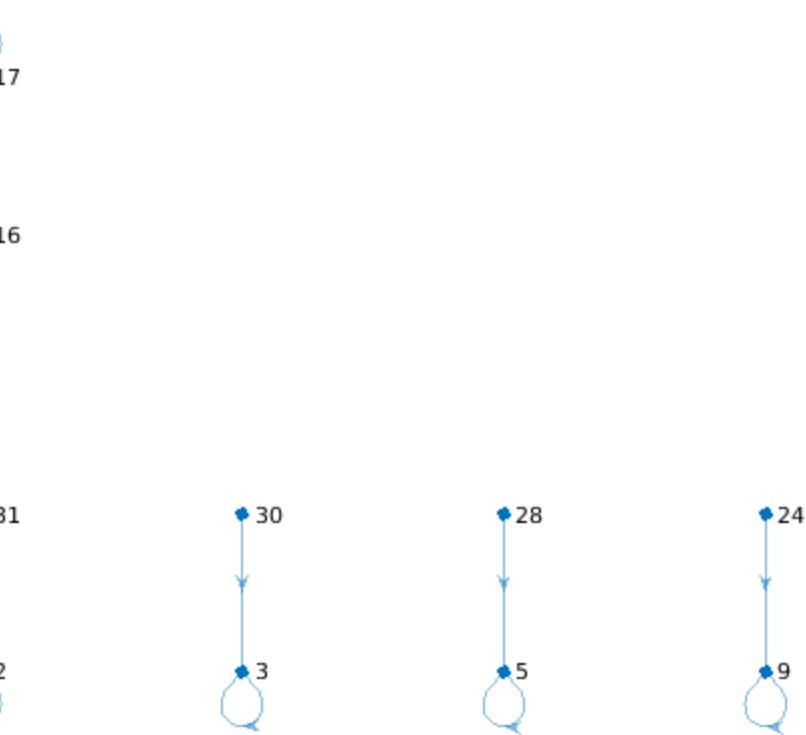
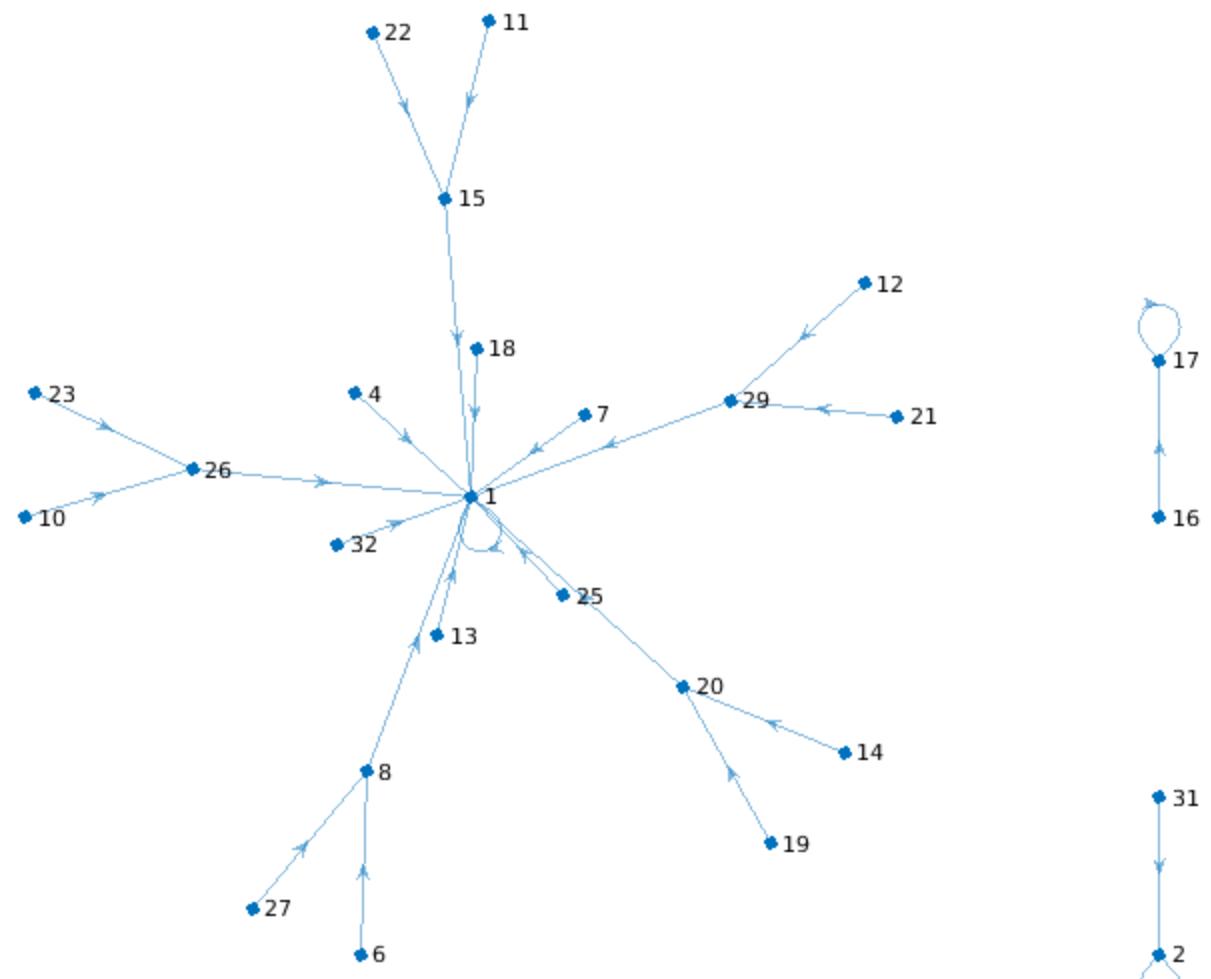


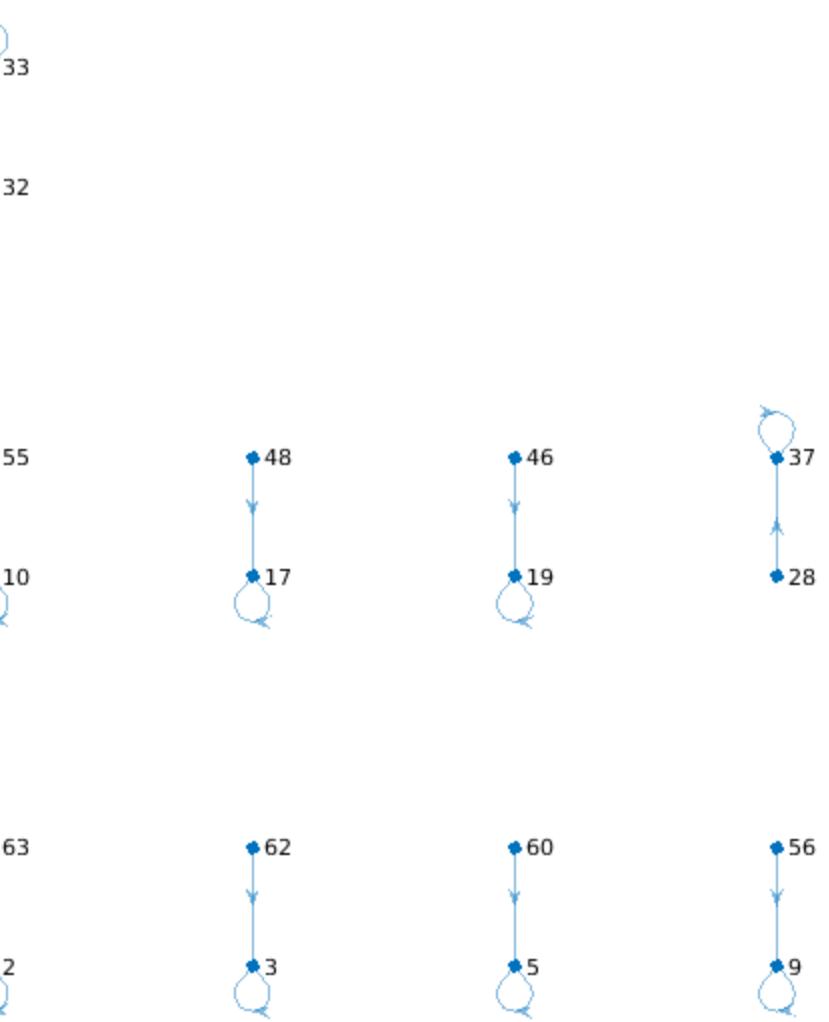
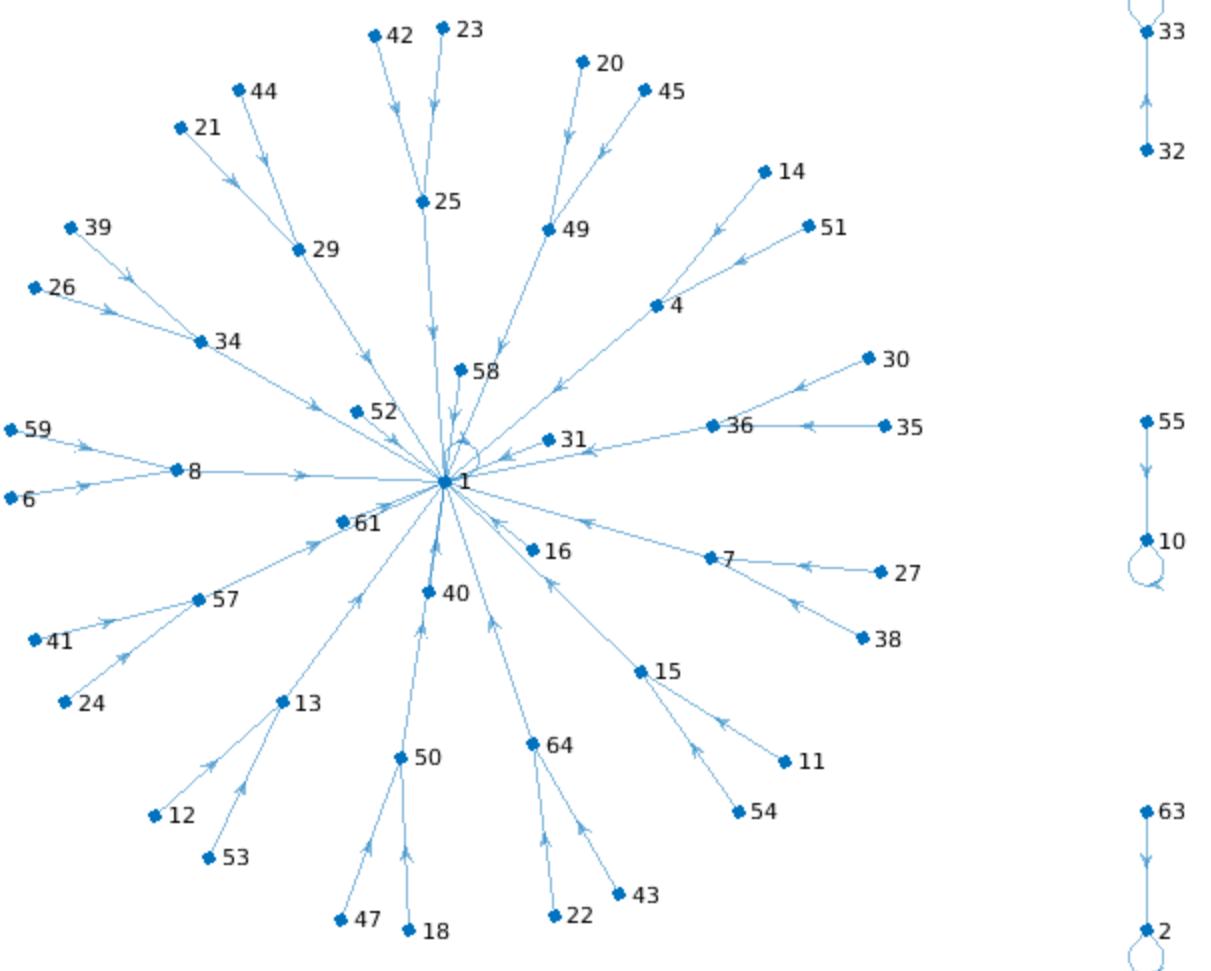


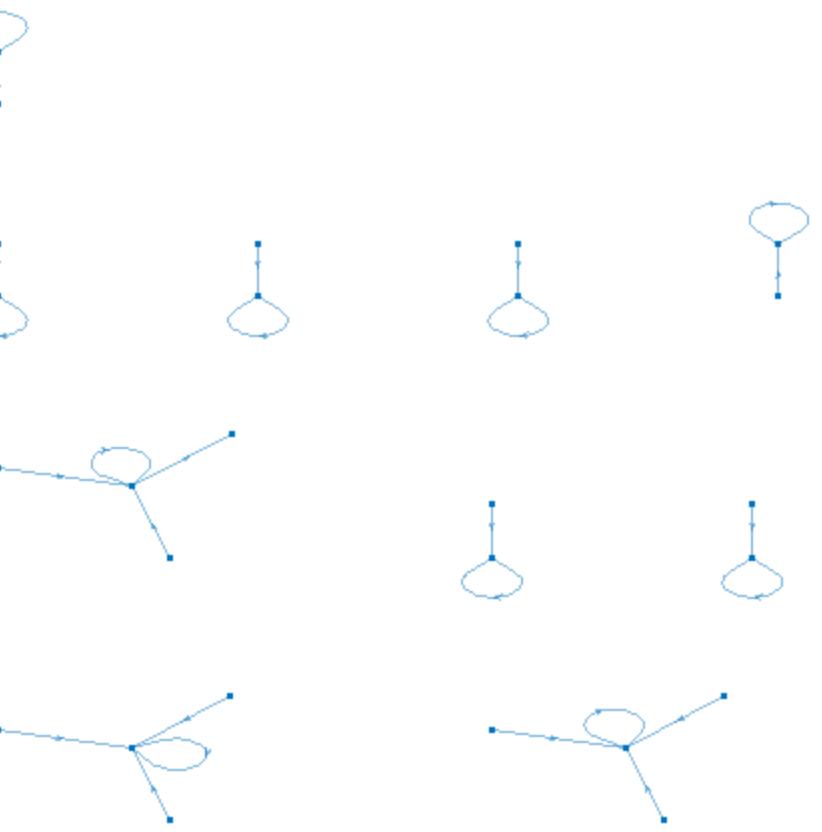
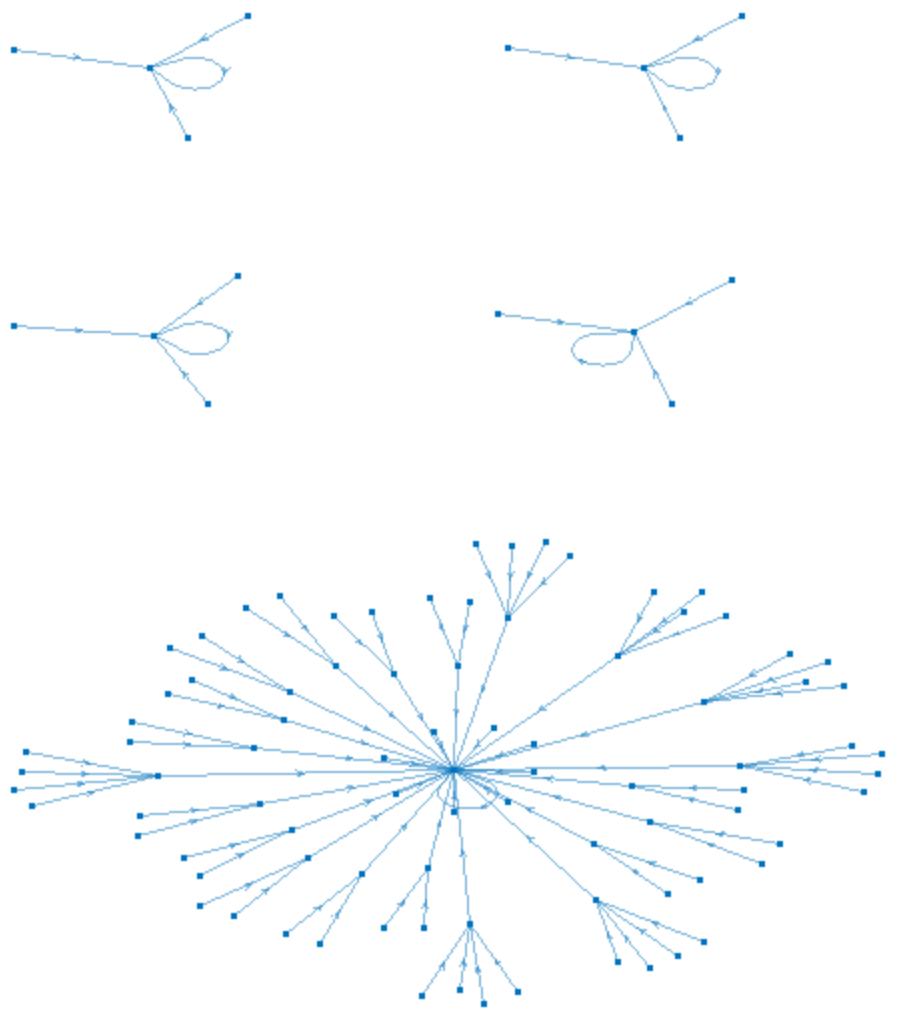


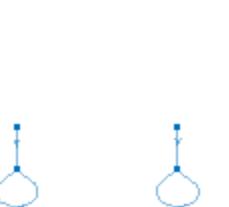
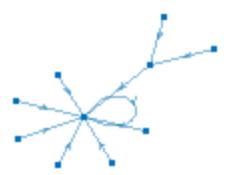
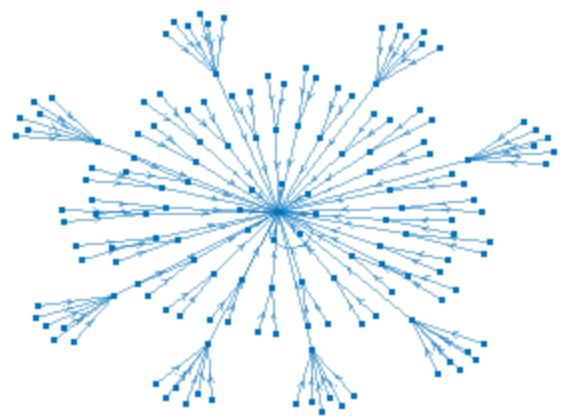
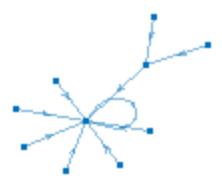


