# Abstract

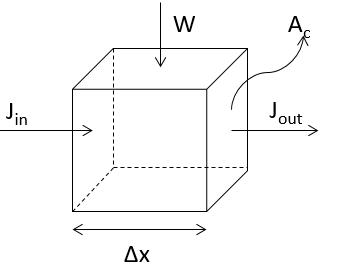
# Introduction

Total discharge of COD and Nitrogen are described below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Reactor 1 | Reactor 2 | Reactor 3 | Reactor 4 | Reactor 5 |
| Length(km) | 52 | 62 | 57 | 157 | 87 |
| W of COD (kg/s) | 4.632 | 4.849 | 4.537 | 5.362 | 5.221 |
| W of Nitrogen(kg/s) | 0.786 | 0.863 | 0.842 | 1.112 | 0.965 |

# Model description

The models covered in the first chapter are all referred to two substantial equations: water balance and substance balance which identify pollutant transport process within different reactors.



*Fig. schematization of physically based model*

The mass balance of water is given by, (eq.1)

The mass balance of pollutant is given by, (eq.2)

Where, , water density,

, concentration of pollutant,

, incoming mass flux rate, , which can decompose to

, incoming mass flux rate,

, cross-sectional area perpendicular to the flux rates,

, loading, which can be specified as impulse loading, step loading and pulse loading.

, first order decay rate

, system volume.

## Completely Mixed System

1. **Physically based theory**

In a completely mixed system, concentration will not be varying through space, which means that in each reactor with space, concentration remains constant.

Furthermore, steady state solution is constructed in our model in stead of monitoring time-variant concentration. These two conditions contribute to our solution to this problem referring to eq.(1),

Rearrange these terms will lead to the solution of concentration at steady state with denoting different reactors.

1. **Model set-up**

First order decay rate needs to be calibrated by fitting observations well. In this perspective, our data are split as calibration period and validation period. From the data we collected, the first three year (2000-2002) is utilized for calibration and the remaining (2003-2004) for validation. Data separation at one station are pasted below and more detailed info are attached in the *appendix*.

*Tab. The description of data used in the model*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Station | Q  () | W\_COD  (mg/l) | W\_NH  (mg/s) | C\_COD  (mg/l) | C\_NH  (mg/l) | Length  (km) | Area  (m^2) | year | type |
| Yueyang | 2650 | 4.632 | 0.786 | 2.798 | 0.270 | 52 | 28445.1 | 2000 | **Calibration** |
| 2650 | 0.938 | 0.220 | 30107.4 | 2001 |
| 2650 | 1.544 | 0.501 | 28614.6 | 2002 |
| 2650 | 2.656 | 0.389 | 29473.9 | 2003 | **Validation** |
| 2650 | 3.520 | 0.080 | 29407.4 | 2004 |

Due to the limitation of data collection, five-year discharge and loading are integrated in the preliminary scrutiny.

To calibrate the model, specific objective function needs to be determined. After considering the sparse data, different reactors and response, multi-objective function could mitigate such issues in a way and then is used in our case, which can also separate into two different estimators RMSR (Root Mean Square Error) and Relative Error Method.

