ENSTA

IP PARIS

MO102 - Projet Matlab

Optimisation Technico-économique d'un Procédé de Fabrication du Bioéthanol

Julian Garrec

(texte original de Patrice Paricaud)

Version 2023-02-10

Introduction **3**

1. Première partie du Projet : étude des équilibres liquide-vapeur du mélange eau-éthanol **4**

1.1. Diagramme de phase température-composition à pression fixée 4

1.2. Résolution de l'équilibre liquide-vapeur 5

a) Équations d'équilibre 5

b) Pressions de vapeur des composés purs* 6

c) Coefficients d'activité* 7

d) Algorithme de calcul d'équilibre à composition et P fixée** 8

e) Ajustement du modèle NRTL sur les données expérimentales 10

2. Deuxième partie du Projet: Dimensionnement et optimisation de la colonne, en utilisant la méthode de McCabe et Thiele **13**

2.1. Mise en équations du problème 15

2.2. Méthode de McCabe et Thiele 17

a) Description de la méthode graphique 18

b) Détermination du taux de reflux minimal Rmin 19

c) Détermination des dimensions réelles de la colonne 20

d) Calcul des coûts annuels 20

Introduction

La production de bioéthanol nécessite un apport d'énergie important notamment lors de l'étape de séparation, entraînant non seulement des coûts de production mais également des émissions de gaz à effet de serre. Il convient donc de d'assurer que la fabrication du biocarburant nécessite le moins possible d'énergie et qu'elle respecte au mieux l'environnement. *Maltab* va nous permettre de réaliser de manière automatique l'optimisation d'une partie du procédé de fabrication du bioéthanol, tenant compte des dépenses énergétiques, des coûts de maintenance et des contraintes environnementales (quotas d'émission de CO2). Nous verrons notamment, au cours de ce projet, comment Matlab permet d'optimiser une fonction à paramètres multiples, de résoudre une équation non linéaire, gérer des fichiers en entrée-sortie, et utiliser les fonctions graphiques. Un problème d'équations différentielles non linéaires est également proposé dans la partie 3.

Le procédé doit fabriquer 100 000 tonnes d'éthanol par an. La **figure 1** montre le schéma simplifié du procédé utilisant la betterave à sucre comme matière première. On suppose que la solution en sortie du fermenteur a une fraction molaire $x_F = 0.03$ en éthanol (correspondant à un degré alcoolique de 10% en vol.). Cette solution aqueuse d'éthanol est envoyée dans une colonne de distillation atmosphérique (**Pression = 760mmHg**). On doit obtenir de l'éthanol de fraction molaire $x_D = 0.84$ en tête de colonne. Le mélange est ensuite déshydraté pour obtenir le bioéthanol à 99%. Au bouilleur, à la sortie basse de la colonne, le flux sortant à une fraction molaire de $x_B = 0.0001$ en éthanol.

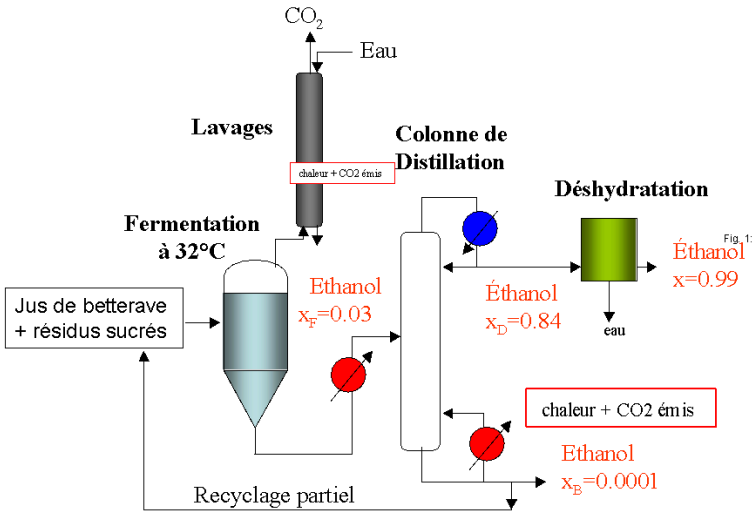


Fig. 1: Schéma de procédé. Fabrication du bioéthanol

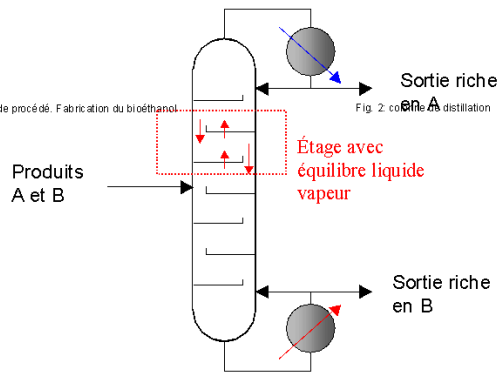


Fig. 2: colonne de distillation

Au cours de ce projet, nous allons nous intéresser uniquement au dimensionnement et à l'optimisation économique de la colonne de distillation (**Figures 1** et **2**). Cette optimisation doit tenir compte des exigences environnementales, économiques, et des contraintes de pureté des produits en sortie du procédé de séparation. Nous ferons l'hypothèse que les solutions ne contiendront que de l'eau et de l'éthanol, donc nous négligerons la présence d'autres résidus (sucre, bactéries, produits annexes.).

La colonne de distillation doit permettre de faire concentrer une solution eau+éthanol contenant 3% en mole d'éthanol pour obtenir une solution à 84% en mole d'éthanol. Pour cela, nous ferons certaines hypothèses de modélisation et utiliserons la méthode de McCabe et Thiele. Lors de l'optimisation du coût, on tiendra compte des émissions de CO_2 venant des dépenses énergétiques.

Notez que dans le descriptif du projet, les objectifs et algorithmes du projet sont bien mis en valeur et encadrés. Nous conseillons vivement de suivre l'ordre de ce descriptif, afin de décomposer le programme en sous-tâches et d'avancer plus rapidement dans le projet, le but étant que les élèves avancent rapidement et utilisent un maximum de fonction matlab.

1. Première partie du Projet : étude des équilibres liquide-vapeur du mélange eau-éthanol

1.1. Diagramme de phase température-composition à pression fixée

La colonne de distillation fonctionne selon l'équilibre liquide-vapeur du mélange eau-éthanol à la pression atmosphérique P = 760 mmHg (on suppose que P est constante le long de la colonne pour les calculs d'équilibre). Il est donc nécessaire de bien comprendre le comportement thermodynamique de ce mélange à cette pression pour tout le domaine de composition. Pour cela, on étudie le **diagramme de phase température – composition** du mélange binaire eau+éthanol à la pression P = 760 mmHg.

Ce diagramme fournit les compositions des deux phases liquides et vapeur et la température d'ébullition du mélange à l'équilibre, à la pression P =760 mmHg : la phase liquide contenant une fraction molaire xliq en éthanol est en équilibre avec une phase vapeur ayant une fraction molaire ygaz en éthanol (**figure 3**). Ce diagramme comporte un *azéotrope* autour de xliq=0.9, qui va limiter la séparation des deux produits. Le premier but du projet va être

d'utiliser matlab afin d'ajuster un modèle thermodynamique (modèle *NRTL*) sur les points expérimentaux. Ce modèle nous servira par la suite pour le dimensionnement de la colonne de distillation.