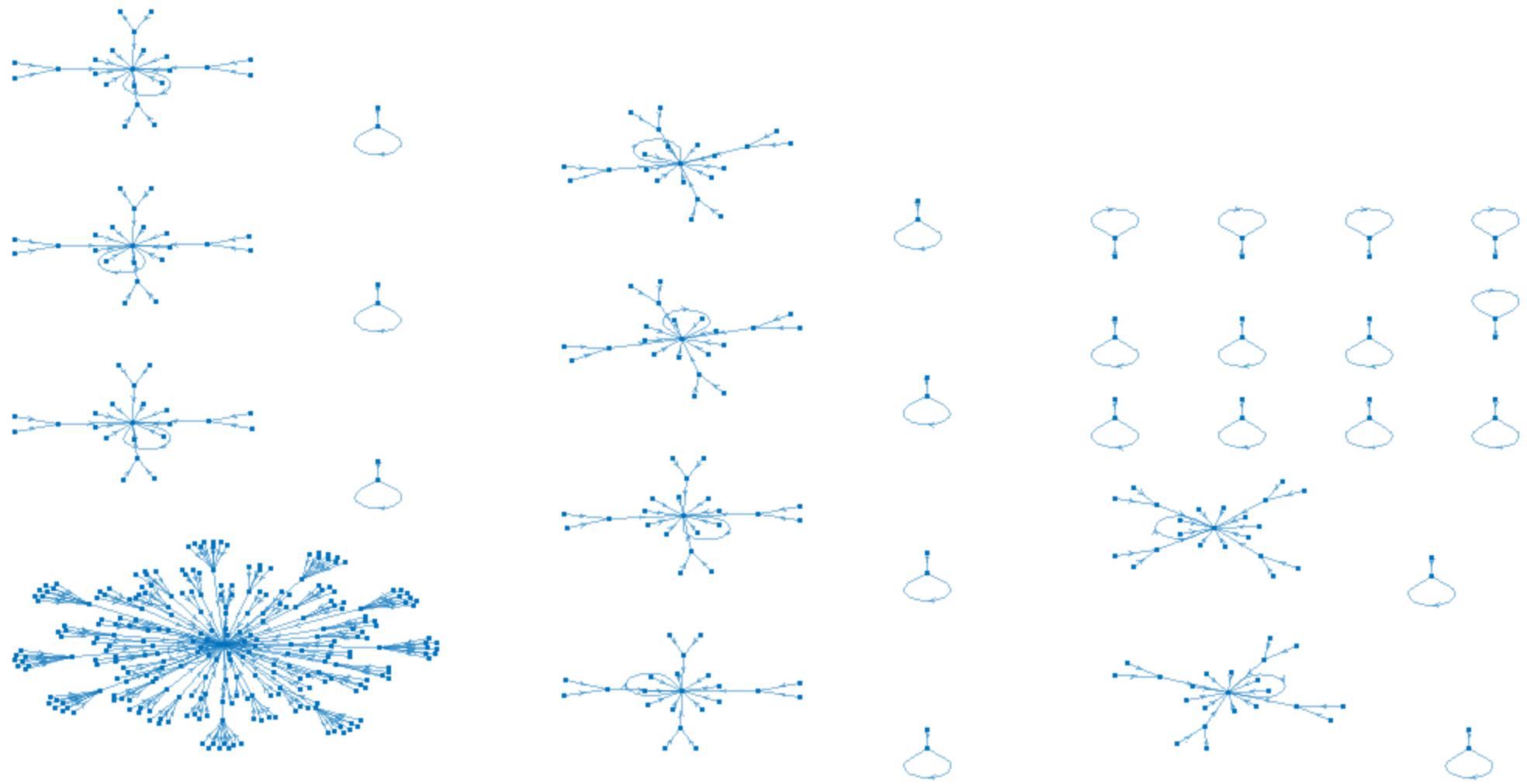


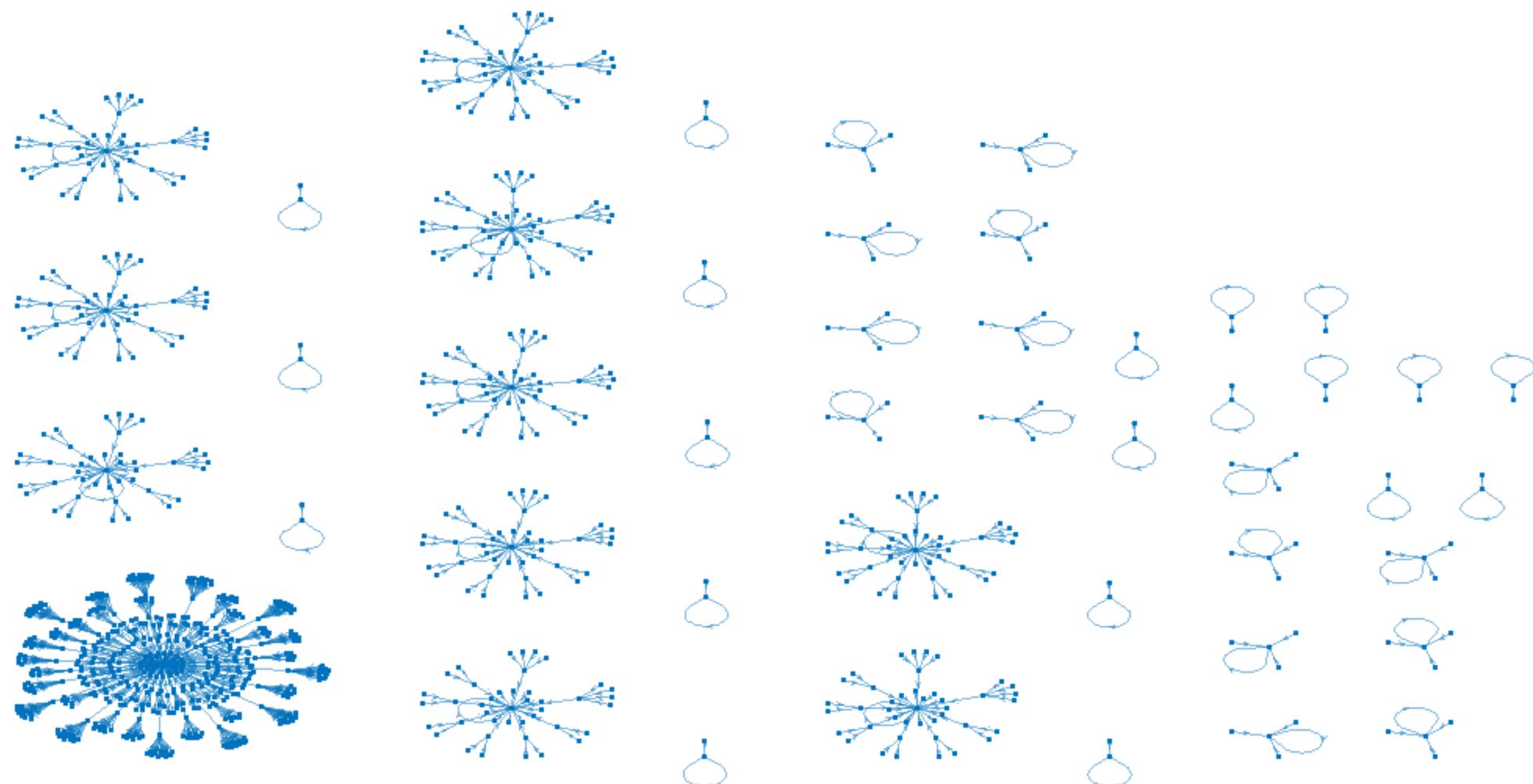


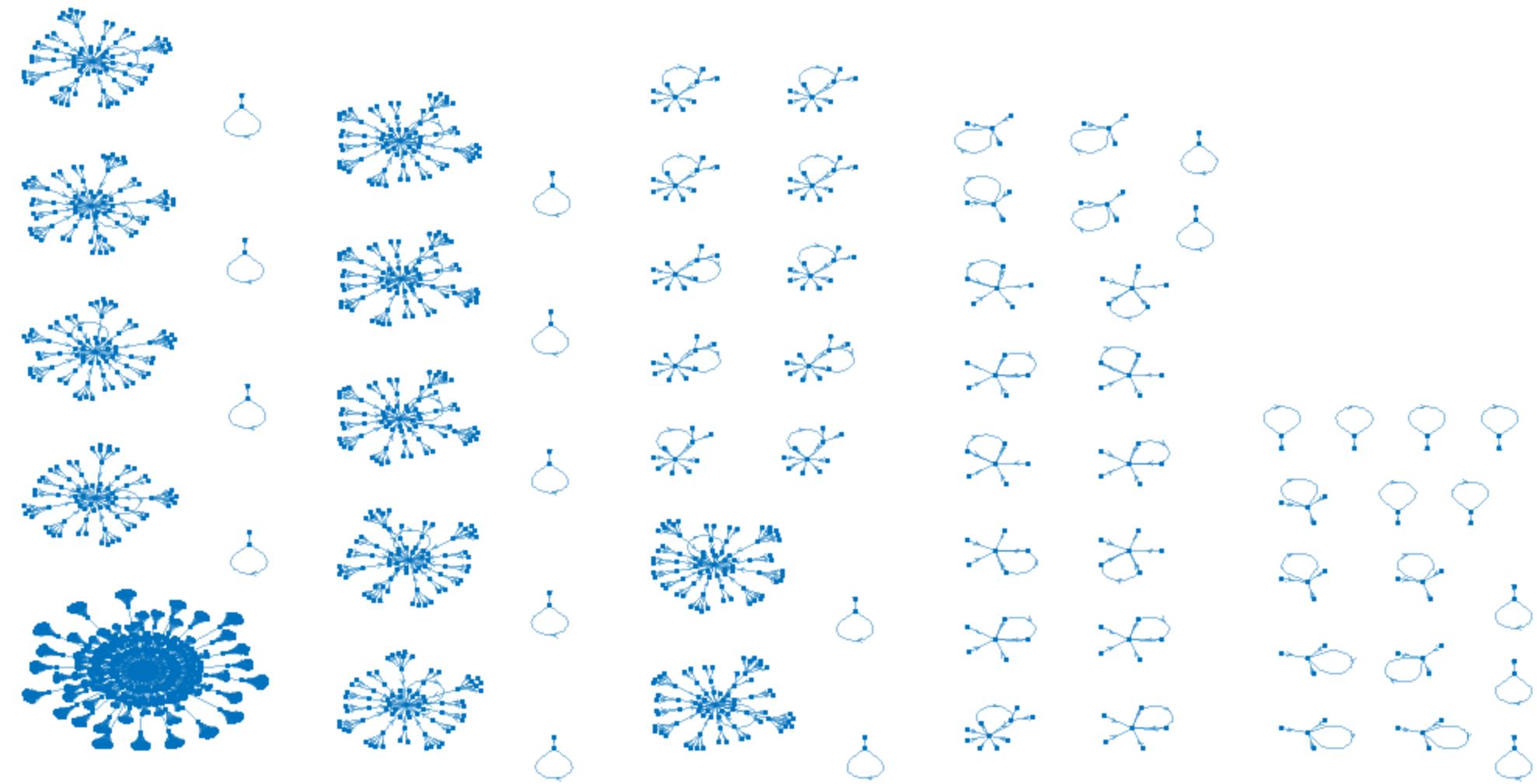


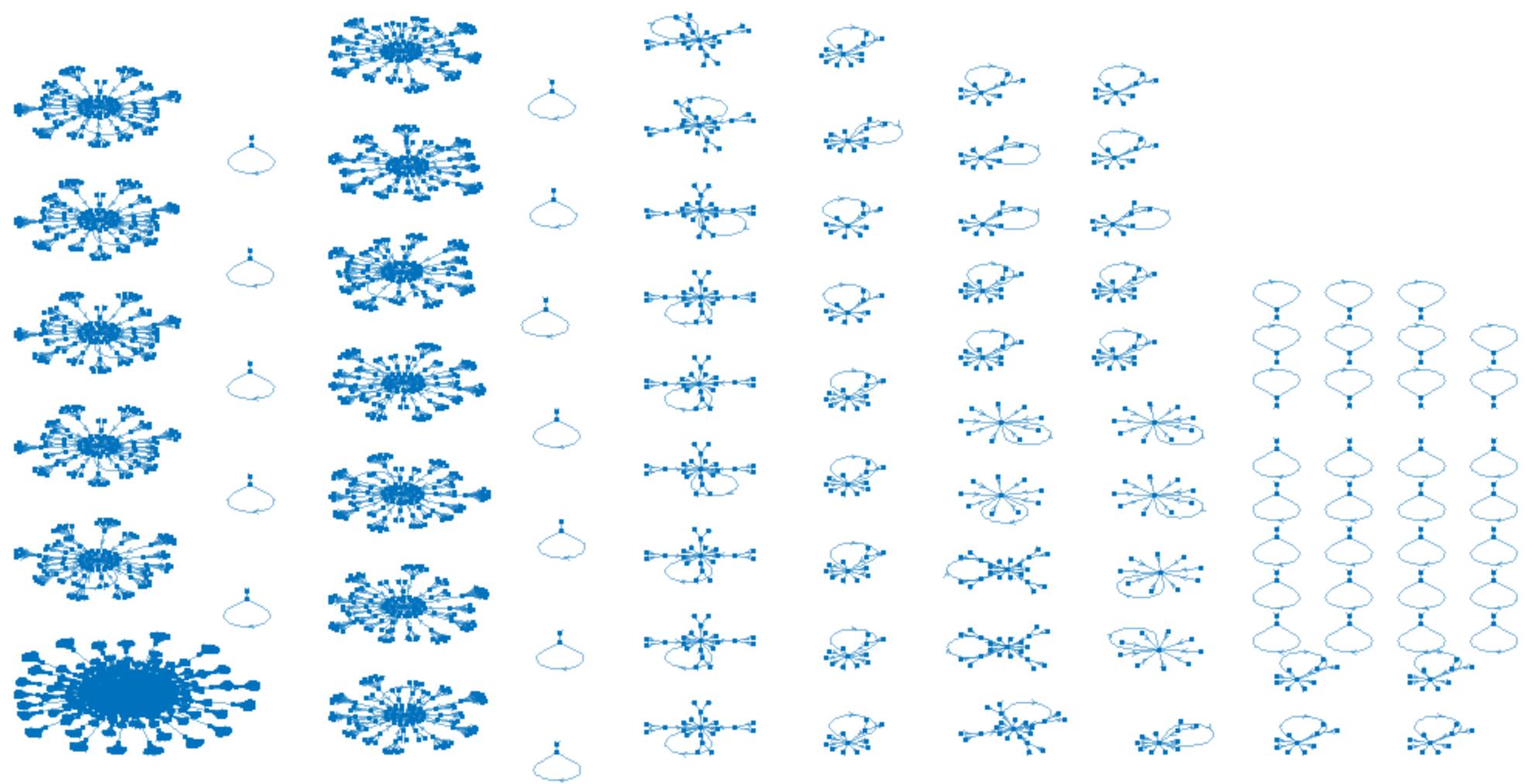


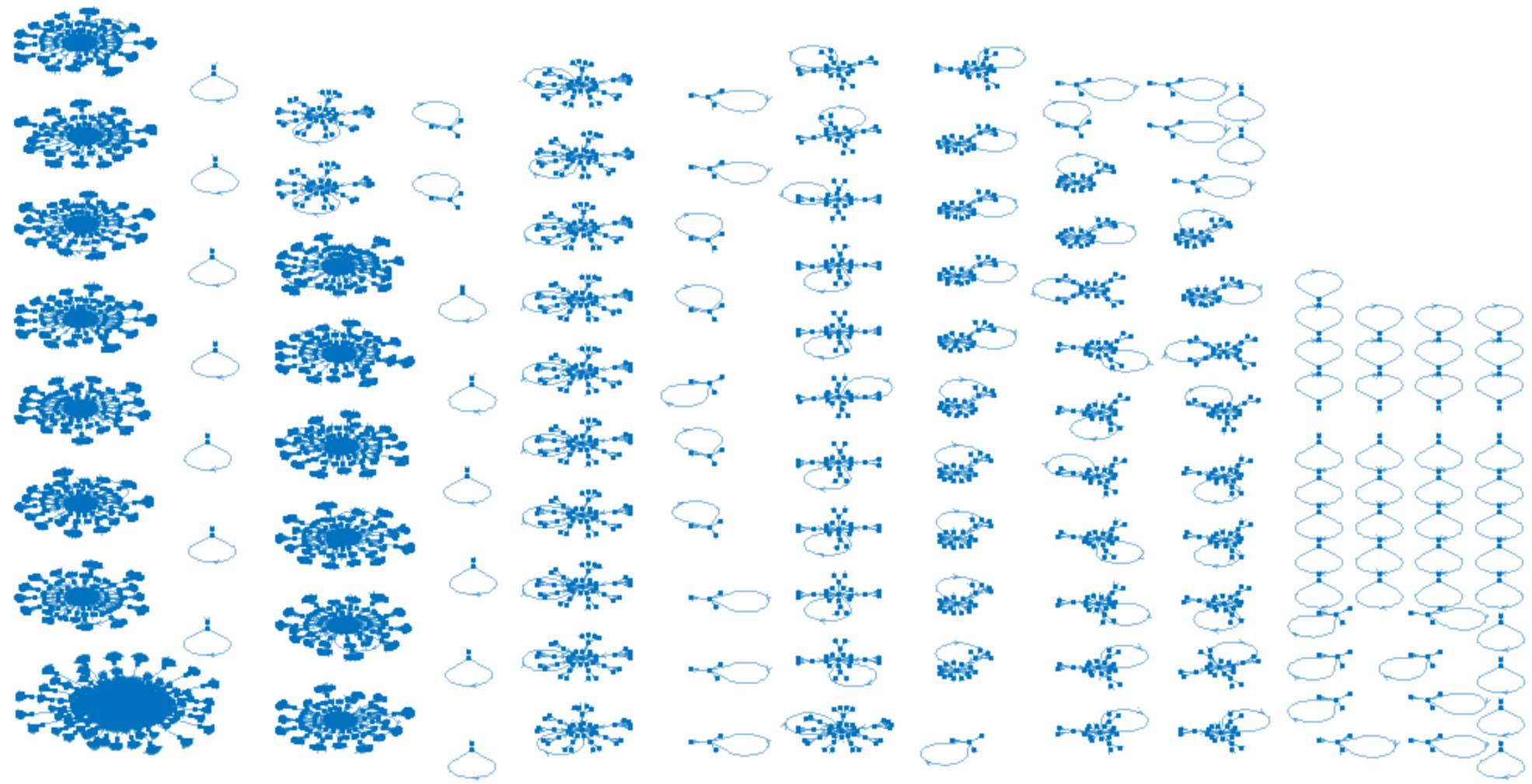


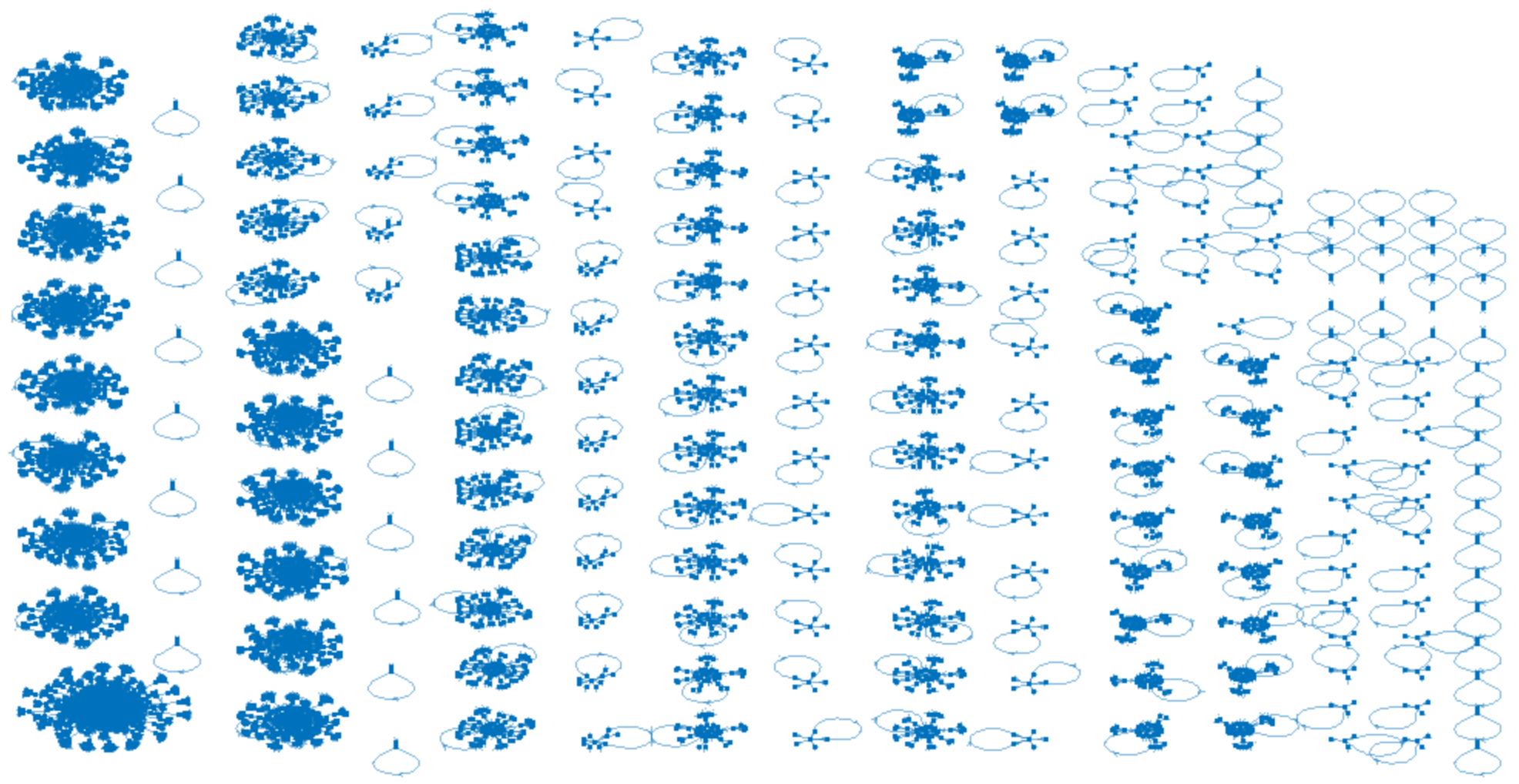


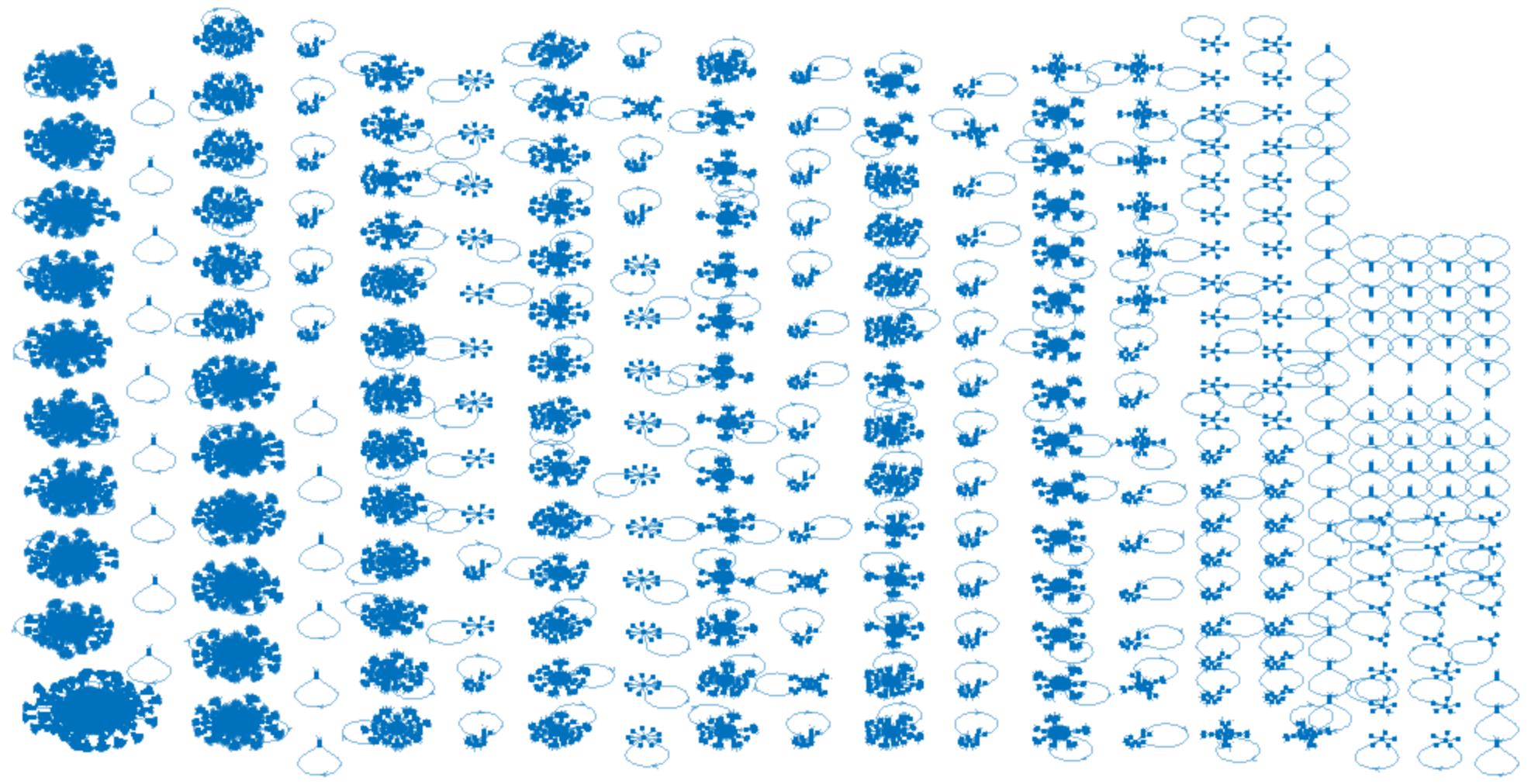


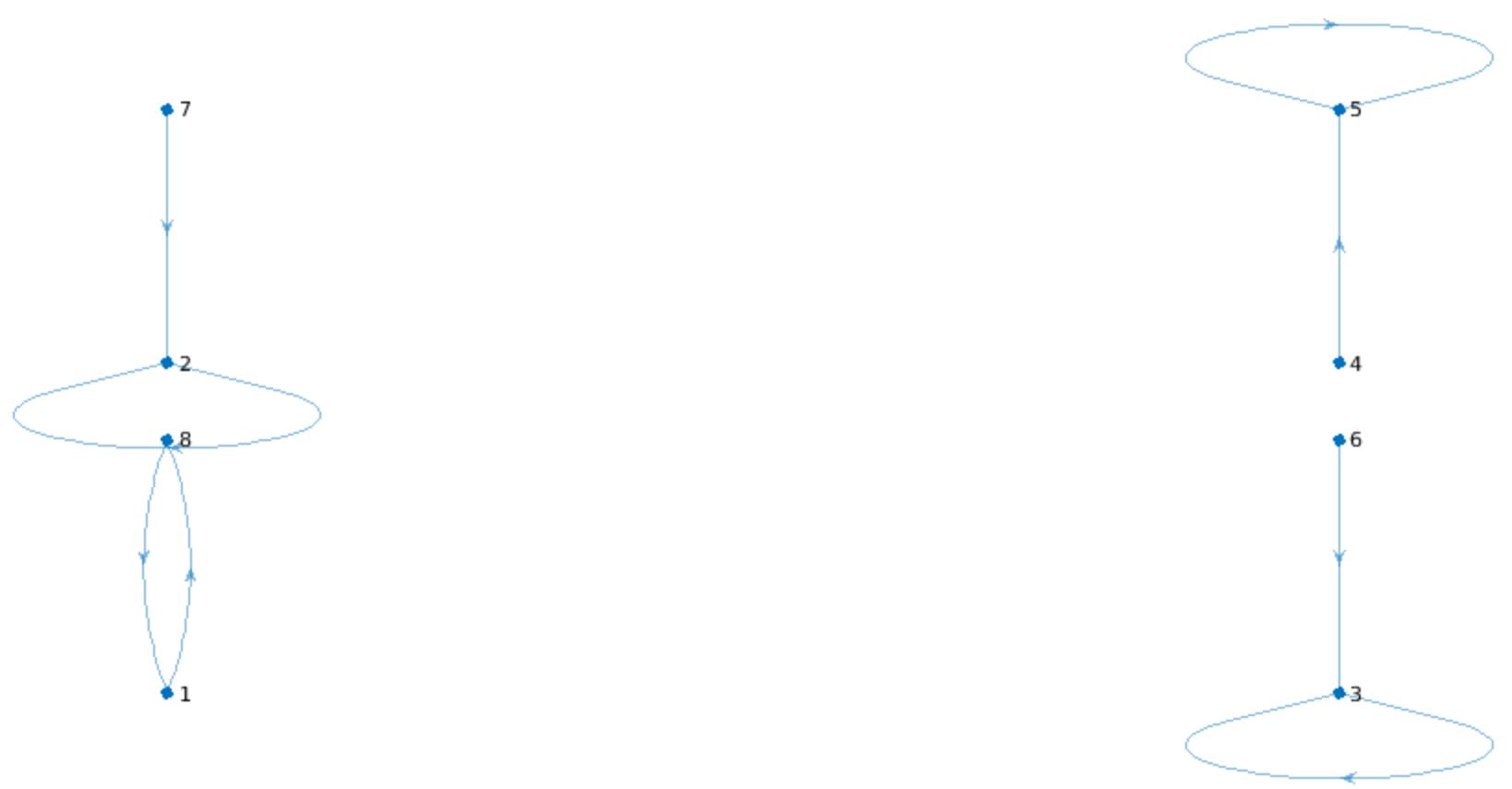
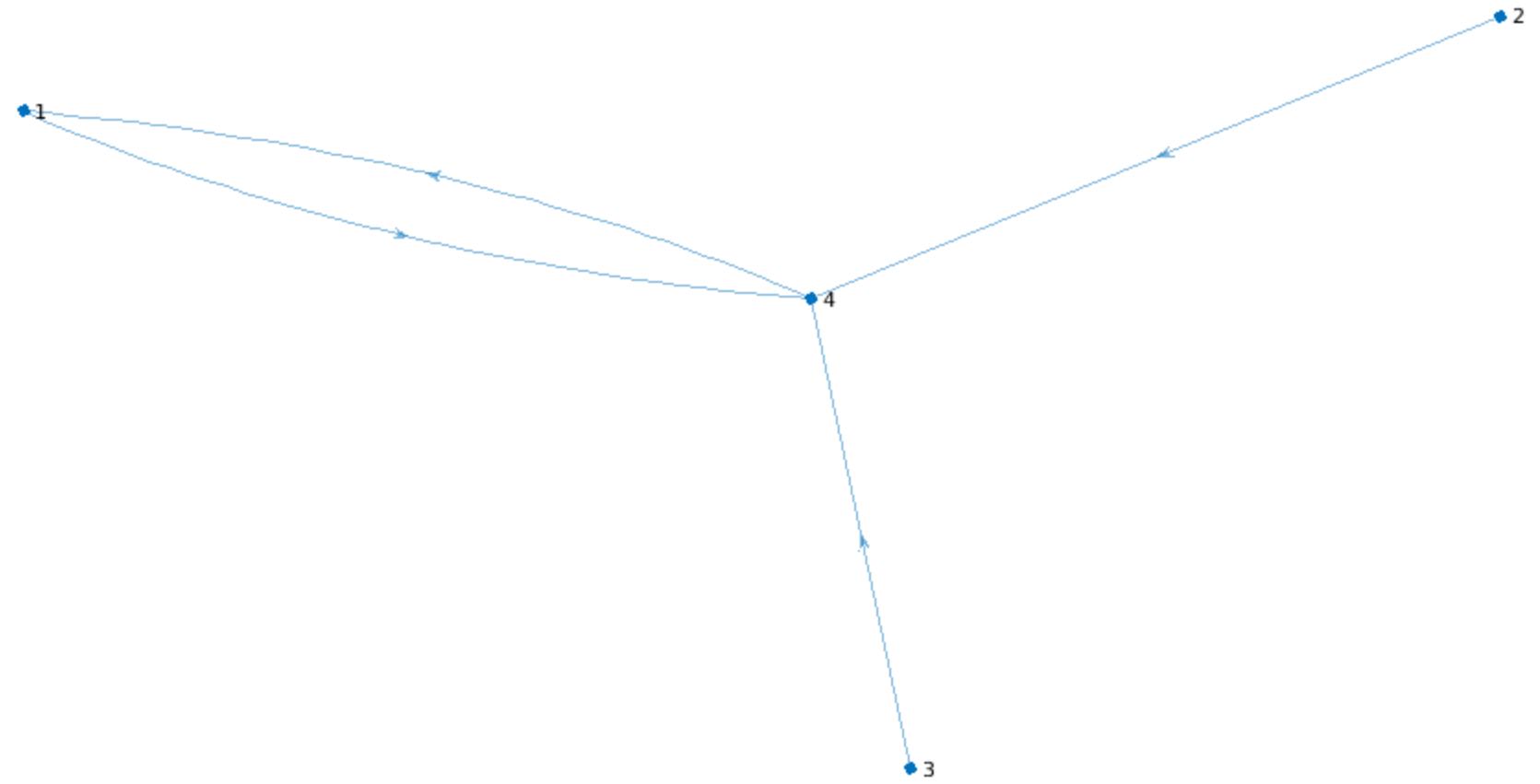


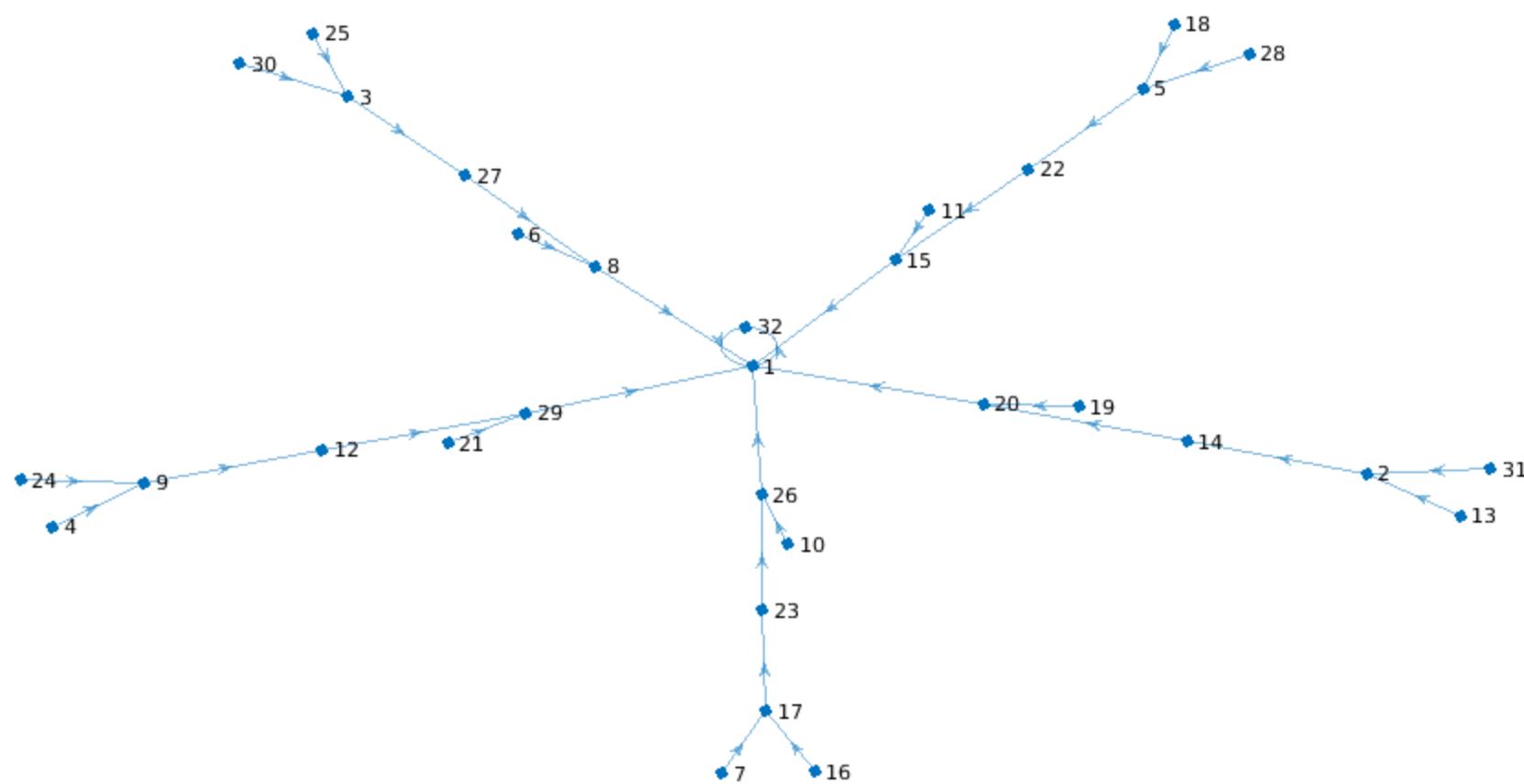
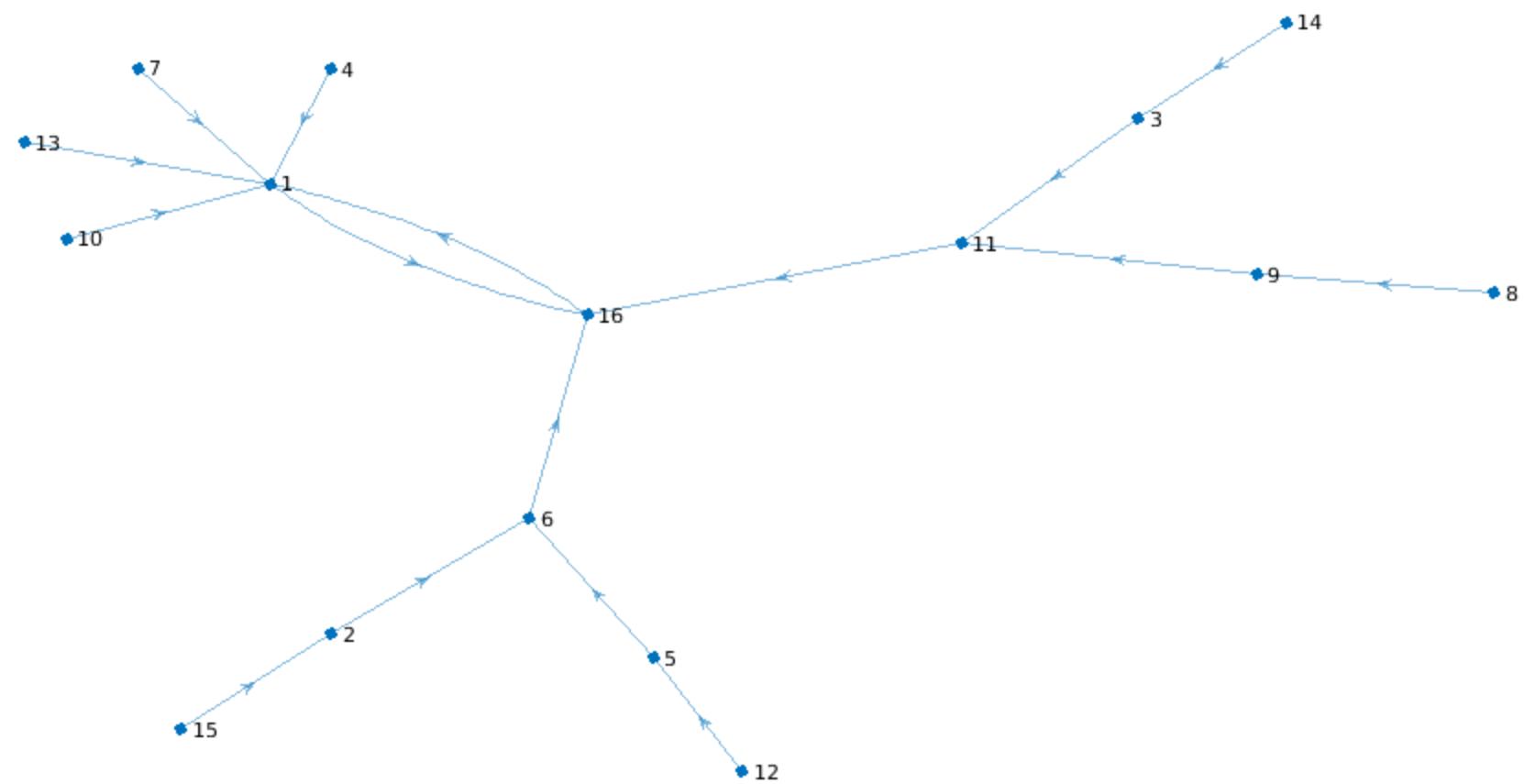


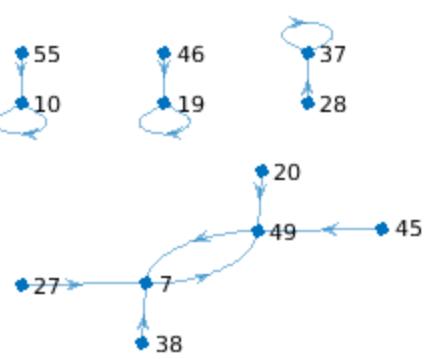
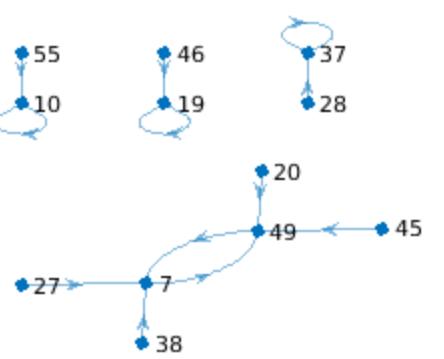
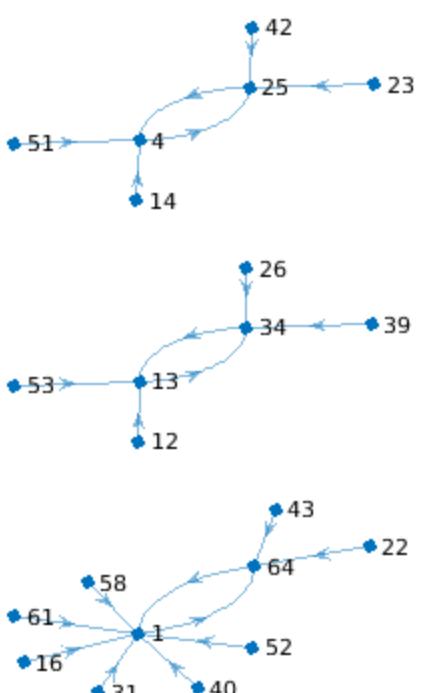
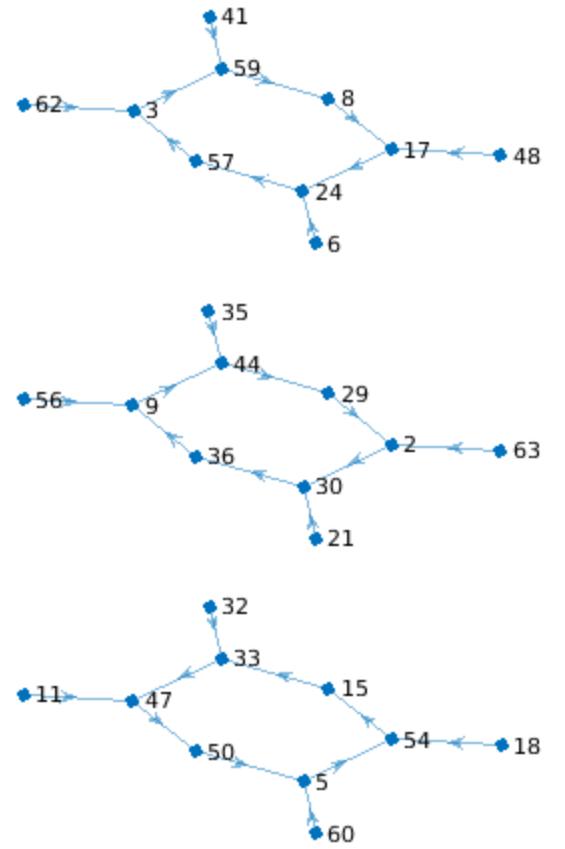


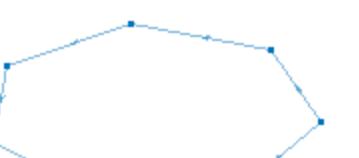
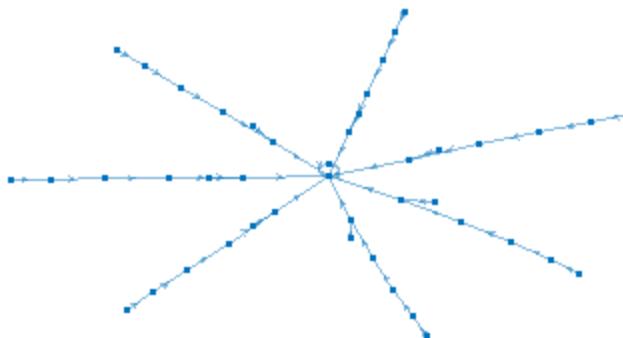


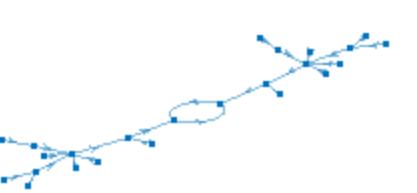
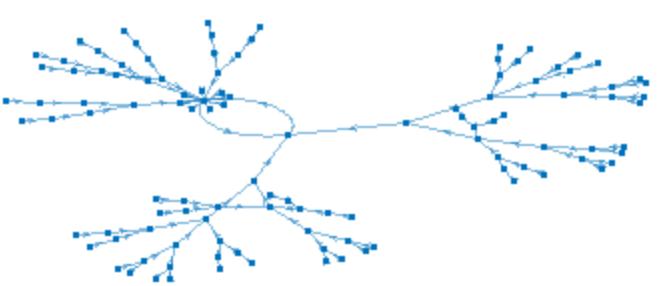
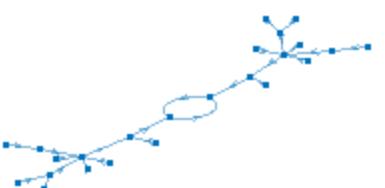
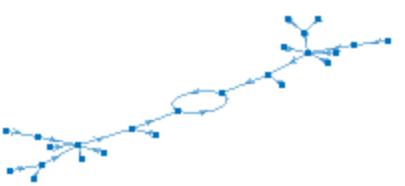