Where, , observed sequence

. Modelled sequence

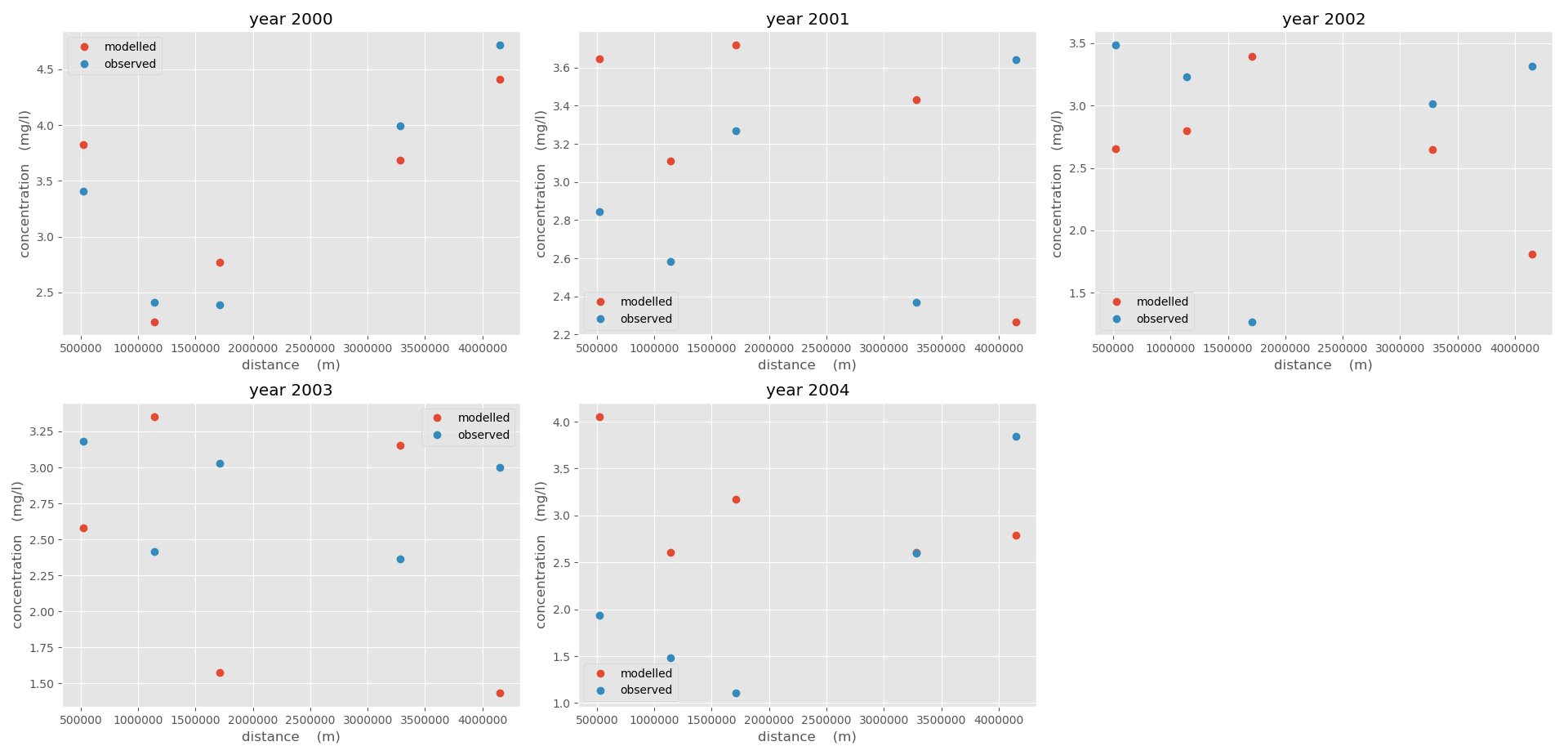
, weights. In this case, ,

1. **Results**

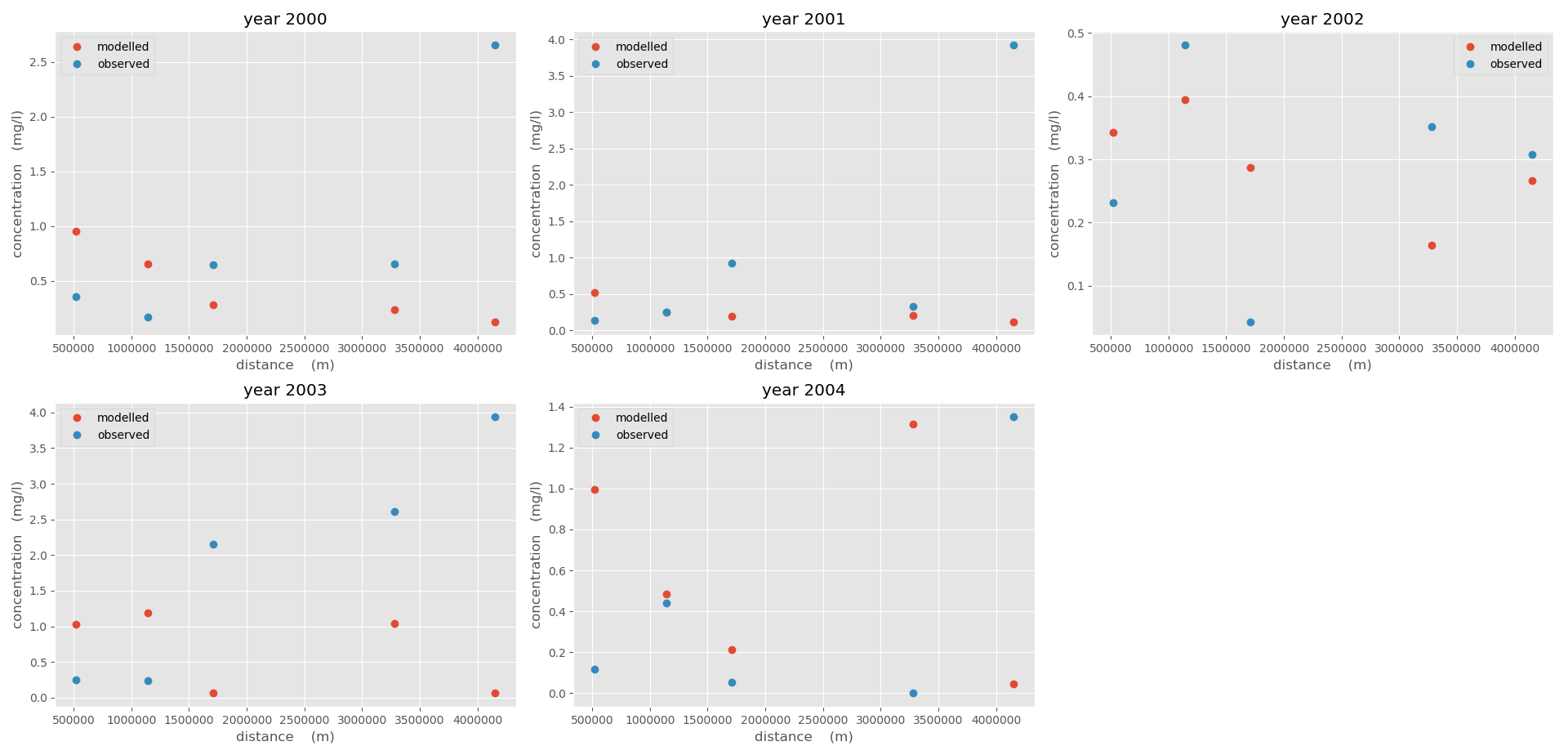
The modeled results are attached in table. During calibration, first order decay can be estimated as 0.0291/d and 0.283/d for COD and Nitrogen respectively. After that, this value we used for testing the robustness of our model. Relative error shows the difference between simulated concentration and observed. Apparently, the error of dissolved nitrogen is relatively larger than COD.

1. **Discussion**

This model performance is not as satisfactory as we may expect. The reason behind it could be discovered according to the mismatch between input data and theoretical solution. That is to say, the data we collected are averaged over years, inconsistent and noisy due to some interferences. Since it is time-consuming to clean data and check the data, this work is not covered in our model configuration. secondly, completely mixed system is somewhat an ideal situation as pollutants are both time and space variant. Besides, pollutant settling is not taken into account in this case. In other words, it is more complicated in real situation and this model is not well representative to some extent. What’s more, the scarcity of data limits the training process, which in a way results in the bad behavior even though it converges to an optimal decay factor.



*Fig. comparison of modelled and observed COD*



*Fig. comparison of modelled and observed Nitrogen*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reactor | K\_COD | K\_NH | Modeled C\_COD | Observed C\_COD | Relative error  (%) | modeled C\_NH | Observed C\_NH | Relative error  (%) | year | Data usage |
| (/d) | (/d) | (mg/l) | (mg/l) | (mg/l) | (mg/l) |
| 1 | 0.0265 | -0.0629 | 3.881 | 3.41 | 12.13604741 | 0.3556 | 0.355555 | 0.012547641 | 2000 | calibration |
| 0.0196 | -0.0314 | 2.368 | 2.844 | 20.10135135 | 0.144 | 0.143823 | 0.12296843 | 2001 |
| 0.0413 | 0.283 | 2.596 | 3.485 | 34.2449923 | 0.281 | 0.232116 | 17.3963136 | 2002 |
| 0.0291 | 0.283 | 3.685 | 3.179 | 13.73134328 | 0.237 | 0.254861 | 7.536316936 | 2003 | validation |
| 4.41 | 1.935 | 56.12244898 | 0.13 | 0.114862 | 11.6442854 | 2004 |
| 2 | 0.0206 | -0.0629 | 3.668 | 2.413 | 34.21483097 | 0.364 | 0.17055 | 53.14555766 | 2000 | calibration |
| 0.0412 | -0.0314 | 3.178 | 2.583 | 18.72246696 | 0.12 | 0.253044 | 110.8702435 | 2001 |
| 0.0823 | 0.283 | 3.621 | 3.232 | 10.7428887 | 0.192 | 0.480799 | 150.4163157 | 2002 |
| 0.0481 | 0.283 | 3.432 | 2.418 | 29.54545455 | 0.206 | 0.239385 | 16.20654831 | 2003 | validation |
| 2.268 | 1.478 | 34.8324515 | 0.121 | 0.438368 | 262.287435 | 2004 |
| 3 | 0.0206 | -0.0629 | 2.675 | 2.392 | 10.57943925 | 0.201 | 0.65236 | 224.5572139 | 2000 | calibration |
| 0.0412 | -0.0314 | 2.887 | 3.269 | 13.23172844 | 0.145 | 0.92287 | 536.462069 | 2001 |
| 0.0823 | 0.283 | 3.266 | 1.269 | 61.14513166 | 0.287 | 0.04365 | 84.79094077 | 2002 |
| 0.0481 | 0.283 | 2.65 | 3.029 | 14.30188679 | 0.165 | 2.1489 | 1202.363636 | 2003 | validation |
| 1.807 | 1.111 | 38.5168788 | 0.266 | 0.053123 | 80.02894737 | 2004 |
| 4 | 0.0206 | -0.0629 | 2.606 | 3.994 | 53.26170376 | 0.547 | 0.656578 | 20.03254113 | 2000 | calibration |
| 0.0412 | -0.0314 | 3.479 | 2.37 | 31.87697614 | 0.393 | 0.331283 | 15.70407125 | 2001 |
| 0.0823 | 0.283 | 1.504 | 3.013 | 100.3324468 | 1.043 | 3.519871 | 237.4756472 | 2002 |
| 0.0481 | 0.283 | 3.153 | 2.367 | 24.92863939 | 0.147 | 2.609751 | 1675.340816 | 2003 | validation |
| 1.434 | 2.56 | 78.52161785 | 0.072 | 0.00035 | 99.51388889 | 2004 |
| 5 | 0.0206 | -0.0629 | 4.094 | 4.717 | 15.2173913 | 0.555 | 2.653542 | 378.1156757 | 2000 | calibration |
| 0.0412 | -0.0314 | 2.695 | 3.64 | 35.06493506 | 0.177 | 3.92365 | 2116.751412 | 2001 |
| 0.0823 | 0.283 | 3.043 | 3.316 | 8.971409793 | 0.212 | 0.3075 | 45.04716981 | 2002 |
| 0.0481 | 0.283 | 2.605 | 2.997 | 15.04798464 | 1.314 | 3.944056 | 200.1564539 | 2003 | validation |
| 2.793 | 3.847 | 37.73720014 | 0.045 | 1.352803 | 2906.229399 | 2004 |

1. System Susceptibility

A system can be considered as susceptible if it assimilates the pollutants at a slow rate. The assimilation factor depends both on the intrinsic properties of a system such as first order decay rate, flow rate, setting velocity as well as the external loading rate of the pollutants.