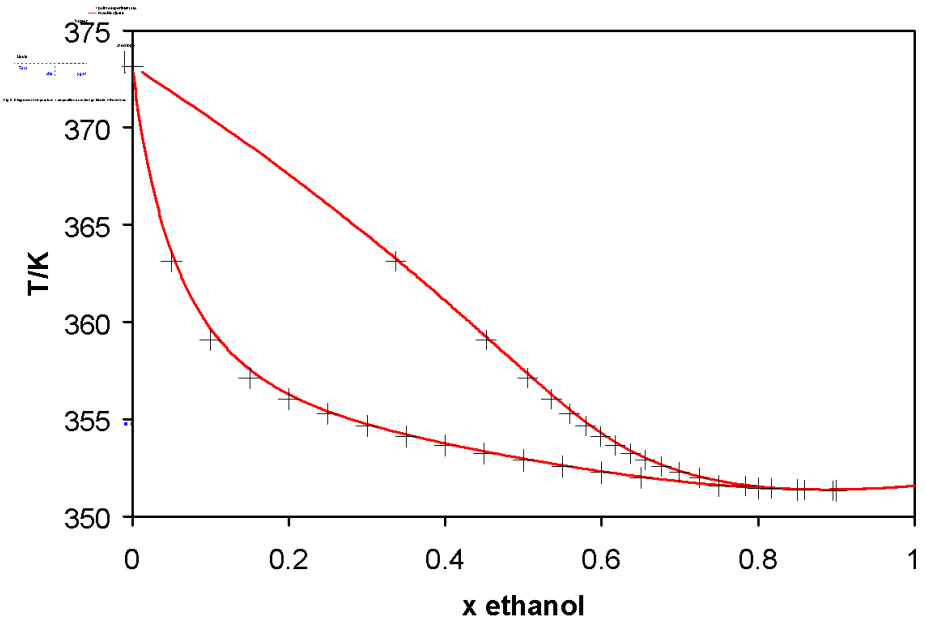


Fig. 3: Diagramme temperature- composition du mélange binaire ethanol-eau.

1.2. Résolution de l'équilibre liquide-vapeur

On suppose que **l'éthanol est le composé 1 et l'eau le composé 2**. La résolution de l'équilibre de phase va consister à résoudre les équations suivantes :

a) Équations d'équilibre

- Égalité des potentiels chimiques des constituants dans les deux phases $\left(\mu_i^{gaz} = \mu_i^{liq}\right)$.

$$Py_1 = P_1^0 x_1 \gamma_1 \quad (1)$$

$$Py_2 = P_2^0 x_2 \gamma_2 \quad (2)$$

avec

P : pression du système

x_1, x_2 : fractions molaires des composés 1 et 2 dans la phase liquide (avec $x_1 + x_2 = 1$)

y_1, y_2 : fractions molaires des composés 1 et 2 dans la phase vapeur (avec $y_1 + y_2 = 1)$.

P_1^0, P_2^0 : pressions de vapeur des composés purs 1 et 2. P_1^0, P_2^0 sont des fonctions de *T*.

γ_1, γ_2 : coefficients d'activités des composés 1 et 2 dans la phase liquide. γ_1, γ_2 sont des fonctions de la température *T* et des fractions molaires x_1, x_2 .

- Égalité de la pression dans les deux phases à la pression fixée

$$P = 760 \text{ mmHg} \qquad (3)$$

b) Pressions de vapeur des composés purs*

Les pressions de vapeur P_1^0, P_2^0 pour l'éthanol et l'eau sont données par la corrélation suivante:

$$\text{Log}_{10}\, P_i^0 = A_i + B_i / T + C_i \,\text{Log}_{10}\, T + D_i T + E_i T^2 \qquad (4)$$

avec

T : température en Kelvin

P_i^0 : pression de vapeur en mmHg

Avec pour l'éthanol

A_1 = 23.8442 ; B_1 = - 2864.2 ; C_1 = - 5.0474; D_1 = 3.7448E-11 ; E_1 = 2.7361E-07

Pour l'eau :

A_2 = 29.8605 ; B_2 = - 3152.2 ; C_2 = - 7.3037 ; D_2 = 2.4247E-09 ; E_2 = 1.809E-06

Travail sous matlab : on écrira une fonction `p0 = psat(T)`, ayant en variable d'entrée la température *T* (scalaire) et en sortie le vecteur p0 de dimension 2, contenant les valeurs des deux pressions de vapeur à la température *T*.

Test de la fonction matlab pour les pressions de vapeur : à *T* = 300K, vérifiez que les pressions de vapeur de l'éthanol et de l'eau sont respectivement P^0_1 = 65.8370 mmHg, P^0_2 =26.5355 mmHg.

Créer le script de test jjpsat.m

Ce script devra contenir :

- le test de psat pour la température T=300K
- le tracé d'un graph p=f(T) pour les deux composés

. Créer un vecteur de températures allant de 300 à 400 K.

. Faire une boucle sur les températures pour créer un vecteur de pressions pour le composé 1. Idem pour le composé 2.

. Tracer les deux courbes.

c) Coefficients d'activité*

Les coefficients d'activités γ_i sont donnés par le *modèle thermodynamique NRTL* selon:

$$\ln\gamma_1 = x_2^2\left(\tau_{21}\left(\frac{G_{21}}{x_1+x_2G_{21}}\right)^2+\frac{\tau_{12}G_{12}}{(x_2+x_1G_{12})^2}\right) \quad (5)$$

$$\ln\gamma_2 = x_1^2\left(\tau_{12}\left(\frac{G_{12}}{x_2+x_1G_{12}}\right)^2+\frac{\tau_{21}G_{21}}{(x_1+x_2G_{21})^2}\right) \quad (6)$$

avec $G_{ij}=\exp(-0.3\tau_{ij})$, $\tau_{ij}=\frac{g_{ij}}{RT}$, *R* = 8.314510 J/mol, tels que $g_{11}=g_{22}=0$. Les paramètres $g_{12}\neq 0$ et $g_{21}\neq 0$ sont les deux paramètres du modèle NRTL qui caractérisent le mélange étudié et doivent être optimisés sur les données expérimentales.

Travail sous matlab : on écrira une fonction `gamma=NRTL(T,x,g)`, ayant en entrée la température `T`, le vecteur de composition `x(i)` de dimension 2 et la matrice de paramètre `g(i,j)`, et en sortie, le vecteur `gamma(i)` tel que

```
gamma=NRTL(T,x,g)
gamma(1)= …
gamma(2)= …
```

Test de la fonction NRTL : Pour T = 300, x_1 = 0.3, x_2 = 0.7, g_{12} = 100, g_{21} = -200 (paramètres de test), vérifiez que :

$$\gamma_1 = 0.9799 \quad \text{et} \quad \gamma_2 = 0.9964$$

Créer le script de test jjNRTL.m contenant le test précédent.

d) Algorithme de calcul d'équilibre à composition et *P* fixée**

Il est possible de montrer à partir des équations (1), (2) et (3) que l'on doit résoudre une équation non linéaire à 1 inconnue (l'inconnue est la température T), donnée par :

$$f(T) = 1 - \sum_{i=1,2} x_i \gamma_i(T, x_i) \frac{P_i^0(T)}{P} = 0 \qquad (7)$$

Le problème est donc de trouver la température T telle que la fonction $f(T) = 0$, pour une pression P donnée et une composition liquide x_i fixée.

Une fois T déterminée et l'Eq. (7) résolue, les y_i sont alors obtenus par

$$y_i = x_i \gamma_i(T, x_i) \frac{P_i^0(T)}{P} \qquad (8)$$

Le problème consiste à résoudre une équation non-linéaire ayant une inconnue T : pour cela on va utiliser la fonction intégrée Matlab `fzero`, qui est basée sur une méthode numérique de type Newton-Raphson.

