游泳池水中余氯浓度分析和控制策略研究

摘要

我是摘要

关键词： 我是关键词

一、 问题的背景与重述

1.1问题背景

我是背景

1.2问题重述

游泳池初始余氯浓度为0.6mg/L，余氯浓度随气温和游泳者人数变化而变化，当余氯浓度下降到0.4mg/L时，需要通过加药设备不断往泳池中泵入消毒剂，使得余氯浓度增高维持在0.4～0.6mg/L。当天气温的变化又与当天的最高最低气温有关，本题需要求解出温度随时间的变化曲线，再根据该曲线和游泳者人数随时间变化求解出当天的游泳池余氯浓度随时间的变化曲线，从而给出最适当的加药时间节点，控制人流量，提高营业收入。

1.2.1 问题一

前提：游泳池室内气温与室外气温相差不大。

条件：给出2022年6月27日最高气温和最低气温，假设平均日长为14小时，平均夜长为10小时。

关键问题：根据最高最低气温建立合理数学模型，寻找气温随时间的变化规律，并预测2022年7月29日和30日游泳馆内各自24小时气温变化曲线。

1.2.2问题二

前提：确在无游泳者情况下，池水中氯的反应速率(mg/(L·h))跟游离态余氯浓度(mg/L)成正比，且受到气温(℃)影响。

条件：当气温控制在（25℃）情况下，余氯从起始浓度0.6mg/L反应降到0.05mg/L 时，统计得平均反应速率V=0.025mg/(L·h)。考虑到气温因素，余氯反应速率跟气温和浓度有如下经验关系式，余氯的反应速率v跟气温T成指数增长关系。已知2022年7月29日9点整初始池水余氯浓度为0.6mg/L。

关键问题：利用给出的经验方程和第一问得到的气温曲线，获得第一次余氯浓度下降到0.4mg/L的时刻，并完整描绘出7月29日一整天池水中余氯浓度随时间变化曲线。

1.2.3问题三

前提：电源模式为“电源一直开启”且能够随时满足用户的热水需求。

条件：给出杭州市此用户家庭夏季、冬季代表性一天的室内温度变化，以及上一问求解的热水器温度变化模型。

关键问题：求出在“电源一直开启”的模式下，找到一个最佳的设定温度（日常恒定温度），既能满足一个人随时洗澡，又能使电量消耗最小。

二、 模型的假设

假设1.。

假设2.。

假设3.。

三、符号说明

|  |  |  |
| --- | --- | --- |
| 符号 | 单位 | 符号说明 |
| v | mg/(L·h) | 余氯反应速率 |
| y | mg/L | 余氯浓度 |
| T0 | ℃ | 上一天的残余温度 |
| Ta | ℃ | 温度振幅 |
| ω1 | h | 太阳能量输入时长 |
| ω2 | h | 太阳能量输入时长 |
| T1 | ℃ | 白天温度上升过程温度 |
| T2 | ℃ | 白天温度下降过程温度 |
| T3 | ℃ | 夜晚温度 |
| t | hh:mm | 一天中的时间 |
| ts | hh:mm | 温度开始衰减的时间 |
| tm | hh:mm | 温度达到最高点的时间 |
| Tsr | hh:mm | 日出时间 |
| Tss | hh:mm | 日落时间 |
| k |  | 衰减常数 |

四、问题分析

4.1问题一分析

4.2问题二分析

4.3问题三分析

针对问题三，问题三要求电源一直开启，此时会面临温度降低导致的加热频繁问题，每次加热都需要消耗电量，如何寻找一个合适的设定温度，使得加热次数减少，成为关键问题，分析温度降低的因素可以知道，在不洗澡时，影响温度降低的因素主要来自于热水器壁体向外散热，而该散热量与内外温差成正比，所以设定温度越低，散热能力越弱，保温效果越好，进而间隔两次加热的时间也能加长，一天内加热所消耗的电量越少，但同时要保证一个人可以随时洗澡，就要保证设定温度低于五度时洗澡，也能洗15分钟，最终热水器中温度降低到夏天37℃，冬天42℃这两个临界条件。综上所述，只需要求得的设定温度低于5摄氏度时开始洗澡，洗澡后温度降到临界条件时的设定温度即为最优的设定温度（日常恒定温度）。