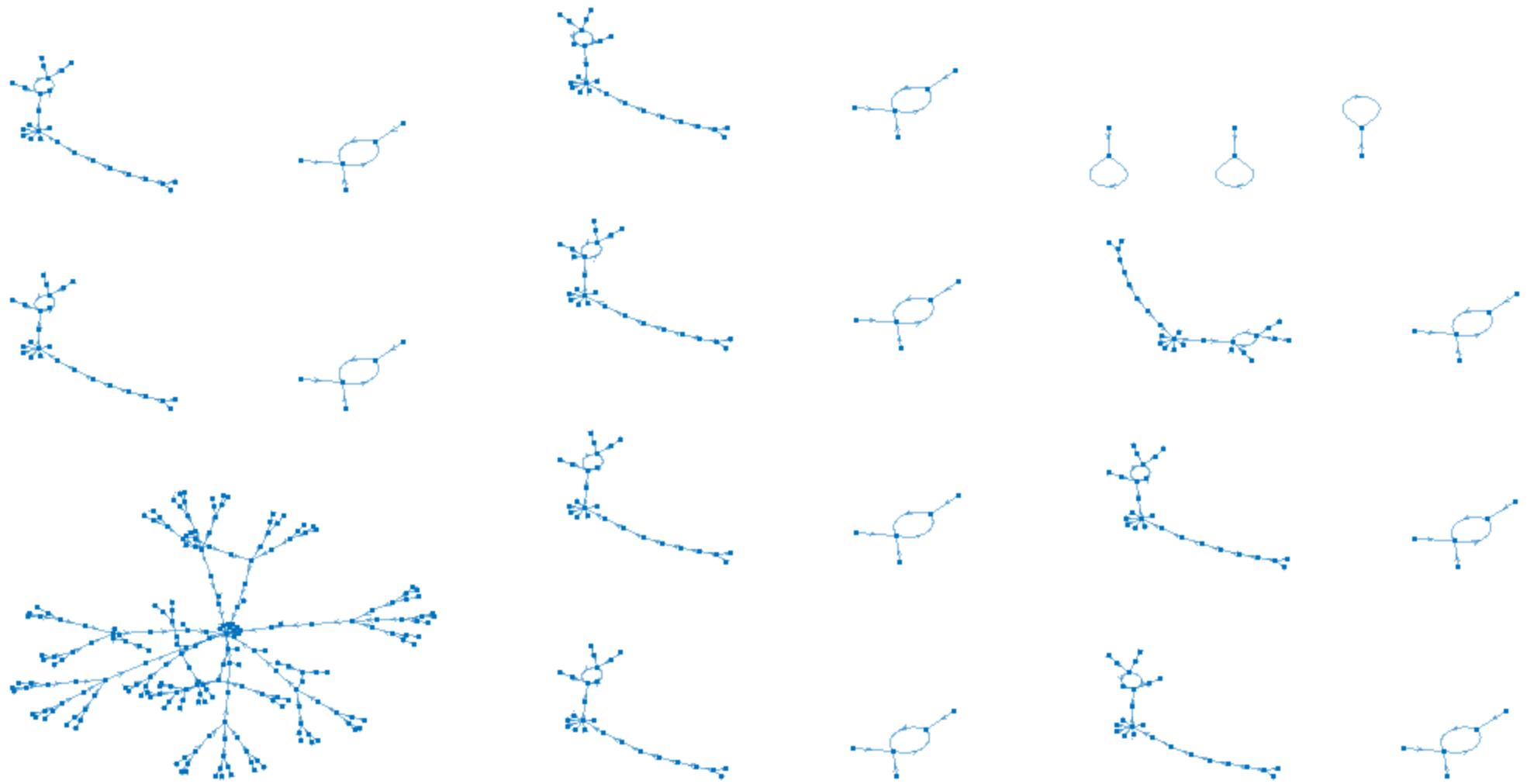


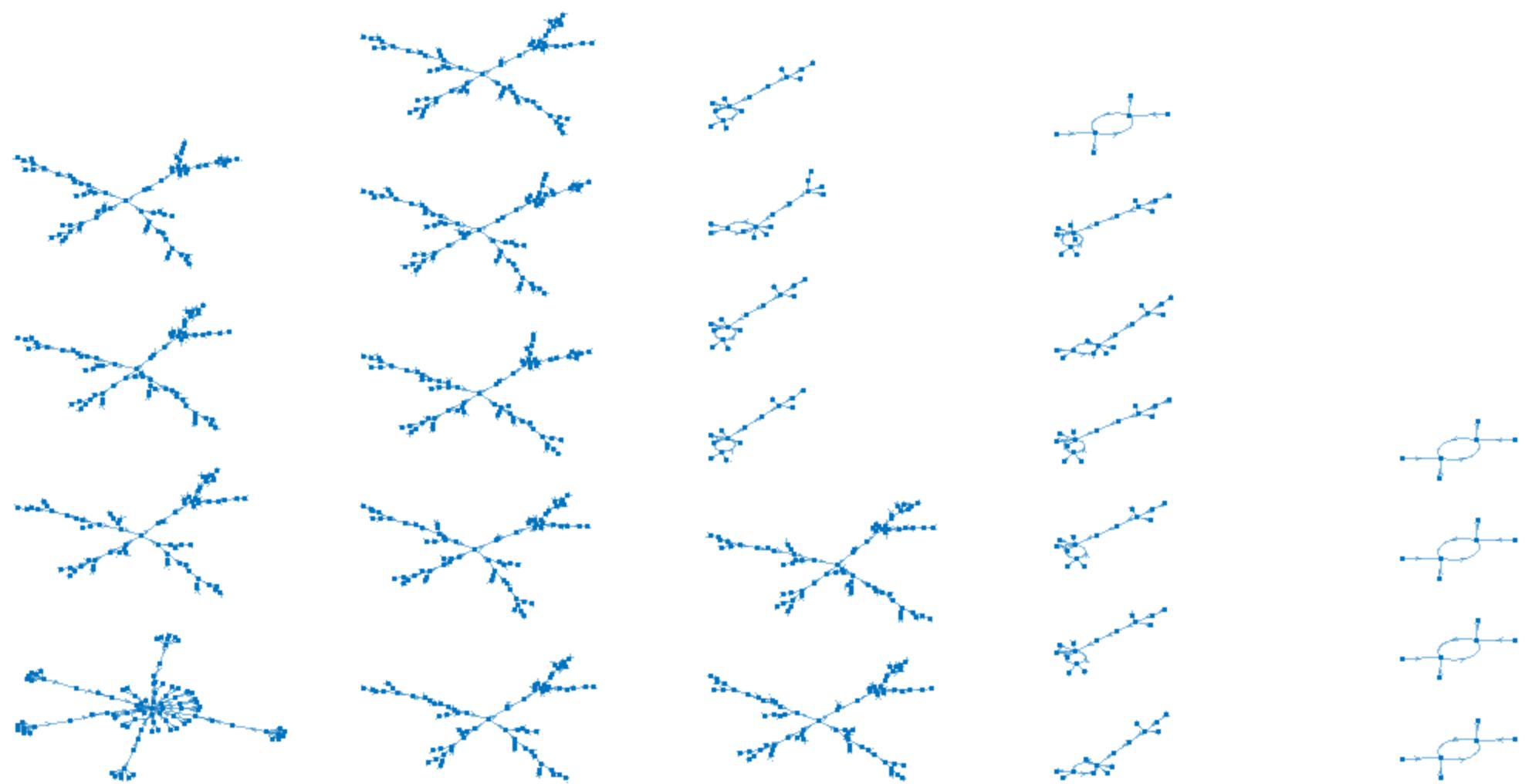


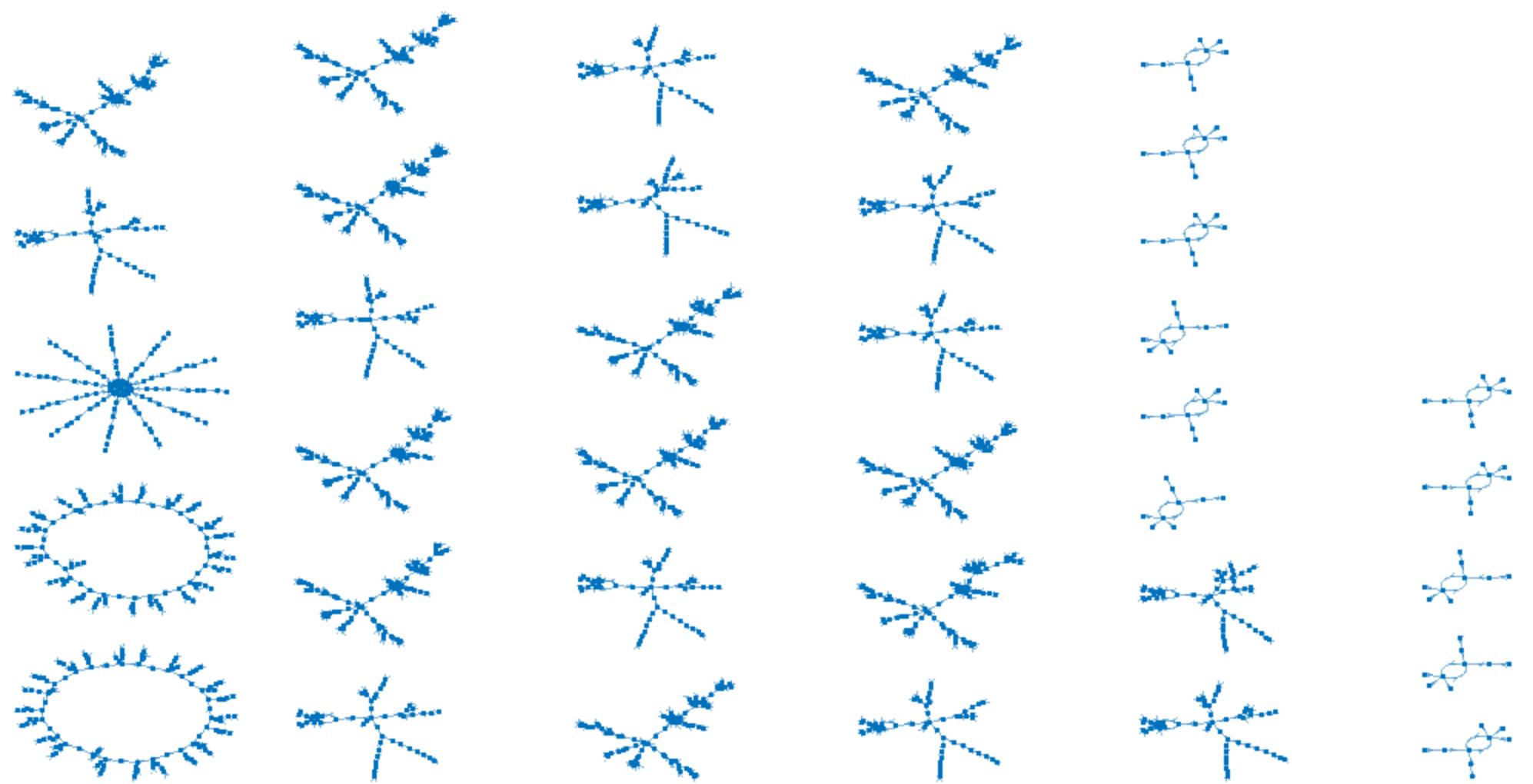


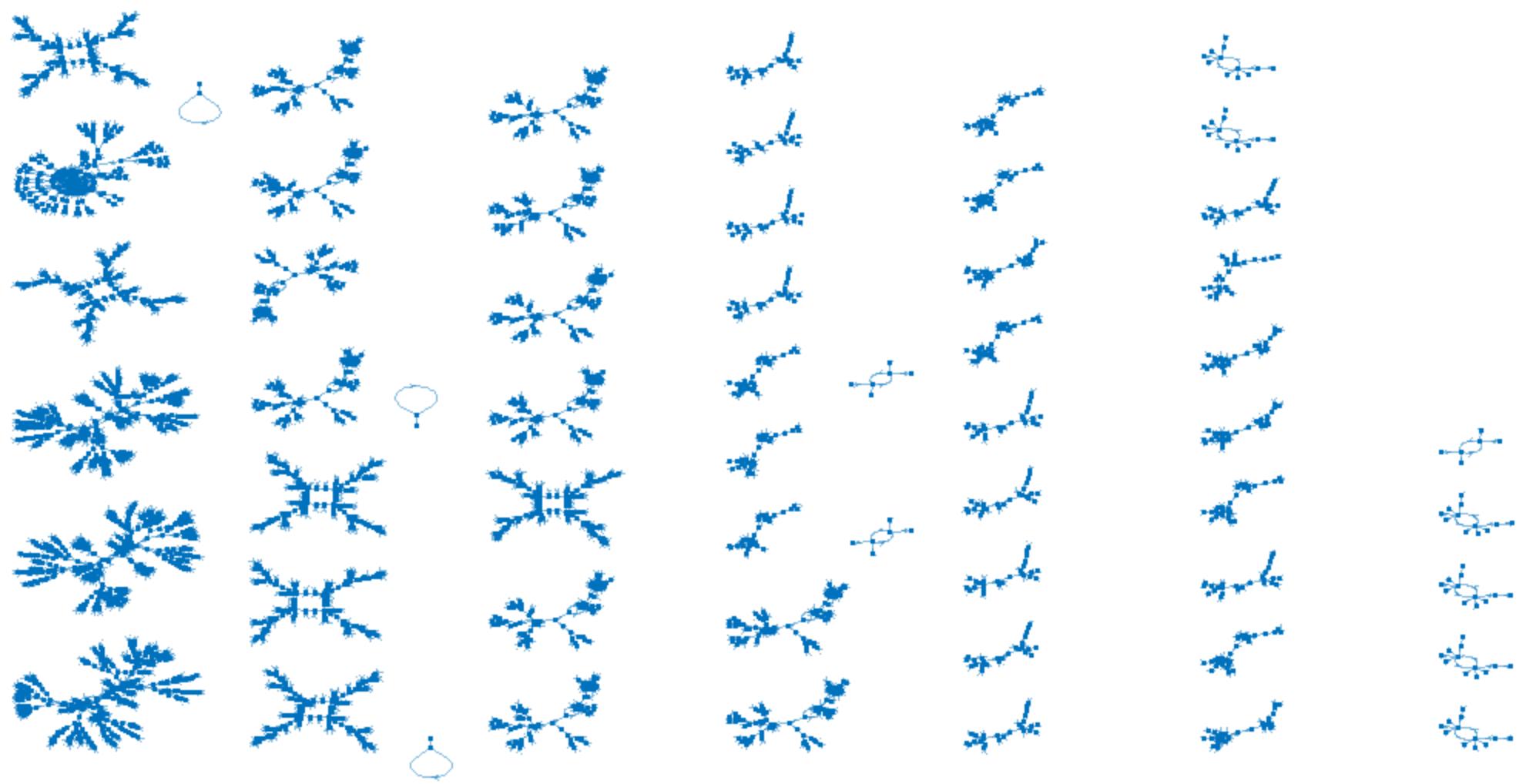


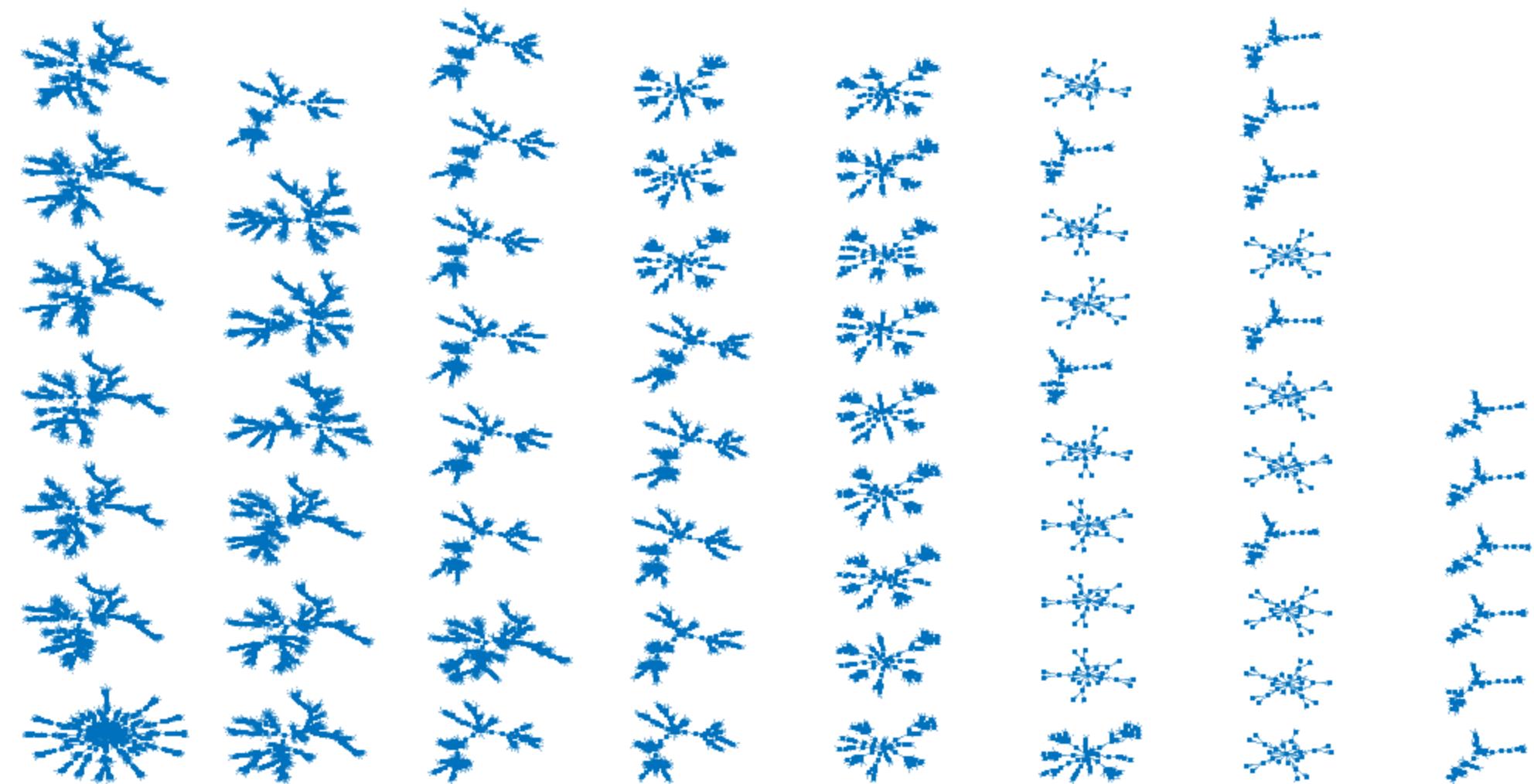


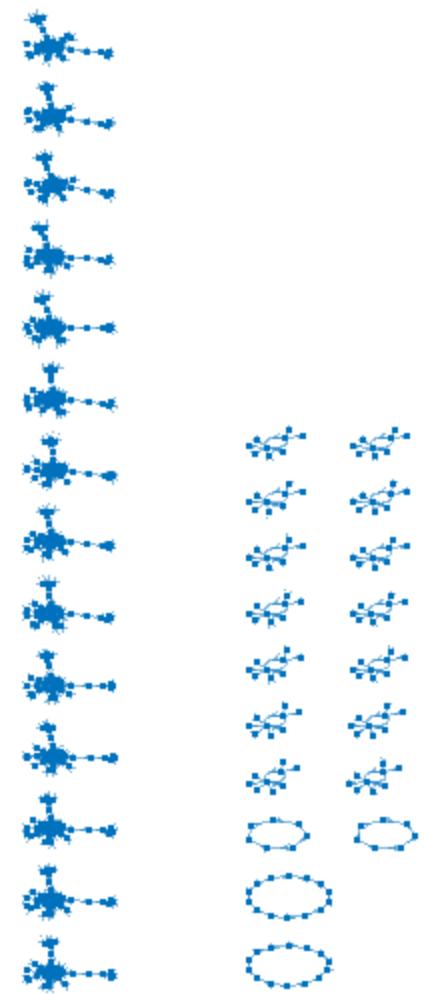
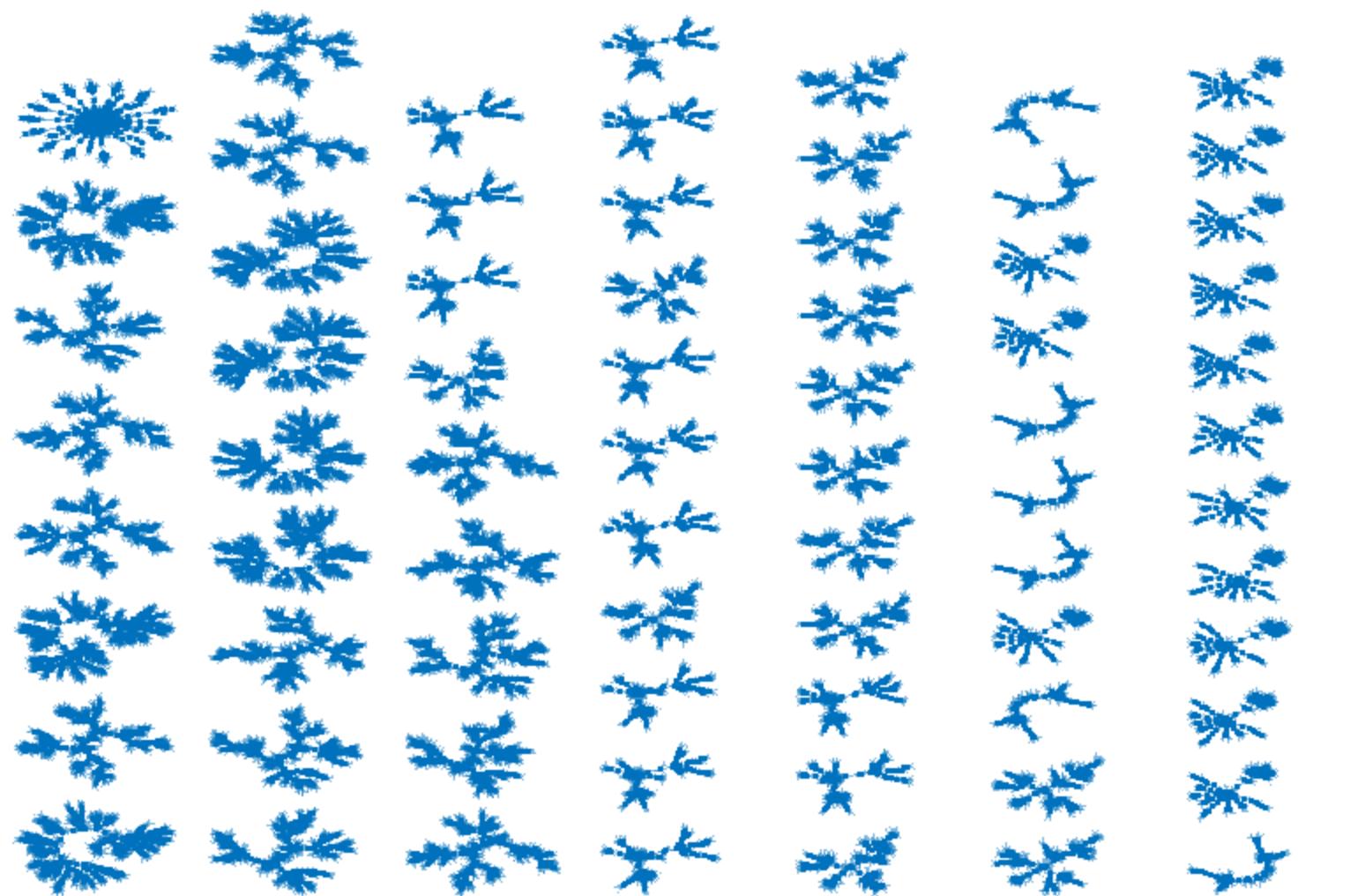




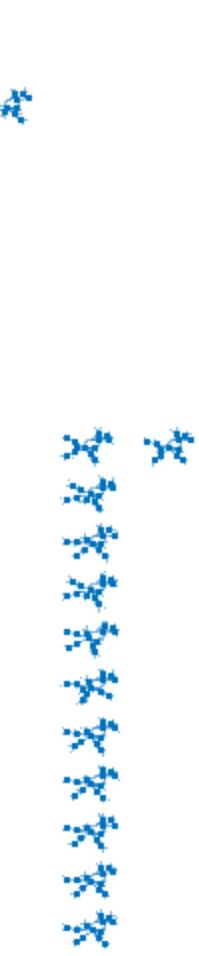
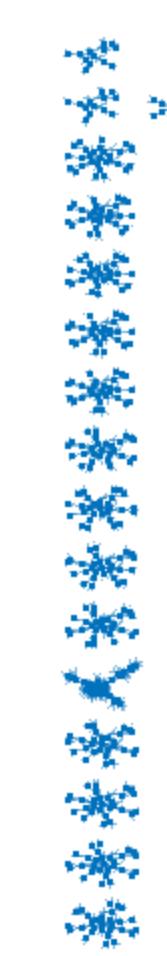
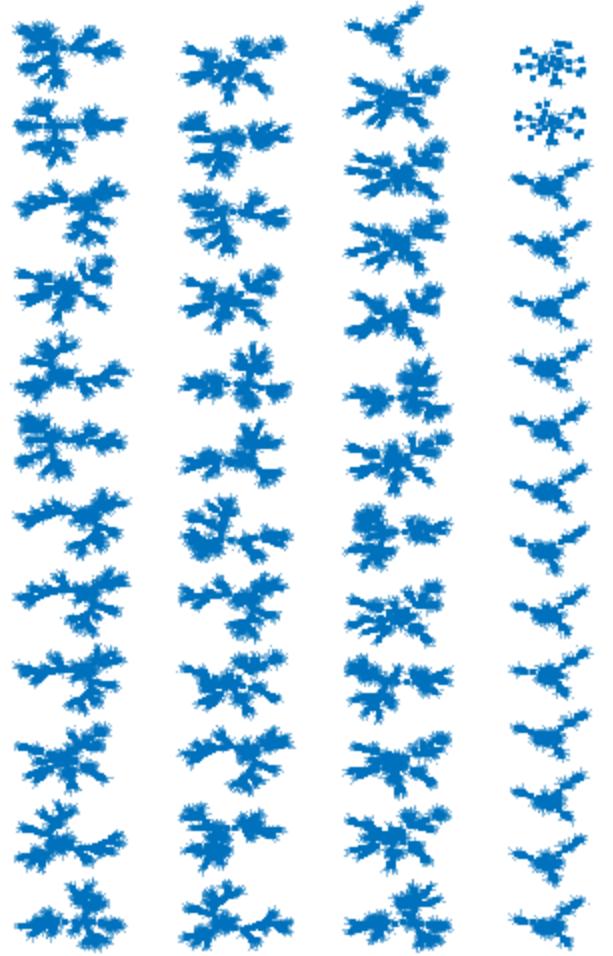
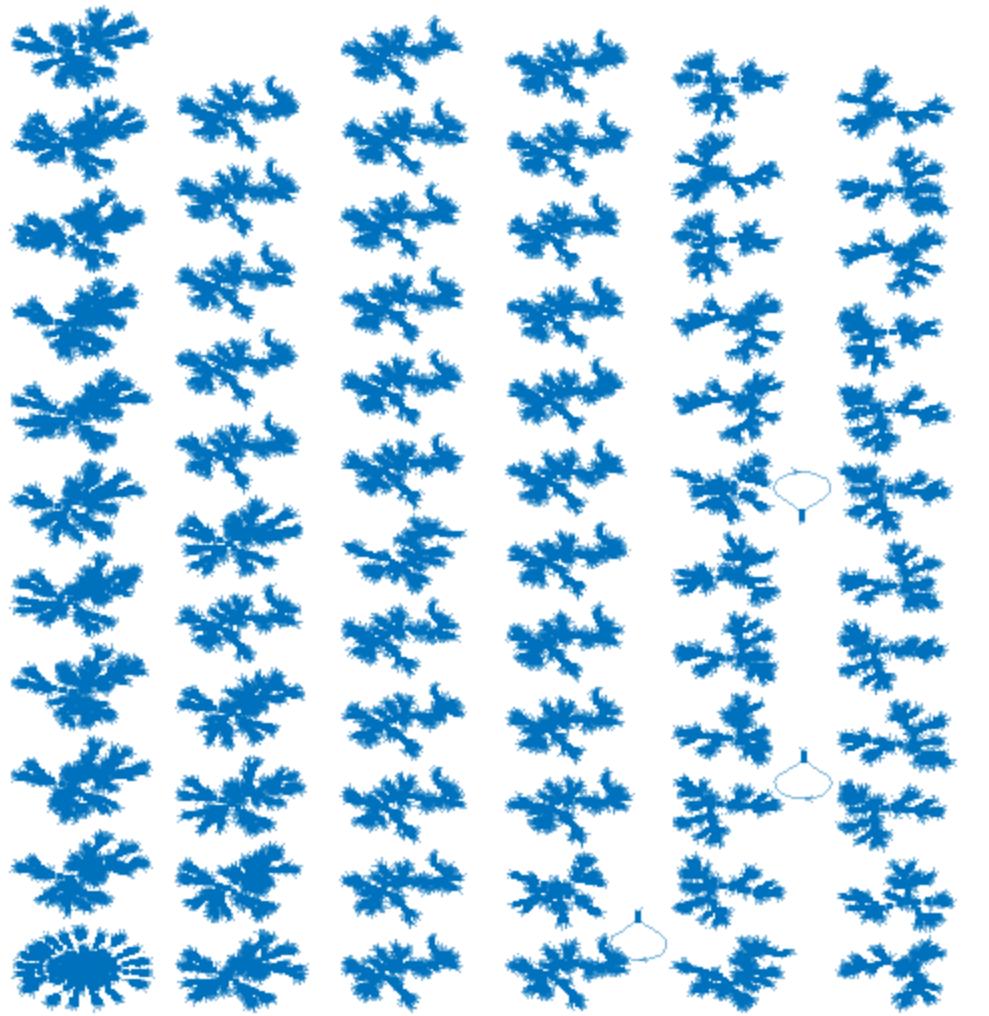






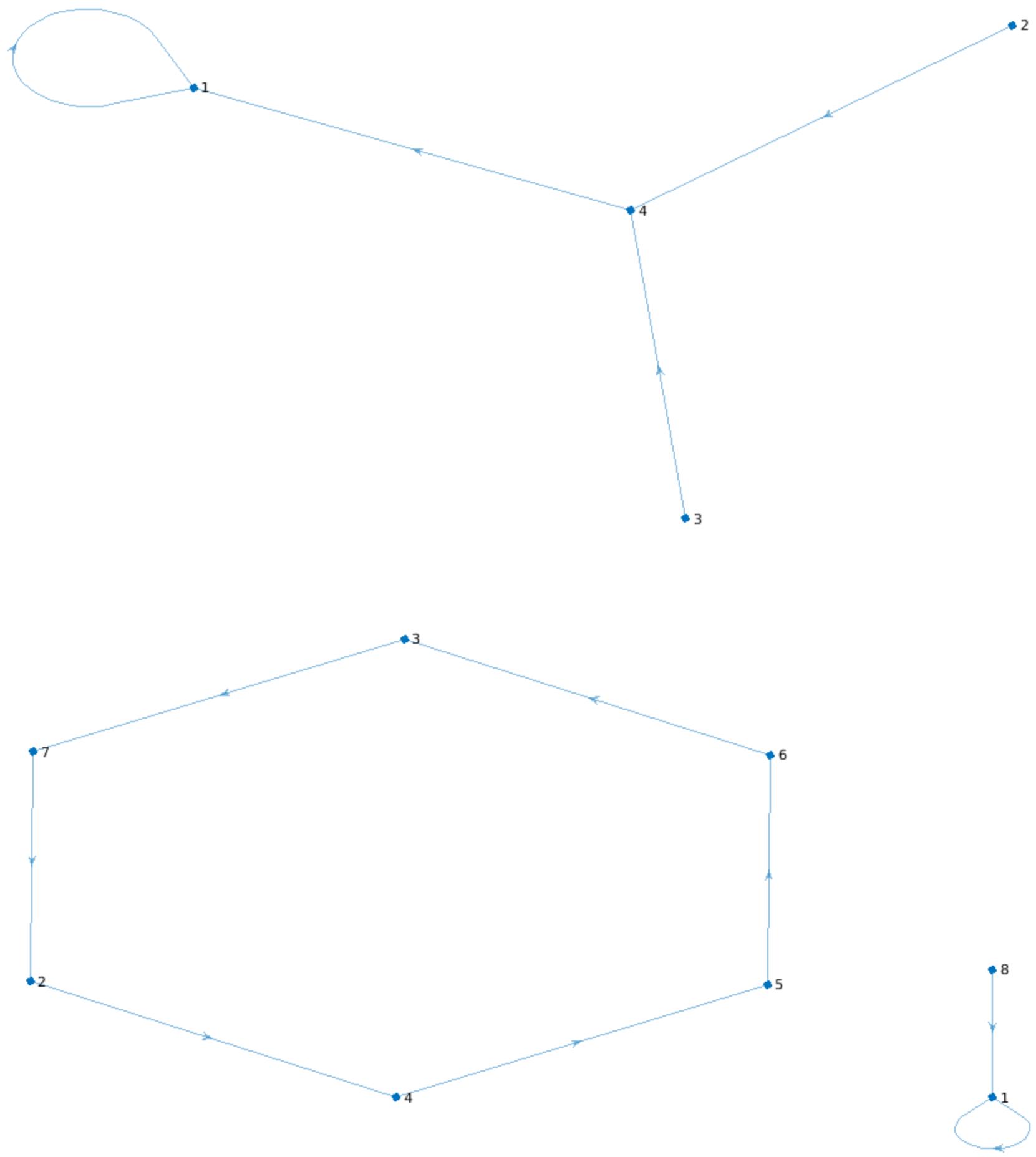


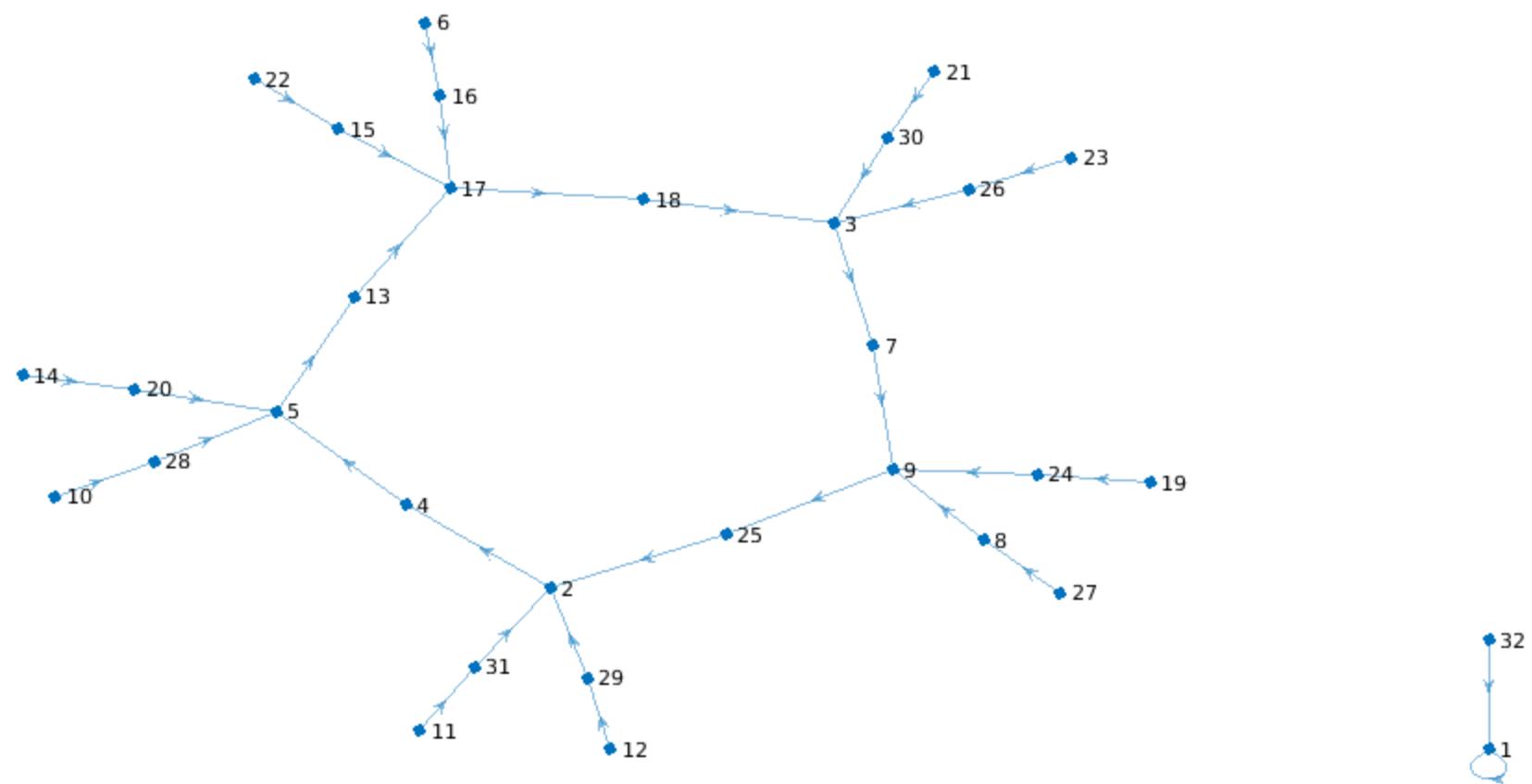
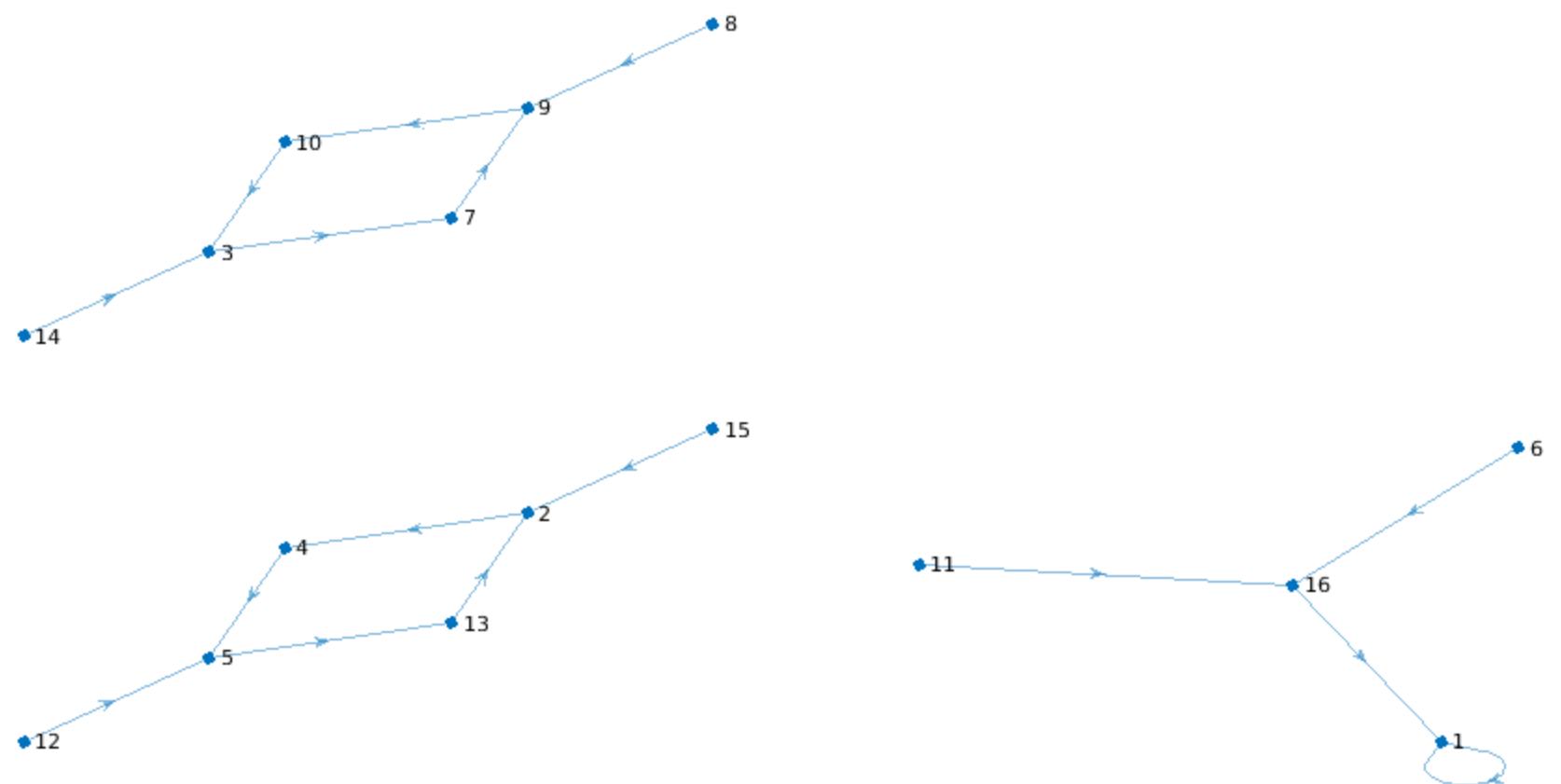
○○○○○○○○○○○○