Travail sous matlab : on écrira une fonction `[y,T] = tsatur(x,g,P)`, dont le but est d'effectuer le calcul d'équilibre à T et P fixés. Elle aura en entrée la pression `P`, le vecteur de composition liquide `x` et la matrice `g` de paramètres NRTL. En sortie, cette fonction donnera la température `T` et le vecteur de composition `y`.

Il s'agit de résoudre un système non linéaire à une équation (**équation (7)**) et une inconnue (*T*). On définira une fonction `f=fsys1(T)` qui aura en entrée la température T et en sortie la valeur f de la fonction à annuler. On utilisera la fonction matlab `fzero` et des variables globales.

Dans un fichier **tsatur.m** :

- Définir la fonction `[y,T]=tsatur(x,g,P)`
- Définir les variables globales `gcom`, `xcom`, `ycom`, et `pcom`
- Sauvegarder `g, x` et `p` dans `gcom`, `xcom`, et `pcom`, respectivement
- Initialiser `T0 = 300`
- Appeler la fonction matlab `T=fzero(@fsys1,T0)`
- `y = ycom`

Dans le fichier **fsys1.m** :

- Définir la fonction `f = fsys1(T)`
- Définir les variables xcom, gcom, ycom et pcom en variables globales
- Appeler la fonction `psat(T)` pour calculer les pressions de vapeur `p0(i)` à la température T
- Appeler la fonction `NRTL(T,x,g)` pour calculer les coefficients d'activité gamma(1) et gamma(2)
- Calculer pour `i =1,2` `ycom(i) = xcom(i)*p0(i)*gamma(i)/pcom`
- Calculer `f = 1 - somme sur i des ycom(i)` (cf **équation (7)**)

Test de fsys1 pour les temperatures de 350 et 370K :

Dans un fichier `jjfsys1.m` :

```
%=======================================
clear all ; close all
global gcom
```

```
global xcom
global ycom
global pcom
disp('')
disp(' -- INPUT --')
x=[.3 .7]
P=760
g=[[0 100];[-200 0]]
%%%%%%%%%%%%%%%%%%%%
xcom=x;
pcom=P;
gcom=g;
%%%%%%%%%%%%%%%%%%%%
disp('')
disp(' -- TEST FSYS1 with 2 TEMP --')
fsys1(350)
fsys1(370)
disp('')
%=======================================
```

Vous devez obtenir :

ans = 0.4364

ans = -0.2076

Test de tsatur : pour P = 760 mmHg, x1=0.3, x2=0.7. g12 = 100, g 21 = - 200, vérifier que T = 364.8054, y1=0.4860. **Créer le script de test jjtsatur.m** contenant ce test.

Algorithme de traçage des diagrammes à P et g données :

On écrira une fonction `tracage(P,g,ifig)` contenant :

- Initier un entier `i=0`
- La boucle de `x1 = 0 à x1 = 1`, avec un pas de `0.01`, contenant :

  `i=i+1`

  `x(1)=x1` et `x(2)=1-x1`

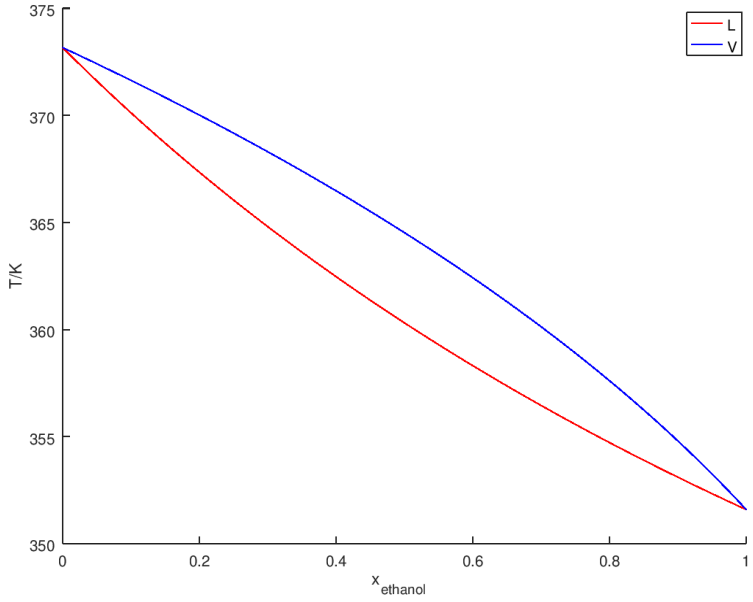
  Utiliser `[y,T]=tsatur(x, g, P)` pour calculer `y1 et T`

  `Enregistrer les valeurs de x1, y1 et T`

  dans `xcal(i), ycal(i)` et `Tcal(i)`, respectivement

- Fin de boucle
- La création de l'objet `figure(ifig); hold on`
- Le tracer, sur le même diagramme, de *T = f(x1)* (en rouge) et *T = f(y1)* (en bleu) à l'aide des vecteurs xcal, ycal, Tcal et de la fonction matlab `plot`.
- La labellisation des axes à l'aide des fonctions matlab `xlabel` et `ylabel`. Utiliser la fonction `legend` pour labelliser les deux courbes.
- La commande de sauvegarde automatique du graphique saveas(ifig,'myfig.png').

Créer le script de test **jjtracage.m** permettant de tracer T = f(x1) et T = f(y1) à partir des valeurs de g *non-optimisées* g_{12} =100 et g_{21}=-200 et pour P = 760 mmHg. Le résultat doit ressembler à ceci :



[Prof. : jjtracage0]

À CE STADE, FAIRE UN BACKUP DE VOTRE RÉPERTOIRE DE TRAVAIL

→ project-ID-YYYY-MM-DD

e) Ajustement du modèle NRTL sur les données expérimentales

Optimiser des données expérimentales à l'aide d'un modèle dépendant de plusieurs paramètres est une tâche très courante pour un ingénieur en R&D. Matlab permet de réaliser facilement cette tâche grâce à ses fonctions intégrées d'optimisation (`fminsearch`,

…), et permet de visualiser rapidement les résultats. Ici, nous allons optimiser les paramètres g12 et g21 du modèle NRTL sur les données expérimentales (cf **Tableau 1)**.

Tableau 1 : Données d'équilibres liquide-vapeur du mélange éthanol-eau à P=760mmHg.

x éthanol = x_1	y éthanol = y_1	T en Kelvin
0	0	373.1
0.05	0.3372	363.1
0.1	0.4521	359.0
0.15	0.5056	357.1
0.2	0.5359	356.0
0.25	0.5589	355.2
0.3	0.5794	354.
0.35	0.5987	354.1
0.4	0.6177	353.6
0.45	0.6371	353.2
0.5	0.6558	352.9
0.55	0.6765	352.5
0.6	0.6986	352.2
0.65	0.725	352
0.75	0.784	351.5
0.8	0.8167	351.4
0.85	0.8591	351.3
0.9	0.8959	351.3
0.95	0.9474	351.3
1	1	351.4

Télécharger le fichier de données sur ce lien. Pour optimiser les paramètres, nous utiliserons la *méthode des moindres carrés* avec la *fonction-objectif* fobj(par) définie par l'**équation (9)** :

$$fobj = \sum_{k=1}^{npointsexp} \left(y_{1cal,k} - y_{1,\exp,k}\right)^2 + \left(T_{cal,k} - T_{\exp,k}\right)^2 \quad (9)$$

