

# model

Author: Daniel ([danielw10001@gmail.com](mailto:danielw10001@gmail.com)), BUAA Dept.6 SCSE

- model
  - 1. 模型
    - 1.1.  $P(N)$ 严格单增
    - 1.2.  $P(N)$ 严格单减
    - 1.3.  $P(N)$ 周期变化
  - 2. 求解

## 1. 模型

一个描述对异性的主动程度与不孤独终老的概率之间关系的模型. 使用余生尝试与异性有效认识的次数来描述对异性的主动程度.

注意到对异性的主动程度对每一次尝试中两人合适成为恋人的概率的影响.

*Let* :  $N$  = 余生尝试与异性有效认识的次数  $\in N_{\text{自然数}}$

*Let* :  $P(N)$  = 余生尝试与异性有效认识  $N$  次时, 每一次尝试中两人合适成为恋人的概率

*maximize* :  $F(N) = (1 - (1 - P(N))^N) \equiv$  不孤独终老的可能性

以下针对 $P(N)$ 严格单增, 严格单减, 周期变化三类典型情况的典型值分别做讨论.

### 1.1. $P(N)$ 严格单增

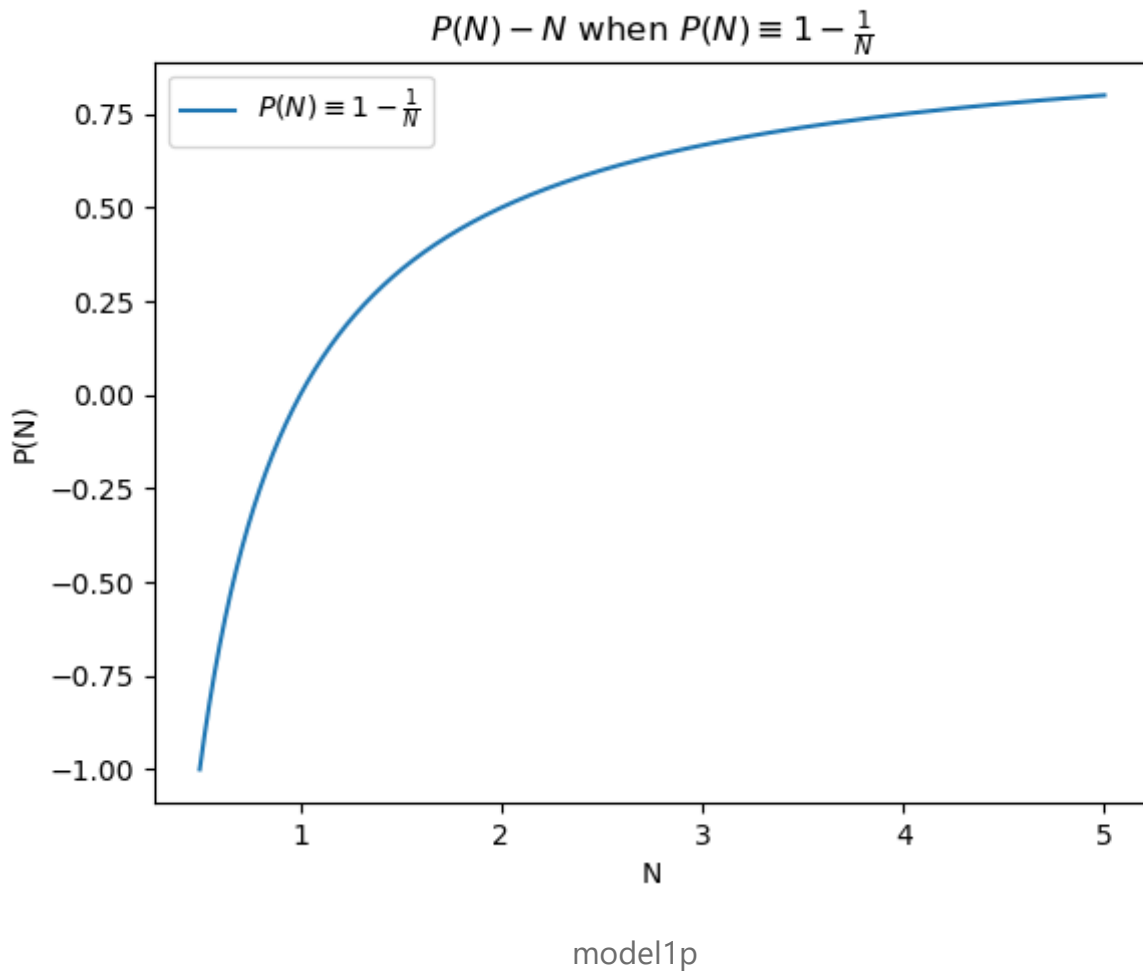
若 $P(N) \equiv 1 - \frac{1}{N}$  (严格单增), 则 $F(N) \equiv 1 - \frac{1}{N^N}$ , 绘制 $P(N)$ 与 $F(N)$ 图像:

```
1  # PRJDIR/src/modelplot1p.py
2
3  import numpy
4  import matplotlib.pyplot
5
6  NFROM = 0.5
7  NTO = 5
8  NSTEP = 0.0001
9
10 nlist = numpy.arange(NFROM, NTO, NSTEP)
11 plist = [(1-1/n) for n in nlist]
12
13 matplotlib.pyplot.title(r"$P(N)$-N$ when $P(N) \equiv 1 - \frac{1}{N}$")
```

```

14 matplotlib.pyplot.xlabel(r"N")
15 matplotlib.pyplot.ylabel(r"P(N)")
16 matplotlib.pyplot.plot(nlist, plist,
17                         label=r"$P(N) \equiv 1 - \frac{1}{N}$")
18 matplotlib.pyplot.legend()
19
20 matplotlib.pyplot.show()

