



ENTPE

L'école de l'aménagement durable des territoires



Travail de fin d'études

pour le diplôme d'ingénieur de l'École nationale des travaux publics de l'État

3ème année - Voie d'approfondissement Environnement CELVN
Master SEER

**Développement et validation de méthodes
pour la détection des polluants dans les
effluents urbains : Modélisation avec
Phreeq-C et techniques d'analyse par
fluorescence**

Préparé par : Axelle BARAT

Encadré par : Katharina TONDERA et Benoit HAUTEFEUILLE

Soutenu le 04/09/2024 devant le jury

Katharina TONDERA, Tuteur école

Benoit HAUTEFEUILLE, Tuteur organisme d'accueil

Denise BLANC, Expert

Année universitaire 2023-2024

Notice analytique

Le projet VAI-TRF vise à développer une méthode innovante pour la détection des polluants dans les eaux urbaines en couplant la fluorescence résolue dans le temps à l'intelligence artificielle (IA). Mon stage s'est concentré principalement sur la modélisation à l'aide du logiciel Phreeq-C, qui a permis d'étudier la spéciation du terbium et a confirmé que les matières organiques naturelles n'affectent pas la formation de complexes Tb-polluant. Les polluants se complexent selon l'ordre croissant de leurs constantes de complexation, quelles que soient les concentrations introduites, fournissant des bases solides pour les techniques de détection par fluorescence. En parallèle, l'utilisation de l'IA pour identifier les polluants a montré des résultats opérationnels pour certains polluants, bien que des confusions subsistent encore pour d'autres en raison du manque de données. Ces défis devraient se résoudre avec l'accumulation de données supplémentaires. Le projet a ainsi établi des fondations robustes pour une surveillance précise des polluants dans les eaux urbaines, avec des perspectives prometteuses pour des améliorations futures.

The VAI-TRF project aims to develop an innovative method for detecting pollutants in urban waters by combining time-resolved fluorescence with artificial intelligence (AI). My internship focused primarily on modeling using the Phreeq-C software, which allowed for the study of terbium speciation and confirmed that natural organic matter does not affect the formation of Tb-pollutant complexes. Pollutants complex according to the increasing order of their complexation constants, regardless of the introduced concentrations, providing a solid foundation for fluorescence detection techniques. Meanwhile, the use of AI for pollutant identification has shown operational results for some pollutants, although confusion still persists for others due to a lack of data. These challenges are expected to be resolved with the accumulation of additional data. The project has thus established a robust foundation for precise monitoring of pollutants in urban waters, with promising prospects for future improvements.

Remerciements

Je remercie sincèrement mon tuteur d'organisme d'accueil, Benoît Hautefeuille, pour son soutien, son accueil chaleureux au sein de son entreprise, ainsi que pour son aide précieuse sur l'automatisation de Phreeq-C.

Je suis très reconnaissante envers ma tutrice d'école, Katharina Tondera, pour son suivi constant et ses retours tout au long de cette expérience.

Un grand merci à Laurent Lassabatere pour m'avoir intégrée au projet et pour son aide avec le logiciel Phreeq-C. Ses conseils ont grandement facilité mon travail sur la modélisation.

Je remercie Denise Blanc pour sa présence dans mon jury de soutenance.

Je suis reconnaissante à Mattéo Martini pour son soutien et son attention durant le stage. Sa collaboration a été très appréciée.

Je souhaite particulièrement remercier Yujiao Zhou pour son encadrement rigoureux, son soutien tout au long de mon stage, ainsi que pour la relecture de ce rapport, et pour son amitié.

Un grand merci à Maxime et Vick de l'équipe de l'UNIL pour les données essentielles fournies, qui ont permis de finaliser la modélisation.

Je remercie également les mathématiciens de M2, Hiba et Moustapha, pour leur entraide, leur amitié et leur travail sur la partie IA.

Enfin, je remercie les stagiaires M1, Miche, Reihane et Salamata, pour leur aide et leur enthousiasme, qui ont rendu cette expérience encore plus enrichissante.

Table des matières

Notice analytique	1
Remerciements	2
Tableau des abréviations	6
Table des figures	6
Introduction	9
1 Présentation du projet de recherche : projet VAI-TRF	11
2 Etat de l'art scientifique et technique du projet	13
2.1 Problématiques et contexte	13
2.1.1 Détection des polluants industriels	13
2.1.2 Objectifs et polluants cibles	15
2.2 Détection par la fluorescence résolue dans le temps	16
2.2.1 Principe de la fluorescence	16
2.2.2 Les additifs chimiques	18
2.2.3 Condition réelle : la matrice IHSS	20
2.2.4 Optimisation par modélisation avec Phreeq-C	20
2.3 Les techniques de détection appliquées dans ce projet	20
2.3.1 Spectres d'émission et d'excitation	20
2.3.2 Identification des polluants par l'addition de Tb	21
2.3.3 Compréhension des complexations par Phreeq-C	22
2.3.4 Identification des images à l'aide de IA	22

3	Modélisation des interactions chimiques des différents mésocosmes avec Phreeq-C	24
3.1	Présentation du logiciel	24
3.2	Objectif et place dans le sujet global	26
3.3	Méthodes utilisées et vérifications	26
3.3.1	Choix des différents mésocosmes	26
3.3.2	Choix des paramètres d'entrée et des polluants	27
3.3.3	Obtention des constantes de complexation	28
3.3.4	Premières modélisations numériques	29
3.3.5	Vérifications expérimentales	33
3.3.6	La matrice IHSS comme association de la SH et des ions en solution	36
3.4	Modélisation de la matrice réelle de l'IHSS	38
3.4.1	Premiers essais : réglage du pH	38
3.4.2	Résultats	41
3.5	Automatisation de Phreeq-C à l'aide de Python	42
3.5.1	Introduction	42
3.5.2	Structure du code	43
3.5.3	Les fonctions clés	43
3.5.4	Utilisation du code	46
4	Identification des polluants dans les effluents industriels et municipaux via spectroscopie de fluorescence et IA	48
4.1	Objectif et place dans le sujet global	48
4.2	Enrichir la base de données de l'IA à l'aide de spectres 3D	48
4.2.1	Présentation des spectres 3D	48
4.2.2	Choix des solutions	49
4.2.3	Utilisation du robot Opentron OT-2 et du spectrophotomètre pour les mesures	49
4.3	Apprentissage de l'IA	50
4.3.1	Génération de données	50
4.3.2	Apprentissage de l'IA	51
4.3.3	Premiers résultats : l'IA est-elle fiable pour l'identification ?	52
4.3.4	Piste de résolution des problèmes de non-identification des polluants	54
5	Vérifier la fiabilité du projet sur le terrain	55
5.1	Présentation du dispositif de terrain	55
5.1.1	L'inibox	56
5.1.2	Le robot Opentron OT-2	57
5.1.3	La pompe-box	57
5.2	Etat d'avancement	58
6	Conclusion et perspectives	60

Annexes	65
A Tableau des constantes de stabilité des polluants	66
B Tableau des constantes de stabilité de la NOM	68
C Tableau des concentrations des ions présents dans l'IHSS	69
D Exemple de code pour le robot Opentron OT-2	70
E Fichier input pour la modélisation de la matrice réelle	77
F Code complet de l'automatisation de Phreeq-C	80
G Résultats des 18 méthodes de traitement d'images par l'IA	83

Tableau des abréviations

Abréviation	Description
VAI-TRF	Validation d'une méthode de fluorescence en temps résolu recourant à l'intelligence artificielle pour le suivi de polluants dans les effluents industriels et municipaux
IA	Intelligence artificielle
TRF	Fluorescence en temps résolu
POP	Polluants organiques persistants
UP	Eau ultra pure
IHSS	International humic substances
SH	Substance humique
CNN	Réseau neuronal convolutif
ICP-MS	Spectrométrie de masse par plasma à couplage inductif
NTA	Acide nitrilotriacétique
DTPA	Acide diéthylènetriaminepentaacétique
DTPMP	Acide diéthylènetriaminepentaméthylène phosphonique)
ATMP	Acide amino triméthylène phosphonique
DOTA	Acide 1,4,7,10-tétraazacyclododécane-1,4,7,10-tétraacétique

Table des figures

1.1	Les quatre étapes du projet VAI-TRF	12
1.2	Les partenaires du projet VAI-TRF	12
2.1	Polluants ciblés pour le projet VAI-TRF	16
2.2	Diagramme de Jablonski [1]	17
2.3	Colonne des lanthanides dans le tableau périodique des éléments- <i>Source : Wikipédia, tableau périodique des éléments</i>	19
2.4	Schéma simplifié du mécanisme de chélation	20
2.5	Spectres d'excitation et d'émission [2]	21
2.6	Effet de la chélation du Tb par un polluant sur l'intensité d'émission et d'excitation - Exemple du DTPMP [3]	22
2.7	Fonctionnement d'un CNN adapté au projet VAI-TRF	23
3.1	Entrées nécessaires au fonctionnement du logiciel	25
3.2	Exemple d'un fichier de sortie avec la distribution des espèces terbiées	25
3.3	Place dans le projet VAI-TRF de la modélisation des interactions chi- miques avec Phreeq-C	26
3.4	Choix des mésocosmes dans l'ordre chronologique avec rappel des ob- jectifs	27
3.5	Paramètres d'entrée systématiques pour les modélisations	28
3.6	Tableau des conditions initiales et des résultats pour un seul polluant	30
3.7	Rappel des constantes de complexation des polluants ajoutés	30
3.8	Tableau des conditions initiales de la compétitivité en UP	31
3.9	Résultats de la compétitivité entre les polluants en UP à concentration égale (à gauche) et avec EDTA en excès (à droite)	31
3.10	Structure moléculaire des acides gallique, vanillique et syringique . . .	32
3.11	Graphique des concentrations des différents complexes obtenus en fonction de $[TbCl_3]$	33

3.12	Exemple d'utilisation du robot Opentron OT-2 pour la préparation d'échantillons	34
3.13	Exemple d'utilisation du robot Opentron OT-2 pour la préparation d'échantillons	35
3.14	Comparaison des pH expérimentaux et théoriques en UP	35
3.15	Schéma simplifié représentatif de l'IHSS utilisé dans la modélisation numérique	36
3.16	Paramètres d'entrée systématiques pour la SH	37
3.17	Courbe de pH en fonction de $[TbCl_3]$ attendue	39
3.18	Schéma représentatif de l'IHSS utilisé dans la première modélisation numérique	39
3.19	Courbe de pH obtenue au 1er essai	40
3.20	Courbe de pH obtenue au 2ème essai	41
3.21	Résultats de la modélisation sur la compétitivité des polluants au sein de la matrice réelle	42
3.22	Code de la fonction <code>calculate_concentrations</code>	43
3.23	Code de la fonction <code>extract_input_data</code>	44
3.24	Code de la fonction <code>extract_output_data</code>	44
3.25	Code de la fonction <code>run_phreeqc_simulation</code>	45
3.26	Code de la fonction <code>convert_and_extract_outputs</code>	46
3.27	Exemple d'un fichier d'entrée avec les concentrations de Tb et de Cl variables	46
3.28	Exemple d'un fichier de sortie en format CSV	47
4.1	Place dans le projet VAI-TRF de l'identification des polluants par l'IA	48
4.2	Exemple de spectres 3D, respectivement N5MD5S11 et N10M0T01R06	49
4.3	Spectres obtenus pour un échantillon	49
4.4	Caractéristiques d'un échantillon	50
4.5	Répartition des classes dans la base d'apprentissage	51
4.6	Images des différentes méthodes	52
4.7	Résultats des 3 méthodes retenues	53
4.8	Dernière matrice de confusion obtenue - <i>Travail réalisé par les mathématiciens</i>	54
5.1	Interface de l'application du projet VAI-TRF mise en place par les mathématiciens	55
5.2	Photographie de l'Inibox - <i>Image réalisée par Hiba</i>	56
5.3	Dispositif TRF à l'intérieur de l'Inibox	56
5.4	Photographie du support - <i>Image réalisée par Hiba</i>	57
5.5	Photographie de la pompe-box - <i>Image réalisée par Hiba</i>	58
5.6	Photographie de l'intérieur d'un cabanon	58

Introduction

Les rivières et les lacs sont aujourd'hui menacés par une pollution croissante qui compromet non seulement les écosystèmes aquatiques mais aussi la santé humaine. L'urbanisation, l'industrialisation et l'intensification de l'agriculture ont aggravé la pollution des eaux naturelles par les rejets de produits chimiques synthétiques, provenant aussi bien de sources diffuses en agriculture que de rejets ponctuels issus d'activités domestiques et industrielles. Cette pollution nuit aux écosystèmes aquatiques et comporte des risques pour la santé humaine, le bien-être et l'environnement, avec des répercussions économiques. Il est urgent de surveiller et de réduire ces rejets pour préserver nos ressources en eau. Pour résoudre ce problème, ce projet vise à développer une méthode novatrice basée sur la fluorescence résolue dans le temps pour une surveillance continue des principaux produits chimiques dans les eaux urbaines. Cette méthode repose sur une approche de la technologie de dosage des molécules chélatantes, perfectionnée grâce à l'intelligence artificielle pour permettre une quantification précise des polluants. Dans le cadre de ce projet, nous nous concentrerons sur le développement et la validation des méthodes de détection des polluants (molécules organiques et pesticides) par la fluorescence des additifs (ions lanthanides) dans les eaux urbaines. En analysant la complexation des ions lanthanides par les polluants, notamment des molécules organiques chélatantes, dans un environnement complexe, nous étudierons les interactions entre ces complexes et les éléments présents dans les eaux urbaines pour comprendre leur comportement et leur potentiel d'application en tant que sondes de détection. Parallèlement, nous explorerons les possibilités d'améliorer la sensibilité et la sélectivité de ces méthodes de détection, en tenant compte des défis spécifiques liés à la surveillance des eaux urbaines. L'objectif de ce stage est de construire un modèle sur Phreeq-C, un logiciel de modélisation géochimique qui simule les réactions chimiques dans les solutions aqueuses, pour étudier la complexation entre les polluants classiques et les ions lanthanides. La modélisation numérique consistera à prendre en compte divers paramètres physico-chimiques susceptibles d'influencer la complexation, tels que le pH et la température

de l'eau polluée. Le modèle sera utilisé pour interpréter les jeux de données expérimentales à disposition.

Ainsi, on peut se demander comment développer et valider une méthode innovante de détection des polluants dans les eaux urbaines et industrielles, en utilisant la modélisation chimique avec Phreeq-C et la spectroscopie de fluorescence soutenue par l'IA, afin d'assurer une surveillance continue et précise des contaminants.

Ce rapport sera structuré autour de la présentation du projet VAI-TRF, l'état de l'art, puis les résultats obtenues par la modélisation avec Phreeq-C et sur l'identification des polluants dans les effluents urbains via spectroscopie de fluorescence et IA, et la présentation d'une application prévue sur le terrain pour la vérification de la fiabilité du projet.

Présentation du projet de recherche : projet VAI-TRF

L'objectif du projet est de développer un nouvel instrument basé sur la fluorescence en temps résolu qui sera validé pour surveiller directement les produits chimiques tels que les phosphonates, les sulfonates et les tensioactifs chélateurs.

Le projet est structuré en quatre étapes distinctes comme on peut le voir à la figure 1.1. La première phase se focalise sur la compréhension de la chélation des lanthanides, afin de développer une méthode d'analyse appropriée d'identification et de quantification des polluants par les analyses spectroscopiques. La deuxième étape consistera à élaborer une intelligence artificielle performante et à l'intégrer dans l'Inibox (dispositif présenté à la section 5.1.1), la plateforme de surveillance mobile. La troisième étape impliquera l'utilisation et le paramétrage de l'Inibox, ainsi que la validation de la méthode développée. Enfin, la quatrième étape consistera à déployer l'Inibox dans les emplacements choisis. Mon travail se concentre principalement sur la première tâche de ce projet, avec une contribution partielle à la quatrième.

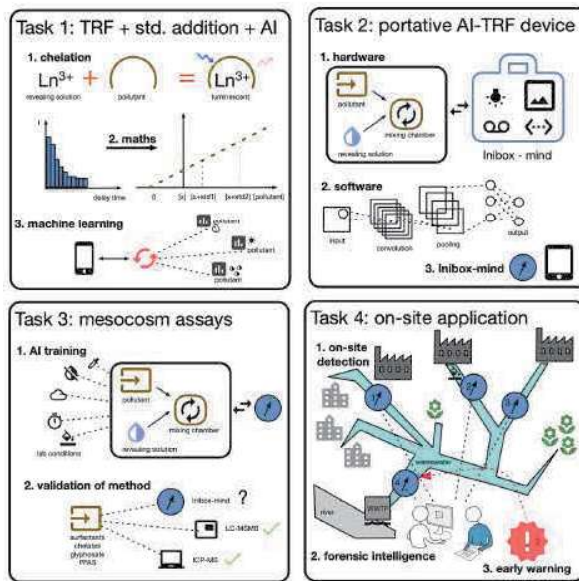


FIGURE 1.1 – Les quatre étapes du projet VAI-TRF

Le projet VAI-TRF rassemble plusieurs partenaires ayant des rôles complémentaires, présentés à la figure 1.2. L'ILM est chargé de la formulation des solutions révélatrices ainsi que de l'échantillonnage et de la conception de prototypes d'appareils de fluorescence à temps résolue. L'École des sciences criminelles de Lausanne (ESC) est responsable de l'analyse des eaux usées et des techniques d'investigation en criminalistique environnementale. Le laboratoire central d'environnement (GR-CEL) développe des outils analytiques pour détecter les traces de polluants, les micropolluants et les nano et micro-plastiques. Le Laboratoire d'Écologie des Hydrosystèmes Naturels et Anthropisés (LEHNA) assure la surveillance de la pollution dans les matrices liquides et solides, et évalue les risques écotoxicologiques. Enfin, la société GLINCS se concentre sur la recherche, le développement et la commercialisation de solutions de traçage et de luminescence personnalisées.



