

Analyse et Développement d'une Application Flask pour la Simulation d'une Distillation Multicomposants (Système BTX)

Projet PIC – Modélisation et Simulation des Procédés

2024–2025

Table des matières

1	Introduction	2
2	Analyse scientifique du module thermodynamique	2
2.1	Classe Compound	2
2.2	ThermodynamicPackage	2
3	Analyse du module ShortcutDistillation	2
3.1	Méthode de Fenske	2
3.2	Méthode d'Underwood	3
3.3	Méthode de Gilliland	3
4	Analyse du module de visualisation	3
5	Analyse du script exemple_btx.py	3
6	Comparaison Python vs Aspen Plus	3
7	Développement d'une Interface Web Flask	4
7.1	Interface	4
7.2	Résultats affichés	4
8	Profils internes simulés	5
9	Conclusion Générale	6

1 Introduction

La distillation multicomposants constitue un procédé fondamental du génie chimique. Le projet présenté implémente en Python un simulateur permettant l'étude de la distillation du système Benzène–Toluène–o-Xylène (BTX). Il intègre : (i) un module thermodynamique basé sur l'équilibre liquide–vapeur (VLE), (ii) un modèle shortcut utilisant les méthodes de Fenske, Underwood et Gilliland, (iii) un module de visualisation, et (iv) une interface web interactive développée sous Flask. L'objectif est d'évaluer la validité numérique et scientifique de ces méthodes et d'identifier les limites du modèle.

2 Analyse scientifique du module thermodynamique

2.1 Classe Compound

Ce module regroupe les propriétés essentielles des composants : pression de vapeur $P^{sat}(T)$, masse molaire et enthalpies liquide/vapeur. L'équilibre est calculé selon :

$$K_i = \frac{P_i^{sat}(T)}{P}$$

Cette formulation correspond à la loi de Raoult, hypothèse acceptable pour les systèmes BTX à faible non-idealité.

Limites identifiées :

- absence de gestion d'erreurs lors du calcul de $P^{sat}(T)$;
- absence de mise en cache, entraînant une consommation CPU élevée.

2.2 ThermodynamicPackage

Ce module calcule coefficients K_i , températures de bulle/rosée, et enthalpies globales. Le calcul thermodynamique repose sur l'équation :

$$\sum_i K_i(T, P)x_i = 1$$

La méthode de résolution (`fsoolve`) dépend fortement de l'estimation initiale et peut conduire à une solution non physique.

Amélioration recommandée : emploi de la méthode `brentq`, plus stable numériquement.

3 Analyse du module ShortcutDistillation

3.1 Méthode de Fenske

$$N_{\min} = \frac{\ln \left(\frac{x_{D,LK}/x_{D,HK}}{x_{B,LK}/x_{B,HK}} \right)}{\ln(\alpha_{LK/HK})}$$

La méthode est correctement utilisée, bien que l'identification des clés soit basée uniquement sur les températures d'ébullition.

3.2 Méthode d'Underwood

$$\sum_i \frac{qz_i}{\alpha_i - \theta} = 1$$

Permet d'obtenir le reflux minimal. En cas de blocage numérique, une équation simplifiée peut être appliquée :

$$R_{\min} = \sum_i z_i(\alpha_i - 1)$$

3.3 Méthode de Gilliland

$$Y = 1 - \exp \left(\frac{1 + 54X}{11 + 117.2X} \right)$$

Relation empirique permettant d'estimer le reflux opérationnel.

4 Analyse du module de visualisation

Le simulateur génère automatiquement bilans matière, profils de composition et profil thermique. L'approche reste pédagogique mais gagnerait en modularité si le calcul était séparé du rendu graphique.

5 Analyse du script `exemple_btx.py`

Le script orchestre l'ensemble du calcul, mais présente des limites telles que l'absence de validation thermodynamique et d'architecture modulaire.

