Concours Commun INP (CCINP), France

Session 2016, filière PC

Modélisation de systèmes physiques ou chimiques

Durée : 4 heures

Ce que traite le sujet

- Parties I et II.1 : présentation et absorption physique du polluant
- Partie II.2 : absorption avec réaction chimique
- Partie III.1 : détermination du produit k a par les moindres carrés
- Parties III.2 et III.3 : aire interfaciale, coefficient de transfert et simulations

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années :
prepahelp.com/annales/ccinp/pc/2016/modelisation

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

CONCOURS COMMUNS POLYTECHNIQUES

EPREUVE SPECIFIQUE - FILIERE PC

MODELISATION DE SYSTEMES PHYSIQUES OU CHIMIQUES

Jeudi 5 mai : 8 h - 12 h

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

Les calculatrices sont autorisées

Détermination du coefficient de transfert d'un polluant dans une colonne d'absorption

I. Présentation du problème

Les procédés d'absorption sont souvent utilisés pour la dépollution des gaz. Ces procédés reposent sur l'absorption préférentielle du gaz polluant par un solvant et sont la plupart du temps mis en œuvre dans des colonnes d'absorption (figure 1). Lors du contact entre les deux phases (gazeuse et liquide), le polluant est transféré du gaz vers le solvant. On récupère un gaz purifié en sortie haute de colonne et le solvant chargé du polluant en pied de colonne.

Un exemple courant est l'absorption du dioxyde de carbone présent dans les gaz de combustion à l'aide d'un solvant aminé pour éviter de rejeter ce gaz à effet de serre directement dans l'atmosphère.

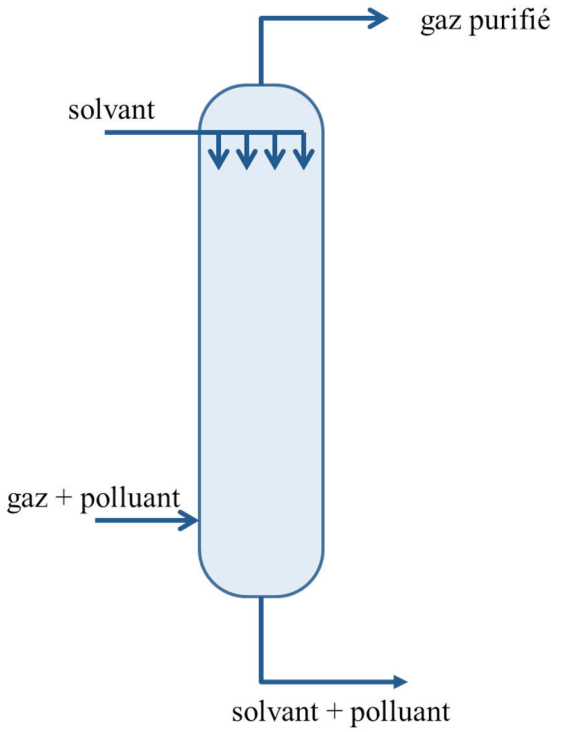


Figure 1 – Schéma d'une colonne d'absorption utilisée dans l'industrie pour la dépollution d'un gaz.

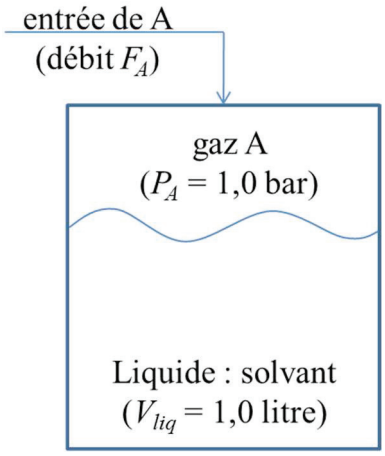
Pour dimensionner les colonnes d'absorption, on a besoin de connaître la conductance de transfert du polluant dans le solvant (notée k_L, unité : $m.s^{-1}$). Pour déterminer ce paramètre, on réalise une étude dans un réacteur de laboratoire dont les conditions de fonctionnement sont beaucoup plus simples et beaucoup mieux définies que dans une véritable colonne d'absorption. Le réacteur de laboratoire utilisé est un réacteur biphasique gaz-liquide.

Une des spécificités de ce réacteur est qu'il est alimenté par un débit variable de polluant qui est ajusté au cours du temps de manière à maintenir la pression de la phase gazeuse constante (figure 2, page 3). Au cours d'une expérience, on enregistre le débit molaire de polluant A qui entre dans le réacteur en fonction du temps. D'après la loi de Dalton, la pression totale est égale à la somme des pressions partielles des espèces dans la phase gazeuse. Comme A est seul dans la phase gazeuse, la pression totale P_{tot} est égale à la pression partielle de A, notée P_A. On suppose que le réacteur est isotherme.

La détermination directe de la conductance de transfert d'un polluant dans un solvant n'est pas aisée. On procède en deux étapes pour la déterminer :

- on détermine d'abord la valeur du produit $k_L \times a$, où a est l'aire interfaciale entre le liquide et le gaz par unité de volume de la phase liquide (unité de a : $m^2.m^{-3}$), grâce à une première expérience où le seul phénomène qui a lieu est l'absorption physique du polluant A dans le solvant ;

- on détermine ensuite les valeurs des deux paramètres k_L et a de manière indépendante grâce à une seconde expérience avec le même solvant mais qui contient cette fois-ci un réactif B qui réagit avec le gaz polluant A (absorption physique avec réaction chimique). On supposera que la réaction chimique en phase liquide s'écrit : $A + B \rightarrow produits$.



F_A : débit molaire de A qui entre dans le réacteur.