```



```

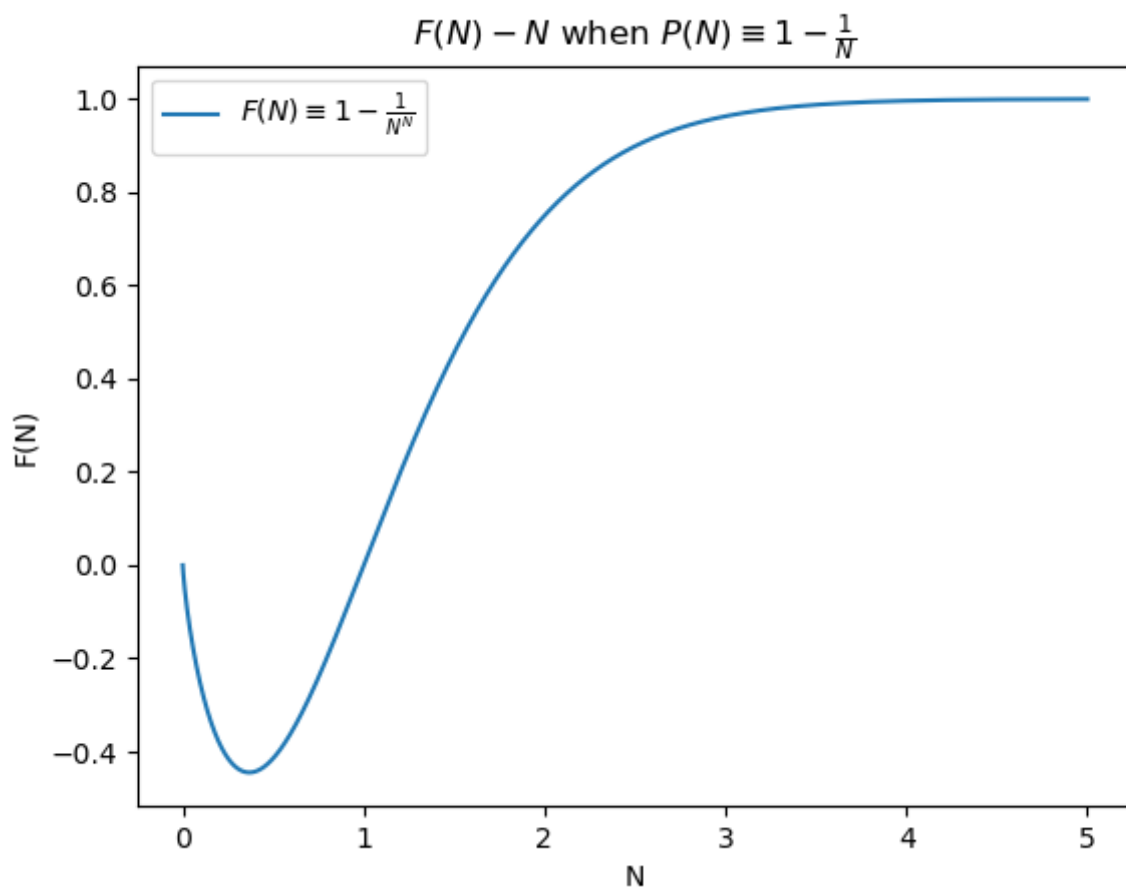
1 # PRJDIR/src/modelplot1f.py
2
3 import numpy
4 import matplotlib.pyplot
5
6 NFROM = 0
7 NTO = 5
8 NSTEP = 0.0001
9
10 nlist = numpy.arange(NFROM, NTO, NSTEP)

