



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Enginyeria
de Manresa



TREBALL FINAL DE GRAU

MODELITZACIÓ DEL DECREIXEMENT DE CLOR EN UN SISTEMA DE SUBMINISTRAMENT D'AIGUA CURS 22/23

Autor: ILIAS MAAROUF EL BOUAZZI

Director: SERGI GRAU TORRENT

Grau d'Enginyeria Electrònica Industrial i Automàtica

EPSEM

Data: 05/05/2023



Vull expressar la meva gratitud més profunda a totes aquelles persones que han contribuït en el treball amb el seu suport, consells i col·laboració. Sense ells, aquest projecte no seria el que és avui. Vull agrair de manera especial a:

Sergi Grau Torrent - Per la seva orientació experta i per l'oportunitat de realitzar aquest estudi.

Edgar Fusté Culla - Per la seva col·laboració i aportacions valuoses durant les reunions i discussions.

I a totes aquelles persones del meu cercle que m'han recolzat, m'han servit de motivació constant i m'han animat a superar tots els reptes.

Moltíssimes gràcies.

RESUM

La cloració és un mètode àmpliament utilitzat per a la desinfecció de l'aigua, però la concentració de clor disminueix gradualment amb el temps, afectant la seva eficàcia com a desinfectant. Aquest estudi es centra en la modelització del decreixement del clor en un sistema de subministrament d'aigua, utilitzant programari especialitzat en xarxes de distribució d'aigua.

S'utilitza software especialitzat com EPANET i GISWATER per simular el flux d'aigua i la dispersió del clor. Mitjançant aquestes eines i l'ús de python, l'estudi pretén millorar la comprensió del decreixement del clor en els sistemes de subministrament d'aigua i proporcionar coneixements pràctics per optimitzar els processos de desinfecció permetent als responsables de l'aigua avaluar i controlar els nivells de clor de manera més efectiva.

Els resultats d'aquest estudi contribueixen al coneixement científic i tecnològic en matèria de desinfecció de l'aigua, millorant la gestió i el manteniment dels sistemes de subministrament d'aquest recurs. Mitjançant la modelització precisa del decreixement del clor, es fa possible assegurar una distribució de l'aigua més segura i eficient, salvaguardant així la salut pública i preservant una font vital per a la vida humana.



ABSTRACT

Chlorination is a widely used method for water disinfection, but the chlorine concentration gradually decreases over time, affecting its performance as a disinfectant. This study focuses on modeling chlorine decay in a water supply system, using specialized software for water distribution networks.

Specialized software such as EPANET and GISWATER is used to simulate water flow and chlorine dispersion. Through the use of these tools and python, the study aims to enhance understanding of chlorine decay in water supply systems and provide practical insights to optimize disinfection processes, enabling water managers to assess and control chlorine levels more effectively.

The results of this study contribute to scientific and technological knowledge in water disinfection, improving the management and maintenance of water supply systems. Through accurate modeling of chlorine decay, it becomes possible to ensure a safer and more efficient water distribution, safeguarding public health and preserving a vital resource for human life.

ÍNDEX

1. INTRODUCCIÓ	1
2. CONEIXEMENTS PREVIS	2
2.1. El clor com a desinfectant	2
2.2. Aplicació en les xarxes de distribució d'aigua	3
2.3. EPANET	5
2.3.1. Característiques del model Hidràulic	6
2.3.2. Característiques del model de Qualitat de l'aigua	7
2.3.3. Exemple de xarxa	8
2.3.4. Model del sistema	10
2.4. GIS	20
2.5. QGIS	21
2.6. COMPLEMENTS DE QGIS (PLUGINS)	22
2.6.1. GISWATER	23
2.6.2. EPATOOLS	27
3. PROCEDIMENT EXPERIMENTAL	28
3.1. Introducció prèvia	28
3.1.1. Normativa del clor	28
3.1.2. Criteris de qualitat	29
3.1.3. Incompliments	31
3.1.4. Anomalies	31
3.2. Implicacions ambientals	32
3.3. Decaïment del clor	33
3.3.1. Comportament matemàtic	33
3.3.2. Comportament físic	35
3.3.3. Simulació amb EPANET	37
3.3.4. Relació entre els emissors i el decreixement del clor	41
3.3.5. codi dels coeficients emissors	42



3.3.6.	Comportament de l'algoritme -----	58
3.3.7.	Aplicació codi del coeficient K i modificacions -----	63
4.	CONCLUSIONS -----	68
5.	BIBLIOGRAFIA -----	70



ÍNDEX DE FIGURES

FIGURA 1. HIPOCLORIT DE SODI [FONT 12].....	3
FIGURA 2. DISPOSITIU DOSIFICADOR DE CLOR [FONT 14]	4
FIGURA 3. XARXA AMB ETIQUETES D'IDENTIFICACIÓ [FONT 9].....	9
FIGURA 4. COMPONENTS FÍSICS EN UN SISTEMA DE DISTRIBUCIÓ D'AIGUA [FONT 9]	10
FIGURA 5. CORBA CARACTERÍSTICA EN UN PUNT [FONT 9].....	19
FIGURA 6. CORBA CARACTERÍSTICA DE 3 PUNTS [FONT 9].....	19
FIGURA 7. CORBA CARACTERÍSTICA DE MULTIPUNT VARIABLE [FONT 9]	19
FIGURA 8. CORBA CARACTERÍSTICA DE VELOCITAT [FONT 9]	19
FIGURA 9. CORBA DE RENDIMENT D'UNA BOMBA [FONT 9]	19
FIGURA 10. CORBA DE VOLUM D'UN TANC [FONT 9]	20
FIGURA 11. CONJUNT DE PROGRAMES NECESSARIS PER TREBALLAR AMB GISWATER I COM ES CLASSIFIQUEN [FONT 5]	25
FIGURA 12. PARÀMETRES ADDICIONALS QUE DETERMINEN L'APTITUD DE L'AIGUA PER AL CONSUM [FONT 16].....	30
FIGURA 13. COMPORTAMENT GRÀFIC DE LES EQUACIONS [FONT 9]	34
FIGURA 14. DIFERÈNCIA ENTRE FLUX PRINCIPAL I CAPA LÍMIT [FONT 9] 35	
FIGURA 15. ARCHIU INP PER LA SIMULACIÓ DE EPANET (ZONA EPSPSEM) [FONT PRÒPIA]	37
FIGURA 16. RENDIMENT DE LA BOMBA [FONT PRÒPIA]	38



FIGURA 17. TEMPS TRANSCORREGUT [FONT PRÒPIA].....	38
FIGURA 18. UNITATS DE L'AGENT QUÍMIC (CLOR) [FONT PRÒPIA].....	39
FIGURA 19. DIFERENTS CONFIGURACIONS DE PATTERN [FONT PRÒPIA]	39
FIGURA 20. CONFIGURACIÓ KB I KW [FONT PRÒPIA].....	40
FIGURA 21.OPCIONS DE QUALITAT PER APLICAR CONCENTRACIÓ INICIAL DE CLOR [FONT PRÒPIA].....	40
FIGURA 22. SIMULACIÓ D'EXEMPLE [FONT PRÒPIA]	41

ÍNDEX DE TAULES

TAULA 1. CARACTERÍSTIQUES DELS NODES [FONT 9]	9
TAULA 2. CARACTERÍSTIQUES DE LES CONDUCCIONS [FONT 9]	9
TAULA 3. COEFICIENT DE RESISTÈNCIA PER A CADA MÈTODE [FONT 9] 15	
TAULA 4. COEFICIENT DE RUGOSITAT PER DIFERENTS MATERIALS [FONT 9]	15
TAULA 5. GRUIX DELS DIFERENTS MATERIALS [FONT 9]	15
TAULA 6. COEFICIENTS DE PÈRDUES MENORS [FONT 9]	16
TAULA 7. EQUACIONS DE REACCIÓ [FONT 9]	34

1. INTRODUCCIÓ

L'aigua potable és un recurs vital per a la vida humana i un dels factors clau per a la salut pública. El manteniment de la qualitat de l'aigua és fonamental per garantir la seva idoneïtat per al consum humà i la prevenció de malalties relacionades amb la contaminació microbiològica.

Un dels mètodes més comuns per desinfectar l'aigua és utilitzant clor. El clor és un agent desinfectant eficaç que pot eliminar o inactivar la majoria dels microorganismes patògens presents en l'aigua. No obstant això, el clor té una vida útil limitada i es degrada amb el temps, disminuint la seva concentració i eficàcia desinfectant.

Per això, la modelització del decreixement del clor en un sistema de subministrament d'aigua és un aspecte crucial per a garantir la seguretat i la qualitat de l'aigua potable. Mitjançant l'estudi i l'anàlisi de les variables que influeixen en el decreixement del clor, és possible comprendre millor com es comporta aquest agent desinfectant en diferents condicions i desenvolupar estratègies de control i optimització del sistema de desinfecció.

En aquest treball, s'abordarà la modelització del decreixement del clor en un sistema de subministrament d'aigua, analitzant els principals factors que afecten a la seva degradació i explorant les diferents metodologies i models disponibles per predir aquest procés. A més, s'exploraran les implicacions pràctiques d'aquesta modelització per a la gestió i el manteniment del sistema de desinfecció d'aigua, així com per a la presa de decisions en matèria de salut pública.

Mitjançant aquest estudi, es pretén contribuir al coneixement científic i tecnològic en l'àmbit de la desinfecció de l'aigua i proporcionar eines i recursos per millorar la gestió de la qualitat de l'aigua potable. Amb una comprensió més profunda del decreixement del clor, es pot assegurar un subministrament d'aigua més segur i més eficient per a la població, preservant la salut pública i garantint un recurs vital per a la vida humana.



2. CONEIXEMENTS PREVIS

2.1. El clor com a desinfectant

El clor és un element químic que es troba en la taula periòdica amb el símbol "Cl" i el número atòmic 17. És un gas verdós i molt tòxic en les seves formes més pures, però també pot ser utilitzat com a agent desinfectant en forma de compostos de clor, com ara el clorur de calci o l'hipoclorit de sodi.

La història del clor es remunta a principis del segle XIX, quan es va descobrir i es va aïllar per primera vegada per part del químic suec Carl Wilhelm Scheele. No obstant això, el seu ús generalitzat com a desinfectant i agent de sanejament va començar a principis del segle XX.

El doctor Ignaz Semmelweis va introduir l'ús de clor com a desinfectant a mitjans del segle XIX. Semmelweis va ser un metge hongarès que va observar que les taxes de mortalitat per febre puerperal eren significativament més baixes en les maternitats on s'utilitzava clor per desinfectar les mans i les eines mèdiques. Aquest descobriment va ser un avenç clau en la prevenció de malalties i va establir les bases per a la desinfecció amb clor en àmbits mèdics.

Des de llavors, l'ús del clor com a agent desinfectant s'ha estès a diversos àmbits, incloent el tractament de l'aigua potable i les piscines, la desinfecció de superfícies i eines mèdiques, i altres aplicacions relacionades amb la salut pública i la higiene. El clor ha demostrat ser eficaç per eliminar microorganismes patògens, com ara bacteris i virus, i contribueix a prevenir la propagació de malalties i a protegir la salut pública.

Avui dia, el clor continua sent un dels agents desinfectants més utilitzats en diferents sectors, gràcies a les seves propietats desinfectants i a la seva eficàcia en la destrucció d'organismes patògens. No obstant això, és important utilitzar el clor de manera adequada i garantir una dosificació i un control adequats per assegurar una desinfecció efectiva i evitar possibles efectes negatius per a la salut i el medi ambient.

Al llarg del temps, s'han desenvolupat diferents formes d'aplicació del clor com a agent desinfectant. Una de les formes més comunes és mitjançant la cloració de l'aigua potable. Aquest procés implica l'addició de clor o compostos de clor a l'aigua per destruir els microorganismes patògens presents i garantir la seva seguretat pel consum humà. La cloració de l'aigua ha estat una pràctica estàndard en els sistemes de tractament d'aigua durant molts anys i ha contribuït a reduir de manera significativa la propagació de malalties relacionades amb l'aigua. [3] [10]



2.2. Aplicació en les xarxes de distribució d'aigua

L'aplicació del clor a nivell de les xarxes de distribució d'aigua s'ha convertit en un pas essencial per a la potabilització de l'aigua i a la garantia de la seva seguretat sanitària. El procés d'aplicació del clor en les xarxes de distribució d'aigua segueix els següents passos:

Preparació del clor: abans de la seva aplicació, es prepara una solució de clor per mitjà de clor líquid o de compostos de clor, com ara hipoclorit de sodi o clorur de calci. Aquesta solució conté una concentració específica de clor actiu que s'ajusta segons les necessitats de desinfecció de cada sistema d'aigua potable.



Figura 1. Hipoclorit de sodi [Font 12]

Dosificació del clor: la dosificació del clor es realitza mitjançant l'ús d'un dispositiu de dosificació, com ara una estació de dosificació o un sistema de bombeig. Aquest dispositiu introdueix de manera controlada la quantitat adequada de clor a l'aigua que flueix a través de les xarxes de distribució.



Figura 2. Dispositiu dosificador de clor [Font 14]

Mesura i control del clor residual: és essencial mesurar i controlar la concentració de clor residual en diverses ubicacions clau de la xarxa de distribució. Això es fa mitjançant l'ús de sondes o analitzadors que proporcionen dades en temps real. Els nivells de clor residual s'ajusten i es monitoren regularment per assegurar que s'aconsegueixi la desinfecció adequada.

Temps de contacte: un factor clau en l'eficàcia de la desinfecció amb clor és el temps de contacte entre el clor i els microorganismes presents a l'aigua. Per assegurar un temps de contacte suficient, és important mantenir una velocitat adequada del flux d'aigua a través de les xarxes de distribució, de manera que es permeti una exposició adequada al clor.

Distribució homogènia: és crucial assegurar una distribució homogènia del clor a través de tota la xarxa de distribució per garantir que tota l'aigua que circula per ella estigui desinfectada. Això es pot aconseguir mitjançant la ubicació estratègica de punts de dosificació, la regulació adequada dels sistemes de bombeig i la monitorització periòdica dels nivells de clor en diferents punts de la xarxa.

Control i manteniment: es realitzen inspeccions i manteniments regulars dels sistemes de cloració per assegurar el bon funcionament dels dispositius de dosificació, així com la integritat i la seguretat de les xarxes de distribució d'aigua. Aquestes tasques inclouen neteja, calibratge i substitució periòdica de les parts del sistema, així com la verificació de la qualitat del clor utilitzat i el seu compliment amb les normatives i regulacions aplicables.

Monitorització de la qualitat de l'aigua: paral·lelament a l'aplicació del clor, es realitzen mostres i anàlisis periòdiques de la qualitat de l'aigua en diferents punts de la xarxa de distribució. Això permet avaluar l'eficàcia de la desinfecció i garantir que l'aigua compleix amb els estàndards de potabilitat establerts.



Subproductes de desinfecció: és important tenir en compte els possibles subproductes de desinfecció que es poden formar com a resultat de la reacció del clor amb la matèria orgànica de l'aigua. Aquests subproductes, com ara els trihalometans, poden tenir impactes negatius per a la salut humana. Per mitigar aquest problema, s'utilitzen estratègies com la optimització de la dosificació de clor, l'ús de tècniques alternatives de desinfecció i l'aplicació de tractaments complementaris per reduir la formació de subproductes.

[11]

Tal i com es pot apreciar, el procediment per “dechlorar” l'aigua i que sigui finalment potable per consumir-la és tot un camí llarg amb diferents etapes. Aquest treball se centre, sobretot, en la part que s'encarrega de calcular, predir i mesurar com va decreixent la concentració d'aquest químic en les xarxes de distribució d'aigua potable de la ciutat.

2.3. EPANET

EPANET és un software d'aigües gratuït desenvolupat per la EPA (*Environmental Protection Agency*) i és reconegut per les enginyeries i constructores més prestigioses.

El programa EPANET permet realitzar simulacions en períodes prolongats (un o varis dies) del comportament hidràulic i l'evolució de la qualitat de l'aigua en xarxes d'abastiment a pressió. Una xarxa pot constar de canonades, nodes (unió de canonades), bombes, vàlvules i tancs o dipòsits d'emmagatzematge. EPANET monitoritza l'evolució dels caudals en les canonades, les pressions en els nodes, els nivells marcats en els dipòsits i la concentració de productes químics en suspensió a l'aigua durant tot el període de simulació. Inclús et permet analitzar-lo en diferents intervals de temps.

El software EPANET proporciona un entorn integrat per manipular les dades d'entrada de la xarxa, és capaç de realitzar simulacions hidràuliques i de qualitat del l'aigua i veure els resultats en una varietat de formats. Aquests inclouen mapes de xarxes codificades per colors, taules de números, gràfiques de tendències i mapes de isolínies.

EPANET va sorgir del laboratori Nacional d'Investigació per la Prevenció de Riscos (NRMRL), que forma part de l'Agència per la Protecció del Medi Ambient de EEUU (USEPA). Com a tal, és un software de domini públic que es pot baixar i distribuir-lo lliurement.