P_A : pression partielle de A dans la phase gazeuse.

V_{liq} : volume de la phase liquide (solvant).

Figure 2 – Schéma du réacteur de laboratoire utilisé pour la détermination de la conductance de transfert du polluant A.

On enregistre le débit molaire de A qui entre dans le réacteur en fonction du temps pour les deux expériences (sans réaction et avec réaction avec le réactif B). Les données se trouvent dans un fichier nommé « `data.txt` ». La première colonne correspond au temps (s). Les deuxième et troisième colonnes correspondent aux débits molaires de A enregistrés au cours des deux expériences (sans et avec réaction respectivement). Les colonnes sont séparées par des espaces (tableau 1, page 7).

II. Modélisation des phénomènes

Dans cette partie, on demande d'établir les modèles qui vont représenter l'évolution du débit molaire d'entrée F_A en fonction du temps. Les modèles sont obtenus en réalisant des bilans de matière.

II.1 Cas où le solvant ne contient pas le réactif B

Q1.1) On considère tout d'abord l'absorption du polluant A dans le solvant (cas où le solvant ne contient pas le réactif B). Le phénomène qui régit le transfert de A à l'interface entre les deux phases est la diffusion, phénomène localisé au niveau de la couche limite située à proximité de l'interface (zone délimitée par des pointillés sur la figure 3, page 4).

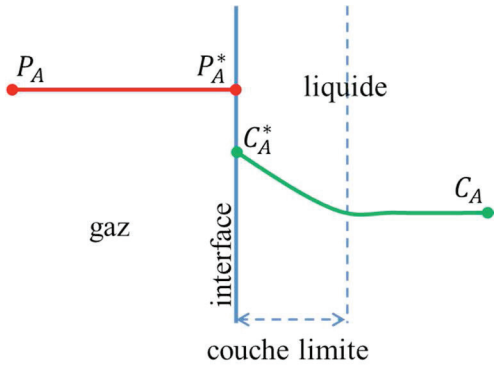


Figure 3 – Schéma représentant l'interface gaz – liquide.

La pression à l'interface P_A^* est égale à la pression partielle P_A dans la phase gazeuse car le polluant A est seul dans cette phase. La pression dans la phase gazeuse est maintenue constante ($P_A = P_A^* = 1{,}0$ bar). A l'interface, la loi de Henry permet de relier la pression P_A^* et la concentration C_A^*. D'après cette loi, la concentration de gaz dissous dans un liquide est proportionnelle à la pression partielle du gaz en contact avec le liquide à l'équilibre thermodynamique et à température constante.

Calculer la valeur de la concentration C_A^* en mol.m^{-3} à partir de la loi de Henry. On donne la constante de Henry pour A : $H = 26{,}0$ bar.L.mol^{-1}. On pourra s'appuyer sur l'analyse dimensionnelle pour écrire la loi de Henry.

Q1.2) Le débit molaire F_A^* qui traverse l'interface gaz-liquide vers le solvant s'écrit : $k_L \times a \times (C_A^* - C_A) \times V_{liq}$, où C_A^* est la concentration du polluant à l'interface, C_A est la concentration de A dans le solvant (figure 3) et V_{liq} le volume de la phase liquide.

Vérifier que l'expression $k_L \times a \times (C_A^* - C_A) \times V_{liq}$ est bien homogène à un débit molaire.

Q1.3) Sachant qu'à l'instant initial la concentration C_A de A dans la phase liquide est nulle, expliquer qualitativement comment C_A va évoluer au cours du temps. Préciser le signe du terme $k_L \times a \times (C_A^* - C_A) \times V_{liq}$.

Q1.4) On considère que la phase gazeuse se comporte comme un réacteur ouvert parfaitement agité en régime permanent (il n'y a donc pas d'accumulation de A dans cette phase) dans lequel il n'y a pas de réaction (le gaz ne fait que transiter dans cette partie du réacteur). On précise que les grandeurs intensives (température, pression, concentration) dans un réacteur ouvert parfaitement agité sont identiques en tout point.

Ecrire le bilan de matière sur le polluant A dans la phase gazeuse. Préciser le terme d'entrée et le terme de sortie intervenant dans le bilan de matière.

Q1.5) La phase liquide est modélisée par un réacteur semi-fermé parfaitement agité fonctionnant en régime transitoire. Ce type de réacteur peut être considéré comme un réacteur fermé parfaitement agité (contenant initialement le solvant dans le cas présent) possédant une entrée permettant l'ajout d'une espèce au cours du temps (le polluant A dans le cas présent).

Pour simplifier le problème, on suppose que le volume de la couche limite (zone à proximité de l'interface dans laquelle on observe un gradient de concentration du polluant dû à sa diffusion de l'interface vers la phase liquide, figure 3, page 4) est très faible et que la concentration de A dans la phase liquide est homogène. En d'autres termes, on considère que la concentration de A passe instantanément de C_A^*, à l'interface, à C_A dans la phase liquide comme indiqué sur le schéma de la figure 4.

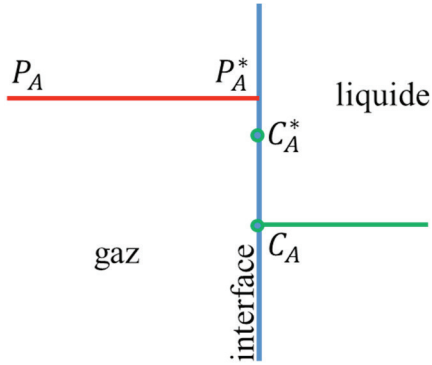


Figure 4 – Schéma simplifié de l'interface gaz–liquide.