FIGURE 1.2 – Les partenaires du projet VAI-TRF

Etat de l'art scientifique et technique du projet

2.1 Problématiques et contexte

2.1.1 Détection des polluants industriels

Présentation des différents polluants et leur origine

La pollution des cours d'eau est générée par une multitude de substances, rejetées directement dans l'eau. Cette pollution peut être subdivisée en 5 catégories : [4]

- Les nutriments, principalement les nitrates (NO_3^-) et les phosphates (PO_4^{3-}) sont des substances nutritives essentielles à la croissance des végétaux présentes naturellement dans notre environnement. Cependant, les activités humaines contribuent largement à l'augmentation de ces éléments nutritifs dans les eaux urbaines. Les fertilisants agricoles répandus sur les cultures en sont une source majeure, de même que les rejets des eaux usées domestiques et industrielles, ainsi que l'écoulement des eaux dans les zones résidentielles et urbaines. Les nutriments sont acheminés jusqu'aux cours d'eau par le ruissellement. [5]
- Les composés organiques (matières organiques, POP, hydrocarbures) sont une deuxième source majeure de la pollution des eaux urbaines. Les matières organiques se forment à partir de la dégradation de la matière vivante. Elles sont naturellement présentes dans les environnements où les déjections animales, les carcasses d'animaux et les résidus végétaux se décomposent. Elles peuvent également provenir des eaux usées, ainsi que des effluents agricoles et industriels, qui transportent eux aussi de la matière vivante en décomposition. On y retrouve aussi les POPs qui sont des substances chimiques qui se dégradent lentement dans l'environnement, s'accumulent dans les organismes vivants et possèdent des propriétés toxiques et mobiles. Leur origine peut être intentionnelle comme les pesticides ou des produits chimiques industriels à l'exemple des PCB, ou non intentionnelle lors de processus industriels comme les combustions ou la production de substances chimiques [6]. Enfin, les hydrocarbures

réunissent les carburants, les huiles moteurs et les graisses. Ils sont caractérisés par leur insolubilité et peuvent former des nappes à la surface de l'eau. Leur présence dans l'environnement provient principalement des gaz d'échappement, des fuites de moteurs, des opérations d'entretien mal gérées, et des lessivages de zones techniques. [7]

- Les micropolluants sont des substances nocives qui peuvent causer des effets néfastes sur l'environnement et les organismes vivants, même en quantités infimes. Ils proviennent exclusivement des activités humaines et incluent les pesticides, les médicaments, les substances issues d'usages industriels et domestiques, telles que les plastifiants, retardateurs de flamme, isolants, produits de combustion, et divers additifs chimiques. On y retrouve également les POPs, les métaux lourds et les PFAS, considérés comme des micropolluants organiques en raison de leur bioaccumulation dans la chaîne alimentaire. Ces micropolluants proviennent de diverses sources, dont les rejets industriels, les effluents domestiques, l'agriculture, et les eaux pluviales urbaines. [8]
- Les métaux lourds désignent les éléments métalliques ayant une densité supérieure à 5 g/cm^3 et sont naturellement présents dans l'environnement tels que le plomb, le mercure, le cadmium, le cuivre, le chrome et le nickel. Une grande partie de la concentration de ces métaux est accentuée par les activités anthropiques. Les processus industriels, tels que la métallurgie et la production de batteries, libèrent ces éléments dans l'air. L'agriculture contribue par l'utilisation de fertilisants et de pesticides contenant des métaux lourds. Les émissions des véhicules s'ajoutent aussi à cette pollution. Les décharges et les incinérateurs de déchets, ainsi que les activités minières, dispersent ces métaux dans le sol et les cours d'eau. Enfin, la production d'énergie, en particulier à partir de charbon, peut émettre des métaux lourds dans l'air et les eaux par les émissions des centrales électriques. [9]
- Les micro-organismes pathogènes sont des organismes vivants caractérisés par leur parasitisme, capables de se multiplier au sein d'un autre organisme vivant, tels que les champignons et les bactéries.[10] Les rejets d'eaux usées, ainsi que les effluents provenant des élevages et de certaines industries, contiennent souvent des germes microbiologiques d'origine fécale ou des matières organiques en décomposition qui peuvent proliférer. [4]

Importance de la détection précoce et continue

Les deux aspects principaux concernent la santé humaine et la préservation de l'environnement. Une gestion inadéquate des eaux usées urbaines, industrielles et agricoles signifie que l'eau potable de centaines de millions de personnes est dangereusement contaminée ou polluée chimiquement. L'OMS estime à un million le

nombre de personnes qui meurent chaque année en raison d'une eau potable ou d'un assainissement insuffisants. [11] Les polluants peuvent également nuire aux écosystèmes aquatiques en perturbant les habitats et en affectant la faune et la flore comme l'eutrophisation ou les perturbateurs endocriniens. Ils peuvent, dans certains cas, entraîner la mort massive d'espèces. Une surveillance continue aide à minimiser ces impacts en permettant des interventions rapides pour protéger les écosystèmes locaux. [12]

2.1.2 Objectifs et polluants cibles

Justification de l'intérêt du projet

La surveillance en temps réel des contaminants chimiques dans les eaux usées industrielles et municipales est cruciale pour maintenir la qualité de ces effluents. Il est essentiel de détecter rapidement les concentrations dangereuses et de comprendre la dynamique ainsi que les causes potentielles des menaces environnementales pour gérer efficacement les impacts sur l'environnement. Cependant, les méthodes analytiques traditionnelles sont souvent inadéquates pour assurer une surveillance continue. Elles nécessitent des procédures coûteuses et longues, comme le transport des échantillons vers des laboratoires externes, la préparation des échantillons, et l'étalonnage des équipements.

Au niveau réglementaire, la gestion quantitative des ressources en eau se déploie à la fois à l'échelle européenne et nationale, et elle vise à atteindre un équilibre entre la demande en eau et les quantités disponibles. Ce processus est crucial pour maintenir la santé des écosystèmes aquatiques, en veillant à ce que les cours d'eau continuent de fonctionner correctement malgré les pressions humaines et environnementales. La gestion de la ressource en eau en Europe est régie par la Directive Cadre sur l'Eau (DCE) de 2000. Cette directive vise à préserver et restaurer l'état des eaux de surface et souterraines, en ciblant notamment maintien d'un bon état chimique des eaux. En France, la loi sur l'eau de 1992 considère l'eau comme un patrimoine commun et confie aux préfets la régulation des prélèvements pour garantir une gestion équilibrée et durable. Ce sont les Schémas Directeurs d'Aménagement et de Gestion des Eaux (SDAGE) qui définissent des plans d'actions à l'échelle des bassins hydrographiques pour respecter cette gestion. [13]

Pour surmonter ces défis et répondre à ces objectifs, le projet VAI-TRF se concentre sur le développement et la validation d'une nouvelle approche utilisant la fluorescence résolue dans le temps et l'intelligence artificielle (IA). Cette approche innovante vise à permettre une surveillance continue et précise des produits chimiques critiques, tels que les phosphonates, les sulfonates, et les tensioactifs chélatants, dans les installations de traitement des eaux usées, tant industrielles que municipales. En utilisant la fluorescence résolue dans le temps combinée à des algorithmes d'IA, le projet cherche à fournir une méthode plus rapide et plus efficace pour surveiller en temps réel la

présence de ces substances chimiques, répondant ainsi aux exigences modernes de gestion des effluents.

Les polluants identifiés dans le projet

Pour le projet, 6 acides chélateurs étaient initialement identifiés en raison de leur(s) fonction(s) acide(s) capable(s) de se lier par liaison de coordination au niveau de l'oxygène après le relâchement d'un proton avec le Tb(III). De plus, ceux-ci possèdent une constante de stabilité avec le Tb(III) plus élevée que celle de la NOM. Ainsi, l'EDTA, l'ATMP, le NTA, le DTPA, le DOTA et le DTPMP étaient de bons candidats.

Cependant, au cours du stage, une analyse approfondie a révélé que seulement 4 de ces acides chélateurs étaient détectables à une concentration de 5 ppm d'IHSS : le NTA, le DTPA, le DOTA et le DTPMP. L'EDTA et l'ATMP ne se sont pas révélés présents dans ces conditions. Cette observation est probablement due à des différences dans les propriétés de solubilité ou de réactivité des acides chélateurs, dont les logK sont inférieurs aux 4 autres polluants retenus.

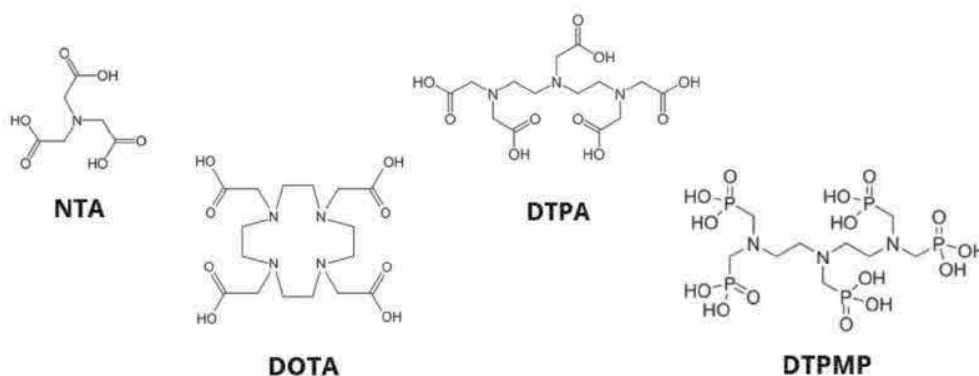


FIGURE 2.1 – Polluants ciblés pour le projet VAI-TRF

Ces acides chélateurs sont principalement classés dans les familles de polluants organiques. Ils peuvent être considérés comme une menace pour l'environnement en raison de leur capacité à se lier aux métaux lourds et de leur persistance dans les milieux aquatiques.

2.2 Détection par la fluorescence résolue dans le temps

2.2.1 Principe de la fluorescence

Une molécule fluorescente, aussi appelée un fluophore, a la capacité d'absorber de l'énergie provenant d'un rayonnement lumineux incident, ce processus s'appelle l'excitation. Après avoir absorbé cette énergie, la molécule la libère rapidement en

l'espace de quelques nanosecondes, sous forme de lumière émise. [1] On différencie la fluorescence de la phosphorescence, où l'émission de lumière persiste plus longtemps après l'arrêt de l'excitation de l'ordre d'au moins quelques microsecondes. [14] Ce principe est représenté plus simplement dans le diagramme de Jablonski :

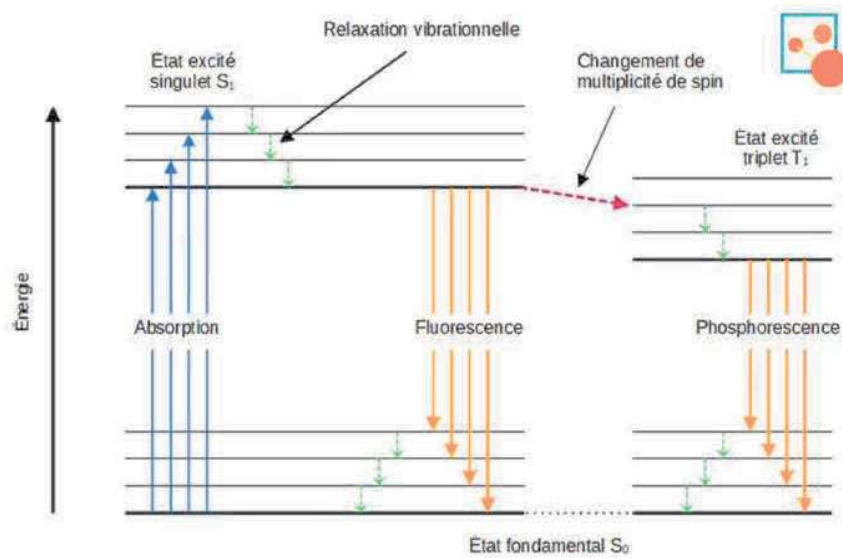


FIGURE 2.2 – Diagramme de Jablonski [1]

La fluorescence est un processus au cours duquel des molécules absorbent de l'énergie, ce qui les propulse vers un état d'énergie plus élevé, l'état d'excitation. Ensuite, ces molécules retrouvent rapidement leur état d'origine en émettant de la lumière. Ce retour à l'état fondamental se produit en très peu de temps. Cette caractéristique de temps extrêmement court rend la fluorescence particulièrement utile pour les applications nécessitant une détection rapide. [15] La phosphorescence est un phénomène lumineux plus lent où la molécule se désexcite via une conversion inter-système (CIS) vers un état triplet (T_1), suivi de l'émission d'un photon lors du retour à l'état fondamental (S_0), prolongeant la durée lumineuse de 10^{-6} à 10 secondes en raison de transitions de spins défavorisées. [16]

Le principe de fluorescence en temps-résolu, utilisé dans le cadre du projet, repose sur le temps de vie, noté τ , qui correspond à la durée de vie de l'état excité. C'est une mesure du temps que met une molécule excitée pour revenir à son état fondamental après avoir absorbé de l'énergie et émis un photon. Plus précisément, il s'agit du temps nécessaire pour que la population d'atomes ou de molécules dans l'état excité diminue de 63% par rapport à sa valeur initiale. [17]

Enfin, le temps de vie est fortement influencée par l'environnement autour de la molécule ou de l'atome émetteur. Dans le cadre de ce projet, les émetteurs sont mis dans l'eau, les molécules de l'eau présentent un effet de réduction de l'intensité d'émission, appelé généralement 'quenching effect'. Un quencher est une substance qui diminue

ou éteint la fluorescence ou la phosphorescence d'une molécule. Le quenching est couramment utilisé pour étudier les interactions moléculaires et les mécanismes photophysiques dans divers domaines scientifiques.[18] La présence de molécules chélatantes autour des émetteurs permet de réduire cet effet de quenching, avec une efficacité variable selon la nature de la molécule chélatante. Ce phénomène offre une méthode d'analyse spectroscopique des émetteurs, qui permet de caractériser différents polluants en fonction de leur impact sur la réduction de la fluorescence ou de la phosphorescence. [19]

2.2.2 Les additifs chimiques

Les émetteurs additifs sont choisis parmi les composés luminescents pour leurs propriétés qui les rendent particulièrement adaptés au projet. Suite aux recherches précédentes menées par l'équipe, les lanthanides se sont révélés être des candidats idéaux en raison de plusieurs avantages remarquables. En effet, ces éléments possèdent un pic d'émission fin et intense, facilitant ainsi leur analyse précise. De plus, ils sont extrêmement stables dans le temps, ce qui garantit une performance fiable et durable. Enfin, leur sensibilité à l'environnement permet d'obtenir des réponses précises aux variations de conditions, tout en restant relativement abordables et non toxiques.

Présentation des lanthanides

Les lanthanides peuvent émettre de la lumière lorsqu'ils sont excités par des photons, ce qui les rend précieux pour les applications en fluorescence et en phosphorescence, c'est pourquoi on s'y intéresse ici. [20] Ils comprennent quinze éléments chimiques dont les numéros atomiques vont de 57 à 71 dans la table de Mendeleïev, allant du Lanthane (La) au Lutécium (Lu).

Tableau périodique des éléments chimiques

Le tableau périodique des éléments chimiques est présenté avec la colonne des lanthanides (éléments 57 à 71) mise en évidence par une bordure rouge. Les éléments sont classés par périodes (1 à 7) et groupes (1 à 18). Les lanthanides sont insérés dans le tableau entre les éléments 56 (Ba) et 72 (Hf) de la 6^{ème} période. Les actinides (éléments 89 à 103) sont placés sous les lanthanides. Le tableau inclut également des informations sur le numéro atomique, le symbole chimique et le nom de l'élément.

FIGURE 2.3 – Colonne des lanthanides dans le tableau périodique des éléments-
Source : Wikipédia, tableau périodique des éléments

Les lanthanides, accompagnés par les actinides (colonne placée sous les lanthanides dans le tableau de Mendeleïev) composent le bloc f. En effet, ces éléments possèdent une configuration électronique particulière, les orbitales 4f commencent à se remplir après l'élément La, avec un nombre croissant d'électrons 4f à mesure que l'on avance dans la série des lanthanides. Cependant, ces orbitales 4f ne sont jamais complètement remplies pour ces éléments (jusqu'au Lu, qui fait donc partie du bloc d). [21]

Les lanthanides, en raison de leurs orbitales 4f profondément enfouies et peu réactives, ont des liaisons covalentes rares avec un ligand. Au lieu de cela, ils privilégient des interactions électrostatiques, basées sur l'attraction entre leur charge positive et les ligands. En raison de leur caractère oxophile, les lanthanides sont plus attirés par les ligands oxygénés et forment souvent des complexes avec des nombres de coordination élevés, généralement entre 8 et 9. Ces caractéristiques sont aussi influencées par les exigences stériques, qui concernent l'espace disponible autour du métal central pour accueillir ces ligands. [16]

Dans le cadre de ce projet, on se focalise sur le Tb(III), qui a une durée de vie de luminescence de plusieurs millisecondes.

Effet chélate des complexes de lanthanides

La chélation est un mécanisme physico-chimique au cours duquel est formé un complexe, le chélate, entre un ligand, dit chélateur, et un cation (ou atome) métallique, alors complexé, dit chélaté. [22]

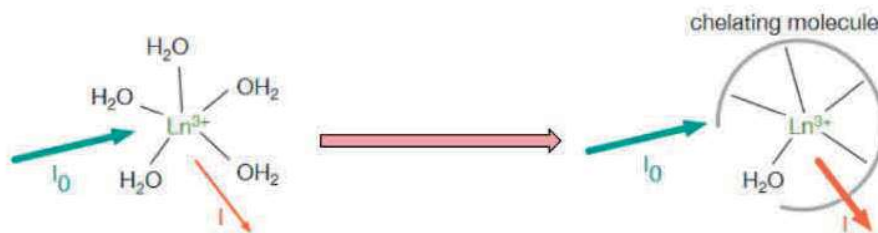


FIGURE 2.4 – Schéma simplifié du mécanisme de chélation

2.2.3 Condition réelle : la matrice IHSS

La substance humique utilisé dans notre projet est la Suwannee river, référencée par le code 2R101N, sur le site de l'IHSS. [23] Cette matrice est considérée comme de la matière organique naturelle aquatique. Les deux groupes fonctionnels qui la caractérise sont les acides carboxyliques et les acides phénoliques. [24] Les concentrations des différents ions qui la composent sont disponibles en Annexe C.

2.2.4 Optimisation par modélisation avec Phreeq-C

La modélisation avec Phreeq-C est essentielle pour comprendre les interactions complexes entre un ion et les acides humiques que sont les carboxyles et les phénols. Elle fournit des informations détaillés sur la formation des complexes et la distribution de l'ion sur les groupements fonctionnels des SH qui sont impossibles à explorer expérimentalement. Cela permet une évaluation précise de l'impact de la NOM sur la spéciation du Tb (dans notre cas) dans les environnements naturels. [25] Dans tous les cas étudiés, les groupes fonctionnels de l'acide humique sont considérés comme des sites actifs de surface qui interagissent avec les ions et les molécules en solution par absorption et complexation plutôt que de simplement se dissoudre dans la solution. [26] [27]

2.3 Les techniques de détection appliquées dans ce projet

2.3.1 Spectres d'émission et d'excitation

Les techniques spectroscopiques sont cruciales pour l'identification des polluants dans divers échantillons. Ces méthodes reposent sur l'analyse des propriétés optiques des substances. La spectroscopie d'excitation mesure la lumière absorbée par un échantillon lorsqu'il est exposé à une source lumineuse à différentes longueurs d'onde. En faisant varier la longueur d'onde de la lumière incidente, on observe quelles longueurs d'onde sont absorbées par l'échantillon. Ces longueurs d'onde spécifiques cor-

respondent aux niveaux d'énergie des électrons dans les molécules du polluant. Ces spectres fournissent des informations sur les polluants en comparant les spectres obtenus à des spectres de référence. La spectroscopie d'émission mesure la lumière émise par un échantillon après excitation par cette même source. Lorsqu'un échantillon est excité, les électrons des molécules passent à un niveau d'énergie plus élevé (partie 2.2.1). En revenant à leur état fondamental, ces électrons émettent de la lumière. Le spectre d'émission obtenu révèle des pics à des longueurs d'onde spécifiques correspondant aux transitions électroniques des polluants. [2] [28]

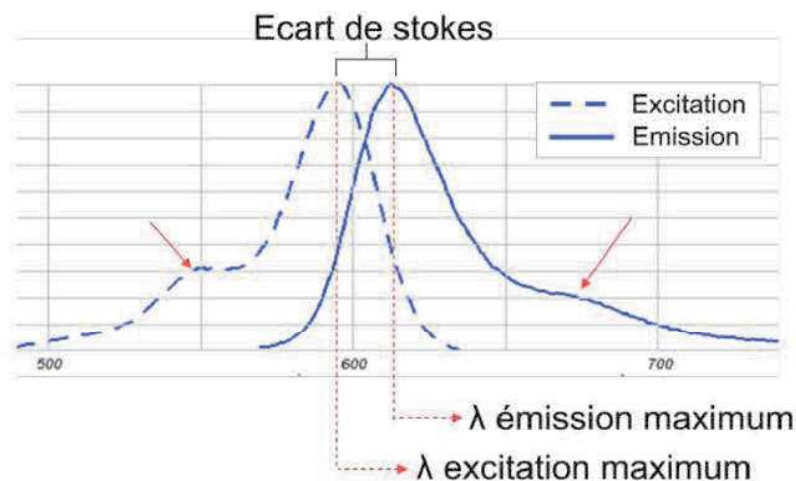


FIGURE 2.5 – Spectres d'excitation et d'émission [2]

2.3.2 Identification des polluants par l'addition de Tb

L'identification de polluants par addition de terbium est une technique puissante dans l'analyse chimique grâce à ses propriétés de luminescence et de sa capacité à former des complexes avec d'autres molécules, dont les polluants. En effet, le terbium peut se lier à des ligands spécifiques, formant des complexes stables. Ces complexes peuvent avoir des caractéristiques de fluorescence particulières qui varient en fonction des propriétés chimiques du ligand avec lequel le Tb se lie. L'échantillon à analyser est préparé en solution contenant les polluants auquel on ajoute une certaine concentration en Tb. La solution est ensuite excitée par une source lumineuse et la fluorescence émise par les complexes de Tb est mesurée à l'aide d'un spectrofluorimètre. Cette méthode présente de nombreux avantages tels que la sensibilité (détection de polluant même à faible concentration) et la rapidité. [29]

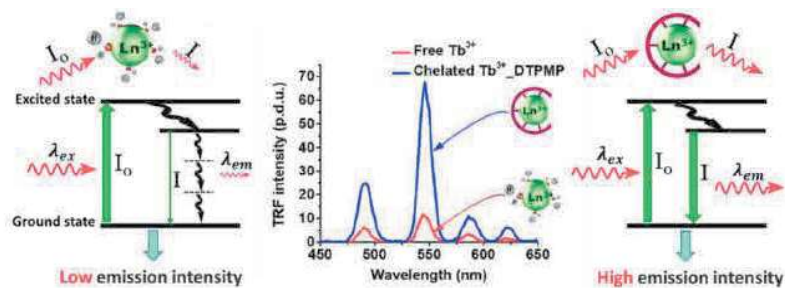


FIGURE 2.6 – Effet de la chélation du Tb par un polluant sur l’intensité d’émission et d’excitation - Exemple du DTPMP [3]

2.3.3 Compréhension des complexations par Phreeq-C

Phreeq-C est un atout essentiel pour optimiser la méthode de détection du projet VAI-TRF. Ce logiciel a déjà fait ses preuves dans de nombreuses recherches. Par exemple, l’article intitulé *hemical Speciation of Some Metal Ions in Groundwaters of Yola Area Using Geochemical Model*[30] explore l’application de PhreeqC pour modéliser la spéciation chimique de divers ions métalliques dans les eaux souterraines de la région de Yola. L’étude met en avant l’utilisation de Phreeq-C pour analyser et comprendre la répartition et la forme des ions métalliques dans les eaux souterraines. Cela a alors permis aux chercheurs de comprendre les interactions entre les ions métalliques et les ligands présents dans les eaux, ainsi que l’impact des conditions environnementales. Phreeq-C est un outil majeur qui aide à interpréter les données expérimentales et à prédire les effets des variations des conditions environnementales sur les systèmes aquatiques.