Modifications de la fonction tracage :

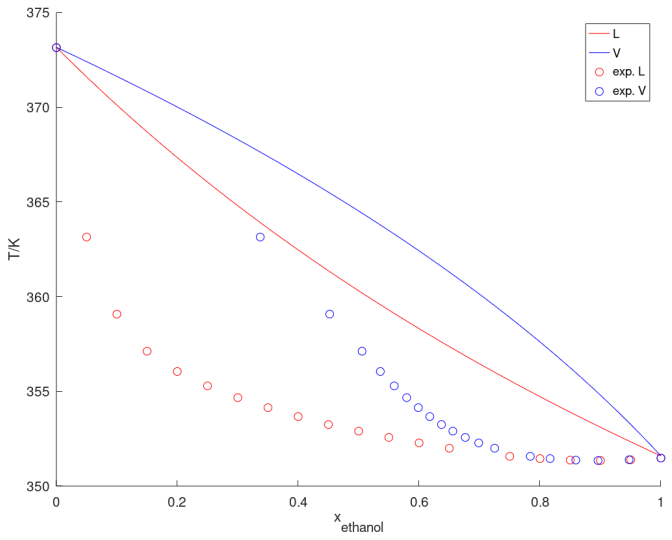
1. Télécharger le fichier de `ethanol_eau.dat` sur **ce lien**. Il contient les données du **Tableau 1**.

2. Modifier la fonction `tracage(P,g,ifig)` et **jjtracage.m** afin de tracer sur le même graphique les courbes *T = f(x1), T = f(y1)* calculées et les données expérimentales :

- Charger les données à partir du fichier `ethanol_eau.dat` dans une matrice `DAT`. Pour cela, utiliser la fonction matlab `load`.

IMPORTANT : DAT devra être définie en variable globale dans jjtracage.m et dans tracage.m.

Le résultat doit ressembler à ceci :



[Prof. : jjtracage1]

À CE STADE, FAIRE UN BACKUP DE VOTRE RÉPERTOIRE DE TRAVAIL

→ project-ID-YYYY-MM-DD

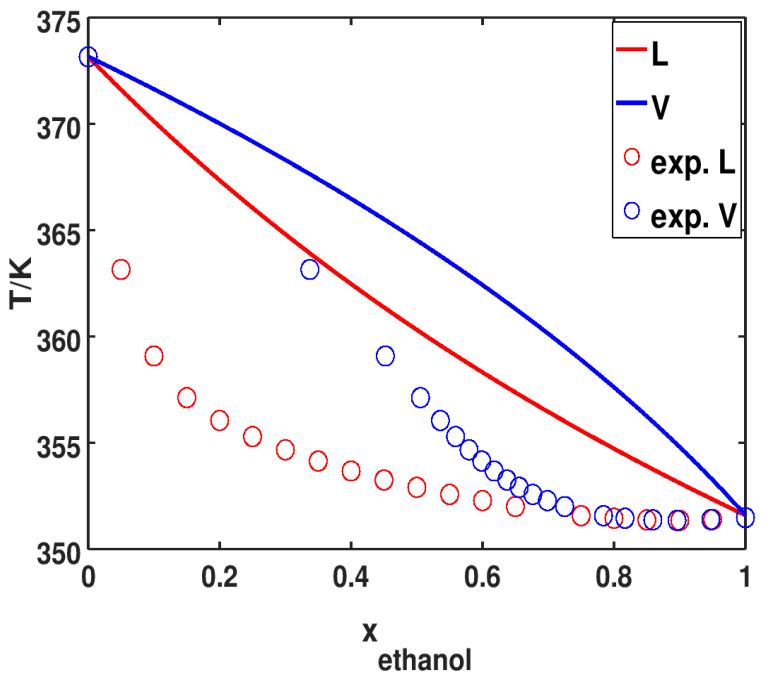
3. Modifier la fonction `tracage(P,g,ifig)` afin de rendre le graphique plus lisible :

- Epaissir les courbes avec plot(...,'linewidth',2)
- Augmenter la taille des points (ronds) expérimentaux avec plot(...,'markersize',12)
- Modifier la police des labels sur les axes avec

myfontsize=12;xlabel(...,'fontname','helvetica','fontsize', myfontsize,'fontweight','bold')

- Modifier la police et l'épaisseur des axes avec set(gca ,'fontname','helvetica','fontsize', myfontsize-2 ,'fontweight','bold','linewidth',1)
- Modifier la police des labels avec hleg=legend('L','V','exp. L','exp. V'); set(hleg,'fontname','helvetica','fontsize', myfontsize ,'fontweight','bold'

Le résultat doit ressembler à ceci :



[Prof. : jjtracage2]

À CE STADE :

- **Faites une copie du fichier myfig.png en x-T-nonopt.png**
- **FAIRE UN BACKUP DE VOTRE RÉPERTOIRE DE TRAVAIL**

→ project-ID-YYYY-MM-DD

On veut maintenant obtenir une figure similaire a x-T-nonopt.png mais avec les courbes théoriques ajustées dans les points expérimentaux. On va pour cela créer un nouveau script à partir de **jjtracage.m**, qui va contenir une étape supplémentaire d'optimisation du modèle NRTL.

Algorithme d'optimisation et de tracage :

1. Copier le script **jjtracage.m** en **prin.m** et modifier ce dernier de façon à ce qu'il contienne les étapes suivantes :

- Définir un vecteur `par0`, de dimension 2 qui contiendra les valeurs initiales de g12 et g21.
- Faire appel à la fonction `parnew = fminsearch(@fobj, par0)`, où `fobj` est la fonction-objectif à optimiser, et `parnew` le jeu de paramètres optimisés.
- Reconstruire la matrice `g` en utilisant les valeurs de `parnew`. Afficher le résultat à l'aide de :

```
disp('')
disp('(@prin.m ) optimized NRTL parameters:')
disp('')
g
```

- L'appel de la fonction `tracage(P,g,1)`, à la fin du script, doit permettre d'obtenir la figure avec les courbes optimisées (voir la fin de cet encadré).

2. Écrire la fonction `f=fobj(par)` dans laquelle on aura :

- La matrice `DAT` sera mise en variable globale de façon à accéder aux valeurs des données expérimentales qui sont chargées dans **prin.m**.
- Utiliser la fonction `size(DAT)` pour déterminer le nombre de points expérimentaux et l'assigner dans la variable nexp.
- Reconstruire la matrice `g` en utilisant les valeurs de `parnew`.
- Fixer la pression `P` à une valeur de `760` mmHG
- Initier `f=0`
- Créer la boucle de `k = 1` à `nexp`

  Fixer la composition `x(1) = DAT(k,1)` et `x(2) = 1-x(1)`

  Calculer y et T en faisant appel à la fonction

  `[y,T] = tsatur(x,g,P)`

  `f = f + (T - DAT(k,3) )^2 + (y(1)-DAT(k,2))^2`

  selon l'**équation (9)**

- Fin de boucle

Test de fobj pour les parametres NRTL initiaux :