EPANET està dissenyat com una eina de desenvolupament per augmentar el coneixement del moviment i el destí dels components de l'aigua en una xarxa de distribució. De fet, es pot utilitzar per moltes aplicacions en l'anàlisi de sistemes de distribució. Dissenyar programes de mostreig, calibrar models hidràulics, analitzar clor residual i avaluar el risc al que s'enfronten els consumidors són algunes de les aplicacions que es poden realitzar. Al mateix temps, EPANET pot servir d'ajuda per l'avaluació de diferents estratègies alternatives de gestió dels sistemes de distribució encaminades totes elles a la millora de la qualitat de l'aigua dins del sistema, això inclou:

- Ús alternatiu de fonts de subministrament en sistemes amb múltiples fonts de subministrament.
- Variació dels esquemes de bombeig, omplert y buidat de tancs.
- Ús de tècniques de tractament via satèl·lit, com la "decloració" en tancs d'emmagatzematge específic
- Identificació de canonades que necessiten ser netejades o reemplaçades.

2.3.1. Característiques del model Hidràulic

Les següents característiques estan extretes del "Manual de Usuario de EPANET" (Rossman et al., 2010). A partir d'aquest document, destaco els punts que es donen a continuació i que són els més rellevants per comprendre com són els models hidràulics a EPANET.

Per la realització del model de qualitat de l'aigua és necessari disposar d'un model hidràulic precís i complet. EPANET és un motor d'anàlisi hidràulic actual que inclou les següents característiques:

- No existeix límit en la mida de la xarxa que es desitja analitzar
- Calcula les pèrdues per fricció en les conduccions mitjançant les expressions de Hazen-Williams, Darcy-Weisbach, o Chezy-Manning
- Inclou pèrdues menors en elements tals com recolzaments, cantonades, etc.
- Modela bombes funcionant tan a velocitat de gir constant com a velocitats de gir variables.
- Calcula l'energia consumida i el cost de bombeig de les estacions.
- Modela diferents tipus de vàlvules, incloent vàlvules de regulació, vàlvules de retenció, vàlvules d'aïllament, vàlvules reductores de pressió, vàlvules de control de cabal, etc.

- Permet l'emmagatzematge d'aigua en tancs que presenten qualsevol geometria (per exemple que la secció del tanc sigui variable amb l'altura d'aquest).
- Considera la possibilitat d'establir diferents categories de consum dels nodes, cada una d'elles amb la seva pròpia corba pròpia de modulació.
- Modelitza consums que depenen de la pressió que surt al exterior del sistema a través d'emissors (ruixadors, aspensors)
- Pot determinar el funcionament del sistema simplement amb el nivell de l'aigua en el tanc i controls de temps o utilitzar un sistema de regulació temporal més complex.

[9]

2.3.2. Característiques del model de Qualitat de l'aigua

A més del model hidràulic, EPANET també té les següents capacitats en la modelització de la qualitat de l'aigua:

- Realitza el seguiment en el temps de substàncies no reactives que es troben a la xarxa.
- Modela el comportament d'un material reactiu tant si augmenta la seva concentració com si es dissipa al llarg del temps.
- Modela l'edat de l'aigua que passa al llarg de la xarxa.
- Realitza el seguiment d'una porció de fluid des de un node donat a través de tots els demés al llarg del temps.
- Modela reaccions en tot el fluid i en la capa de la paret de la canonada.
- Utilitza equacions cinètiques polinòmiques per modelar les reaccions en tot el fluid.
- Utilitza coeficients i equacions lineals per modelar les reaccions a la paret de la canonada.
- Té en compte les limitacions derivades d'aplicar la conservació de la massa en la modelització de les reaccions de la paret de la canonada.
- Permet el creixement o descens de la reacció fins a una concentració límit.
- Realitza coeficients general en les reaccions que poden ser modificades canonada a canonada.
- Permet que els coeficients de les reaccions de paret siguin correlatives amb la rugositat de la canonada.



- Permet les entrades de concentració o massa en qualsevol punt de la instal·lació al llarg del temps.
- Modela els dipòsits de tres formes; de mescla completa, de flux en pistó, o amb dos compartiments de mescla.

Gràcies a totes aquestes característiques, EPANET pot estudiar qualsevol fenomen de qualitat de l'aigua, tals com:

- Barreges d'aigua de diferents fonts.
- Edat de l'aigua al llarg del sistema.
- Disminució del **clor residual**.
- Creixement del subproductes de desinfecció.
- Seguiment de possibles situació de propagació de la contaminació.

Una eina molt important de EPANET que es fa servir en aquest treball és la capacitat que té aquest programa de simular el decreixement del nivell de clor.

[9]

2.3.3. Exemple de xarxa

En aquests passos analitzarem la xarxa de distribució mostrada de la figura 2. Està formada per un dipòsit font (*source reservoir*), per exemple, un pou de decantació, des del qual es bombeja a una xarxa de distribució. També hi ha una canonada connectada a un dipòsit d'emmagatzematge que servirà com a font secundària. Les etiquetes d'identificació (ID labels) dels diferents components es mostren a la figura. Els nodes de la xarxa tenen les característiques que es mostren a la Taula 1 i les característiques de les canonades es troben en la taula 2. A més, la bomba (ID 9) tenen una altura en funció del cabal donat pels punts (H0-Q0, H1-Q1, H2-Q2), el tanc (Node 10) té un diàmetre de 20 m, el nivell del aigua és de 2,25 m, el nivell màxim és de 7 m i el mínim de 2 m.

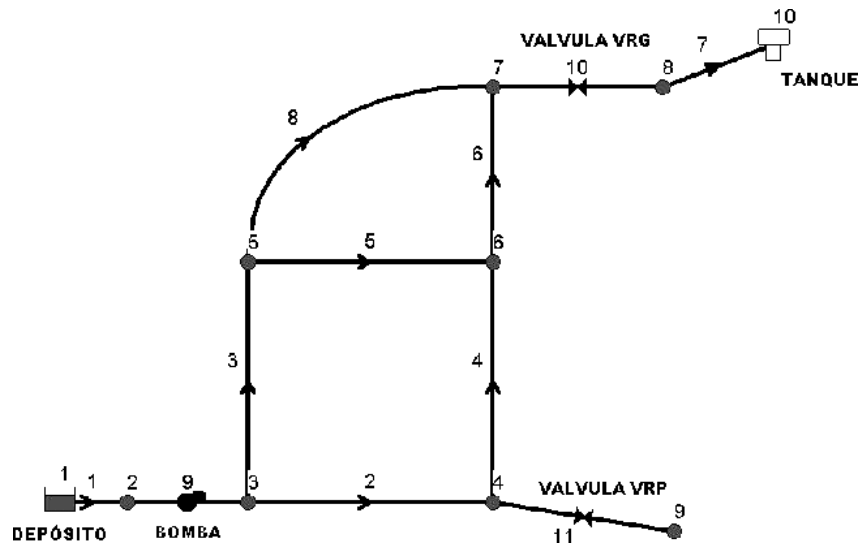


Figura 3. Xarxa amb etiquetes d'identificació [Font 9]

Nudo	Altura (m)	Demanda (l/s)
2	0,5	0
3	3	10
4	2	5
5	0,5	5
6	4	20
7	17	0
8	17	0
9	2	40

Taula 1. Característiques dels nodes [Font 9]

Conducció	Longitud (m)	Diàmetre (mm)	Rugositat (mm)
1	1	300	1,2
2	500	250	1,2
3	500	250	1,2
4	500	200	1,2
5	500	150	1,2
6	500	200	1,2
7	500	300	1,2
8	750	150	1,2

Taula 2. Característiques de les conduccions [Font 9]



[9] [12]

2.3.4. Model del sistema

En aquest apartat es donarà una breu explicació sobre com EPANET modelitza els objectes físics que constitueixen el sistema de distribució així com els seus paràmetres operacionals. També es donaran els fonaments bàsics de com EPANET simula els comportaments hidràulics i de qualitat de l'aigua.

Components físics

EPANET modelitza un sistema de distribució d'aigua com una sèrie de línies connectades a nodes. Les línies representen canonades (*pipes*), bombes (*pumps*), i vàlvules (*valves*) de control. Els nusos representen connexions (*junctions*), tancs (*tanks*) i dipòsits (*reservoirs*). La figura de sota separa per colors cada part de la xarxa i ens mostra com aquests objectes poden ser connectats per representar una xarxa.

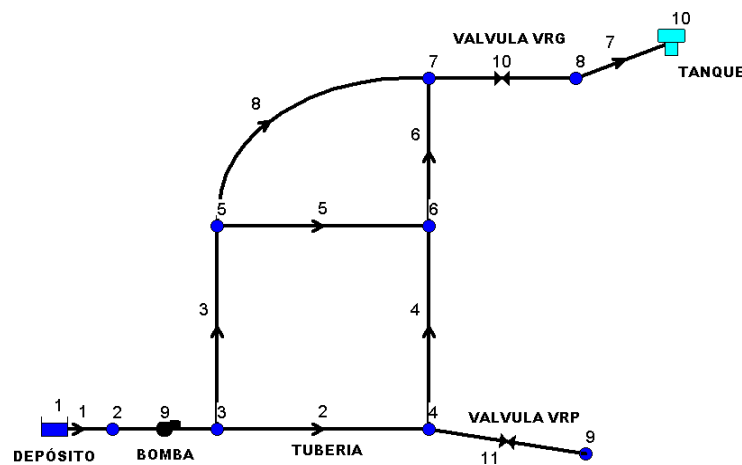


Figura 4. Components físics en un sistema de distribució d'aigua [Font 9]

Connexions

Les connexions són punts a la xarxa on s'uneixen les línies o per on entra o surt l'aigua de la xarxa. La informació bàsica que es requereix per les connexions es:



- Cota, normalment sobre el nivell del mar.
- Demanda d'aigua (règim d'evacuació des de la xarxa).
- Qualitat de l'aigua inicial.

Els resultats que s'obtenen de les connexions al llarg de tota la simulació que realitzaríem són:

- Altura piezomètrica (energia interna per unitat de massa del fluid).
- Pressió.
- Qualitat de l'aigua.

Les connexions també poden:

- Tenir una demanda que variï amb el temps.
- Tenir diferents categories de demanda assignades.
- Tenir una demanda negativa indicant que l'aigua entra a la xarxa.
- Ser font de qualitat de l'aigua per on els constituents entren a la xarxa.
- Contenir emissors (o aspersors) fent que el caudal descarregat depengui de la pressió.

Dipòsits

Els dipòsits són un tipus de node que representa una font externa infinita o una espècie d'embornal per el sistema. Són utilitzats per modelitzar llacs, rius, aqüífers subterranis i connexions a altres sistemes. Els dipòsits també poden servir com a font de qualitat de l'aigua.

Les principals característiques d'un dipòsit és l'altura per d'amunt del nivell del mar si no es troba sota pressió i la seva qualitat inicial per l'anàlisi de qualitat de l'aigua.

Com que el dipòsit és considerat un punt frontera de la xarxa, la seva altura i qualitat de l'aigua no poden veure's afectades per el que passi en la resta del sistema. Per lo tant, no s'ordenen propietats o valors de sortida. Malgrat tot això, es pot variar la seva altura amb el temps si li assignem un patró de temps.

Tancs

Els tancs son nodes amb capacitat d'emmagatzematge, on el volum d'aigua emmagatzemada pot variar en el temps al llarg de la simulació. Les característiques principals pels tancs són:

- Cota (on el nivell d'aigua és zero).
- Diàmetre (o geometria si no és cilíndric).
- Valors inicials màxims i mínims del nivell de l'aigua.
- Qualitat de l'aigua inicial.

Els principals valors que es demanen al llarg del temps són:

- Nivell de la superfície lliure d'aigua.
- Qualitat de l'aigua.

Els tancs operen limitats pels nivells mínims i màxims, lògicament. EPANET atura l'aportació de cabal si el nivell del tanc està al mínim i atura el consum de cabal si el nivell del tanc es troba en el seu punt màxim. Els tancs també poden servir com a font de qualitat d'aigua.

Emissors

Els emissors són mecanismes associats a les connexions que modelitza l'alliberament de cabal a l'atmosfera a través d'uns orificis. El cabal que travessa l'emissor varia en funció de la pressió del node:

$$q = cp^\gamma$$

On q = cabal, p = pressió, C = coeficient de cabal (descarrega), y γ = exponent de la pressió. Pels polvoritzadors i ruixadors γ és igual a 0.5 i el fabricant normalment dona el valor del coeficient de cabal en unitats de cabal/pressió^{0.5} i representa el flux a través del mecanisme que genera una pèrdua unitària elevada al quadrat.

Els emissors són utilitzats per modelitzar el cabal a través de ruixadors o sistemes de reg a la xarxa. També poden ser utilitzats per modelitzar pèrdues en les

connexions entre canonades (efectivament, el coeficient de descarrega i l'exponent de pressió de trencament de la connexió o junta poden estimar-se) o per modelitzar una boca d'extinció de focs (assegurar un cabal disponible a una pressió mínima). En aquest últim cas es pot utilitzar un coeficient de descarrega molt alt (per exemple, 100 vegades el cabal màxim esperat) i modificar l'altura de la connexió per incloure la càrrega equivalent deguda a l'existència d'aquesta boca antiincendis. EPANET considera els emissors com una característica de les connexions i no com un element més a la xarxa.

Canonades

Les canonades són línies que porten l'aigua d'un punt de la xarxa a un altre. EPANET assumeix que totes les canonades es troben completament plenes en tot moment. La direcció del cabal va des de l'extrem amb altura piezomètrica (energia intern per unitat de pes de l'aigua) major fins l'extrem de la conducció amb menor altura, seguint sempre el sentit de la disminució de l'altura. Els paràmetres hidràulics més importants per la canonada són:

- Nodes d'entrada i sortida.
- Diàmetre.
- Longitud.
- Coeficient de rugositat (per determinar pèrdues).
- Estat (oberta, tancada o amb una vàlvula).

Els paràmetres d'estat de les canonades es fan servir per contemplar elements tals com vàlvules de tall o seccionament o vàlvules de retenció. Les característiques de qualitat de l'aigua per les canonades són les següents:

- Coeficient de reacció del flux (*bulk reaction coefficient*).
- Coeficient de reacció de paret (*wall reaction coefficient*).

Aquests coeficients s'explicaran més detingudament en els següents apartats.

Els principals valors que es poden obtenir són:

- Cabal.
- Velocitat.
- Pèrdues.
- Factor de fricció de Darcy-Weisbach.
- Variació de la velocitat de reacció (al llarg de la seva longitud).
- Variació de la qualitat de l'aigua (al llarg de la seva longitud).

Les pèrdues de carrega en la conducció degut a la rugositat de les parets de la canonada poden mesurar-se utilitzant tres mètodes diferents:

- Mètode de Hazen-Williams
- Mètode de Darcy-Weisbach
- Mètode de Chezy-Manning

El mètode de Hazen-Williams és el més utilitzat en l'àmbit anglosaxó. No es pot usar amb un altre líquid que no sigui aigua i va ser desenvolupada originalment únicament per règim turbulent. El mètode de Darcy-Weisbach és el més exacte teòricament. Es pot aplicar per qualsevol líquid en qualsevol règim que es trobi. El mètode de Chezy-Manning és el més utilitzat per conductes funcionant en làmina lliure.

Cada mètode utilitza la següent equació per calcular les pèrdues entre el node inicial i el node final de la canonada:

$$h_L = A \cdot q^B$$

On h_L = pèrdues (longitud), q = cabal (Volum/Temps), A = coeficient de resistència, i B = exponent del cabal. La taula 3 conté les expressions pel càlcul del coeficient de resistència i els valors de l'exponent del cabal per cada un dels mètodes. Cada mètode utilitza un coeficient de rugositat diferent que ha de ser estimat empíricament. La taula 4 conté valors típics d'aquests coeficients per diferent tipus de materials. S'ha de tenir en compte que el coeficient de rugositat d'una canonada al llarg del temps.

Amb el mètode de Darcy-Weisbach, EPANET utilitza diferents expressions corresponents a les diferents zones en les que el factor de fricció f presenta comportaments diferents. Així, depenent del règim de treball:

- Per règim laminar ($Re < 2.000$) utilitza la fórmula de Hagen-Poiseuille.
- Per règim turbulent ($Re < 4.000$) utilitza l'aproximació de Swamee i Jain de l'equació de Colebrook-White.
- Per flux transitori ($2.00 < Re < 4.000$) utilitza una interpolació cúbica del àbac de Moody.

