

Fiche-guide de TD sur la trigénération par micro-turbine à gaz et cycle à absorption

1) Objectifs du TD

L'objectif de ce TD est de montrer comment, moyennant un travail marginalement réduit, il est possible d'ajouter des composants multi-fonctionnels personnalisés, relativement compliqués sur le plan thermodynamique, à ceux plus simples prédéfinis dans le noyau de Thermoptim, que l'on peut qualifier de mono-fonctionnels étant donné qu'ils ne mettent en jeu qu'au plus une seule forme d'énergie (soit mécanique, soit thermique). En procédant de la sorte, il est possible de modéliser facilement des installations complexes mettant en jeu de nombreux composants, comme c'est le cas d'une centrale de trigénération.

2) Références

GICQUEL R., Systèmes Energétiques, Tomes 1 et 2, Presses de l'Ecole des Mines de Paris, février et novembre 2001.

ASHRAE, Fundamentals Handbook (SI), Thermodynamics and Refrigeration Cycles, 2001.

3) TD principal

3.1 Enoncé

Pour étendre le noyau de Thermoptim, on lui ajoute des modules externes écrits dans le langage Java, qui en définissent à la fois les équations et l'interface graphique. Ces éléments complémentaires sont chargés dynamiquement lors du lancement du progiciel, et apparaissent dans ses écrans de manière transparente pour l'utilisateur, comme s'ils en faisaient partie intégrante. La mise en place du mécanisme d'extension de Thermoptim par ajout de classes externes a conduit à introduire deux distinctions sémantiques qui ont leur importance : d'une part celle qui vient d'être définie entre les composants mono et multi-fonctionnels, et d'autre part celle qui est liée à une notion nouvelle dans le progiciel, celle de thermocoupleur. Les thermocoupleurs sont destinés à compléter les échangeurs de chaleur classiques en permettant à des composants autres que des transfos "échange" de se connecter à une ou plusieurs transfos "échange" pour représenter un couplage thermique.

Un tel mécanisme présente un grand intérêt, car il permet de représenter de nombreux couplages thermiques qui ne constituent pas pour autant un échange de chaleur classique, comme par exemple le refroidissement des parois de la chambre de combustion d'un moteur alternatif, une compression refroidie, et surtout, comme on le verra plus loin, des apports ou des extractions de chaleur au niveau des composants externes multi-fonctionnels.

On peut ainsi continuer à bénéficier de l'ensemble de l'environnement de Thermoptim, à savoir tous les composants disponibles ainsi que l'éditeur de schémas qui permet de décrire très aisément la structure interne du système étudié. Non seulement une telle manière de faire simplifie notablement la démarche de modélisation et facilite ultérieurement l'utilisation et la maintenance du modèle, mais surtout elle sécurise sa construction en automatisant l'établissement des couplages entre les différents éléments qui le

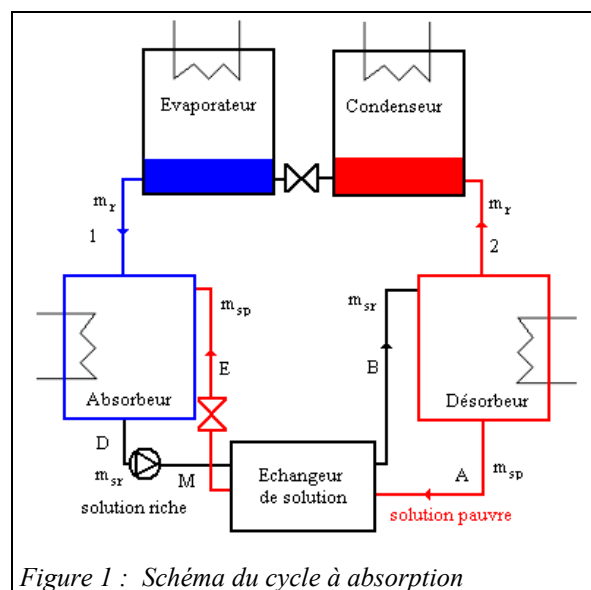


Figure 1 : Schéma du cycle à absorption

composent et en garantissant leur cohérence. Ce point est d'autant plus important que le système étudié comprend un nombre de composants élevé.