2.3.4 Identification des images à l’aide de IA

Certains sujets de recherche ont déjà associé la spectroscopie par photoluminescence et le machine learning pour l’identification d’un élément. L’IA joue un rôle crucial dans l’interprétation des données complexes issues des spectres 3D, en extrayant des motifs spécifiques et en les classant efficacement. Beaucoup d’études ont prouvé son efficacité, notamment dans l’identification des plastiques ou de polluants du sol.[31] [32] [33] Cependant, aucun ne traite l’identification des polluants dans les eaux urbaines, ce qui est un argument de taille pour le projet VAI-TRF.

Les étapes de cette démarche reposent sur la collecte de spectres de photoluminescence pour divers éléments à identifier, le prétraitement des données pour réduire le bruit de ces spectres, ainsi que l’entraînement de modèles d’IA pour les identifier. Il existe différentes techniques de machine learning, telles que les random forests ou les réseaux de neurones (CNN). [31]

Dans le cadre du projet, c’est la méthode par CNN qui a été retenue.

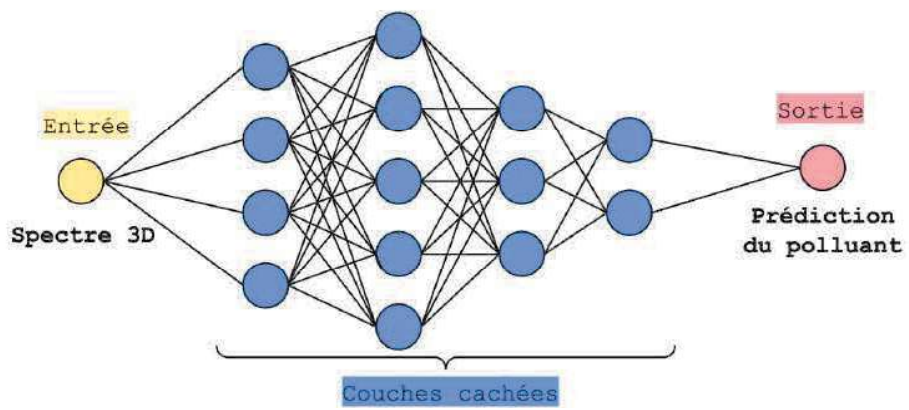


FIGURE 2.7 – Fonctionnement d'un CNN adapté au projet VAI-TRF

Modélisation des interactions chimiques des différents mésocosmes avec Phreeq-C

3.1 Présentation du logiciel

Phreeq-C est un programme informatique largement utilisé pour la modélisation géochimique des processus en solution aqueuse. Il permet de simuler des réactions chimiques dans des systèmes aqueux et d'étudier une variété de processus géochimiques. C'est un atout crucial pour comprendre la spéciation d'un élément et voir les interférences qui peuvent exister en présence de différents polluants.[34]

Dans le cadre du projet, le logiciel Phreeq-c nous permet de reproduire notre environnement c'est-à-dire l'eau de la Suwanne River afin de faire les analyses des processus chimiques intrinsèques et de vérifier les complexations Tb-polluant malgré l'environnement.

Il prend en compte des réactions chimiques complexes telles que des réactions de précipitation et de dissolution des ions, les échanges ioniques, les réactions de surfaces, les réactions entre l'eau et l'atmosphère et enfin les réactions d'oxydoréduction et de complexation. Dans notre cas, nous portons notre attention sur les réactions de complexation entre le terbium et les polluants étudiés. Cela nous permet de comprendre la spéciation d'un élément et de voir les interférences qu'il peut y avoir en présence de différents polluants.

```

Name of input file?
/Users/axellebarat/Desktop/Phreeqc/bin/EPcompet.txt
Input file: /Users/axellebarat/Desktop/Phreeqc/bin/EPcompet.txt

Name of output file?
Default: /Users/axellebarat/Desktop/Phreeqc/bin/EPcompet.txt.out
/Users/axellebarat/Desktop/Phreeqc/output/EPcompetout.txt
Output file: /Users/axellebarat/Desktop/Phreeqc/output/EPcompetout.txt

Name of database file?
Default: phreeqc.dat
/Users/axellebarat/Desktop/Phreeqc/database/llnl.dat
Database file: /Users/axellebarat/Desktop/Phreeqc/database/llnl.dat

```

FIGURE 3.1 – Entrées nécessaires au fonctionnement du logiciel

Le choix de la base de données llnl.dat a été fait ainsi car cette base offre la plus grande quantité d'informations pertinentes, notamment en ce qui concerne les lanthanides, qui sont au cœur du projet. Phreeq-C renvoie alors un fichier de sortie sous format texte avec des valeurs d'intérêt telles que le pH et la distribution des espèces présentes en solution.

-----Distribution of species-----							
Species	Molality	Activity	Log Molality	Log Activity	Log Gamma	mole V cm ³ /mol	
Tb(3)	5.000e-04						
TbNta+2	3.881e-04	2.510e-04	-3.411	-3.600	-0.189	(0)	
Tb(Nta)2+	1.009e-04	9.052e-05	-3.996	-4.043	-0.047	(0)	
Tb+3	9.309e-06	3.569e-06	-5.031	-5.447	-0.416	(0)	
TbIhss+2	1.672e-06	1.081e-06	-5.777	-5.966	-0.189	(0)	
TbCl+2	1.337e-08	8.679e-09	-7.874	-8.062	-0.188	(0)	
Tb(Ihss)2+	1.422e-11	1.275e-11	-10.847	-10.894	-0.047	(0)	
TbOH+2	8.952e-12	5.812e-12	-11.048	-11.236	-0.188	(0)	
TbCl2+	7.090e-12	6.359e-12	-11.149	-11.197	-0.047	(0)	
TbCl3	3.336e-15	3.336e-15	-14.477	-14.477	0.000	(0)	
Tb0+	3.123e-18	2.801e-18	-17.505	-17.553	-0.047	(0)	
TbCl4-	1.828e-18	1.640e-18	-17.738	-17.785	-0.047	(0)	
Tb02H	4.102e-25	4.102e-25	-24.387	-24.387	0.000	(0)	
Tb02-	3.457e-32	3.100e-32	-31.461	-31.509	-0.047	(0)	

FIGURE 3.2 – Exemple d'un fichier de sortie avec la distribution des espèces terbiées

Les espèces sont rangées dans le sens décroissant selon leur concentration, on espère donc obtenir les complexes Tb-polluant en tête de liste. Phreeq-C est en premier lieu un logiciel connu pour son interface plutôt intuitive, qui permet de configurer de manière simple des modélisations diverses. Cette interface graphique est conçue pour simplifier l'entrée des données et la visualisation des résultats. Cependant, il peut arriver que ce logiciel ait des problèmes de compatibilité ou d'installation sur certains ordinateurs, ce qui a été mon cas, d'où le choix d'utiliser directement la console.

3.2 Objectif et place dans le sujet global

L'objectif de la modélisation des interactions chimiques dans l'environnement naturel est de comprendre la spéciation du Tb(III) et voir les interférences qui peuvent exister en présence de différents polluants. Celui-ci répond à la tâche 1 du projet VAI-TRF, et plus particulièrement sur la partie compréhension de la chélation des lanthanides avant de développer une nouvelle méthode d'identification des polluants complexe.

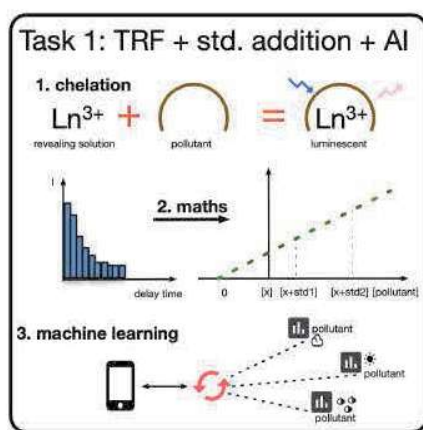


FIGURE 3.3 – Place dans le projet VAI-TRF de la modélisation des interactions chimiques avec Phreeq-C

3.3 Méthodes utilisées et vérifications

3.3.1 Choix des différents mésocosmes

Les différents polluants étudiés pour le projet ne sont pas répertoriés dans la base de données utilisée. Dans un premier temps, l'acétate et l'UP ont été utilisés. L'acétate étant déjà présent dans la base de données, cela a permis de s'assurer de la bonne prise en main du logiciel en considérant l'acétate comme un de nos acides. Ensuite, l'UP a été combiné avec chaque polluant pour vérifier les premières hypothèses, en particulier la compétitivité entre les polluants, liée à leurs constantes respectives. Il est important de noter qu'au moment de ces premiers essais Phreeq-C, l'EDTA et le DTPA faisaient encore partie de nos candidats (partie 2.1.2), d'où leur utilisation en UP. Enfin, la matrice réelle étant une matrice complexe à modéliser, une matrice simplifiée a d'abord été modélisée en s'appuyant sur un mélange de 3 acides aux propriétés similaires à celles de l'IHSS.

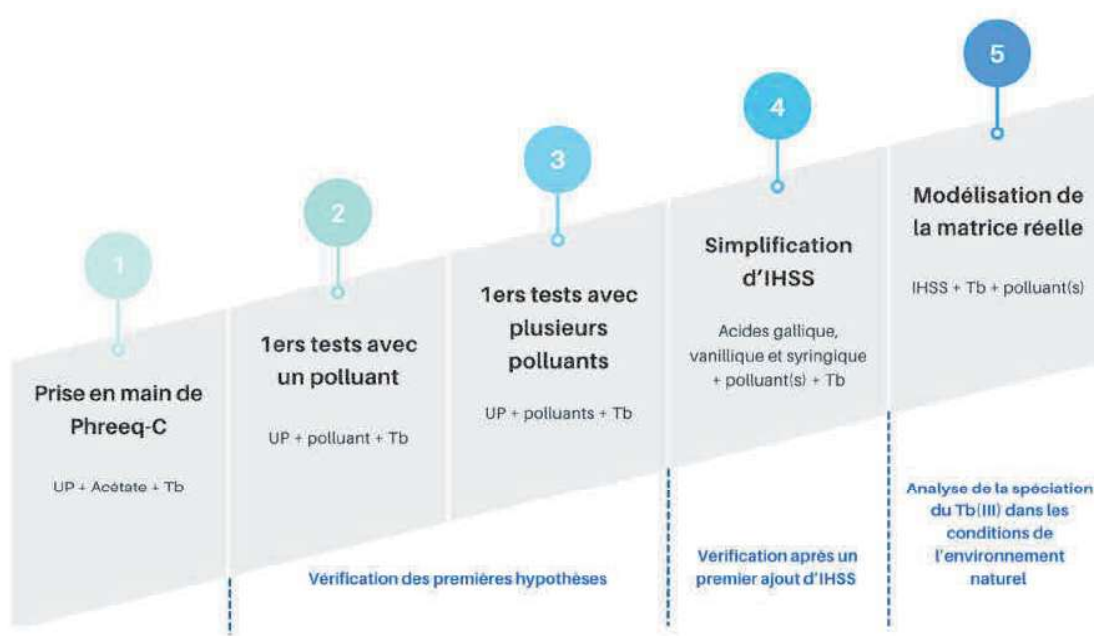


FIGURE 3.4 – Choix des mésocosmes dans l'ordre chronologique avec rappel des objectifs

A partir de l'étape 2, les données sur les polluants, manquantes à la base de données, ont été rajoutées à la main dans chaque fichier input, avec la prise en compte des équations de complexation acide-Tb et des logK associées, point-clé de cette partie. Ce choix a été établi pour éviter tout problème au sein de la base de données.

3.3.2 Choix des paramètres d'entrée et des polluants

Plusieurs paramètres ont été fixés dans toutes les modélisations sur Phreeq-C :

- La température : fixée à 25, ce qui correspond à une moyenne annuelle de l'eau du Mississippi,
- Le pH : L'utilisation du mot-clé "charge" indique au logiciel qu'il faut ajuster automatiquement le pH de la solution en fonction des espèces présentes. Sa formule est la suivante : $\text{pH} = -\log_{10}[H^+]$,
- Le potentiel électrochimique : conventionnellement fixé à 4. Il est régi par la formule : $\text{pE} = -\log_{10}[\text{lectrons}]$,
- Le potentiel rédox : La condition "pe" spécifie que le logiciel doit utiliser le pE pour déterminer les états d'oxydation des espèces redox présentes dans la solution.
- La densité : qui a toujours une valeur de 1kg/L pour l'eau,

— La quantité d'eau : fixée à 1L pour toutes les modélisations.

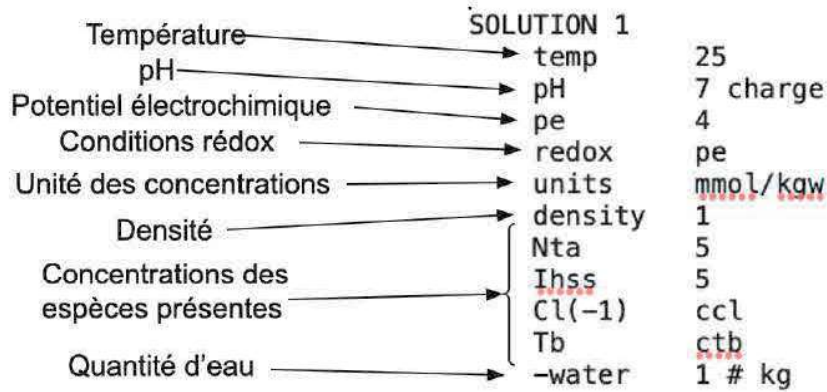
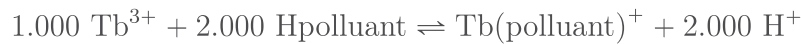


FIGURE 3.5 – Paramètres d'entrée systématiques pour les modélisations

On ajoute à ces paramètres fixes les concentrations nécessaires aux différentes modélisations selon les espèces souhaitées (polluants, Tb, ions). A cela s'additionne les équations de complexation TB-polluant comme suit :



3.3.3 Obtention des constantes de complexation

Les constantes de stabilité, notées $\log K$ ou $\log \beta$, quantifient la stabilité des complexes formés entre un ion métallique et des ligands. Dans le cas général, l'équation de la réaction de formation du complexe est :



L'équation de la constante de stabilité pour ce complexe est alors :

$$\beta_n = \frac{[ML_n]}{[M][L]^n}$$

où :

- β_n : la constante de stabilité globale du complexe.
- $[ML_n]$: concentration du complexe formé.
- $[M]$: concentration du métal libre.
- $[L]$: concentration du ligand libre.

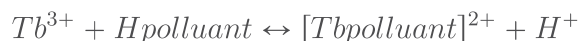
— n : nombre de ligands liés au métal.

Ces constantes sont indispensables pour les modélisations Phreeq-C suivantes. La méthode de détermination de ces constantes utilisée dans le cadre du projet est celle émise par les membres de l'équipe basée à Lausanne. Leur méthode se base sur la méthode d'Irving et Rossotti. [35] Celle-ci est une technique classique basée sur la titration potentiométrique pour déterminer les constantes de stabilité. Cette méthode mesure le potentiel d'une solution contenant le complexe et ses composants en fonction de la concentration d'un des réactifs, généralement les ions H^+ . La courbe de titration obtenue, qui trace le potentiel mesuré contre le volume de titrant ajouté, permet de déterminer la constante de stabilité du complexe. [36] [37] [35] L'UNIL a alors déterminé les constantes de stabilité pour les polluants et la NOM (associés au Tb). Ces résultats sont disponibles aux Annexe A et Annexe B.

3.3.4 Premières modélisations numériques

Modélisation en eau ultra pure avec un polluant

En se basant sur les résultats de l'Université de Lausanne, on fait l'hypothèse que les polluants utilisés sont des mono-acides (voir Annexe A). La complexation entre le polluant et le Tb est alors régie par l'équation suivante :



On introduit dans la solution le polluant et le $TbCl_3$. Dans un premier temps, la concentration en polluant est fixée à 1 mmol/L et la concentration en Tb varie de 0,1 mmol/L à 1 mol/L. Dans un second temps, c'est la concentration en Tb qui est fixée à 20 mmol/L et la concentration en polluant varie de 1 mmol/L à 1 mol/L. Pour chacun des polluants, on relève la concentration du complexe formé Tb-polluant²⁺.

pH initial	Température (en °C)	[ATMP] (en mmol/L)	[Tb ³⁺] (en mmol/L)	[Cl ⁻] (en mmol/L)	pH final	% erreur	[TbATMP ⁺⁺] (en mmol/L)
7	25	1	0,1	0,3	3,017	0	0,1
7	25	1	0,5	1,5	3,022	0	0,5
7	25	1	1	3	3,027	0	1
7	25	1	5	15	3,057	0	1
7	25	1	10	30	3,071	0	1
7	25	1	90	270	3,109	0	1
7	25	1	1000	3000	3,112	0	1
7	25	1	20	60	3,086	0	1
7	25	5	20	60	2,386	0	5
7	25	10	20	60	2,083	0	10
7	25	20	20	60	1,778	0	20
7	25	30	20	60	1,604	0	20
7	25	100	20	60	1,09	0	20
7	25	1000	20	60	0,089	0	20

FIGURE 3.6 – Tableau des conditions initiales et des résultats pour un seul polluant

En présence d'un unique polluant dans la solution, quelque soit le polluant, les résultats sont identiques. Il est important de noter que dans tous les cas, la concentration du complexe Tb-polluant est toujours égale à la plus faible concentration initialement ajoutée dans la solution entre le polluant et le Tb, indiquant que le facteur limitant est cette concentration. Néanmoins, le complexe est toujours entièrement formé ; ce qui est logique. Un autre point est que le pH augmente légèrement quand la concentration en Tb^{3+} augmente. Cela s'explique par l'hydrolyse du Tb^{3+} . En effet, la formation de $Tb(OH)_3$ implique la consommation de H^+ pour produire OH^- , ce qui basifie légèrement le pH. De l'autre côté, quand on augmente la concentration en polluant (comme l'ATMP à la figure ci-dessus) le pH diminue de 3 points, ce qui s'explique simplement par l'ajout d'un acide dans la solution jusqu'à une concentration 1000 fois plus importante qu'au départ.