Ecrire le bilan de matière sur le polluant A dans la phase liquide. Préciser le terme d'entrée et le terme d'accumulation.

Q1.6) Simplifier le bilan en considérant que le volume de la phase liquide reste constant au cours du temps.

Q1.7) Résoudre l'équation différentielle obtenue à la question précédente en considérant que la concentration initiale de A dans la phase liquide est nulle. On pourra effectuer un changement de variable.

Q1.8) Donner l'expression de la concentration de A en fonction du temps. Vérifier que C_A évolue bien de la manière prévue à la question **Q1.3)**, page 4.

Q1.9) Etablir l'expression du débit molaire F_A en fonction du temps. Vérifier l'homogénéité de la relation.

Q1.10) Préciser vers quelle valeur tend le débit molaire F_A pour des temps importants et indiquer vers quelle valeur tend la concentration C_A dans ce cas.

II.2 Cas où le réactif B est présent dans le solvant

Q2.1) On considère maintenant le cas où le réactif B est présent dans le solvant (à la concentration initiale C_{B0} de 1 000,0 mol.m^{-3}) et où il y a réaction entre A et B dans la phase liquide. Pour simplifier, on suppose que le réactif B ne passe pas dans la phase gazeuse (figure 5, page 6). On suppose également que la phase liquide est parfaitement agitée et que le volume reste constant. On précise que, comme dans le cas précédent, la concentration C_A de A dans la phase liquide est nulle à l'instant initial.

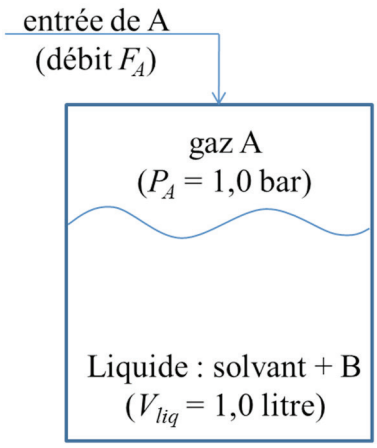


Figure 5 – Schéma du réacteur dans le cas de la deuxième expérience.

Le bilan de matière sur la phase gazeuse est-il modifié ? Justifier.

Q2.2) Des études préalables ont montré que la vitesse apparente de la réaction dans la phase liquide (notée r_{app}) est impactée par les phénomènes de diffusion de A dans la couche limite et qu'elle peut s'écrire sous la forme $r_{app} = a \times C_A^* \times \sqrt{D_A \times k \times C_B}$ pour une réaction d'ordre un par rapport à chacun des deux réactifs. D_A est la diffusivité de A dans la phase liquide ($1{,}83 \times 10^{-9}$ m^2.s^{-1}), k est la constante cinétique vraie de la réaction entre A et B (k = 0,10 m^3.mol^{-1}.s^{-1}) et C_B est la concentration de B dans la phase liquide.

Indiquer la dimension de r_{app}. Préciser son unité dans le système SI.

Q2.3) Ecrire le nouveau bilan de matière sur A dans la phase liquide en prenant en compte le nouveau terme correspondant à la réaction chimique (on fera particulièrement attention au signe).

Q2.4) Simplifier le bilan en considérant que le volume de la phase liquide reste constant.

Q2.5) Ecrire le bilan de matière sur B dans la phase liquide en considérant que la phase liquide est un réacteur fermé parfaitement agité.

Q2.6) Simplifier ce bilan en considérant que le volume de la phase liquide reste constant.

Q2.7) Donner le système de deux équations différentielles ordinaires qui régit l'évolution des concentrations de A et de B dans la phase liquide en fonction du temps.

III. Traitement numérique des données expérimentales

Dans cette partie, on propose de comparer les modèles établis dans la partie II aux résultats expérimentaux pour déterminer la valeur de l'aire interfaciale a et celle du coefficient de transfert k_L en utilisant des outils numériques.

Les portions de programme demandées au candidat peuvent être réalisées dans le langage Python ou dans le langage Scilab. Cependant, toutes les questions seront traitées dans le même langage. On veillera à apporter les commentaires suffisants à la compréhension du programme et utiliser des noms de variables explicites. Il est demandé de répondre précisément aux questions posées (par exemple, on écrira une fonction uniquement lorsque cela est explicitement demandé). Des annexes sont disponibles à la fin de l'énoncé.

III.1 Détermination de la valeur du produit $k_L \times a$

Dans cette partie, on va déterminer la valeur du produit $k_L \times a$ à partir des données expérimentales enregistrées en l'absence de réaction dans la phase liquide. Si l'expression du débit molaire de A demandée à la question **Q1.9)**, page 5, n'a pas été trouvée, on utilisera la notation formelle suivante pour écrire le code : $F_A = f(t, (k_L \times a))$.

Q3.1) Les données expérimentales sont disponibles sous forme d'un fichier texte nommé « `data.txt` » dans lequel les informations sont présentées sous forme de colonnes (tableau 1). La première colonne du fichier correspond au temps (unité : s), la deuxième au débit molaire de A (unité : mol.s^{-1}) entrant dans le réacteur enregistré au cours de la première expérience (cas de l'absorption dans le solvant sans réaction) et la troisième colonne au débit molaire de A (unité : mol.s^{-1}) entrant dans le réacteur enregistré au cours de la seconde expérience (en présence du réactif B).

```
00.0 1.9231E-04 1.9230E-04
10.0 1.8293E-04 1.9108E-04
20.0 1.7401E-04 1.8991E-04
30.0 1.6552E-04 1.8878E-04
40.0 1.5745E-04 1.8771E-04
50.0 1.4977E-04 1.8667E-04
```

Tableau 1 – Données expérimentales contenues dans le fichier « `data.txt` ».