<i>Formula</i>	<i>Coefficiente de Resistencia (A)</i>	<i>Exponente de Caudal (B)</i>
Hazen-Williams	$4.727 C^{-1.852} d^{-4.871} L$	1.852
Darcy-Weisbach	$0.0252 f(\epsilon, d, q) d^{-5} L$	2
Chezy-Manning	$4.66 n^2 d^{-5.33} L$	2

Notas: C = coeficiente de rugosidad de Hazen-Williams
 ϵ = coeficiente de rugosidad de Darcy-Weisbach (ft)
 f = factor de fricción (depende de ϵ , d, y q)
 n = coeficiente de rugosidad de Manning
 d = diámetro de la tubería (ft)
 L = longitud de la tubería (ft)
 q = caudal (cfs)

Taula 3. Coeficient de resistència per a cada mètode [Font 9]

<i>Material</i>	<i>Hazen-Williams C (adimensional)</i>	<i>Manning's n (adimensional)</i>
Hierro Colado	130 – 140	0,012 - 0,015
Hormigón o Revestido de Hormigón	120 – 140	0,012 - 0,017
Hierro galvanizado	120	0,015 - 0,017
Plástico	140 – 150	0,011 - 0,015
Acero	140 – 150	0,015 - 0,017
Arcilla Vitrificada	110	0,013 - 0,015

Taula 4. Coeficient de rugositat per diferents materials [Font 9]

<i>Material</i>	<i>Darcy-Weisbach ϵ (mm)</i>
PVC y PE	0,0025
Fibroemento	0,025
Fundición revestida	0,03
Fundición no revestida	0,15
Hormigón armado	0,1
Hormigón liso	0,025

Taula 5. Guix dels diferents materials [Font 9]



Les canonades poden obrir-se o tancar-se en determinats instants de temps o quan existeixin unes condicions específiques, com per exemple quan el nivell del tanc cau o arriba a un punt determinat, o quan la pressió en els nodes cau o arriba a un valor determinat.

Pèrdues menors

Les pèrdues menors (o pèrdues locals) es deuen a l'existència de turbulències en cantonades i connexions. La importància d'incloure aquestes pèrdues depèn de la distribució en planta de la xarxa i el grau d'exactitud requerit. Poden comptabilitzar-se assignant a la canonada un coeficient de pèrdues menors. Les pèrdues menors es calculen multiplicant aquest coeficient per l'energia cinètica d'entrada a la canonada, és a dir:

$$h_L = k \left(\frac{v^2}{2 \cdot g} \right)$$

On K = coeficient de pèrdues menors, v = velocitat del flux (longitud/temps), y g = acceleració de la gravetat (longitud/temps). La taula 6 mostra pèrdues menors d'alguns connectors.

<i>Conectores</i>	<i>Coeficiente de pérdidas</i>
Válvula de Globo, totalmente abierta	10.0
Válvula de Ángulo, totalmente abierta	5.0
Válvula de Retención de Clapeta, totalmente abierta	2.5
Válvula de Compuerta, totalmente abierta	0.2
Codo de radio pequeño	0.9
Codo de radio medio	0.8
Codo de radio grande	0.6
Codo a 45°	0.4
Codo cerrado con inversión del flujo	2.2
Te estándar – dirección de paso	0.6
Te estándar – dirección desvío	1.8
Entrada Recta	0.5
Salida brusca	1.0

Taula 6. Coeficients de pèrdues menors [Font 9]



Bombes

Les bombes són elements que aporten energia al fluid incrementant la seva altura piezomètrica. Les característiques més importants per una bomba són la seva entrada, la sortida i la seva corba característica (relació entre l'altura i cabal de la bomba).

Els paràmetres de sortida més importants són el cabal i la càrrega. El cabal flueix de manera unidireccional dins de la bomba i EPANET no permet que una bomba treballi fora del rang que determina la seva corba característica.

També es pot considerar la variació de velocitat de les bombes, la qual pot fixar-se i modificar-se sota les condicions descrites anteriorment. Per definició, la corba característica que aporta el programa té un connector de velocitat igual a 1. Si es dobla la velocitat de la bomba. Llavors el corrector (*relative setting*) serà de 2; si la velocitat és igual a la meitat, el corrector tindrà el valor de 0.5. canviant la velocitat es modifica la posició i la curvatura de la corba característica.

De la mateixa manera que les bombes poden activar-se (arrenca) i desactivar-se (desconnectar-se) en determinats moments establerts per l'usuari o bé quan existeixen certes condicions a la xarxa. El mode d'operació de la vàlvula pot descriure's assignant un patró de temps amb els correctors de velocitat. EPANET també pot tenir en compte el consum d'energia i el cost de la bomba. A cada bomba se li pot assignar una corba de eficiència i una llista de preus d'energia. Si no es disposa d'aquests valors llavors s'utilitzen unes condicions energètiques generals.

El cabal que travessa una bomba és unidireccional. Si les condicions del sistema requereixen que la bomba treballi fora de les seves possibilitats, EPANET intentarà desconnectar-la. Si el sistema requereix un cabal major que el que pot trasbalsar, EPANET extrapolarà una corba pel cabal requerit tot i que produeixi una altura negativa. Ambdós casos apareixerà un missatge d'avertència.

Vàlvules

Les vàlvules són línies que limiten la pressió i el cabal en punts específics de la xarxa. Els seus principals paràmetres característics són:

- Nodes d'entrada i sortida
- Diàmetre
- Consigna (tarada)
- Estat



Components no físics

A més dels components físics, EPANET utilitza tres tipus d'objectes informatius; corbes, patrons y controls. Aquests descriuen el comportament i els aspectes operacionals d'un sistema de distribució.

Corbes

Les corbes són objectes que representen la relació existent entre dos dades per mitjà de dos magnituds o quantitats. Dos o més objectes poden formar part de la mateixa corba. Un model de EPANET pot utilitzar els següents tipus de corbes:

- Corba de característica d'una bomba (*Pump Curve*).
- Corba de rendiment (*Efficiency Curve*).
- Corba de volum (*Volume Curve*).
- Corba de pèrdues (*Head Loss Curve*).

Corba característica d'una bomba

La corba característica representa la relació entre l'altura i el cabal que pot desenvolupar a la seva velocitat nominal. L'altura és l'energia que la bomba aporta a l'aigua i es representa en l'eix vertical (Y) en metres (peus). El cabal es representa en l'eix horitzontal (X) en unitats de cabal. Una corba característica vàlida ha de disminuir l'altura a mesura que augmenta el cabal.

EPANET utilitzarà diferents tipus de corbes característiques depenent del número de punts subministradors. En les següents figures es veuen les diferents corbes possibles:

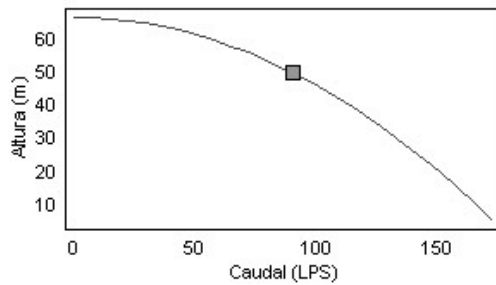


Figura 5. Corba característica en un punt [Font 9]

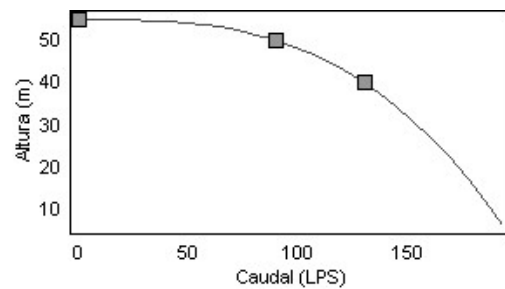


Figura 6. Corba característica de 3 punts [Font 9]

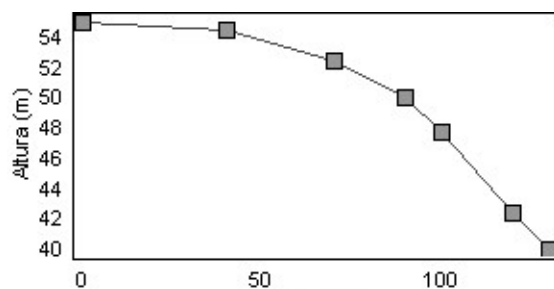


Figura 7. Corba característica de multipunt variable [Font 9]

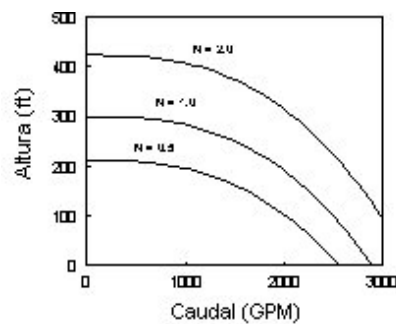


Figura 8. Corba característica de velocitat [Font 9]

Corba de rendiment

La corba de rendiment determina el rendiment de la bomba (eix Y en tant per cent) com a funció del cabal de la bomba (eix X en unitats de cabal). Un exemple de corba de rendiment es mostra en la figura 8. Representa el rendiment total, és a dir, contempla tant el rendiment mecànic de la bomba com el rendiment elèctric del motor de la mateixa. Aquesta corba s'utilitza únicament per càlculs energètics. Si no determinem la corba de rendiment per una bomba, el programa utilitzarà una corba de rendiment genèric prefixat.

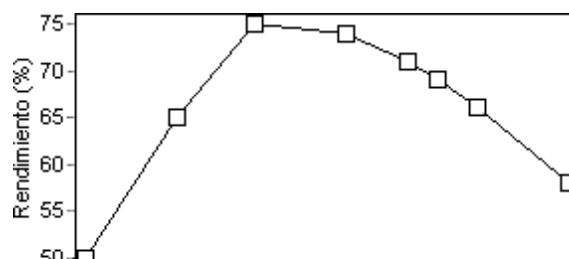


Figura 9. Corba de rendiment d'una bomba [Font 9]

Corba de volum

La corba de volum determina com el volum d'aigua en tanc (eix Y en metres cúbics o peus cúbics) varia en funció del nivell de l'aigua (eix X en metres o peus). S'utilitza quan és necessari representar exactament tancs llurs secció transversal varia amb l'altura. Els valors mínims i màxims de nivells d'aigua representats per la corba han de ser els nivells mínim i màxim entre els que treballa el tanc. Un exemple de la corba de volum d'un tanc es mostra a continuació (figura 9).

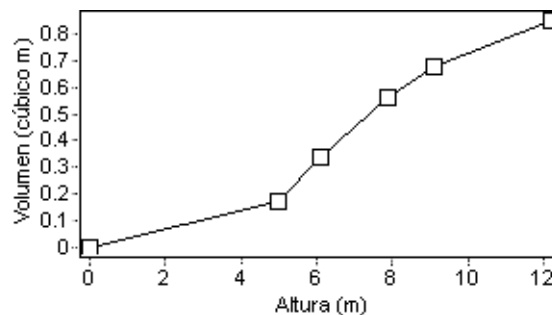


Figura 10. Corba de volum d'un tanc
[Font 9]

Corba de pèrdues

La corba de pèrdues s'utilitza per representar les pèrdues (eix Y en unitats de pressió, peus o metres) en una vàlvula de propòsit general (VPG) en funció del cabal (eix X en unitats de cabal). Això ens dona la oportunitat de modelitzar dispositius i situacions amb una relació de pèrdues-cabal específic, tals com vàlvules de control de flux o de control de flux invers, turbines i descens dinàmic del nivell de pous.

[9] [11]

2.4. GIS

Un GIS (*Geographical Information System*) també conegut com SIG (Sistema d'Informació Geogràfica), és una eina indispensable pel tractament de dades que ens permet analitzar-les i representar-les de múltiples maneres. Aquest instrument també ens permet recopilar i gestionar les nostres dades per poder representar-les utilitzant mapes i escenes 3D que ajuden a tenir la nostre informació més clara i fàcil d'ubicar.



Per tant, un GIS és un sistema de base de dades amb capacitats específiques per dades geo-referenciades, i al mateix temps, té tot un conjunt d'operacions per treballar amb les dades esmentades. De certa manera, es pot dir que és un mapa d'ordre superior.

De la mateixa manera que els textos han passat del paper cap a l'ordinador, els mapes també han donat aquest salt qualitatiu amb l'aparició de GIS. Però el GIS no només és una nova forma de cartografiar, va més enllà. A part de totes les funcions que es poden programar, existeix també la capacitat de crear mapes en paper des del nostre ordinador. Això fa que GIS sigui una eina integradora i completa per tasques d'informació geo-referenciada.

Actualment, en el món de la informació espacial i del mapeig, moltes organitzacions nacionals i internacionals estan utilitzant GIS per fer mapes que es comuniquen, realitzen anàlisis, comparteixen informació i resolen problemes complexos.

Una manera de comprendre un GIS és visualitzar-lo com una sèrie de subsistemes, cada un d'ells encarregat de funcions específiques. Se sol dividir en tres subsistemes fonamentals:

- Subsistema de dades: encarregat de les operacions d'entrada i sortida de les dades i de la seva gestió dins del GIS. Permet que la resta de subsistemes tinguin accés a les dades i puguin realitzar les seves funcions en base a aquestes.
- Subsistema de visualització i creació cartogràfica: s'encarrega de generar representacions (llegendes, mapes, etc.) a partir de les dades. També permet l'edició de les mateixes.
- Subsistema d'anàlisi. Conté mètodes i processos per l'anàlisi de dades geogràfiques.

[7]

2.5. QGIS

Existeixen tot una sèrie de GIS disponibles en l'actualitat. La comunitat lliure ha tingut una gran participació en el desenvolupament d'aquest sistema de software per sistemes d'informació geogràfica de **codi obert**. Aquí és on entra una de les eines més poderoses de software lliure, el QGIS.



QGIS (Quantum GIS) és un software GIS de codi lliure, amb la llicència GNU-GPL, desenvolupat per Open *Source Geospatial Foundation* (OSGeo) a C++, que permet la visualització, edició i anàlisi de dades geogràfiques.

QGIS permet la creació de mapes amb nombroses capes que poden ser programades en diferents formats, depenent de l'aplicació. Però les veritables característiques que fan de QGIS una eina potent són les següents:

- Tenen un suport per l'extensió espacial de SpatialLite, ORACLE Spatial i PostGIS, que afegeix suport a objectes geogràfics en una base de dades, convertint-la en una base de dades espacial.
- Permet la creació de mapes a través de capes raster (cel·les) o de capes vectorials (línies i polígons), suportant nombrosos formats, Shapefile, ArcInfo, MapInfo, GRASS GIS, etc.
- Pot treballar sota qualsevol sistema operatiu.

Una de les característiques més sorprenents de QGIS es la seva capacitat per adaptar-se al usuari. Amb l'arquitectura de complement extensible i les llibreries amb les que compta és possible crear complements enfocats a la necessitats de cada projecte, permetent crear capes amb informació realment interessant en cada mapa. Inclús és possible crear noves aplicacions amb codi C++ i Python.

QGIS també serveix per publicar mapes utilitzant diversos tipus de dades del estàndard definit per la OGC (*Open Geospatial Consortium*), i així poder realitzar tant la funció de client, consumint les dades d'alguna font, com la de servidor, oferint a altres clients els mapes realitzats amb QGIS.

Existeixen una infinitat de complements (*plugins*) per QGIS, com per exemple QOSM, que permet obtenir les imatges del mapa de Open Street Map. Inclús es poden desenvolupar els teus propis complements utilitzant Python o C++ com a llenguatges de programació. Es tractarà aquest tema més àmpliament en el següent apartat.

[8]

2.6. COMPLEMENTS DE QGIS (PLUGINS)

Un *plugin* és una extensió que s'afegeix al programa principal de QGIS per proporcionar funcionalitats addicionals i ampliar les seves capacitats. Aquestes extensions són desenvolupades majoritàriament per la comunitat d'usuaris i programadors de QGIS i estan disponibles per ser descarregades i instal·lades des del seu repositori oficial (software lliure).



Els *plugins* són una de les característiques més destacades de QGIS, ja que permeten adaptar el programa a les necessitats específiques de cada usuari o projecte. Amb els *plugins*, és possible afegir noves eines, funcions i algorismes a QGIS sense necessitat de modificar el codi font original. Això proporciona una gran flexibilitat i personalització per als usuaris.

Els *plugins* per a QGIS es poden classificar en diferents categories segons la seva funcionalitat. Algunes de les categories més comunes se separen de la següent forma:

Per processament de dades: hi ha *plugins* que ofereixen algorismes addicionals per realitzar anàlisis espacials avançades, com ara interpolació, anàlisi de xarxes, anàlisi hidrològic, etc.

Per visualització: aquests *plugins* proporcionen eines per a la millora de la visualització de les dades espacials, com ara mòduls per a la creació de mapes temàtics, representació tridimensional, etc.

Per edició i digitalització: hi ha *plugins* que ofereixen eines per a la digitalització de dades geogràfiques, com ara línies, polígons o punts, i per a la realització de tasques d'edició com ara la fusió de capes o l'edició topològica.

Per connectivitat i integració: aquests *plugins* permeten la connexió de QGIS amb altres fonts de dades i programes externs, com ara bases de dades espacials, serveis web de mapes, *Google Earth*, entre d'altres.

Per anàlisi especialitzada: aquesta categoria inclou *plugins* desenvolupats per a tasques d'anàlisi especialitzada, com ara l'anàlisi de la vegetació, l'avaluació de riscos, l'estudi d'hàbitats, etc.