Compétitivité des polluants en eau ultra pure

Une hypothèse émise par l'UNIL est que si le $\log\beta$ d'un ligand $<$ $\log\beta$ d'un autre ligand, uniquement celui ayant le $\log\beta$ le plus élevé pourra être détecté par fluorescence.

Polluant	Log β
EDTA	17,9
ATMP	18,8
DTPA	22,81
DOTA	24,2
DTPMP	30,64

FIGURE 3.7 – Rappel des constantes de complexation des polluants ajoutés

On s'attend donc logiquement à ce que le complexe entre le Tb et le DTPMP soit majoritaire (ou du moins soit formé en premier) et le complexe Tb-EDTA minoritaire.

pH initial	Température (en °C)	[EDTA] (en mmol/L)	[DTPA] (en mmol/L)	[DOTA] (en mmol/L)	[ATMP] (en mmol/L)	[DTPMP] (en mmol/L)	[Tb+3] (en mmol/L)	[Cl(-1)] (en mmol/L)
7	25	1	1	1	1	1	0,1	0,3
7	25	1	1	1	1	1	0,5	1,5
7	25	1	1	1	1	1	1	3
7	25	1	1	1	1	1	3	9
7	25	1	1	1	1	1	5	15
7	25	1	1	1	1	1	10	30
7	25	1	1	1	1	1	90	270
7	25	1	1	1	1	1	1000	3000
7	25	20	1	1	1	1	20	60
7	25	100	1	1	1	1	20	60
7	25	1000	1	1	1	1	20	60
7	25	5000	1	1	1	1	20	60

FIGURE 3.8 – Tableau des conditions initiales de la compétitivité en UP

La première modélisation permet d'étudier la compétitivité des polluants pour une même concentration donnée (ici, 1 mmol/L) sans prendre en compte les constantes de complexation. Pour la deuxième modélisation, l'EDTA, qui a le plus petit logK acceptable des polluants étudiés, est mis en large excès. Cela permettra de déterminer s'il existe une corrélation entre la concentration du polluant et la concentration du complexe Tb-polluant formé, ou si la concentration du complexe est toujours principalement dictée par la valeur du logK la plus élevée.

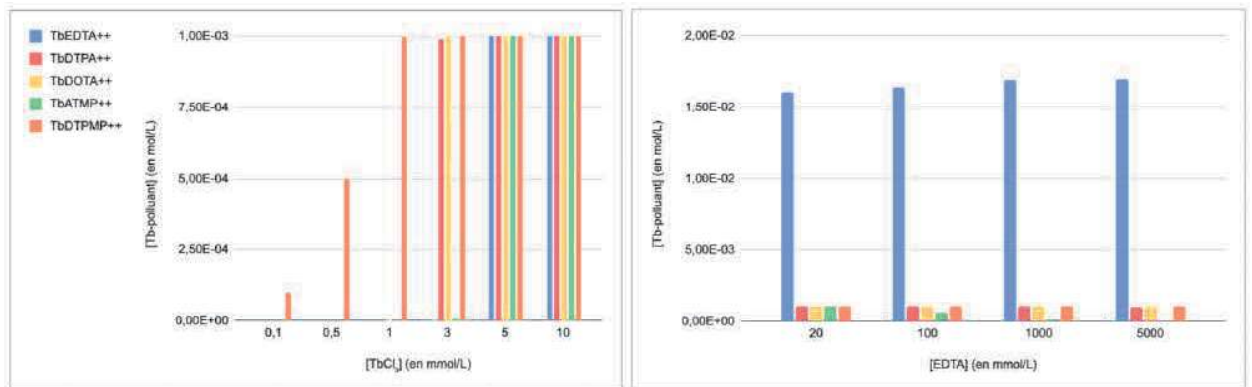


FIGURE 3.9 – Résultats de la compétitivité entre les polluants en UP à concentration égale (à gauche) et avec EDTA en excès (à droite)

Les résultats affirment l'hypothèse émise par l'UNIL. En effet, à concentration égale des différents polluants, c'est d'abord celui avec le plus fort logK qui va entièrement complexé (ici, le DTPMP), puis le deuxième, etc...

En revanche, dans le cas où l'EDTA a été mis en excès, ce sont toujours les logK qui régissent la spéciation. Cependant, pour une concentration d'EDTA à plus de 100mmol/L, les concentrations du complexe Tb-ATMP diminuent en faveur du complexe Tb-EDTA. On peut donc en conclure que la présence d'un polluant en excès ne change pas la spéciation du Tb, sauf dans le cas où le logK d'un autre polluant est très proche de celui du polluant qui se trouve en excès et que la concentration du polluant soit largement en excès, ce qui est assez rare dans l'environnement naturel.

Modélisation de la complexation dans une matrice naturelle - Modélisation avec 3 acides

Les acides gallique, vanillique et syringique ont été choisis car ces trois molécules possèdent un groupement phénolique et un groupe carboxylique. Ces groupements fonctionnels sont présents dans la NOM comme les acides humiques. Etant donné qu'ils sont souvent utilisés comme proxies pour la NOM dans les études de fluorescence, nous avons décidé de commencer les modélisations avec ces composés modèles avant de travailler avec la matrice réelle.

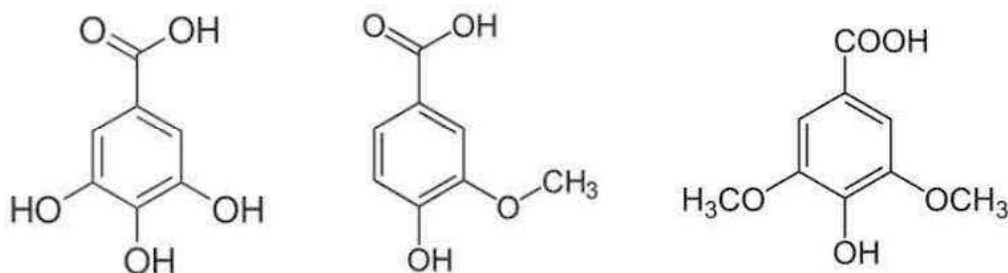
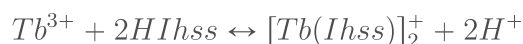


FIGURE 3.10 – Structure moléculaire des acides gallique, vanillique et syringique

Dans ce cas, la NOM a été modélisée comme étant la moyenne des masses molaires et des logK des 3 acides. Voir Annexe B

La NOM est donc modélisée comme étant un unique composé de masse molaire égale à 178,81 et de constantes de complexation avec le Tb égales à 7,26 et 13,25, en supposant que ce composé est un di-acide.

Les complexations sont régies par les équations suivantes :



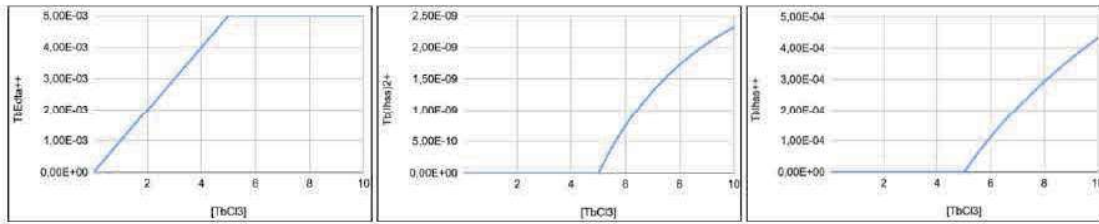


FIGURE 3.11 – Graphique des concentrations des différents complexes obtenus en fonction de $[\text{TbCl}_3]$

Ces résultats montrent que le Tb se complexe d’abord entièrement avec le polluant avant de se lier à la NOM. Cependant, ces résultats ont été invalidés par des données de temps de vie incohérentes obtenues par un stagiaire en M1 de Chimie.

3.3.5 Vérifications expérimentales

Les produits chimiques utilisés en laboratoire

- $\text{TbCl}_3 \cdot 6\text{H}_2\text{O}$

Le chlorure de terbium hexahydraté ($\text{TbCl}_3 \cdot 6\text{H}_2\text{O}$) est un composé chimique disponible chez Sigma-Aldrich avec une pureté de 99,9%. Ce sel, formé par l’association de Tb(III) avec trois ions chlorure et six molécules d’eau, se présente généralement sous forme de cristaux (comme c’est le cas ici) ou de poudre cristalline. Il correspond au TbCl_3 dans les modélisations précédentes.

- Les polluants cibles

Méthodologie et matériel

1. Préparation des solutions et dilutions : Pour assurer la précision et la fiabilité des résultats, il est essentiel de préparer les solutions à des concentrations exactes. Les dilutions sont effectuées de manière à ajuster les concentrations des produits chimiques trop élevées au départ.
2. Utilisation du robot Opentron OT-2 : Afin de maximiser la précision et la reproductibilité des préparations d’échantillons, un robot pipeteur OT-2 conçu par la société Opentron est utilisé. Ce robot automatisé permet de réaliser des pipetages avec une grande exactitude, éliminant ainsi les erreurs humaines potentielles et assurant la conformité des volumes ajoutés, tout en étant un gain de temps précieux. Grâce à l’Opentron, la manipulation des échantillons est standardisée, ce qui garantit que chaque étape de la préparation est exécutée avec la même précision, d’un échantillon à l’autre.



FIGURE 3.12 – Exemple d’utilisation du robot Opentron OT-2 pour la préparation d’échantillons

Le robot est équipé de deux emplacements pour pipettes, l’un de $1000 \mu\text{L}$ et l’autre de $300 \mu\text{L}$, pour des prélèvements de volumes précis. Il comprend également 11 positions pour les racks et un emplacement dédié pour la poubelle. Son fonctionnement est piloté par un logiciel basé sur Python, ce qui permet d’ajuster les protocoles de pipetage selon les besoins spécifiques des expériences. Le codage de celui-ci a été réalisé par les mathématiciens, qui m’ont appris à l’utiliser et à l’implémenter dans le logiciel *Opentron*. Un exemple de code correspondant à l’exemple d’utilisation du robot de la figure précédente est disponible en Annexe D.

3. Mesure du pH : La mesure du pH est un paramètre crucial dans cette étude, car elle permet de valider les modélisations Phreeq-C et donc la spéciation du Tb(III). Un pH-mètre est utilisé pour mesurer le pH des solutions après chaque préparation. Chaque échantillon est réalisé à l’identique 3 fois et 3 mesures de pH sont réalisées pour chaque échantillon, ce qui fait 9 mesures de pH pour une même composition. La moyenne des 9 pH obtenus est enfin calculée afin de la comparer aux pH théoriques.



FIGURE 3.13 – Exemple d'utilisation du robot Opentron OT-2 pour la préparation d'échantillons

Comparaison des pH expérimentaux et théoriques

	Echantillon	pH théorique	pH expérimental	Delta
EP Acétate	EP14	3,574	3,25	0,324
	EPA1	3,309	2,98	0,329
	EPA8	2,719	2,39	0,329
EP polluant unique	EPE6	3,086	2,81	0,276
	EE2	2,386	2,15	0,236

FIGURE 3.14 – Comparaison des pH expérimentaux et théoriques en UP

Un écart constant d'environ 0,3 a été observé pour chaque mesure. Pour éviter de surcharger l'analyse, seules trois valeurs correspondant à des polluants spécifiques ont été sélectionnées (les valeurs correspondent à l'EDTA dans le tableau), car les résultats étaient similaires pour l'ensemble des polluants testés. Cet écart faible et régulier valide la modélisation effectuée, confirmant la cohérence entre les données expérimentales et les prédictions théoriques.

3.3.6 La matrice IHSS comme association de la SH et des ions en solution

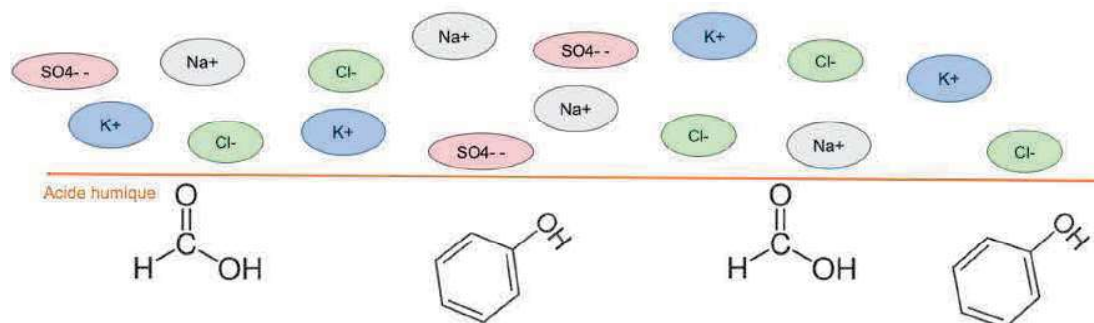


FIGURE 3.15 – Schéma simplifié représentatif de l’IHSS utilisé dans la modélisation numérique

La SH comme une surface spécifique

Les groupes carboxyliques et phénoliques dans les substances humiques (SH) constituent des sites actifs de surface qui interagissent avec les ions et les molécules en solution par adsorption et complexation, plutôt que de simplement se dissoudre dans l’eau, d’où l’hypothèse de les traiter comme une surface spécifique dans le logiciel Phreeq-C. [38] [39]

Prise en compte des ions en solution

La capacité des substances humiques à capturer et à former des complexes avec les ions métalliques est essentielle pour la gestion des métaux dans les systèmes aquatiques et a un impact majeur sur la qualité environnementale. [39] Cependant, les constantes sont indisponibles pour Na⁺ et K⁺ car ces ions ont une faible affinité pour la complexation avec de nombreux ligands, ce qui rend les constantes de stabilité très faibles et souvent négligées dans les études détaillées de complexation. [40] Additionné au fait que les concentrations des ions dans la matrice de l’IHSS sont très faibles (voir Annexe C [39]), l’hypothèse de l’absence de complexation entre les ions métalliques en solution et les groupes fonctionnels de la surface humique a été émise.

En revanche, quelques constantes de stabilité entre les ions Na⁺ et K⁺ et les polluants ont été trouvés. [41]

Pour faciliter les modélisations et par manque de données, nous travaillons uniquement avec les deux cations (Na⁺ et K⁺) et les deux anions (SO₄²⁻ et PO₄²⁻) ayant les concentrations les plus élevées.

TABLE 3.1 – Constantes de complexation Na^+ et K^+ pour les polluants cibles

Acide	Na^+	K^+
EDTA	1.8	1.0
DTPA	≈ 2.0	0.9
NTA	1.2	0.5
DOTA	–	–
ATMP	–	–
DTPMP	3.71	3.32

Choix des paramètres Phreeq-C pour la SH

Plusieurs paramètres ont été fixés pour les prochaines modélisations concernant la SH, en plus des paramètres initialement introduits à la partie 3.3.2.

- La densité des sites pour les groupes phénoliques et carboxyliques [24]
- La surface spécifique : fixée à $600 \text{ m}^2/\text{g}$ par défaut [34]
- La masse : fixée à 0.33g par défaut.

```

SURFACE 1
  -equilibrate 1
  -sites_units density
  Humin_h 0.00247 600.0 0.33
  Humin_c 0.01121 600.0 0.33
SURFACE_MASTER_SPECIES
  Humin_h Humin_hOH
  Humin_c Humin_cCOOH
EQUILIBRIUM_PHASES 1
  Tb(OH)3 0.0
  CO2(g) -3.5 10
SURFACE_SPECIES
  Humin_hOH = Humin_hOH
  log_k 0.0
  Humin_cCOOH = Humin_cCOOH
  log_k 0.0

  Humin_hOH = Humin_hO- + H+
  log_k -7.5 # Constante de dissociation du groupe phénolique
  Humin_cCOOH = Humin_cCOO- + H+
  log_k -4.5 # Constante de dissociation du groupe carboxylique

  Humin_cCOOH + Tb+++ = Humin_cCOOTb++ + H+
  log_k 7.26

  Humin_hOH + Tb+++ = Humin_hOTb++ + H+
  log_k 8
  
```

1 Densité des sites (en mol/g)
 2 Surface spécifique (en m^2/g)
 3 Masse (en g)

FIGURE 3.16 – Paramètres d’entrée systématiques pour la SH

De cela découle les équations associés à cette surface :

— Les équations de définition des espèces de surface :



$$\log K = 0.0 \quad (3.2)$$



$$\log K = 0.0 \quad (3.4)$$

— Les équations de dissociation :



$$\log K = -7.5 \quad (\text{Constante de dissociation du groupe phénolique}) \quad (3.6)$$

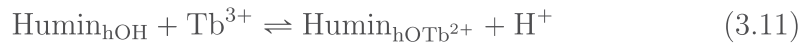


$$\log K = -4.5 \quad (\text{Constante de dissociation du groupe carboxylique}) \quad (3.8)$$

— Les équations de complexation avec le Tb :



$$\log K = 7.26 \quad (3.10)$$



$$\log K = 8 \quad (3.12)$$

Des études de sensibilité ont été menées sur les constantes de complexation pour différents acides phénoliques et carboxyliques, sur la valeur de la surface spécifique de 600 à 800 m^2/g , sur la masse (de 0 à 0.33g) et sur les constantes de dissociation (+/- 0.5). Aucun changement n'a été observé.

3.4 Modélisation de la matrice réelle de l'IHSS

3.4.1 Premiers essais : réglage du pH

Avant de commencer l'exploitation des fichiers de sortie et l'ajout des différents polluants, il fallait s'assurer que les modèles reflétaient correctement la matrice réelle étudiée. Dans notre cas, cela étant vérifiable par calcul de pH, le pH a été mesuré pour une gamme de concentrations en Tb allant de 0 à 20ppm avec une concentration d'IHSS fixée à 5ppm. Cette valeur sera notre valeur de référence pour toute la suite de la section.

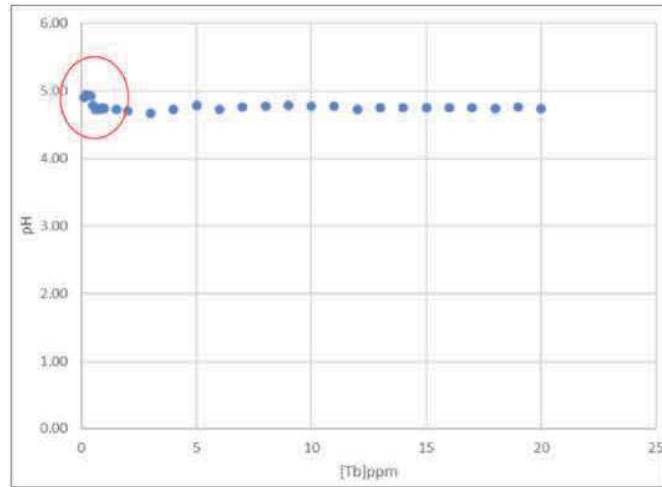


FIGURE 3.17 – Courbe de pH en fonction de $[TbCl_3]$ attendue

Phase 1 : pH trop élevé avec allure de courbe similaire

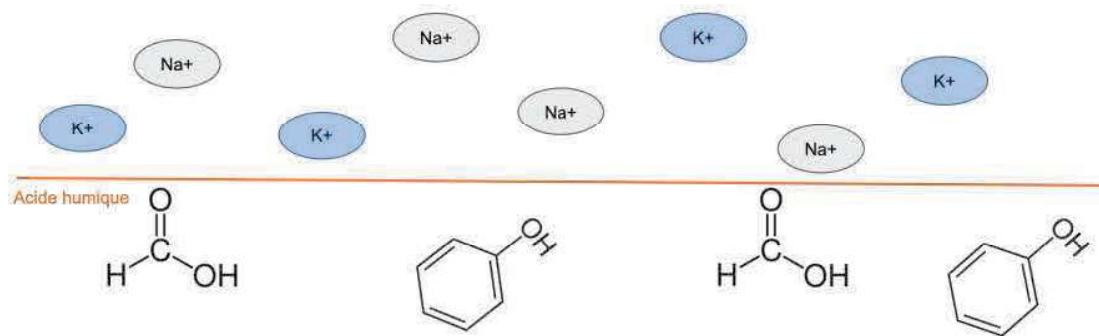


FIGURE 3.18 – Schéma représentatif de l’IHM utilisé dans la première modélisation numérique

Au début des modélisations de la matrice réelle de l’IHM, l’ajout des anions a été omis, l’électroneutralité n’était donc plus respecté. La formule de l’électroneutralité exprime le principe selon lequel, dans une solution électriquement neutre, la somme des charges positives doit être égale à la somme des charges négatives, comme suit :

$$\sum_i z_i C_i = 0 \tag{3.13}$$

où :

- z_i : La valence de l’ion i . La valence est le nombre de charges électriques que porte l’ion.