With reference to eq.(1), one can assess the steady state concentration of the system as a function of intrinsic system characteristics and the loading rate.

Where, is called the assimilation factor, which if higher leads to higher rate of assimilation of the lake.

*Table. Assimilation factor of COD*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | Reactor 1 | Reactor 2 | Reactor 3 | Reactor 4 | Reactor 5 |
| 2000 | 34.3147 | 34.311 | 34.313 | 34.3136 | 34.3133 |
| 2001 | 34.3148 | 34.3112 | 34.3132 | 34.3136 | 34.3132 |
| 2002 | 34.3147 | 34.311 | 34.313 | 34.3137 | 34.3134 |
| 2003 | 34.3148 | 34.311 | 34.3131 | 34.3135 | 34.3132 |
| 2004 | 34.3148 | 34.311 | 34.313 | 34.3136 | 34.3135 |
| Mean | 34.31476 | 34.31104 | 34.31306 | 34.3136 | 34.31332 |
| std | 5.48E-05 | 8.94E-05 | 8.94E-05 | 7.07E-05 | 0.00013 |

*Table. Assimilation factor of Nitrogen*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | Reactor 1 | Reactor 2 | Reactor 3 | Reactor 4 | Reactor 5 |
| 2000 | 197404 | 122002 | 152898 | 165097 | 160037 |
| 2001 | 201335 | 124532 | 156260 | 166811 | 157076 |
| 2002 | 197818 | 121798 | 153359 | 167412 | 160801 |
| 2003 | 199871 | 121517 | 155021 | 163996 | 157164 |
| 2004 | 199715 | 121315 | 153052 | 165335 | 162902 |
| Mean | 199228.6 | 122232.8 | 154118 | 165730.2 | 159596 |
| std | 1442.576944 | 1173.317 | 1311.824 | 1229.728 | 2228.953 |

With reference to the table, assimilation factor of DIN(Dissolved Inorganic Nitrogen) is much larger than COD, which shows the higher capacity of dissolving Nitrogen of this river. While the standard deviation demonstrates the variation of DIN is obvious through the selected years, which may be due to different amount of sewage discharge. Within each reactor, the first segment of the modeled river generally is more powerful transferring pollutants whereas the second one is considered as the most susceptible in the system.

1. System Recovery Time

Resilience of a system is measured by how a system recovers after a pollution accident. In stead of steady state we mentioned above, this process considers substance decaying with time. This includes how fast it declines or how resiliently the system responds to change.

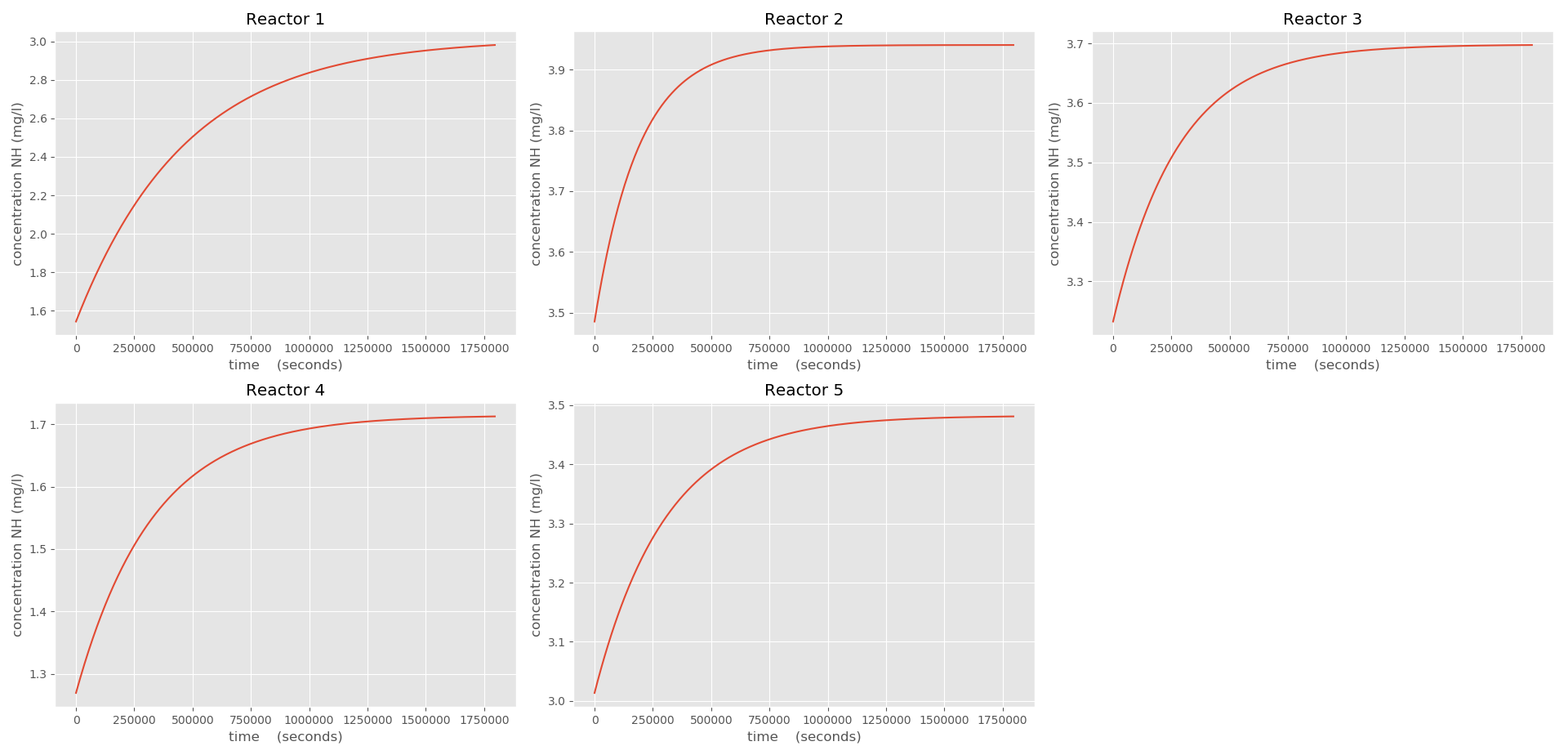
The governing equation for this assessment can be derived as,

And for a completely mixed system, this leads to simplified form of temporal variation of pollutant.