4.4问题四分析

1. 问题三的模型建立与求解

7.1 洗澡过程热水器温度变化与耗电量

由第二问所得“电源一直开启”模式下水温变化方程

此方程描述了水温变化与室温、加热时间之间的关系。求解在不同室温Te的情况下，水温由预设温度低5摄氏度时开始洗澡，保证出水时间在15分钟的情况下，在洗澡结束后温度正好降为37摄氏度时的设定温度，即为最优的预设温度。

7.2 日常恒定过程热水器温度变化与耗电量

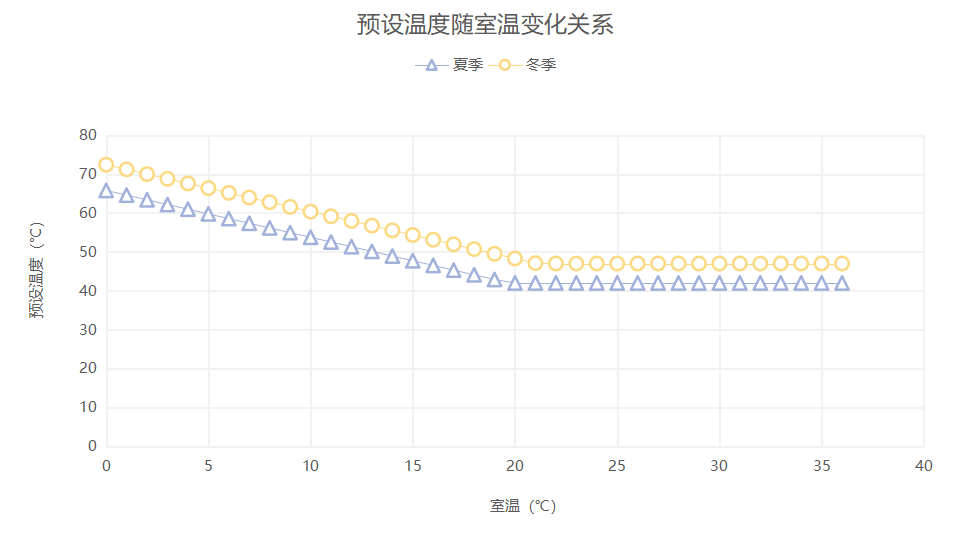
由第一问可得，耗电加热时间随水温变化方程为：

由第二问可得，降温时间随水温变化的方程：

通过计算在不同室温Ts情况下，由预设温度降低5摄氏度所需要的时间，即可求得每天所需的加热次数，进而得到每天加热所需的耗电量。

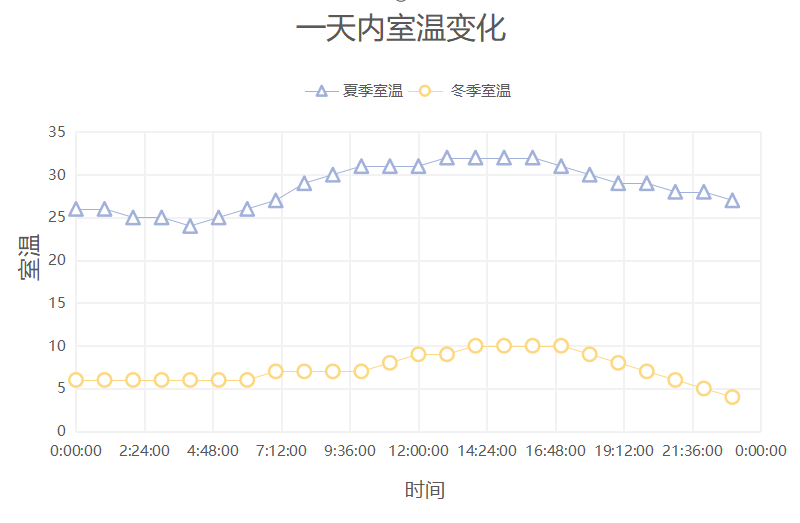
7.3 模型求解

采用向后差分的方法来求解该微分方程，求得结果如下：



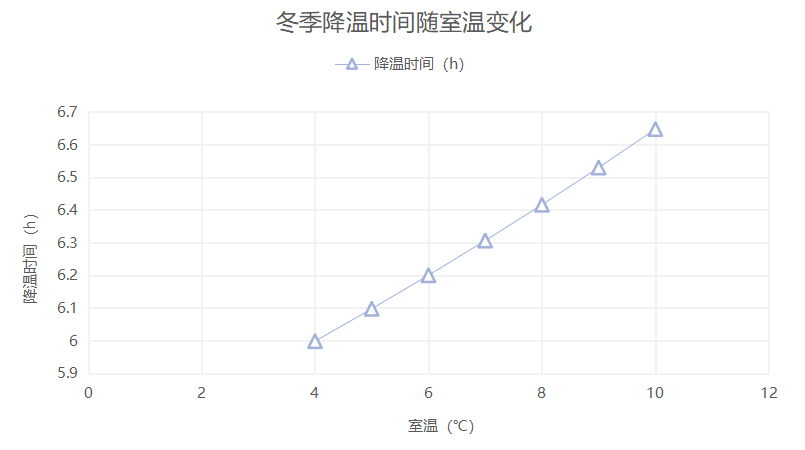