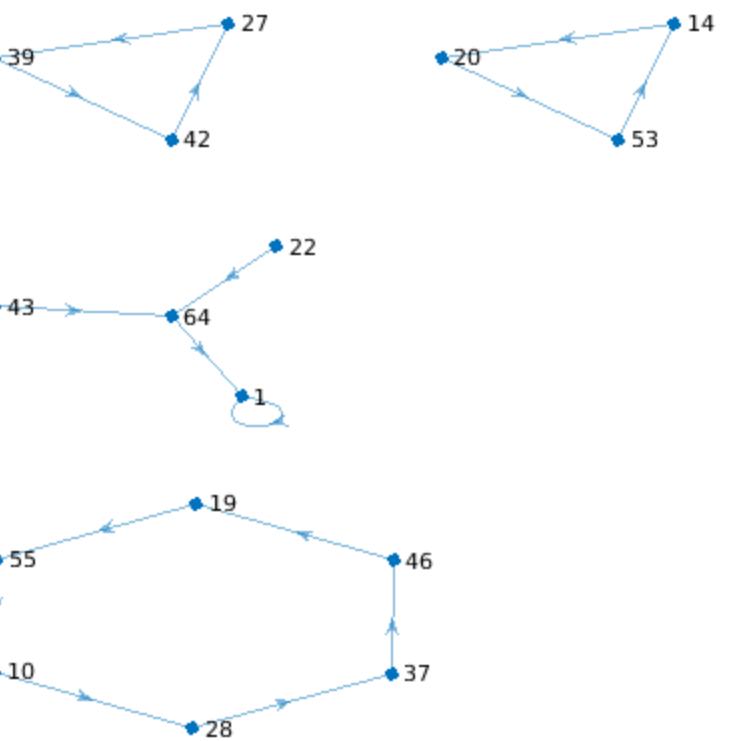
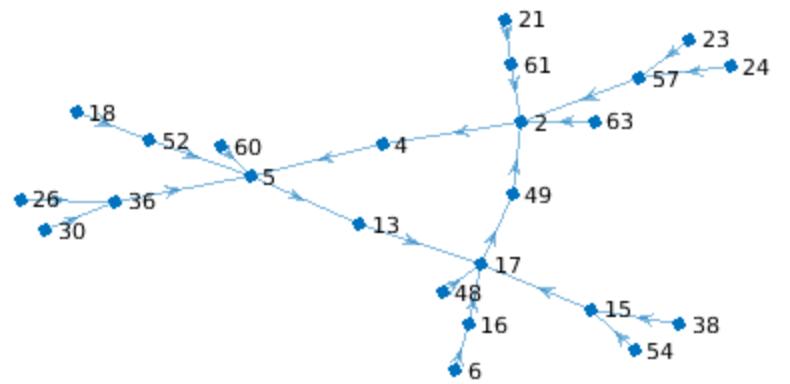


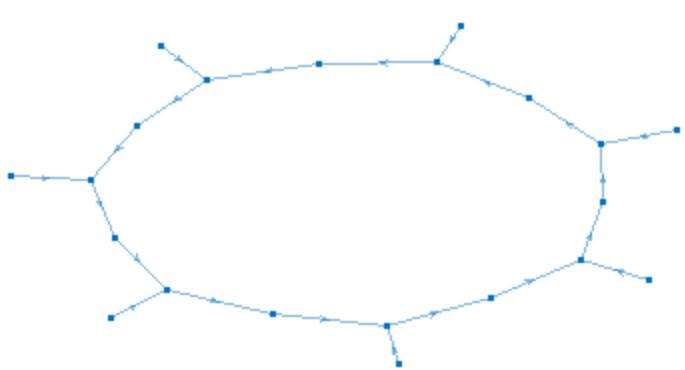
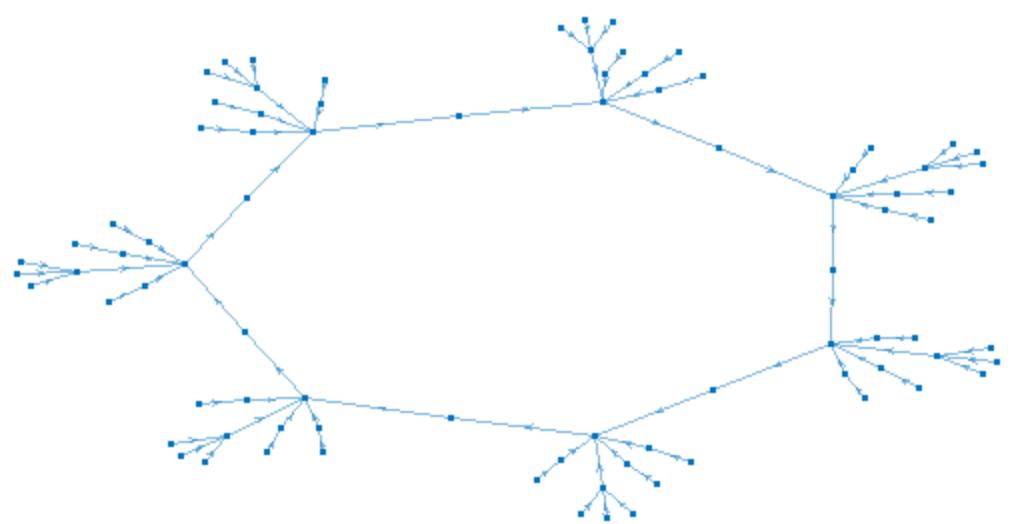
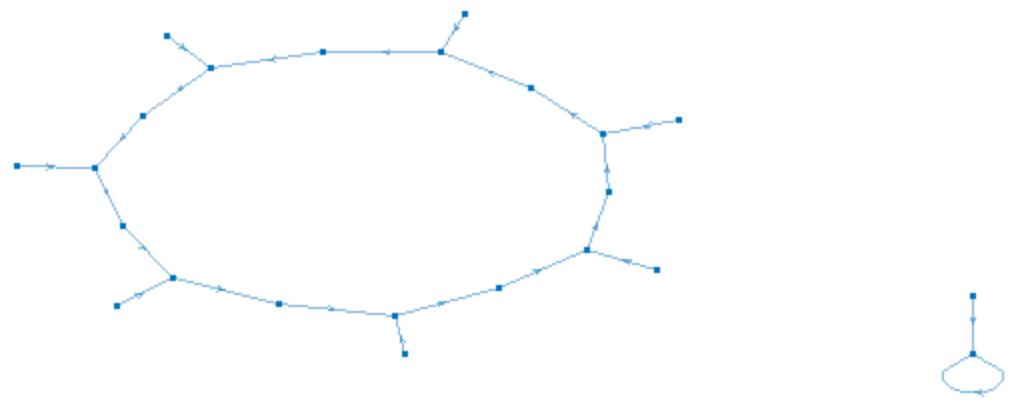
新郎新郎新郎新郎新郎新郎

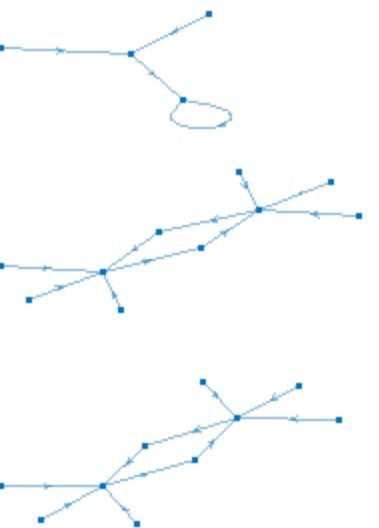
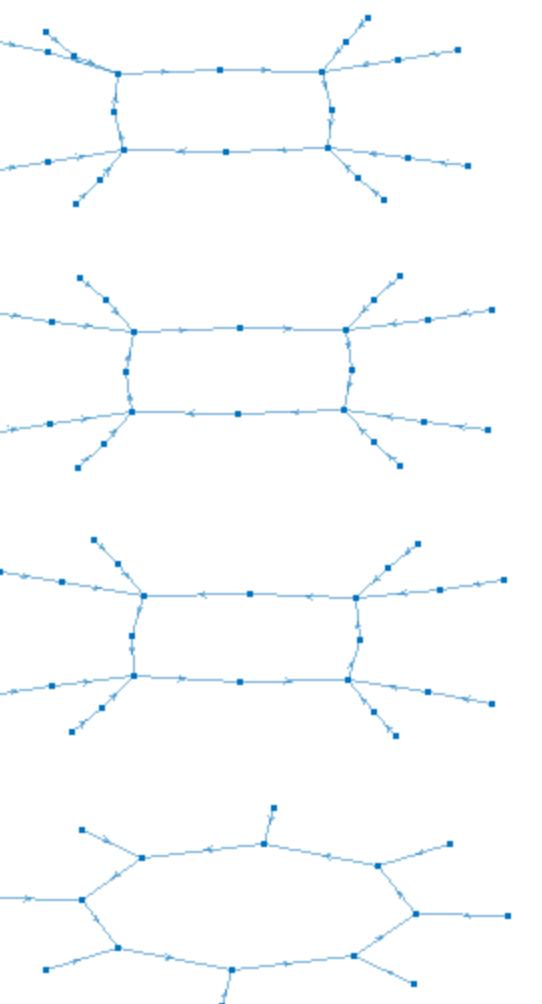
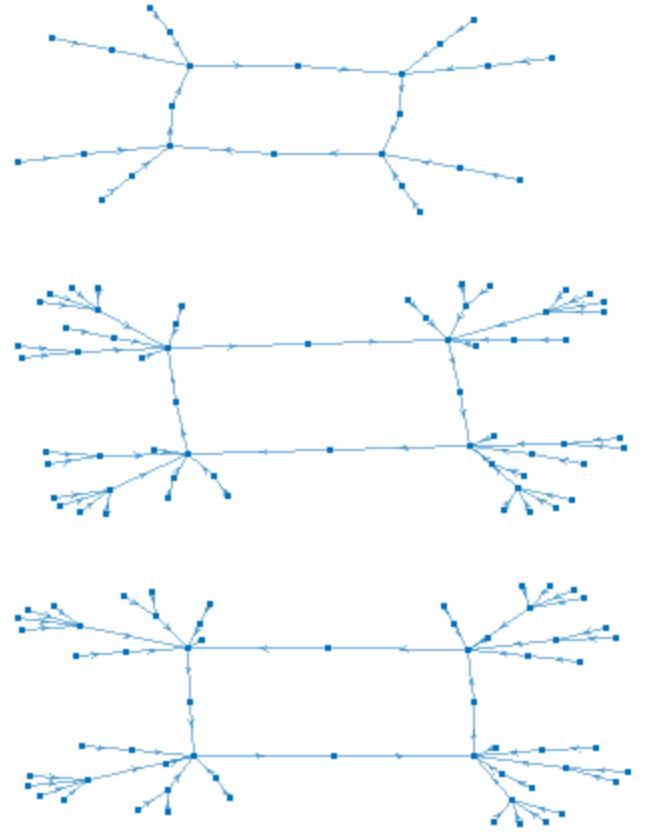
新郎

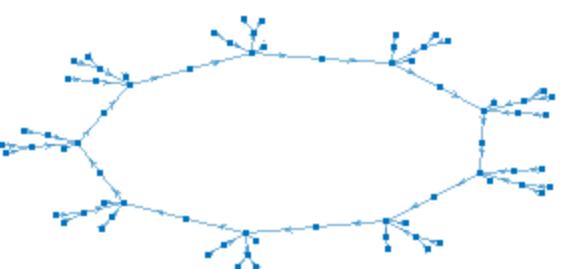
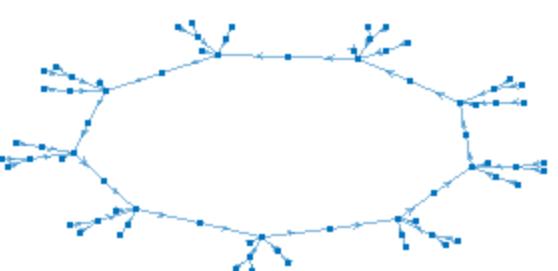
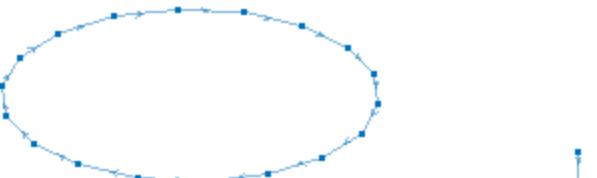
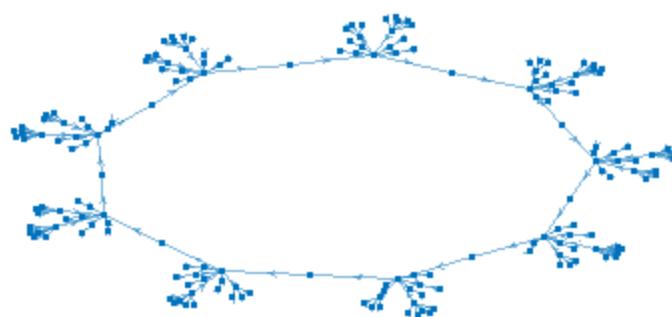
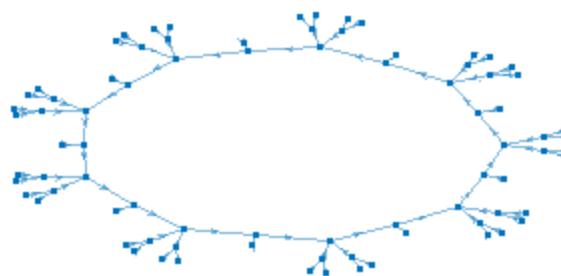
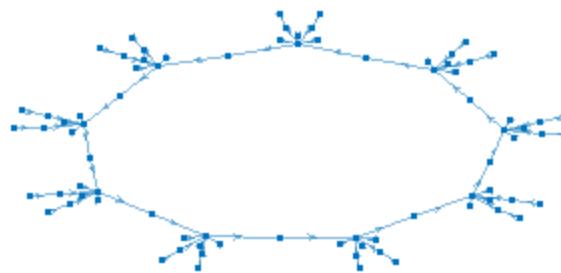


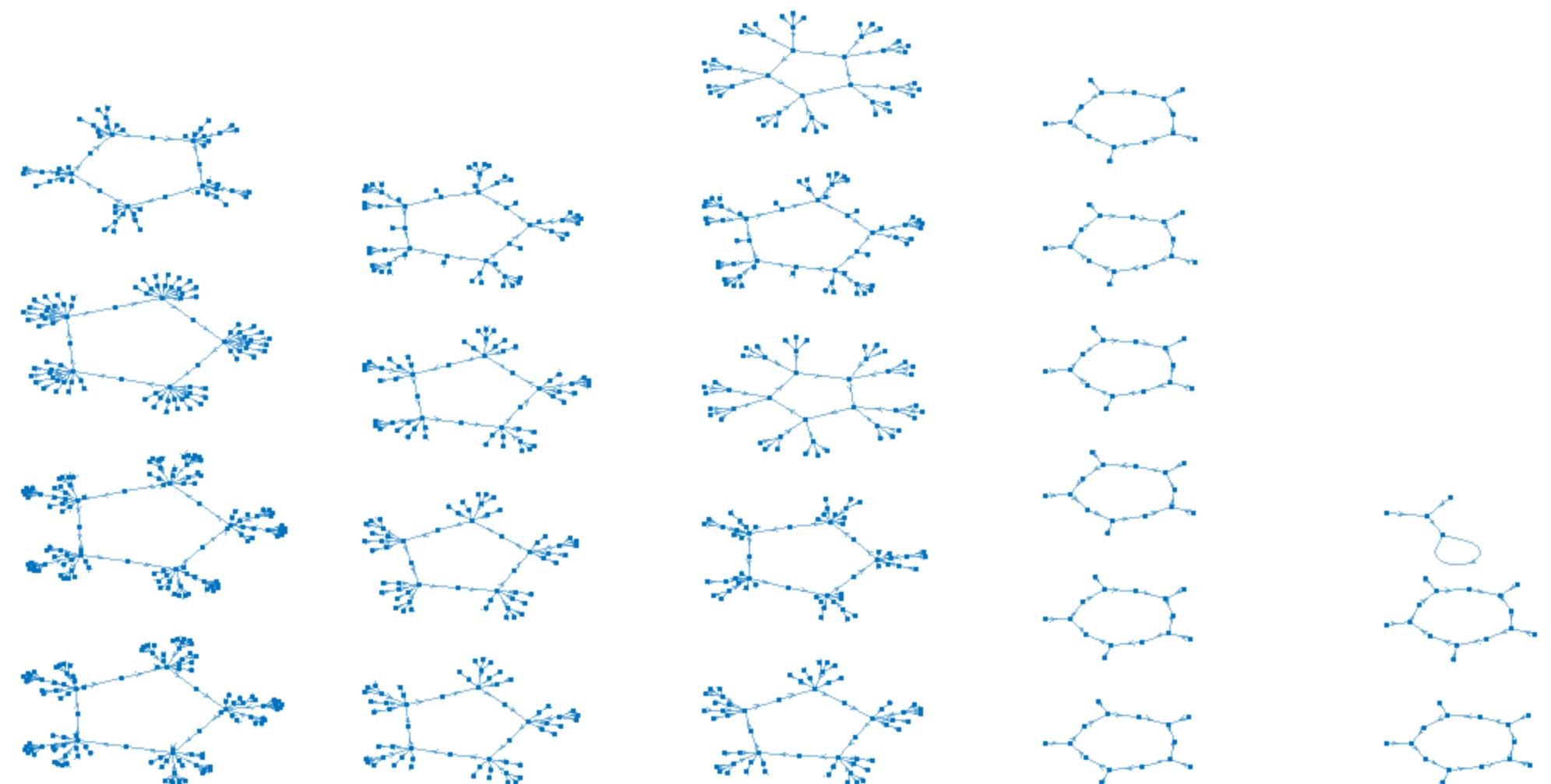


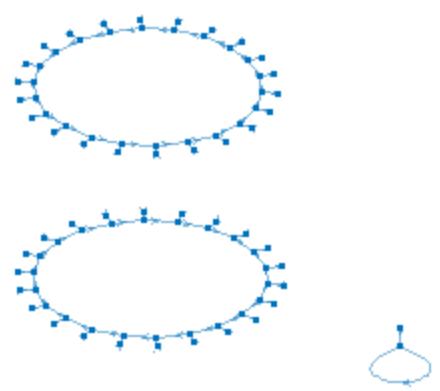
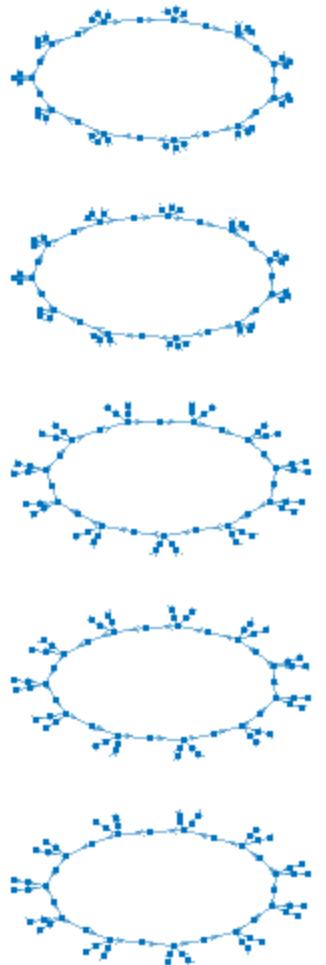
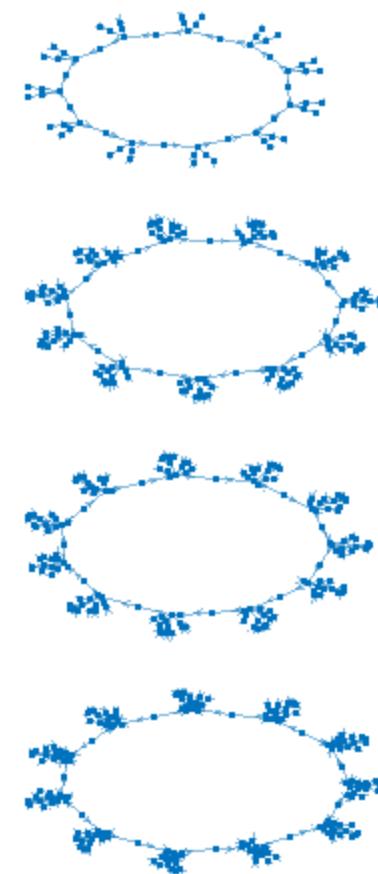
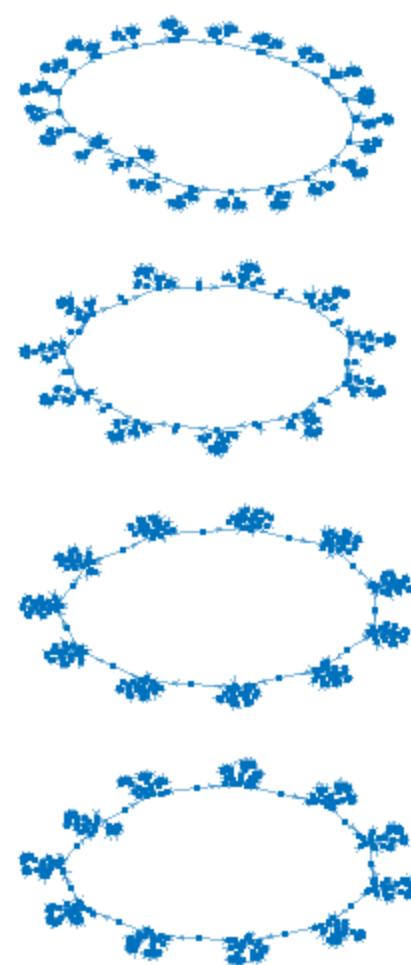