— C_i : La concentration de l'ion i dans la solution, exprimée en moles par litre (mol/L).

Cet oubli explique en partie les résultats faussés obtenus suivants :

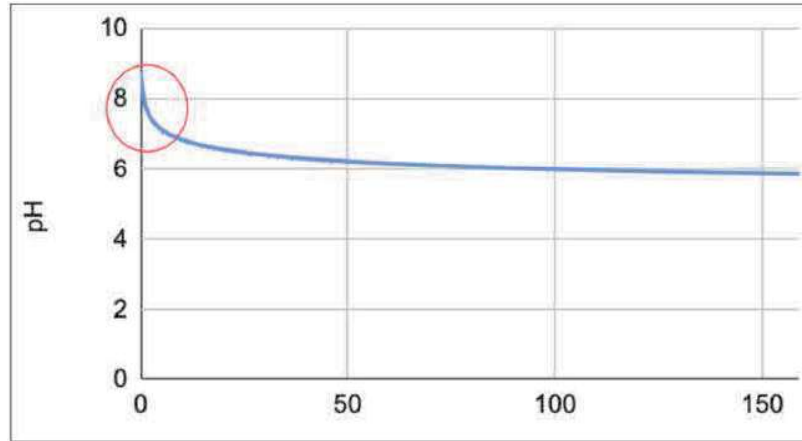


FIGURE 3.19 – Courbe de pH obtenue au 1er essai

On observe un pH 2 points au-dessus du pH attendu en l'absence de Tb et une variation de 1.5 points entre 0 et 20ppm de Tb. Malgré ce pH trop élevé, l'allure de courbe est similaire.

Plusieurs hypothèses ont alors été émises concernant le choix des groupes carboxyliques et phénoliques et des paramètres de surface. L'étude de sensibilité évoquée à la partie 3.3.6. a écarté ces hypothèses.

Phase 2 : pH élevé avec un off-set par rapport à la courbe attendue

Après plusieurs discussions, l'équipe s'est rendue compte que cette erreur provenait de l'absence de la prise en compte des anions dans la modélisation, et donc du non-respect de l'électroneutralité. Cette nouvelle étape se base alors sur la composition de l'IHSS représentée par le schéma en début de partie 3.3.6. En revanche, en ne prenant en compte que les ions Na^+ , K^+ , SO_4^{2-} et PO_4^{2-} , l'électroneutralité n'était toujours pas respecté car la concentration des deux anions est plus faible que celle des deux cations. Pour compenser cela, 0.00506 mmol/L ont été additionnés à la concentration initiale de Cl^- .

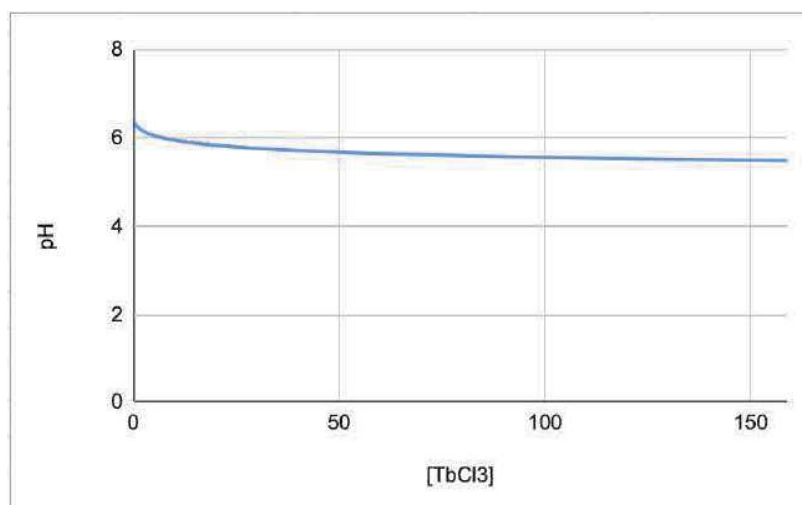


FIGURE 3.20 – Courbe de pH obtenue au 2ème essai

Malgré une étude de sensibilité approfondie, je n'ai pas réussi à améliorer ces résultats. Après comparaison entre les pH expérimentaux et théoriques, on retrouve malheureusement ce point d'écart.

3.4.2 Résultats

Tous les polluants ont été mis en solution à une même concentration de 1mmol/L en faisant varier la concentration en Tb de 0 à 3 mmol/L pour voir la compétitivité des polluants au sein d'une matrice réelle et les potentielles interférences que peut causer celle-ci sur la spéciation du Tb(III). Le fichier input de cette modélisation est disponible en Annexe E.

Tb(3)		1.010e-03	
TbDtpmp+2	1.000e-03		
TbDota+2	9.640e-06		
TbDtpa+2	3.964e-07		
TbNta+2	9.960e-10		
Tb+3	1.028e-19		
TbCl+2	3.368e-22		
TbSO4+	5.262e-23		
TbCl2+	3.929e-25		
TbOH+2	2.682e-25		
TbCl3	3.907e-28		
Tb(SO4)2-	2.209e-28		
TbCl4-	4.342e-31		
TbO+	2.447e-31		
TbO2H	8.079e-38		
TbO2-	0.000e+00		

Humin_c		3.685e-06 moles	
Species	Moles	Mole Fraction	
Humin_cCOOH	3.654e-06	0.992	
Humin_cCOO-	3.117e-08	0.008	
Humin_cCOOTb+2	8.528e-16	0.000	

Humin_h		8.121e-07 moles	
Species	Moles	Mole Fraction	
Humin_hOH	8.120e-07	1.000	
Humin_hO-	6.926e-12	0.000	
Humin_hOTb+2	1.041e-15	0.000	

Tb(3)		2.995e-03	
TbDtpmp+2	1.000e-03		
TbDota+2	9.979e-04		
TbDtpa+2	9.508e-04		
TbNta+2	4.630e-05		
Tb+3	5.845e-15		
TbCl+2	4.848e-17		
TbSO4+	2.225e-18		
TbCl2+	1.502e-19		
TbOH+2	1.371e-20		
TbCl3	4.184e-22		
Tb(SO4)2-	8.550e-24		
TbCl4-	1.375e-24		
TbO+	1.180e-26		
TbO2H	3.873e-33		
TbO2-	8.264e-40		

Humin_c		3.685e-06 moles	
Species	Moles	Mole Fraction	
Humin_cCOOH	3.654e-06	0.991	
Humin_cCOO-	3.187e-08	0.009	
Humin_cCOOTb+2	3.908e-11	0.000	

Humin_h		8.121e-07 moles	
Species	Moles	Mole Fraction	
Humin_hOH	8.120e-07	1.000	
Humin_hOTb+2	4.773e-11	0.000	
Humin_hO-	7.083e-12	0.000	

FIGURE 3.21 – Résultats de la modélisation sur la compétitivité des polluants au sein de la matrice réelle

On compare ici deux fichiers de sortie pour deux concentrations en Tb différentes, de 1mmol/L et de 3mmol/L. Tout d’abord, on remarque que les 4 complexes qui nous intéressent sont placés en premier dans la spéciation du Tb(III). La simulation montre comment le terbium interagit avec ces acides complexants avec un pH initialement neutre mais qui devient acide en raison des interactions chimiques fortes avec les agents chélatants. D’autre part, on note que la présence de la NOM n’altère absolument pas la spéciation du Tb, puisque les concentrations des complexes Tb-SH est de l’ordre d’au moins 10^{-12} mol/L, ce qui est un très bon point pour le projet.

3.5 Automatisation de Phreeq-C à l’aide de Python

Code complet : Annexe F

3.5.1 Introduction

L’automatisation de Phreeq-C via Python est une initiative qui permet de résoudre deux problèmes majeurs lors de l’utilisation de Phreeq-C, à savoir la redondance et les éventuelles erreurs liées au changement manuel des fichiers d’entrée pour un même tableau. En générant un grand nombre de simulations de façon efficiente, ce code apporte un gain de temps certain et assure la reproductibilité des simulations.

3.5.2 Structure du code

Ce script se divise en deux parties distinctes, ce qui sera plus facile à manipuler pour les prochains utilisateurs :

1° Section à modifier "PARAMS" : C'est dans cette section que l'utilisateur doit ajuster les paramètres selon ses différents besoins (chemins d'accès, concentrations des éléments, nom des valeurs à prélever dans les fichiers d'entrée et de sortie...)

2° Section non modifiable : Cette partie inclut les fonctions de base et les étapes de traitement essentielles au bon fonctionnement du script et qui ne doivent pas être modifiées.

3.5.3 Les fonctions clés

La fonction "calculate_concentrations"

```
def calculate_concentrations(params):  
    A = np.arange(params["elt_start"], params["elt_end"], params["elt_step"])  
    B = [3 * Tb_conc + params["cl_electroneutrality_adjustment"] for Tb_conc in A]  
    return A, B
```

FIGURE 3.22 – Code de la fonction calculate_concentrations

La fonction "calculate_concentrations" calcule deux gammes de concentrations pour les éléments chimiques en prenant les paramètres :

- "elt_start" : La concentration initiale de l'élément cible.
- "elt_end" : La concentration finale de l'élément cible.
- "elt_step" : Le pas entre chaque concentration de l'élément cible.

Elle génère une série de valeurs pour l'élément de départ (par exemple, le terbium Tb) et ajuste les concentrations en ions chlorure (Cl) pour maintenir l'électroneutralité. Le paramètre "cl_electroneutrality_adjustment" est optionnel, il m'a été utile pour compenser le manque d'anions par rapport aux cations lors des simulations de la matrice IHSS. Cette fonction renvoie les listes A et B qui sont respectivement la liste de concentrations de l'élément cible et la liste de concentrations correspondantes d'ions chlorure.

Les fonctions "extract_input_data" et "extract_output_data"

```
def extract_input_data(input_content, params_to_extract):
    lines = input_content.split('\n')
    data = {}
    for line in lines:
        for param in params_to_extract:
            if f' {param} ' in line:
                data[param] = str(line.split()[1])
    return data
```

FIGURE 3.23 – Code de la fonction extract_input_data

```
def extract_output_data(output_content, params_to_extract):
    lines = output_content.split('\n')
    data = {}
    for line in lines:
        for param in params_to_extract:
            if param["search_str"] in line and param["key"] not in data:
                value = line.split('=')[1].split()[0].strip() if '=' in line else line.split()[1].strip()
                data[param["key"]] = str(value)
    return data
```

FIGURE 3.24 – Code de la fonction extract_output_data

Ces fonctions extraient les données spécifiques à partir du contenu d'un fichier d'entrée et de sortie à partir des paramètres suivants :

- "input_content" : Le contenu du fichier d'entrée. Il est important que celui-ci soit converti sous forme de chaîne de caractères.
- "params_to_extract" : Une liste des paramètres à extraire du contenu du fichier.

Elles fonctionnent en parcourant chaque ligne du contenu du fichier et en cherchant les paramètres indiqués. Si un paramètre est trouvé dans une ligne, sa valeur est extraite et stockée dans une liste. Ces fonctions renvoient respectivement une liste avec les paramètres et leurs valeurs extraites.

La fonction "Run_phreeqc_simulation"

```
def run_phreeqc_simulation(params, tb_concentrations, cl_concentrations):
    results = []

    with open(params["input_file_path"], 'r') as f:
        input_content = f.read()

    for i in range(len(tb_concentrations)):
        modified_content = input_content.replace('c{tb}', str(tb_concentrations[i])).replace('c{cl}', str(cl_concentrations[i]))
        modified_content = f'DATABASE {params["database_path"]}\n' + modified_content

        # Définir le chemin complet pour le fichier temporaire
        temp_input_file_path = os.path.join(params["output_directory"], f'temporaire{i}.txt')

        with open(temp_input_file_path, 'w') as f:
            f.write(modified_content)

        output_file_path = f'{params["output_file_base"]}{i}.txt'
        command = f'{params["phreeqc_executable_path"]} {temp_input_file_path} {output_file_path} {params["database_path"]}'

        process = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        output = process.communicate()

        input_data = extract_input_data(modified_content, params["input_params"])
        results.append(input_data)

    df = pd.DataFrame(results)
    return df
```

FIGURE 3.25 – Code de la fonction run_phreeqc_simulation

La fonction "run_phreeqc_simulation" exécute des simulations Phreeq-C pour différentes concentrations de l'élément cible et d'ions chlorure, en utilisant les paramètres spécifiés :

- "input_file_path" : Chemin d'accès du fichier input.
- "output_file_base" : Chemin d'accès du dossier output.
- "database_path" : Chemin d'accès de la base de données.
- "tb_concentrations" : Une liste de concentrations de l'élément cible (créée à partir de la fonction calculate_concentrations) .
- "cl_concentrations" : Une liste de concentrations de chlorure (créée à partir de la fonction calculate_concentrations).

C'est grâce à la fonction "subprocess.Popen" dans la bibliothèque "subprocess" que le logiciel externe Phreeq-c peut être exécuté.

Pour chaque paire de concentrations, elle modifie le fichier d'entrée Phreeq-C, exécute la simulation, et extrait les données pertinentes après modifications en appelant la fonction "extract_input_data". Pour chaque simulation, un fichier d'entrée temporaire est enregistré avec les nouvelles valeurs de concentrations. Les résultats sont stockés dans un DataFrame "df" et renvoyés. Cela est possible grâce à la bibliothèque "pandas".

La fonction "convert_and_extract_outputs"

```
def convert_and_extract_outputs(params, num_files):
    results = []
    for i in range(num_files):
        out = os.path.join(params["output_directory"], f'IHSSTbEN{i}.txt')
        with open(out, 'r', encoding='iso-8859-1') as file:
            content = file.read()
        with open(out, 'w', encoding='utf-8') as file:
            file.write(content)

        print(f"Le fichier {out} a été converti en UTF-8.")

    output_data = extract_output_data(content, params["output_params"])
```

FIGURE 3.26 – Code de la fonction convert_and_extract_outputs

La fonction "convert_and_extract_outputs" traite les fichiers de sortie générés par Phreeq-C pour extraire les données importantes à partir de :

- "output_params" : Les paramètres à extraire des fichiers de sortie.
- "num_files" : Le nombre de fichiers de sortie à traiter.

Elle lit chaque fichier de sortie, convertit son encodage en UTF-8 pour éviter les erreurs, et extrait les données pertinentes en appelant la fonction "extract_output_data".

3.5.4 Utilisation du code

Pour utiliser le code, il suffit de modifier les valeurs explicitées dans la section "PARAMS" et de lancer le script via un environnement Python. Il faut indiquer le chemin d'accès du fichier d'entrée et remplacer, dans ce fichier, les concentrations variables entre chaque simulation par "ctb" (pour l'élément cible) et "ccl"(pour les ions chlorure) comme fait à la figure ci-dessous.

```
SOLUTION 1
pH 7 charge
temp 25
pe 4
redox pe
units mmol/kgw
Na 0.00304
K 0.00239
Tb ctb
Cl(-1) ccl
S 0.00038
-water 1 # kg
```

FIGURE 3.27 – Exemple d'un fichier d'entrée avec les concentrations de Tb et de Cl variables

Celui-ci renverra alors un fichier CSV contenant tous les résultats de la gamme de concentrations indiquée au début des paramètres pour faciliter ainsi l'interprétation des simulations (Figure 3.18).

validationprgm

Tb	Cl(-1)	pH	Erreur	Total Tb	Tb3+	Tb3+ (%)
0.0	0.00506	6.385	0.0	1.937e-05	7.649e-06	39.488900361383585
0.0005	0.00656	6.369	0.0	5E-07	4.825e-07	96.5
0.001	0.00806	6.354	0.0	1E-06	9.662e-07	96.62
0.0015	0.00956	6.341	0.0	1.5e-06	1.451e-06	96.73333333333332
0.002	0.01106	6.329	0.0	2E-06	1.936e-06	96.8
0.0025	0.01256	6.317	0.0	2.5e-06	2.422e-06	96.88
0.003	0.01406	6.307	0.0	3E-06	2.909e-06	96.96666666666668
0.0035	0.01556	6.297	0.0	3.5e-06	3.396e-06	97.02857142857142
0.004	0.01706	6.288	0.0	4E-06	3.884e-06	97.10000000000002
0.0045	0.01856	6.279	0.0	4.5e-06	4.372e-06	97.15555555555557

FIGURE 3.28 – Exemple d'un fichier de sortie en format CSV

Identification des polluants dans les effluents industriels et municipaux via spectroscopie de fluorescence et IA

4.1 Objectif et place dans le sujet global

Cette partie correspond à la tâche 3 du projet, à savoir le développement de l'IA pour l'identification des polluants.

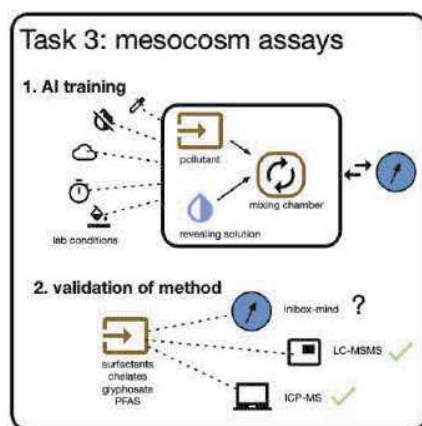


FIGURE 4.1 – Place dans le projet VAI-TRF de l'identification des polluants par l'IA

4.2 Enrichir la base de données de l'IA à l'aide de spectres 3D

4.2.1 Présentation des spectres 3D

[42]

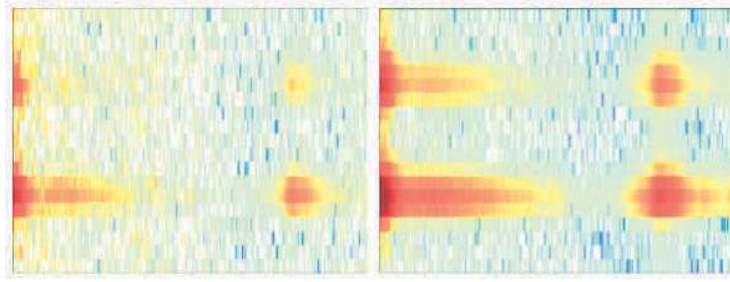


FIGURE 4.2 – Exemple de spectres 3D, respectivement N5MD5S11 et N10M0T01R06

Ces spectres sont générés à partir de l'analyse TRF par spectrophotomètre de chaque échantillon. Chacun est exposé à une excitation couvrant une large gamme de longueurs d'onde, allant de 230 à 520 nm. Par la suite, l'intensité de fluorescence a été enregistrée pour chaque longueur d'émission située entre 520 et 610 nm.