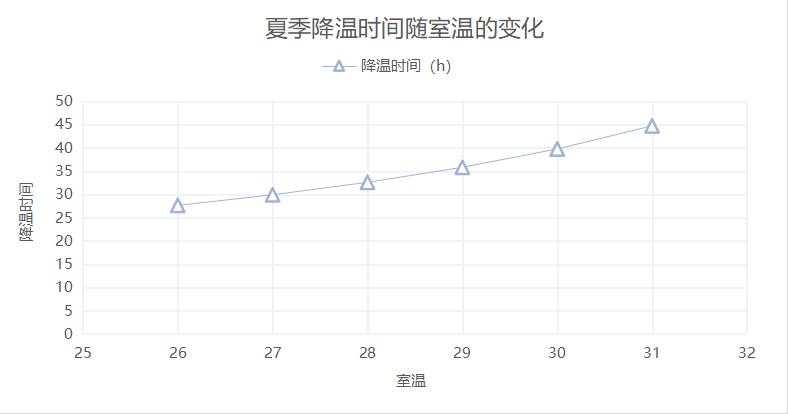
从图中可以看出，不同的室温下，最低可以洗澡的预设温度有变化且随着室温的逐渐升高而降低，最终降到洗澡水温加五摄氏度。

一天之内的室温变化如下图：

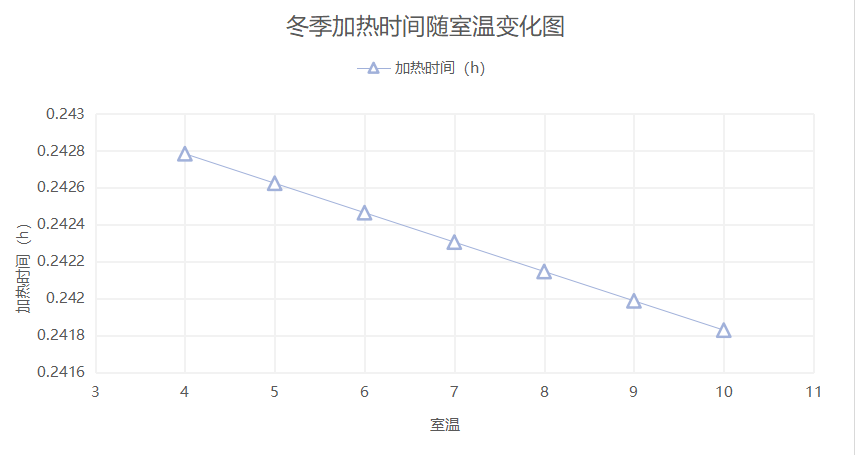


由于需要保证一天之内随时可以洗澡，则说明需要在最低室温的情况下，也能完成洗澡，即一天之内的预设温度需要在最低室温的情况下，求解预设温度。

结合上面两个图表数据可得，在夏季时，一天内最低温度为26℃，对应的最低预设温度为42℃，冬天的最低温度为4℃，对应的最低预设温度为67.59℃。



上面两图为冬夏两季，在分别满足各自最低预设温度的情况下，一天之内降温时间随温度变化的曲线，由图中数据可得，在夏季最低预设温度（42℃）的情况下，降温时间都超过了一天，所以在夏天时，只需要每天洗澡时加热一次即可满足一天24小时都在保温温度范围内。但在冬天时，在预设温度为最低预设温度（67.59℃）的情况下，一天之内降温时间在6h到7h之间，这就意味着一天至少需要加热四次。