Une micro-turbine à gaz est une turbine à gaz de petite puissance (quelques dizaines de kW), fonctionnant généralement avec un faible rapport de compression et avec un régénérateur pour en améliorer les performances.

Nous nous intéressons ici à une installation de trigénération où les gaz sortant d'une micro-turbine de 125 kW_e sont utilisés d'une part pour fournir la chaleur nécessaire au désorbeur d'une machine à absorption LiBr-H₂O, et d'autre part pour produire 0,5 kg/s d'eau chaude sanitaire à 80 °C environ. La turbine aspire 0,78 kg/s d'air qui est comprimé à 5 bars, puis passe dans un régénérateur avant d'être porté à 950 °C dans la chambre de combustion brûlant du gaz naturel. Les gaz sont détendus jusqu'à la température de 650 °C puis traversent successivement le régénérateur, le désorbeur et l'échangeur de cogénération. Un compresseur de gaz est nécessaire pour rehausser la pression du gaz naturel du réseau GDF. Il s'agit d'une variante de l'exemple de cogénération 8.5.1 de [1].

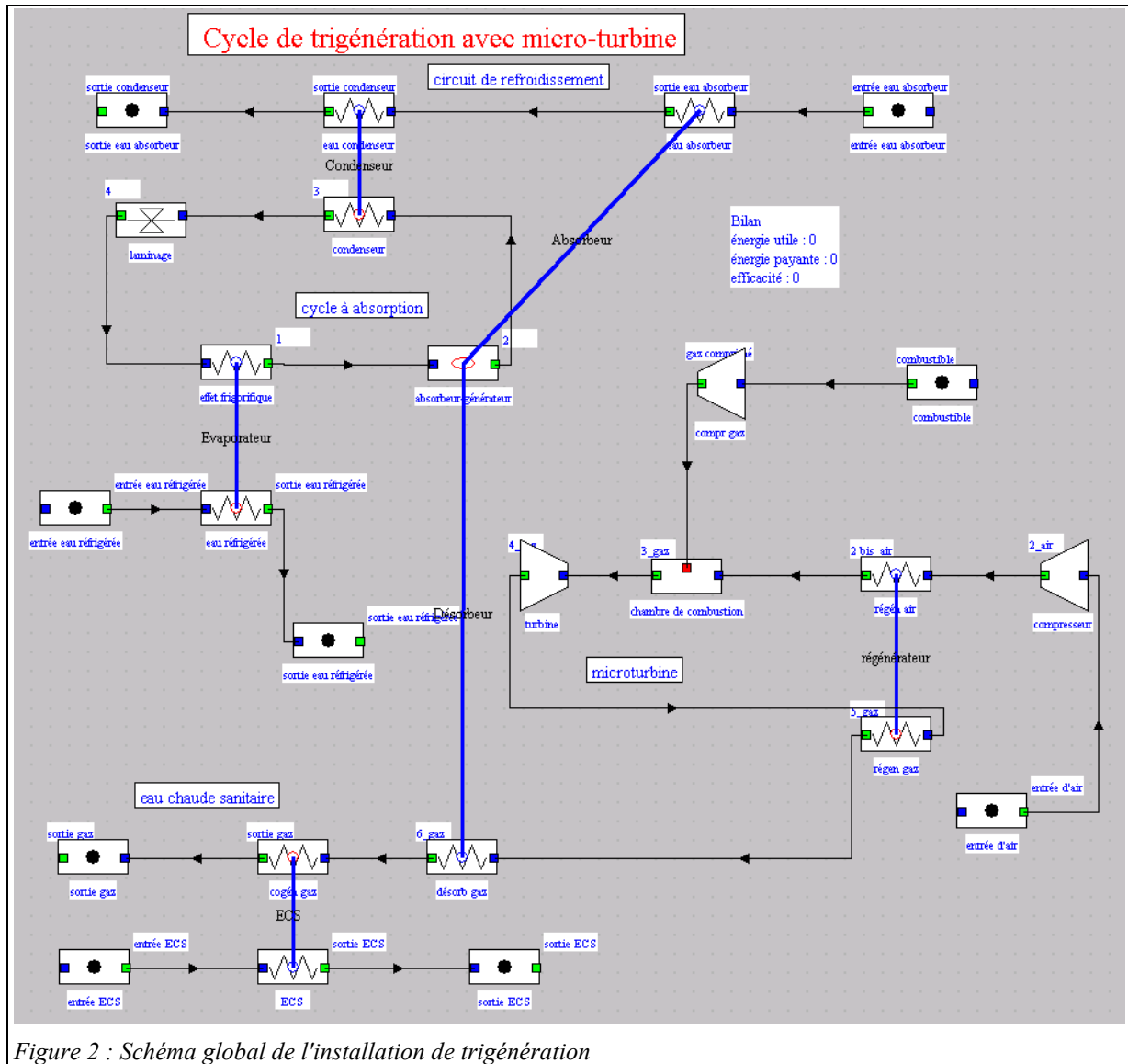
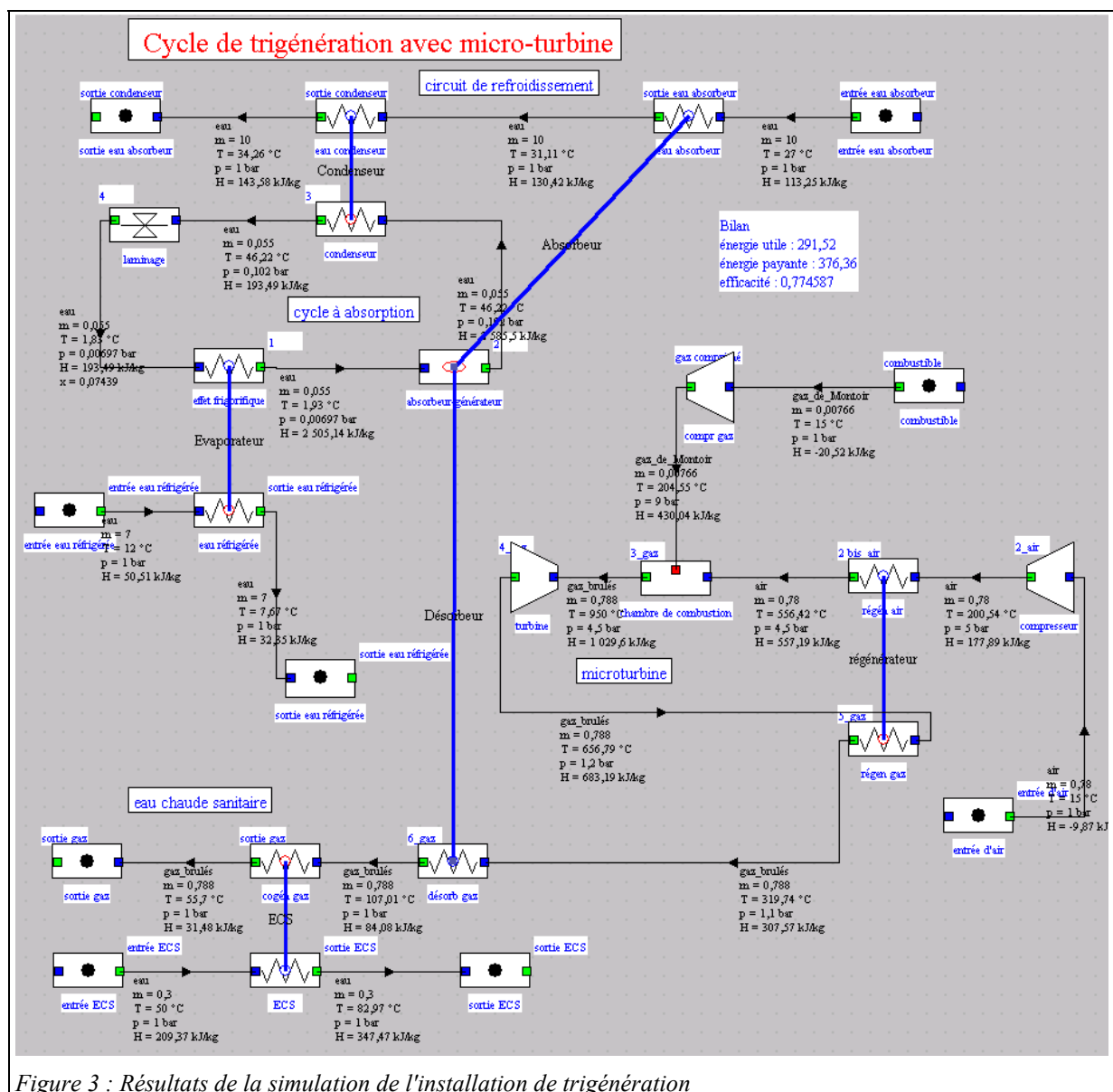