4.2.2 Choix des solutions

Pour chaque solution, on introduit une concentration fixe de 5ppm d'IHSS, une concentration variable du polluant cible allant de 2 à 80 ppm et une quantité de solution révélatrice assurant le maintien des mêmes rapports molaires Tb/polluant entre les différentes séries.

4.2.3 Utilisation du robot Opentron OT-2 et du spectrophotomètre pour les mesures

Pour préparer les cuvettes, le robot Opentron OT-2 a été utilisé de la même manière qu'à la partie 3.3.5. Ensuite, c'est le spectrophotomètre qui permet d'obtenir les spectres 3D.

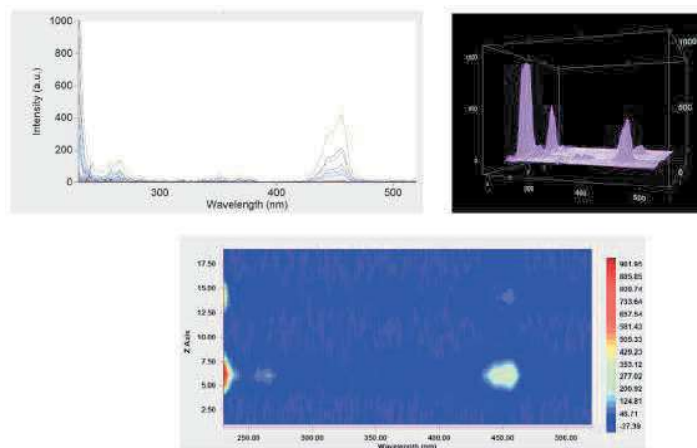


FIGURE 4.3 – Spectres obtenus pour un échantillon

4.3 Apprentissage de l'IA

4.3.1 Génération de données

Une fois que les échantillons ont été analysés, les données collectées sont d'abord sauvegardées dans un fichier au format CSV. Ensuite, elles sont converties en une image au format PNG. Cette image est partagée avec les mathématiciens du laboratoire, qui se chargent de traiter et d'analyser ces informations. Le but de cette démarche est de fournir les données nécessaires pour alimenter une IA. Celle-ci aura pour mission de déterminer la nature précise des polluants présents dans les échantillons analysés.

A la fin juillet, nous possédons une base de données contenant 1008 échantillons, chacun étant associé à cinq caractéristiques spécifiques :

- La matrice
- La solution révélatrice
- Le ratio ion/polluant
- Un fichier CSV représentant les mesures d'un spectre 3D

La dernière caractéristique est la classe à prédire qui correspond au polluant en excès contenu dans l'échantillon. Il y a 5 classes possibles : Il existe 5 classes de réponse possibles :

- DTPMP
- NTA
- DOTA
- DTPA
- Aucun : cette classe concerne les échantillons ne contenant pas de polluants.


Sample	Pollutant	Concentration Pollutant ppm	Matrix	Revelatrice	Ratio mol ion/poll	CSV
D05M0T01S01	DTPMP	5	IHSS	Tb	0.45	

FIGURE 4.4 – Caractéristiques d'un échantillon

Ces caractéristiques fournissent des informations essentielles pour chaque échantillon, permettant une analyse détaillée et comparative au sein de la base de données.

Dans notre base de données qui compte 1008 échantillons, seuls ceux qui présentent une matrice de type "IHSS" et un ratio ion/polluant inférieur à 1 seront sélectionnés pour alimenter notre modèle de classification. Au final, cette sélection réduit notre jeu de données à 288 échantillons. Voici la répartition de ces échantillons :

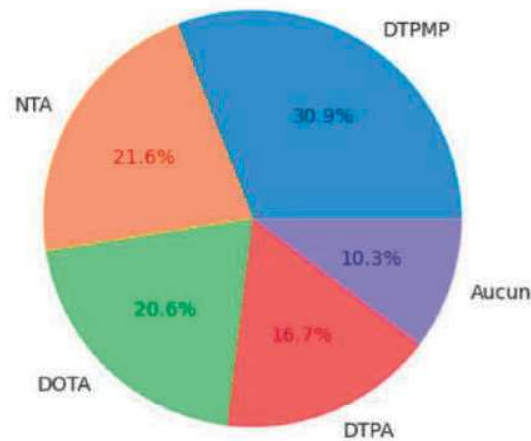


FIGURE 4.5 – Répartition des classes dans la base d'apprentissage

De la base de données considérée, 80% formeront la base d'apprentissage qui permet de construire le modèle et 20% la base de test qui permet d'évaluer le modèle. Ces deux étapes constituent les phases dites d'apprentissage et d'entraînement, respectivement.

4.3.2 Apprentissage de l'IA

Le modèle d'IA sera formé pour identifier les polluants en analysant des images qui, en réalité, correspondent à des spectres 3D. Ces images représentent visuellement les données spectrales, permettant à l'IA de reconnaître les signatures spécifiques des différents polluants. 18 méthodes de traitements d'images (A_1, A_2, \dots, A_{12}) ont été considérées, établissant chacune une fenêtre spécifique et un niveau de couleurs différents. En effet, chaque méthode consiste à découper un spectre en une partie de l'image donnée (soit l'image complète, soit la moitié, soit le quart), permettant à l'IA d'examiner différentes parties de l'image avec précision. Ces segments sont ensuite analysés avec deux niveaux de couleurs différents : le 1 correspondant à l'image de base, et le 2 correspondant à la même image mais à un niveau de couleur normalisé, ce qui aide à accentuer certains détails pour une meilleure reconnaissance des polluants. Par exemple, la méthode D1 se concentre uniquement sur le quart supérieur gauche de l'image, permettant ainsi d'extraire des informations spécifiques de cette zone (la

méthode D2 est donc la même image mais en couleurs normalisés).

On peut observer les différentes images obtenues, correspondant chacune à une méthode :

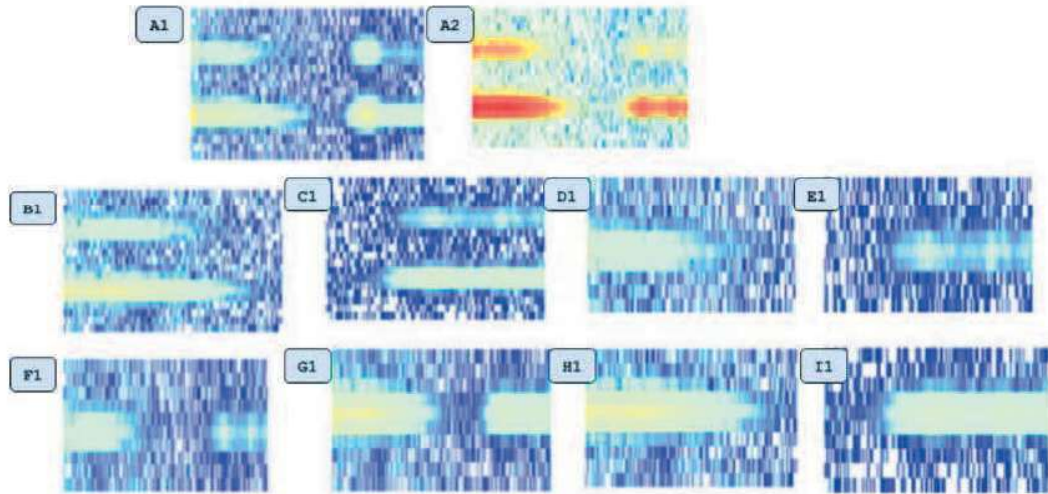


FIGURE 4.6 – Images des différentes méthodes

Pour chaque méthode de traitement d'images, un modèle de réseau de neurones convolutifs à 8 couches a été développé. Bien que l'architecture de base reste identique pour tous les modèles, les hyperparamètres des différentes couches, ainsi que la durée d'apprentissage, sont ajustés pour chaque modèle, ce qui les rend distincts les uns des autres. Ces ajustements permettent d'optimiser la performance du réseau en fonction des spécificités de chaque méthode, garantissant ainsi une meilleure précision dans l'identification des polluants. Ce travail de fond est réalisé par les mathématiciens.

4.3.3 Premiers résultats : l'IA est-elle fiable pour l'identification ?

Après comparaison des résultats des 18 modèles de traitement, les trois meilleurs méthodes ont été retenues : A2, C2 et I2. Les résultats complets sont disponibles en annexe

Méthode nom	Accuracy	Loss	Matrice de confusion	Graph évaluer overfitting	Graph évaluer overfitting
A2	1.0000	0.0655	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 0 0 0 6 0 4 0 0 0 0 12 </pre>		
C2	0.96491229534 14917	0.0599228 292703628 54	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 10 2 0 3 0 0 0 6 0 4 0 0 0 0 12 </pre>		
I2	1.0	0.0241837 538778781 9	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 0 0 0 6 0 4 0 0 0 0 12 </pre>		

FIGURE 4.7 – Résultats des 3 méthodes retenues

La première colonne renvoie l'accuracy, qui mesure le taux de prédictions correct sur l'ensemble des échantillons. Dans le cas général, sa formule est la suivante : [43]

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}} = \frac{TP+TN}{TP+TN+FP+FN}$$

où :

- TP : Vrai positif, bonne prédiction ;
- TN : Vrai négatif, bonne prédiction ;
- FP : Faux positif, mauvaise prédiction ;
- FN : Faux négatif, mauvais prédiction.

Plus simplement, plus il y a d'erreurs de prédictions, plus l'accuracy diminuera et plus il y a de prédictions correctes , plus l'accuracy augmentera.

On retiendra en particulier I2, qui a les meilleurs scores et de belles courbes. Les courbes de perte pour les jeux de données d'entraînement et de test convergent bien vers 0 ainsi le modèle apprend efficacement et réduit l'erreur au fil du temps. La perte d'entraînement (courbe bleue) diminue régulièrement, tandis que la perte de test (courbe orange) suit une tendance similaire, suggérant que le modèle généralise bien aux données inconnues. En plus, il n'y a aucun signe de sur-apprentissage, car les courbes d'entraînement et de test se suivent étroitement au fil du temps. Cela implique que le modèle peut être appliqués à de nouvelles données.

Predicted	0	1	2	3	4
True					
0	63	0	0	0	6
1	0	65	0	0	1
2	2	0	55	0	0
3	2	0	2	13	3
4	23	1	3	1	72

FIGURE 4.8 – Dernière matrice de confusion obtenue - *Travail réalisé par les mathématiciens*

Les classes sont les suivantes :

- 0 : DTPMP,
- 1 : NTA,
- 2 : DOTA,
- 3 : DTPA,
- 4 : Aucun polluant.

Le NTA est toujours correctement prédit à 100% tandis que le DOTA présente seulement deux erreurs, probablement dues à des raisons chimiques spécifiques.

4.3.4 Piste de résolution des problèmes de non-identification des polluants

Pour résoudre les problèmes d'identification des polluants, il est crucial de traiter plusieurs défis clés. La principale difficulté est le nombre insuffisant d'échantillons, ce qui compromet la précision des résultats. On remarque ensuite que le DTPMP est souvent confondu avec des échantillons non polluants, et inversement. Pour améliorer la précision, il faudrait impérativement augmenter le volume des données, équilibrer les classes, et ajuster les caractéristiques chimiques qui pourraient affecter les prédictions.

Vérifier la fiabilité du projet sur le terrain

5.1 Présentation du dispositif de terrain

Pour répondre à la détection des polluants de manière continue, il était primordiale d'imaginer un système facilement transportable capable de gérer un flux d'eau et de l'analyser rapidement. Cela a été possible grâce au travail des mathématiciens, accompagnés de GLINCS et AXINT. La création d'une application simple d'utilisation pour répondre à la demande et coordonner cette circuit-box a été réalisée.

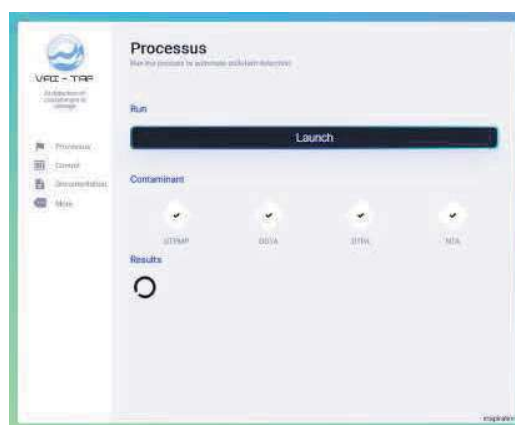


FIGURE 5.1 – Interface de l'application du projet VAI-TRF mise en place par les mathématiciens

Il suffit de prendre en compte les polluants dont on souhaite analyser la présence dans l'environnement étudié, et l'application renvoie dans la partie "Results" le(s) polluant(s) identifié(s).

Ce système, en plus d'un écran, se compose de 3 éléments reliés entre eux :

- L'inibox
- Le robot Opentron-OT2

5.1.1 L'inibox

L'inibox est un détecteur mobile et résistant développé par GLINCS et AXINT dans le cadre du projet *Détection de traces d'éléments lanthanides par fluorescence en temps résolu : application industrielle au marquage anti contrefaçon et à l'analyse chimique* d'Arthur Marais. [21] L'objectif était d'imaginer une analyse par TRF facilement déplaçable et manipulable et capable de résister à des conditions variables de l'environnement extérieur. Certains appareils existent déjà sur le marché mais aucun ne correspond au cahier des charges pour un usage industriel de cette méthode.

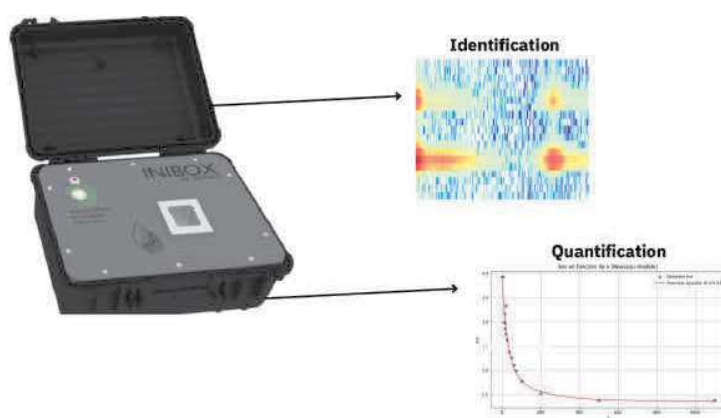


FIGURE 5.2 – Photographie de l'Inibox - Image réalisée par Hiba

Le système Inibox remplace finalement le spectrophotomètre en générant des spectres 3D et des courbes de TRF. Il possède un porte-cuvette adapté aux échantillons liquides (remarquable par le carré blanc sur la figure ci-dessus).

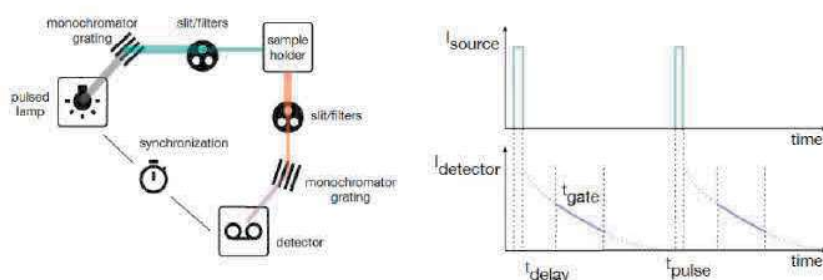


FIGURE 5.3 – Dispositif TRF à l'intérieur de l'Inibox

Sous la plaque en métal, on retrouve les éléments nécessaires à un système TRF : la lumière pulsée émise par une source passe d'abord à travers un monochromateur à

réseau, qui la disperse en un spectre de longueurs d'onde distinctes. Un filtre initial est utilisé pour ajuster cette lumière avant qu'elle ne traverse l'échantillon. Après cela, la lumière modifiée par le liquide passe par un deuxième filtre pour éliminer les longueurs d'onde indésirables, puis elle peut être analysée par un second monochromateur à réseau pour une séparation plus précise des longueurs d'onde. Enfin, un détecteur mesure l'intensité lumineuse pour chaque longueur d'onde, fournissant des données sur certaines caractéristiques de l'échantillon.

L'Inibox est ainsi nécessaire pour répondre aux questions d'identification et de quantification des polluants. Actuellement, la fonction mapping 3D est encore manquante sur celle-ci.

5.1.2 Le robot Opentron OT-2

Le robot, à l'aide des micro-pipettes, va venir prélever un échantillon d'eau à analyser pour le placer dans un nouveau contenant tenu par un support adapté. Celui-ci a été imaginé et réalisé à l'imprimante 3D par Benoit Hautefeuille.



FIGURE 5.4 – Photographie du support - *Image réalisée par Hiba*

L'Opentron va ensuite venir prélever une dose de Tb placé dans un réservoir au préalable pour l'ajouter à cet échantillon. Enfin, l'échantillon va être homogénéisé grâce à agitateur magnétique contenu dans le support (visible sur la figure ci-dessus), avant d'être envoyé vers l'Inibox via la pompe-box.

Ce robot est alors indispensable pour ajouter le Tb de façon indépendante, essentiel à l'identification et à la quantification des polluants.

5.1.3 La pompe-box

La pompe-box établie les connexions entre l'Inibox, le robot et l'environnement naturel. Il va d'abord pomper l'eau directement du cours d'eau à analyser pour en prélever un échantillon, puis le faire circuler jusqu'à l'Opentron où l'ajout de Tb est effectué, pour enfin l'envoyer à l'Inibox pour l'analyse finale, avant d'être acheminé dans une poubelle. Ce dernier dispositif a été conçu par les 2 mathématiciens, avec l'aide de Benoit Hautefeuille.

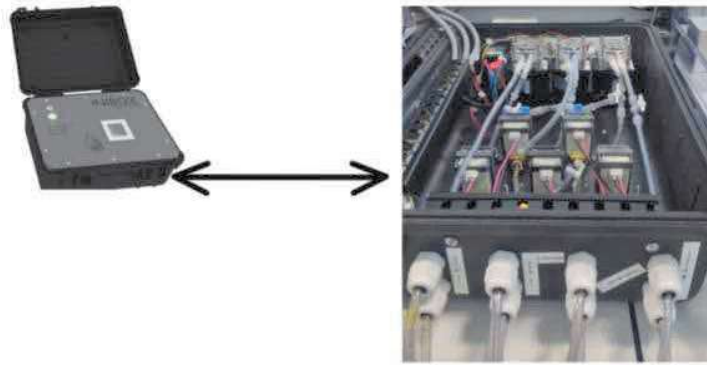


FIGURE 5.5 – Photographie de la pompe-box - *Image réalisée par Hiba*

5.2 Etat d'avancement

Une première visite de terrain a été concrétisée avec toute l'équipe sur le site *Django Rheinhardt* à Chassieu, afin d'établir un premier contact et de trouver un emplacement adéquat au dispositif de terrain. Ce site est idéal puisqu'il propose deux cabanons déjà exploités pour un suivi de l'eau en continu en effectuant, à intervalle de temps régulier, des prélèvements de pH ou de conductivité.

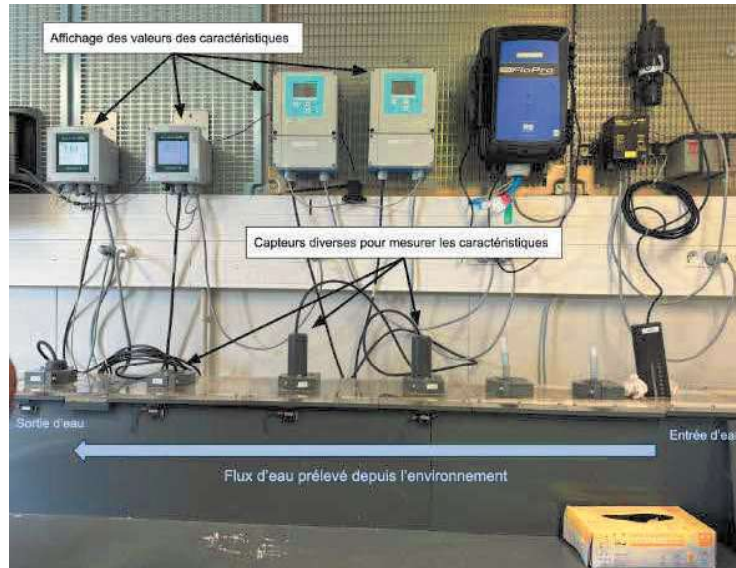


FIGURE 5.6 – Photographie de l'intérieur d'un cabanon

Il est donc plus facile, pour le projet VAI-TRF, de placer le dispositif dans le cabanon proposant le plus d'espace et de placer l'arrivée d'eau de la pompe-box

directement dans le bac où le flux d'eau circule.
Ces premiers tests devraient être réalisés au cours du mois de septembre 2024.