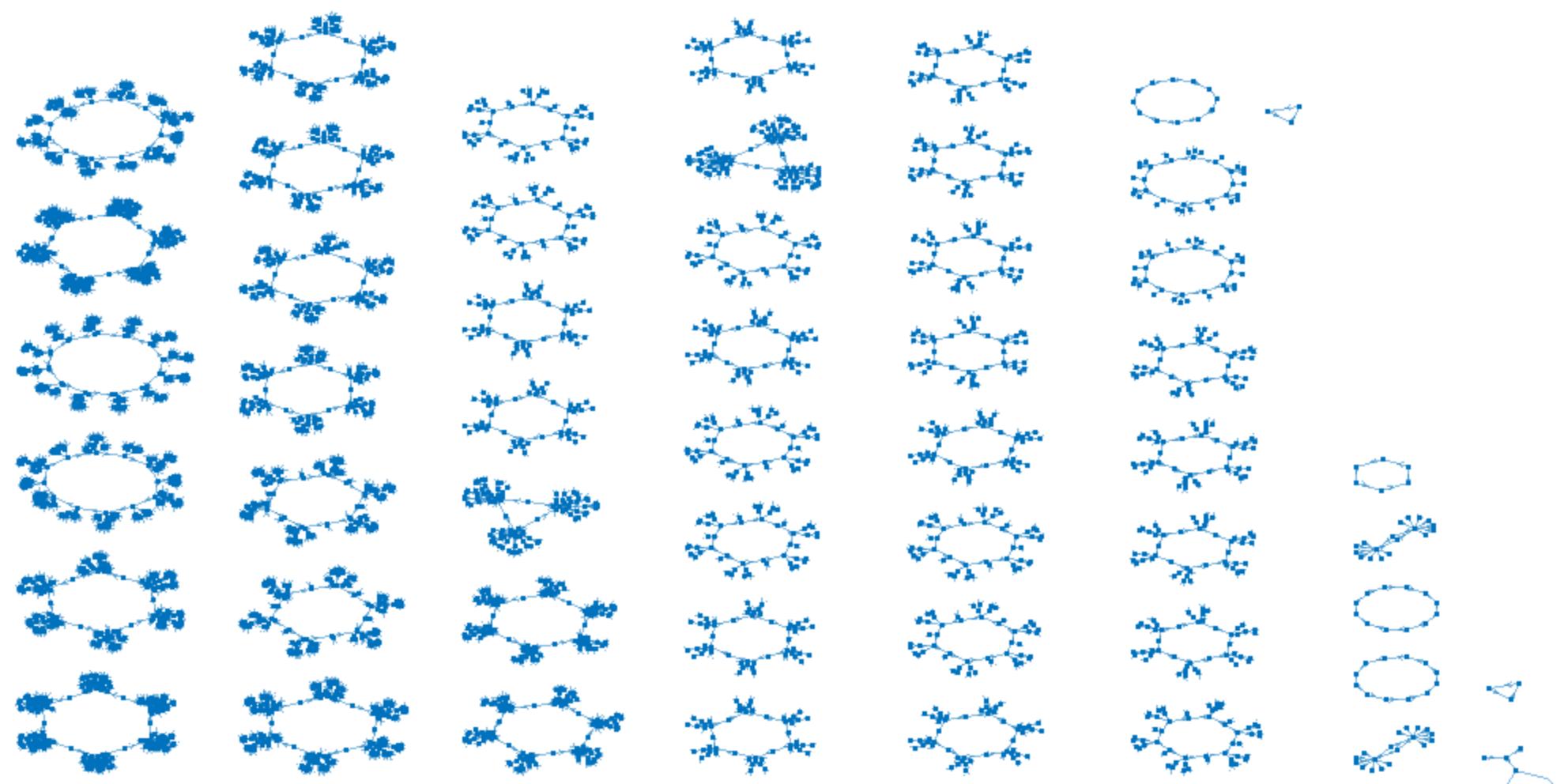


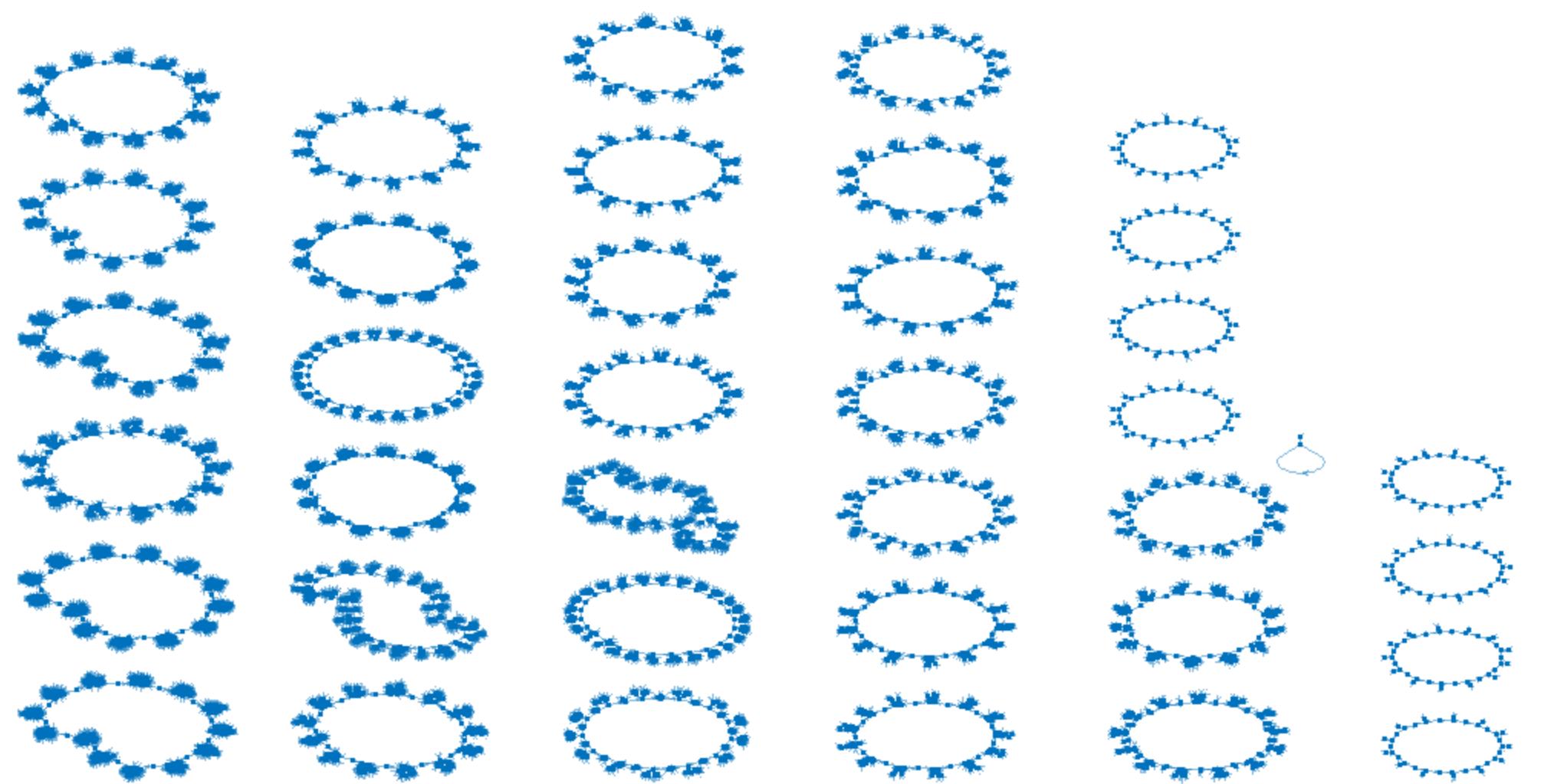


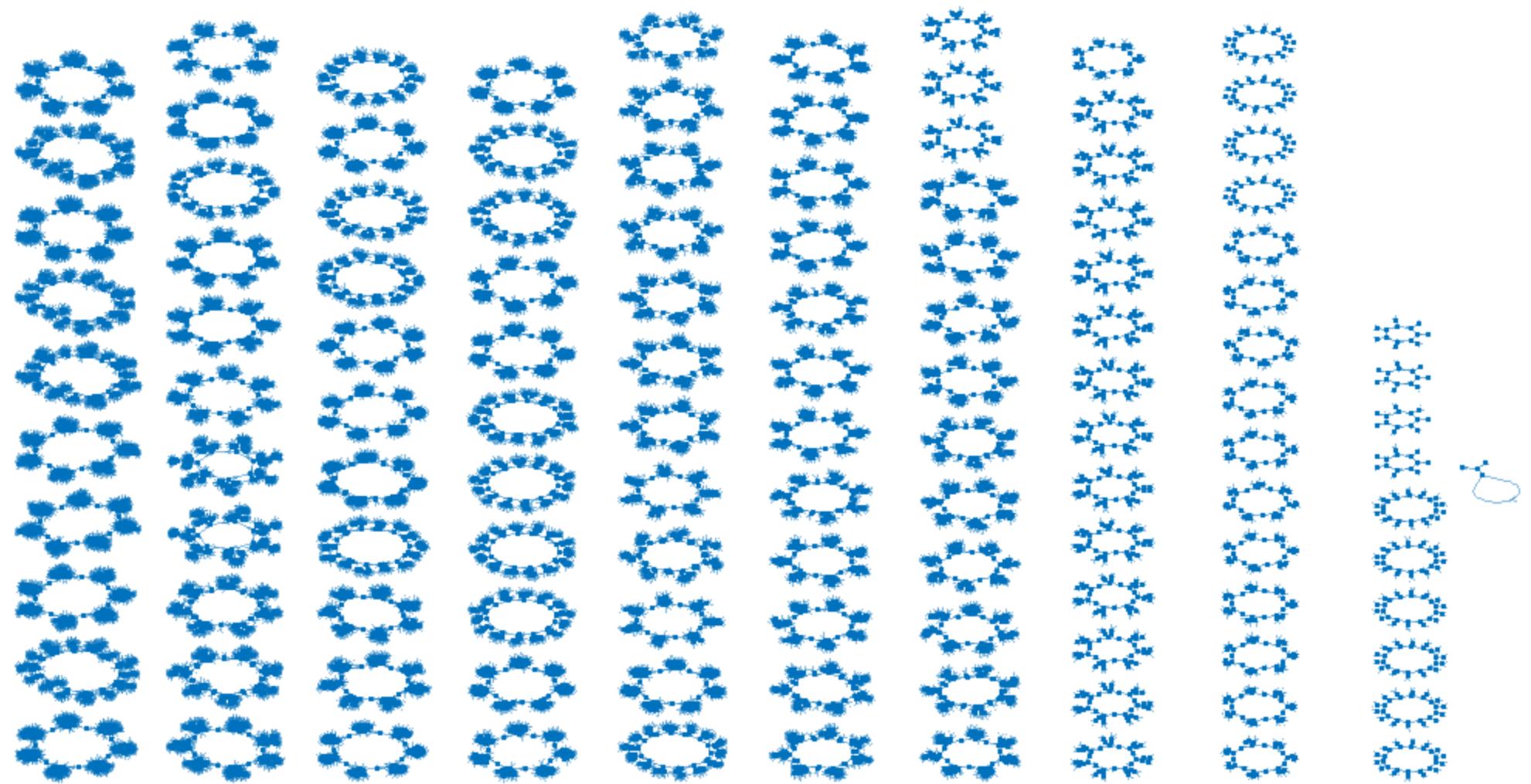




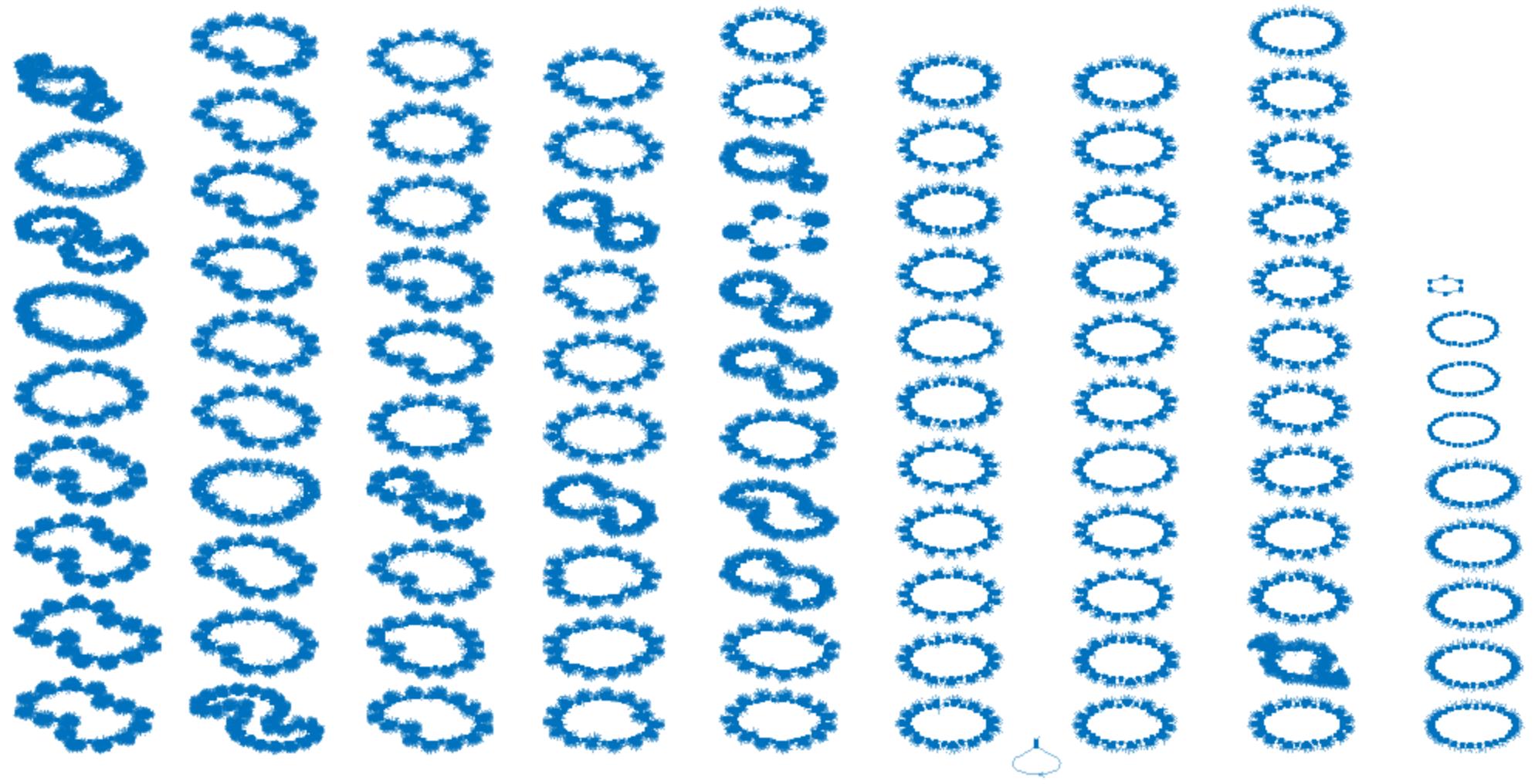


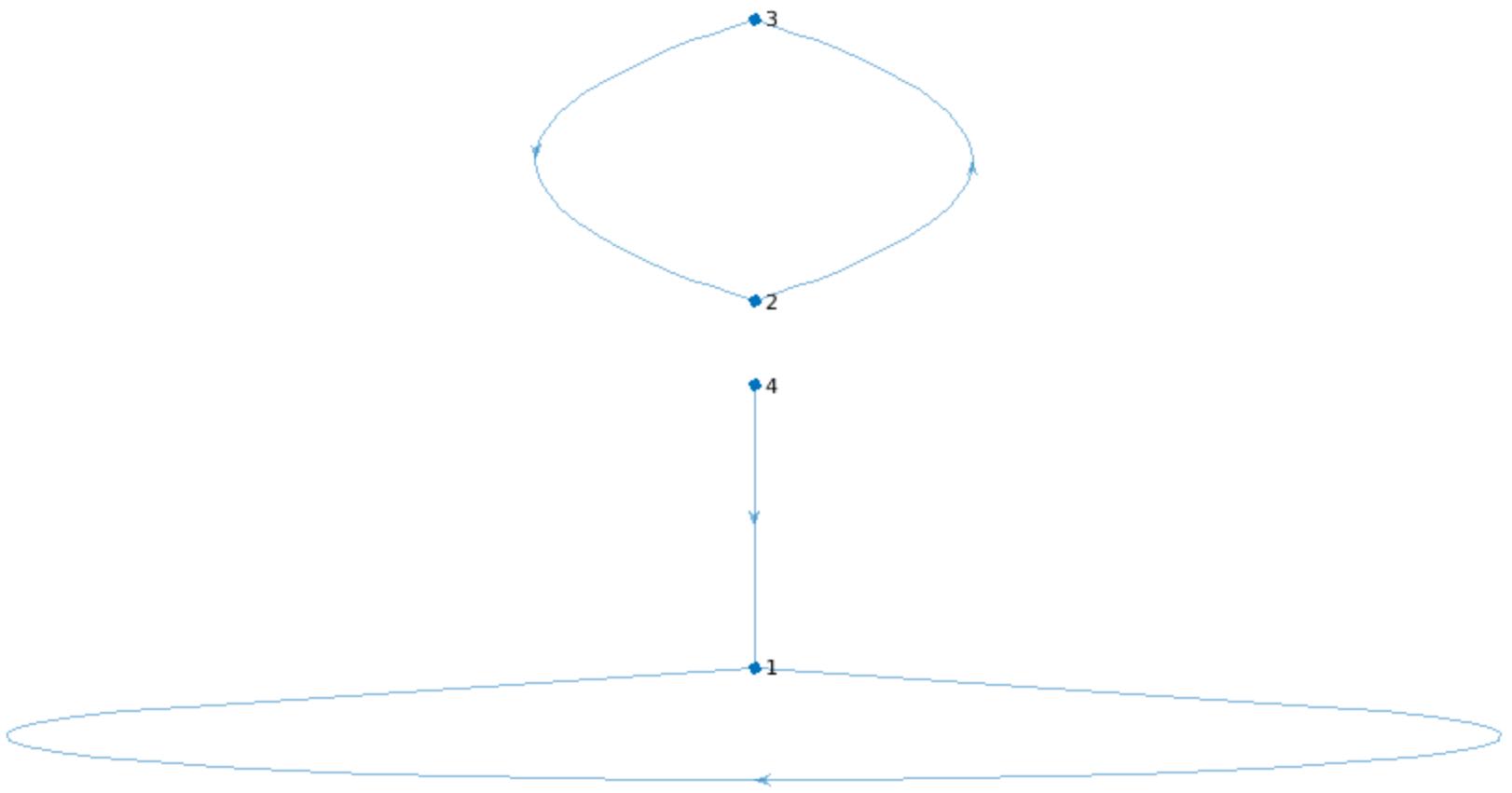


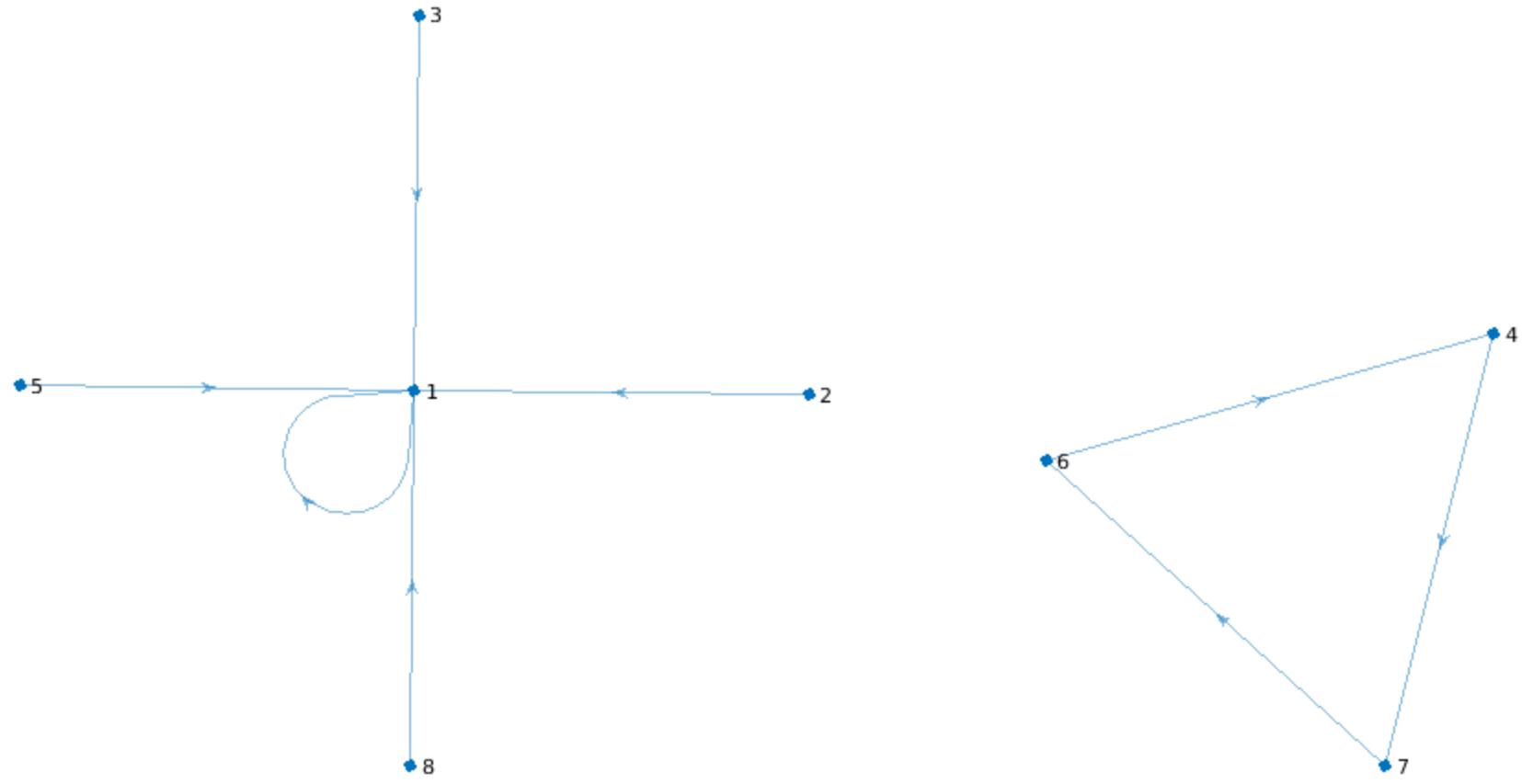


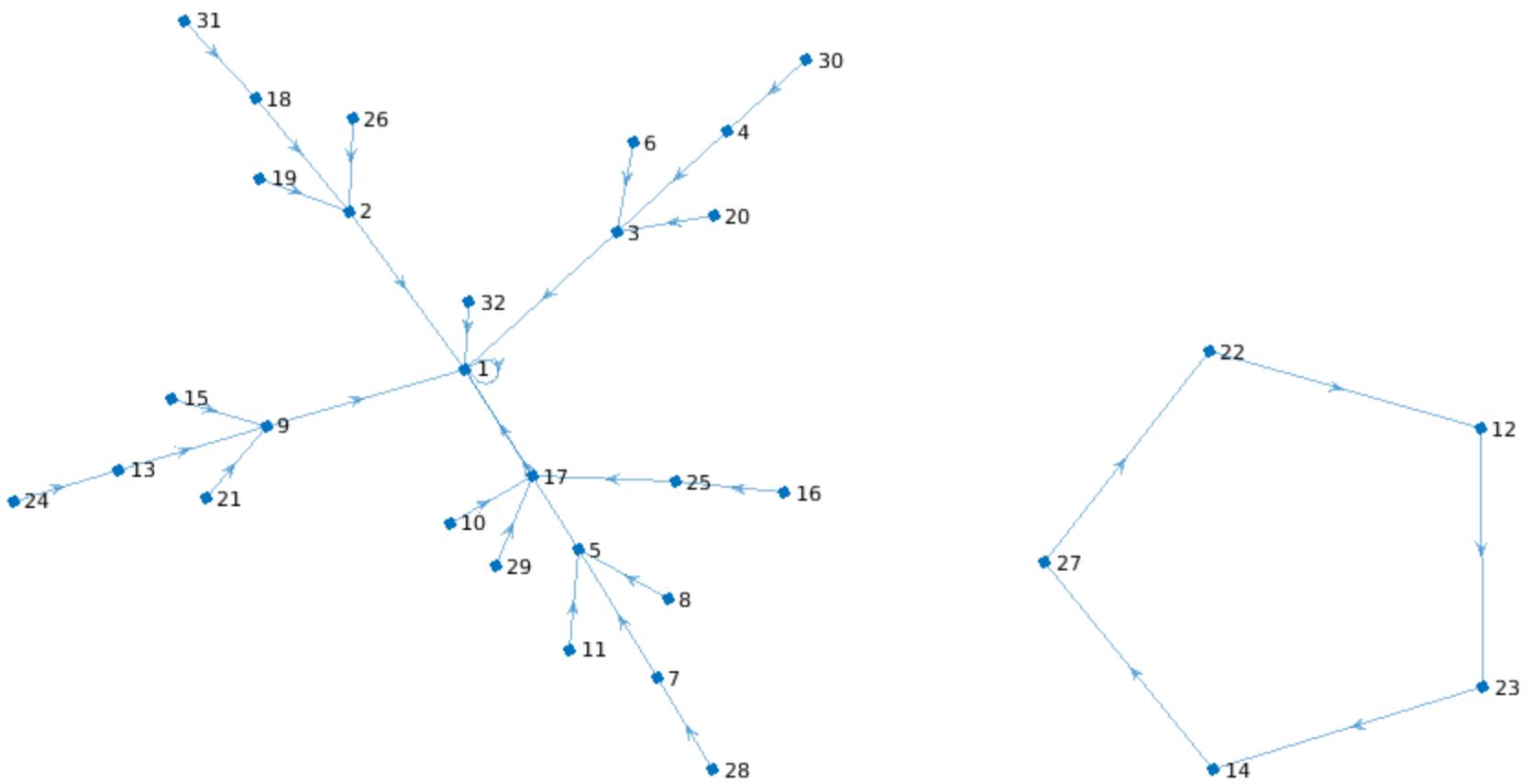
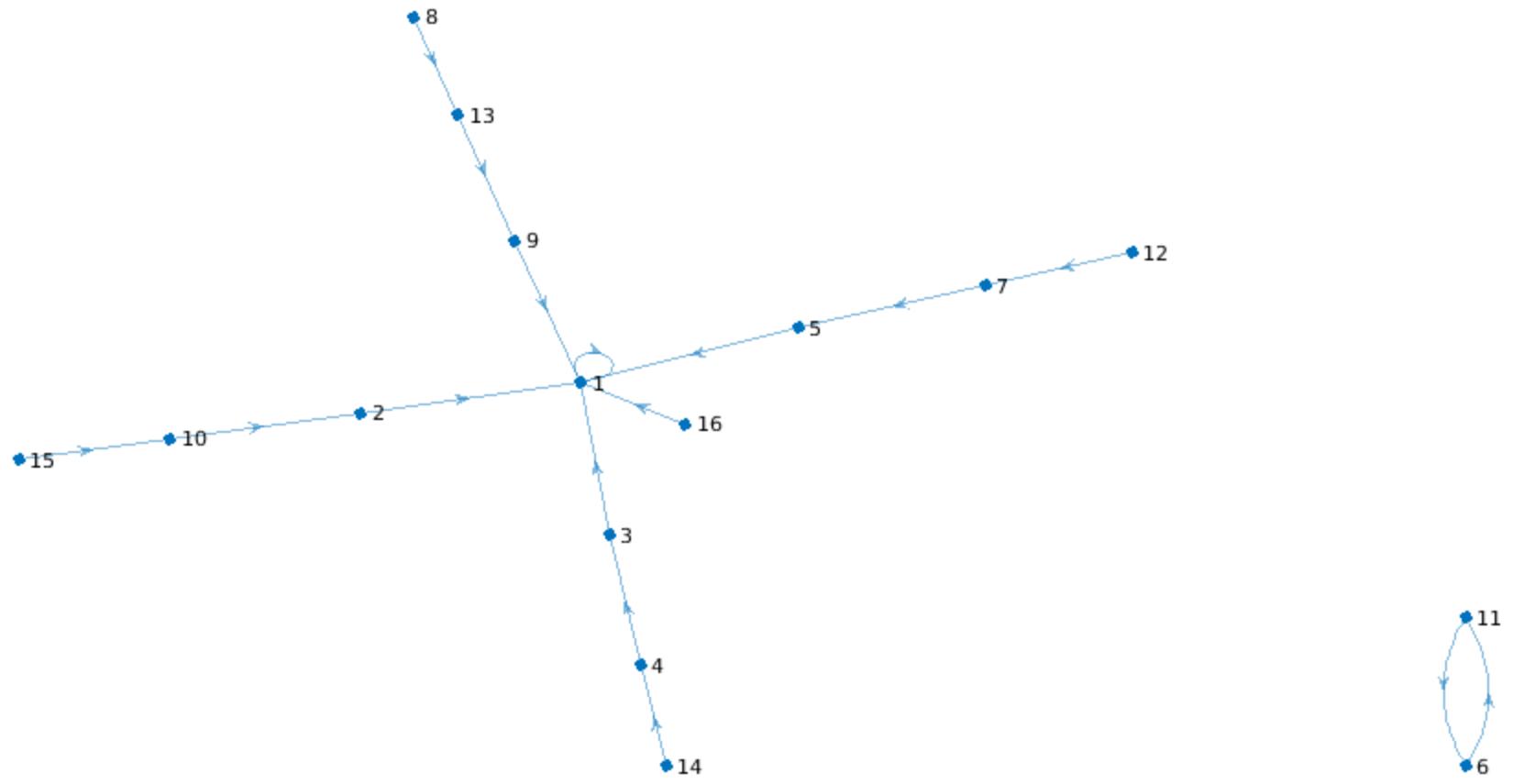


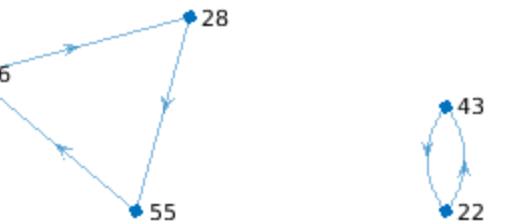
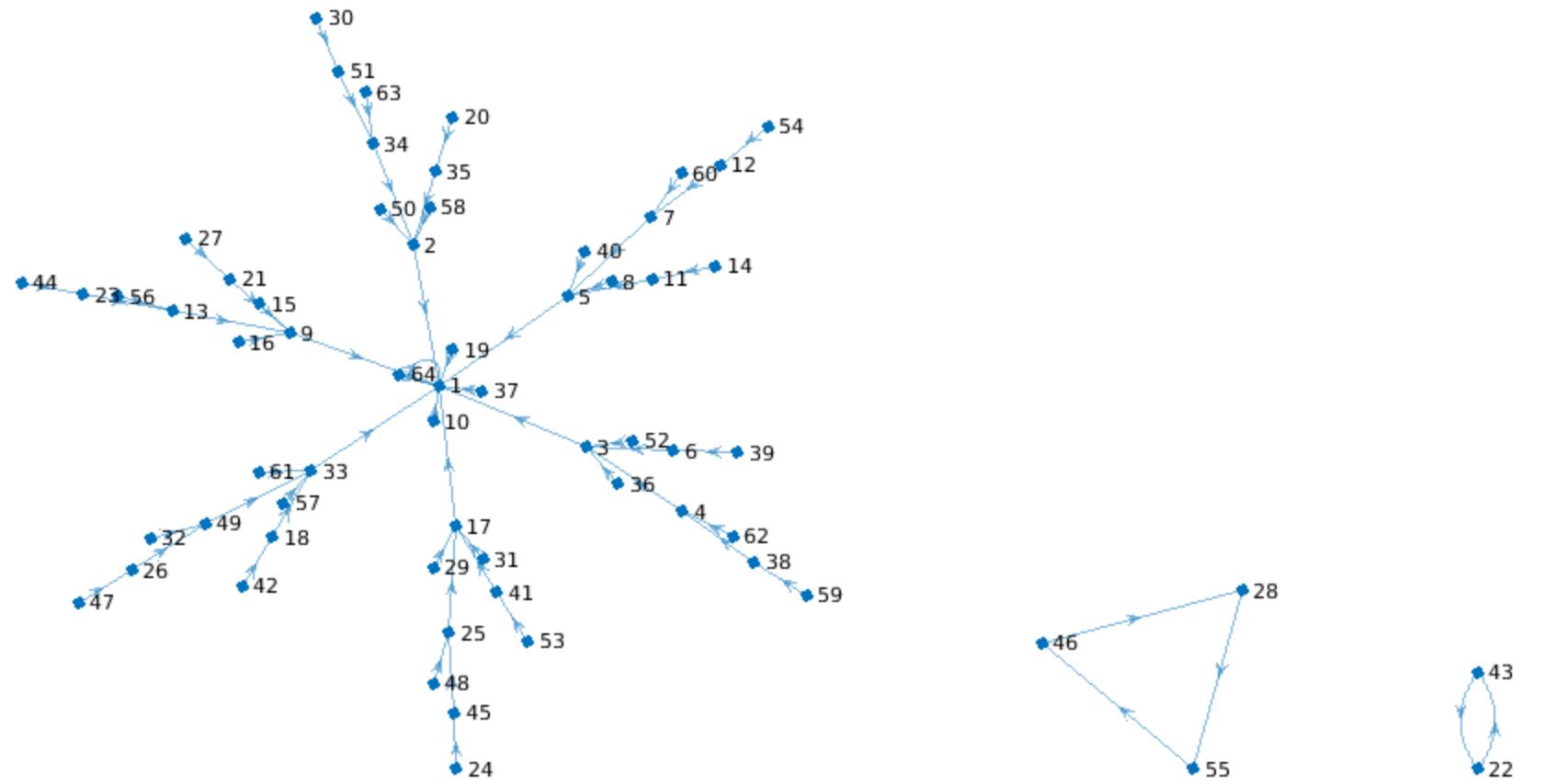
10

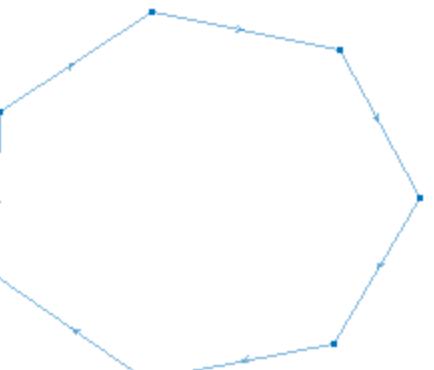
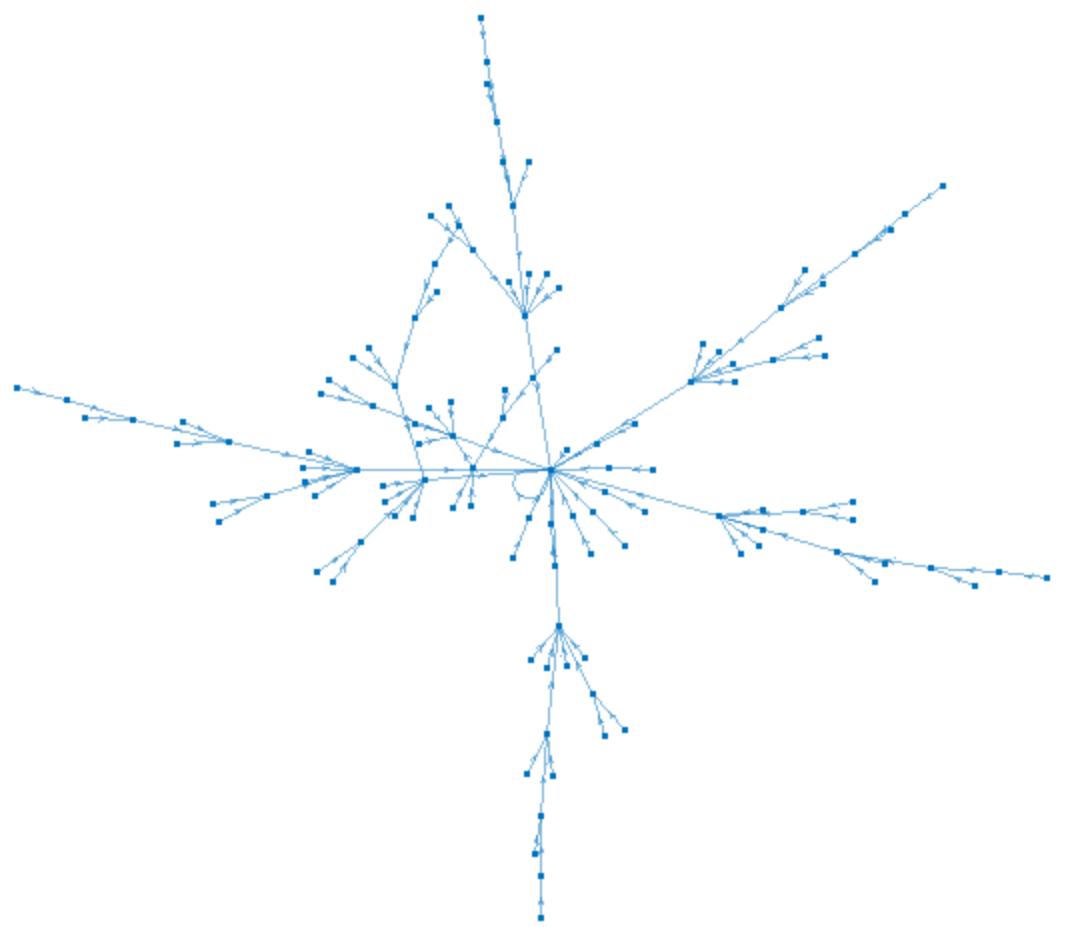


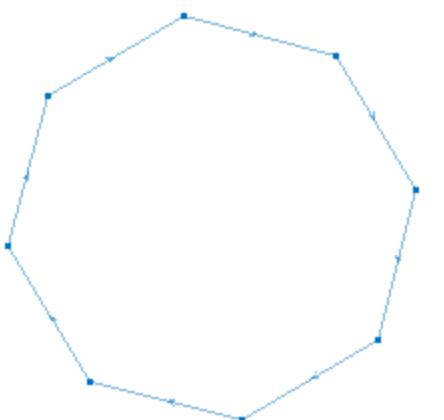
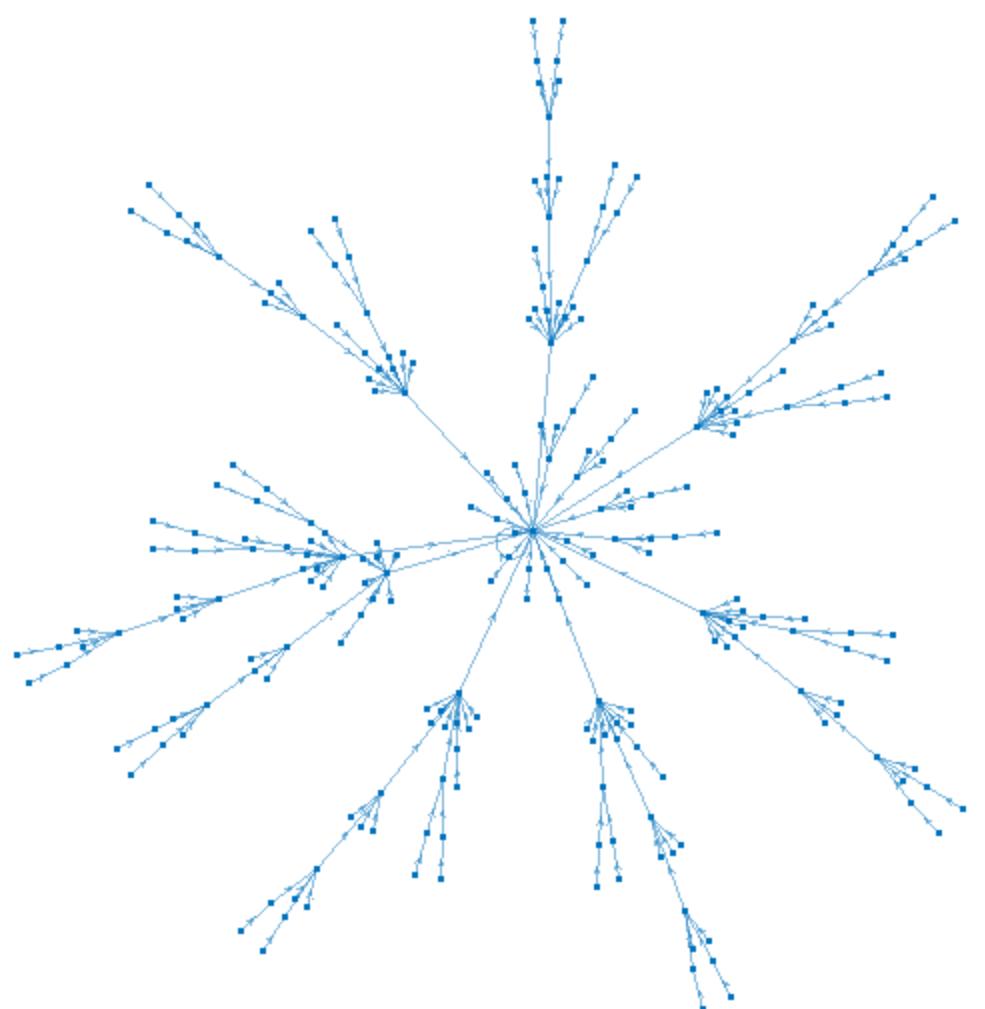


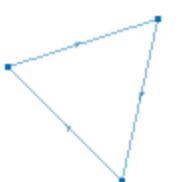
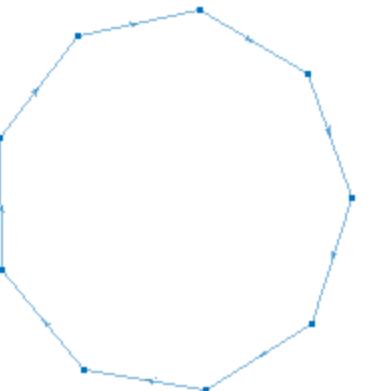
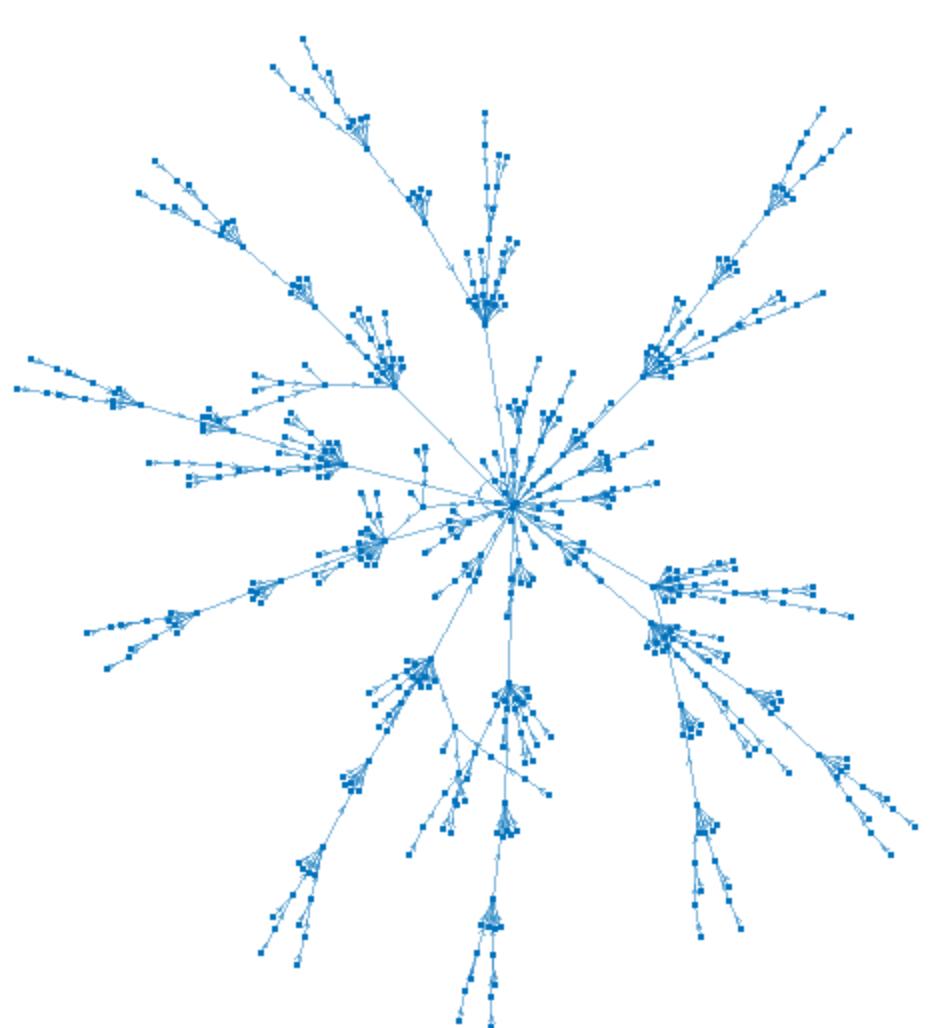


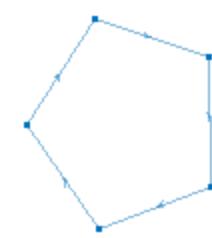
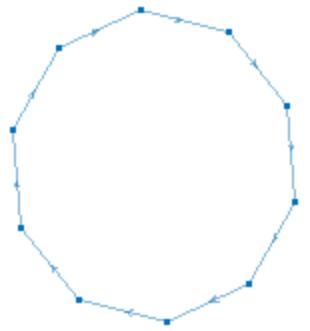
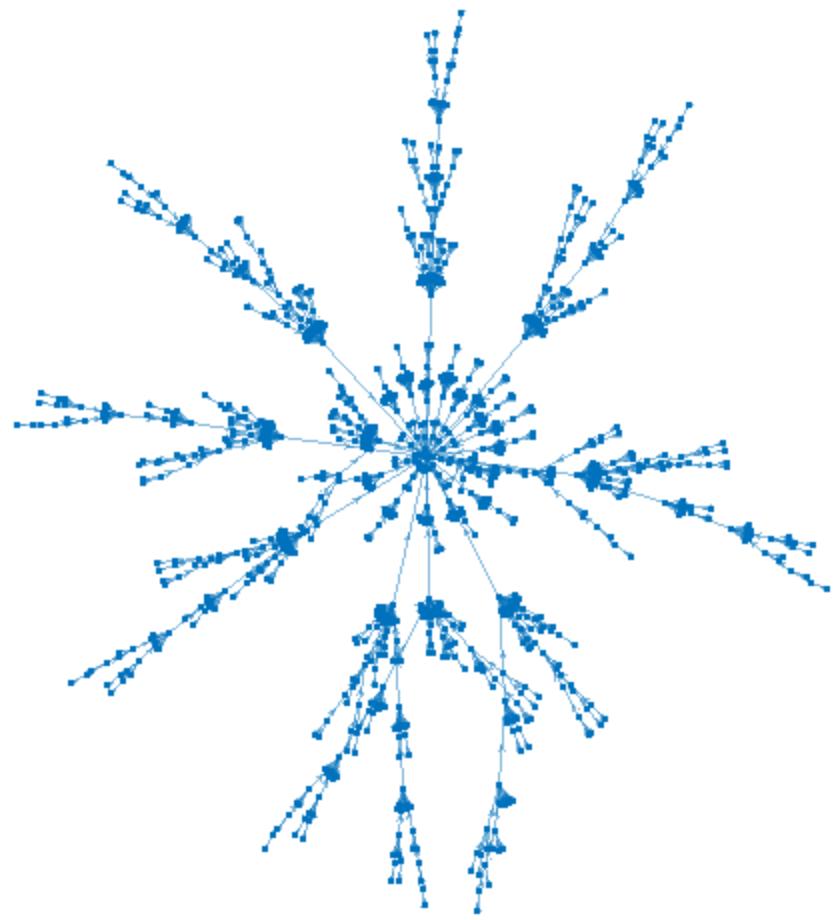