Q3.1.a) Indiquer la syntaxe à utiliser pour charger le fichier « `data.txt` » qui contient les données expérimentales. Donner le code permettant de créer trois vecteurs t^{exp}, F_A^{exp1} et F_A^{exp2} correspondant respectivement aux données des première, deuxième et troisième colonnes. Donner également le code qui permet de déterminer le nombre de points expérimentaux n.

Q3.1.b) Donner la syntaxe permettant de tracer sur un graphe l'évolution du débit molaire de A en fonction du temps dans le cas où seul le phénomène d'absorption physique est étudié.

Q3.2) On souhaite déterminer la valeur du produit $k_L \times a$ grâce au modèle établi à la question **Q1.9)**, page 5, par une méthode d'optimisation numérique de paramètres. La méthode utilisée est celle des moindres carrés. Il s'agit d'une méthode qui permet de comparer des données expérimentales à un modèle mathématique la plupart du temps issu d'une théorie.

Dans le cas présent, le modèle mathématique correspond à l'expression du débit molaire F_A, que l'on notera F_A^{th1} par la suite, déterminé à la question **Q1.9)**, page 5. Ce débit molaire est une fonction du temps t et du produit $k_L \times a$ que l'on souhaite déterminer. La méthode des moindres carrés donne la valeur optimale du produit $k_L \times a$ qui permet de calculer un débit molaire théorique F_A^{th1} représentant le mieux le débit molaire expérimental F_A^{exp1}. Cette valeur optimale est celle qui permet de minimiser la somme quadratique des déviations des mesures aux prédictions. Cette somme, notée $S(k_L \times a)$, peut se mettre sous la forme suivante :

$$S(k_L \times a) = \sum_{i=1}^{n} \left(F_{Ai}^{exp1} - F_{Ai}^{th1}(k_L \times a) \right)^2$$

avec n le nombre de points expérimentaux, i l'indice correspondant au point expérimental enregistré au temps t_i^{exp} $(1 \leq i \leq n)$, F_{Ai}^{exp1} le débit molaire expérimental obtenu au temps t_i^{exp} et $F_{Ai}^{th1}(k_L \times a)$ le débit molaire théorique calculé au temps t_i^{exp} grâce au modèle établi à la question **Q1.9)**, page 5.

Ecrire une fonction `smc(kla,t_exp,Fa_exp1)` qui retourne la valeur de la quantité S. Cette fonction aura comme argument d'entrée le produit $k_L \times a$ et les vecteurs t^{exp} et F_A^{exp1} créés à la question **Q3.1.a)**, page 7.

Q3.3) Une des méthodes utilisées pour trouver le minimum d'une fonction f est celle proposée par Nelder-Mead. Il s'agit d'un algorithme d'optimisation non linéaire basé sur le concept de simplexe à $n+1$ sommets dans un espace à n dimensions. Dans le cas d'une fonction d'une variable $(n = 1)$, un simplexe est un segment.

L'algorithme de Nelder-Mead est une procédure itérative. Partant d'un segment initial $[MN]$, l'algorithme va générer une succession de segments par des transformations simples au cours des itérations : le segment se déplace et se réduit jusqu'à ce que ses extrémités se rapprochent d'un point où la fonction présente un minimum (ce minimum peut être un minimum local).

Soient x_M et x_N les abscisses des points M et N. Les transformations subies par le segment (illustrées à la figure 6, page 9) sont basées sur la comparaison des valeurs de la fonction f aux extrémités du segment.

- La première étape consiste à réindexer si nécessaire les deux extrémités du segment de manière à ce que $f(x_M) \leq f(x_N)$.
- L'extrémité N pour laquelle la fonction f est maximale est remplacée par une nouvelle extrémité. On introduit alors le point R, réflexion de N par rapport à M, tel que $x_R = x_M + (x_M - x_N)$.
- Si $f(x_R) < f(x_M)$, le segment est étiré dans cette direction (car la réflexion se rapproche du minimum). On introduit un point E (étirement du segment) tel que $x_E = x_M + 2(x_M - x_N)$ qui permet éventuellement de se rapprocher encore un peu plus du minimum. Le point N est substitué par le point R si $f(x_R) < f(x_E)$, sinon par E.
- Si $f(x_R) > f(x_M)$, le segment est réduit dans la direction opposée de la réflexion (car la réflexion s'éloigne du minimum). On introduit le point C_1 (contraction du segment) qui est défini par $x_{C_1} = x_N + 1/2\,(x_M - x_N)$. Si $f(x_{C_1}) < f(x_M)$, N est remplacé par C_1. Sinon N est remplacé par C_2, homothétie de rapport -1 et de centre M du point C_1 $(x_{C_2} = x_M + 1/2\,(x_M - x_N))$.