2.6.1. GISWATER

Es tracta d'un *plugin* (extensió) de QGIS i és una eina que també està basada en software lliure creada per treballar de forma integral en les àrees relacionades amb l'abastiment, el sanejament, el drenatge urbà o l'estudi de riscos d'inundació. Actua com enllaç entre programes de **simulacions hidràuliques** (EPANET), bases de dades espacials i sistemes d'informació geogràfica -SIG o GIS- amb la finalitat d'oferir una solució de format obert en la gestió de l'aigua.



GISWATER permet utilitzar de forma integrada Serveis Web de Mapes (WMS), això permet visualitzar capes d'informació geogràfica de fonts externes directament dins de la interfície de QGIS. També facilita l'obtenció de dades actualitzades i rellevants per realitzar anàlisis i presa de decisions.

Permet utilitzar també sistemes de Control y Adquisició de Dades (SCADA) o Sistemes de Relació amb Clients (CRM). SCADA permet l'adquisició de dades en temps real de sensors i dispositius de control distribuïts a la xarxa hidràulica. Aquesta funcionalitat facilita la supervisió i gestió en temps real dels sistemes de subministrament i sanejament, millorant la capacitat de resposta davant d'incidències o esdeveniments imprevistos.

Al mateix temps, GISWATER també actua com a *driver* de connexió entre la base de dades espacial i eines d'anàlisi hidràulic.

Tal i com es pot apreciar en la figura 1, GISWATER s'emmarca dins d'una agrupació d'aplicacions que conjuntament utilitzats permeten una gestió global i sòlida per poder treballar sobre models d'abastiment d'aigües i drenatges urbans.

L'element central d'aquest conjunt és la base de dades, on es troba tota la informació i gran part de les funcionalitats de cada projecte que es crea amb aquest programa. La base de dades per utilitzar GISWATER es PostgreSQL, que juntament amb la seva extensió PostGis permet enllaçar còmodament tot el conjunt:

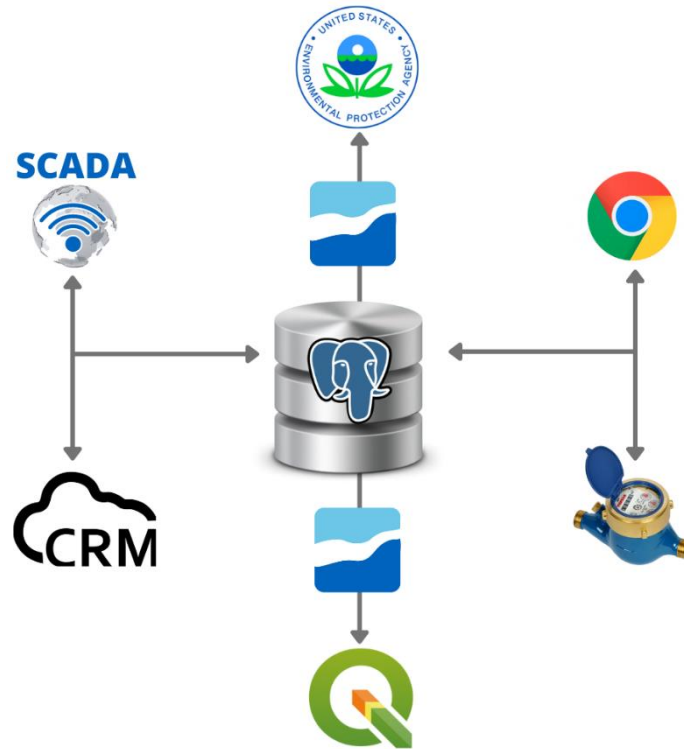


Figura 11. Conjunt de programes necessaris per treballar amb GISWATER i com es classifiquen [Font 5]

El QGIS és el sistema d'informació geogràfica del qual s'ha parlat anteriorment. Aquest programa és amb el que es basa GISWATER pel desenvolupament del seu projecte, sobretot en la branca visual de major operativitat per part del usuari. QGIS es relaciona mitjançant PostGIS amb la base de dades, mostrant de forma organitzada les dades espacials.

L'eix principal d'aquest projecte (Base de dades- GIS) també permet connectar amb SCADA, per actualitzar en temps real la informació que prové directament dels elements físics de la xarxa. D'aquesta manera, GISWATER es presenta com un sistema de gestió global i que dona la possibilitat als seus usuaris de treballar sempre amb dades que s'actualitzen automàticament.

Una altra avantatge de GISWATER és, com s'ha esmentat anteriorment, el seu enfocament en l'ús d'estàndards oberts, lo qual fomenta la interoperabilitat amb altres sistemes i la reutilització de dades i funcionalitats. Això facilita la col·laboració entre diferents actors involucrats en la gestió de l'aigua, com ara administracions públiques (Ajuntament), empreses de serveis i consultors.



Cal esmentar també que GISWATER no només ofereix capacitats de gestió de dades en l'entorn del programa de QGIS per a escriptori (PC), sinó que també proporciona opcions per treballar amb les dades a través d'una interfície web i aplicacions mòbils. Això ofereix als usuaris la flexibilitat d'accedir i actualitzar la informació des de diferents dispositius i ubicacions, la qual cosa resulta especialment útil per al personal de camp que necessita realitzar inspeccions i recopilar dades in situ.

[5]

Una gran avantatge de tenir el *plugin* de GISWATER i les seves extensions realitzades internament amb el llenguatge de programació *python* és la seva flexibilitat i facilitat d'ús. *python* és un llenguatge de programació popular i àmpliament utilitzat en el camp de la ciència i l'enginyeria, i ofereix una sèrie d'avantatges significatives.

En primer lloc, *python* és un llenguatge de programació de codi obert i té una gran comunitat de desenvolupadors. Això significa que hi ha una gran quantitat de recursos, biblioteques i mòduls disponibles per als desenvolupadors que treballen amb *python*. Aleshores, permet als desenvolupadors extreure el màxim rendiment de les eines i recursos disponibles per millorar i ampliar les funcionalitats del *plugin* de GISWATER i les seves extensions.

En segon lloc, *python* és conegut per la seva sintaxi clara i llegible, la qual cosa facilita la comprensió i el manteniment del codi. Això és especialment important en el desenvolupament de *plugins/extensions*, ja que permet als desenvolupadors treballar de manera eficient i col·laborar en equips de treball.

A més, *python* ofereix una gran varietat de biblioteques i eines específiques per a l'anàlisi i la manipulació de dades, com ara NumPy, Pandas i Matplotlib. Això és fonamental en el camp de la geoinformàtica i l'anàlisi de dades geoespacionals, ja que permet als desenvolupadors gestionar, processar i visualitzar dades geogràfiques amb facilitat.

Finalment, *python* és multi plataforma, lo qual significa que el *plugin* de GISWATER i les seves extensions desenvolupades en *aquest llenguatge* poden ser executades en diferents sistemes operatius com Windows, macOS i Linux, sense haver de fer grans adaptacions.

2.6.2. EPATOOLS

Epatools és una extensió del propi *plugin* de Giswater que s'utilitza per a l'anàlisi i el disseny de xarxes de subministrament d'aigua potable. Proporciona eines i funcionalitats addicionals per a la simulació hidràulica i la gestió eficient de les xarxes d'aigua.

Les funcions principals de epatools inclouen:

Simulació hidràulica: epatools permet simular el comportament hidràulic de les xarxes de subministrament d'aigua, incloent-hi el flux, la pressió i altres paràmetres importants. Això permet als usuaris comprendre millor el funcionament de les xarxes i identificar problemes com ara mancances de pressió, sobrepressions o zones amb un control insuficient del cabal, etc.

Optimització del disseny: L'extensió ofereix eines per a l'optimització del disseny de les xarxes de subministrament d'aigua. Això inclou la possibilitat d'avaluar diverses alternatives de disseny per identificar la més eficient en termes de costos, eficiència hidràulica i qualitat de l'aigua.

Gestió de l'energia: L'extensió ofereix eines per a la gestió eficient de l'energia en les xarxes de subministrament d'aigua, permetent l'optimització dels sistemes de bombament i la reducció del consum energètic.



3. PROCEDIMENT EXPERIMENTAL

3.1. Introducció prèvia

Per iniciar amb la modelització del clor en la nostre xarxa de distribució, hem de començar entenent quina funcionalitat té aquest producte químic, com es tracte tan qualitativament com quantitativament. En aquest apartat previ, s'ha investigat segons la normativa governamental de quina manera es tolera aquest component per a que al final l'aigua de consum pugui ser apte per el cos humà. (informació extreta de la Generalitat de Catalunya).

3.1.1. Normativa del clor

Es consideren aigües de consum humà:

- Les subministrades per a beure, cuinar, preparar aliments o altres usos domèstics mitjançant xarxes de distribució o cisternes.
- Les subministrades per a beure, cuinar, preparar aliments o altres usos higiènics en activitats comercials o públiques:
 - Establiments turístics i de lleure (hotels, càmpings, albergs, cases de colònies, etc.)
 - Centres d'ensenyament
 - Centres socio-sanitaris (geriàtrics, hospitals, etc.) o penitenciaris
 - Locals públics (centres comercials, edificis administratius)
 - Habitatges de lloguer
 - Establiments de restauració comercial i social.
- Les utilitzades en indústries alimentàries per a fins de fabricació, tractament, conservació o comercialització de productes o substàncies destinats al consum humà i les que s'utilitzen per a la neteja de superfícies objectes i materials que puguin estar en contacte amb els aliments.

Restaran excloses d'aquesta consideració les aigües de les indústries alimentàries i dels establiments de restauració comercial i social, destinades als usos següents:

- Els establerts per la normativa (producció de vapor, calderes, etc.)
- Les zones exteriors de les indústries
- Els corrals
- El rentat de vehicles de transport d'animals
- Aquells que disposin d'autorització expressa d'acord amb la valoració del Pla d'autocontrol basat en el sistema APPCC, sempre que la qualitat de les aigües utilitzades no afecti la salubritat del producte alimentari.

3.1.2. Criteris de qualitat

Les aigües que es destinen al consum humà han de tenir garantida una qualitat que permeti assegurar un nivell alt de protecció de la salut de les persones que les consumeixen.

A Catalunya, es considerarà que l'aigua és apta per al consum quan estigui desinfectada i no tingui cap tipus de microorganisme, paràsit o substància en una concentració que pugui suposar un perill per a la salut humana i quan els resultats analítics:

- Compleixin els valors establerts en les parts A i D de l'Annex I.
- Compleixin els valors establerts en la part B1 de l'Annex I o els valors autoritzats/excepcionals pel director general de Salut Pública.
- Compleixin els valors establerts en la part B2 de l'Annex I, d'acord amb la migració màxima dels productes comercials en contacte amb l'aigua de consum utilitzat, segons les especificacions d'ús del fabricant.
- Compleixin els valors dels paràmetres addicionals (Figura 12).

Annex	Núm.	Paràmetre	Valor
B	nota 7 del paràmetre 26	Clorit	700 µg/l
C	31	Bacteris coliformes	10 UFC en 100 ml
C	32	Recòmpte de colònies a 22° C	100 UFC en 1 ml a la sortida del tractament
C	34	Amoni	0,50 mg/l
C	35	Carboni orgànic total (COT)	7 mg/l
C	42	Manganès	400 µg/l
C	44	Oxidabilitat	5 mg/l
C	49	Terbolesa	1 UNF a la sortida del tractament de desinfecció 5 UNF en xarxa de distribució

Figura 12. paràmetres addicionals que determinen l'aptitud de l'aigua per al consum [Font 16]

Per tant, l'aigua que incompleix algun dels criteris definits per a valorar-ne l'aptitud s'ha de qualificar com a no apta per al consum humà.

En els supòsits que, tot i ser apta per al consum, un o diversos paràmetres superen els valors paramètrics (VP) de l'Annex I de l'RD 140/2003, cal indicar-ne de forma expressa el motiu:

- Si la mostra supera algun valor paramètric de la part C de l'Annex I de l'RD 140/2003, sense sobrepassar el valors de la Taula 1, s'ha d'indicar el paràmetre o paràmetres que els superen:
- Si l'aigua de la zona de subministrament on s'ha pres la mostra disposa d'autorització d'excepció per a algun paràmetre de la part B i el resultat analític es manté dins del marge de l'excepció, s'ha d'indicar el paràmetre excepcional:

S'entén que la qualificació de l'aigua correspon a la qualificació de la mostra quan la valoració és apta per al consum o quan els resultats analítics que determinen la no aptitud de l'aigua hagin estat confirmats, d'acord amb el que estableix l'article 27 de l'RD 140/2003 i l'apartat 3.2.2 d'aquest programa de vigilància.

En qualsevol cas, l'aigua destinada al consum humà ha de contenir els nivells suficients de clor lliure residual. Els nivells de clor que es consideren mínims són:

- 0,5 mg/l, a la sortida del tractament de desinfecció
- 0,2 mg/l, en la xarxa de distribució.

L'interval recomanat de clor lliure residual en la xarxa de distribució és 0,2-0,6 mg/l.

3.1.3. Incompliments

Es considera com a incompliment qualsevol resultat analític que impliqui la valoració de l'aigua com a no apta per al consum.

De tots els incompliments, cal prendre'n una nova mostra per a la seva confirmació abans de 24 hores després que aquest s'hagi detectat.

Resten exclosos d'aquesta confirmació els resultats analítics de paràmetres que són motiu d'un seguiment específic requerit pel Departament de Salut.

[16]

3.1.4. Anomalies

Es considerarà anomalia qualsevol resultat analític dels paràmetres de la part C, Annex I, de l'RD 140/2003 que no compleixi el VP fixat normativament.

Pel que fa als paràmetres de la Taula 1 d'aquest document, es consideraran anomalies els resultats compresos entre el VP de l'RD 140/2003 i el valor establert en la taula.

La primera vegada que es detecti una anomalia en una zona de subministrament cal que aquesta sigui confirmada. La confirmació de les anomalies posteriors serà necessària quan ho determinin els serveis territorials de Salut.

[16]



3.2. Implicacions ambientals

Avui en dia, és crucial considerar les implicacions del medi ambient en tots els processos industrials i d'enginyeria. A mesura que augmenta la consciència sobre el canvi climàtic i la degradació ambiental, és important adoptar pràctiques sostenibles per minimitzar l'impacte negatiu de les activitats humanes. Això implica reduir la contaminació, preservar els recursos naturals i complir amb les regulacions ambientals. Tenir en compte les implicacions del medi ambient també pot generar estalvis de recursos, millorar la imatge corporativa, fomentar la innovació i millorar la competitivitat de les organitzacions.

El procés de cloració utilitzat com a desinfectant en les xarxes de distribució d'aigua potable pot tenir implicacions en el medi ambient. Encara que el clor és un agent desinfectant eficaç i àmpliament utilitzat, el seu ús pot tenir alguns efectes secundaris, que s'han comentat en apartats anteriors. A continuació s'expliquen aquests efectes amb més precisió.

La formació de subproductes de desinfecció: quan el clor reacciona amb la matèria orgànica present a l'aigua, pot formar subproductes de desinfecció com ara els trihalometans (THM) i els àcids haloacètics (AHA). Aquests subproductes poden ser perjudicials per al medi ambient i la salut humana. Els THM, per exemple, s'han associat a riscos potencials per a la salut, com ara l'aparició de certs tipus de càncer.

L'impacte en els ecosistemes aquàtics: quan l'aigua tractada amb clor es descarrega a rius, llacs o cossos d'aigua naturals, pot afectar els ecosistemes aquàtics. El clor residual pot interactuar amb els organismes aquàtics i causar efectes adversos, com ara la mortalitat de peixos, invertebrats aquàtics i microorganismes beneficiosos presents en els sistemes aquàtics.

La contaminació del sòl i aigües subterrànies: si hi ha fuites o vessaments de clor durant el procés de desinfecció o a les instal·lacions de tractament de l'aigua, es pot produir contaminació del sòl i les aigües subterrànies circumdants. Això pot tenir efectes negatius en la qualitat de l'aigua subterrània i afectar els ecosistemes que en depenen.

Per **mitigar aquests impactes ambientals**, s'estan desenvolupant i promocionant alternatives més sostenibles a l'ús del clor en la desinfecció de l'aigua. Algunes d'aquestes alternatives inclouen l'ús de mètodes de desinfecció no químics, com ara la radiació ultraviolada (UV) o l'ozonització. Aquests mètodes eviten la formació de subproductes de desinfecció i redueixen els efectes negatius en el medi ambient.



És important destacar que, tot i que hi ha impactes ambientals associats a l'ús del clor, la desinfecció de l'aigua potable és **essencial** per garantir la protecció de la salut pública en controlar la propagació de malalties transmeses per l'aigua. Els beneficis de la desinfecció superen els impactes negatius, i és responsabilitat de les autoritats i organismes competents implementar pràctiques de desinfecció adequades que minimitzin els efectes en el medi ambient.