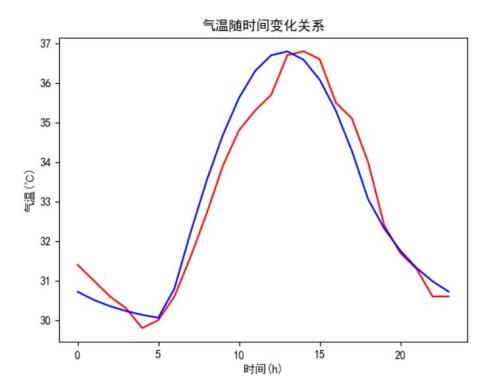


从上图可以看出，冬季加热时间随室温变化差别不大，基本维持在0.24h。一天保温所耗费的电量大约为1.5KW\*4\*0.25h=1.44KW\*h。

九、模型的评价与改进

**9.1问题一**

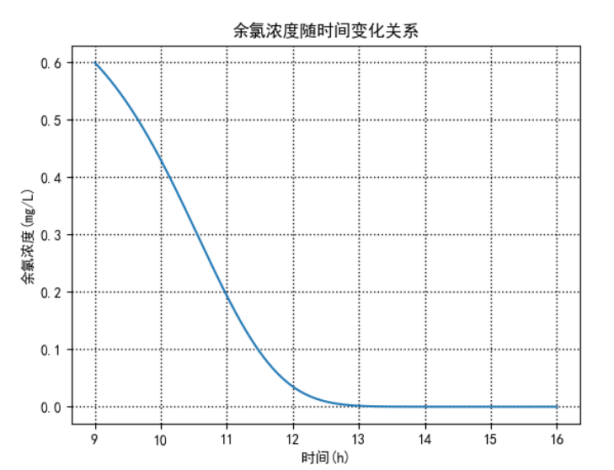
在问题一中，建立模型结果能通过很少的数据，较为准确的得出一天的温度，我们仅在最高最低温度之上，引进日出日落时间，来估算在高温出现的时间，进而使得模型更加准确。如图所示。



此外该模型适用于一年中的大多数时间。除了极端情况外，都能有良好的效果。但是始终与真实情况有一定的差距，但这都是在允许范围内，综上我们认为模型一表现优异。

**9.2问题二**

问题二中，通过第一部分问题确定出温度与余氯浓度之间的数学关系，再通过第二问的经验模型，结合第一问的时间温度关系，建立出温度余氯弄得的模型。通过化学动力学方程，有限差分等方法，不断优化模型结果，得出模型结果，通过对比模型结果与实际情况，可以认为模型表现良好。



曲线大体与经验公式保持一致走势，可以认为模型正确，通过不断的进行调参，优化，再次基础上得出结果。

**9.3 问题三**

问题三中，引入了泳池人数对于余氯浓度的影响，需要在问题二的基础上建立一个包含泳池人数的模型关系，我们通过拟合构造出人数与浓度的关系式。模型能够完整的预测出某天余氯浓度与时间的关系，通过设置边界条件，即可求出加药时间点。

**9.4 问题四**

问题四，在问题三的基础，通过拟合构造出，加药后余氯浓度与时间的关系，能够得出一般情况下，泳池的余氯浓度情况。模型符合实际，具有较好的前景。但是不足之处在于我们假定人数在一个小时固定，但是实际中人数是随时动态变化。

参考文献

[1] 孟翔晨,刘昊,程洁.基于FY-2F数据的中国区域地表温度日变化模型评价及特征研究[J].遥感学报,2019,23(04):570-581.

[2] Frank-M Göttsche, Folke S Olesen,Modelling of diurnal cycles of brightness temperature extracted from METEOSAT data,Remote Sensing of Environment,Volume 76,Issue3,2001,Pages337-348,ISSN0034-4257,https://doi.org/10.1016/S0034-4257(00)00214-5.