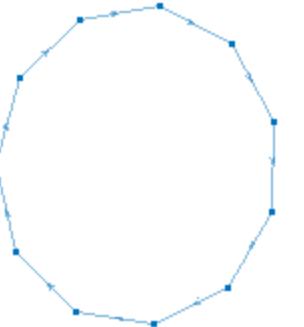
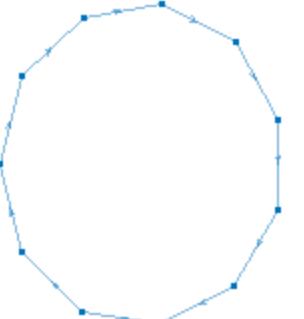
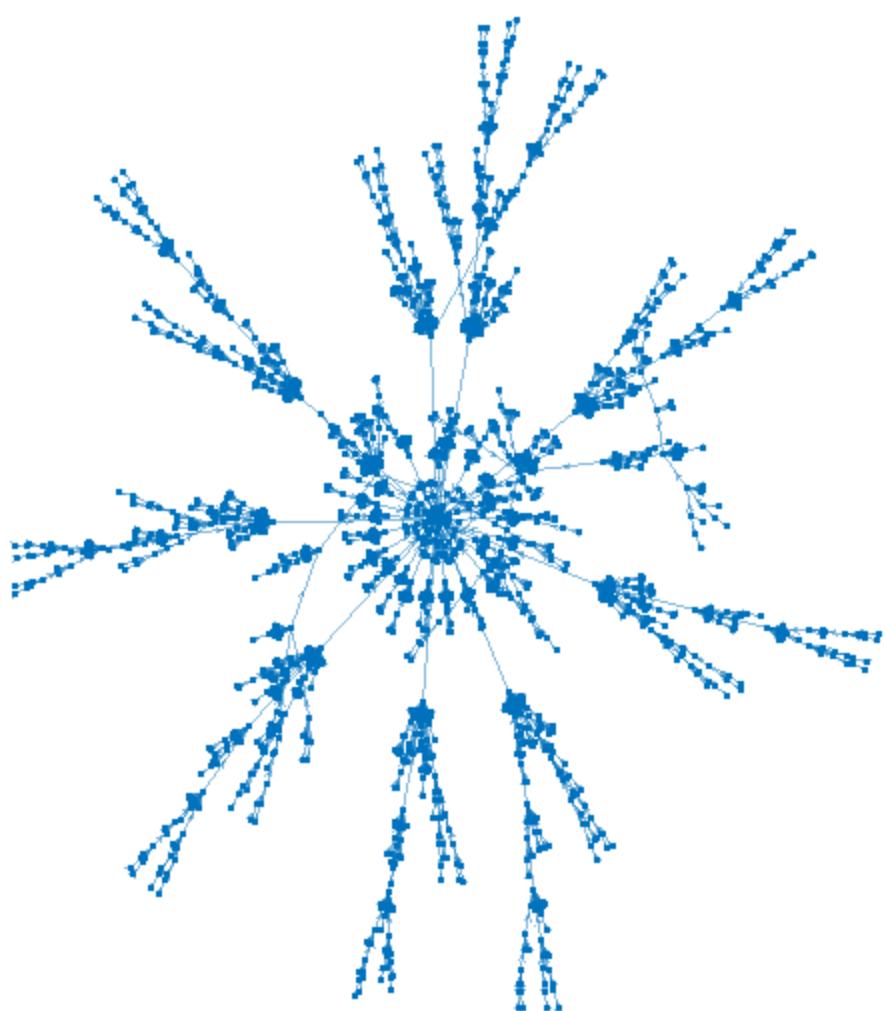


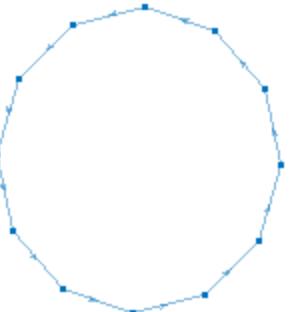
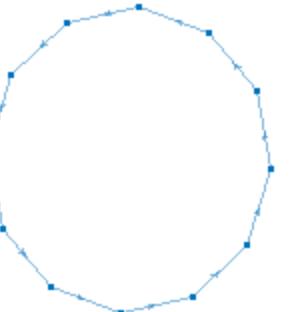
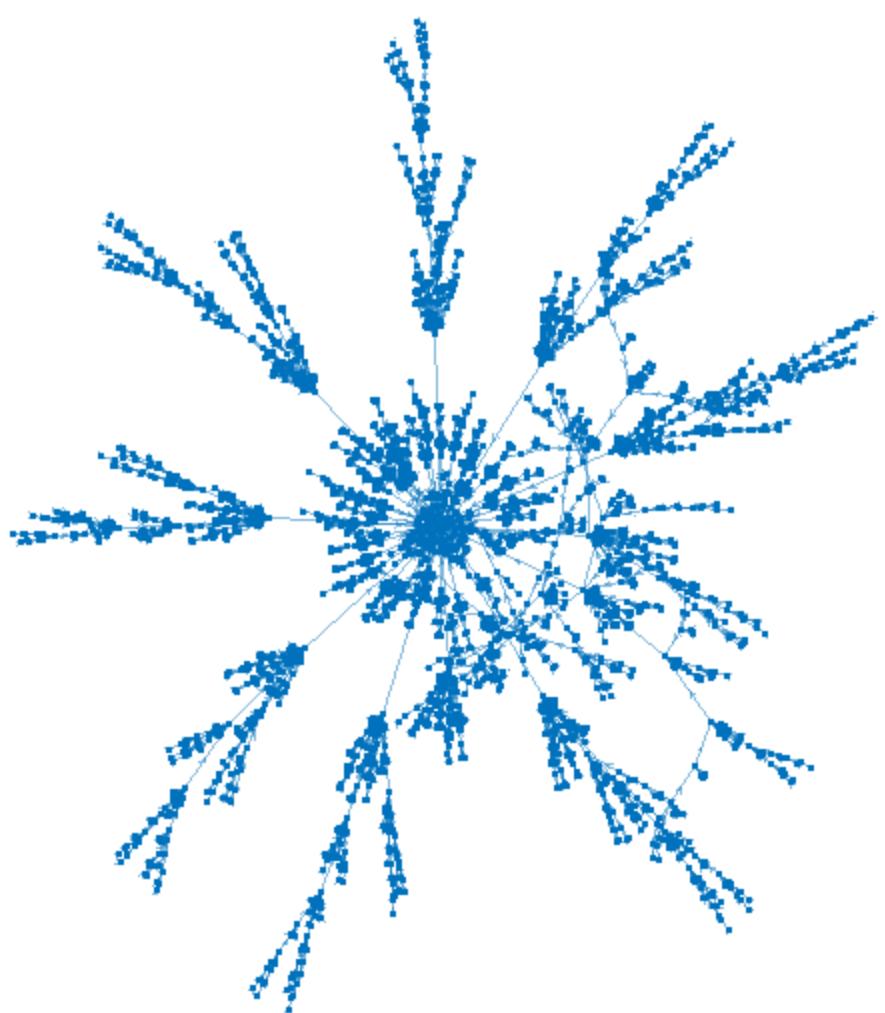


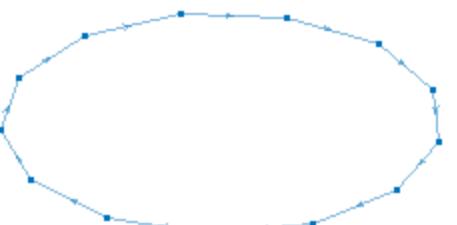
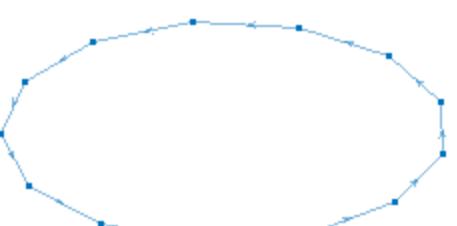
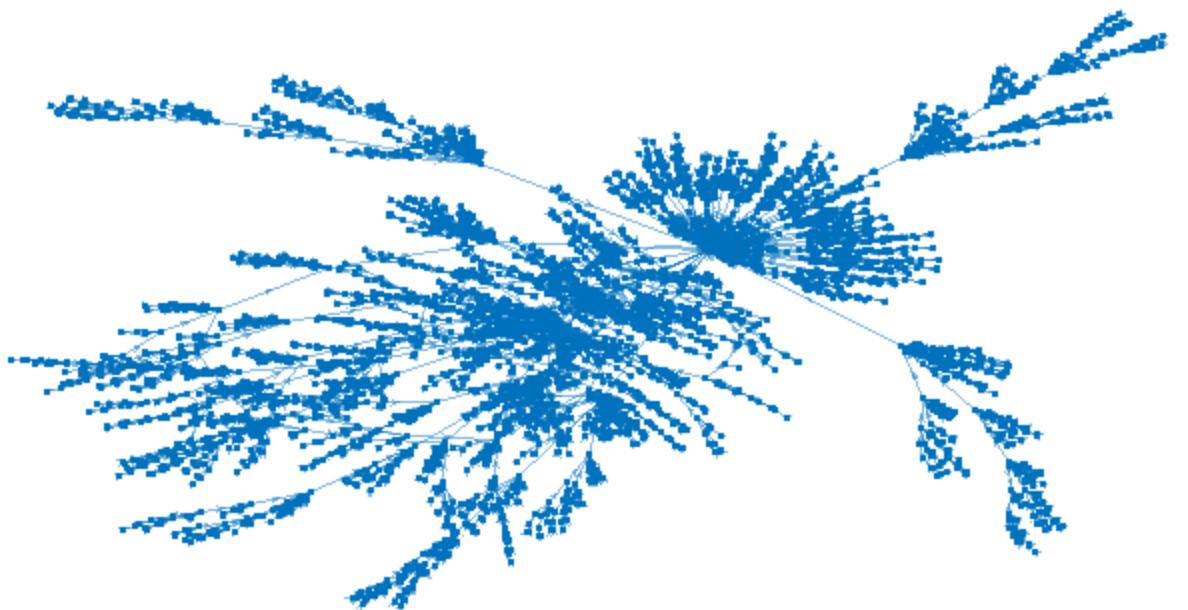
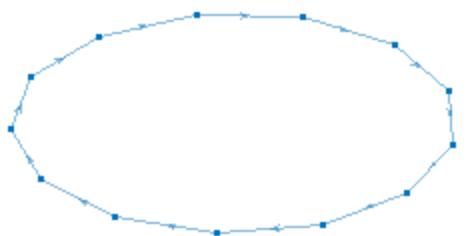


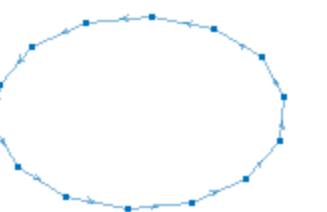
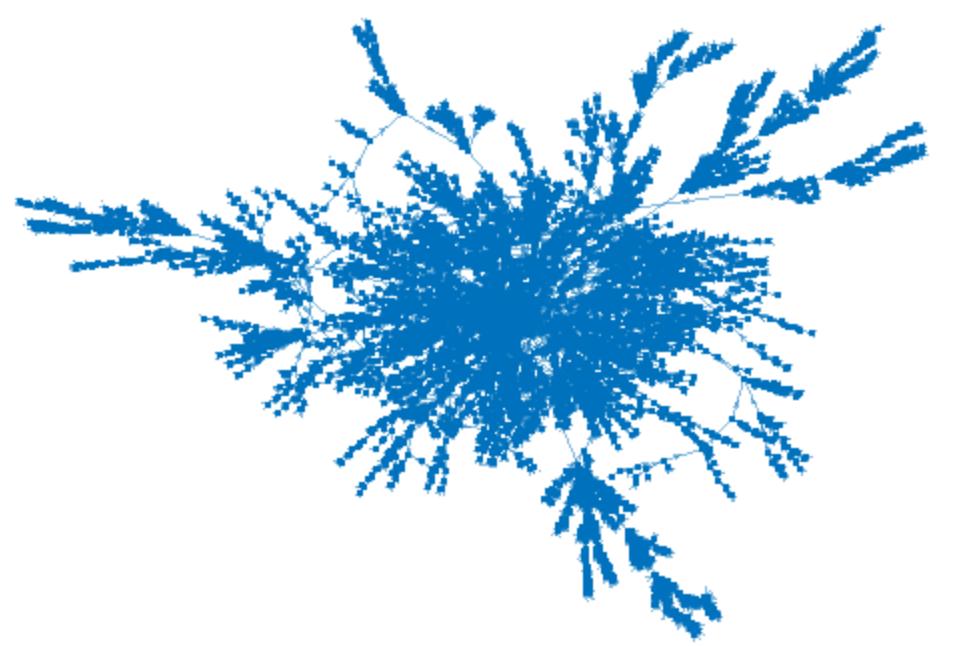
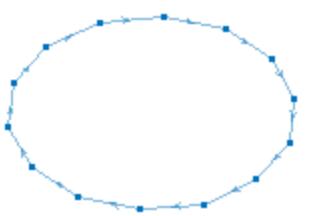
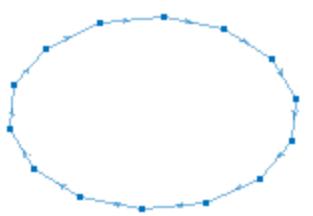




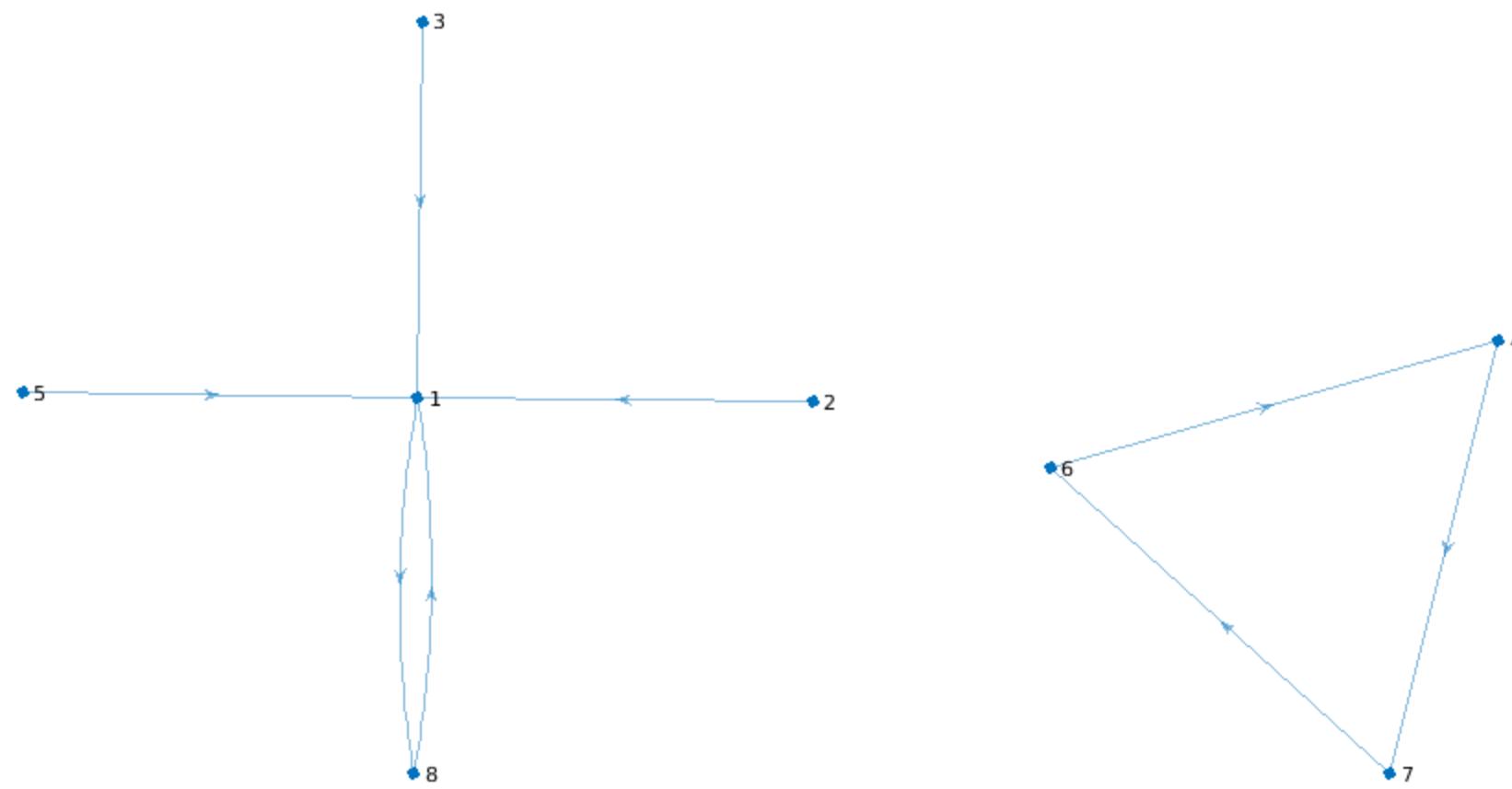
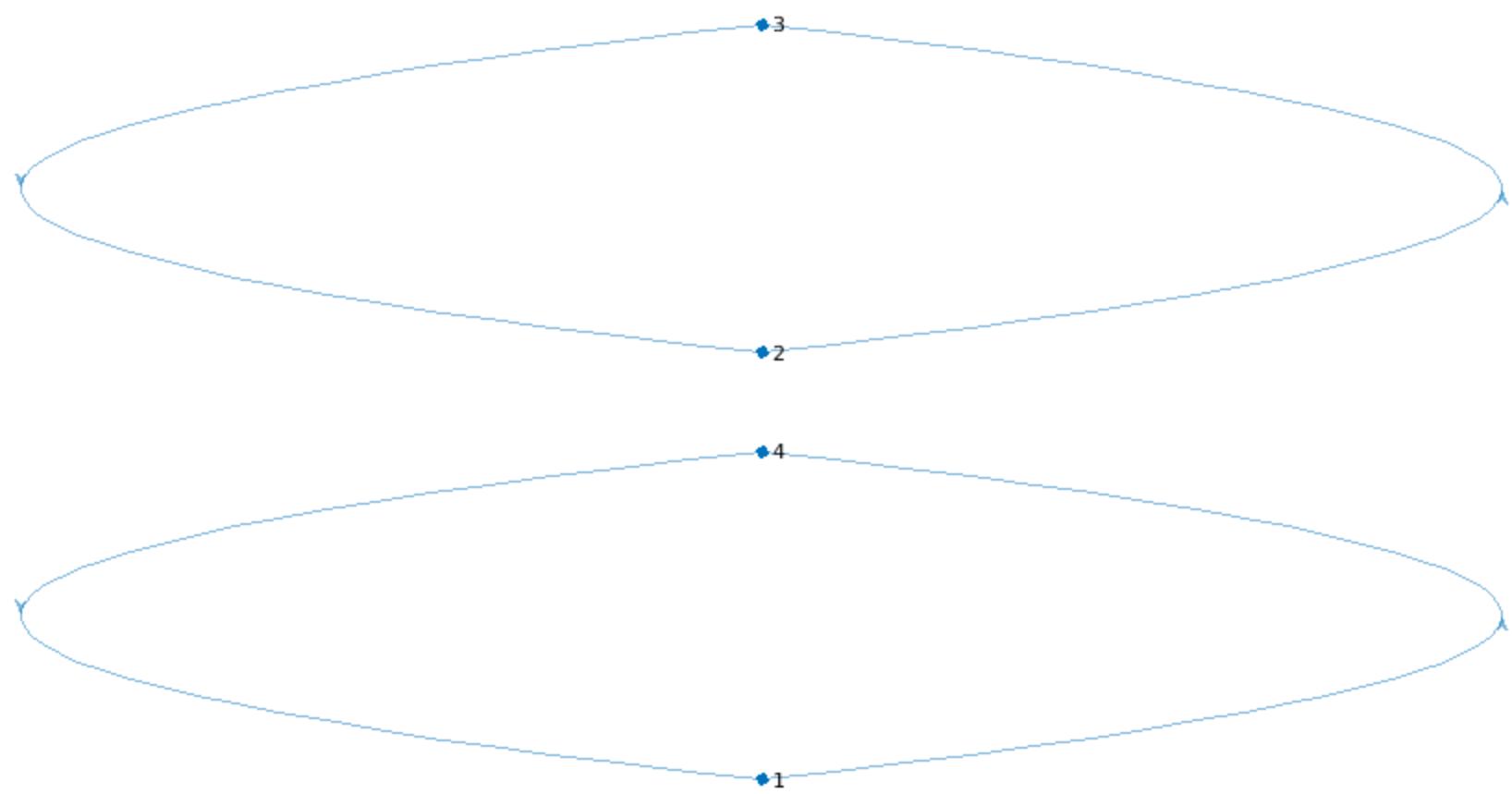


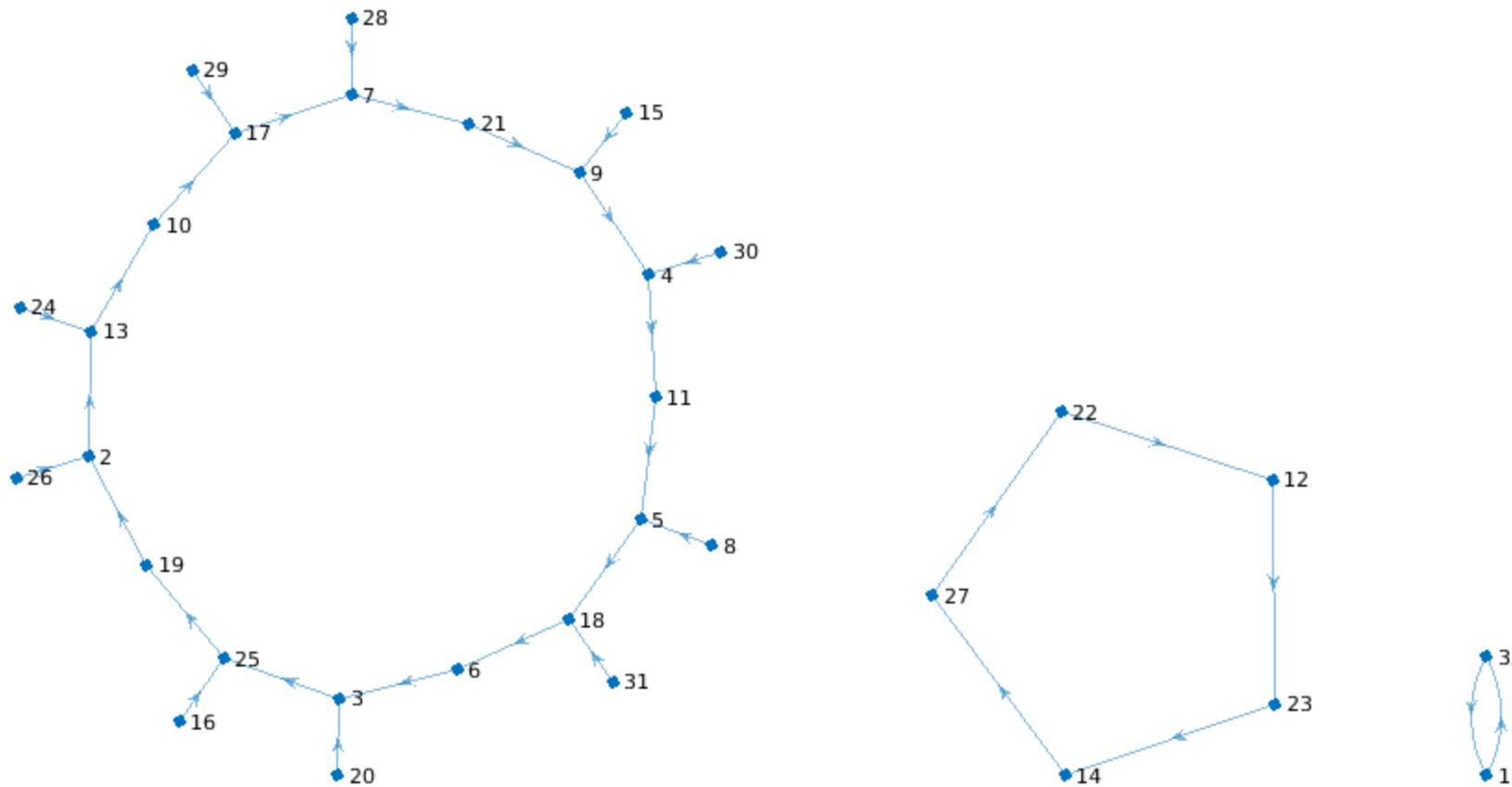
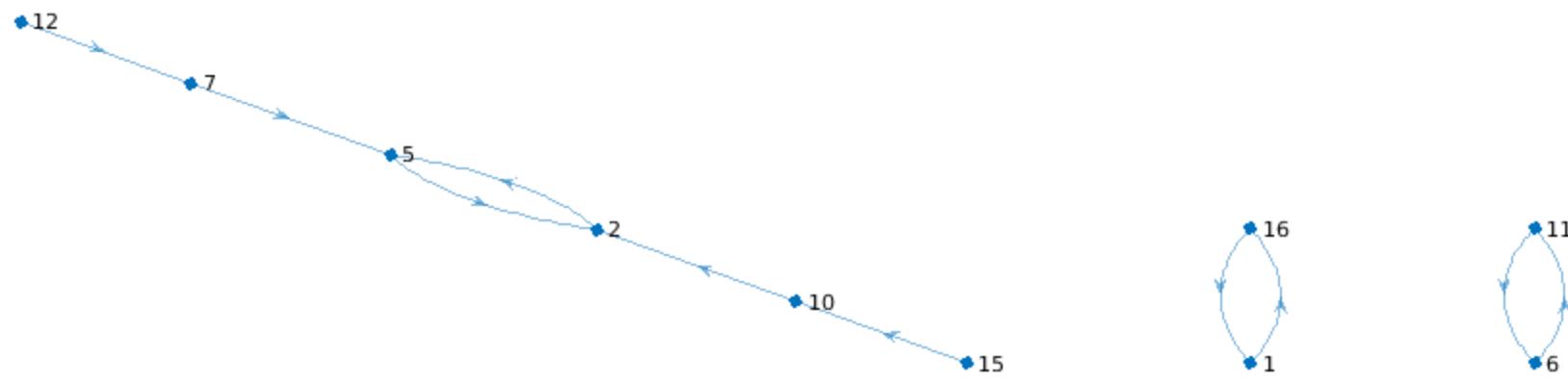
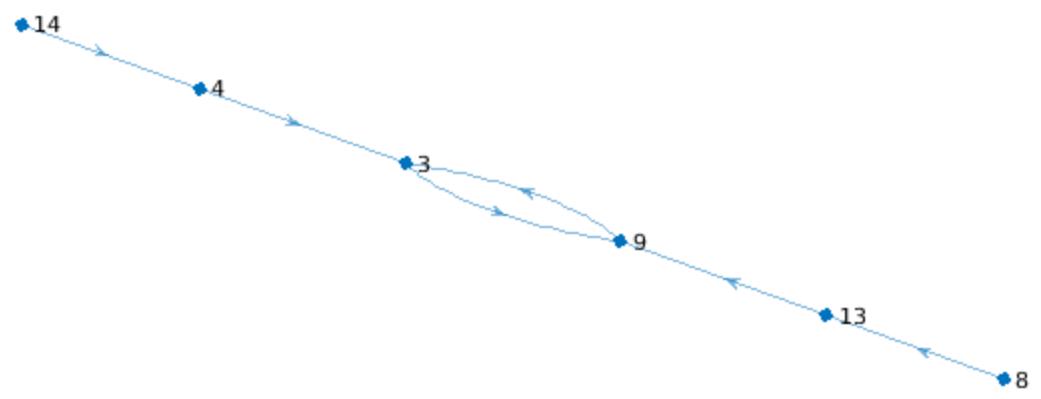


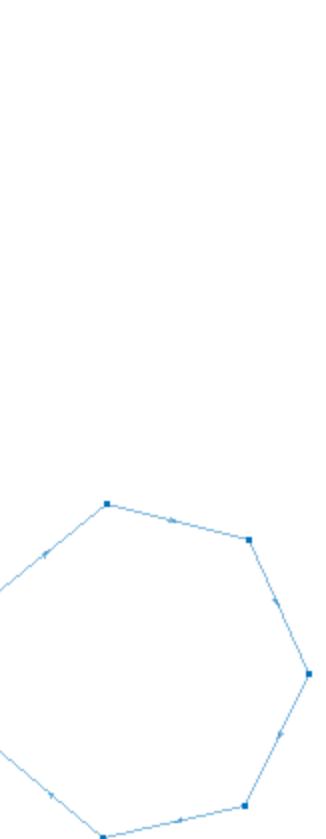
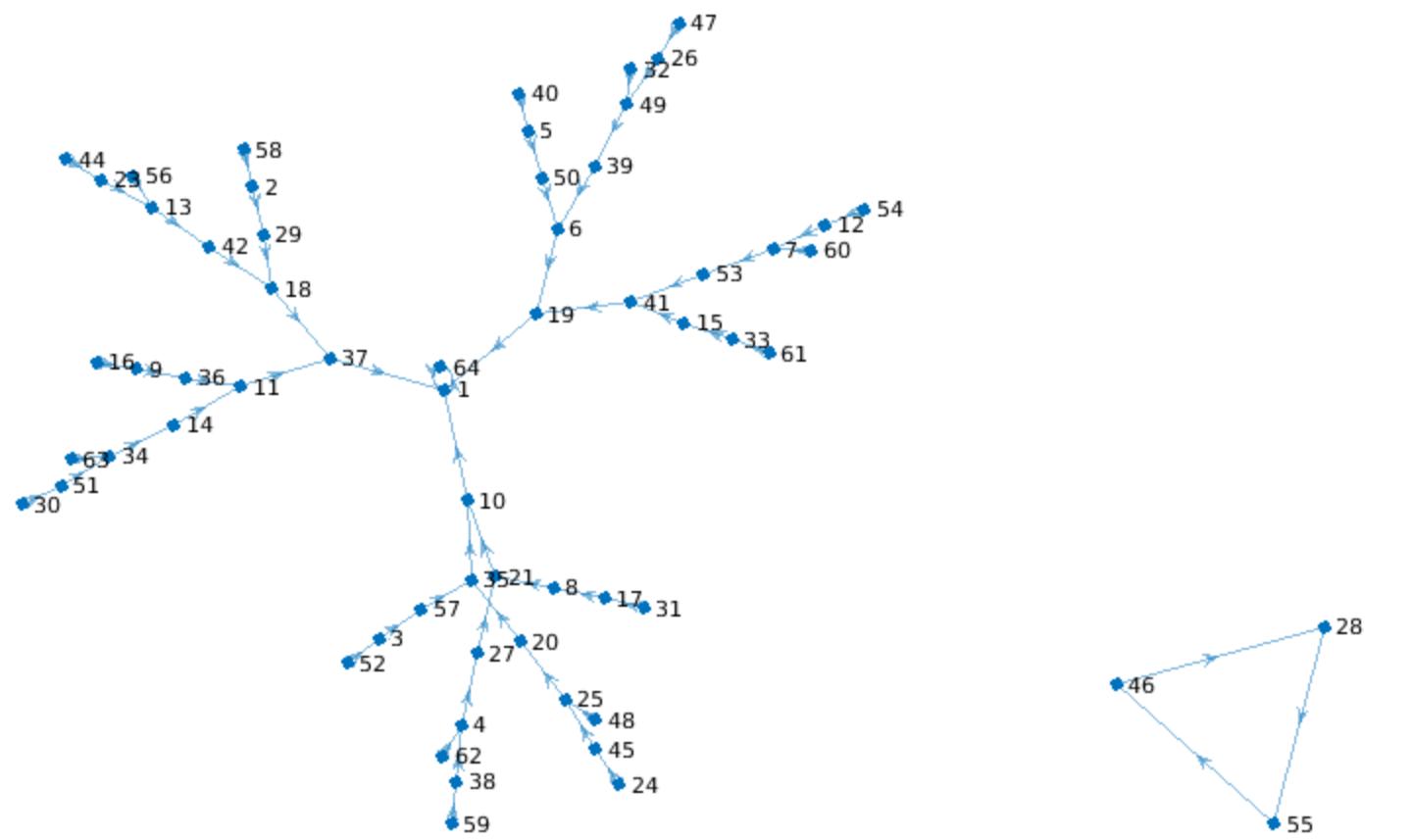


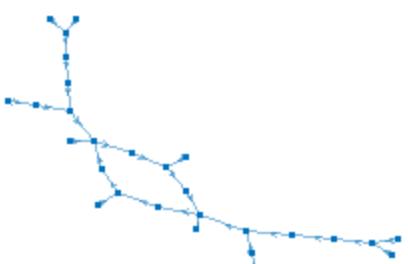
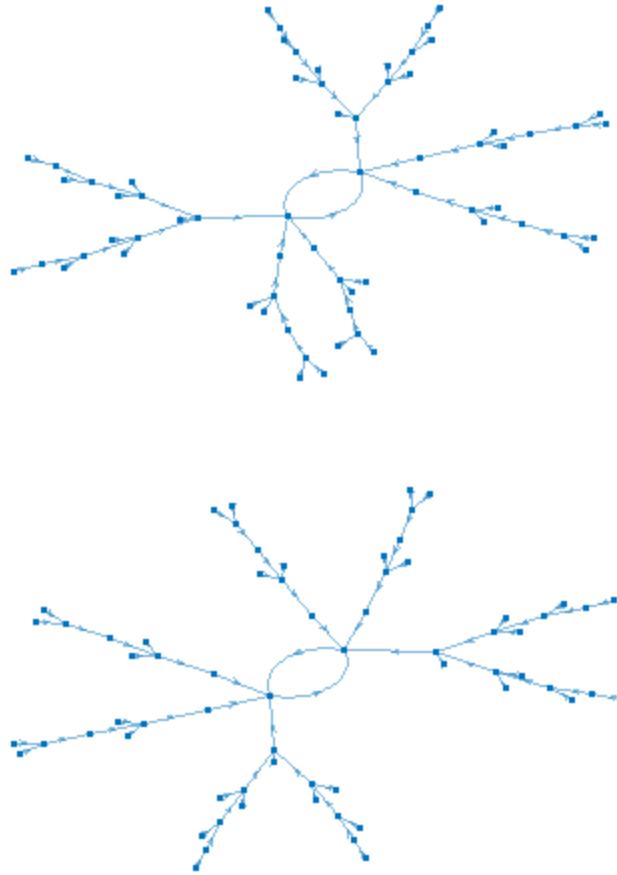


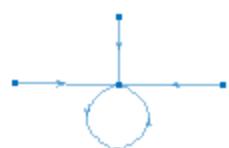
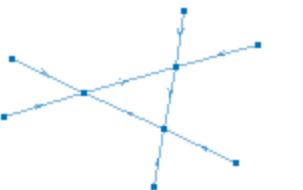
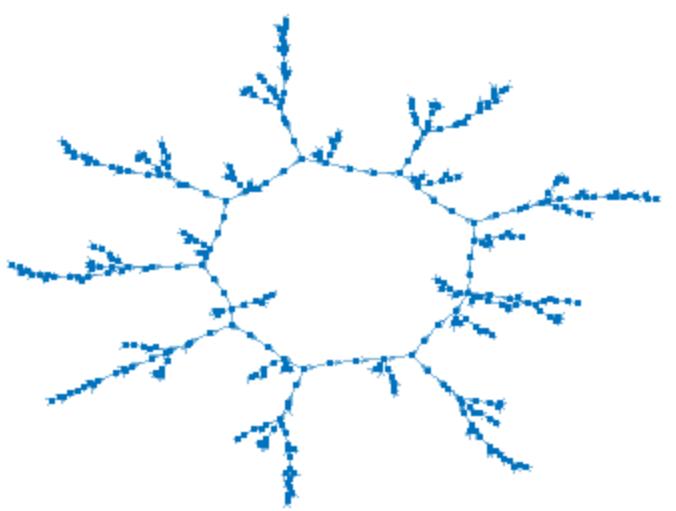
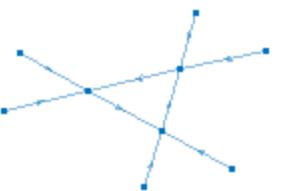
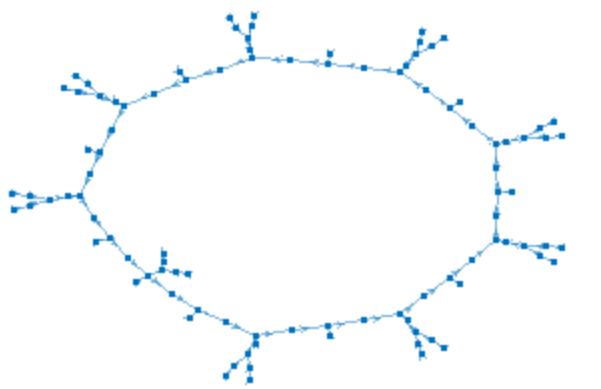
41