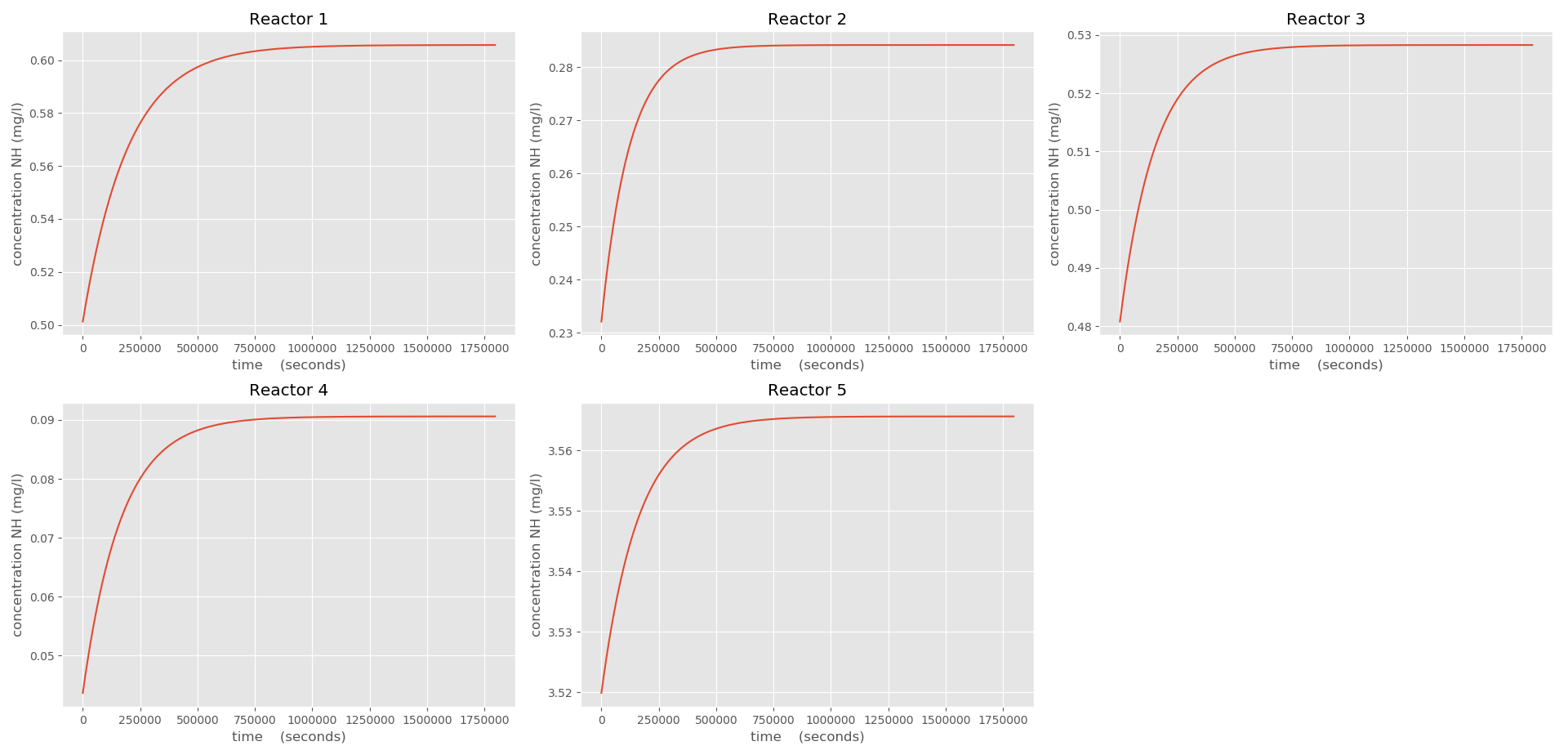
Where,

The solution to this first order differential equation can be generalized according to different loading type. In this model configuration, step loading is included and then solution to this shows below.

The response of pollutant through time is illustrated below (only year 2002 is selected)

****

*Fig. System recovery time for COD*



*Fig. system recovery time for Nitrogen*

## Incompletely Mixed System

1. **Physically based theory**

Before the substance gets well mixed, there is a transient process physically called diffusion and advection. With analogy, mass balance of water and pollutant are taken into account whereas these equations involved are specified both in space and time.

Referring back to eq.(2), mass balance of substance is obtained and expressed by mass flux. For **diffusion** process, we introduce Fick’s first law.

Where, , diffusion coefficient

In a differential element, we consider not varying too much with . Hence, we approximate by using Taylor series expansion. This trick will lead us to diffusion only concentration formula.

For **advection** process, mass flux term can be given by . Following the same mathematical derivation, we obtain advection only concentration formula.

Where, , flow velocity,

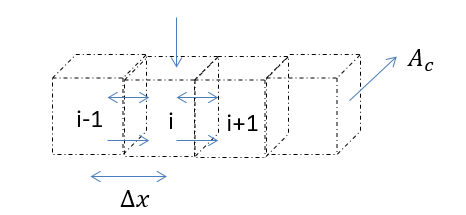
Combining the two dominated process together, this yields to advection, dispersion with decay equation (ADE)

1. **Numerical solutions**
2. Euler Explicit Method

For a time and space derivative problem, one of the most obvious approach is step forward method. The basic idea is to approximate time and space varying problem by,

As Euler explicit method is conditionally stable, much attention should be put when choosing time step

1. Control Volume Approach



*Fig. schematization of numerical approach*

We can rearrange ADE equation by numerical consideration.

While in our model, we only take steady state into account, then this leads to time-invariant numerical solution.

Rewrite this in a matrix form according to n elements.

This work has been done in python scripts, which are attached in the appendix.

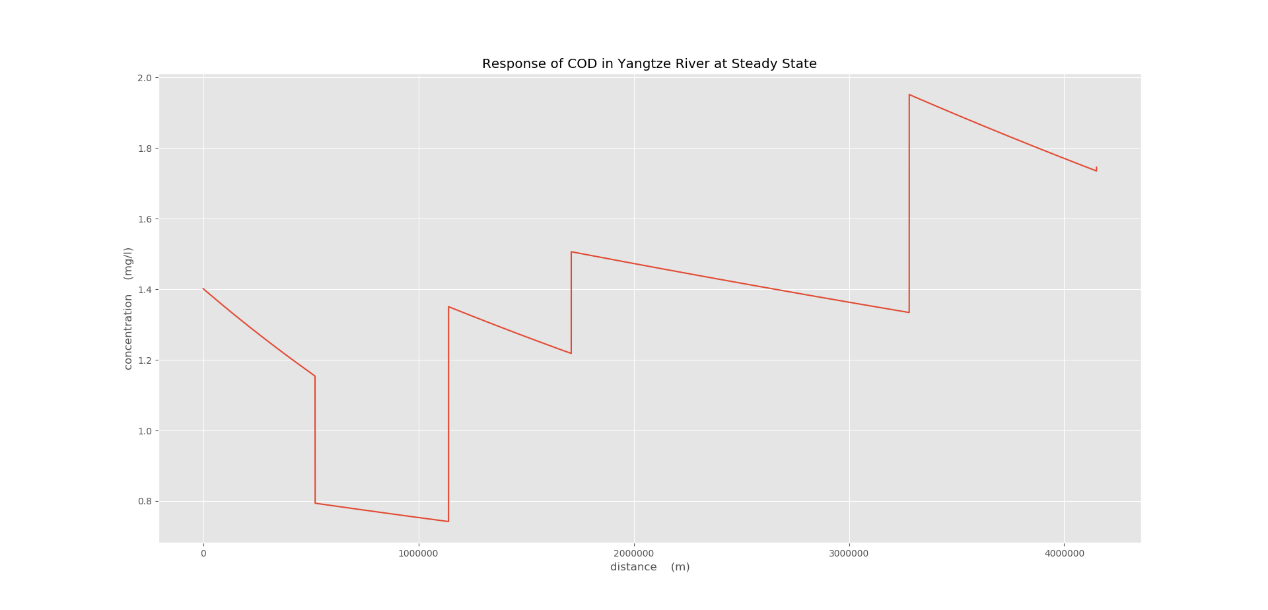
1. Model set-up

The data we selected are in year 2004 as we can predict the system response in a way.