```

```

11 flist = [(1-1/n**n) for n in nlist]
12
13 matplotlib.pyplot.title(r"$F(N)-N$ when $P(N) \equiv 1 - \frac{1}{N}$")
14 matplotlib.pyplot.xlabel(r"$N$")
15 matplotlib.pyplot.ylabel(r"$F(N)$")
16 matplotlib.pyplot.plot(nlist, flist,
17                          label=r"$F(N) \equiv 1 - \frac{1}{N^N}$")
18 matplotlib.pyplot.legend()
19
20 matplotlib.pyplot.show()

```



model1f

得最优解  $N_{max} \rightarrow +\infty$

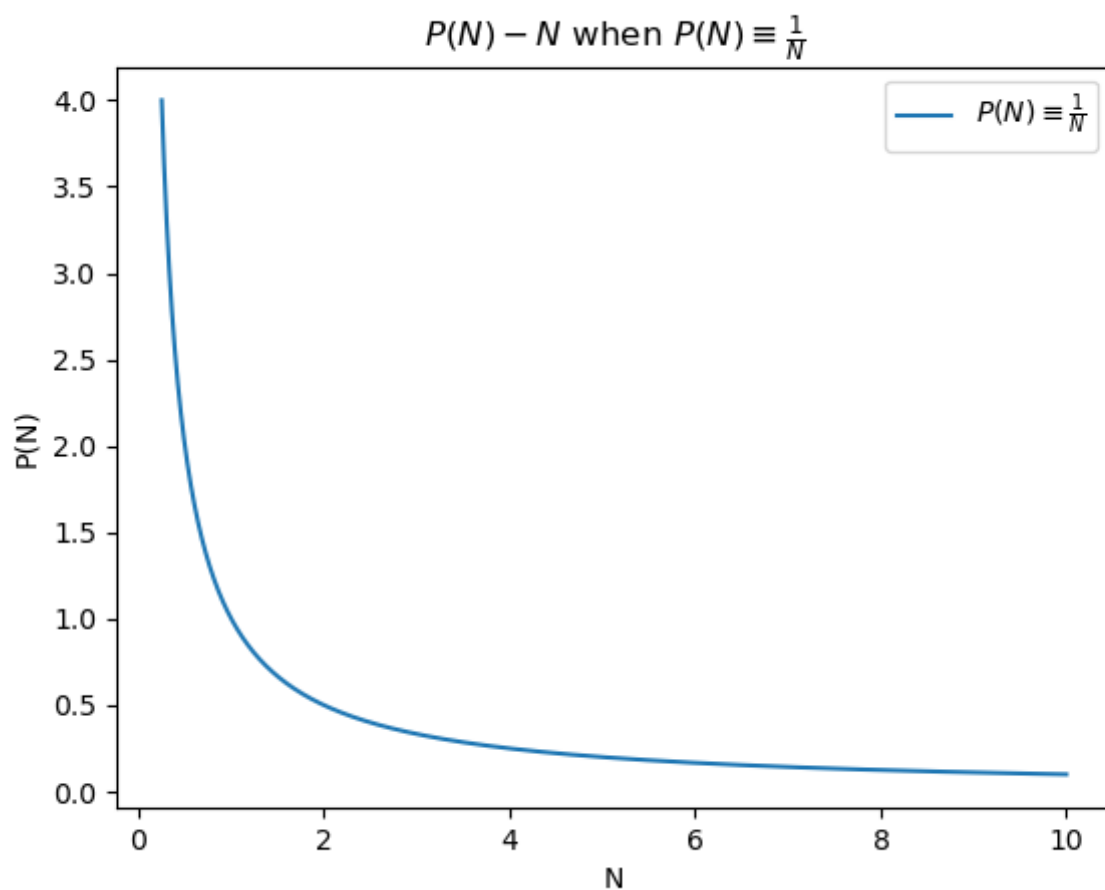
## 1.2. $P(N)$ 严格单减

若  $P(N) \equiv \frac{1}{N}$  (严格单减), 则  $F(N) \equiv 1 - \frac{(N-1)^N}{N^N}$ , 绘制  $P(N)$  与  $F(N)$  图像:

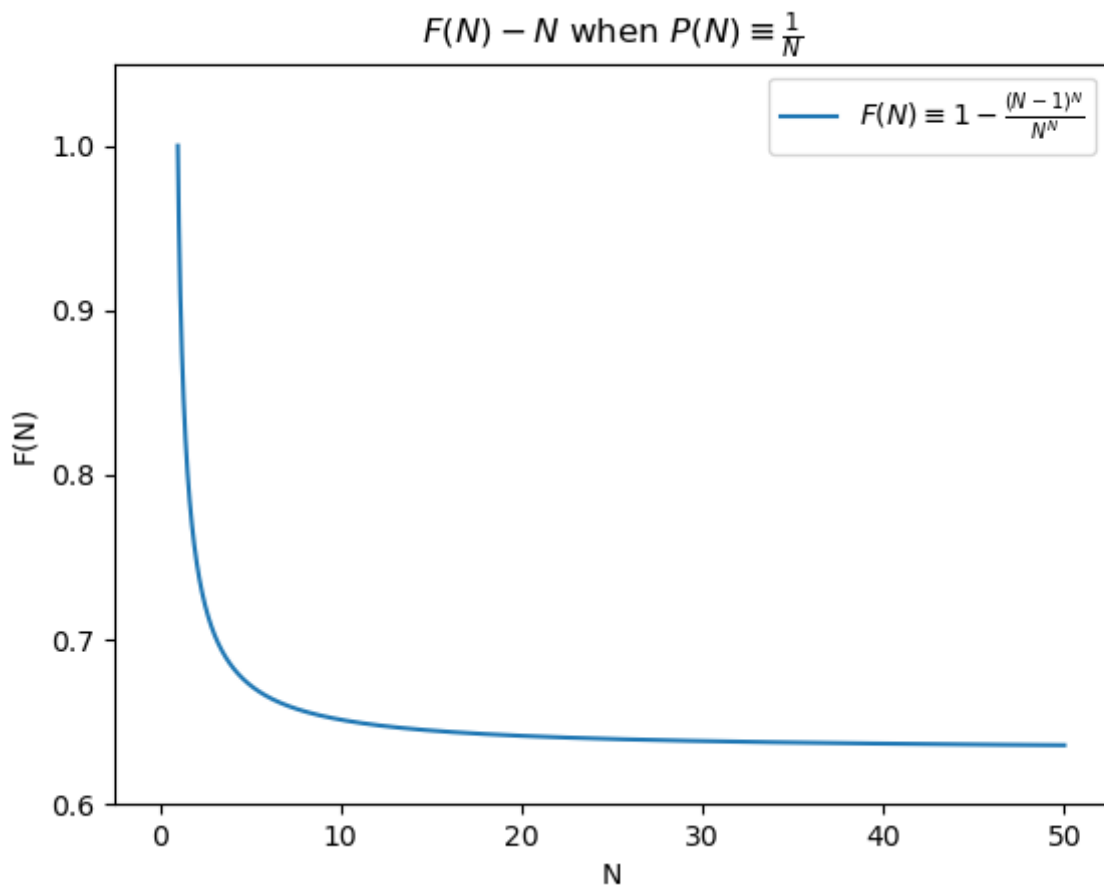
```

1 # PRJDIR/src/modelplot2p.py
2
3 import numpy
4 import matplotlib.pyplot
5
6 NFROM = 0.25
7 NTO = 10
8 NSTEP = 0.0001
9
10 nlist = numpy.arange(NFROM, NTO, NSTEP)
11 plist = [1/n for n in nlist]
12
13 matplotlib.pyplot.title(r"$P(N)-N$ when $P(N) \equiv \frac{1}{N}$")
14 matplotlib.pyplot.xlabel(r"N")
15 matplotlib.pyplot.ylabel(r"P(N)")
16 matplotlib.pyplot.plot(nlist, plist,
17                         label=r"$P(N) \equiv \frac{1}{N}$")
18 matplotlib.pyplot.legend()
19
20 matplotlib.pyplot.show()