Figure 2 : Schéma global de l'installation de trigénération

La modélisation de cette installation fait appel à une trentaine de composants, ce qui représente plusieurs centaines d'équations couplées. Tous ces composants étant disponibles dans le noyau de Thermoptim, à l'exception du sous-système absorbeur-échangeur de solution-désorbeur du cycle de réfrigération à absorption (les trois éléments inférieurs du schéma de la figure 1), qui vient remplacer le compresseur d'un cycle à compression de vapeur, on comprend l'intérêt que présente le mécanisme des classes externes dans un cas comme celui-là : en ajoutant un composant spécialisé pour représenter le module manquant, le travail à effectuer est beaucoup plus réduit que s'il fallait écrire un programme pour modéliser tout le cycle.

Il suffit donc de créer un composant externe permettant de représenter le module inexistant, qui d'une part met en jeu le couple LiBr-H₂O, dont les propriétés peuvent être modélisées soit directement dans le composant externe soit dans un corps externe particulier, et d'autre part requiert à la fois un apport de chaleur à haute température au niveau du désorbeur, et une extraction de chaleur à moyenne température au niveau de l'absorbeur. La représentation de ces couplages thermiques est possible en utilisant deux thermocoupleurs, nommés "Absorbeur" et "Désorbeur" sur le schéma de la figure 2 : le composant externe "absorbeur-générateur" calcule les énergies thermiques qui doivent être échangées, et chaque thermocoupleur recalcule la transfo "échange" à laquelle il est connecté.

Dans ce TD, nous expliquons comment faire en pratique pour créer ce composant, après avoir donné les résultats de la simulation du système complet. Afin d'être facilement compréhensible, le modèle présenté ici est relativement simple sur le plan thermodynamique, calculable sans difficulté avec les hypothèses retenues. Il présente de ce fait diverses limitations : en particulier, tel qu'il est implémenté, il permet un calcul direct du composant connaissant la température de l'absorbeur et celle du désorbeur ainsi que l'efficacité de l'échangeur de solution, mais pas un calcul inverse où l'on chercherait à déterminer, pour des conditions de fonctionnement données, l'une de ces grandeurs. En revanche, il présente le grand intérêt d'être déjà suffisamment compliqué pour bien montrer les problèmes réels qui se posent lors de la création d'un composant externe, et les solutions qui permettent de les résoudre.