Conclusion et perspectives

Ce stage a permis de répondre efficacement à la problématique posée : comment développer et valider une méthode innovante de détection des polluants dans les eaux urbaines et industrielles, en utilisant la modélisation chimique avec Phreeq-C et la spectroscopie de fluorescence soutenue par l'intelligence artificielle, afin d'assurer une surveillance continue et précise des contaminants ?

La première partie du travail s'est concentrée sur la modélisation de la spéciation du Tb(III) avec le logiciel Phreeq-C. Cette modélisation a démontré que la NOM n'interfère pas avec la formation des complexes Tb-polluant. De plus, il a été confirmé que les polluants se complexent avec les ions lanthanides selon l'ordre croissant de leurs constantes de complexation, indépendamment des concentrations présentes. Ces résultats fournissent une base solide pour l'application des techniques de détection des polluants par fluorescence.

La seconde partie du stage, axée sur l'identification des polluants à l'aide de l'IA, a montré des résultats opérationnels pour certains polluants, bien que des confusions persistent pour d'autres, probablement en raison d'un manque de données. Cette difficulté devrait se résoudre avec l'intégration de nouvelles données, améliorant ainsi la précision de l'IA pour un plus grand nombre de polluants.

En conclusion, ce stage a permis de développer une méthode innovante pour la détection des polluants, répondant ainsi à la problématique initiale. Les perspectives post-stage incluent la quantification des polluants et les premiers essais sur le terrain, qui seront essentiels pour valider les méthodes développées dans des conditions réelles. Ces étapes renforceront la fiabilité et l'impact du projet, ouvrant la voie à une surveillance continue et précise des contaminants dans les eaux urbaines.

Bibliographie

- [1] Lachimie.fr. Spectre de fluorescence et de phosphorescence, 2024. Consulté le 28 juillet 2024.
- [2] Aurélie Le Ru. Bases de la fluorescence, n.d. Consulté le 30 juillet 2024.
- [3] Thi Thanh Huong Lai. *Quantification et marquage des produits chimiques dans l'industrie pétrolière par fluorescence résolue en temps des ions lanthanides*. PhD thesis, Université de Lyon, 2019. Thèse de doctorat en Physique Chimie.
- [4] Eaufrance. Les substances polluantes des milieux aquatiques, 2023. Consulté le 27 juillet 2024.
- [5] Commission canadienne de conservation de l'environnement (CCME). Phosphore, n.d. Consulté le 27 juillet 2024.
- [6] Ministère de la Transition Écologique. Polluants organiques persistants (pop), 2023. Consulté le 27 juillet 2024.
- [7] April Marine. Pollution par hydrocarbures : comment prévenir les risques en bateau ?, n.d. Consulté le 27 juillet 2024.
- [8] Les Agences de l'eau. Micropolluants : qui sont-ils ?, 2019. Consulté le 27 juillet 2024.
- [9] Lenntech. Métaux lourds, 2024. Consulté le 27 juillet 2024.
- [10] Suez. Micro-organismes pathogènes, n.d. Consulté le 27 juillet 2024.
- [11] World Health Organization. Drinking-water, 2023. Consulté le 28 juillet 2024.
- [12] Eaufrance. Les impacts de la pollution de l'eau, 2023. Consulté le 28 juillet 2024.
- [13] Office Français de la Biodiversité. La gestion de la ressource en eau, 2024. Consulté le 30 juillet 2024.
- [14] LibreTexts Chemistry. Fluorescence and phosphorescence, n.d. Consulté le 28 juillet 2024.

- [15] Alexander P. Demchenko. Fluorescence detection in sensor technologies. In *Introduction to Fluorescence Sensing*, chapter 3, pages 55–110. Springer, Cham, 2020.
- [16] Estelle Godart. *Synthèses de cryptophanes hydrosolubles adaptés à l’encapsulation du xénon, et d’hémicryptophanes fonctionnalisés pour la complexation de lanthanides (III), en vue d’applications en imagerie médicale*. PhD thesis, Université de Lyon, 2017. Téléchargé depuis HAL, Consulté le 28 juillet 2024.
- [17] David P. Millar. Time-resolved fluorescence spectroscopy. *ScienceDirect*, 6(5) :637–642, 1996.
- [18] M. Esmaelpourfarkhani, M. Ramezani, M. Alibolandi, K. Abnous, and M. Taghdisi. Time-resolved fluorescence dna-based sensors for reducing background fluorescence of environment. *Journal of Fluorescence*, 33(11) :2145–2160, 2023. Consulté le 28 juillet 2024.
- [19] İlhami Gulcin and Saleh H. Alwasel. Metal ions, metal chelators and metal chelating assay as antioxidant method. *Processes*, 10(1) :132, 2022.
- [20] Jingli Yuan and Guilan Wang. Lanthanide complex-based fluorescence label for time-resolved fluorescence bioassay. *Journal of Fluorescence*, 15 :559–568, 2005.
- [21] Arthur Marais. *Détection de traces d’éléments lanthanides par fluorescence en temps résolu : application industrielle au marquage anti contrefaçon et à l’analyse chimique*. PhD thesis, Université de Lyon, 2018. Téléchargé depuis HAL, Consulté le 29 juillet 2024.
- [22] Techniques de l’Ingénieur. Chélation : définition et propriétés, n.d. Consulté le 29 juillet 2024.
- [23] International Humic Substances Society. Elemental compositions and stable isotopic ratios of ihss samples, 2024. Consulté le 29 juillet 2024.
- [24] International Humic Substances Society. Acidic functional groups of ihss samples, 2024. Consulté le 29 juillet 2024.
- [25] Charlotte Catrouillet. *Les interactions entre l’arsenic, le fer et la matière organique en milieu anoxique*. PhD thesis, Université de Rennes 1, 2015. Consulté le 29 juillet 2024.
- [26] Rémi Marsac, Mélanie Davranche, Gérard Gruau, Martine Bouhnik-Le Coz, and Aline Dia. An improved description of the interactions between rare earth elements and humic acids by modeling : Phreeqc-model vi coupling. *Elsevier*, 75(19) :5625–5637, 2011.
- [27] E. Tipping. Humic ion-binding model vi : An improved description of the interactions of protons and metal ions with humic substances. *Aquatic geochemistry*, 4 :3–47, 1998.
- [28] David E. Wolf. Chapter 4 - fundamentals of fluorescence and fluorescence microscopy. In *Methods in Cell Biology*, volume 114, pages 69–97. Elsevier, 2013.

- [29] Arunkumar Gowri, Annasamy Kathiravan. A new approach for detection of environmental pollutants based on lanthanide ions. In Pooja Kumar Singh Patil, editor, *Sensors in water pollutants monitoring : Role of material*, pages 147–160. Springer, Singapore, 2019.
- [30] G. Obiefuna and D. Orazulike. Chemical speciation of some metal ions in groundwaters of yola area using geochemical model. *Journal of Applied Sciences and Environmental Management*, 14(2), 2010.
- [31] B. Lotter, S. Konde, J. Nguyen, M. Grau, M. Koch, and P. Lenz. Identifying plastics with photoluminescence spectroscopy and machine learning. *Scientific reports*, 12(18840), 2022.
- [32] X. Jia, D. O’Connor, Z. SHi, and D. Hou. Virs based detection in combination with machine learning for mapping soil pollution. *Environmental Pollution*, 268, 2022.
- [33] A. Michel, A. Morrison, V. Preston, C. Marx, B. Colson, and H. White. Rapid identification of marine plastic debris via spectroscopic techniques and machine learning classifiers. *Environmental Science and Technology*, 54(17) :10630–10637, 2020.
- [34] David L. Parkhurst and C.A.J. Appelo. User’s guide to phreeqc (version 2)—a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. Water-Resources Investigations Report 99-4259, U.S. Geological Survey, Denver, Colorado, 1999.
- [35] Francis J. C. Rossotti. *The Determination of Stability Constants and Other Equilibrium Constants in Solution*. McGraw-Hill, New York, 1961.
- [36] Arthur E. Martell and R. J. Motekaitis. *The Determination and Use of Stability Constants*. VCH Publishers, New York, 1988.
- [37] M. T. Beck. *Chemistry of Complex Equilibria*. Horwood, Chichester, 1990.
- [38] Jorge C. Masini, Gilberto Abate, Elizabete C. Lima, Lucas C. Hahn, Mary S. Nakamura, Jaim Lichtig, and Hélio R. Nagatomy. Comparison of methodologies for determination of carboxylic and phenolic groups in humic acids. *Analytica Chimica Acta*, 364 :223–233, 1998.
- [39] Keshia M. Kuhn, Elisabeth Neubauer, Thilo Hofmann, Frank von der Kammer, George R. Aiken, and Patricia A. Maurice. Concentrations and distributions of metals associated with dissolved organic matter from the suwannee river (ga, usa). *Environmental Engineering Science*, 32(1), 2015.
- [40] Robert M. Smith and Arthur E. Martell. Critical stability constants, enthalpies and entropies for the formation of metal complexes of aminopolycarboxylic acids and carboxylic acids. *Science of The Total Environment*, 64 :125–147, 1987.
- [41] Robert M. Smith and Arthur E. Martell. Critical stability constants, enthalpies and entropies for the formation of metal complexes of aminopolycarboxylic acids and carboxylic acids. *Science of The Total Environment*, 3 :180–186, 1987.

- [42] S. Mounier, N. Patel, L. Quilici, J.-Y. Benaim, and C. Benamou. Fluorescence 3d de la matière organique dissoute du fleuve amazone : Three-dimensional fluorescence of the dissolved organic carbon in the amazon river. *Water research*, 33(6) :1523–1533, 1999.
- [43] Zeljko D. Vujović. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6) :3, 2021.

Annexes



A

Tableau des constantes de stabilité des polluants

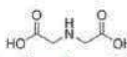
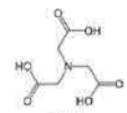
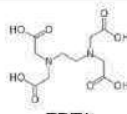
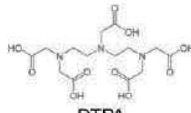
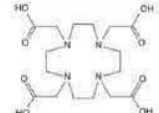
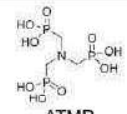
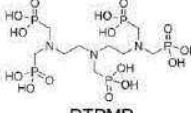


Compound	CAS	MM [g/mol]	Stability Constant (Tb ³⁺ , I=0.1M)
 Iminodiacetic acid (IDA)	142-73-4	133.10	$\log\beta=7.00, \log\beta_2=10.42$
 NTA	139-13-9 (free acid) 5064-31-3 (trisodium)	191.14	$\log\beta=11.25, \log\beta_2=20.21$
 EDTA	60-00-4 (free acid) 6381-92-6 (dihydrate/ diNa salt)	292.244 372.24	$\log\beta=17.90$
 DTPA	67-43-6	393.35	$\log\beta=22.81$
 DOTA	60239-18-1	404.42	$\log\beta=24.2$
 ATMP	6419-19-8	299.048	$\text{Log}\beta=18.88$
 DTPMP	15827-60-8	573.20	$\text{Log}\beta=30.64$
 Triethylenetetramine	112-24-3	146.23	$\log\beta=14.09, \log\beta_2=17.35$
 PFOS	1763-23-1	500.13	$\text{Log}\beta=5.00, \text{Log}\beta_2=10.89$

Tableau des constantes de stabilité de la NOM

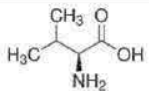
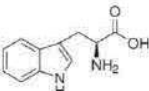
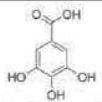
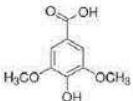
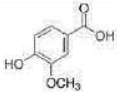
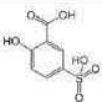
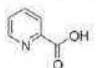
 <p>Valine</p>	72-18-4	117.15	$\log\beta=3.16, \log\beta_2=6.88$
 <p>Tryptophan</p>	73-22-3	204.23	$\log\beta=5.10, \log\beta_2=8.16$
 <p>Gallic acid</p>	149-91-7 5995-86-8 (*H ₂ O)	170.12	$\log\beta=7.49, \log\beta_2=13.67$
 <p>Syringic Acid</p>	530-57-4	198.17	$\log\beta=9.19, \log\beta_2=12.84$
 <p>Vanillic acid</p>	121-34-6	168.15	$\log\beta=9.80, \log\beta_2=13.25$
 <p>5-sulfosalicylic acid</p>	97-05-2	218.185	$\log\beta=5.99, \log\beta_2=8.88$
 <p>Picolinic Acid</p>	98-98-6	123.11	$\log\beta=7.79, \log\beta_2=11.02$



Tableau des concentrations des ions présents dans l'IHSS

Composition	µg/mg de C	ppm en IHSS(5ppm)	ppb en IHSS(5ppm)	Masse molaire(g/mol)	mmol/L en IHSS(5ppm)
Ca	2,488	6,31E-03	6,30708	12,01	5,25E-04
Mg	0,3064	7,77E-04	0,77672	24,31	3,20E-05
Na	27,6	7,00E-02	69,96600	22,99	3,04E-03
K	36,9	9,35E-02	93,54150	39,10	2,39E-03
Si	11,8	2,99E-02	29,91300	28,09	1,08E-03
Cl	4,85	1,23E-02	12,29475	35,45	3,47E-04
PO4	0,087	2,21E-04	0,22055	94,97	2,32E-06
SO4	14,4	3,65E-02	36,50400	96,06	3,80E-04
Fe	2,303	5,84E-03	5,83811	55,85	1,05E-04
Al	1,226	3,11E-03	3,10791	26,98	1,15E-04
Cu	0,524	1,33E-03	1,32834	63,54	2,09E-05
Mn	0,01	2,54E-05	0,02535	54,94	4,61E-07
Ba	0,00584	1,48E-05	0,01480	137,34	1,08E-07
Zn	0,0089	2,26E-05	0,02256	65,37	3,45E-07
Pb	5,95E-03	1,51E-05	0,01508	207,19	7,28E-08
V	1,44E-02	3,65E-05	0,03650	50,94	7,17E-07
Ni	8,74E-03	2,22E-05	0,02216	58,71	3,77E-07
As	4,87E-03	1,23E-05	0,01235	74,92	1,65E-07
Cr	3,84E-02	9,73E-05	0,09734	51,99	1,87E-06
Y	1,66E-04	4,21E-07	0,00042	88,9	4,73E-09
Cd	Faible				
Sc	Faible				
Pd	Faible				
U	Faible				

Faible = < 0,1 ng/mg de C



Exemple de code pour le robot Opentron OT-2

```
from opentrons import protocol_api
import numpy as np
import pandas as pd

metadata = {
    'apiLevel': '2.12',
    'protocolName': 'standard200TPA',
    'description': '''On considère 4 solutions mères : l'eau , solution inconnue
, solution standard , solution révélatrice .
La méthode consiste à distribuer le même volume de solution
polluée à concentration inconnue dans plusieurs échantillons,
ensuite, un volume fixe de solution révélatrice (contenant du lanthanide)
est ajouté à chaque échantillon, ainsi qu'une solution standard du même polluant, à une concentration co
'auteurs': 'Moustaphâ & Hiba'
}

def run(protocol: protocol_api.ProtocolContext):

tips = protocol.load_labware('opentrons_96_tiprack_300ul', 8) # place 8
tips1 = protocol.load_labware('opentrons_96_tiprack_1000ul', 7) # place 7
reservoir1 = protocol.load_labware('opentrons_2_reservoir_100000ul', 4) # place 4
```

```

if name=='AA007':
    if len(distribuate_1000) != 0 :
        p1000.distribute(list(distribuate_1000), reservoir2['A2'],bottom(z=15),
        [plate.wells_by_name()[well_name].bottom(z=55) for well_name
        in W_1000],disposal_volume = 0)

    if len(distribuate_300) != 0 :
        p300.distribute(list(distribuate_300), reservoir2['A2'],bottom(z=15),
        [plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
        in W_300],disposal_volume = 0)

if name=='AA1':
    if len(distribuate_1000) != 0 :
        p1000.distribute(list(distribuate_1000), reservoir3['A1'],bottom(z=15),
        [plate.wells_by_name()[well_name].bottom(z=55) for well_name
        in W_1000],disposal_volume = 0)

    if len(distribuate_300) != 0 :
        p300.distribute(list(distribuate_300), reservoir3['A1'],bottom(z=15),
        [plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
        in W_300],disposal_volume = 0)

volume_AA005 = [0]*3+[8000]*3+[0]*6
volume_AA007 = [0]*6+[8000]*3+[0]*3
volume_AA1 = [0]*9+[10000]*3
volume_Tb = [5000]*3+[0]*9
volume_Tb1 = [0]*3+[3000]*9

# visualisation des volumes préétablies
matrice_volume = np.transpose(np.array([volume_EP, volume_AA001, volume_AA005, volume_AA007, volume_AA1,
volume_Tb, volume_Tb1]))
colonnes = ['EP', 'AA0001', 'AA0005', 'AA007', 'AA1', 'Tb',
'Tb1']
volume = pd.DataFrame(matrice_volume, columns=colonnes)
print(volume)

```

```

for name in volume.columns :
liste=list(volume[name])

distribuate_1000=[] # stockera les volumes a distribuer selon la pipette p1000
distribuate_300=[] # stockera les volumes a distribuer selon la pipette p300
w_300=[] # stockera le nom des wells (A1, ..., D4) ou déposer les volumes aspire par p1000
w_1000=[] # stockera le nom des wells (A1, ..., D4) ou déposer les volumes aspire par p1000

for i in range (0,len(liste)) : #
on parcourt un a un les volumes de la liste

V=liste[i] # pour un tel volume V

S_1000=0 # comptera le nombre de transfert dans un meme well pour distribuer V avec p1000
S_300=0 # comptera le nombre de transfert dans un meme well pour distribuer V avec p300
# on reecrit V = q*1000 + r avec r < 1000
q = V // 1000
r = V % 1000
# on reecrit V=q1*300+r1 avec r1 < 300
q1 = V // 300
r1 = V % 300
# on reecrit r= q2*300 + r2 avec r2 <300
q2 = r // 300
r2= r % 300

if V>=1000 : # si le volume est plus grand ou egal a 1000
distribuate_1000 += [1000] * q # on distribue dans un premier temps q fois un volume
de 1000 avec p1000
S_1000=q

if r!=0 : # si il reste du volume a distribuer, r
if r%100==0 or r%
100 == 50: # si celui ci est un multiple de 100 ou qui termine par 50
distribuate_1000.append(r) # on distribue r avec p1000
S_1000=S_1000+1
else : # sinon on le distribue avec p300
if r<300 : # si on peut distribuer r en une seule fois
distribuate_300.append(r) # on le distribue en une fois
S_300=S_300+1
else : # sinon on le distribue en plusieurs fois
distribuate_300 += [300] * q2 # on distribue dans un premier temps q2 fois un volume
de 300
S_300=q2
distribuate_300.append(r2) # puis on distribue ce qui l'en reste, r2 en une fois