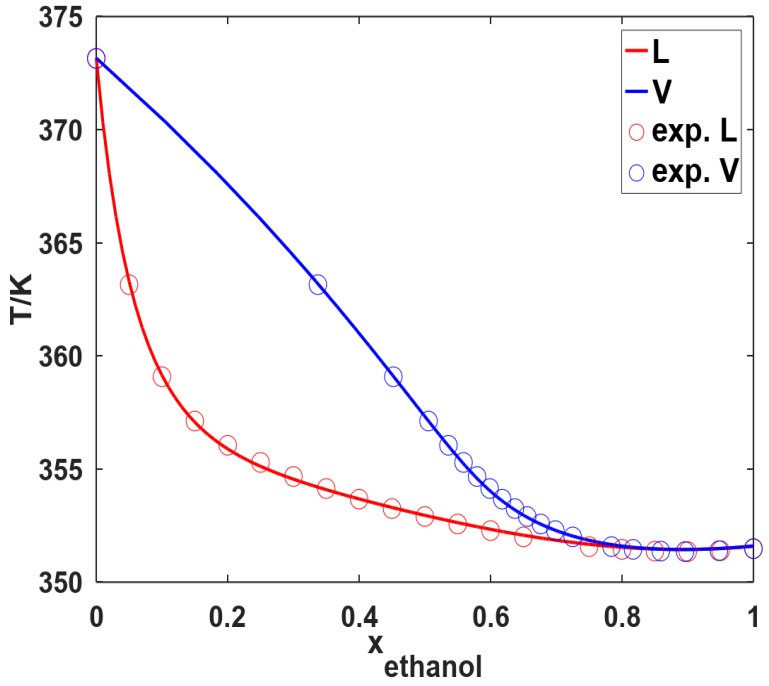
Dans un fichier **jjfobj.m** :

- Definir la variable `DAT` en global
- Assigner les valeurs `[100 -200]` au vecteur `par`
- Charger le fichier de données expérimentales les placer dans la matrice `dat`
- évaluer `fobj(par)`

La valeur retournée doit être de 1109.3

Exécution du script prin.m :

Noter les valeurs de la matrice g optimisée. Le résultat doit ressembler à ceci :



[Prof. : prin]

À CE STADE :

- **Faites une copie du fichier myfig.png en X-T-opt.png**
- **FAIRE UN BACKUP DE VOTRE RÉPERTOIRE DE TRAVAIL**

→ project-ID-YYYY-MM-DD

2. Deuxième partie du Projet: Dimensionnement et optimisation de la colonne, en utilisant la méthode de McCabe et Thiele

La **figure 4** décrit le système à étudier et les variables du problème.

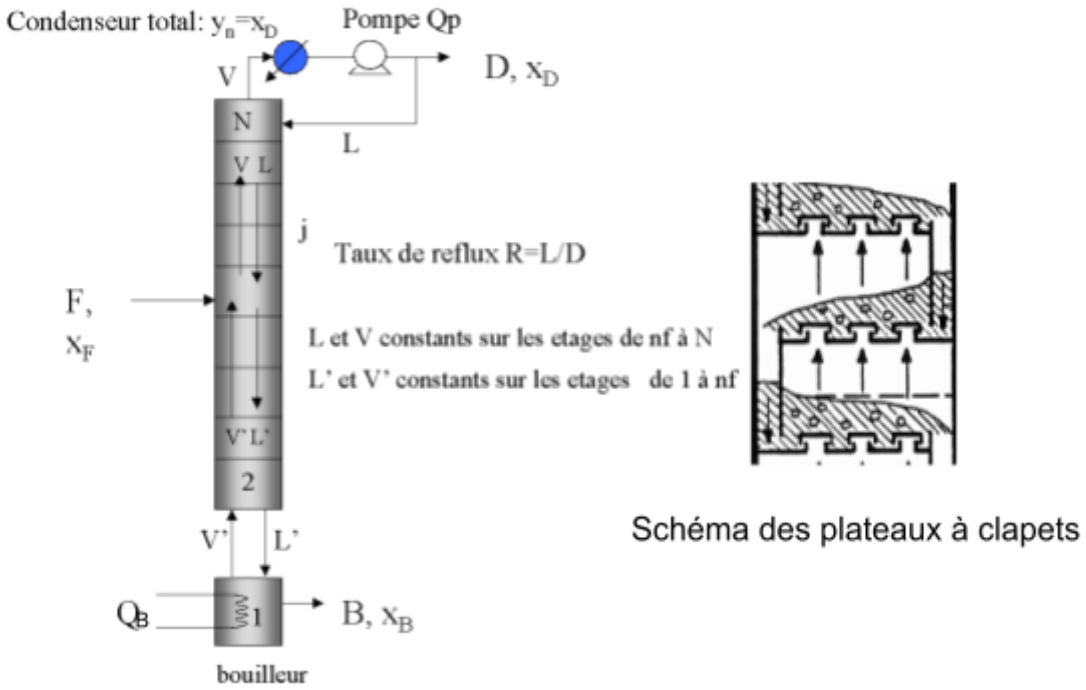


Fig. 4 : schéma de la colonne de distillation.

Les variables et données du problème sont :

F : débit molaire total en entrée (en kmol/h). On suppose que *F* est liquide à saturation, et L' = L + F.

x_F: fraction molaire en éthanol dans le débit d'entrée *D*. On fixe **x_f = 0.03**.

D: débit de distillat (en kmol/h). *D* est tel que *D* = 295.43 kmol/h (correspondant aux **100 000 tonnes d'éthanol** produites par an).

x_D: fraction molaire en éthanol dans le débit de distillat *D*. On fixe x_D=0.84.

B: débit en sortie de bouilleur (en kmol/h)

x_B: fraction molaire en éthanol dans le débit de sortie du bouilleur B. On fixe **x_B=0.0001**.

x_k, y_k : fractions molaires en éthanol du liquide et de la vapeur en équilibre à l'étage *k*.

T_k : température à l'étage *k*. T_k et y_k dépendent de x_k via l'équilibre liquide vapeur.

N : nombre d'étages de la colonne.

Q_B : puissance fournie au bouilleur (grâce à un échangeur de chaleur avec de la vapeur surchauffée)

P : pression constante tout le long de la colonne : ***P* =760 mmHg**.

R : taux de reflux. ***R = L/D***

L : débit de liquide renvoyé dans l'étage N, avec $L = RD$. Le débit L est supposé constant tout le long de la colonne au-dessus de l'alimentation F.

V : débit de vapeur sortant de l'étage N. On sait que : $V = L + D$. Le débit V est supposé constant tout le long de la colonne au-dessus l'alimentation F.

L' : débit de liquide rentrant dans le bouilleur, avec $L' = L + F$. Le débit L' est supposé constant tout le long de la colonne au-dessous l'alimentation F.

V' : débit de vapeur sortant du bouilleur avec $V' = V$. $V'=V$ est supposé constant tout le long de la colonne et $L' = V + B = F + L$

Q_p : puissance fournie par la pompe en sortie de condenseur.

Q_C : Puissance récupérable au condenseur.

Il est possible de montrer, à x_D, x_B, P et D fixés, que **l'unique degré de liberté restant du problème est le taux de reflux *R***. Le but de ce projet est alors de calculer le coût de revient annuel de la colonne, pour un taux de reflux R donné, puis de trouver la valeur optimale de R (voir **figure 5**). *Rmin* est le taux de reflux R minimal en dessous duquel la colonne ne peut pas fonctionner.