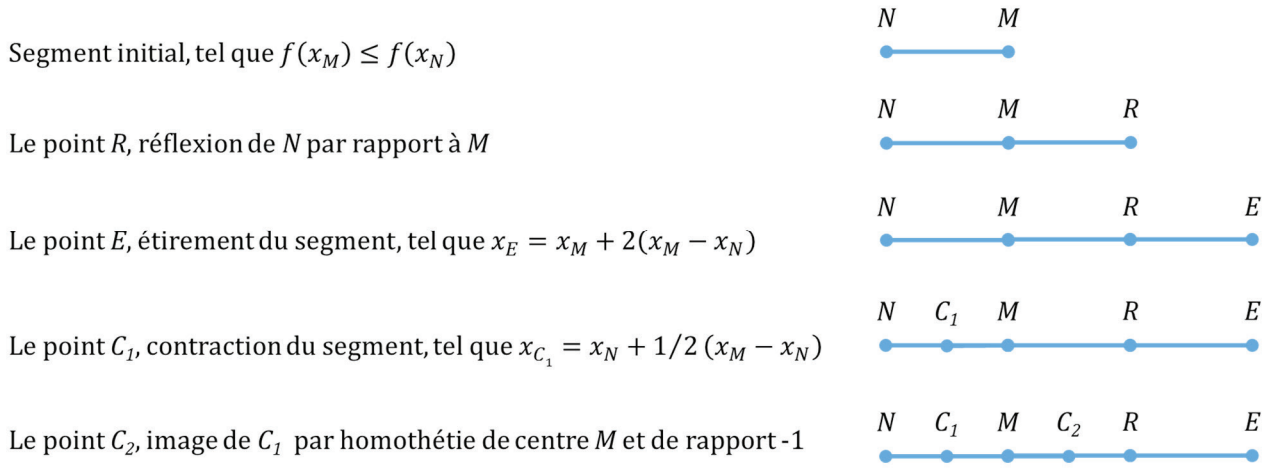


Figure 6 – Illustration des transformations subies par le segment initial $[MN]$ dans la méthode d'optimisation de Nelder-Mead.

Q3.3.a) Soit $f(x)$ une fonction dont on cherche le minimum. Soient x_M et x_N les abscisses des extrémités d'un segment $[MN]$. Ecrire le code permettant de réindexer les deux extrémités du segment de manière à ce que $f(x_M)$ soit inférieure ou égale à $f(x_N)$.

Q3.3.b) Partant du segment $[MN]$ réindexé, écrire le code permettant de transformer le segment $[MN]$ au cours d'une itération de la méthode de Nelder-Mead. Chaque étape du code devra être commentée.

Q3.3.c) Construire une boucle permettant de réaliser des itérations successives de la méthode de Nelder-Mead. On prendra $x_M^0 = 1$ et $x_N^0 = 10^{-5}$ pour les abscisses des deux extrémités du segment initial $[MN]$. Cette boucle sera interrompue lorsque l'écart relatif $abs\left(\frac{x_M - x_N}{x_M}\right)$ sera inférieur à 10^{-4} ou lorsque le nombre maximum d'itérations, fixé par l'utilisateur, sera atteint soit $Itmax = 1\,000$.

Q3.4) La méthode de Nelder-Mead appliquée à la fonction $S(k_L \times a)$ conduit à $(k_L \times a)_{opt} = 5{,}0 \times 10^{-3}$. Donner le code permettant de calculer le débit molaire théorique grâce à la valeur optimisée du produit $(k_L \times a)_{opt}$.

Q3.5) Indiquer la syntaxe à utiliser pour comparer sur un même graphe les débits molaires théorique et expérimental. Expliquer l'intérêt de comparer la courbe théorique et les points expérimentaux.

Q3.6) On souhaite développer un algorithme de tri permettant de réindexer les sommets d'un simplexe à $p + 1$ sommets ($X = \{x_1, x_2, \ldots x_{p+1}\}$) dans le cadre de la généralisation de la méthode de Nelder-Mead appliquée à une fonction quelconque $z(x)$ de p variables ($x \in \mathbb{R}^p$). On propose d'utiliser la méthode du tri par sélection pour réindexer les sommets du simplexe de sorte que les valeurs que prend la fonction $z(x)$ en chaque sommet du simplexe soient classées de manière croissante.

La méthode du tri par sélection repose sur la recherche du plus petit élément d'une liste. Une fois identifié, on échange cet élément avec le premier élément de la liste. Il s'agit d'une procédure itérative (figure 7). Lors de la première itération, on échange le premier élément ($i = 1$) avec le plus petit de la liste. Lors de la i^e itération, on échange le i^e élément par le plus petit élément en ne considérant que ceux à partir de la i^e position.

Vecteur initial	10	2	- 3	5	- 1	20	60	48	- 15	9
1re itération	**- 15**	2	- 3	5	- 1	20	60	48	**10**	9
2e itération	*- 15*	**- 3**	**2**	5	- 1	20	60	48	10	9
3e itération	*- 15*	*- 3*	**- 1**	5	**2**	20	60	48	10	9

Figure 7 – Représentation schématique des trois premières itérations de la méthode de tri par sélection.

Ecrire le code permettant de réindexer les sommets du simplexe $X = \{x_1, x_2, \dots x_{p+1}\}$ de sorte que les valeurs que prend la fonction $z(x)$ en chaque sommet du simplexe soient classées de manière croissante. On supposera que la fonction $z(x)$ est déjà définie.

III.2 Détermination des valeurs de a et k_L

Dans cette partie, on souhaite déterminer la valeur de l'aire interfaciale a en comparant les débits molaires expérimentaux obtenus lors de la deuxième expérience (en présence du réactif B dans le solvant) avec le débit molaire théorique calculé à partir du modèle établi à la question **Q2.7)**, page 6. Ce modèle est un système de deux équations différentielles ordinaires que l'on peut écrire sous la forme suivante :

$$\begin{cases} \dfrac{dC_A}{dt} = g\big(t, C_A, C_B, a, (k_L \times a)\big) \\ \dfrac{dC_B}{dt} = h(t, C_A, C_B, a) \end{cases}$$

La fonction g est une fonction du produit $k_L \times a$ dont la valeur a été déterminée grâce à la première expérience ($(k_L \times a)_{opt} = 5{,}0 \times 10^{-3}$ s^{-1}), du temps t, des concentrations C_A et C_B de A et B dans le solvant et de l'aire interfaciale a. La fonction h est une fonction du temps t, des concentrations C_A et C_B de A et B dans le solvant et de l'aire interfaciale a.