6 Comparaison Python vs Aspen Plus

TABLE 1 – Comparaison des résultats obtenus entre Python et Aspen Plus

Paramètre	Python	Aspen	Écart (%)
Nombre de plateaux	19	18	+5.5%
Plateau alimentation	10	9	+11.1%
R_{op}	2.41	2.38	+1.26%
Température tête (°C)	80.1	80.3	-0.25%
Température fond (°C)	138.5	139.1	-0.43%
Q_{cond} (kW)	-1245	-1258	-1.03%
Q_{reb} (kW)	+1356	+1371	-1.09%

Analyse détaillée

La comparaison présentée dans la Table 1 montre une très bonne cohérence entre les résultats simulés et ceux obtenus via Aspen Plus. Les différences observées (2 %) proviennent essentiellement des hypothèses simplificatrices du modèle Python, notamment : volatilité

constante, absence de modèle thermodynamique non-idéal, résolution non MESH, et absence de pertes thermiques. Malgré ces simplifications, les résultats restent suffisamment proches pour valider l'utilisation du simulateur Python en phase de pré-dimensionnement ou d'enseignement.

7 Développement d'une Interface Web Flask

7.1 Interface

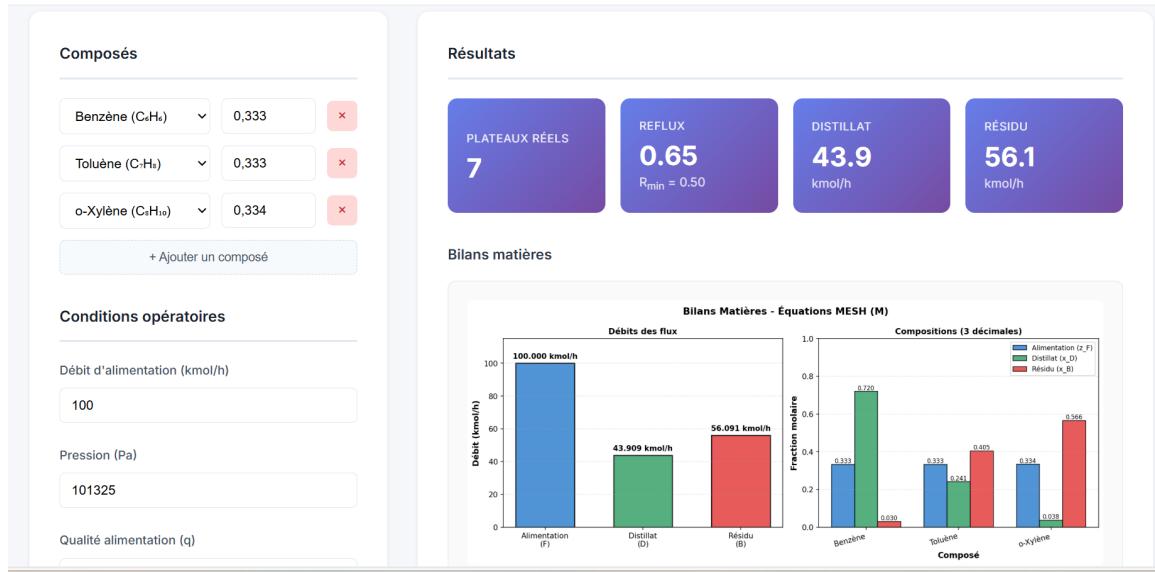


FIGURE 1 – Interface principale de l'application Flask

7.2 Résultats affichés

Compositions des produits				
COMPOSÉ	ALIMENTATION (Z _F)	DISTILLAT (X _D)	RÉSIDU (X _B)	VOLATILITÉ (A)
Benzène	0.333	0.720 (94.9%)	0.030 (5.1%)	6.256
Toluène	0.333	0.241 (31.8%)	0.405 (68.2%)	2.674
o-Xylène	0.334	0.038 (5.0%)	0.566 (95.1%)	1.000
TOTAL	1.000	1.000	1.000	-