[3] Van Den Bergh F, Van Wyk M A and Van Wyk B J. 2006. Comparison of data-driven and model-driven approaches to brightness temperature diurnal cycle interpolation//Proceedings of the 17th Annual Symposium of the Pattern Recognition Association of South Africa. Parys, South Africa: [s.n.]

附录

模型代码

import numpy as np

import matplotlib.pyplot as plt

import math

from pylab import mpl

mpl.rcParams['font.sans-serif'] = ['SimHei']

tan=math.tan

cos=math.cos

arctan=math.atan

pi=np.pi

e=math.e

ts=18

h=0.001

k=-0.11295

t0=9

tn=24

y=np.ones(24000)

y[0]=0.6

N=(24-t0)/h

T0=27.182

Ta=36.72-27.182

tm=(19.04+4.96)/2+3

w=14

m=4

PeopleNum=[0,10,20,40,60,80,100,120,140,160,180,200,220,240,260,280,300,320,340,360,380,400,420,440,460,480,500]

Swpepple=[80,190,0,220,340,0,280,380,0,320,480,520]

PeopleTest=[520,520,0,520,520,0,520,520,0,520,520,520]

km=(w/pi)\*(arctan(pi/w\*(ts-tm)))

Time=[]

concent=[]

concent.append(0.6)

Tem=[]

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t0 - tm)))

# 第四问前两个小时

for i in range(1, 2000, 1):

t = h \* i

pnum=520

ploss=0.0004\*pnum+0.0015

addmed=1.4754

y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h + addmed \* h

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)))

if y[i]>0.6:

y[i]=0.6

concent.append(y[i])

# else:

# y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h

# Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)))

# concent.append(y[i])

y[0]=0.6

concent.append(0.6)

# \*\*\*\*\*\*#

# 十一点到十二点

t1=11

for i in range(1, 1000, 1):

t=h\*i

pnum=520

ploss=0.0004\*pnum+0.0015

addmed=1.4754

y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t1 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h+addmed\*h

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t1 + t - tm)))

if y[i] > 0.6:

y[i] = 0.6

concent.append(y[i])

y[0]=0.6

concent.append(0.6)

p=[520,450]

# 十二点到两点

t2=12

for i in range(1, 2000, 1):

t=h\*i

if t<=1:

pnum=p[0]

else:

pnum=p[1]

ploss=0.0004\*pnum+0.0015

addmed=1.4754

y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t2 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h+addmed\*h

if(y[i]>0.6):

y[i]=0.6

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t2 + t - tm)))

concent.append(y[i])

concent.append(0.4)

#二点到六点

y[0]=0.4

t2=14

p=[0,520]

for i in range(1, 4000, 1):

t=h\*i

if t<2:

pnum=0

elif t>=2and t<3:

pnum=212

else :

pnum=0

ploss=0.0004\*pnum+0.0015

addmed=1.4754

y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t2 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h+addmed\*h

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t2 + t - tm)))

if y[i]>0.6:

y[i]=0.6

concent.append(y[i])

#六点到九点

concent.append(0.6)

y[0]=0.6

t2=18

for i in range(1, 3000, 1):

t=h\*i

pnum=520

ploss=0.0004\*pnum+0.0015

addmed=1.4754

y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h+addmed\*h

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)))

if y[i]>0.6:

y[i]=0.6

concent.append(y[i])

t=np.arange(9,21,0.001)

plt.plot(t, concent)

plt.grid(color="black", which="both", linestyle=':', linewidth=1)

plt.fill(color='g', alpha=0.3)

plt.title("时间-余氯浓度")

plt.xlabel("时间/h")

plt.ylabel("余氯浓度 mg/l")

plt.show()

import math

import matplotlib

import matplotlib.pyplot as plt

from numpy import \*

from pylab import \*

from pylab import mpl

mpl.rcParams['font.sans-serif'] = ['SimHei']

pi=math.pi

tsr=5.25

tss=18.92

tmax=36.36

tmin=27.36

tmax1=36.72

tmin1=27.18

ts=tss-1

T0=tmin

tm=(tss+tsr)/2+2

Ta=tmax-tmin

Tb=tmax1-tmin1

e=math.e

T=[]

tan=math.tan

cos=math.cos

acos=math.acos

arctan=math.atan

asin=math.asin

w=2/15\*acos(-tan(30.3/360\*2\*pi)\*tan(23.33521955/360\*2\*pi))

w1=14

w2=14

k=(w2/pi)\*(arctan(pi/w2\*(ts-tm)))

for t in arange(0,7):