On propose d'utiliser la méthode d'Euler pour résoudre le système d'équations différentielles.

Q3.7) Ecrire deux fonctions `G(t,CA,CB,A,kla)` et `H(t,CA,CB,A)` permettant de calculer les valeurs des fonctions $g\big(t, C_A, C_B, a, (k_L \times a)\big)$ et $h(t, C_A, C_B, a)$.

Q3.8) La méthode utilisée pour résoudre le système d'équations différentielles est la méthode d'Euler à pas fixe. On note t_0 le temps initial, t_f le temps final d'intégration et Δt le pas d'intégration.

Q3.8.a) Soient n le nombre d'expériences réalisées et Δt^{exp} l'intervalle de temps entre deux mesures expérimentales successives du débit molaire du polluant A. Pour réaliser la comparaison entre les débits molaires théoriques et expérimentaux, on ne s'intéresse qu'aux valeurs du débit molaire théorique F_{Ai}^{th2} aux instants particuliers $t_i^{exp} = \Delta t^{exp} \times i$ avec i variant de 1 à n.

La méthode d'Euler ne permet d'obtenir un résultat correct que si le pas d'intégration, Δt, est suffisamment faible. Par conséquent, on souhaite utiliser un pas d'intégration Δt cent fois plus petit que l'intervalle de temps entre deux mesures expérimentales Δt^{exp}. Le temps d'intégration est donc discrétisé en intervalles de durée Δt et les valeurs des concentrations C_A et C_B sont calculées aux instants particuliers $t_j = \Delta t \times j$ avec j variant de 0 à m.

Donner la syntaxe permettant de calculer le pas de temps Δt ainsi que le nombre d'intervalles m (m est un nombre entier) en fonction de t_0 (le premier élément du vecteur t^{exp}), t_f (le dernier élément du vecteur t^{exp}) et Δt^{exp} (l'intervalle de temps entre deux mesures expérimentales successives du débit molaire).

Q3.8.b) Donner une expression de $C_A(t + \Delta t)$ à l'ordre 1 ($o(\Delta t)$) en fonction de C_A et $\frac{dC_A}{dt}$ évaluées en t à l'aide d'un développement limité de la fonction $t \mapsto C_A(t)$.

Q3.8.c) En déduire une valeur approchée de $\left.\frac{dC_A}{dt}\right|_t$ à l'ordre 0 ($o(1)$) en fonction de $C_A(t)$, $C_A(t + \Delta t)$ et Δt. Faire de même pour donner une valeur approchée de $\left.\frac{dC_B}{dt}\right|_t$ à l'ordre 0 ($o(1)$) en fonction de $C_B(t)$, $C_B(t + \Delta t)$ et Δt.

Q3.8.d) On note C_{Aj} et C_{Bj} les concentrations $C_A(t_j)$ et $C_B(t_j)$. De même, on note C_{Aj+1} et C_{Bj+1} les concentrations $C_A(t_{j+1})$ et $C_B(t_{j+1})$. Donner une expression de la dérivée par rapport au temps de C_A évaluée à l'instant t_j, notée $\left.\frac{dC_A}{dt}\right|_{t_j}$, en fonction de Δt, C_{Aj} et C_{Aj+1}. De même, donner une expression de la dérivée par rapport au temps de C_B évaluée à l'instant t_j, notée $\left.\frac{dC_B}{dt}\right|_{t_j}$, en fonction de Δt, C_{Bj} et C_{Bj+1}.

Q3.8.e) Donner le système de deux équations permettant de calculer C_{Aj+1} et C_{Bj+1} en fonction de $C_{Aj}, C_{Bj}, \Delta t, g\left(t_j, C_{Aj}, C_{Bj}, a, (k_L \times a)\right)$ et $h(t_j, C_{Aj}, C_{Bj}, a)$.

Q3.8.f) Ecrire une fonction `Euler(G,H,t0,tf,CA0,CB0,Dt,m,a,kla)` qui retourne deux vecteurs : le vecteur temps et le vecteur correspondant à la concentration C_A calculée par la méthode d'Euler.

Q3.9) On souhaite utiliser la méthode des moindres carrés pour déterminer la valeur optimale de l'aire interfaciale a en utilisant une stratégie similaire à celle utilisée pour obtenir la valeur optimisée du produit $(k_L \times a)$. Pour cela, on a besoin de connaître les valeurs du débit molaire de A F_{Ai}^{th2} calculé aux instants particuliers $t_i^{exp} = \Delta t^{exp} \times i$ avec i variant de 1 à n.

Donner le code permettant de créer un vecteur F_A^{th2} contenant les valeurs du débit molaire de A calculé aux instants particuliers t_i^{exp} à partir des valeurs de la concentration de A (C_A^{th2}) calculées en utilisant la fonction `Euler` de la question **Q3.8.f)**.

Q3.10) La valeur optimale de l'aire interfaciale a est celle qui permet de minimiser la fonction $S_2(a)$ qui peut se mettre sous la forme suivante :

$$S_2(a) = \sum_{i=1}^{n} \left(F_{Ai}^{exp2} - F_{Ai}^{th2}(a) \right)^2$$

avec F_{Ai}^{exp2} le débit molaire expérimental obtenu au temps t_i^{exp}, $F_{Ai}^{th2}(a)$ le débit molaire théorique calculé au temps t_i^{exp} (question **Q3.9)**, page 11) et n le nombre de points expérimentaux.

Donner le code permettant de construire une fonction `smc2(a,t_exp,Fa_exp2)` qui retourne la valeur de la quantité S_2. Cette fonction aura comme argument d'entrée l'aire interfaciale a et les vecteurs t^{exp} et F_A^{exp2} créés à la question **Q3.1.a)**, page 7.