[12]

3.3. Decaïment del clor

El clor és utilitzat en tot el món com a desinfectant residual per a contrarestar la contaminació microbiana en els sistemes d'abastiment d'aigua potable. Tot i així, la concentració residual de clor decau a mesura que l'aigua recorre les canonades principals degut a la seva reacció amb la matèria orgànica natural de l'aigua (descomposició en massa) i amb els materials de la superfície interna de les canonades (descomposició de la paret).

Mantenir una concentració residual eficaç del desinfectant, i al mateix temps de subproductes de la desinfecció, és essencial per la seguretat de l'aigua (declarat per l'organització Mundial de la Salut, 2011). Per tant, la gestió de la concentració residual de clor dins dels límits definits en els sistemes d'aigua potable és una de les principals preocupacions de les empreses de serveis públics.[9]

3.3.1. Comportament matemàtic

Dins de la massa del fluid es produeixen reaccions químiques, que són funció de la concentració del producte afegit, de la taxa de reacció i del número d'ordre de la reacció.

	Expressió de velocitat	Expressió integrada
Reacció d'ordre zero	$R_c = \frac{d[c]}{dt} = -k$	$C = C_0 - kt$
Reacció de primer ordre	$R_c = \frac{d[c]}{dt} = -k[C]$	$C = C_0 \cdot e^{-kt}$

Reacció de segon ordre	$R_c = \frac{d[c]}{dt} = -k[C]^2$	$\frac{1}{C} - \frac{1}{C_0} = kt$
------------------------	-----------------------------------	------------------------------------

Taula 7. Equacions de reacció [Font 9]

Si $k < 0$, és una reacció de decaïment

Si $k > 0$, és una reacció de deformació

Tal i com hem comentat anteriorment, el clor a mesura que passa temps en una xarxa es va consumint al entrar en contacte amb certs elements amb els quals reacciona. El comentari que observem sota la taula vol dir que quan el coeficient k sigui menor que zero, estarem davant d'un decaïment perquè la concentració final de clor serà menor que la del principi, lògicament. I si és més gran de zero, llavors com que és positiva resultarà un creixement de la concentració d'aquest producte. Com que el clor al entrar en una xarxa entra automàticament en contacte amb tot el "medi" de l'aigua, aquest sempre decreixerà la seva concentració.

És per això que el clor és considerada una substància no conservativa. Pot ser degradada simplement pels processos naturals de auto-purificació i les seves concentracions es redueixen amb el temps.

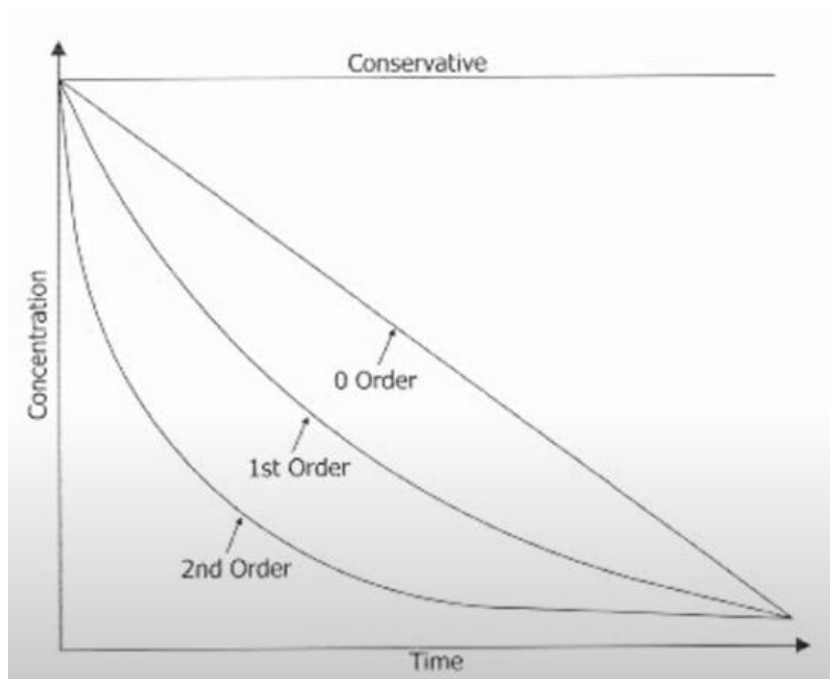


Figura 13. Comportament gràfic de les equacions [Font 9]



Aquestes son les corbes típiques, l'equació d'ordre 0 és una equació lineal, en canvi les equacions de 1r y 2n ordre son funcions exponencials. Aquesta és la mostra clara de decaïment tal i com es comporten les equacions, és la corba típica. En el temps zero és on existeix la concentració màxima i a mesura que va avançant el temps la concentració va disminuint.

3.3.2. Comportament físic

Els desinfectant com el clor, tal i com s'ha esmentat anteriorment, és una de les substàncies químiques més comunes que s'afegeixen al aigua. El comportament més habitual que tenen i sobre el qual s'estudia la seva conducta és el següent (veure figura 12):

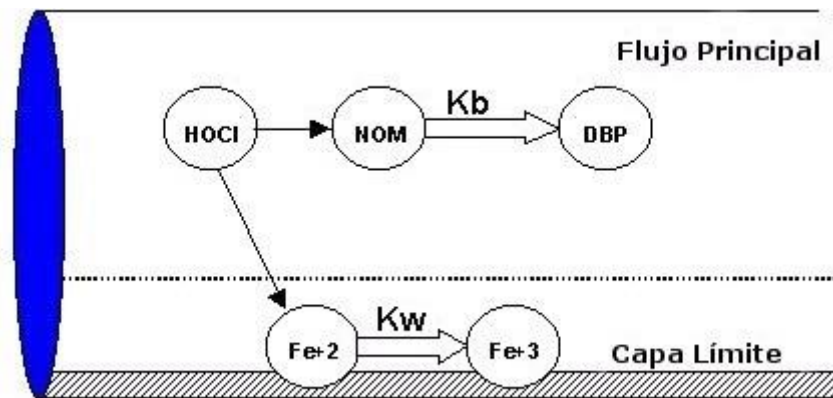


Figura 14. Diferència entre flux principal i capa límit [Font 9]

La K depèn de Kb (coeficient de reacció en massa o *bulk coefficient*) i de la Kw (coeficient en paret o *wall coefficient*).

Quan el clor entra en la xarxa de distribució, la seva concentració es consumirà de dos maneres: quan entra en contacte amb la pròpia massa de l'aigua i l'altre part de l'aigua es consumirà en entrar en contacte amb la superfície interna de la canonada. És a dir a través de l'aigua y a través de la canonada.

En les parets de la canonades solen haver-hi capes de matèria biològica o els anomenats *biofilms*, que vindria a ser el mateix. I inclús el propi material de la canonada també és un factor que fa reduir la concentració.

Aquests coeficients esmentats juguen un paper en la següent equació per determinar la constant K. Abans d'explicar-la cal tenir en compte que al final el programa EPANET el que fa és simplificar totes aquestes formules y oferir la possibilitat de configurar Kb y Kw per aplicar el decaïment que desitgem. Però interiorment el programa està utilitzant la següent formula:

$$k = k_b + \frac{k_w + k_f}{R_H(k_w + k_f)}$$

K_b: coeficient de reacció de massa, 1/T

K_w: coeficient de reacció de paret, L/T

K_f: coeficient de transferència de massa des de l'interior del fluid a la paret de la canonada, L/T

R_H: radi hidràulic, L

La velocitat de decaïment d'aquest desinfectant en la canonada també depèn de què tant ràpid arriba a les parets i de la velocitat de les reaccions que passen dins la canonada. Es podria resumir en que depèn del transcurs del temps, però també existeixen factor implicats com el tipus de líquid, la temperatura, el material de la canonada, etc. coeficient de transferència de massa des de l'interior del fluid a la paret de la canonada té en compte aquests fenòmens, y es calcula com:

$$K_f = \frac{S_H + d}{D}$$

S_H: número de Sherwood

d: difusivitat molecular del reactiu, L²/T

D: diàmetre de la canonada, L

El número Sherwood té valors diferents que depenen del tipus de règim del fluid (estacionari, turbulent o laminar). Per règim estacionari S_H=2.

[9]



3.3.3. Simulació amb EPANET

Un cop ha quedat clara la teoria, toca explicar la part experimental o pràctica del treball. A partir dels coneixements obtinguts s'ha fet una recerca per trobar una manera de modelitzar el decreixement del clor mitjançant algun software. EPANET és un programa que ja ve integrat amb la possibilitat de fer una simulació en la qual introdueixes un químic (que sol ser clor) en el dipòsit de la xarxa. La metodologia de com funciona cada part de la xarxa (dipòsits, nodes, vàlvules, etc) s'ha explicat en el capítol de EPANET. A continuació s'explicarà com s'introdueixen aquests paràmetres a nivell pràctic amb captures de pantalla de cada pas que es segueix. D'aquesta manera es mostrarà com es fa la simulació per obtenir el coeficient de decreixement amb aquest software.

Cal dir abans de començar que l'arxiu .inp que s'ha agafat per realitzar la simulació es tracta de la xarxa d'aigua que es troba en la zona de Manresa que agafa des de l'entorn de l'EPSEM (part alta del mapa) fins a L'Althaia (part baixa del mapa). L'arxiu té el nom de "elevada.inp" i el mapa és el següent:

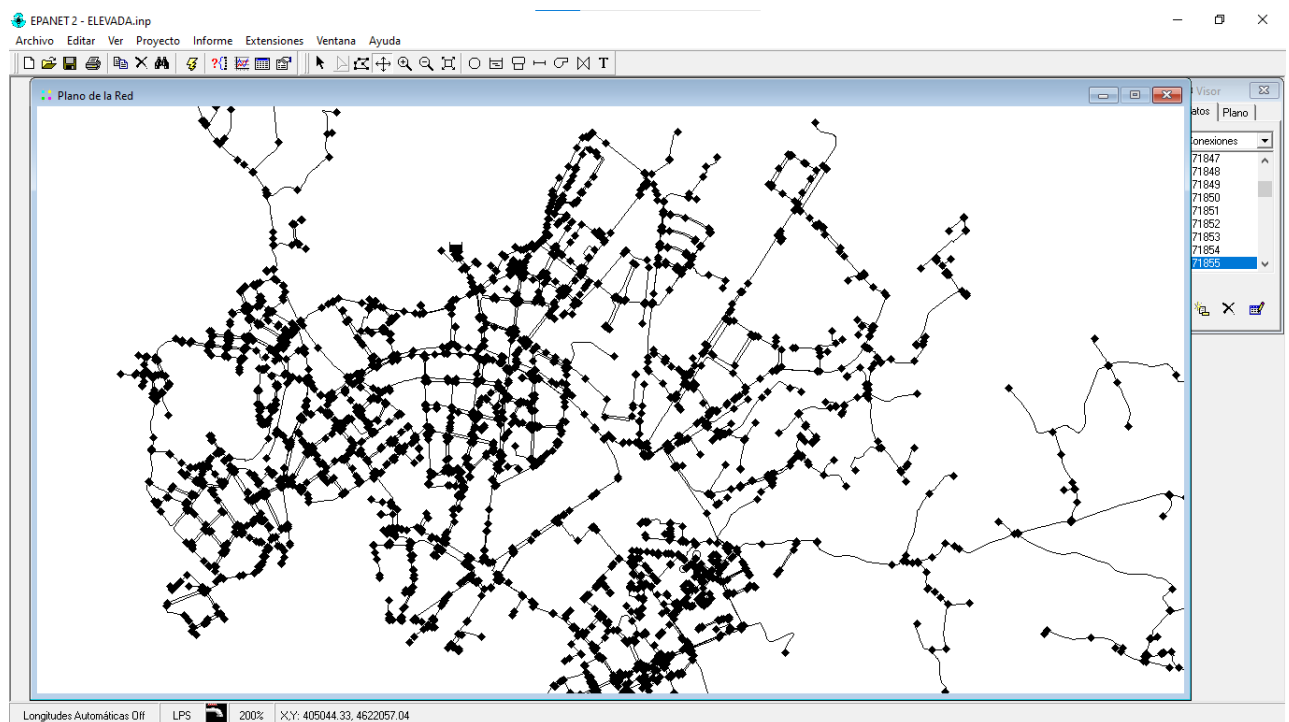
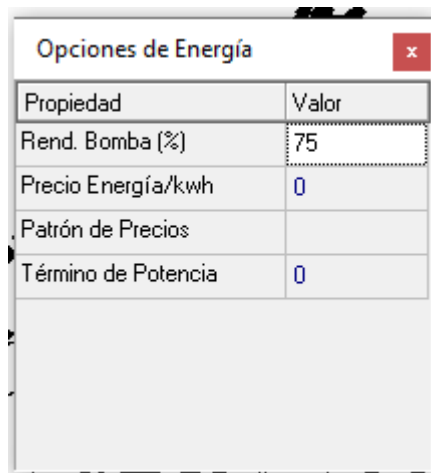


Figura 15. Arxiu inp per la simulació de EPANET (zona EPSPSEM) [Font pròpia]

Per fer la simulació del decreixement del clor es configuren els següents paràmetres:

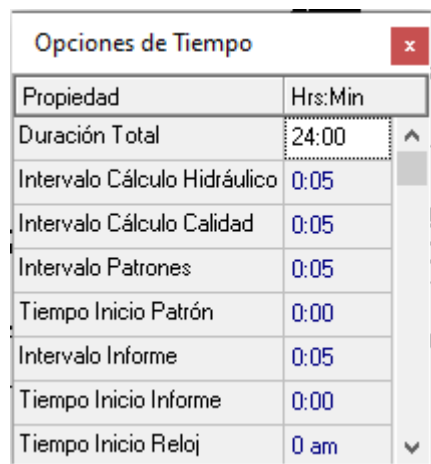
Valors per defecte:



Propiedad	Valor
Rend. Bomba (%)	75
Precio Energía/kwh	0
Patrón de Precios	
Término de Potencia	0

Figura 16. Rendiment de la bomba [Font pròpia]

Temps a 1 dia tot i que es pot allargar per confirmar que la concentració va disminuint:



Propiedad	Hrs:Min
Duración Total	24:00
Intervalo Cálculo Hidráulico	0:05
Intervalo Cálculo Calidad	0:05
Intervalo Patrones	0:05
Tiempo Inicio Patrón	0:00
Intervalo Informe	0:05
Tiempo Inicio Informe	0:00
Tiempo Inicio Reloj	0 am

Figura 17. Temps transcorregut [Font pròpia]



Propiedad	Valor
Parámetro	Ninguno
Unidades Densidad	mg/L
Difusividad Relativa	1
Nudo Procedencia	
Tolerancia de Calidad	0.01

Figura 18. Unitats de l'agent químic (clor) [Font pròpia]

Els *patterns* (patrons) serveixen per definir el rendiment inicial que tindran els nodes, va molt bé aplicar diferents patrons per veure com reacciona al sistema d'avant de canvis ja coneguts:

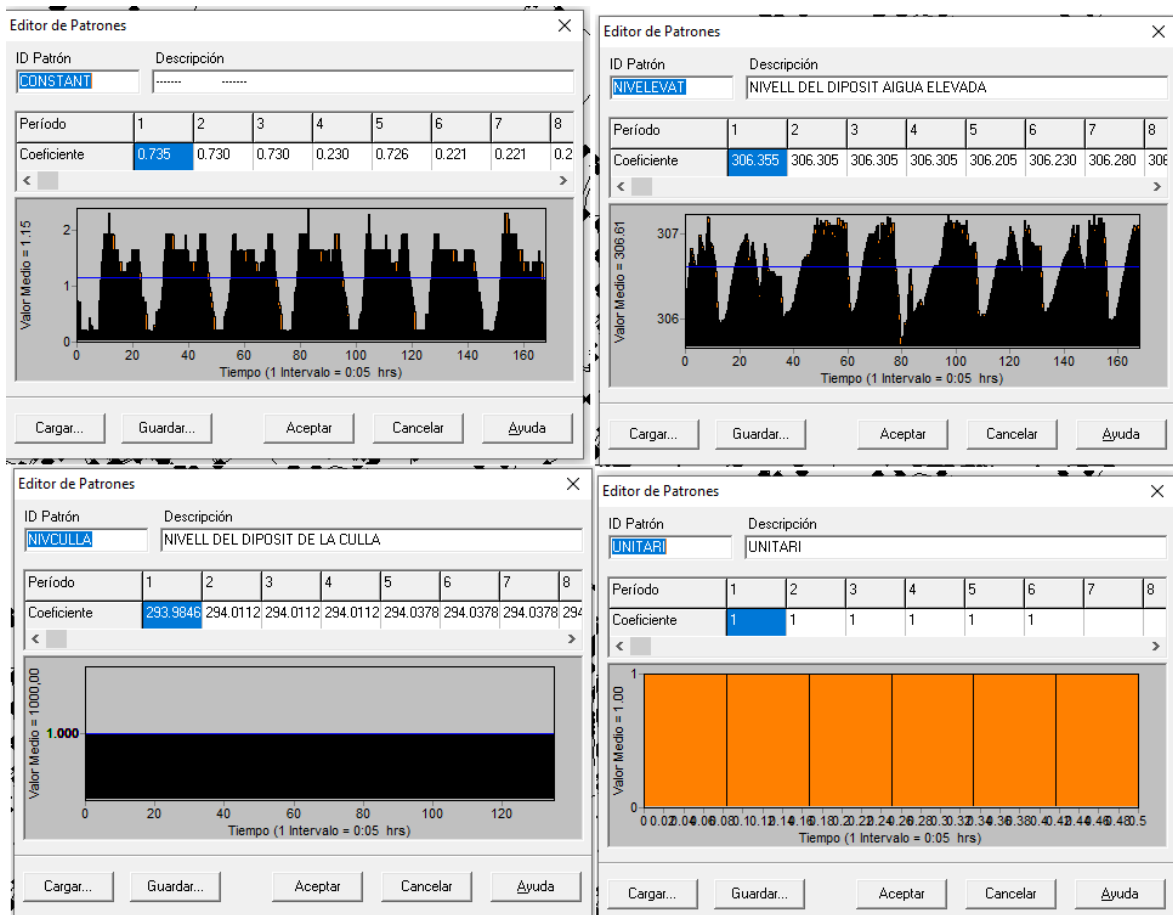


Figura 19. Diferents configuracions de pattern [Font pròpia]