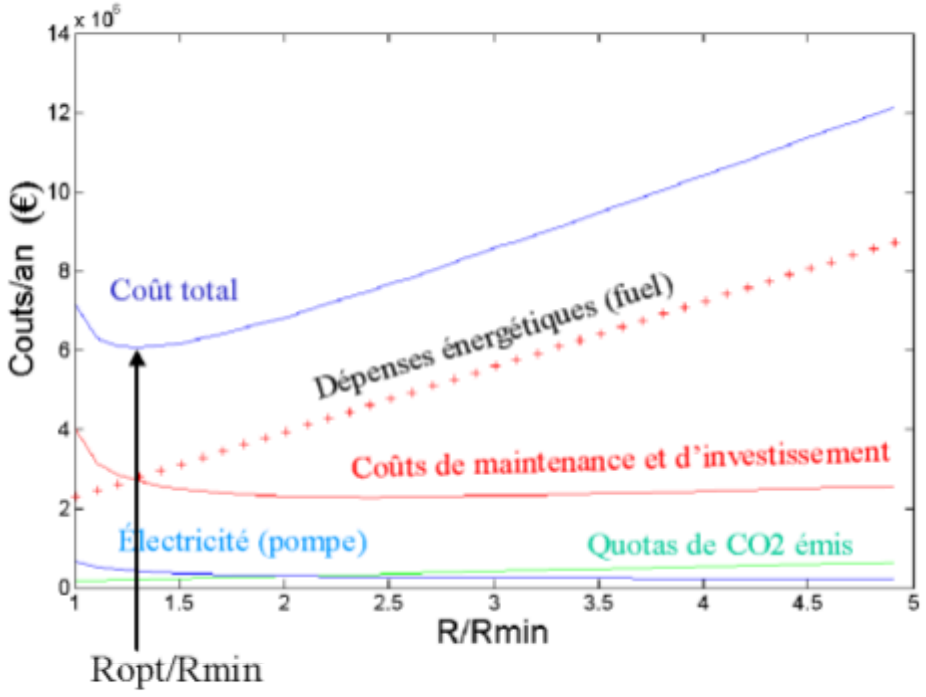


Fig.5 : Différents coûts en fonction du rapport R/Rmin et valeur optimale de R

2.1. Mise en équations du problème

- Bilan total en nombre de moles sur la colonne : débit total molaire entrant = somme des débits molaires sortant

$$F = B + D \qquad (10)$$

- Bilan en nombre de mole d'éthanol sur la colonne : débit molaire en éthanol entrant = débits molaires en éthanol sortants

$$Fx_F = Bx_B + Dx_D \qquad (11)$$

- Bilan molaire sur le condenseur : $V = L + D$

- Bilan molaire sur le bouilleur : $L' = V + B.$

Soit le taux de reflux R avec $R = L/D$ (12)

- Bilan molaire en éthanol sur la partie haute de la colonne, sur les étages au-dessus de l'alimentation (voir Fig. 6 ci-dessous). On montre que

$$y_k = \frac{L}{V}x_{k+1} + \frac{D}{V}x_D \qquad (13)$$

y_k : fraction molaire en éthanol de la phase vapeur à l'étage k.

x_{k+1} : fraction molaire en éthanol de la phase liquide à l'étage k+1.

x_D : fraction molaire en éthanol du distillat (sortie tête de colonne et condenseur)

D'après les Eq. (12) et (13),

$$y_k = \frac{R}{R+1}x_{k+1} + \frac{1}{1+R}x_D \qquad (14)$$

L'équation (14) définit **la droite opératoire supérieure** reliant x_{k+1} à y_k .

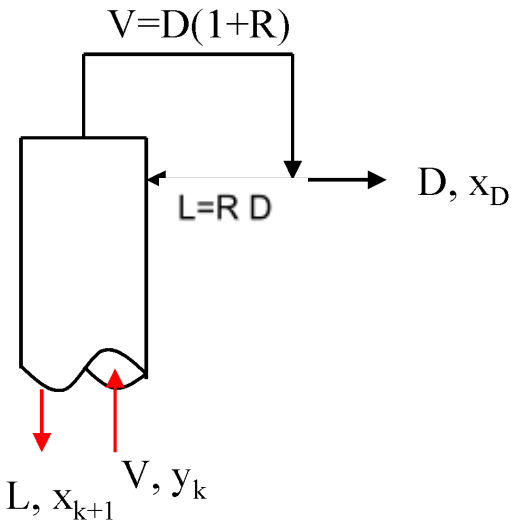


Figure 6 : Partie haute de la colonne, au-dessus de l'étage d'alimentation. Bilan de matière sur la partie encadrée (moles en entrée = moles en sortie). On en déduit les Eq. (13) et (14).

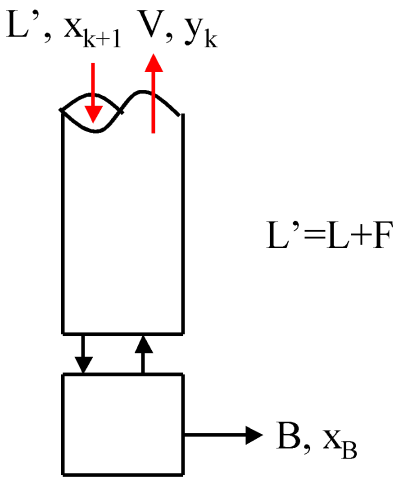


Fig. 7 : Partie basse de la colonne, au-dessous de l'étage d'alimentation.

- Bilan molaire en éthanol sur la partie basse de la colonne, sur les étages au-dessous de l'alimentation *F* (voir Fig. 6 ci-dessous). On montre que

$$y_k = \frac{L'}{V} x_{k+1} - \frac{B}{V} x_B \qquad (15)$$

avec

y_k : fraction molaire en éthanol de la phase vapeur à l'étage k.

x_{k+1} : fraction molaire en éthanol de la phase liquide à l'étage k+1.

x_B : fraction molaire en éthanol en sortie de bouilleur.

L'équation (15) constitue **la droite opératoire inférieure**. Supposons que l'on fixe les fractions molaires x_D, x_B , le taux de reflux $R = L/D$, et le débit D. On peut alors en déduire les autres valeurs F, B, L, V et L', à partir des équations de bilan de masse établies auparavant (Eq. (10) , (11) et (12)):

Flux de matière :

$$F = D\frac{x_D - x_B}{x_F - x_B} \quad (16)$$

$$B = F - D = D\frac{x_D - x_F}{x_F - x_B} \quad (17)$$

$$L = RD \quad (18)$$

$$V = D(1+R) \quad (19)$$

$$L' = L + F \quad (20)$$

Pour ce projet, on fixe les fractions molaires en éthanol :

$$x_D = 0.84 \text{ , } x_B = 0.0001 \text{ et } x_F = 0.03 \text{ .}$$

En supposant que l'usine fonctionne tous les jours et 24h/24, pour produire 100000 tonnes d'éthanol par an, il faut que

$$x_D D = \frac{100000 \times 10^6}{46} = 2.173 \times 10^9 \text{ mol/an}$$

donc le dépit D en kmol/h vaut

$$D = \frac{100000 \times 10^3}{46 \times 0.84 \times 365 \times 24} = 295.43 \text{ kmol/h}$$

(masse molaire de l'éthanol = 46g/mol).

2.2. Méthode de McCabe et Thiele

La détermination du nombre d'étages de la colonne se fait de manière graphique grâce aux droites opératoires et au diagramme d'équilibre $y = f(x)$ du mélange binaire éthanol-eau (déterminé auparavant dans la partie I du projet). Il s'agit de la *méthode de McCabe et Thiele*. Matlab va nous permettre d'automatiser cette méthode et ainsi d'avoir des résultats très précis et instantanés. **Matlab va nous permettre également de minimiser les coûts**

de la colonne et les dépenses énergétiques en faisant varier le paramètre R appelé taux de reflux.