```



```
1 # PRJDIR/src/modelplot2f.py
2
3 import numpy
4 import matplotlib.pyplot
5
6 NFROM = 0
7 NTO = 50
8 NSTEP = 0.0001
9
10 nlist = numpy.arange(NFROM, NTO, NSTEP)
11 flist = [(1-((n-1)**n)/(n**n)) for n in nlist]
12
13 matplotlib.pyplot.title(r"$F(N)$-N$ when $P(N) \equiv \frac{1}{N}$")
14 matplotlib.pyplot.xlabel(r"$N$")
15 matplotlib.pyplot.ylabel(r"$F(N)$")
16 matplotlib.pyplot.plot(nlist, flist,
17                        label=r"$F(N) \equiv 1 - \frac{(N-1)^N}{N^N}$")
18
19 matplotlib.pyplot.ylim(0.6, 1.05)
20 matplotlib.pyplot.legend()
21 matplotlib.pyplot.show()
```



得最优解  $N_{max} \equiv 1$

### 1.3. $P(N)$ 周期变化

若  $P(N) \equiv \sin(N)$  (周期变化), 则  $F(N) \equiv 1 - (1 - \sin(N))^N$ , 绘制  $P(N)$  与  $F(N)$  图像:

```

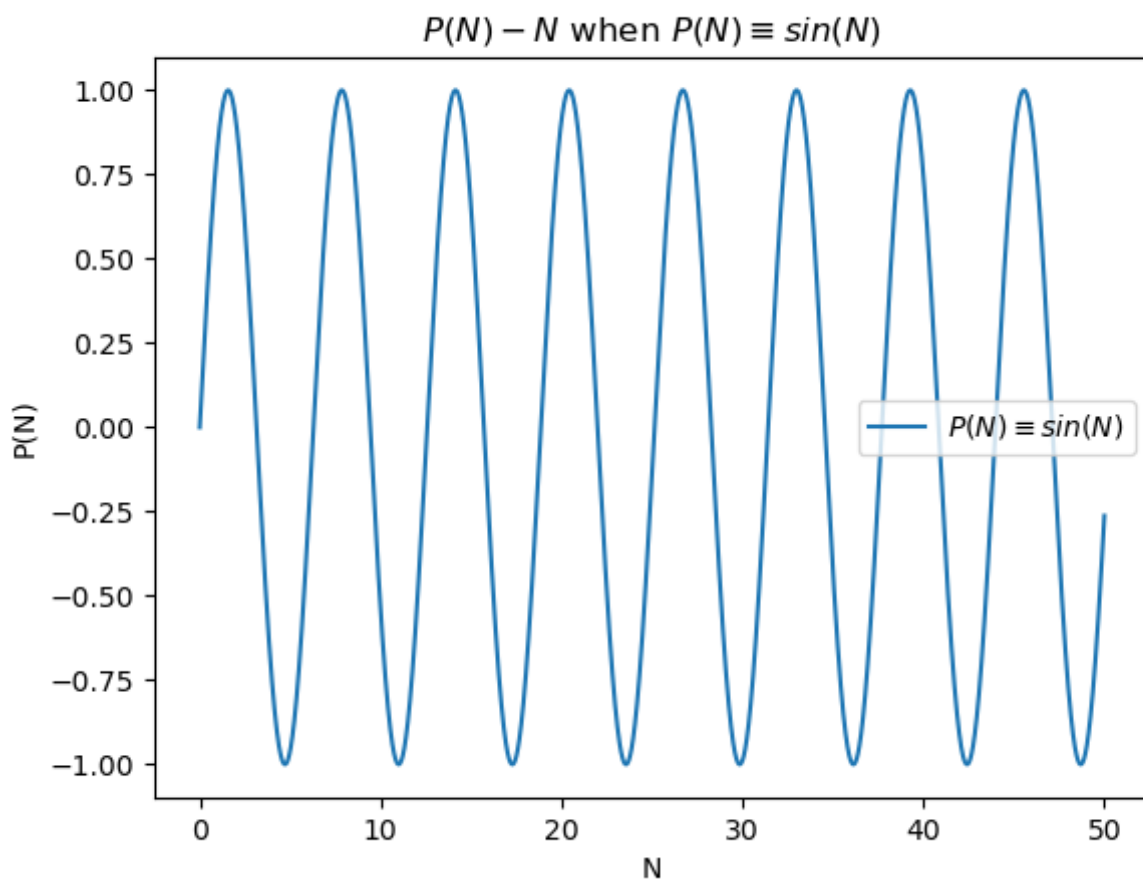
1  # PRJDIR/src/modelplot3p.py
2
3  import numpy
4  import math
5  import matplotlib.pyplot
6
7  NFROM = 0
8  NTO = 50
9  NSTEP = 0.0001
10
11 nlist = numpy.arange(NFROM, NTO, NSTEP)
12 plist = [math.sin(n) for n in nlist]
13

```

```

14 matplotlib.pyplot.title(r"$P(N)-N$ when $P(N) \equiv \sin(N)$")
15 matplotlib.pyplot.xlabel(r"$N$")
16 matplotlib.pyplot.ylabel(r"$P(N)$")
17 matplotlib.pyplot.plot(nlist, plist,
18                         label=r"$P(N) \equiv \sin(N)$")
19 matplotlib.pyplot.legend()
20
21 matplotlib.pyplot.show()