Configuració de K_b i K_w :

Propiedad	Valor
Orden Reacción Flujo	1
Orden Reacción Pared	Primer
Coef. Global Flujo	0.3
Coef. Global Pared	0.5
Concentración Límite	0
Coef. Correlación Pared	0

Figura 20. Configuració K_b i K_w [Font pròpia]

Paràmetre (substància) i concentració inicial:

Propiedad	Valor
Coordenada-X	402695,55
Coordenada-Y	4621576,52
Descripción	
Etiqueta	
*Altura Total	1
Patrón de Altura	NIVELEVAT
Calidad Inicial	2
Fuente de Calidad	

Figura 21. Opcions de qualitat per aplicar concentració inicial de clor [Font pròpia]

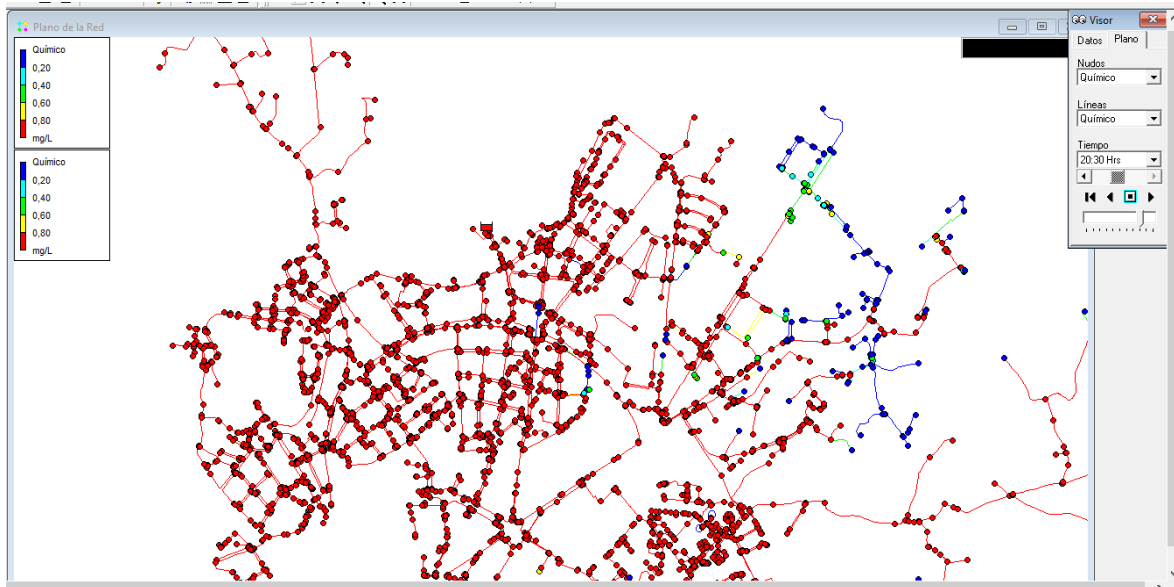


Figura 22. Simulació d'exemple [Font pròpia]

3.3.4. Relació entre els emissors i el decreixement del clor

Tal i com s'ha explicat en aquest treball, els emissors y el decreixement de clor es comporten de la mateixa manera matemàticament parlant. Tots dos funcionen seguint l'equació de primer orde: $x = x_0 \cdot y^{\gamma}$.

Això fa pensar que si existeix un algoritme capaç de poder calcular els coeficients emissors, podrem fer servir aquest procediment per aplicar-ho al càlcul del coeficient de decreixement del clor. Efectivament existeix aquest algoritme, es tracta del emitter calibration, una extensió de epatools (*plugin* que forma part de GISWATER) que realitza el càlcul dels coeficients emissors. Permet a l'usuari introduir les dades de cabal i pressió dels emissors de reg i comparar-les amb els valors esperats. A partir d'aquesta comparació, l'algoritme genera una corba de calibratge que s'utilitza per ajustar els emissors i garantir que funcionin de manera eficient i efectiva.



3.3.5. codi dels coeficients emissors

emitter_calibration_execute.py és el següent::

```
import pandas as pd
import matplotlib.pyplot as plt
import os
import numpy as np
import re

from qgis.PyQt.QtCore import pyqtSignal, QObject
from qgis.core import QgsTask

from copy import deepcopy

from ... import global_vars
from ...settings import tools_log, tools_gw

try:
    import wntr
except ImportError:
    wntr = None

class EmitterCalibrationExecute(QgsTask, QObject):
    """ This shows how to subclass QgsTask """

    task_finished = pyqtSignal(list)

    def __init__(self, description, input_file, network,
emtin_data, coeff_dma_result, results, trials, accuracy,
dlg_vol_cal, output_file, result_id, period_id, efficiency,
emitter, timer, duration=0):

        QObject.__init__(self)
        super().__init__(description, QgsTask.CanCancel)
        self.exception = None
        self.duration = duration
        self.timer = timer

        self.input_file = input_file
        self.network = network
        self.emtin_data = emtin_data
        self.method = 'linear'
        self.delete_extra_points = False
        self.coeff_dma_result = coeff_dma_result
        self.results = results
        self.trials = trials
```

```
self.accuracy = accuracy
self.dlg_vol_cal = dlg_vol_cal
self.output_file = output_file
self.result_id = result_id
self.period_id = period_id
self.efficiency = efficiency
self.emitter = emitter

self.current_dma = None
self.current_trial = None

self.result_log = ''

def run(self):

    # Step 2: Run EPANET iterations to find emitter
    coefficient
    self._run_iterations()
    # Step 3: generate .emtout file
    self._generate_emtout(self.coeff_dma_result)
    # Step 4: generate .log file
    # self._ife_calculation()
    # Step 5:
    self._generate_log(self.coeff_dma_result)
    # Step 6:
    self._move_emitters()

    self.task_finished.emit([True, self.result_log])

    return True

def finished(self, result):

    if self.timer:
        self.timer.stop()

    if result:
        tools_log.log_info(f"Task '{self.description()}'
completed")
    else:
        if self.exception is None:
            tools_log.log_info(f"Task
'{self.description()}' not successful but without exception")
        else:
            tools_log.log_info(f"Task
'{self.description()}' Exception: {self.exception}")
```

```

def _run_iterations(self):

    network_junctions = {}
    network_junctions['all_network'] = []
    junctions_by_dma = {}
    network_demand = 0
    effc_by_dma = {'0': f"{self.efficiency}"}
    demand_by_dma = {'0': 0}
    sum_dem_x_c0 = 0
    sum_dem = 0
    junctions_coeff_cal = {}
    network = deepcopy(self.network)

    # Populate all_juntions list and junctions_by_dma
    json variables
    for k, v in self.emtin_data['JUNCTIONS'].items():
        dma_id = v['dma_id']
        if junctions_by_dma.get(dma_id) is None:
            junctions_by_dma[dma_id] = []
        if network_junctions.get(dma_id) is None:
            network_junctions[dma_id] = []
        junctions_by_dma[dma_id].append(k)

network_junctions['all_network'].append(v['node_id'])
    if v['dma_id'] is not None:

network_junctions[dma_id].append(v['node_id'])
        else:
            network_junctions['0'].append(v['node_id'])

    # Populate effc_by_dma and set keys for demand_by_dma
    for k, v in self.emtin_data['DMA'].items():
        dma = v['dma_id']
        effc_by_dma[dma] = v['effc']
        demand_by_dma[dma] = 0

    for junction in network_junctions['all_network']:
        try:
            demand = self.results.node['demand'].loc[:,
junction]
            junction_demand = demand.sum().sum()
            network_demand += junction_demand
        except:
            continue
    for dma_id in effc_by_dma:
        dma_demand = 0

```

```

        if network_junctions.get(dma_id) is None:
            continue
        for junction in network_junctions[dma_id]:
            try:
                demand =
self.results.node['demand'].loc[:, junction]
                junction_demand = demand.sum().sum()
                dma_demand += junction_demand
            except:
                continue
        # Multiply demands for work with wntr units
        demand_by_dma[dma_id] = dma_demand * 1000

        sum_dem_x_c0 +=
float(demand_by_dma[dma_id])*float(effc_by_dma[dma_id])
        sum_dem += float(demand_by_dma[dma_id])
        c0_m = float(sum_dem_x_c0) / float(sum_dem)
        expected_demand = float(sum_dem) / float(c0_m)
        demand_by_dma = dict(sorted(demand_by_dma.items(),
key=lambda item: item[1], reverse=True))

# -----
# RECURSIVITY FOR ALL NETWORK
# -----

self.current_dma = 'All Network'

# 2 ----- add emitters
self.min_coeff_mult = 0
self.mid_coeff_mult = 1
self.max_coeff_mult = 2

df_dict = {'coeff': [],
           'demand': []}

idx = 0
interpolated_idx = -1
for l_coeff in [self.min_coeff_mult,
self.mid_coeff_mult, self.max_coeff_mult]:
    # Set emitter coefficients
    for junction_name, junction in
network.junctions():
        junction_json =
self.emtin_data['EMITTERS'].get(junction_name)
        if junction_json is None or
junction_json['coef'] is None:
            junction.emitter_coefficient =
float(self.emitter) * l_coeff
        else:
            junction.emitter_coefficient =

```

```

float(junction_json['coef']) * l_coeff

    # Run simulation
    self.sim = wntr.sim.EpanetSimulator(network)
    self.results = self.sim.run_sim()

    # Get demand for all network
    first_calculated_demands =
self._get_total_demand(network_junctions['all_network'])
    # Add values to df_dict
    if idx > 0 and df_dict['demand'][idx - 1] <
expected_demand < first_calculated_demands:
        df_dict['coeff'].append(None)
        df_dict['demand'].append(expected_demand)
        interpolated_idx = idx
    df_dict['coeff'].append(l_coeff)
    df_dict['demand'].append(first_calculated_demands)

    idx += 1

# Check that there is data
if len(df_dict['demand']) == 0:
    raise Exception("No demands")
# Check that objective demand is between limits
if expected_demand not in df_dict['demand']:
    raise Exception(f"Demand ({expected_demand}) is
outside the range of coeff allowed")
demands_list_plt = []
# 3 ----- run iterations interpolating
for trial in range(self.trials):

    # Set current trial
    self.current_trial = trial

    # Create dataframe
    df = pd.DataFrame(df_dict)
    # Interpolate NaN values
    interpolated_df =
df.interpolate(method=self.method)

    # Run epanet with coeff = interpolated value
    coeff =
interpolated_df['coeff'][interpolated_idx]
    network = deepcopy(self.network)
    # Set emitter coefficients
    for junction_name, junction in
network.junctions():
        junction_json =
self.emtin_data['EMITTERS'].get(junction_name)

```



```

        if junction_json is None:
            continue
        junction.emitter_coefficient =
float(junction_json['coef']) * coeff
        # Run simulation
        self.sim = wntr.sim.EpanetSimulator(network)
        self.results = self.sim.run_sim()

        # Get demand for all network
        calculated_demands =
self._get_total_demand(network_junctions['all_network'])
        demands_list_plt.append(calculated_demands)
        if abs(calculated_demands - expected_demand) <
self.accuracy:

            for junction_name, junction in
network.junctions():
                junctions_coeff_cal[junction_name] =
junction.emitter_coefficient

            if self.delete_extra_points is False:
                plt.cla()
                plt.plot(demands_list_plt,
label='demand')

                ax = plt.subplot()
                ax.plot(trial, expected_demand, "or",
label='expected_demand')
                plt.legend()
                folder_dir =
os.path.join(global_vars.plugin_dir,
f'resources{os.sep}emitter_calibration{os.sep}png')
                plt.savefig(os.path.join(folder_dir,
f'demand_all_network_plt.png'))

                self.coeff_dma_result['all_network'] = {}
                self.coeff_dma_result['all_network']['coeff']
= coeff

self.coeff_dma_result['all_network']['sum_demands'] =
calculated_demands

        # Fill info log
        self.result_log = f'{self.result_log}\n' \
            f'- All Network: (Trial:{trial})\n' \
            f'- Demand without emitters:
{"{: .8f}".format(sum_dem)}\n' \
            f'- Demand expected:
{"{: .8f}".format(expected_demand)}\n' \
            f'- Demand result:

```

```

{"{: .8f}".format(calculated_demands)}\n' \
        f'- Coef:
{"{: .3f}".format(coeff)}\n\n'
        break

        df_dict['coeff'][interpolated_idx] = coeff
        df_dict['demand'][interpolated_idx] =
calculated_demands

        # Delete farthest point from dataframe
        if self.delete_extra_points:
            if interpolated_idx == 1:

df_dict['coeff'].pop(len(df_dict['coeff']) - 1)

df_dict['demand'].pop(len(df_dict['demand']) - 1)
            elif interpolated_idx == 2:
                df_dict['coeff'].pop(0)
                df_dict['demand'].pop(0)
                interpolated_idx -= 1

        # Now with this dataframe we insert [None, p_f]
and interpolate again
        for i in range(len(df_dict['coeff']) - 1):
            if df_dict['demand'][i] < expected_demand <
df_dict['demand'][i + 1]:
                df_dict['coeff'].insert(i + 1, None)
                df_dict['demand'].insert(i + 1,
expected_demand)

                interpolated_idx = i + 1
                break
            self.trial = trial + 1

        self.network = network

# -----
# RECURSIVITY FOR EACH DMA
# -----

for dma_id in effc_by_dma:

    # Manage demand_by_dma[dma_id] if is None:
    if network_junctions.get(dma_id) is None:
        continue

    # Set current dma
    self.current_dma = dma_id

```

```

# Get junction demands sum (without emitters) -
First epanet
# Multiply demands by effc - get total_demand
if effc_by_dma[dma_id] in (None, ""):
    effc = self.efficiency
else:
    effc = effc_by_dma[dma_id]
expected_demand = float(demand_by_dma[dma_id]) /
float(effc)

# 2 ----- add emitters
self.min_coeff_mult = 0
self.mid_coeff_mult = 1.5
self.max_coeff_mult = 3

df_dict = {'coeff': [],
           'demand': []}

idx = 0
interpolated_idx = -1
for l_coeff in [self.min_coeff_mult,
self.mid_coeff_mult, self.max_coeff_mult]:
    # Set emitter coefficients
    for junction_name, junction in
network.junctions():
        if junction_name not in
network_junctions[dma_id]:
            continue
        junction_coeff =
junctions_coeff_cal[junction_name]
        junction.emitter_coefficient =
float(junction_coeff) * l_coeff

# Run simulation
self.sim = wntr.sim.EpanetSimulator(network)
self.results = self.sim.run_sim()

# Get demand by dma
first_calculated_demands =
self._get_total_demand(network_junctions[dma_id])
# Add values to df_dict
if idx > 0 and df_dict['demand'][idx - 1] <
expected_demand < first_calculated_demands:
    df_dict['coeff'].append(None)
    df_dict['demand'].append(expected_demand)
    interpolated_idx = idx

df_dict['coeff'].append(l_coeff)

df_dict['demand'].append(first_calculated_demands)

```

```

        idx += 1

    # Check that there is data
    if len(df_dict['demand']) == 0:
        raise Exception("No demands")
    # Check that objective demand is between limits
    if expected_demand not in df_dict['demand']:
        raise Exception(f"Demand ({expected_demand})
is outside the range of coeff allowed")
    demands_list_plt = []
    # 3 ----- run iterations interpolating
    for trial in range(self.trials):

        # Set current trial
        self.current_trial = trial

        # Create dataframe
        df = pd.DataFrame(df_dict)
        # Interpolate NaN values
        interpolated_df =
df.interpolate(method=self.method)

        # Run epanet with coeff = interpolated value
        coeff =
interpolated_df['coeff'][interpolated_idx]
        network = deepcopy(self.network)
        # Set emitter coefficients
        for junction_name, junction in
network.junctions():

            if junction_name not in
network_junctions[dma_id]:
                continue

            junction_coeff =
junctions_coeff_cal[junction_name]
            junction.emitter_coefficient =
float(junction_coeff) * coeff

        # Run simulation
        self.sim = wntr.sim.EpanetSimulator(network)
        self.results = self.sim.run_sim()

        # Get demand by dma
        calculated_demands =
self._get_total_demand(network_junctions[dma_id])
        demands_list_plt.append(calculated_demands)
        if abs(calculated_demands - expected_demand)