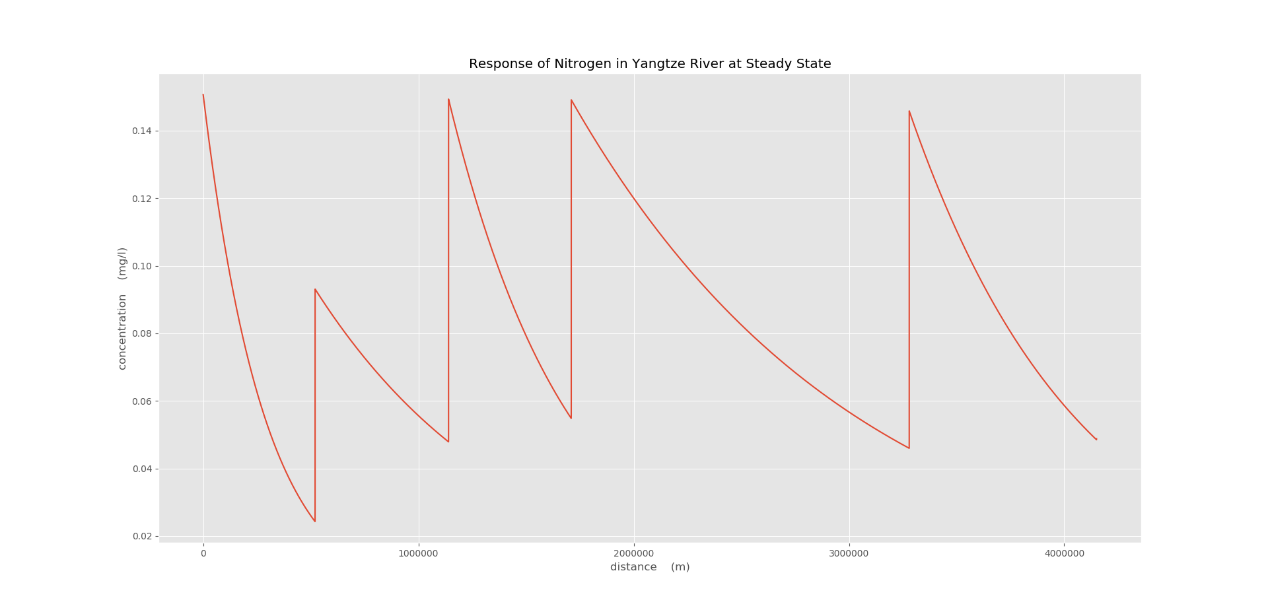
*Tab. Model set-up for incompletely system*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Segment | Length (km) |  | boxes | k\_COD(/d) | k\_NH(/d) | Dispersion |
| 1 | 52 | 100 | 520 | 0.0291 | 0.283 | 0.1 |
| 2 | 62 | 100 | 620 |
| 3 | 57 | 100 | 570 |
| 4 | 157 | 100 | 1570 |
| 5 | 87 | 100 | 870 |

1. **Results**



*Fig. Response of COD in Yangtze River at steady state in incompletely mixed system*



*Fig. Response of DIN in Yangtze River at steady state in incompletely mixed system*

1. **Discussion**

The results of the river response of COD is not expected as exponential decay with space because in this case, the decay factor we use is too small to result in fast decay while for nitrogen, the exponential decay with space is apparent.

Along each reactor, the rate of change in space is not the same because of the impact of different magnitude of discharge, which will flush the pollutant quickly once it has a large value, step loading and even the total volume of each reactor. The rising or falling limbs are associated with the loading as the external force.

Other factors which may affect the result such as precipitation and evaporation etc are neglected in this case since the discharge, total volume are averaged over years. If we need to dive into higher resolution, much more detailed information need to be collected.

1. Economic Impact of Yangtze basin

In this case, we study the relationship of economic development and environmental quality change of Hubei province, specifically the middle reach of Yangtze river from 2000 to 2004 and make a prediction for the economic development condition in 2005 according to the predictive water quality condition.

We divided socio-economic research into three parts: social-economic modelling, environmental prediction and social-environmental issue over generations.

* 1. social-economic modelling

3.1.1 Producer function

In this case, we select cotton production and industry output value in Hubei province, China as two producers; land and capital input as two resources. Cotton production and industry would utilize land and capital respectively. The yields of two producers obey the functions below:

In which, , the yield of cotton crop (tons);

, the land use of cotton crop ();

, the capital input of cotton crop (billion);

, the yield of industry output value (billion);

, the land use of industry ();

, the capital input of industry (billion);

, the coefficients of yield functions

From the collected data of the capital input and land use for cotton crop and industry in 2000, we calibrated theses coefficients and lead to the producer functions,

Controlling the different levels of cotton and industry yield, the contour line with respect to land use and capital were obtained as below.

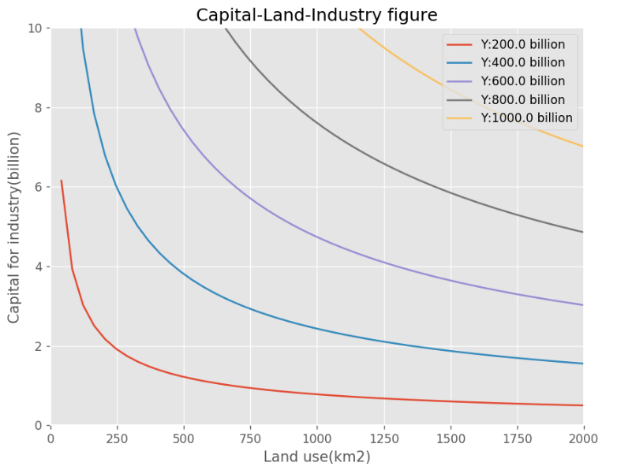
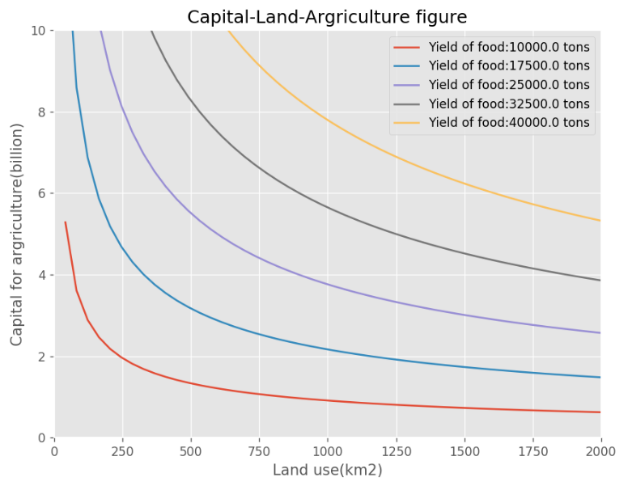


Fig. fig.

3.2 Edgeworth Box

The total resources of land use and capital are confined as 2000 and 10 billion respectively. furthermore, by limiting one resource and moving the other one until they are barely touched, we could obtain one tangent point of well-distributed resources. Replicating the procedure, it yields to the graph of edgeworth box.

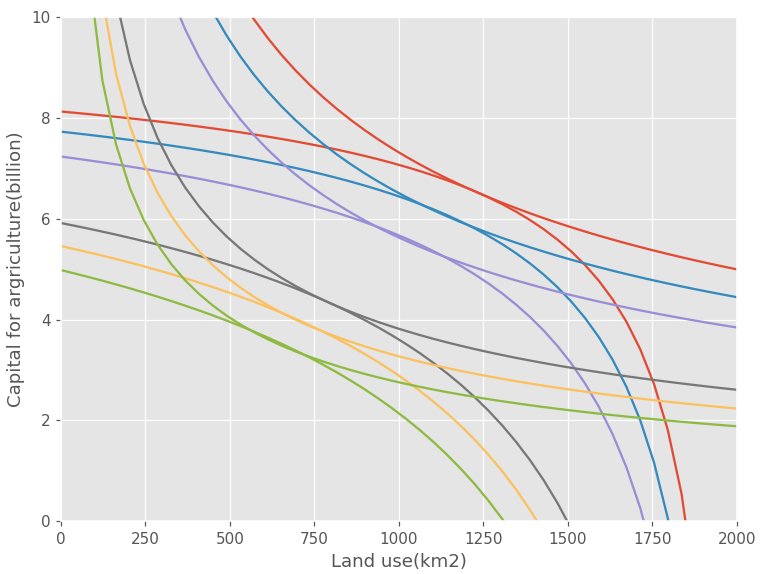


Fig.