```

```

S_300=S_300+1
else : # si le volume initial V est inferieure a 1000.
if (V%100==0) or (V%
100 == 50) : # si celui-ci est un multiple de 100 ou qui termine par 50
distribuate_1000.append(V) # on distribue V avec p1000
S_1000=S_1000+i
else : # sinon on le distribue avec une p300
if r<300 : # si on peut distribuer r en une seule fois
distribuate_300.append(r) # on le distribue en une fois
S_300=S_300+i
else : # sinon on le distribue en plusieurs fois
distribuate_300 += [300] * q1 # on distribue dans un premier temps q2 fois un volume
de 300
S_300=q1
distribuate_300.append(r1) # puis on distribue ce qui l en reste, r2 en une fois
S_300=S_300+1
# permet la construction de W_1000, W_300
S_300=int(S_300)
S_1000=int(S_1000)
if i<7 : # si V etait un volume que lon souhaitait distribuer dans la premiere ligne du plate
for k in range (0,S_1000) :
W_1000.append('A{}'.format(i+1))
# ajoute le nom de well (A) S_1000 fois, car on distribue dans celui-ci "une partie" de V selon p1000 et
for k in range (0,S_300) :
W_300.append('A{}'.format(i+1))
# ajoute le nom de well (A) S_300 fois, car on distribue dans celui-ci "une partie" de V selon p300 en
elif i<14 : # si V etait un volume que lon souhaitait distribuer dans la seconde ligne du plate
for k in range (0,S_1000) :
W_1000.append('B{}'.format(i-6))
for k in range (0,S_300) :
W_300.append('B{}'.format(i-6))
elif i<21 : # si V etait un volume que l on souhaitait distribuer dans la troisieme ligne du plate
for k in range (0,S_1000) :
W_1000.append('C{}'.format(i-13))
# ajoute le nom de well S_1000 fois
for k in range (0,S_300) :
W_300.append('C{}'.format(i-13))
else : # si V etait un volume que lon souhaitait distribuer dans la derniere ligne du plate
for k in range (0,S_1000) :
W_1000.append('D{}'.format(i-20))
for k in range (0,S_300) :
W_300.append('D{}'.format(i-20))

```

```

# afin de verifier la repartition
print("la liste de volumes donnee:")
print(liste)
print('\n')
print("Selon p1000, les volumes a distribuer sont :")
print(distribuate_1000)
print("respectivement dans les plates : ")
print(W_1000)
print('\n')
print("Selon p300 les volumes a distribuer sont :")
print(distribuate_300)
print("respectivement dans les plates : ")
print(W_300)
print('\n')

# distribution

if name=='EP' :
    if len(distribuate_1000) != 0 :
        p1000.distribute(list(distribuate_1000), reservoir1['A1'].bottom(z=15),
            [plate.wells_by_name()[well_name].bottom(z=55) for well_name
            in W_1000],disposal_volume = 0)

    if len(distribuate_300) != 0 :
        p300.distribute(list(distribuate_300), reservoir1['A1'].bottom(z=15),
            [plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
            in W_300],disposal_volume = 0)
        if name=='AA0001':
            if len(distribuate_1000) != 0 :
                p1000.distribute(list(distribuate_1000), reservoir1['A2'].bottom(z=15),
                    [plate.wells_by_name()[well_name].bottom(z=55) for well_name
                    in W_1000],disposal_volume = 0)

            if len(distribuate_300) != 0 :
                p300.distribute(list(distribuate_300), reservoir1['A2'].bottom(z=15),
                    [plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
                    in W_300],disposal_volume = 0)

```

```

if name=='AA0005':
if len(distribuate_1000) != 0 :
p1000.distribute(list(distribuate_1000), reservoir2['A1'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=55) for well_name
in W_1000],disposal_volume = 0)

if len(distribuate_300) != 0 :
p300.distribute(list(distribuate_300), reservoir2['A1'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
in W_300],disposal_volume = 0)

if name=='AA007':
if len(distribuate_1000) != 0 :
p1000.distribute(list(distribuate_1000), reservoir2['A2'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=55) for well_name
in W_1000],disposal_volume = 0)

if len(distribuate_300) != 0 :
p300.distribute(list(distribuate_300), reservoir2['A2'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
in W_300],disposal_volume = 0)

if name=='AA1':
if len(distribuate_1000) != 0 :
p1000.distribute(list(distribuate_1000), reservoir3['A1'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=55) for well_name
in W_1000],disposal_volume = 0)

if len(distribuate_300) != 0 :
p300.distribute(list(distribuate_300), reservoir3['A1'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
in W_300],disposal_volume = 0)

if name=='Tb' :
if len(distribuate_1000) != 0 :
p1000.distribute(list(distribuate_1000), reservoir3['A2'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=55) for well_name
in W_1000],disposal_volume = 0)

if len(distribuate_300) != 0 :
p300.distribute(list(distribuate_300), reservoir3['A2'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
in W_300],disposal_volume = 0)

```

```
if name=='Tb1' :
if len(distribuate_1000) != 0 :
p1000.distribute(list(distribuate_1000), reservoir4['A1'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=55) for well_name
in W_1000],disposal_volume = 0)

if len(distribuate_300) != 0 :
p300.distribute(list(distribuate_300), reservoir4['A1'].bottom(z=15),
[plate.wells_by_name()[well_name].bottom(z=51.75) for well_name
in W_300],disposal_volume = 0)
```



E

Fichier input pour la modélisation de la martice réelle

```

SOLUTION 1
  pH 7 charge
  temp 25
  pe 4
  redox pe
  units mmol/kgw
  Na 0.00304
  K 0.00239
  Tb ctb
  Cl(-1) ccl
  Dtpa 1
  Nta 1
  Dota 1
  Dtpmp 1
  S 0.00038
  -water 1 # kg

SOLUTION_MASTER_SPECIES
  Dtpa HDtpa 0.0 Dtpa 393.35
  Nta HNta 0.0 Nta 191.14
  Dota HDota 0.0 Dota 404.42
  Dtpmp HDTpmp 0.0 Dtpmp 573.2

SURFACE 1
  -equilibrate 1
  -sites_units density
  Humin_h 0.00247 600.0 0.33
  Humin_c 0.01121 600.0 0.33

SURFACE_MASTER_SPECIES
  Humin_h Humin_hOH
  Humin_c Humin_cCOOH

EQUILIBRIUM_PHASES 1
  Tb(OH)3 0.0
  CO2(g) -3.5 10

SURFACE_SPECIES
  Humin_hOH = Humin_hOH
  log_k 0.0
  Humin_cCOOH = Humin_cCOOH
  log_k 0.0

  Humin_hOH = Humin_hO- + H+
  log_k -7.5 # Constante de dissociation du groupe phénolique
  Humin_cCOOH = Humin_cCOO- + H+
  log_k -4.5 # Constante de dissociation du groupe carboxylique

  Humin_cCOOH + Tb+++ = Humin_cCOOTb++ + H+
  log_k 7.26

  Humin_hOH + Tb+++ = Humin_hOTb++ + H+
  log_k 8

SOLUTION_SPECIES
  HDtpa = HDtpa
  log_k 0
  -delta_H 0

  HDtpa = Dtpa- + H+
  log_k 7
  -delta_H 0

  1.000 Tb+++ + 1.000 HDtpa = TbDtpa++ + 1.000 H+
  log_k 22.81

```

```

HNta = HNta
      log_k  0
      -delta_H  0

HNta = Nta- + H+
      log_k  7
      -delta_H  0

1.000 Tb+++ + 1.000 HNta = TbNta++ + 1.000 H+
      log_k  11.25

1.000 Tb+++ + 2.000 HNta = Tb(Nta)2+ + 2.000 H+
      log_k  20.21

HDota = HDota
      log_k  0
      -delta_H  0

HDota = Dota- + H+
      log_k  7
      -delta_H  0

1.000 Tb+++ + 1.000 HDota = TbDota++ + 1.000 H+
      log_k  24.2

HDtmp = HDtmp
      log_k  0
      -delta_H  0

HDtmp = Dtmp- + H+
      log_k  7
      -delta_H  0

1.000 Tb+++ + 1.000 HDtmp = TbDtmp++ + 1.000 H+
      log_k  30.64

```



Code complet de l'automatisation de Phreeq-C

```
### Axelle BARAT, 28/05/2024
### Ce script permet d'automatiser et d'exécuter des simulations chimiques sur le logiciel PHREEQC
### Il modifie les fichiers d'entrée de Phreeqc selon une gamme de concentrations pour un élément donné
### Il extrait les données pertinentes des nouveaux fichiers d'entrée pour les mettre dans des colonnes
###comme des concentrations
### Il lance les simulations à partir du logiciel et récupère les résultats pertinents dans les fichiers
###de sortie tels que le pH ou les concentrations souhaitées.
### Il renvoie un fichier .csv contenant les colonnes extraites dans les fichiers d'entrée et de sortie
### Possibilité d'inclure des calculs de pourcentage entre les différentes colonnes

import subprocess
import os
import numpy as np
import pandas as pd

# Paramètres à modifier
PARAMS = {
    "elt_start": 0.0, #Début de la gamme de concentrations ciblée
    "elt_end": 1.0, #Fin de la gamme de concentrations ciblée
    "elt_step": 0.0005, #Pas entre deux concentrations
    "cl_electroneutrality_adjustment": 0.00506, #Si besoin d'ajuster l'électroneutralité
    #Chemin vers la base de données
    "database_path": '/Users/axellebarat/Desktop/Phreeqc/database/llnl.dat',
    #Chemin vers l'exécutable Phreeq-C
    "phreeqc_executable_path": '/Users/axellebarat/Desktop/Phreeqc/bin/phreeqc',
    #Chemin vers le fichier d'entrée
    "input_file_path": '/Users/axellebarat/Desktop/Phreeqc/bin/IHSSTbEN.txt',
    #Chemin vers le fichier de sortie
    "output_file_base": '/Users/axellebarat/Desktop/Phreeqc/output/IHSSTbEN',
    "num_files": 2000, #Nombre de fichiers
    "output_directory": '/Users/axellebarat/Desktop/Phreeqc/output/', #Chemin vers le dossier de sortie
    #Chemin du 1er .csv avec les colonnes des données d'entrée
    "csv_results_path": '/Users/axellebarat/Desktop/Phreeqc/output/resultsIHSSTbEN.csv',
    #Chemin du 2nd .csv avec les colonnes des données de sortie
    "csv_ph_output_path": '/Users/axellebarat/Desktop/Phreeqc/output/phIHSSTbEN.csv',
    #Chemin du .csv final
    "csv_final_path": '/Users/axellebarat/Desktop/Phreeqc/output/validationprgm.csv',

    "input_params": ["Tb", "Cl(-)"], #Nom des valeurs à prélever dans les fichiers d'entrée
    "output_params": [
        {"key": "pH", "search_str": " pH = "},
        {"key": "Erreur", "search_str": " Percent error, 100*{(Cat-|An|)}/{Cat+|An|} = "},
        {"key": "Total Tb", "search_str": "Tb(3) "},
        {"key": "Tb3+", "search_str": "Tb+3 "}],
    #Nom des valeurs à prélever dans les fichiers de sortie
    "percentages_to_calculate": [
        {"numerator": "Tb3+", "denominator": "Total Tb", "result_column": "Tb3+ (%)"},
        # Ajouter d'autres calculs de pourcentage si nécessaire
    ]
}
```

```

# Fonction pour calculer les gammes de concentrations
def calculate_concentrations(params):
    A = np.arange(params["elt_start"], params["elt_end"], params["elt_step"])
    B = [3 * Tb_conc + params["cl_electroneutrality_adjustment"] for Tb_conc in A]
    return A, B

# Fonction pour extraire les données d'entrée
def extract_input_data(input_content, params_to_extract):
    lines = input_content.split('\n')
    data = {}
    for line in lines:
        for param in params_to_extract:
            if f' {param} ' in line:
                data[param] = str(line.split()[1])
    return data

# Fonction pour exécuter une simulation PHREEQC
def run_phreeqc_simulation(params, tb_concentrations, cl_concentrations):
    results = []

    with open(params["input_file_path"], 'r') as f:
        input_content = f.read()

    for i in range(len(tb_concentrations)):
        modified_content = input_content.replace('ctb', str(tb_concentrations[i])).replace('ccl', str(cl_concentrations[i]))
        modified_content = f'DATABASE {params["database_path"]}\n' + modified_content

        # Définir le chemin complet pour le fichier temporaire
        temp_input_file_path = os.path.join(params["output_directory"], f'temporaire{i}.txt')

        with open(temp_input_file_path, 'w') as f:
            f.write(modified_content)

        output_file_path = f'{params["output_file_base"]}{i}.txt'
        command = f'{params["phreeqc_executable_path"]} {temp_input_file_path} {output_file_path} {params["database_path"]}'

        process = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        output = process.communicate()

        input_data = extract_input_data(modified_content, params["input_params"])
        results.append(input_data)

    df = pd.DataFrame(results)
    return df

# Fonction pour extraire les données de sortie
def extract_output_data(output_content, params_to_extract):
    lines = output_content.split('\n')
    data = {}
    for line in lines:
        for param in params_to_extract:
            if param["search_str"] in line and param["key"] not in data:
                value = line.split('=')[1].split()[0].strip() if '=' in line else line.split()[1].strip()
                data[param["key"]] = str(value)
    return data

# Fonction pour convertir et extraire les fichiers de sortie
def convert_and_extract_outputs(params, num_files):
    results = []
    for i in range(num_files):
        out = os.path.join(params["output_directory"], f'IHSSTbEN{i}.txt')
        with open(out, 'r', encoding='iso-8859-1') as file:
            content = file.read()
        with open(out, 'w', encoding='utf-8') as file:
            file.write(content)

        print(f"Le fichier {out} a été converti en UTF-8.")

```

```

output_data = extract_output_data(content, params["output_params"])

# Calculer les pourcentages dynamiquement
for percentage_calc in params["percentages_to_calculate"]:
    numerator = float(output_data[percentage_calc["numerator"]])
    denominator = float(output_data[percentage_calc["denominator"]])
    if denominator != 0:
        output_data[percentage_calc["result_column"]] = (numerator / denominator) * 100
    else:
        output_data[percentage_calc["result_column"]] = 0

results.append(output_data)

df_out = pd.DataFrame(results)
return df_out

# Fonction principale pour exécuter l'ensemble du pipeline
def main(params):
    tb_concentrations, cl_concentrations = calculate_concentrations(params)
    df_results = run_phreeqc_simulation(params, tb_concentrations, cl_concentrations)
    df_results.to_csv(params["csv_results_path"], index=False)

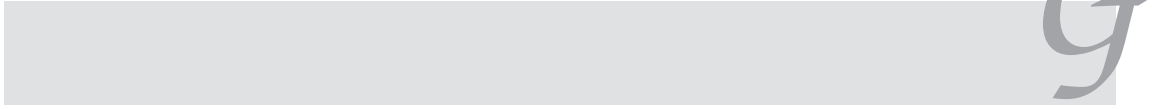
    df_out = convert_and_extract_outputs(params, params["num_files"])
    df_out.to_csv(params["csv_ph_output_path"], index=False)

    df_results = pd.read_csv(params["csv_results_path"])
    df_out = pd.read_csv(params["csv_ph_output_path"])

    df_conc = pd.concat([df_results, df_out], axis=1)
    df_conc.to_csv(params["csv_final_path"], index=False)
    print("Pipeline complété avec succès.")

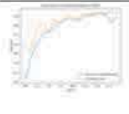
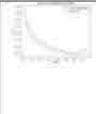
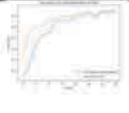
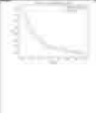
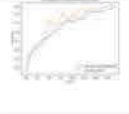
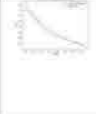


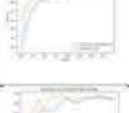

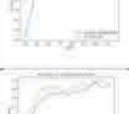

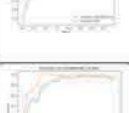

# Exécuter le script principal
main(PARAMS)

```



Résultats des 18 méthodes de traitement d'images par l'IA

Méthode nom	Accuracy	Loss	Matrice de confusion	Graph évaluer overfitting	Graph évaluer overfitting
A1	0.9122806787490845	0.3413132429122925	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 2 0 1 1 2 4 0 0 0 0 12 </pre>		
A2	1.0000	0.0655	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 0 0 0 6 0 4 0 0 0 0 12 </pre>		
B2	0.8771929740905762	0.2940249741077423	<pre> Predicted 0 1 2 4 True 0 17 0 0 1 1 0 9 0 0 2 0 0 12 0 3 4 0 2 0 4 0 0 0 12 </pre>		
B1	0.8310	0.4355	<pre> Predicted 0 1 2 3 4 True 0 16 0 0 1 1 1 0 9 0 0 0 2 0 1 11 0 0 3 2 0 1 1 2 4 2 0 0 0 10 </pre>		
C1	0.9649122953414917	0.15200679004192352	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 1 0 1 4 0 4 0 0 0 0 12 </pre>		
C2	0.9649122953414917	0.059922829270362854	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 10 2 0 3 0 0 0 6 0 4 0 0 0 0 12 </pre>		
D1	0.9649122953414917	0.17193230986595154	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 1 0 1 4 0 4 0 0 0 0 12 </pre>		
D2	0.9473684430122375	0.14365249872207642	<pre> Predicted 0 1 2 3 4 True 0 18 0 0 0 0 1 0 9 0 0 0 2 0 0 12 0 0 3 0 0 3 3 0 4 0 0 0 0 12 </pre>		
E1	0.9473684430122375	0.1832851618528366	<pre> Predicted 0 1 2 3 4 True 0 17 0 0 1 0 1 0 9 0 0 0 2 0 0 12 0 0 3 0 0 2 4 0 4 0 0 0 0 12 </pre>		
E2	0.9649122953414917	0.12845642864704132	<pre> Predicted 0 1 2 3 4 True 0 17 0 0 1 0 1 0 9 0 0 0 2 0 0 11 1 0 3 0 0 0 5 0 4 0 0 0 0 12 </pre>		

F1	0.92982459068 29834	0.2488110 214471817	Predicted: 0 1 2 3 4 True: 0 15 0 0 3 0 1 0 0 0 0 0 2 0 0 12 0 0 3 0 0 1 5 0 4 0 0 0 0 12 <figure size 800x600 with 0 Axes>		
F2	0.9558	0.1316	Predicted: 0 1 2 3 4 True: 0 18 0 0 0 0 1 0 0 0 0 0 2 0 0 12 0 0 3 0 0 2 4 0 4 0 0 0 0 12 <figure size 800x600 with 0 Axes>		
G1	0.91228067874 90845	0.3323173 522949219	Predicted: 0 1 2 3 4 True: 0 17 0 0 1 0 1 0 3 0 0 0 2 0 0 12 0 0 3 2 0 1 5 0 4 1 0 0 0 11 <figure size 800x600 with 0 Axes>		
G2	0.94736844301 22375	0.2158001 363277435 3	Predicted: 0 1 2 3 4 True: 0 17 0 0 0 1 1 0 0 1 0 0 2 0 0 12 0 0 3 0 0 1 5 0 4 0 0 0 0 11 <figure size 800x600 with 0 Axes>		
H1	0.92982459068 29834	0.1081367 433071136 5	Predicted: 0 1 2 3 4 True: 0 28 0 0 0 0 1 0 0 0 0 0 2 0 0 11 1 0 3 1 0 2 3 0 4 0 0 0 0 12 <figure size 800x600 with 0 Axes>		
H2	0.96491229534 14917	0.1087578 386068344 1	Predicted: 0 1 2 3 4 True: 0 17 0 1 0 0 1 0 3 0 0 0 2 0 0 12 0 0 3 0 0 1 5 0 4 0 0 0 0 12 <figure size 800x600 with 0 Axes>		
I1	0.96491229534 14917	0.1185713 931918144 2	Predicted: 0 1 2 3 4 True: 0 18 0 0 0 0 1 0 3 0 0 0 2 0 0 12 0 0 3 1 0 1 4 0 4 0 0 0 0 12 <figure size 800x600 with 0 Axes>		
I2	1.0	0.0241837 538778781 9	Predicted: 0 1 2 3 4 True: 0 18 0 0 0 0 1 0 0 0 0 0 2 0 0 12 0 0 3 0 0 0 0 0 4 0 0 0 0 12 <figure size 800x600 with 0 Axes>	