```

```

< self.accuracy:
    if self.delete_extra_points is False:
        plt.cla()
        plt.plot(demands_list_plt,
label='demand')
        ax = plt.subplot()
        ax.plot(trial, expected_demand, "or",
label='expected demand')
        # plt.axhline(y=expected_demand,
linewidth=3, color='r', linestyle='dotted')
        plt.legend()
        folder_dir =
os.path.join(global_vars.plugin_dir,
f'resources{os.sep}emitter_calibration{os.sep}png')
        plt.savefig(os.path.join(folder_dir,
f'demand_dma_{dma_id}_plt.png'))

        self.coeff_dma_result[dma_id] = {}
        self.coeff_dma_result[dma_id]['coeff'] =
coeff

self.coeff_dma_result[dma_id]['sum_demands'] =
calculated_demands

        # Fill info log
        self.result_log = f'{self.result_log}\n'
\
        f'- DMA: {dma_id}
(Trial:{trial})\n' \
        f'- Demand without emitters:
{"{: .8f}".format(demand_by_dma[dma_id])}\n' \
        f'- Demand expected:
{"{: .8f}".format(expected_demand)}\n' \
        f'- Demand result:
{"{: .8f}".format(calculated_demands)}\n' \
        f'- Coef:
{"{: .3f}".format(coeff)}\n\n'
        break

        df_dict['coeff'][interpolated_idx] = coeff
        df_dict['demand'][interpolated_idx] =
calculated_demands

        # Delete farthest point from dataframe
        if self.delete_extra_points:
            if interpolated_idx == 1:

df_dict['coeff'].pop(len(df_dict['coeff']) - 1)

```



```

df_dict['demand'].pop(len(df_dict['demand']) - 1)
        elif interpolated_idx == 2:
            df_dict['coeff'].pop(0)
            df_dict['demand'].pop(0)
            interpolated_idx -= 1

        # Now with this dataframe we insert [None,
p_f] and interpolate again
        for i in range(len(df_dict['coeff']) - 1):
            if df_dict['demand'][i] < expected_demand
< df_dict['demand'][i + 1]:
                df_dict['coeff'].insert(i + 1, None)
                df_dict['demand'].insert(i + 1,
expected_demand)

                interpolated_idx = i + 1
                break
        self.trial = trial + 1

self.network = network

def _move_emitters(self):

    file_out = f"{self.output_file}.emtout"
    read = False
    new_text = ''
    with open(file_out, 'r') as f:
        lines = f.readlines()
        for line in lines:
            if '[EMITTERS]' in line:
                read = True
            elif read and '[' in line:
                read = False
            if read:
                new_text += f"{line}"

    f.close()
    file_in = f"{self.input_file}"
    read = False
    old_text = ''
    full_text = ''
    with open(file_in, 'r') as f:
        lines = f.readlines()
        for line in lines:
            full_text += line
            if '[EMITTERS]' in line:
                read = True
            elif read and '[' in line:
                read = False

```

```
        if read:
            old_text += f"{line}"
    f.close()
    updated_text = full_text.replace(old_text, new_text)
    inp_out = f"{self.output_file}.inp"
    with open(inp_out, 'w') as f:
        f.write(updated_text)
    f.close()

    def _generate_emptout(self, coeff_dma_result):
        path = f"{self.output_file}.emptout"
        coeff_dma_result = str(coeff_dma_result).replace("'",
'''')
        json_result =
self._call_getepacalfile(result_type='coeff',
result_value=coeff_dma_result)
        self._fill_inp_file(path,
json_result['body']['file'])

    def _generate_log(self, coeff_dma_result):
        path = f"{self.output_file}.log"
        coeff_dma_result = str(coeff_dma_result).replace("'",
'''')
        json_result =
self._call_getepacalfile(result_type='log',
result_value=coeff_dma_result)
        self._fill_inp_file(path,
json_result['body']['file'])

    def _call_getepacalfile(self, result_type=None,
result_value=None):

        extras = f'"type": "volume", "resultId":
"{self.result_id}", "periodId": "{self.period_id}"'
        if None not in (result_type, result_value):
            extras += f', "resultType": "{result_type}",
"resultValue": {result_value}'
        # if result_type == 'log' and self.ife is not None:
        #     extras += f', "resultIfe":{self.ife}'

        self.body = tools_gw.create_body(extras=extras)
        json_result =
tools_gw.execute_procedure('gw_fct_getepacalfile', self.body,
log_sql=True, is_thread=True)
        return json_result
```

```

def _ife_calculation(self):

    # Run simulation
    self.sim = wntr.sim.EpanetSimulator(self.network)
    self.results = self.sim.run_sim()

    # Qpr = Perdidas reales anuales de agua (m3/año)
    Qpr = self._get_leaks()
    # Ndp = Numero de dias en los que el sistema esta
presurizado
    Ndp = 365
    # PRAA = Perdidas reales anuales (m3/año)
    PRAA = Qpr / Ndp

    # P = Presion media de operacion (mca)
    P = self._get_avg_pressure()
    # Lm = Longitud de tuberias (km)
    Lm = self._get_pipes_length()
    # Nc = Numero de acometidas (tuberias a linea de
propiedad)
    Nc = int(self.emtin_data['TITLE'].get('conec_num'))
    # Lt = Longitud total en km de las acometidas, desde
la tuberia al contador
    Lt =
float(self.emtin_data['TITLE'].get('conec_meters')) / 1000
    # PRAI = Perdidas reales anuales inevitables o
umbral minimo de fugas (m3/año)
    PRAI = P * (6.57 * Lm + 0.256 * Nc + 9.13 * Lt)
    self.ife = PRAA / PRAI

def _get_total_demand(self, junctions):

    demands = 0
    for junction in junctions:
        try:
            demand = self.results.node['demand'].loc[:,
junction]
            junction_demand = demand.sum().sum()
            demands += junction_demand
        except:
            continue
    # Multiply demands for work with wntr units
    return demands * 1000

def _get_leaks(self):
    leaks = -1

```



```
sector = None
sectors = None

valid_nodes = 0
nodes = 0

simulation_time =
int(self.network.options.time.duration / 3600)
base_demand =
self.network.query_node_attribute("base_demand")
node_type =
self.network.query_node_attribute("node_type")
pat = self.network.get_pattern("P0")

if sector:
    sim_nodes = set(self.network.node_name_list)
    sector_nodes = set(sectors[sector]["nodes"])
    sim_sector = sim_nodes & sector_nodes
    sector_demands = (
        self.results.node["demand"].iloc[:-1].loc[:,
sim_sector]
    )
else:
    sector_demands =
self.results.node["demand"].iloc[:-1]

# Calculate indicator
for node in sector_demands:

    # Get simulation results
    demand = self.results.node["demand"][node]

    if node not in base_demand:
        continue
    node_base_demand = base_demand[node]
    nodes = nodes + 1
    # Manage valid nodes (only nodes with demand)
    if node_base_demand == 0.0 or node_type[node] !=
"Junction":
        continue
    else:
        valid_nodes = valid_nodes + 1

i = 0
# Calculate indicator
fuites_node = np.zeros((nodes, 1))
demandes_node = np.zeros((nodes, 1))
for node in sector_demands:
```

```

        if node not in base_demand:
            continue
        node_base_demand = base_demand[node]
        # Read the demand of the node (168)
        demanda_node =
np.array(self.results.node["demand"][node])
        # Read the pressure of the node (168)
        pressure_node =
np.array(self.results.node["pressure"][node])

        # demands_json =
self.emtin_data['DEMANDS'].get(node)
        # WIP: get pat_multi
        pat_multi = 1

        # Calculate the provided consume (168) (m3/s?)
        consum_node_S = np.minimum(
            node_base_demand
            # * pat.multipliers[0:simulation_time]
            * pat_multi
            * np.power((np.maximum(pressure_node[:-1], 0)
- 0) / (10 - 0), 0.5),
            # node_base_demand *
pat.multipliers[0:simulation_time],
            node_base_demand * pat_multi,
        )
        # Calculate the node leakage
        fuita_node = demanda_node[:-1] - consum_node_S
        demandes_node[i] = sum(demanda_node)
        fuites_node[i] = sum(fuita_node)

        i = i + 1

    Vl = sum(fuites_node)
    VS = sum(demandes_node)
    indicator_result = Vl / VS

    return Vl[0]

def _get_avg_pressure(self):
    avg_pressure = -1
    pressure = self.results.node['pressure']
    avg_pressure = pressure.mean().mean()
    return avg_pressure

def _get_pipes_length(self):
    total_length = 0
    for pipe_name, pipe in self.network.pipes():
        total_length += pipe.length

```

```
        return total_length / 1000

    def _fill_inp_file(self, folder_path=None,
all_rows=None):

        tools_log.log_info(f"Write inp file.....:
{folder_path}")

        # Generate generic INP file
        file_inp = open(folder_path, "w")
        read = True
        for row in all_rows:
            # Use regexp to check which targets to read
            (everyone except GULLY)
            if bool(re.match('\[(.*?)\]', row['text'])) and \
                ('GULLY' in row['text'] or 'LINK' in
row['text'] or
                'GRATE' in row['text'] or 'LXSECTIONS'
in row['text'])):
                read = False
            elif bool(re.match('\[(.*?)\]', row['text'])):
                read = True
            if 'text' in row and row['text'] is not None and
read:
                line = row['text'].rstrip() + "\n"
                file_inp.write(line)

        self._close_file(file_inp)

    def _close_file(self, file=None):

        if file is None:
            file = self.file_rpt

        try:
            if file:
                file.close()
                del file
        except Exception:
            pass
```

3.3.6. Comportament de l'algoritme

Per entendre tot aquest codi, que no és precisament curt, es pot separar-lo en una estructura general amb 6 funcions principals, que són les següents:

1. `_run_iterations()`, realitza iteracions EPANET per trobar el coeficient emissor.
2. `_generate_emptout()` per generar un fitxer ".emptout" amb els resultats obtinguts en el pas anterior.
3. `_ife_calculation()` per calcular l'índex d'eficiència de la fuga (IFE).
4. `_generate_log()` per generar un fitxer de registre (.log) amb els resultats del càlcul de l'IFE.
5. `_move_emitters()` per actualitzar els coeficients dels emissors.
6. `task_finished` (té l'argument `[True, self.result_log]`) per indicar que la tasca s'ha completat amb èxit i per proporcionar els resultats de la tasca.

Finalment hi ha un `return True`

1. Aquesta funció és la que més ens interessa ja que és el "nucli" principal del programa. S'encarrega d'executar les iteracions de l'algoritme per trobar el coeficient de l'emissor. Inicialment, es realitza una simulació amb uns coeficients d'emissió prèviament definits. A continuació, es compara la demanda calculada amb la demanda esperada i es determina si s'ha assolit la precisió desitjada. Si no és així, es realitza una interpolació per trobar nous valors de coeficient i es repeteix el procés fins a assolir la precisió o el nombre màxim d'iteracions.

A nivell matemàtic, el que fa l'algoritme de `_run_iterations` és utilitzar el mètode d'interpolació per trobar els nous valors de coeficients d'emissió que s'apropen cada vegada més a la solució desitjada. Per aconseguir aquest objectiu, la seqüència del codi realitza el següent:

Inicialització: l'algoritme comença amb una configuració inicial de coeficients d'emissió prèviament definits (valors predeterminats).

Simulació inicial: es realitza una simulació del sistema de subministrament d'aigua utilitzant els coeficients d'emissió actuals. Aquesta simulació proporciona una demanda calculada per cada zona de la xarxa.

comparació amb la demanda esperada: es compara la demanda calculada amb la demanda esperada. Si la diferència entre la demanda calculada i la demanda esperada és inferior a la tolerància que definim al principi, s'ha assolit la precisió desitjada i l'algoritme finalitza.

Interpolació: si la precisió no s'ha assolit, s'aplica un mètode d'interpolació per determinar els nous valors de coeficients d'emissió. Aquesta interpolació es basa en els resultats de la simulació anterior i utilitza tècniques matemàtiques per aproximar-se a la solució òptima.

Simulació amb nous coeficients: es realitza una nova simulació amb els nous coeficients d'emissió obtinguts mitjançant la interpolació. Això proporciona una nova demanda calculada.

Iteració: el procés d'interpolació i simulació es repeteix fins a assolir la precisió desitjada o fins a arribar al nombre màxim d'iteracions establert.

Resultats finals: quan s'assoleix la precisió desitjada o s'acaba el nombre màxim d'iteracions, s'obtenen els coeficients d'emissió finals que s'apropen a la solució optimitzada.

A nivell de programació, l'algoritme `_run_iterations` es realitza mitjançant un bucle que es repeteix fins a assolir la precisió desitjada o el límit d'iteracions. Dins del bucle, es calculen els nous coeficients d'emissió mitjançant la interpolació i es realitza una nova simulació amb aquests coeficients. La diferència entre la demanda calculada i la demanda esperada es compara amb la tolerància per determinar si s'ha assolit la precisió desitjada, tal i com s'ha explicat. El bucle es repeteix fins que es compleixi aquesta condició.

2. Aquesta funció genera un fitxer `.emtout` que conté els resultats de la simulació del sistema de subministrament d'aigua. Utilitza les dades de la simulació emmagatzemades a `self.results` i les guarda en el fitxer especificat a `self.output_file`.

3. Correspon a aquesta funció del codi emitter:

```
def _ife_calculation(self):

    # Run simulation
    self.sim = wntr.sim.EpanetSimulator(self.network)
    self.results = self.sim.run_sim()

    # Qpr = Perdidas reales anuales de agua (m3/año)
    Qpr = self._get_leaks()
    # Ndp = Numero de dias en los que el sistema esta
presurizado
    Ndp = 365
    # PRAA = Perdidas reales anuales (m3/año)
    PRAA = Qpr / Ndp

    # P = Presion media de operacion (mca)
    P = self._get_avg_pressure()
    # Lm = Longitud de tuberias (km)
    Lm = self._get_pipes_length()
    # Nc = Numero de acometidas (tuberias a linea de
propiedad)
    Nc = int(self.emtin_data['TITLE'].get('conneq_num'))
    # Lt = Longitud total en km de las acometidas, desde la
tuberia al contador
    Lt = float(self.emtin_data['TITLE'].get('conneq_meters'))
/ 1000
    # PRAI = Perdidas reales anuales inevitables o umbral
minimo de fugas (m3/año)
    PRAI = P * (6.57 * Lm + 0.256 * Nc + 9.13 * Lt)
    self.ife = PRAA / PRAI
```

Aquesta és la funció per realitzar un càlcul relacionat amb la infraestructura de les xarxes d'aigua. Utilitza la llibreria “wntr” per simular la xarxa d'aigua i calcular certs paràmetres relacionats amb les fugues i la pressió.

La funció calcula la relació entre les pèrdues reals d'aigua i les pèrdues inevitables d'aigua, coneguda com a Índex de Fugues d'Infraestructura (IFI) o Taxa de Fugues d'Infraestructura (TFI). L'IFI/TFI és una mètrica comunament utilitzada per avaluar el rendiment dels sistemes de distribució d'aigua, on un valor més baix indica un millor rendiment.

Específicament, la funció calcula l'IFI/TFI utilitzant la següent fórmula:



$$IFI/TFI = (Q_{pr} / N_{dp}) / (P * (6.57 * L_m + 0.256 * N_c + 9.13 * L_t))$$

on:

- Q_{pr} = Pèrdues anuals reals d'aigua (m³/any)
- N_{dp} = Nombre de dies que el sistema està pressuritzat (generalment 365 dies)
- P = Pressió de funcionament mitjana (mca)
- L_m = Longitud de canonades (km)
- N_c = Nombre de connexions (connexions de servei als clients)
- L_t = Longitud total de les connexions des de la canonada principal fins al comptador (km)

La funció retorna el valor calculat de l'IFI/TFI.

4. Aquesta funció genera un fitxer de registre (.log) amb informació detallada sobre l'execució de l'algorisme de calibració de l'emissor. Utilitza les dades emmagatzemades a `self.result_log` i les guarda en el fitxer especificat.
5. Serveix per moure els emissors a les seves posicions desitjades. Bàsicament rep com a entrada una llista de coordenades que representen les posicions a les quals s'han de moure els emissors.
6. Aquesta funció s'utilitza per a notificar que una tasca ha finalitzat, enviant una llista amb un valor booleà i un registre de resultats. Altres objectes es poden connectar a aquesta senyal i realitzar accions específiques com a resposta, utilitzant els valors de la llista emesa.