3.3 Production Possibility Frontier

Collecting the yields value of two resources at each tangent point, plot them into one graph and model a function line trying to fit these points.

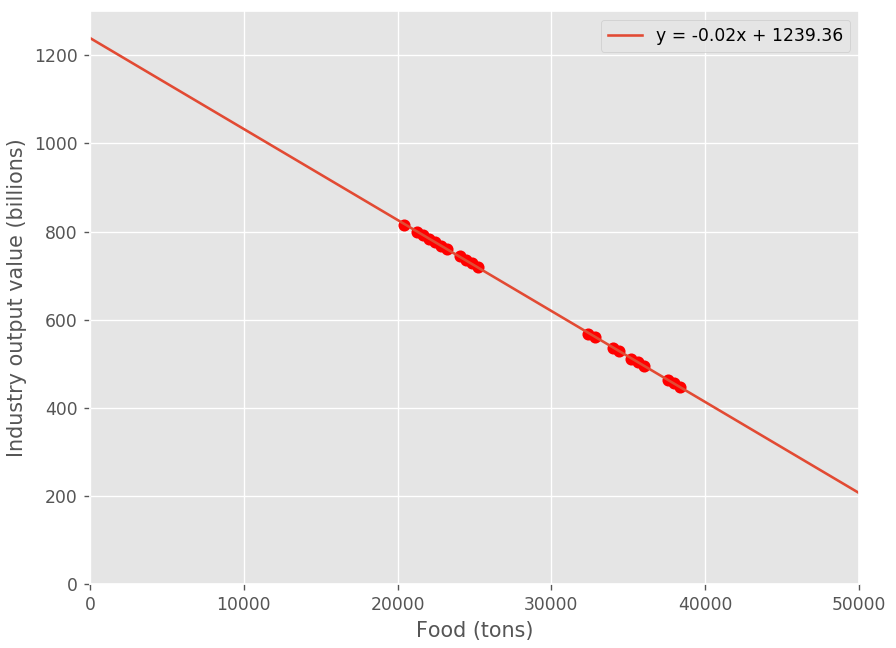


Fig.

The possible function to describe this line follows,

3.4 The equilibrium of Production Possibility Function and Utility Function

1. social-environmental modelling over generations

in this chapter, we construct a model referring to social capital and environment quality together and monitor the evolution of capital and environmental quality as they should hit to an equilibrium point in a long run.

* 1. Model description

As we still deal with two functions, utility function and producer function and seek for a balance between residents’ happiness and social total production, environment and consumption are taken as input in the utility function with a weight, and the production of all commodities as producer.

These parameters will be described later.

Environment evolution with time is quantified below,

Model configuration:

|  |  |  |
| --- | --- | --- |
| Parameter set | Parameter description | Value |
|  | Preference for environment quality | 0.5 |
|  | Degree of habit formation for quality | 0.9 |
|  | Environment self rate of degradation | 0.2 |
|  | Environmental maintenance efficiency | 0.5 |
|  | Tax rate to conserve environment per unit consumption | 0.1 |
|  | Capital share of output | 0.025 |
|  | Degradation of environment cause by 1 unit of consumption | 0.1 |
|  | Capital share of output | 0.36 |
|  | Total factor of productivity | 8.45 |

Running it with these configurations in Matlab, the results are shown below.