3.2 Résultats globaux

Une fois le composant manquant créé, la modélisation de l'installation de trigénération ne pose pas de problème particulier, et conduit au synoptique de la figure 2, dans lequel on reconnaît dans la partie supérieure gauche le cycle de réfrigération avec son circuit de refroidissement, dans la partie centrale droite la micro-turbine, et dans la partie inférieure gauche l'échangeur de production d'eau chaude sanitaire.

Le paramétrage de la micro-turbine correspond à celui d'une machine industrielle existante, mais l'installation de trigénération elle-même est fictive : on a simplement cherché à récupérer au mieux l'enthalpie disponible dans les gaz d'échappement de la turbine pour produire du froid et chauffer de l'eau sanitaire.

Dans le synoptique de la figure 3, on a comptabilisé dans l'énergie utile non seulement les travaux de compression et de détente, mais aussi l'énergie frigorifique et l'énergie thermique fournie au circuit d'eau, ce qui conduit, pour les conditions d'utilisation retenues, à un rendement global de près de 78 % qui correspond à environ 125 kW de puissance électrique, 176 kW de chaleur apportée au désorbeur, fournissant 127 kW de puissance frigorifique, et 40 kW de chaleur pour l'ECS, pour une consommation de 375 kW de combustible. Le rendement mécanique vaut 32,7 %, et le rapport chaleur-force 1,76. Le COP du cycle à absorption à simple effet est quant à lui égal à 0,72.

3.3 Conception du composant externe

Généralités

Le composant externe manquant, qui est implémenté dans la classe Java LiBrAbsorption, correspond au sous-système absorbeur-échangeur de solution-désorbeur du cycle à absorption utilisant le couple LiBr-H₂O (figure 1). Il inclut les propriétés du fluide thermodynamique. Le modèle retenu est celui d'une machine à simple effet avec les principales hypothèses suivantes (on notera que dans ce qui suit, pour des raisons d'homogénéité de notations, la richesse ou la pauvreté de la solution est exprimée par rapport au réfrigérant (l'eau) et non au solvant (le bromure de lithium), contrairement à la convention habituelle aux Etats Unis, qui est celle retenue dans la référence) :

- le désorbeur et le condenseur sont à la même pression
- l'absorbeur et l'évaporateur sont à la même pression
- la vapeur de réfrigérant sortant de l'évaporateur est saturée
- le liquide de réfrigérant sortant du condenseur est saturé
- la vapeur de réfrigérant sortant du désorbeur est saturée, à la température d'équilibre de la solution pauvre en réfrigérant à la pression du désorbeur
- la solution pauvre sort bouillante du désorbeur, supposé isotherme
- la solution riche sort saturée de l'absorbeur, supposé isotherme

Les paramètres du modèle sont les suivants :

- la température de l'absorbeur
- la température du désorbeur
- l'efficacité de l'échangeur de solution

Les données d'entrée du modèle sont les suivantes (fournies par les autres composants du système) :

- la pression de l'absorbeur
- la pression du désorbeur
- le débit de réfrigérant

Les sorties sont :

- la charge thermique de l'absorbeur
- la charge thermique du désorbeur
- la concentration de la solution riche
- la concentration de la solution pauvre
- le débit de la solution riche
- le débit de la solution pauvre

Interface graphique

Une interface graphique possible pour le composant s'en déduit (figure 4). Elle permet de construire le tiers inférieur droit de l'écran (en anglais sur la figure), le reste étant défini de manière standard dans Thermoptim.

Les paramètres correspondent aux trois premières lignes ajoutées, les résultats des calculs aux autres. Tout en bas sont listés les deux types de thermocoupleurs que ce composant définit.

Les données d'entrée sont fournies par les autres composants du modèle global : le débit de réfrigérant, fixé par la transfo amont "effet frigorifique" vaut ici 0,055 kg/s, la pression de l'absorbeur (point 1) est égale à 0,00697 bar, correspondant à une température d'évaporation de 1,93 °C, et la pression du désorbeur (point 2) est égale à 0,102 bar, correspondant à une température de condensation de 46,22 °C.