T.append(T0+Tb\*cos(pi/w2\*(ts-tm))\*pow(e,-(t+24-ts)/k))

for t in arange(7,24):

if t<tm:

T.append(T0+Ta\*cos(pi/w1\*(t-tm)))

elif t>=tm and t<ts:

T.append(T0+Ta\*cos(pi/w2\*(t-tm)))

else:

T.append(T0+Ta\*cos(pi/w2\*(ts-tm))\*pow(e,-(t-ts)/k))

print(T)

x=arange(0,24)

plt.title('温度-时间曲线')

plt.plot(x, T, 'ro-', color='#4169E1', alpha=0.8, linewidth=1)

plt.xlabel('时间')

plt.ylabel('温度')

plt.show()

import numpy as np

import matplotlib.pyplot as plt

import math

from pylab import mpl

mpl.rcParams['font.sans-serif'] = ['SimHei']

tan=math.tan

cos=math.cos

arctan=math.atan

pi=np.pi

e=math.e

ts=18

h=0.001

k=-0.11295

t0=9

tn=24

y=np.ones(24000)

y[0]=0.6

N=(24-t0)/h

T0=27.182

Ta=36.72-27.182

tm=(19.04+4.96)/2+3

w=14

m=4

PeopleNum=[0,10,20,40,60,80,100,120,140,160,180,200,220,240,260,280,300,320,340,360,380,400,420,440,460,480,500]

Swpepple=[80,190,0,220,340,0,280,380,0,320,480,520]

PeopleTest=[520,520,0,520,520,0,520,520,0,520,520,520]

km=(w/pi)\*(arctan(pi/w\*(ts-tm)))

Time=[]

concent=[]

concent.append(0.6)

Tem=[]

Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t0 - tm)))

# 计算加药时间

# for pnum in PeopleNum:

# ploss=pnum\*0.0004+0.0015

# for i in range(1, 5000, 1):

# t = h \* i

# y[i] = y[i - 1] + h \* k \* 10 \*\* (0) \* y[i - 1] - ploss \* h

#

# concent.append(y[i])

# if (y[i] < 0.4):

# Time.append(t0 + t)

# print(t0 + t)

# break

# print(Time)

#第三题第二问

# for i in range(1, 12000, 1):

# t = h \* i

# pnum=Swpepple[i//1000]

# ploss=0.0004\*pnum+0.0015

# if(pnum!=0):

# if (y[i - 1] < 0.4):

# y[i - 1] = 0.6

# Time.append(t0 + t)

# print(t0 + t)

# if(i//1000<9):

# y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)) - m - 25) / 5) \* y[i - 1] - ploss \* h

# Tem.append(T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)))

# concent.append(y[i])

# else:

# y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (ts - tm))\* pow(e, -(t0+t - ts) / km) - m - 25) / 5) \* y[i - 1] - ploss \* h

# Tem.append(T0 + Ta \* math.cos(pi / 14 \* (ts - tm))\* pow(e, -(t0+t - ts) / km))

# concent.append(y[i])

# print(Tem)

# print('Time:',Time)

# t=np.arange(9,21,0.001)

# plt.plot(t, concent)

# plt.grid(color="black", which="both", linestyle=':', linewidth=1)

# plt.fill(color='g', alpha=0.3)

# plt.title("7月30号余氯浓度随时间变化")

# plt.show()

# #没有人数干扰

# for i in range(1,9000,1):

# t = h \* i

# y[i] = y[i - 1] + h \* k \* 10 \*\* ((T0 + Ta \* math.cos(pi / 14 \* (t0 + t - tm)) - m - 25) / 5) \* y[i - 1]

# concent.append(y[i])

# # 第二问 22个小时

# for i in range(1,15000,1):

# t = t0+h \* i

# y[i] = y[i - 1] + h \* k \* 10 \*\* (0) \* y[i - 1]

# concent.append(y[i])

#

# t=np.arange(9,24,0.001)

# plt.plot(t, concent)

# plt.grid(color="black", which="both", linestyle=':', linewidth=1)

# plt.fill(color='g', alpha=0.3)

# plt.title("picture")

# plt.show()

# print(concent[999])