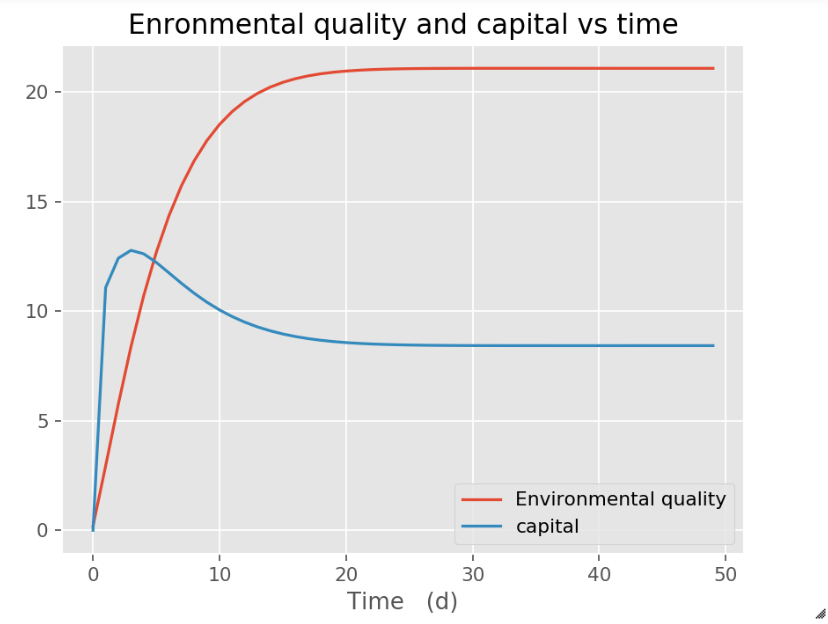


Fig. environment quality and capital develops in time

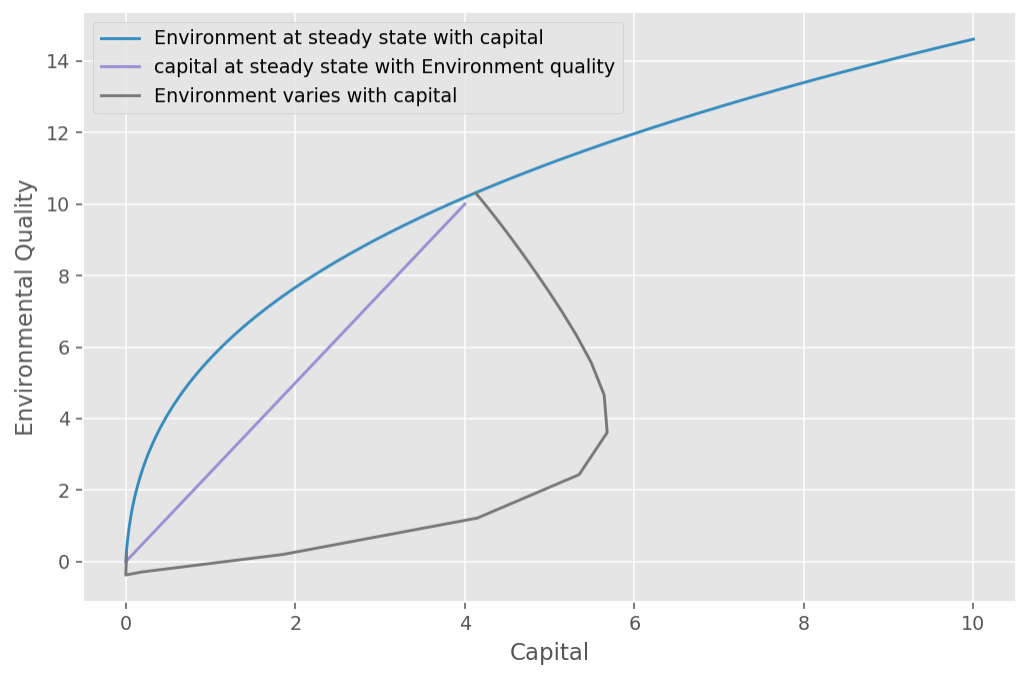
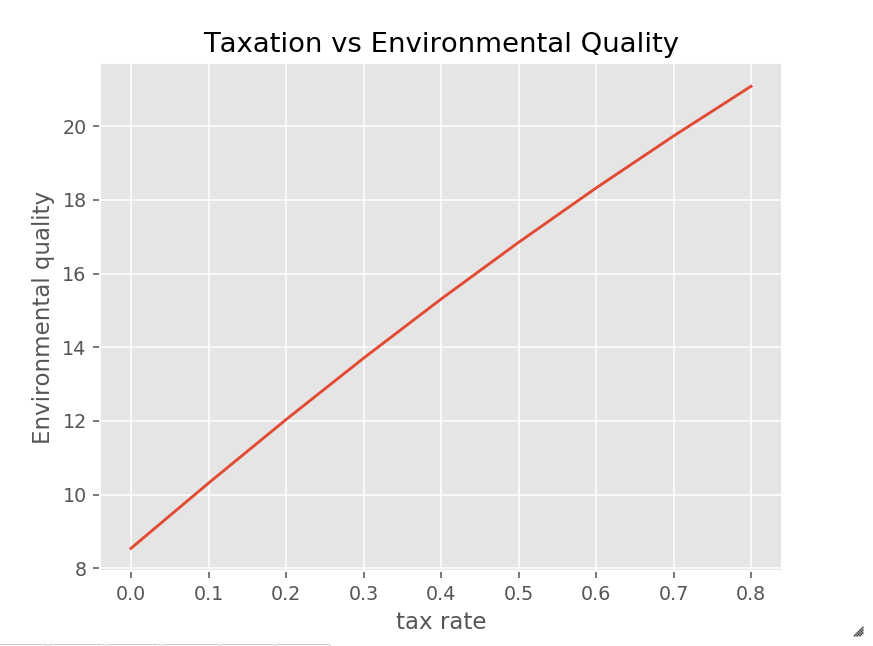


Fig. evolution of capital and environment quality to achieve steady state

In this case, environment quality and capital will eventually reach around 20 and 8 units and the trajectory of the dynamic model to steady state is pasted beforehand. Therefore, this society will be considered as healthy because it does not lead to societal collapse.

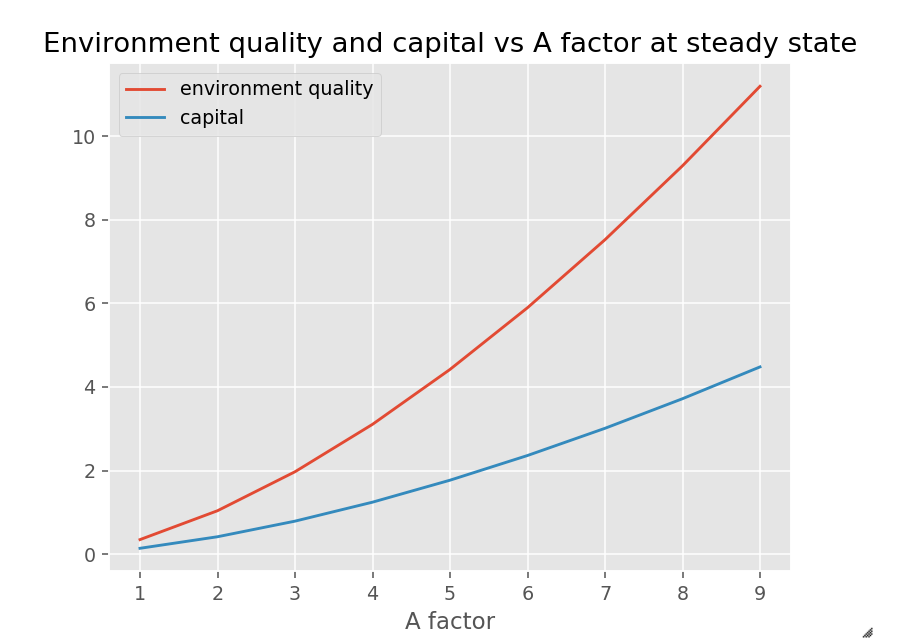
4.2 Taxation affect our model



As it illustrates, increased tax rate results in more money put into the improvement of environment quality so that an expected better environment quality occurs with higher tax rate. But note, this can only explain the situation associated with these parameters set up and is not representative of all societies,

* 1. technology effect

Technology factor represents the level of production and will promote the development of such a society. On the other hand, more resources could be transferred to enhance the environment quality. Hence, huge improvement will be observed as time.



# Eco-system Impact of Yangtze basin

# Further Consideration

# Conclusion

# Reference

### Appendix

1.Python Codes

Object Orientated Programming is established to solve for this model in part 1.

*class model(object):*

*def \_\_init\_\_(self, k,D=0.1):*

*self.k = k*

*self.C = []*

*self.Q = []*

*self.W = []*

*self.D = D*

*self.A\_c = []*

*self.L = []*

*def add\_elements(self,c,Q,W,A\_c=0,delx=20 ,L=100):*

*self.C=np.array(c)*

*self.Q=np.array(Q)*

*self.W=np.array(W)*

*self.L =np.array(L)*

*self.A\_c=np.array(A\_c)*

*self.V\_tot = self.A\_c\*self.L if len(self.A\_c)==len(self.L) else self.A\_c[:-1]\*self.L*

*self.delx= delx*

*self.V = self.A\_c\*self.delx*

*def solve(self,method='CMSS'):*

*# build matrix*

*if method=='CMSS':*

*size = len(self.Q)*

*C\_new = np.zeros(size)*

*for i in range(size):*

*C\_new[i] = (self.W[i]+(self.C[i]\*self.Q[i]))/(self.Q[i]+self.k\*self.V\_tot[i])*

*return C\_new*

*if method=='IMSS':*

*size = len(self.C)-1*

*C\_new = []*

*for i in range(size):*

*n = int(self.L[i]/self.delx)*

*D\_aver = self.D\*self.A\_c[i]/self.delx*

*low\_diag = (-self.Q[i]-D\_aver)\*np.ones(n-1)*

*up\_diag = -D\_aver\*np.ones(n-1)*

*main\_diag = (self.Q[i]+D\_aver+D\_aver+self.k\*self.A\_c[i]\*self.delx)\*np.ones(n-2)*

*main\_diag = np.hstack((self.C[i]\*self.Q[i]+self.Q[i]+self.k\*self.A\_c[0]\*self.delx,main\_diag, D\_aver+self.k\*self.A\_c[i]\*self.delx+self.C[i+1]\*self.Q[i+1]))*