Une particularité de ce composant est qu'il ne modifie pas son point aval 2, dont l'état est considéré comme une donnée d'entrée. Il serait bien sûr possible d'aborder le problème sous un autre angle, mais, comme nous l'avons dit plus haut, nous n'avons pas voulu trop compliquer les choses dans cet exemple.

The screenshot shows the 'Transfo' software interface. The window title is 'Transfo'. The interface is divided into several sections:

- Configuration:** 'transfo' is set to 'absorbeur-générateur', 'type' is 'externe'. There are buttons for '<', '>', 'Sauver', 'Supprimer', and 'Fermer'.
- Energy Type:** 'type énergie' is 'autre'. There is a 'débit imposé' checkbox (unchecked) and a 'débit' input field with the value '0,055'.
- System Type:** Radio buttons for 'système fermé' (unchecked) and 'système ouvert' (checked). There is also an 'observée' checkbox (unchecked).
- Points and Enthalpy:** 'point amont' is '1' (with an 'afficher' button), 'point aval' is '2' (with an 'afficher' button). 'm ΔH' is '4,42'. A 'Calculer' button is present.
- Thermodynamic Properties (Left):**

T (°C)	1,93
P (bar)	0,00697
h (kJ/kg)	2 505,14
titre	1

T (°C)	46,22
P (bar)	0,102
h (kJ/kg)	2 585,5
titre	1
- LiBr Absorption Parameters (Right):**

absorber temperature (°C)	40.000
desorber temperature (°C)	100.000
solution exch. efficiency	0.800
absorber load	-171.707
desorber load	176.037
Rich solution fraction	0.399
Poor solution fraction	0.364
Rich solution flow	1.006
Poor solution flow	0.951
reference	
- Bottom Section:** A large empty box with labels 'absorber' and 'desorber' below it.

Figure 4 : Interface graphique du composant externe

Modèle thermodynamique

Les équations du modèle sont obtenues comme suit.

Mélange LiBr-H₂O

Le mélange LiBr-H₂O est modélisé en utilisant les équations retenues par l'ASHRAE.

Absorbeur

La solution pauvre en réfrigérant est pulvérisée en pluie et vient asperger la vapeur de réfrigérant à basse pression, qui est absorbée, libérant son enthalpie de condensation et sa chaleur de dilution. Cette chaleur Q_{abs} est extraite par refroidissement par de l'eau qui refroidit ensuite le condenseur.

Avec l'hypothèse que l'absorbeur est à température constante T_{abs} et que la solution riche est saturée, les équations de l'absorbeur sont les suivantes :

L'équation de pression de vapeur saturante de la solution $P_{abs} = P(x_{sr}, T_{abs})$ fournit la concentration de la solution riche saturée, et donc son enthalpie $h_{srD} = h(x_{sr}, T_{abs})$.

Données connues : $m_r, h_{r1}, T_{abs}, P_{abs}$

Inconnues : $m_{sr}, m_{sp}, x_{sp}, h_{spE}$

Conservation de la masse : $m_r + m_{sp} = m_{sr}$

Conservation de la masse de solution : $(1 - x_{sp}) m_{sp} = (1 - x_{sr}) m_{sr}$

Ces deux équations fournissent m_{sp} et m_{sr} si x_{sr}, x_{sp} et m_r sont connus :

$$m_{sr} = m_r \frac{1 - x_p}{x_r - x_p}$$

$$m_{sp} = m_r \frac{1 - x_r}{x_r - x_p}$$

Conservation de l'enthalpie : $m_r h_{r1} + m_{sp} h_{spE} = m_{sr} h_{srD} + Q_{abs}$

Soit 8 variables, 4 données et 3 équations.

Générateur

La solution riche en réfrigérant est introduite à haute pression dans le générateur à haute température, où elle bout par contact avec des tubes chauffés soit directement par un combustible, soit par de la vapeur d'eau. La vapeur produite est quasiment du réfrigérant pur, du fait de la différence de tension saturante entre les deux fluides. Elle est ensuite dirigée vers le condenseur. La solution appauvrie est extraite pour être recyclée.