Després d'explicar què fa el codi `emitter calibration`, cal mencionar que aquesta només és una de les moltes funcions de `epatools`, que és la extensió de GISWATER. Aquesta és la dimensió del codi del *plugin* en el qual treballem...



Cal afegir també el fet de que instal·lar l'extensió de epatools amb la funció emitter calibration requereix moltes configuracions que no s'han mencionat en el procediment però que cal remarca-les o, si més no, donar una breu explicació.

El primer que s'ha de fer és tenir una versió de QGIS compatible amb la última versió de GISWATER. En cada actualització de GISWATER, les versions de QGIS compatibles són més recents. Després, al afegir el *plugin* de GISWATER, es necessita configurar una base de dades sobre la qual es treballarà. Per fer-ho còmode, BGEO recomana utilitzar pgAdmin, és un programa de base de dades que està basat en SQL. Un cop s'ha configurat això, s'ha d'instal·lar epatools i crear una altra base de dades per utilitzar aquest software. Per poder utilitzar la funció emitter encara no seria possible, abans s'ha de generar un model de xarxa de distribució d'aigua (arxiu .inp) amb la qual treballar. Tal i com es pot apreciar, és un procés no molt intuïtiu, però gràcies a que treballes amb dades que s'emmagatzemaran en el teu servidor, és un sistema molt segur ja que només tu hauries de tenir accés amb la contrasenya personal.

Un aspecte important a mencionar és que la funció emitter calibration no és del tot estable encara. Aquest fet m'ha portat a comunicar-me amb un desenvolupador de BGEO (creadors de GISWATER). Es tracte den Edgar Fusté Culla, tècnic dev-ops amb estudis en l'àmbit de la ciberseguretat i programador multiplataforma. Va començar sent un dels principals programadors de BGEO, fins que aquest rol el va agafar en Sergi Maspons i actualment està de reforç amb ells. Ell està més centrat en bases de dades, com a curiositat. Amb l'ajuda de l'Edgar he pogut, a part de realitzar les configuracions prèvies per tenir la interfície de QGIS preparada, solucionar varis errors que he detectat en la funcionalitat del emitter. Ell mateix em va comentar que és un software que està en continu desenvolupament i que encara s'han d'optimitzar varis punts. Sobretot ha de millorar l'aspecte de que pugui treballar sobre diferents xarxes .inp que no siguin configurades pels propis treballadors de GISWATER.

Aquests problemes d'inestabilitat que han sorgit són també degut a que la funcionalitat del emitter és una versió poc desenvolupada en aquest moment. Va ser un aprofitament del projecte hydroleaks que forma part d'una investigació sobre el emissors de reg en les canonades d'aigua.

3.3.7. Aplicació codi del coeficient K i modificacions

Pel nostre càlcul de la K de decreixement s'aprofita aquest algoritme dels emissors (funció `_run_`iterations), i la diferència en el nostre cas és que els nostres paràmetres, seran: la concentració de clor inicial, la concentració final desitjada i el temps que ha transcorregut. Donant una precisió o tolerància (la que més s'adapti) i el número d'iteracions que volem que realitzi, el programa ha de poder fer el mateix procediment però per calcular el coeficient K.

S'ha realitzat primer una versió "primitiva" del codi per calcular la K. Aquest programa aplica la formula per calcular el coeficient directament. Sense fer iteracions ni ajusts de tolerància, no realitza cap simulació amb EPANET si no que simplement demana el paràmetres d'entrada i et realitza el càlcul.

Aquest és el codi:

```
# Demanem els valors d'entrada necessaris
c1 = float(input("Ingrese la concentración de cloro
en el punto inicial (mg/L): "))
c2 = float(input("Ingrese la concentración de cloro
en el punto final (mg/L): "))
t = float(input("Ingrese el tiempo transcurrido
(horas): "))

# Calculem la tasa de disminució del clor residual
(aproximació)
clor_residual = c1 - c2
tassa_disminucio = clor_residual / t

# Ajustem la tasa de disminució del clor residual
utilitzant un model (equació) de primer ordre
ajust = 0.9 # valor d'ajust per un model de primer
ordre
tassa_disminucio_ajustada = tassa_disminucio * ajust

# Assignem el coeficient de decreixement del clor
coeficient_decreixement = -1 *
tassa_disminucio_ajustada

# Imprimim el resultat
print("El coeficiente de decaimiento del cloro es:
", coeficient_decreixement, "1/hora")
```

I aquests serien els canvis oportuns en la funció principal de les iteracions:

```
import numpy as np
from scipy.optimize import curve_fit

class ChlorineDecay:
    def __init__(self, network, emtin_data,
method='time', trials=10, accuracy=0.001,
delete_extra_points=False):
        self.network = network
        self.emtin_data = emtin_data
        self.method = method
        self.trials = trials
        self.accuracy = accuracy
        self.delete_extra_points =
delete_extra_points
        self.current_trial = None
        self.result_log = ''
        self.coeff_dma_result = {}

    def _get_cl_concentration(self):
        # Funció per obtenir la concentració actual
de clor (per exemple, al final de la canonada).
        # Aquí pots substituir aquesta funció amb el
teu propi codi per obtenir les dades de concentració
de clor.
        return 1.0 # és un exemple, hauries de
substituir-lo per la concentració real.

    def _chlorine_decay(self, t, k):
        c0 = self._get_cl_concentration()
        return c0 * np.exp(-k * t)

    def _run_iterations(self):
        # Pas 1: Generar un dataframe les dades
inicials.
        df_dict = {
            'time': [0.0],
            'conc': [self._get_cl_concentration()]
        }
```

```
}

# Inicialitzar variables
interpolated_idx = 0
demands_list_plt = []

# 3 ----- Executa les iteracions
interpolant.
for trial in range(self.trials):
    # agafa la iteració actual.
    self.current_trial = trial

    # Crea dataframe
    df = pd.DataFrame(df_dict)

    # interpola els valors NaN
    interpolated_df =
df.interpolate(method=self.method)

    # Ajusta la corba de decreixement
    popt, pcov =
curve_fit(self._chlorine_decay,
interpolated_df['time'], interpolated_df['conc'])

    # calcula la concentració actual pel
temps final
t_final = interpolated_df['time'].iloc[-
1]
c_final = self._chlorine_decay(t_final,
popt[0])

    # afegeix els resultats a la dataframe
df_dict['time'].append(t_final + 1) #
afegeix una t més

df_dict['conc'].append(self._get_cl_concentration())

    # comprova si la concentració ha
assolit el valor desitjat
if abs(c_final -
self.target_concentration) < self.accuracy:
    # guarda resultats
```

```
        self.coeff_dma_result['trial_' +
str(trial)] = {}
        self.coeff_dma_result['trial_' +
str(trial)]['k'] = popt[0]
        self.coeff_dma_result['trial_' +
str(trial)]['final_conc'] = c_final

        # info log
        self.result_log += f'Trial {trial}:
K = {popt[0]}, final concentration = {c_final}\n'
        break

        # elimina el punt més llunyà de la
dataframe
        if self.delete_extra_points:
            if interpolated_idx == 1:

df_dict['time'].pop(len(df_dict['time']) - 1)

df_dict['conc'].pop(len(df_dict['conc']) - 1)
            elif interpolated_idx == 2:
                df_dict['time'].pop(0)
                df_dict['conc'].pop(0)
                interpolated_idx -= 1

        # amb aquest dataframe insertem [None,
p_f] i interpolem un altre cop
        for i in range(len(df_dict['time']) -
1):
            if df_dict['conc'][i] <
self.target_concentration < df_dict['conc'][i + 1]:
                df
```



Aquesta és la part a “atacar” per modificar les iteracions que executa el emitter i fer el nostre càlcul de la K. Si es volgués implementar aquesta funció juntament amb les diferents tasques de epatools, es podria assolir aquest objectiu si es realitza més desenvolupament i recursos per poder aplicar la funció de la K de decreixement dins la interfície, però s’ha de treballar conjuntament amb BGEO per assolir aquest objectiu ja que s’han de configurar compatibilitats amb EPANET per a que sigui possible fer les simulacions internes del codi i realitzar el càlcul correctament.



4. CONCLUSIONS

En quant a resultats quantitius, s'ha entès com es pot modelitzar el decreixement del clor en una xarxa de distribució d'aigua. S'ha assolit el que demanava el treball des d'un primer moment, l'estudi i disseny d'un algoritme de control hidràulic per a la millora de la qualitat en una xarxa de distribució d'aigua. Sens dubte, el control d'una substància tant important com és el clor en les canonades d'aigua potable, és clau per millorar la qualitat d'aquesta font tan bàsica per a tota la població. S'ha aconseguit modelitzar la fórmula de primer ordre que defineix el decreixement de la concentració de clor. S'ha realitzat un programa *python* amb l'objectiu de calcular qualsevol K que es demani, és a dir que s'ha aconseguit trobar una manera de modelitzar el decreixement de clor a nivell de software. Tot i les dificultats que s'han trobat per tenir un funcionament estable amb el programari que tenia a la meua disposició, he sabut comprendre com funciona l'algoritme i com aplicar el càlcul desitjat. Cal destacar també que és un treball que es pot actualitzar contínuament i que pots seguir avançant, millorant i modificant-lo cada cop més.

S'han assolit i après un munt d'habilitats i coneixements que em serviran pel futur. El món de les xarxes de distribució d'aigua es molt extens i m'he focalitzat en un tema molt important com és la desinfecció de l'aigua mitjançant el clor. El fet de col·laborar conjuntament amb el desenvolupador de l'extensió sobre la qual he treballat, m'ha ajudat molt per comprendre com funciona la programació per a *plugins* de QGIS, com administrar bases de dades amb pgAdmin, comprendre d'on prové el càlcul de coeficients emissors i com funciona l'algoritme...

És un treball llarg que té molt de camí a recórrer i que pot tenir una millora contínua ja que sempre es pot anar més enllà. El següent pas pot ser el de implementar el codi per afegir una nova opció a epatools amb la possibilitat de fer el càlcul de la K des de la pròpia interfície, per exemple.

El programa creat és una eina amb molt potencial que et permet aplicar una funcionalitat que té EPANET però directament des del *plugin* de GISWATER. Per tant aquest codi arribarà a ser molt interessant per tots els desenvolupadors que vulguin seguir amb aquesta feina i assolir un software capaç de calcular paràmetres tant importants com és la constant de decreixement del clor. És una variable clau per entendre com va canviant el teu sistema en quant a la concentració d'aquest químic i t'ajuda a conèixer en tot moment quina K tindries en un punt en concret fent simulacions. D'aquesta manera, podràs modelitzar el teu sistema com vulguis fins a trobar l'esquema o distribució que més s'adapti a les teves necessitats. Per aquests motius, s'ha aportat també per a la comunitat de desenvolupadors d'aquest software el programa que pot ser implementat com a eina afegida per epatools. He

penjat el codi a la plataforma github i així si algun programador té el temps i els recursos necessaris ho podrà fer. O també pot ser una ajuda pels mateixos desenvolupadors de GISWATER.

La única desavantatge que trobo en aquest software és que, tot i tenir aplicacions molt potents, de codi obert i capaç d'adaptar moltes funcionalitats en la teva interfície de QGIS, el fet d'afegir i/o configurar el *plugin* de GISWATER és un procediment força complex i lent perquè s'han de preparar totes les configuracions prèvies tals com: la connexió amb la base de dades (crear una de nova amb identificador i contrasenya personal), descarregar una versió compatible de QGIS amb Postgres y GISWATER, establir un arxiu .inp compatible amb les funcionalitats de epatools, etc.

Després d'avaluar i reflexionar sobre la feina realitzada i els resultats obtinguts, tenint en compte que en aquest treball he partit des de la base pràcticament, puc concloure que he assolit un munt de coneixements, habilitats i virtuts que no tenia abans de començar-lo. El software de QGIS era tot un camp desconegut per la meva part, i aquest treball m'ha ajudat a comprendre aquest món i a tenir més experiència en aquest àmbit que, sens dubte, em servirà pel futur.

5. BIBLIOGRAFIA

- [1] B. Kowalska (2018). Simulation of chlorine concentration changes in a real water supply network using epanet 2.0 and watergems software packages.pdf. Consultat el 20/01/2023
- [2] L. Monteiro. (2014.). Modeling of Chlorine Decay in Drinking Water Supply Systems Using EPANET MSX-annotated.pdf. Consultat el 20/01/2023
- [3] Autor desconocido. (2012). Application of EPANET for the determination of chlorine dose and prediction of THMs in a water distribution System. Consultat el 29/01/2023
- [4] Autor desconocido. (2015). Numeric Modelling of the Residual Chlorine Concentration Evolution in a Water Distribution Network. Consultat el 20/01/2023
- [5] GISwater. (s.f.). Recuperat de <<https://www.giswater.org/>>. Consultat el 10/08/2022.
- [6] Geoinnova. (2021.). ¿Qué es un SIG, GIS o Sistema de Información Geográfica? Recuperat de <<https://geoinnova.org/blog-territorio/que-es-un-sig-gis-o-sistema-de-informacion-geografica/>> . Consultat el 05/12/2022.
- [7] ESRI España. (s.f.). ¿Qué es un SIG? Recuperat de <<https://www.esri.es/es-es/descubre-los-gis/qu-es-sig/que-es-sig#liSwitcher>> .Consultat el 15/02/2023.
- [8] QGIS. (s.f.). Recuperat de <<https://www.qgis.org/es/site/about/index.html>> Consultat el 20/07/2022.
- [9] United States Environmental Protection Agency. (2021). EPANET User's Manual (versió 2.2.0-1). Recuperat de <https://www.epa.gov/system/files/documents/2021-07/epanet_users_manual_2.2.0-1.pdf> . Consultat el 11/03/2023
- [10] NosoloSIG. (2016). GISwater: solución abierta para la gestión del ciclo de agua. Recuperat de <<https://www.nosolosig.com/articulos/639-giswater-solucion-abierta-para-la-gestion-del-ciclo-agua>> . Consultat el 25/01/2023.
- [11] Agua y desastres. (1991). Recuperat de <[TREBALL FINAL DE GRAU: ILIAS MAAROUF EL BOUAZZI](https://cidta.usal.es/riesgos/CD2/AGUA_DESASTRES/doc14585.pdf#:~:text=En%201846%2C%20el%20doctor%20Ignaz%20Semmelweis%20introdujo%20uno,distribuci%C3%B3n%20de%20agua%20potable%20en%20Maidstone%2C%20Ken%20%28Inglaterra%29.> https://cidta.usal.es/riesgos/CD2/AGUA_DESASTRES/doc14585.pdf#:~:text=En%201846%2C%20el%20doctor%20Ignaz%20Semmelweis%20introdujo%20uno,distribuci%C3%B3n%20de%20agua%20potable%20en%20Maidstone%2C%20Ken%20%28Inglaterra%29.> > Consultat el 10/07/2022.</p>
</div>
<div data-bbox=)

- [12] Aguas Residuales. (2015). La desinfección del agua mediante el uso de cloro. Recuperat de <<https://www.aguasresiduales.info/revista/blog/la-desinfeccion-del-aqua-mediante-el-uso-de-cloro>> .Consultat el 05/03/2023.
- [13] Amazon.es. (s.f.). Fuensantica Hipoclorito Sódico Cloro Líquido. Recuperat de <<https://www.amazon.es/Fuensantica-Hipoclorito-S%C3%B3dico-Cloro-Liquido/dp/B0727R4M1G>> .Consultat el 18/12/2022.
- [14] Amazon.es. (s.f.). Dosificador Clor. Recuperat de <https://www.amazon.es/s?k=dosificador+clor&adgrpid=1304021225768824&hva did=81501415383541&hvbmt=bp&hvdev=c&hvlocphy=3173&hvnetw=o&hvqmt=p &hvtargid=kwd-81501561560496%3Aloc-170&hydadcr=11349_1883626&tag=bingamazonest-21&ref=pd_sl_7sskyyguh4_p> .Consultat el 02/02/2023.
- [15] Universidad Politécnica de Valencia. (s.f.). Recuperat de <<https://ita.upv.es/index-es.php>> . Consultat el 18/01/2023.
- [16] Departament de Salut. (s.f.). Programa de vigilància i control sanitari aigües de consum humà a Catalunya. Recuperat de <https://salutweb.gencat.cat/web/.content/_ambits-actuacio/Per-perfils/Empreses-i-establiments/gestor-subministrament-aigua/programa-vigilancia-control-sanitari-aigues-consum-huma-catalunya/documents/arxiu/3.pdf> .Consultat el 30/03/2023.
- [17] Técnica Industrial. (2019). Recuperat de <<https://www.tecnicaindustrial.es/wp-content/uploads/Numeros/20/42/a42.pdf>> . Consultat el 15/04/2023.
- [18] CIDTA - Universidad de Salamanca. (1995.). Cloración. Recuperat de <<https://cidta.usal.es/cursos/etap/modulos/libros/CLORACION.pdf>> .Consultat el 05/09/2022.