*matrix = diags([up\_diag,main\_diag,low\_diag],[1,0,-1])*

*matrix += np.diag(low\_diag,-1)*

*matrix += np.diag(main\_diag,0)*

*rhs = np.hstack((self.W[i],np.zeros(n-1)))*

*print(rhs.shape,matrix.shape)*

*with open("results\_IMSS.csv",'w') as f:*

*f.write(str(matrix))*

*f.close()*

*C\_new.append(np.linalg.solve(matrix,rhs))*

*return C\_new*

*if method == 'IMSS full':*

*#try:*

*# self.Q.shape == self.W.shape+1*

*# self.C.shape == self.Q.shape*

*#except:*

*# raise RuntimeError("The shape of Q or C should include boundary condition")*

*# C is from the first to the second last*

*# the same for Q W,A*

*size = len(self.C[:-1])*

*n = [int(self.L[i]/self.delx) for i in range(size)]*

*Q\_1 = np.repeat(self.Q[:-1],n)*

*A\_1 = np.repeat(self.A\_c,n)*

*num = np.cumsum(n)-1*

*num = np.hstack((0,num))*

*W\_1 = np.repeat([0,0,0,0,0],n)*

*for i in range(size):*

*W\_1[num[i]:num[i+1]:n[i]] =self.W[i]*

*D\_aver = self.D\*A\_1[1:]/self.delx*

*low\_diag = (-Q\_1[1:]-D\_aver)*

*up\_diag = -D\_aver*

*main\_diag = Q\_1[2:]+D\_aver[1:]+D\_aver[:-1]+self.k\*A\_1[1:-1]\*self.delx*

*main\_diag = np.hstack((self.Q[1]+D\_aver[0]+self.k\*self.delx\*self.A\_c[0],main\_diag, self.Q[-1]+D\_aver[-1]+self.k\*self.A\_c[-1]\*self.delx))*

*matrix = diags([up\_diag,main\_diag,low\_diag],[1,0,-1],format='csr')*

*rhs\_1 = W\_1*

*rhs\_2 = np.zeros(len(W\_1))*

*rhs\_2[0] = self.C[0]\*self.Q[0]*

*rhs = rhs\_1+rhs\_2*

*C\_new = spsolve(matrix,rhs)*

*with open("results\_full.csv",'w') as f:*

*f.write(str(C\_new))*

*f.close()*

*return C\_new*

*def draw(self):*

*plt.figure()*

*plt.plot(np.cumsum(self.L),self.solve())*

*plt.show()*

*def residence(self):*

*# The resisdence time of a pollutant is the average time that a pollutant spends within a completely mixed system*

*residence = np.zeros(len(self.Q))*

*for i in range(len(self.Q)):*

*residence[i] = self.V\_tot[i]/(self.Q[i]+self.k\*self.V\_tot[i])*

*return residence*

*def resilience(self,t, to, loadtype='stepload'):*

*# t is nd-array*

*#resilience of a system is measured by how a system recovers after a pollution accident*

*lab = np.zeros(len(self.Q))*

*res = np.zeros((len(t),len(self.Q)))*

*if loadtype == 'stepload':*

*for i in range(len(self.Q)):*

*for j in range(len(t)):*

*lab[i] = self.Q[i]/self.V\_tot[i]+self.k*

*res[j,i] = self.W[i]/lab[i]/self.V\_tot[i]\*np.exp(-lab[i]\*(t[j]-to))+self.C[i]*

*return res*

*def calibration(self, observed):*

*obs = observed*

*k = self.k*

*def simulated(k):*

*ml.k = k*

*return ml.solve()*

*def objective(k,o):*

*#print((simulated(k)-o)\*\*2).sum()*

*#print(np.abs(simulated(k)-o)/o)*

*return np.sqrt((np.abs(simulated(k)-o)\*\*2).sum())*

*bnds = (0, None)*

*k = minimize(objective, x0=1e6, args=(obs) , method='Nelder-Mead' ).x*

*print(objective(k,observed))*

*#print((np.abs(simulated(k)-observed)/observed\*100).sum())*

*return k*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Q(m^3/s)** | **COD(mg/L)** | **NH4(mg/L)** | **W\_COD(kg/s)** | **W\_NH4(kg/s)** | **Length(km)** | **water depth (m)** | **width (m)** | **Cross-sectional Area(m^2)** | **year** |
| **Below Yueyang** | 2650 | 2.79847789 | 0.2697661 |  |  | 52 | 28.32276946 | 1004.32 | 28445.12 | 2000 |
| 2650 | 0.93812236 | 0.2199294 | 52 | 29.97784712 | 1004.32 | 30107.35 | 2001 |
| 2650 | 1.54391944 | 0.5012351 | 4.632 | 0.786 | 52 | 28.49153955 | 1004.32 | 28614.62 | 2002 |
| 2650 | 2.65626317 | 0.3893347 | 52 | 29.34716202 | 1004.32 | 29473.94 | 2003 |
| 2650 | 3.52037068 | 0.0799647 | 52 | 29.28090058 | 1004.32 | 29407.39 | 2004 |
| **Hong Hu** | 9960 | 3.40988441 | 0.3555554 | 62 | 26.7360119 | 1220.66 | 32635.58 | 2000 |
| 9960 | 2.8435457 | 0.1438229 | 62 | 27.67211434 | 1220.66 | 33778.24 | 2001 |
| 9960 | 3.48518589 | 0.2321164 | 62 | 26.6615582 | 1220.66 | 32544.7 | 2002 |
| 9960 | 3.17886005 | 0.2548611 | 4.849 | 0.863 | 62 | 26.55957195 | 1220.66 | 32420.21 | 2003 |
| 9960 | 1.93533619 | 0.1148624 | 62 | 26.48616683 | 1220.66 | 32330.6 | 2004 |
| **Yanwu** | 8830 | 2.41269628 | 0.1705502 | 57 | 24.08166162 | 1969.55 | 47430.04 | 2000 |
| 8830 | 2.58288487 | 0.2530443 | 57 | 25.16597819 | 1969.55 | 49565.65 | 2001 |
| 8830 | 3.23244859 | 0.4807993 | 57 | 24.22744104 | 1969.55 | 47717.16 | 2002 |
| 8830 | 2.41806828 | 0.2393855 | 4.537 | 0.842 | 57 | 24.76061247 | 1969.55 | 48767.26 | 2003 |
| 8830 | 1.47981563 | 0.4383678 | 57 | 24.13035992 | 1969.55 | 47525.95 | 2004 |
| **Dazui** | 10700 | 2.39185234 | 0.65236 | 157 | 22.985967 | 1065.44 | 24490.17 | 2000 |
| 10700 | 3.26864551 | 0.92287 | 157 | 23.51195038 | 1065.44 | 25050.57 | 2001 |
| 10700 | 1.26920302 | 0.04365 | 157 | 23.69921009 | 1065.44 | 25250.09 | 2002 |
| 10700 | 3.02887648 | 2.1489 | 5.362 | 1.112 | 157 | 22.65493205 | 1065.44 | 24137.47 | 2003 |
| 10700 | 1.10758219 | 0.053123 | 157 | 23.05836121 | 1065.44 | 24567.3 | 2004 |
| **Huangqu** | 9990 | 3.99370334 | 0.656578 | 87 | 21.93581151 | 1759.93 | 38605.49 | 2000 |
| 9990 | 2.37036445 | 0.331283 | 87 | 21.10037063 | 1759.93 | 37135.18 | 2001 |
| 9990 | 3.01338786 | 3.519871 | 87 | 22.1569574 | 1759.93 | 38994.69 | 2002 |
| 9990 | 2.36672251 | 2.609751 | 5.221 | 0.965 | 87 | 21.12451815 | 1759.93 | 37177.67 | 2003 |
| 9990 | 2.59945634 | 0.00035 | 87 | 22.77746643 | 1759.93 | 40086.75 | 2004 |
| **Wuxue** | 9930 | 4.71666967 | 2.653542 |  | 20.89743308 | 1053.28 | 22010.85 | 2000 |
| 9930 | 3.64046045 | 3.92365 |  | 21.10725076 | 1053.28 | 22231.85 | 2001 |
| 9930 | 3.31646124 | 0.3075 |  | 21.58382727 | 1053.28 | 22733.81 | 2002 |
| 9930 | 2.99748559 | 3.9440558 |  |  |  | 21.99902239 | 1053.28 | 23171.13 | 2003 |
| 9930 | 3.84681554 | 1.3528032 |  | 20.13583898 | 1053.28 | 21208.68 | 2004 |