```



model3p

```

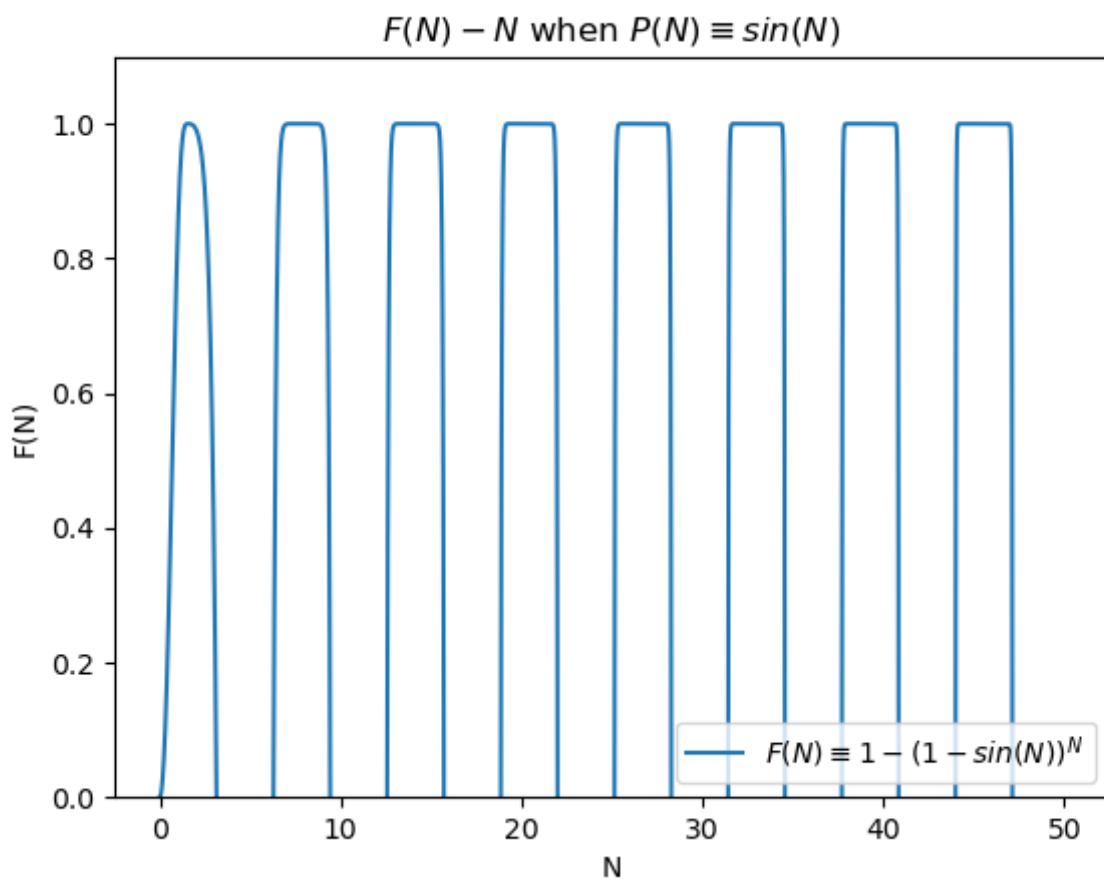
1 # PRJDIR/src/modelplot3f.py
2
3 import numpy
4 import math
5 import matplotlib.pyplot
6
7 NFROM = 0
8 NTO = 50
9 NSTEP = 0.0001

```

```

10
11 nlist = numpy.arange(NFROM, NTO, NSTEP)
12 flist = [(1-(1-math.sin(n))**n) for n in nlist]
13
14 matplotlib.pyplot.title(r"$F(N)-N$ when $P(N) \equiv \sin(N)$")
15 matplotlib.pyplot.xlabel(r"$N$")
16 matplotlib.pyplot.ylabel(r"$F(N)$")
17 matplotlib.pyplot.plot(nlist, flist,
18                         label=r"$F(N) \equiv 1 - (1 - \sin(N))^N$")
19 matplotlib.pyplot.ylim(0, 1.1)
20 matplotlib.pyplot.legend()
21
22 matplotlib.pyplot.show()

```



得最优解  $N_{max} \equiv \frac{\pi}{2} + 2\pi k, k \in \mathbb{Z}$

根据上述讨论, 考虑到  $P(N)$  未知, 典型的最优解包括  $N_{max} \equiv 1, N_{max} \rightarrow +\infty,$   
 $N_{max} \equiv N(k), k \in \mathbb{Z}$



## 2. 求解

最优化 (Mathematical Optimization) 常用的方法包括:

- 解析优化 Analytical Optimization
  - 线性规划 (Linear Programming):
    - 单纯性法 (Simplex Algorithm)
  - 凸优化 (Convex Optimization):
    - 捆集法 (Bundle Adjustment)
    - 次梯度法 (Subgradient Method)
    - 内点法 (Interior Point Method)
    - 切割平面法 (Cutting-Plane Method)
    - Ellipsoid Method
    - Drift-Plus-Penalty Method
- 数值优化 Numerical Optimization
  - 梯度下降法 Gradient Descent
  - 牛顿-拉弗森法 Newton-Raphson Method
  - 演化算法 Evolutionary Algorithm
    - 遗传算法 Genetic Algorithm

但, 由于模型无法确定化, 暂时无法求解.

即, 由于  $P(N)$  无法确定, 无法采用解析法求解; 由于缺少 (大量) 数据, 无法对  $P(N)$  进行拟合, 也无法采用数值法求解.