Q3.11) La méthode de Nelder-Mead appliquée à la fonction $S_2(a)$ conduit à a_{opt} = 10,0 $m^2.m^{-3}$. Indiquer le code pour calculer le coefficient de transfert k_L et afficher les valeurs de a_{opt} et k_L à l'écran.

III.3 Utilisation du modèle pour réaliser des simulations

Maintenant que l'on connaît les valeurs des paramètres a et k_L, on souhaite utiliser le modèle pour réaliser des prédictions. On propose d'étudier l'influence de la concentration initiale C_{B0} du réactif B dans la phase liquide sur le débit molaire en entrée du réacteur (F_A^{th2}).

Q3.12) Expliquer qualitativement comment doit varier le débit molaire en entrée du réacteur avec la concentration initiale du réactif B.

Q3.13) Justifier le concept « d'accélération du transfert par la réaction chimique ».

Q3.14) Ecrire le code qui permettrait de réaliser une prédiction du débit molaire F_A^{th} à l'aide du modèle et des paramètres optimisés (k_L et a) en prenant C_{B0} comme valeur pour la concentration initiale du réactif B, une valeur rentrée par l'utilisateur (unité : $mol.m^{-3}$). On prendra le même pas Δt que précédemment et comme durée d'intégration 1 000 s.

Q3.15) On souhaite calculer la quantité de matière de polluant A, n_A, à partir du débit molaire F_A^{th} obtenu à la question précédente par intégration sur l'intervalle 0 – 1 000 s. La méthode utilisée pour réaliser l'intégration est la méthode des trapèzes. Donner le code permettant de calculer n_A par cette méthode.

ANNEXE A : COMMANDES ET FONCTIONS USUELLES DE SCILAB

A=[a b c d;e f g h;i j k l]

Description : commande permettant de créer une matrice dont la première ligne contient les éléments a, b, c, d, la seconde ligne contient les éléments e, f, g, h et la troisième, les éléments i, j, k, l.

Exemple : A=[1 2 3 4 5;3 10 11 12 20;0 1 0 0 2]

⇨ 1. 2. 3. 4. 5.
3. 10. 11. 12. 20.
0. 1. 0. 0. 2.

A(i,j)

Description : fonction qui retourne l'élément (i, j) de la matrice A. Pour accéder à l'intégralité de la ligne i de la matrice A, on écrit $A(i,:)$. De même, pour obtenir toute la colonne j de la matrice A, on utilise la syntaxe $A(:,j)$.

Arguments d'entrée : les coordonnées de l'élément dans la matrice A.

Argument de sortie : l'élément (i, j) de la matrice A.

Exemple : A=[1 2 3 4 5;3 10 11 12 20;0 1 0 0 2]
A(2,4)
⇨ 12
A(2,:)
⇨ 3. 10. 11. 12. 20.
A(:,3)
⇨ 3.
11.
0.

x=[x1:Dx:x2]

Description : commande permettant de créer un vecteur dont les éléments sont espacés de Dx et dont le premier élément est x_1 et le dernier élément est le plus grand multiple de Dx inférieur ou égal à x_2.

ATTENTION : le vecteur ainsi créé est un vecteur ligne. Pour convertir un vecteur ligne en un vecteur colonne, on le transpose en utilisant l'apostrophe « ' » : x_trans=x'.

Exemple *:* x=[2:0.5:4.2]

⇨ 2. 2.5 3. 3.5 4.
x_trans=x'
⇨ 2.
2.5
3.
3.5
4.

zeros(n,m)

Description : fonction créant une matrice (tableau) de dimensions $n \times m$ dont tous les éléments sont nuls.

Arguments d'entrée : deux entiers n et m correspondant aux dimensions de la matrice à créer.

Argument de sortie : un tableau (matrice) d'éléments nuls.

Exemple : zeros(3,4)

⇨ 0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0.

plot(x,y)

Description : fonction permettant de tracer sur un graphique n points dont les abscisses sont contenues dans le vecteur x et les ordonnées dans le vecteur y.
Arguments d'entrée : un vecteur d'abscisses x (tableau de dimension n) et un vecteur d'ordonnées y (tableau de dimension n).
Exemple : x= [3:0.1:5]
y=sin(x)
plot(x,y)

read("nom_fichier",m,n)

Description : fonction permettant de lire les données sous forme de matrice dans un fichier texte et de les stocker dans une matrice.
Arguments d'entrée : un nom de fichier contenant des données sous forme de matrice de dimension (m,n) et les dimensions de la matrice m (nombre de lignes) et n (nombre de colonnes). On prend m = -1 si le nombre de lignes n'est pas connu a priori.
Exemple : data= read("fichier.txt",-1,2)
//dans cet exemple, data est une matrice constituée des deux premières
//colonnes se trouvant dans le fichier nommé fichier.txt.

sum(x)

Description : fonction permettant de faire la somme des éléments d'un vecteur ou tableau
Arguments d'entrée : un vecteur ou un tableau de réels, entiers ou complexes.
Argument de sortie : un scalaire y qui est la somme des éléments de x.
Exemple : y= sum(x)
//y retourne la somme des éléments de x.