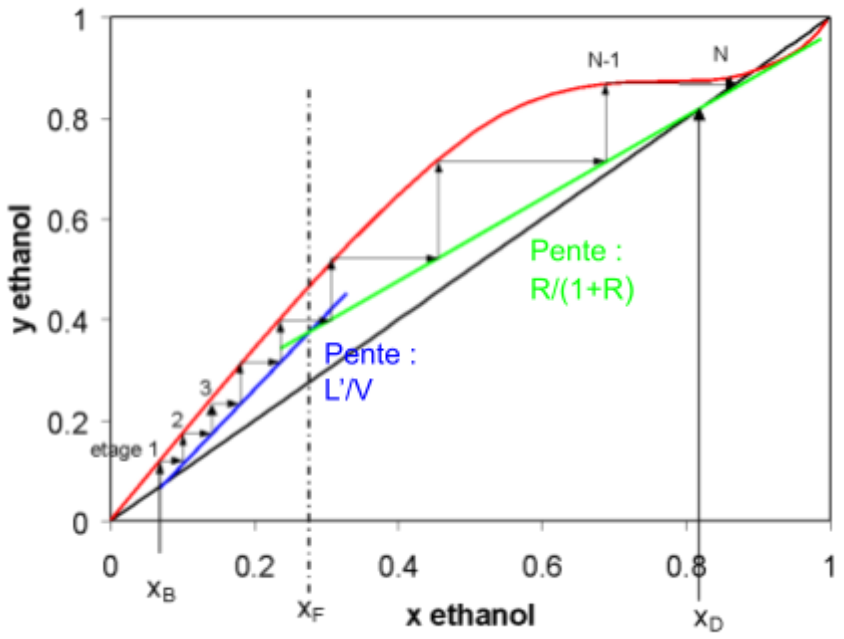


Figure. 8 : Équilibre liquide-vapeur du mélange éthanol-eau. Méthode de McCabe-Thiele. Courbe rouge : courbe y = f(x) d'équilibre liquide vapeur. Droite bleue : droite opératoire inférieure. Droite verte : droite opératoire supérieure. x_B : composition en éthanol au bouilleur (1er étage). xD : composition du distillat. x_F :composition de l'alimentation liquide saturée. N : nombre d'étages. Les flèches indiquent les itérations successives pour déterminer les étages.

a) Description de la méthode graphique

1) On place tout d'abord x_B et x_D sur la droite y = x.
2) On trace la droite verticale passant par x_F. Pour une valeur de R choisie > Rmin (voir comment déterminer Rmin dans la section .3 b)), on trace les deux droites opératoires passant par les points (x_D, x_D), (x_B, x_B), et de pentes respectives L/V = R/(1+R) et L'/V (voir la **figure 8**). les trois droites doivent se couper au même point.
3) On trace la courbe d'équilibre y = f(x).
4) Partant de x_B, on détermine la composition y_B de la vapeur en équilibre au bouilleur (y_B = f(x_B), obtenue avec la fonction `[y,T]=tsatur(x, g, P)`.). Le bouilleur correspond à l'étage 1 de la colonne. La composition du liquide x_2 à l'étage 2 est donnée par la droite opératoire inférieure. La composition de la vapeur y_2 à l'étage 2 est donnée par la courbe d'équilibre y_2 = f(x_2).
5) On itère de cette manière jusqu'à ce que $x_i > x_F$. On obtient alors le nombre d'étage en dessous de l'alimentation F.
6) On utilise la droite opératoire supérieure pour obtenir x_{i+1} à partir de y_i. Puis y_{i+1} est obtenu par la courbe d'équilibre y_{i+1}=f(x_{i+1}).

7) On itère jusqu'à ce que $x_i > x_D$. On obtient alors le nombre total d'étage N avec $x_N > x_D$.

NB : Pour le dernier étage, x_N n'est pas exactement égale à la consigne x_D: cela est tout à fait normal car le nombre d'étage est un nombre entier et il est impossible d'obtenir exactement x_D.

b) Détermination du taux de reflux minimal R_{min}

D'après les **figures 8** et **9**, il existe un taux de reflux minimal correspondant à un nombre d'étages nécessaires infinis pour respecter les consignes de composition x_B et x_D en sortie de colonne. Il s'agit donc dans un premier temps de déterminer R_{min}. Le taux R_{min} est obtenu lorsque les droites opératoires se croisent au point (x_F, y_F) de la courbe d'équilibre liquide vapeur y = f(x) (voir la **figure 9**). On obtient donc R_{min} selon la formule

$$R_{\min} = \frac{x_D - y_F}{y_F - x_F} \qquad (22)$$

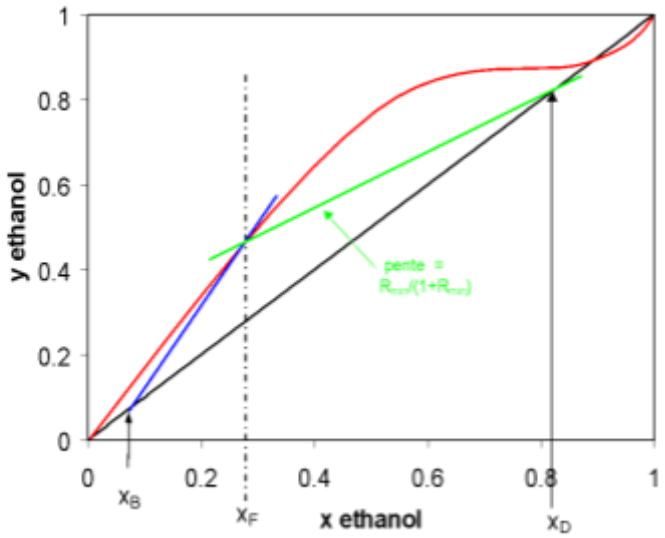


Figure 9 : Détermination du taux de reflux minimal R_{min}

c) Détermination des dimensions réelles de la colonne

Une fois le nombre de plateaux (ou étages) théoriques déterminé, il est possible de déterminer les dimensions réelles de la colonne (hauteur et diamètre intérieur). On définit alors une hauteur équivalente de plateaux théoriques **HETP**, telle que la hauteur réelle **H** de la colonne vaut

$$\mathbf{H = HETP\ N_{dec}} \qquad (23)$$

où N_{dec} le *nombre décimal d'étages.* Cette notion sera précisée lors de l'écriture de l'algorithme. Ici, on prendra des plateaux ayant une **HETP de 0.7 mètres**. Les colonnes ne doivent pas être trop hautes **: H < 70m**, sinon la colonne n'est pas concevable.

Le diamètre optimal $\mathbf{Diam_{opt}}$ est déterminé selon les propriétés hydrodynamiques de la colonne (étude de mécanique des fluides), afin d'éviter les phénomènes d'engorgement et permettre une efficacité optimale. Le diamètre dépend de la vitesse de circulation de la vapeur dans la colonne et donc du débit V. Des études expérimentales sur pilote montrent que dans notre cas

$$\mathrm{Diam}_{\mathrm{opt}} = 0.1\sqrt{V} \qquad (24)$$

où V est le débit de vapeur exprimé en kmol/heure. Le diamètre varie en général entre 2 et 6m.

d) Calcul des coûts annuels

Une fois le nombre de plateaux (ou étages) déterminés pour un taux de reflux *R > Rmin* donné, nous devons calculer les coûts d'investissement et de fonctionnement d'une telle colonne. **Ensuite, nous minimiserons ces coûts en faisant varier R**.