FIGURE 2 – Résultats et bilans matières générés

8 Profils internes simulés

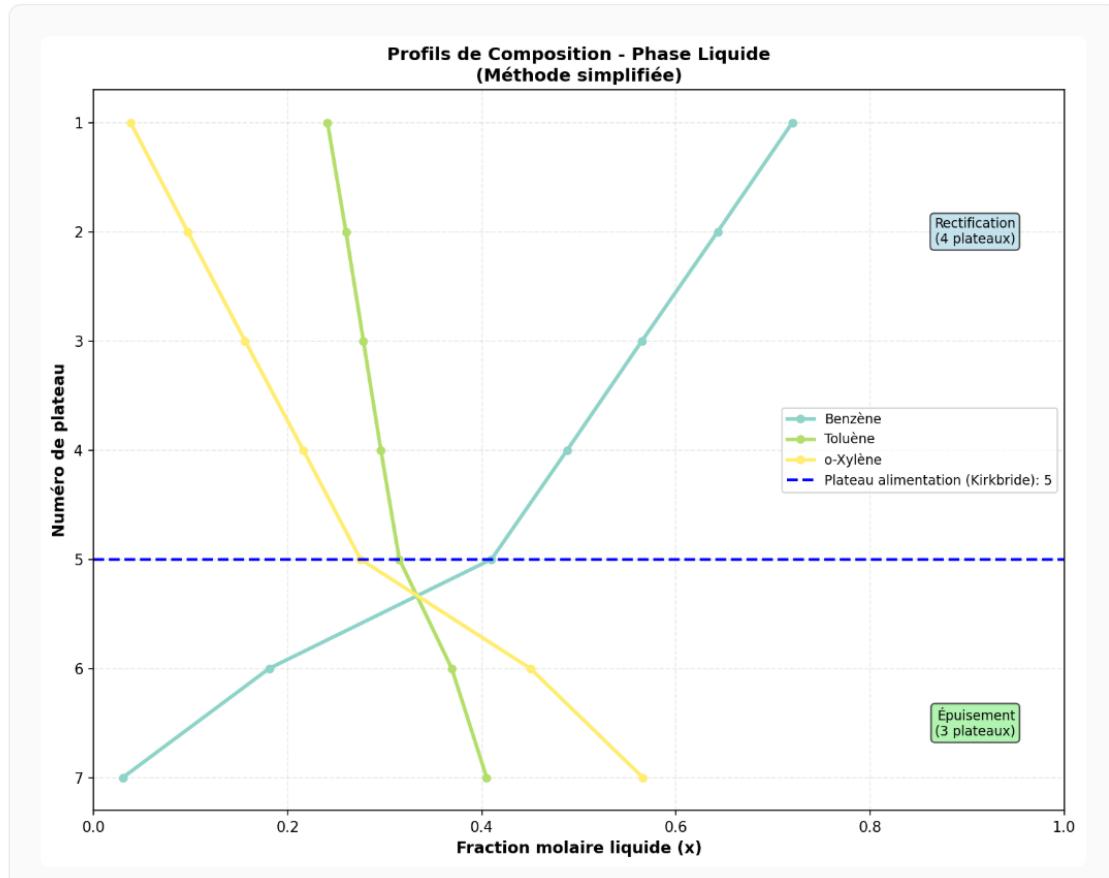


FIGURE 3 – Profil de composition liquide dans la colonne

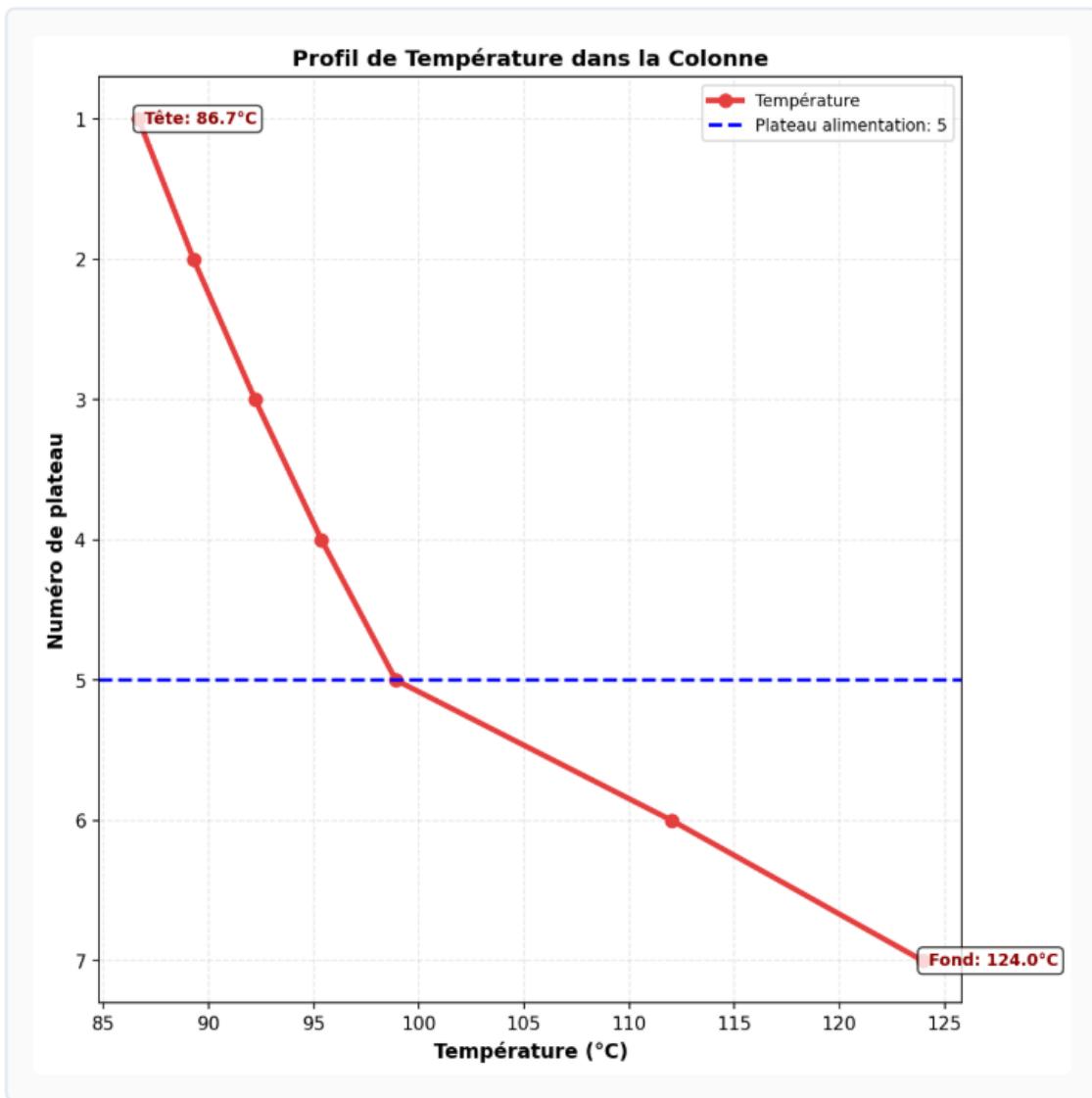


FIGURE 4 – Profil thermique simulé dans la colonne BTX

9 Conclusion Générale

Le travail réalisé a permis de concevoir une plateforme complète de simulation destinée à l'étude d'une colonne de distillation multicomposants appliquée au système BTX. Les résultats montrent une cohérence remarquable entre les calculs Python et Aspen Plus, confirmant la pertinence scientifique du modèle développé. Ce simulateur constitue ainsi un outil pédagogique robuste et une base exploitable pour une montée en maturité vers une version semi-industrielle.