ANNEXE B : FONCTIONS DE PYTHON

B.1. BIBLIOTHEQUE NUMPY DE PYTHON

Dans les exemples ci-dessous, la bibliothèque numpy a préalablement été importée à l'aide de la commande : **import numpy as np**

On peut alors utiliser les fonctions de la bibliothèque, dont voici quelques exemples :

np.array(liste)

Description : fonction permettant de créer une matrice (de type tableau) à partir d'une liste.
Argument d'entrée : une liste définissant un tableau à 1 dimension (vecteur) ou 2 dimensions (matrice).
Argument de sortie : un tableau (matrice).
Exemple : np.array([4,3,2])
⇨ [4 3 2]
np.array([[5],[7],[1]])
⇨ [[5]
[7]
[1]]
np.array([[3,4,10],[1,8,7]])
⇨ [[3 4 10]
[1 8 7]]

$A[i,j]$.

Description : fonction qui retourne l'élément $(i+1, j+1)$ de la matrice A. Pour accéder à l'intégralité de la ligne i+1 de la matrice A, on écrit $A[i,:]$. De même, pour obtenir toute la colonne j+1 de la matrice A, on utilise la syntaxe $A[:,j]$.
Arguments d'entrée : une liste contenant les coordonnées de l'élément dans le tableau A.
Argument de sortie : l'élément $(\mathrm{i}+1, \mathrm{j}+1)$ de la matrice A.
ATTENTION : en langage Python, les lignes d'un tableau A de dimension $n \times m$ sont numérotées de 0 à $n-1$ et les colonnes sont numérotées de 0 à $m-1$

Exemple :
```
A=np.array([[3,4,10],[1,8,7]])
A[0,2]
⇨ 10
A[1,:]
⇨ [1  8  7]
A[:,2]
⇨ [10  7]
```

np.zeros((n,m))

Description : fonction créant une matrice (tableau) de dimensions $n \times m$ dont tous les éléments sont nuls.
Arguments d'entrée : un tuple de deux entiers correspondant aux dimensions de la matrice à créer.
Argument de sortie : un tableau (matrice) d'éléments nuls.

Exemple :
```
np.zeros((3,4))
⇨ [[0 0 0 0]
   [0 0 0 0]
   [0 0 0 0]]
```

np.linspace(Min,Max,nbElements)

Description : fonction créant un vecteur (tableau) de $nbElements$ nombres espacés régulièrement entre Min et Max. Le 1er élément est égal à Min, le dernier est égal à Max et les éléments sont espacés de $(Max - Min)/(nbElements - 1)$:
Arguments d'entrée : un tuple de 3 entiers.
Argument de sortie : un tableau (vecteur).

Exemple :
```
np.linspace(3,25,5)
⇨ [3  8.5  14  19.5  25]
```

np.loadtxt('nom_fichier',delimiter='string',usecols=[n])

Description : fonction permettant de lire les données sous forme de matrice dans un fichier texte et de les stocker sous forme de vecteurs.
Arguments d'entrée : le nom du fichier qui contient les données à charger, le type de caractère utilisé dans ce fichier pour séparer les données (par exemple un espace ou une virgule) et le numéro de la colonne à charger (ATTENTION, la première colonne porte le numéro 0).
Argument de sortie : un tableau.

Exemple :
```
data=np.loadtxt('fichier.txt',delimiter=' ',usecols=[0])
 #dans cette exemple data est un vecteur qui correspond à la première
 #colonne de la matrice contenue dans le fichier fichier.txt
```

B.2. BIBLIOTHEQUE MATPLOTLIB.PYPLOT DE PYTHON

Cette bibliothèque permet de tracer des graphiques. Dans les exemples ci-dessous, la bibliothèque matplotlib.pyplot a préalablement été importée à l'aide de la commande :
import matplotlib.pyplot as plt

On peut alors utiliser les fonctions de la bibliothèque, dont voici quelques exemples :

plt.plot(x,y)

Description : fonction permettant de tracer un graphique de n points dont les abscisses sont contenues dans le vecteur x et les ordonnées dans le vecteur y. Cette fonction doit être suivie de la fonction **plt.show()** pour que le graphique soit affiché.
Arguments d'entrée : un vecteur d'abscisses x (tableau de dimension n) et un vecteur d'ordonnées y (tableau de dimension n).
Argument de sortie : un graphique.
Exemple :

```
x= np.linspace(3,25,5)
y=sin(x)
plt.plot(x,y)
plt.title('titre_graphique')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

plt.title('titre')

Description : fonction permettant d'afficher le titre d'un graphique.
Argument d'entrée : une chaîne de caractères.

plt.xlabel('nom')

Description : fonction permettant d'afficher le contenu de nom en abscisse d'un graphique.
Argument d'entrée : une chaîne de caractères.

plt.ylabel('nom')

Description : fonction permettant d'afficher le contenu de nom en ordonnée d'un graphique.
Argument d'entrée : une chaîne de caractères.

plt.show()

Description : fonction réalisant l'affichage d'un graphe préalablement créé par la commande **plt.plot(x,y)**. Elle doit être appelée après la fonction plt.plot et après les fonctions plt.xlabel et plt.ylabel.

B.3. FONCTION INTRINSEQUE DE PYTHON

sum(x)

Description : fonction permettant de faire la somme des éléments d'un vecteur ou tableau.
Arguments d'entrée : un vecteur ou un tableau de réel, entier ou complexe.
Argument de sortie : un scalaire y qui est la somme des éléments de x.
Exemple :

```
y= sum(x)
 //y retourne la somme des éléments de x.
```

FIN

IMPRIMERIE NATIONALE – 16 1221 – D'après documents fournis

Travailler les chapitres de ce sujet

Prepa Help suit le programme de PC chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Équations différentielles linéaires du premier ordre
- Grandeurs, unités, analyse dimensionnelle et ordres de grandeur
- Cinétique : évolution temporelle d'un système chimique
- Bases de Python : expressions, types, instructions, fonctions
- Tris par sélection, à bulles, par insertion et par comptage
- Fichiers texte et lecture de données

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/ccinp/pc