Le coût total en euros par an, C_{tot} , est estimé selon la formule suivante :

$$C_{tot} = C_{inv} + C_{ener} + C_{elec} + C_{CO2} \qquad (25)$$

avec

C_{inv} : prix d'achat et de montage de la colonne divisé par 10 (amortissement sur 10 ans) plus coût de maintenance. On suppose que le coût C_{inv} est proportionnel au volume de la colonne.

On estime que

$$C_{inv} = 20000\left(Diam_{opt}\right)^2 H \qquad (26)$$

où H et $Diam_{opt}$ sont en mètres et C_{inv} en euros/an.

C_{elec} : Consommation en électricité pour faire fonctionner les pompes. Ces pompes permettent de compenser les pertes de charges dans la colonne. En appliquant la loi de Poiseuille, on trouve la perte de charge, et le coût C_{elec} est estimé à

$$C_{elec} = 2.12\mathrm{V}^2 \frac{\mathrm{H}}{(Diam_{opt})^4} \qquad (27)$$

où H et $Diam_{opt}$ sont en mètres, V est le débit de vapeur en kmol/h, C_{elec} est en euros/an.

C_{ener} : dépenses énergétiques de chauffage de la colonne. La colonne est chauffée au bouilleur via un circuit de vapeur d'eau, laquelle est produite en brûlant du gaz naturel. Compte tenu du prix du gaz naturel, C_{ener} est évalué à

$$C_{ener} = 0.15Q \qquad (28)$$

où C_{ener} est en euros/ an et Q est la puissance nécessaire apportée sous forme de chaleur à la colonne (énergie entrante – énergie sortante), exprimée ici en kJ/h.

Hypothèses simplificatrices : on suppose que l'enthalpie de vaporisation ne dépend pas de la composition du mélange ni de la température.

$$Q = Q_B = V\ \Delta H_{vap} \qquad (29)$$

V est le débit de vapeur en kmol/h, et ΔH_{vap} = 40000 kJ/kmol.

C_{CO2} : Le prix de la tonne de CO2 émise à l'atmosphère est actuellement de **22 euros par tonne de CO2 émises**. Sachant que la masse de CO_2 produite pour obtenir 1kJ par combustion du gaz naturel est de 57 milligrammes, la masse (en tonnes) de CO_2 émis par an est de

$$m_{CO2} = 57 \times 10^{-9} Q \times 24 \times 365 \qquad (30)$$

La somme à payer par an en euros est donc de

$$C_{CO2} = 22 m_{CO2} \qquad (31)$$

Travail sous Matlab : Écrire une fonction `ctot=calc_cout(alpha)` ayant en entrée la variable `alpha` telle que alpha = R/Rmin, et en sortie la valeur du coût total `ctot`.

Ordre des opérations dans le fichier **calc_cout.m** :

- Mettre gcom en global
- Fixer les variables P, xB, xD, D, xF (valeur de l'énoncé).
- En déduire les débits F et B en kmol/h (Équations (16) et (17)),
- `Fixer la valeur des paramètres NRTL avec g=gcom`
- Initialiser le vecteur de composition `x` à partir de `xF`
- Faire appel à la fonction tsatur pour calculer la fraction en éthanol `yF` et la température `T`
- En déduire la valeur de Rmin (Eq. (22))
- Calculer R = alpha * Rmin
- En déduire L, Lprim et V en kmol/h (Eq. (18), (19) et (20))

```
% ====================================
% Droite opératoire inférieure (Fig. 8)
% ====================================
```

- Initier `k=0`
- Initier `xk = xB`
- Boucle while xk<xF

  Vecteur de composition pour tsatur : `x(1)=xk, x(2)=1-xk`

  Trouver yk en équilibre avec xk en appelant la fonction

  `[y, T] = tsatur(x, g, P)` tel que `yk=y(1)`

  Incrémenter le nombre d'étages `k = k + 1`

  Sauvegarder les valeurs de la température, de `x(1)` et de `y(1)`

  dans des tableaux `xstage(k),ystage(k) et Tstage(k)`

  Utiliser la droite opératoire inférieure :

  `xk = V/Lprim*(yk +B/V*xB)`

- fin de la boucle

- Nombre d'étages sous l'alimentation : Nlow = k

```
% ====================================
% Droite opératoire inférieure (Fig. 8)
% ====================================
```

- Réinitialiser xk en utilisant l'autre droite d'opération : `xk = V/L*(yk -D/V*xD)`
- Boucle `while  xk<xD`

    Vecteur de composition pour tsatur : `x(1)=xk,  x(2)=1-xk`

    Trouver yk en équilibre avec xk en appelant la fonction

    `[y,T] = tsatur(x, g, P)  tel que  yk=y(1)`

    Incrémenter le nombre d'étages `k = k + 1`

    Sauvegarder les valeurs de la température `T,`

    de `x(1) et de  y(1)` dans des tableaux `xstage(k),`

    `ystage(k) et Tstage(k)`

    Utiliser la droite opératoire supérieure :

    `xk = V/L*(yk -D/V*xD)`
- Fin de la boucle

- Nombre entier total d'étage : Ntotal = k
- Nombre d'étages au-dessus de l'alimentation : NSup = k-Nlow
- **Pour le calcul des coûts, il vaut mieux utiliser un nombre décimal d'étages : astuce pour lisser la courbe de coût total = f(alpha) :**

`Ndec = Ntotal + (xD-xstage(k))/(xk-xstage(k))`

- Dimensions de la colonne (utiliser Ndec) :
- Calculer la hauteur H : Eq. (23)
- Calculer le diamètre diamopt : Eq. (24)
- Calculer des coûts :
    - Coût d'investissement et maintenance : Eq. (27)
    - Coût de dépenses en électricité: Eq. (27)
    - Calculer la puissance énergétique de chauffage en kJoules par heure : Eq. (29)
    - Coût dû à cette puissance : Eq. (28)
    - Masse de CO2 émise par an : Eq. (30)
    - Coût de quotas de CO2 émis : Eq. (31)
    - Cout total ctot = … Eq. (25)

Dans le fichier **calc_cout.m**, on mettra en variable globale toutes les variables que l'on souhaite afficher à partir du fichier principal.

!! attention : la fonction `calc_cout` plante pour des valeurs de alpha inférieur ou égales à 1 (boucles while infinies) !!

Créer le script de test jjcalc_cout.m appelant `calc_cout` pour une valeure de alpha=2.

- Le calcul doit converger et le coût total affiché doit être de 1.7526e+07
- Observez ce qui se passe pour alpha = 0.5. (Tapez contrôle C pour arrêter le programme).

Copier **prin.m** en **Princolonne.m** et ajouter les éléments suivants :

- Déclarer les variables globales communes à **Princolonne.m** et **calc_cout.m**
- Après la commande tracage(...) boucler sur `alpha=1.001:0.01:3` et calculer tous les coûts pour chaque valeur de alpha
- Tracer sur le même graphe les différents coûts en fonction de alpha (Faire un graph similaire à la **figure 5**). Utilisez le mot clef 'legend 'pour identifier les différents coûts sur le graph.
- Faire appel à `fminsearch(@calc_cout,alpha0)` pour déterminer la valeur de alpha qui minimise le coût total.
- Mettre en place dans la fonction calc_cout une astuce avec un 'if' pour éviter que le programme plante si la fonction fminsearch teste une valeur de alpha inférieure ou égale à 1.
- Prendre comme valeur initiale alpha0=2.

BONUS : Tracer les profiles de température et de concentration en fonction des étages, correspondant à la valeur de alpha optimale (on utilisera les tableaux `Tstage(k),xstage(k)` et `ystage(k)` et un vecteur `ik = [1:k]` => `plot(ik,Tstage)`