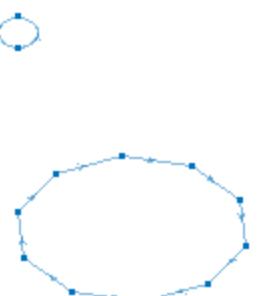
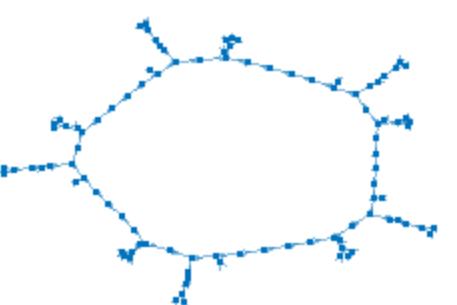
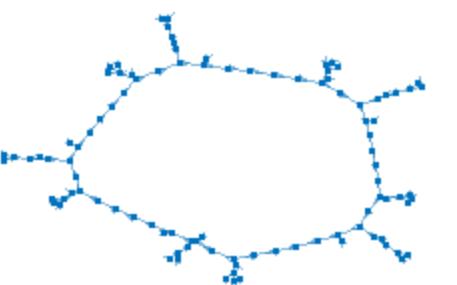
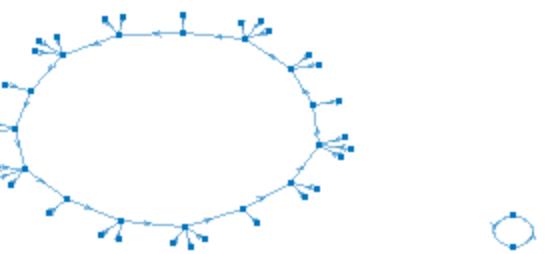
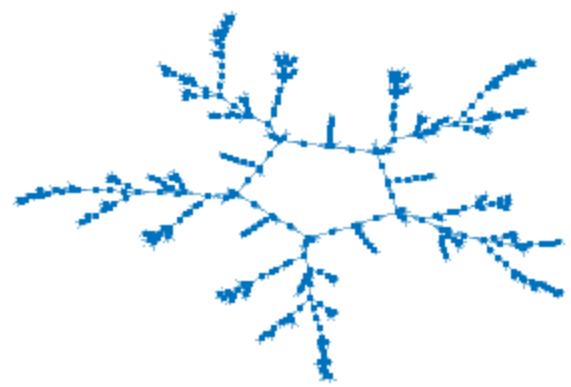
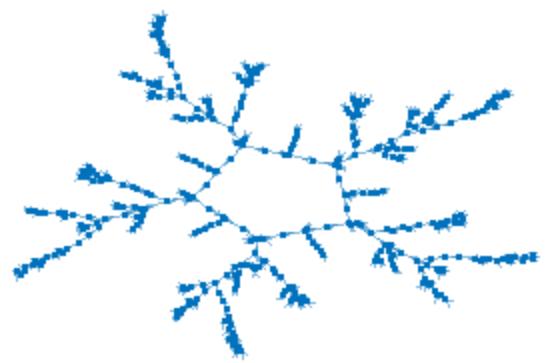


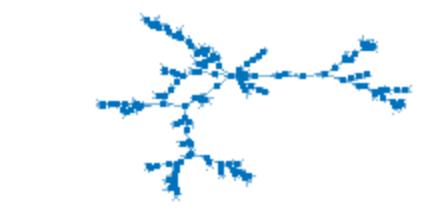
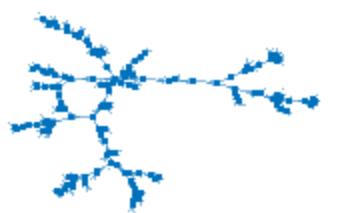
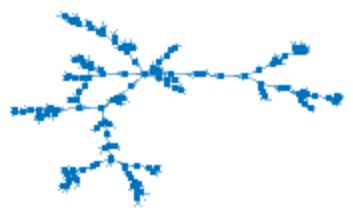




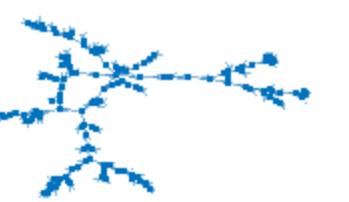
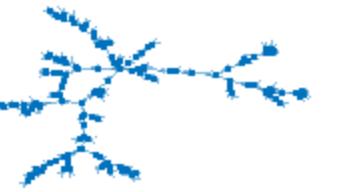
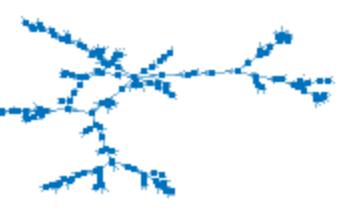
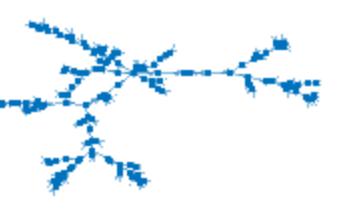
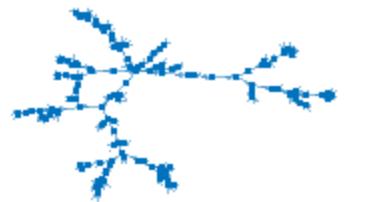
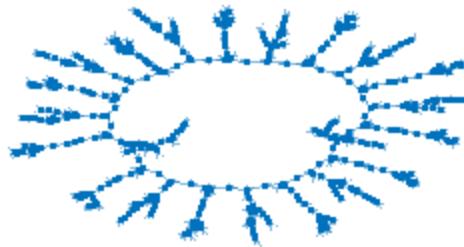
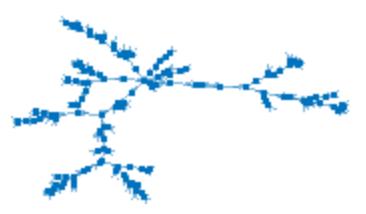
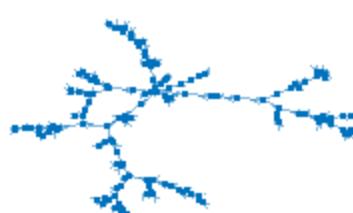


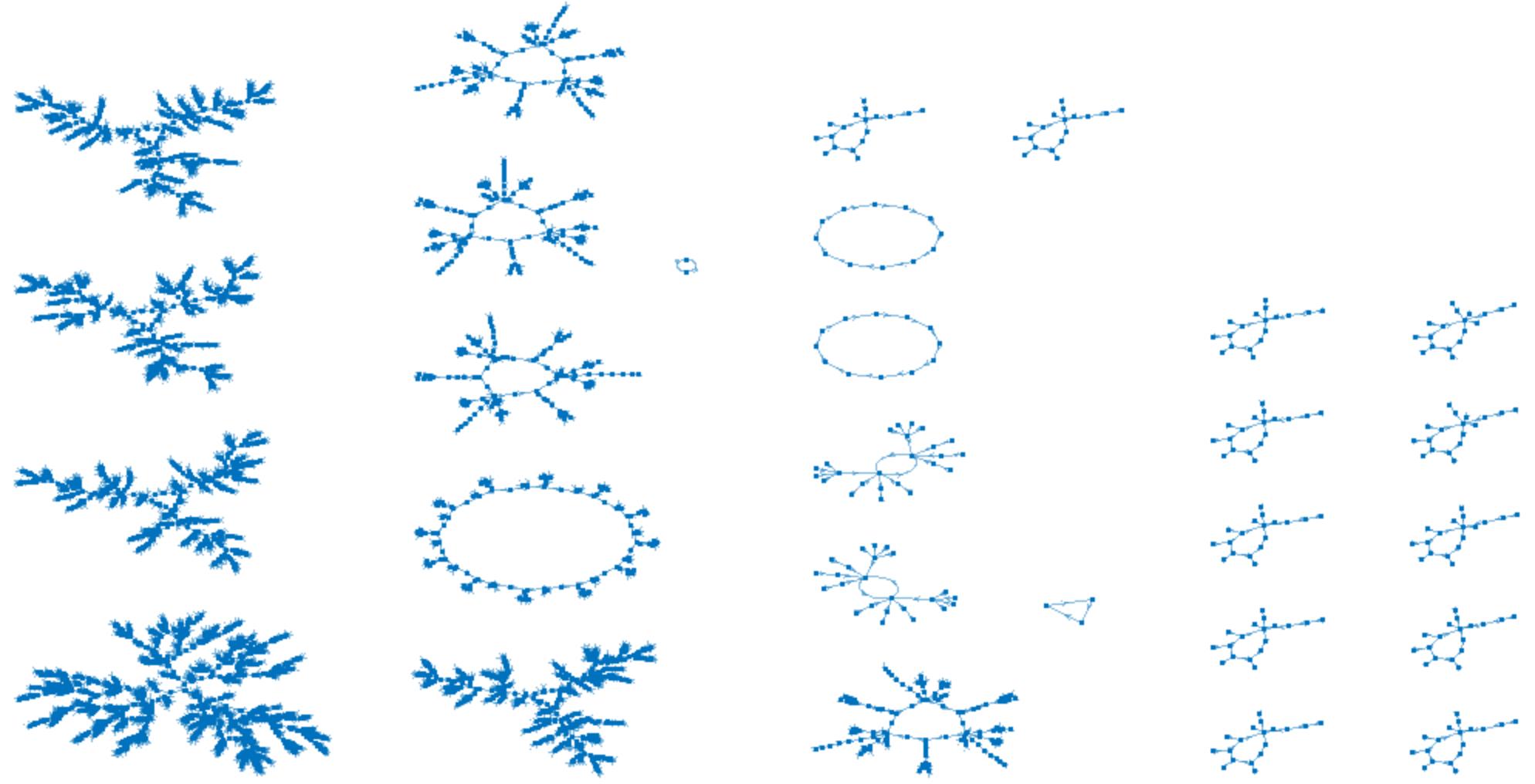


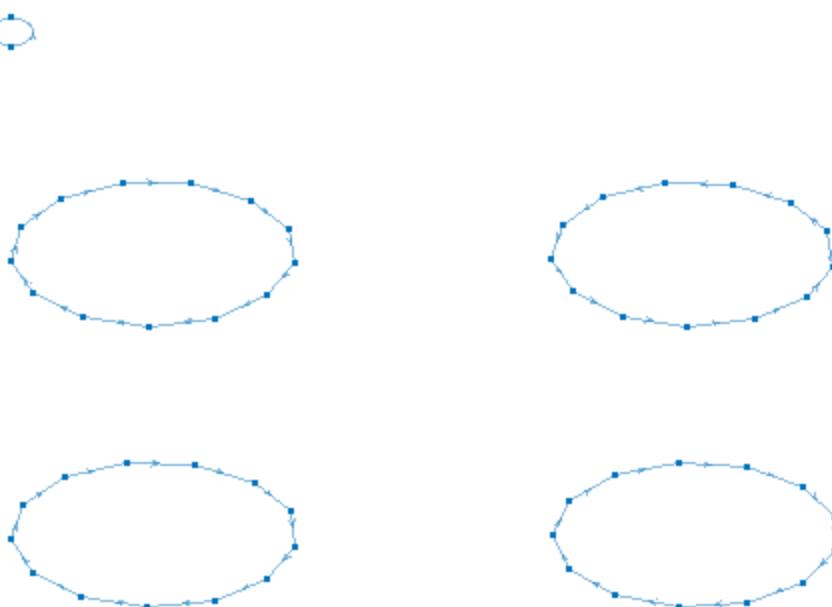
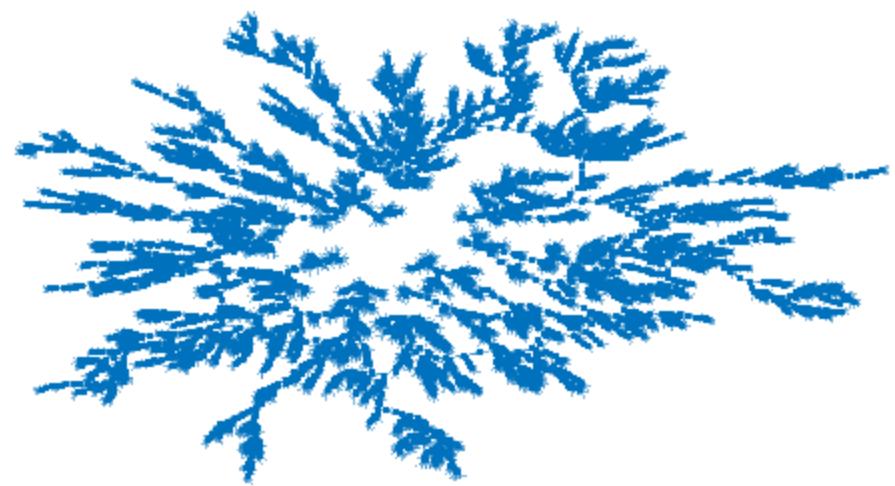
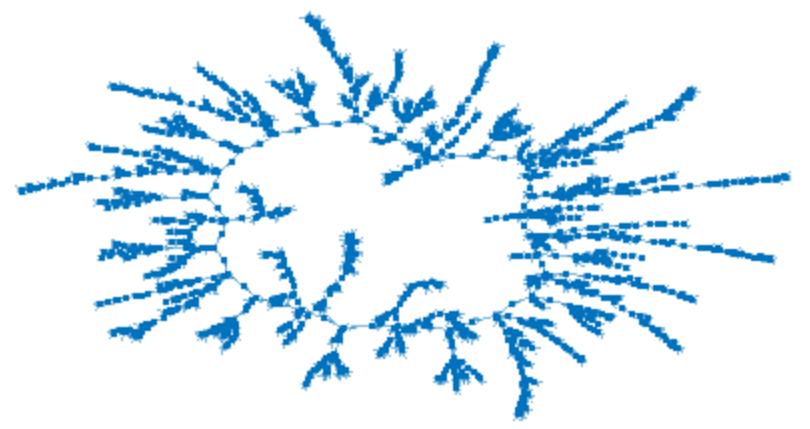


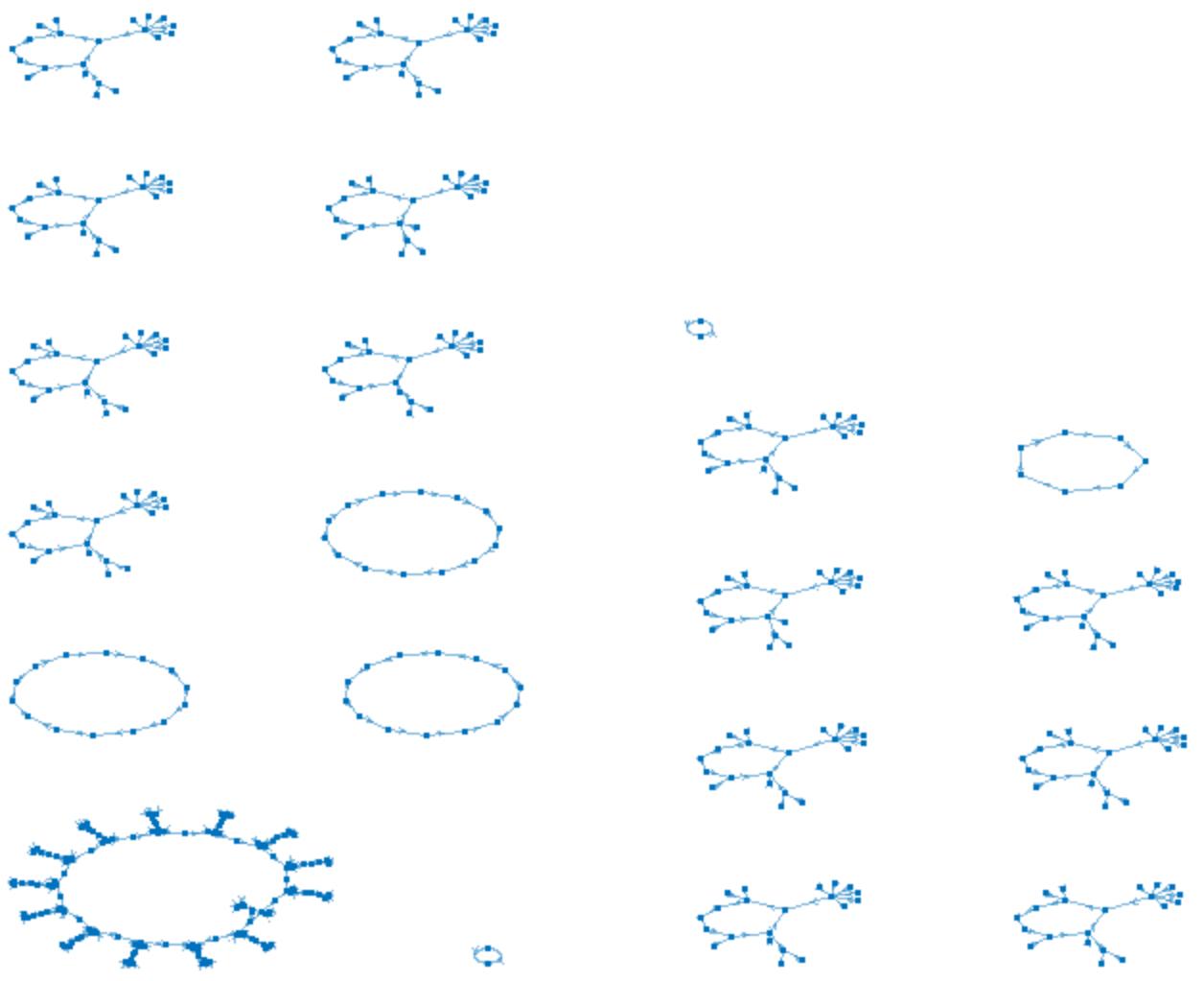
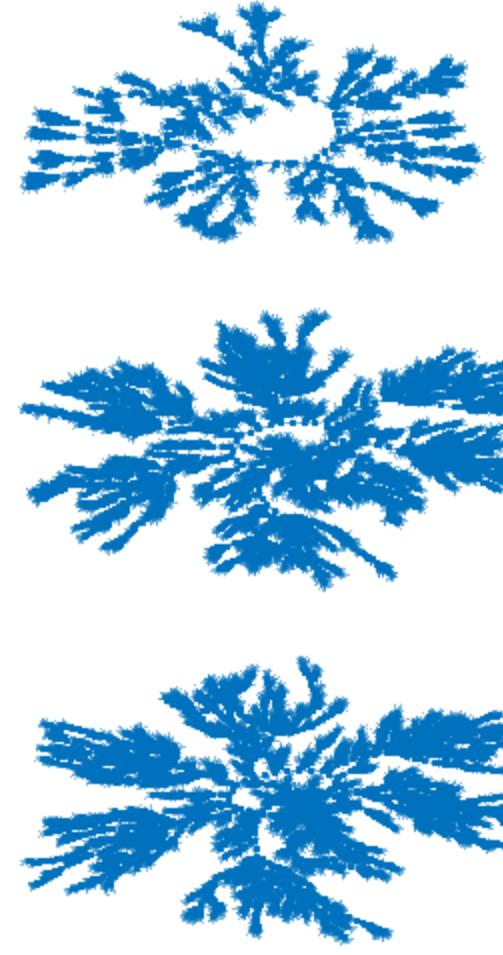


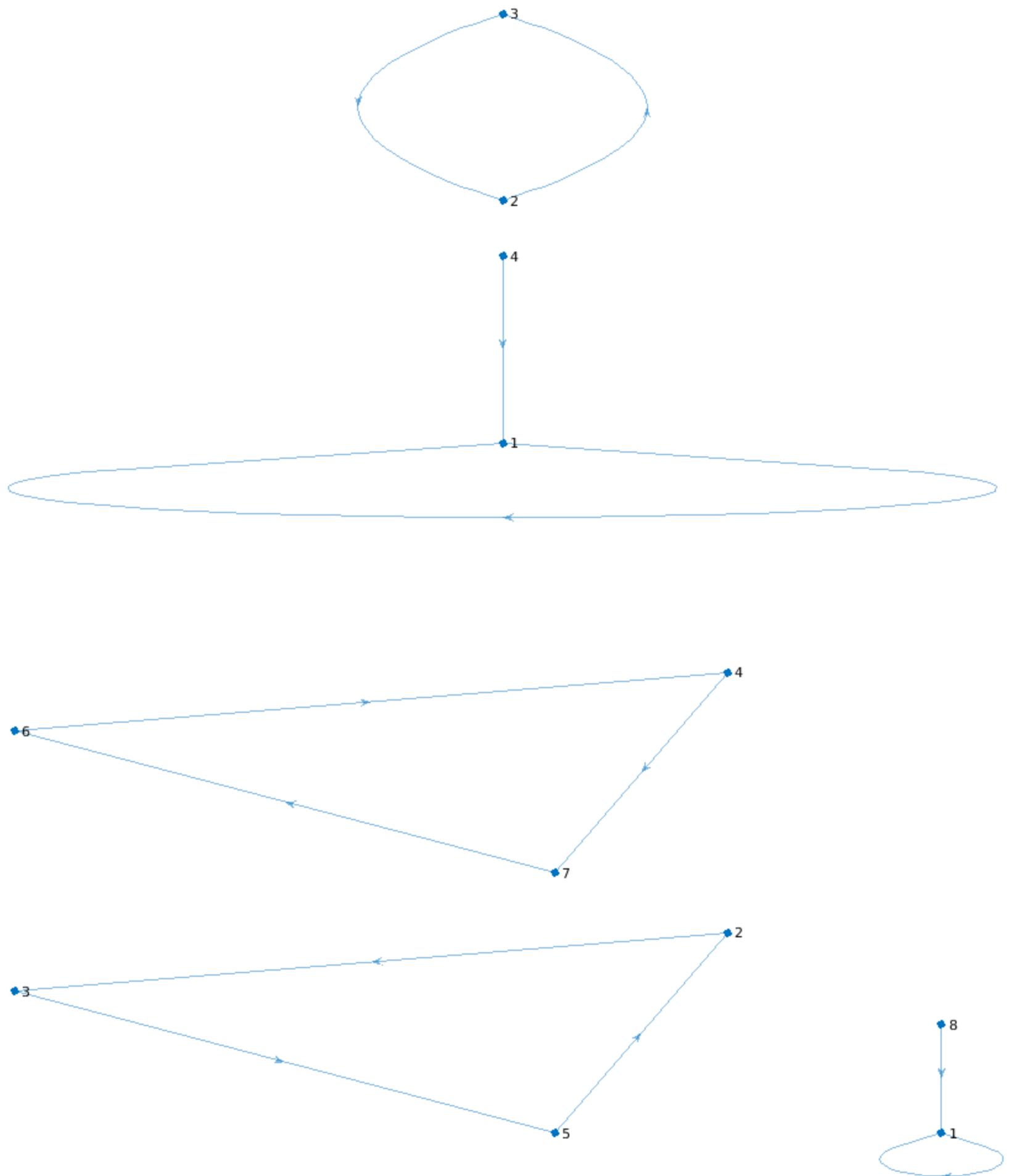
.

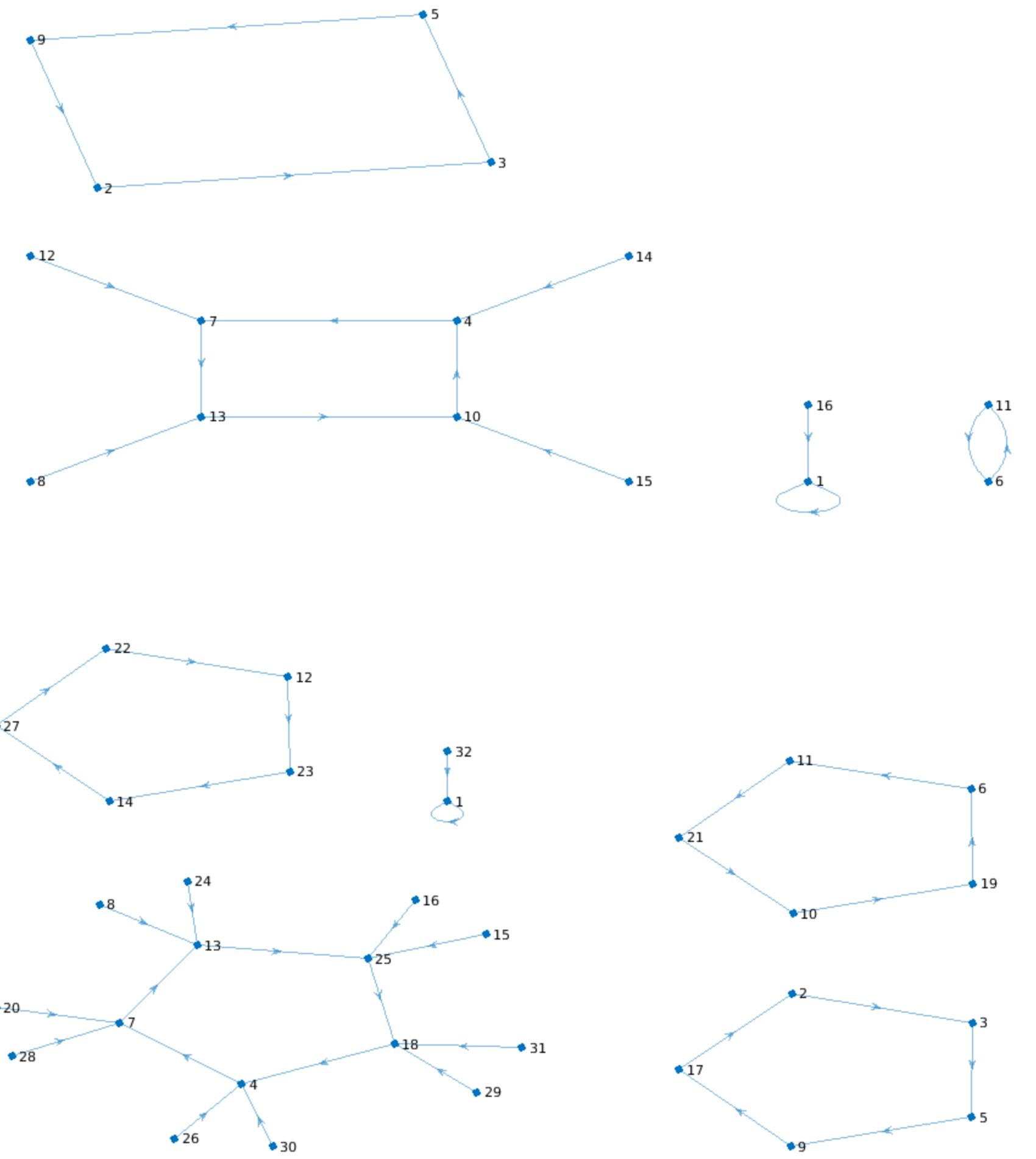


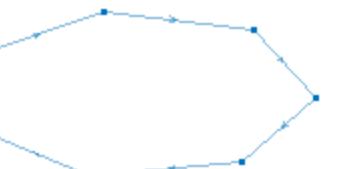
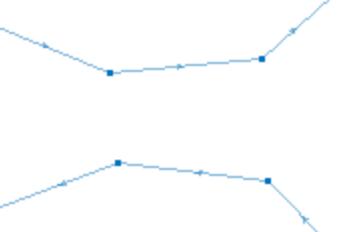
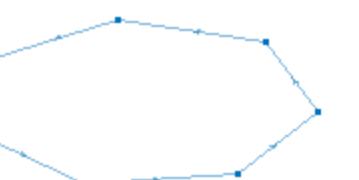
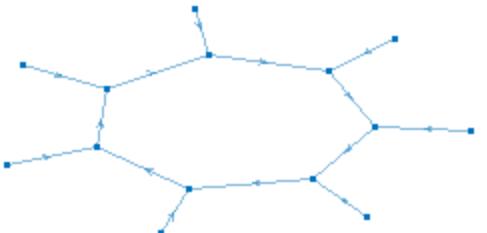
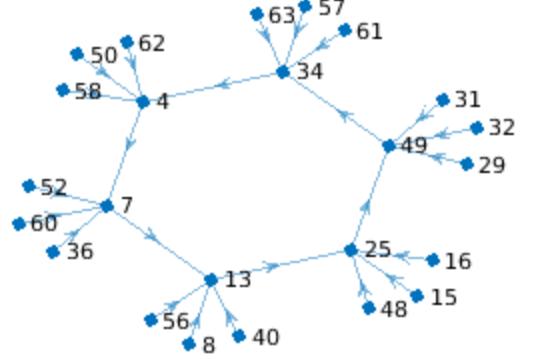
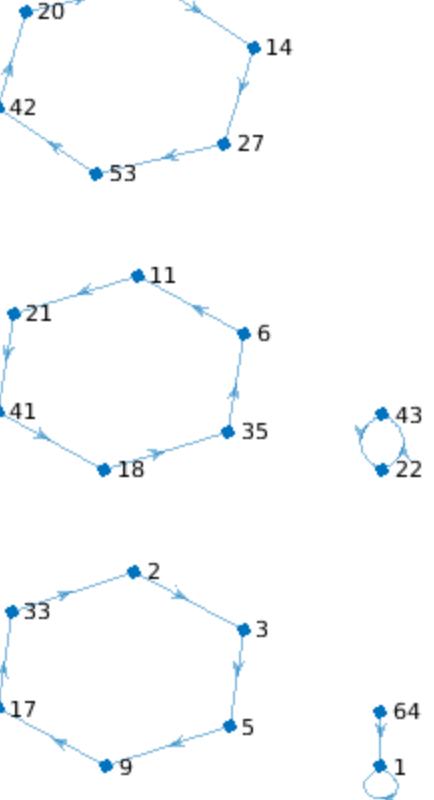
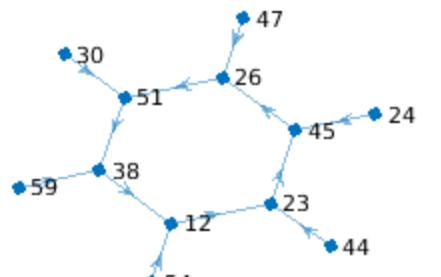


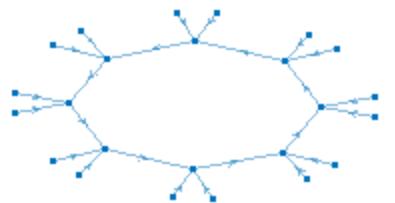


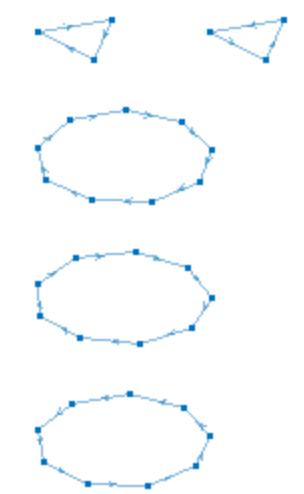
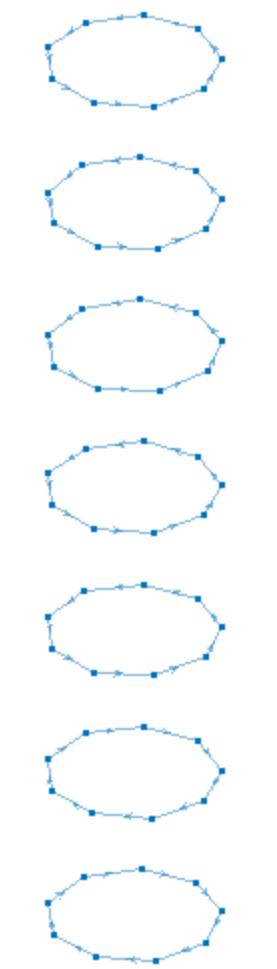
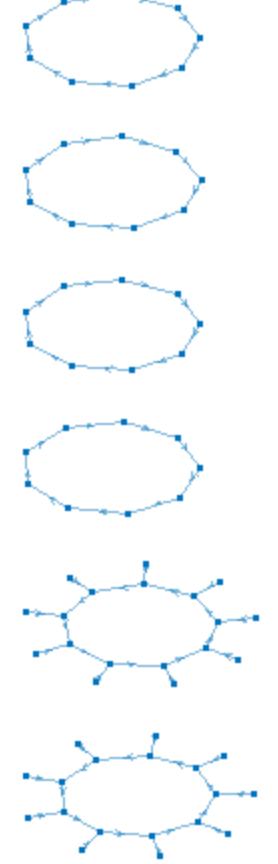
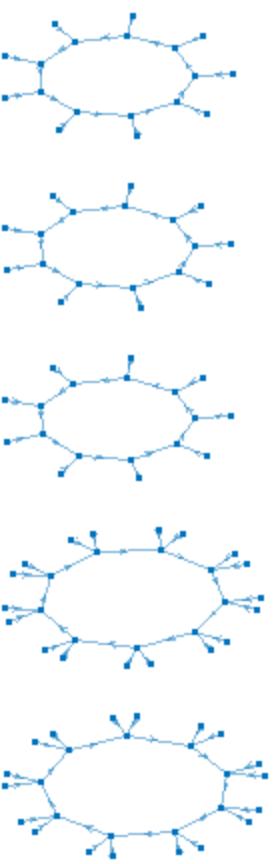
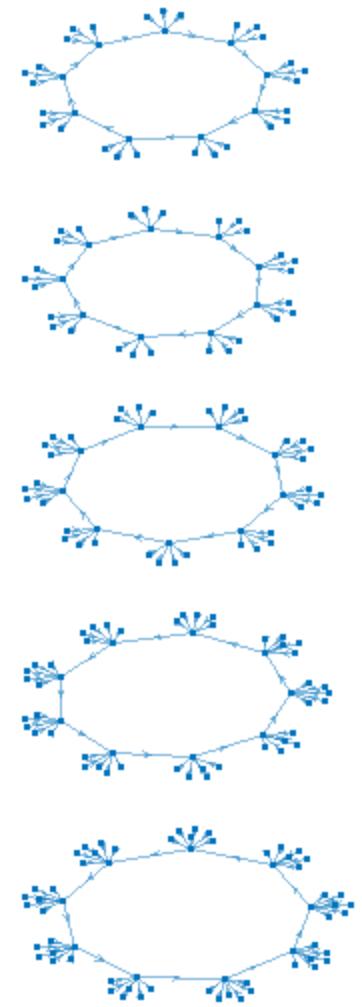


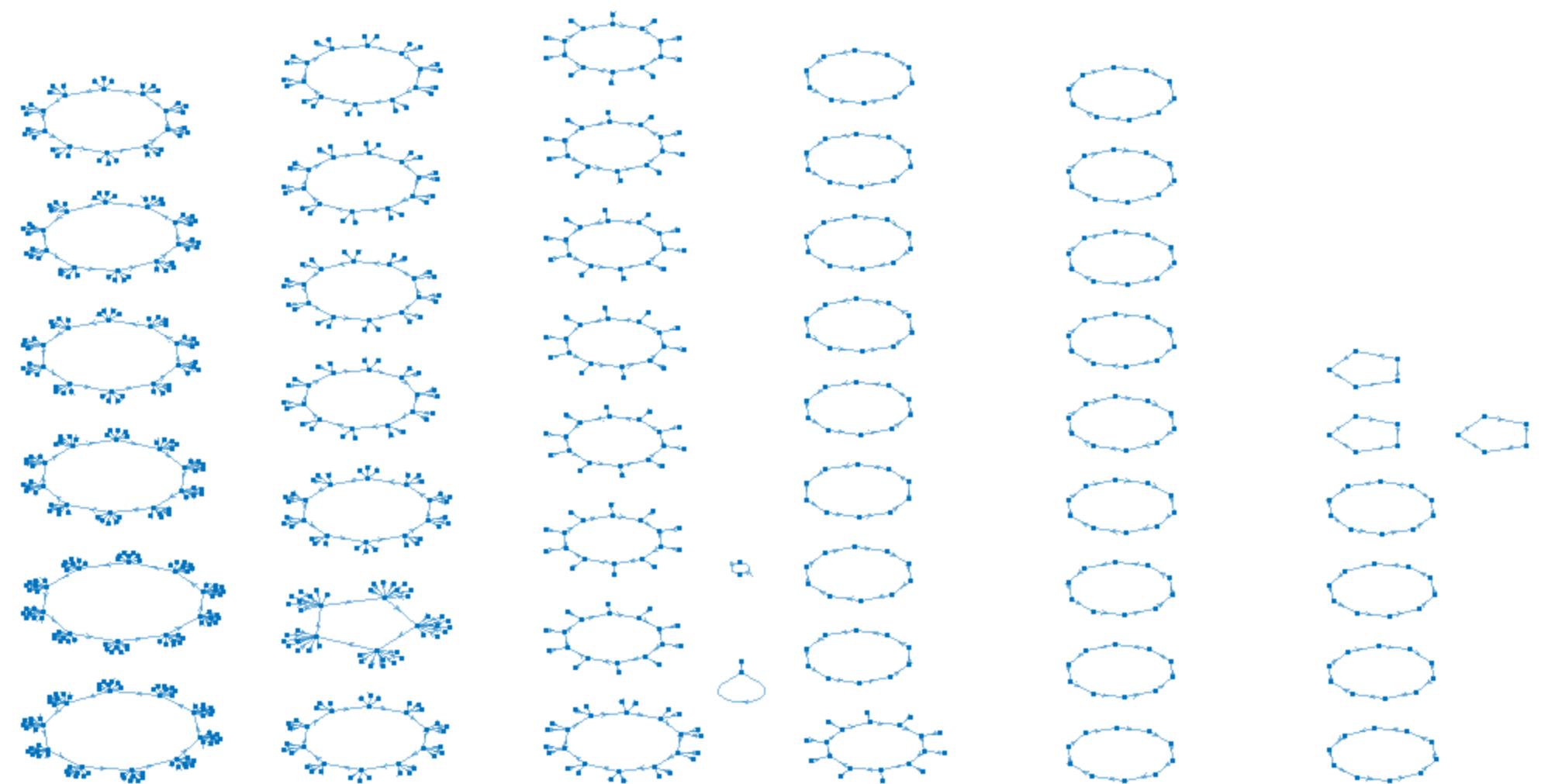


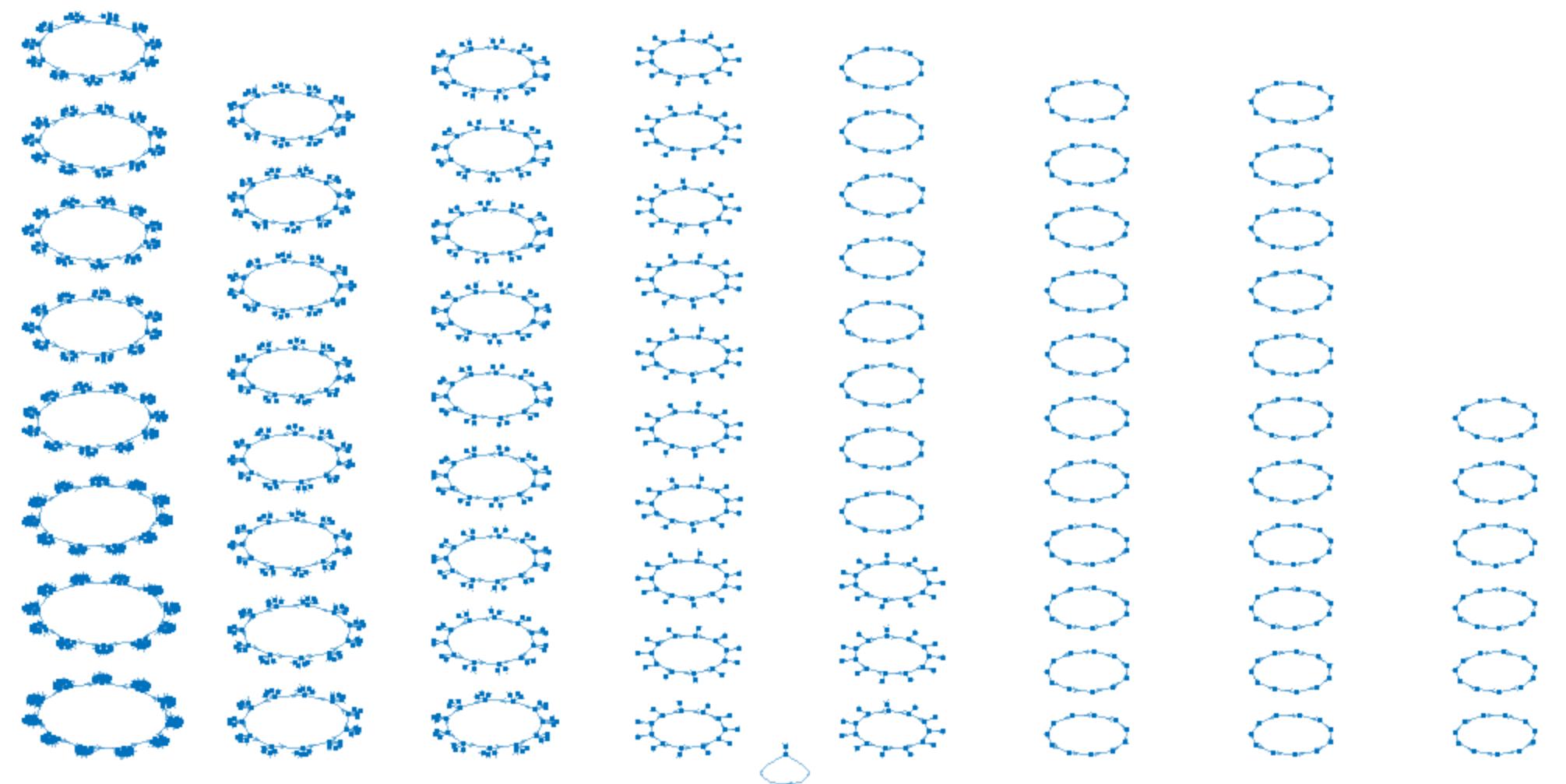


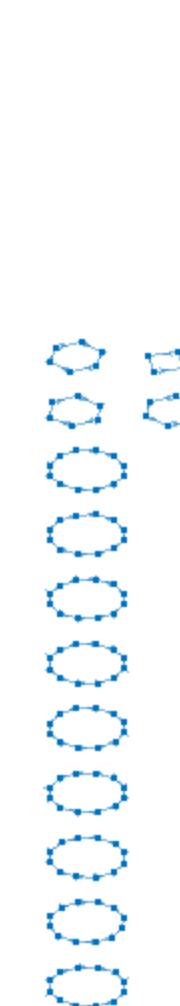
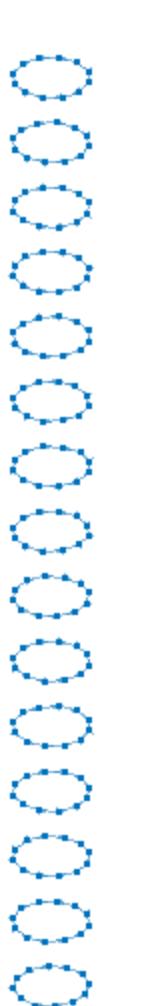
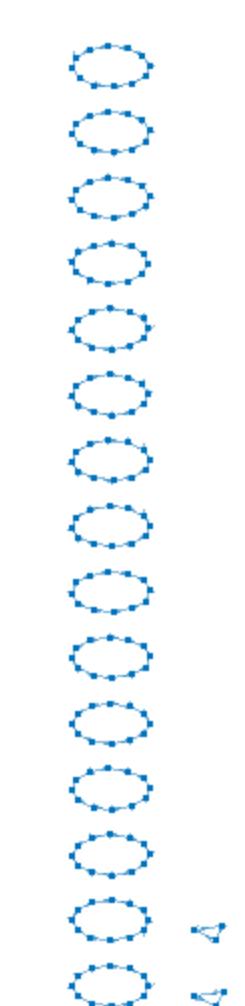
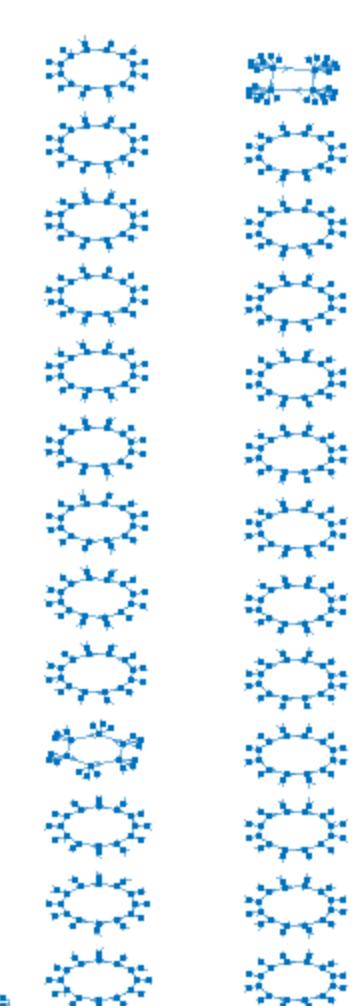
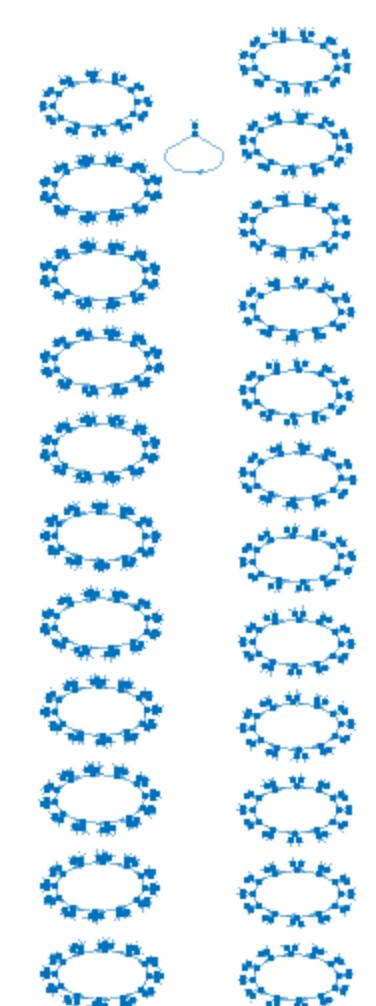
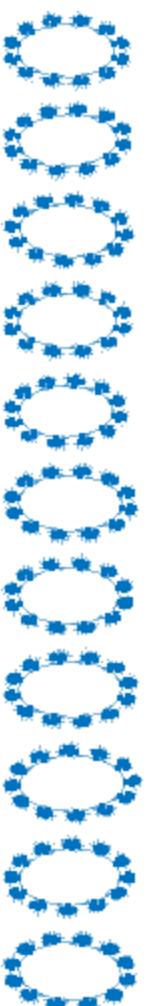
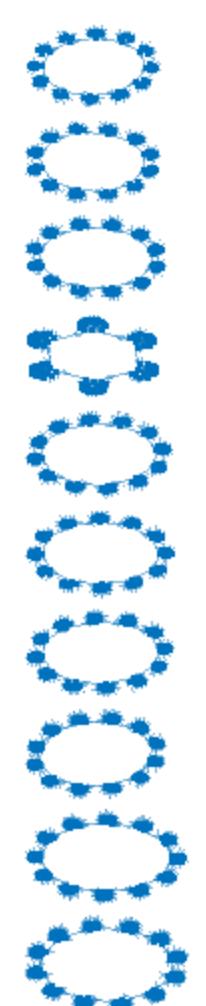


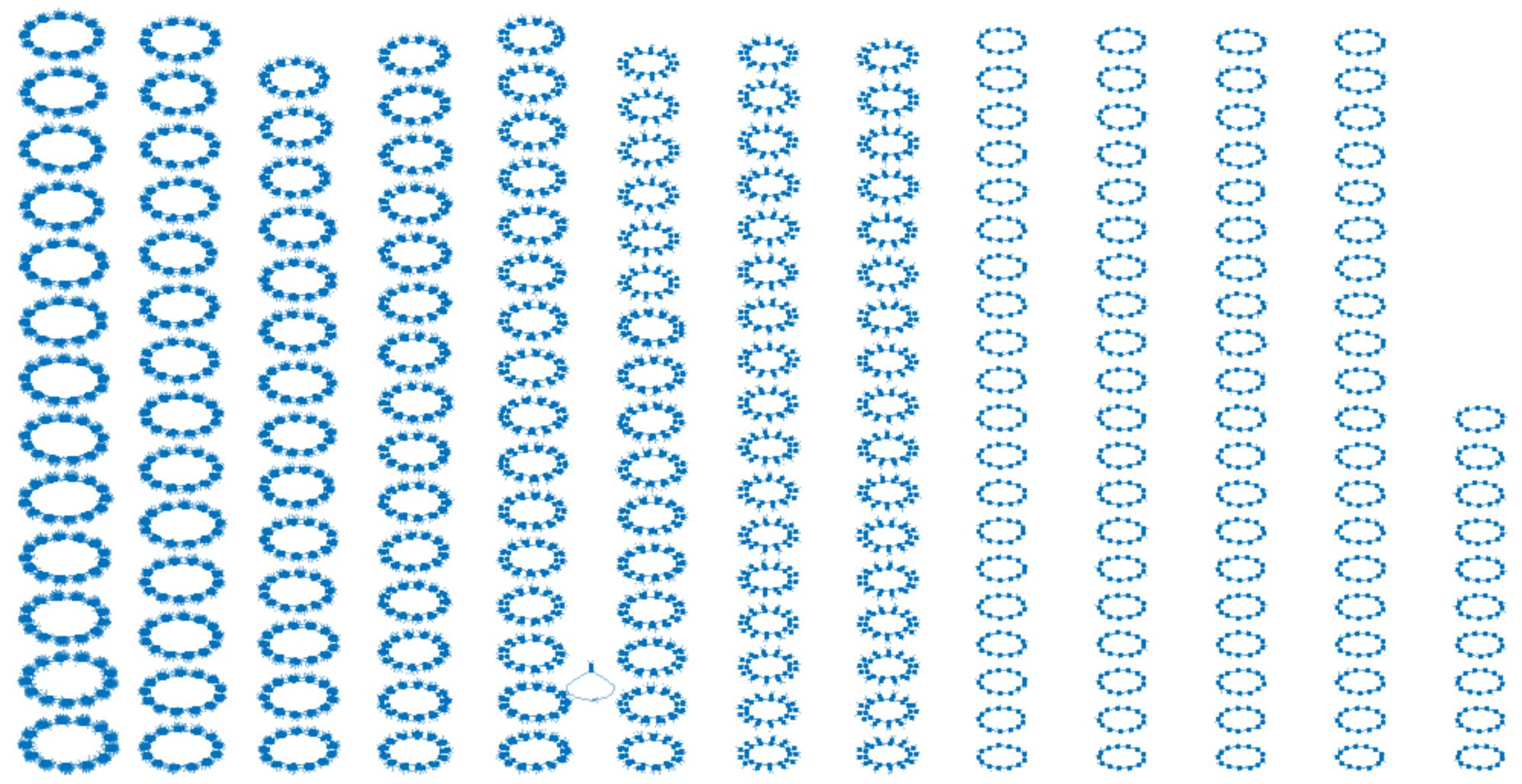


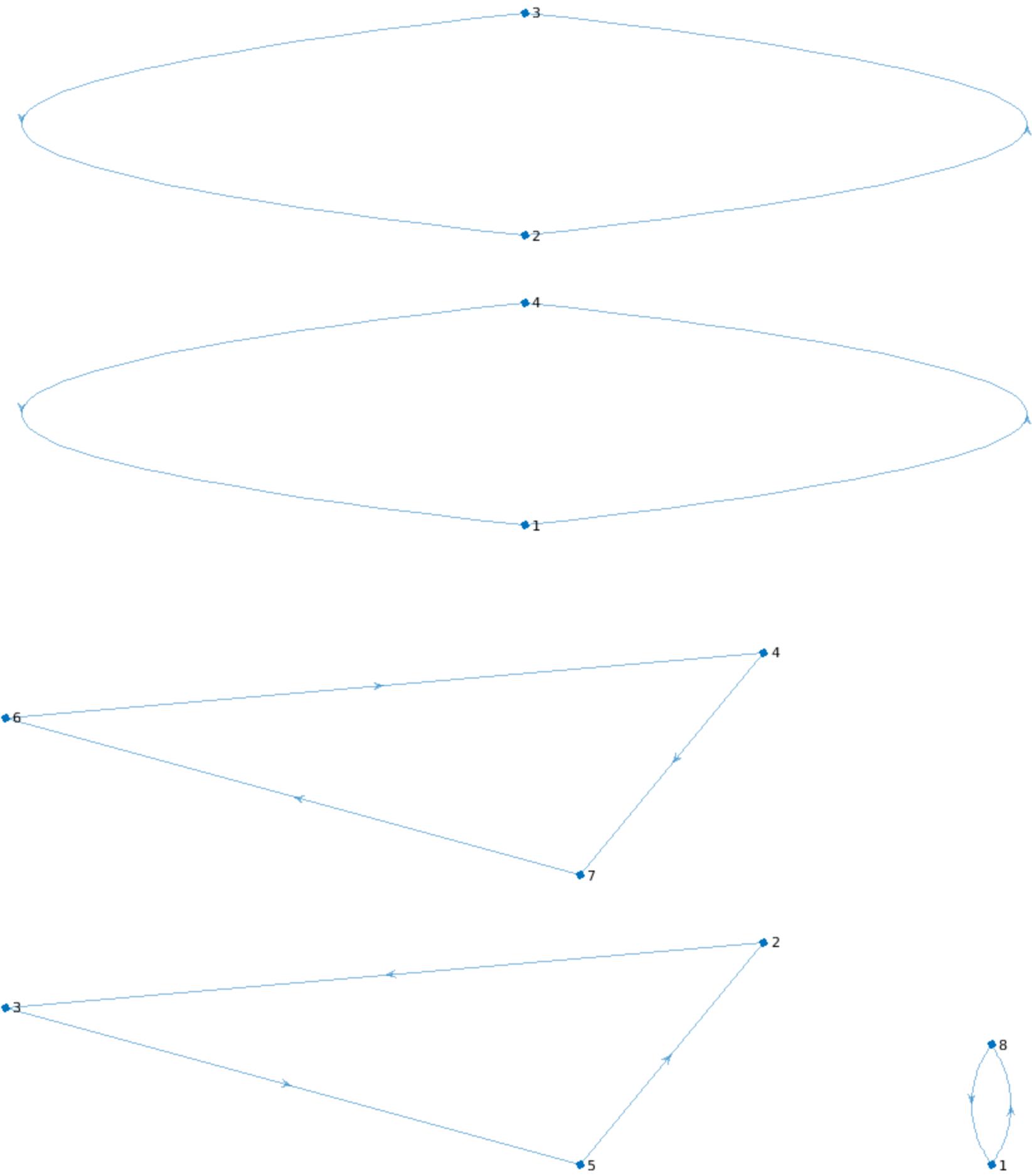


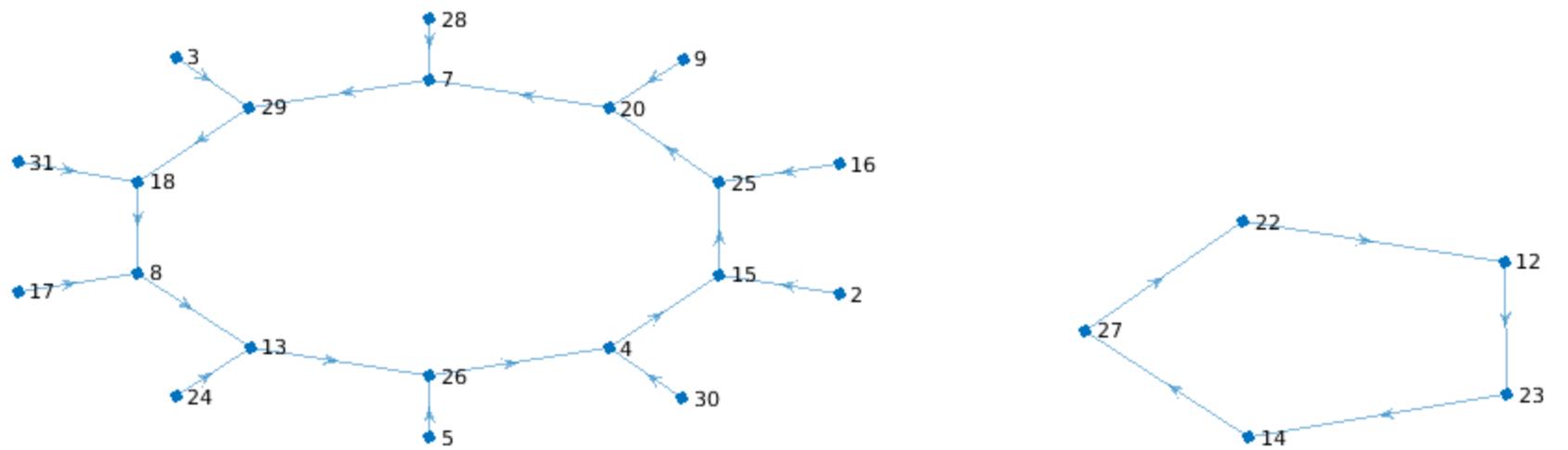
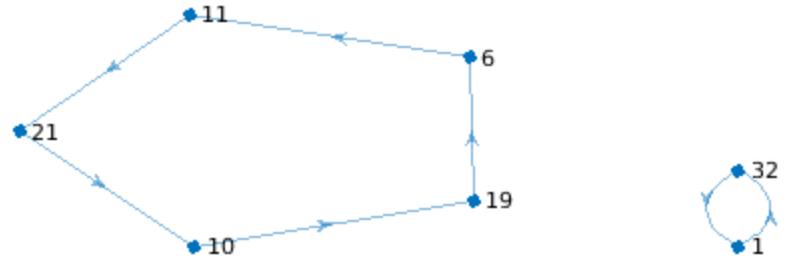
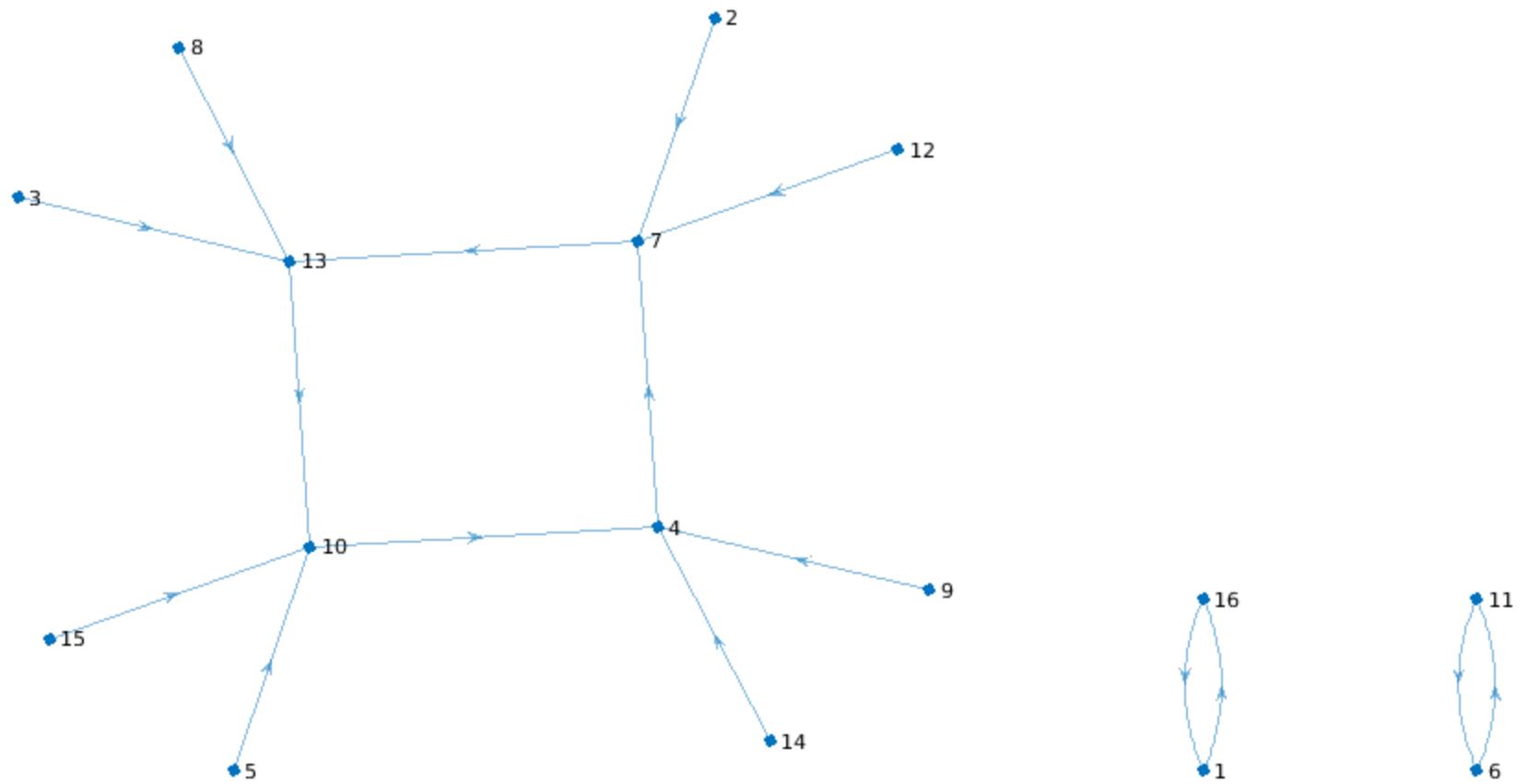


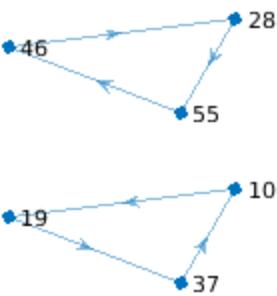
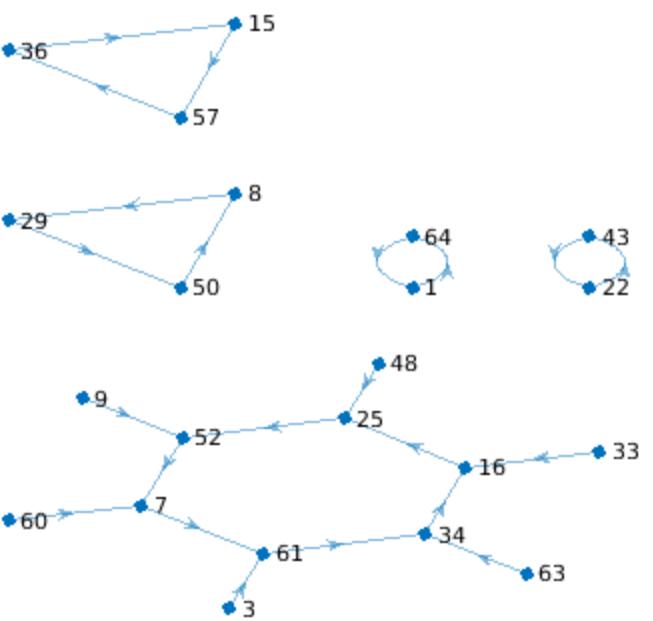
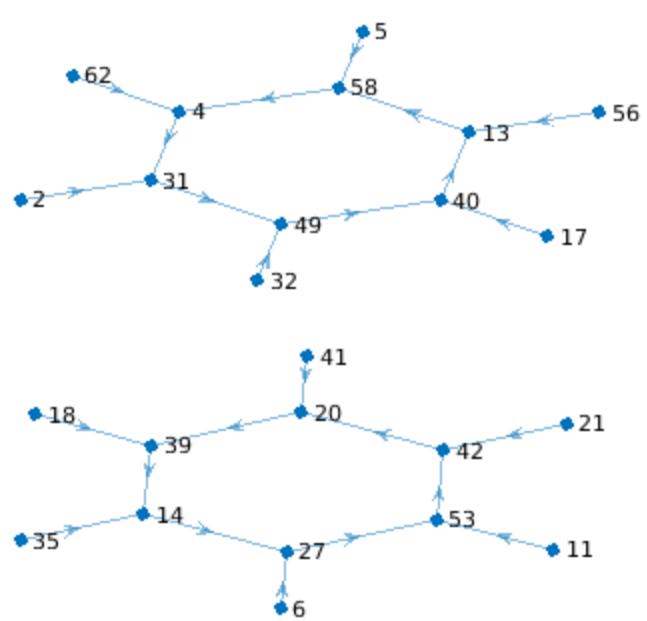
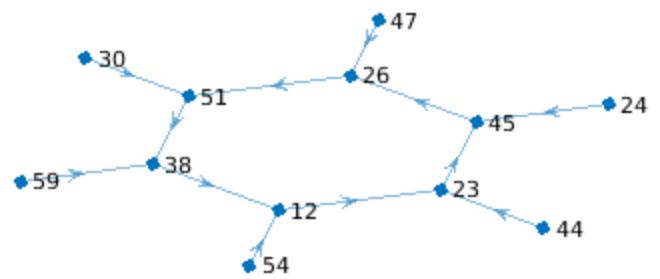


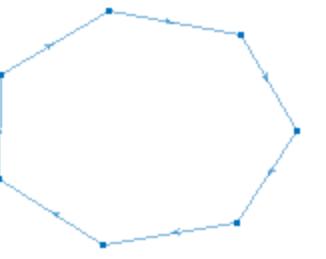
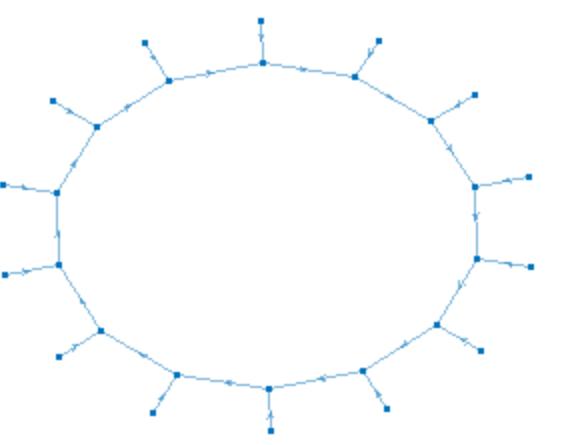
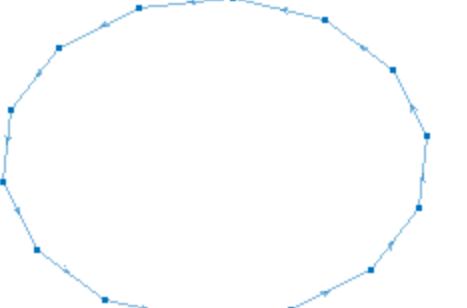
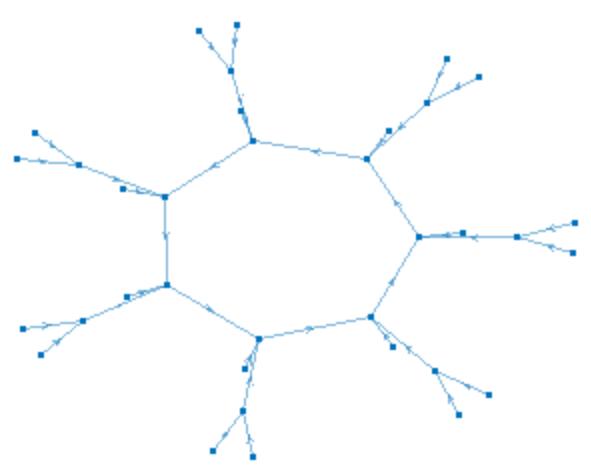
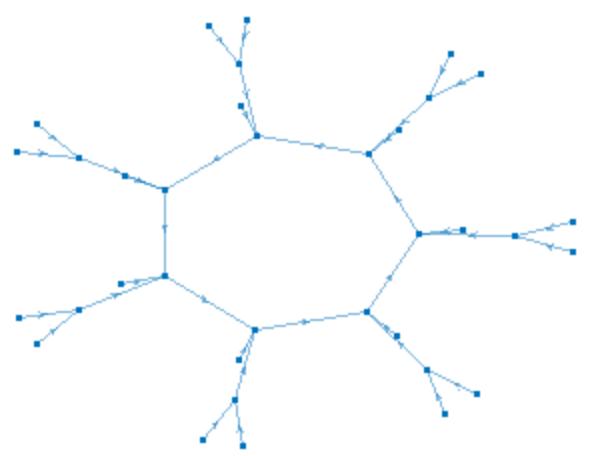


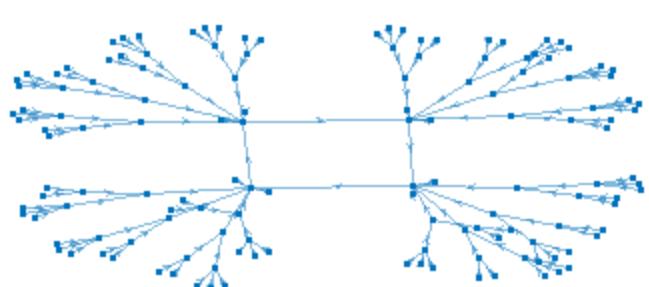
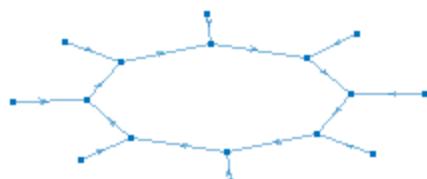


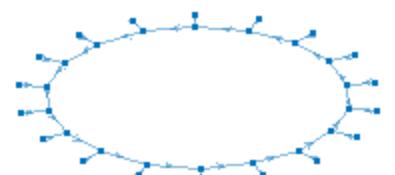












o