Avec l'hypothèse que le générateur est à température constante T_{gen} et que la solution pauvre est saturée, les équations sont les suivantes :

L'inversion de l'équation de pression de vapeur saturante de la solution $P_{gen} = P(x_{sp}, T_{gen})$ fournit la concentration x_{sp} , et son enthalpie h_{spA}

Données nouvelles : h_{r2}, T_{gen}, P_{gen}

Inconnues nouvelles : h_{srB}, h_{spA}

Conservation de l'enthalpie: $m_r h_{r2} + m_{sp} h_{spA} = m_{sr} h_{srB} + Q_{gen}$

Soit 5 variables, 3 données et 1 équation.

Echangeur de solution

A ce stade, il manque deux équations pour résoudre le modèle. Elles correspondent à l'échangeur de solution :

$$\varepsilon = f(T_A, T_B, T_D, T_E)$$

$$m_{sp} (h_{spA} - h_{spE}) = m_{sr} (h_{srB} - h_{srD})$$

Enchaînement des calculs

Concrètement, l'enchaînement des calculs est le suivant :

- 1) mise à jour du composant avant calcul par chargement des valeurs de m_r , h_{r1} , et P_{abs} à partir du point amont
- 2) lecture de T_{abs} , T_{gen} et ϵ sur l'écran du composant externe
- 3) inversion de $P_{abs} = P(x_{sr}, T_{abs})$ pour obtenir x_{sr} puis h_{srD}
- 4) chargement des valeurs de h_{r2} et P_{gen} à partir du point aval
- 5) inversion de $P_{gen} = P(x_{sp}, T_{gen})$ pour obtenir x_{sp} puis h_{spA}
- 6) calcul de m_{sp} et m_{sr}
- 7) calcul de T_B et T_E par les équations de l'échangeur de solution
- 8) calcul de h_{srB} et h_{spE} en faisant l'hypothèse que ces points sont à l'équilibre à leur température et concentration respectives
- 9) calcul des charges thermiques Q_{abs} et Q_{gen}
- 10) mise à jour de l'écran du composant externe
- 11) mise à jour et calcul des thermocoupleurs associés

Examinons maintenant les problèmes rencontrés en pratique à chacune de ces étapes. Dans ce qui suit on a cherché à donner des explications aussi simples que possible pour bien montrer le principe de construction du modèle. Pour davantage de précisions on se reportera à la documentation de ThermoOptim relative aux classes externes. Quelques extraits du code sont donnés ci-dessous, mais il est recommandé de lire la suite de cette note en ayant sous les yeux l'intégralité de la classe LiBrAbsorption.

1) mise à jour du composant par chargement des valeurs de m_r , h_{r1} , et P_{abs} à partir du point amont

La petite difficulté est ici qu'un composant externe n'a pas un accès direct aux variables du simulateur : ces grandeurs sont obtenues par des méthodes particulières génériques, qui construisent des Vector de structure différente selon l'objet désiré.

La marche à suivre n'est pas compliquée mais demande à être respectée :

```
String[] args=new String[2]; //tableau des arguments
args[0]="process";//type d'élément désiré (ici une transfo)
args[1]=tfe.getCompName();//nom de la transfo (ici le nom du composant externe (tfe) lui-même)
Vector vProp=proj.getProperties(args);//méthode du simulateur exécutable par le composant externe
Double f=(Double)vProp.elementAt(3);
double flow=f.doubleValue();//valeur du débit, propagé automatiquement depuis la transfo amont
String amont=(String)vProp.elementAt(1);//nom du point amont (ici le point 1)
getPointProperties(amont);//décodage automatique du Vector (méthode de ExtProcess)
Tamont=Tpoint;//ici T1
Pamont=Ppoint;//ici P1 et donc Pabs
double Hamont=Hpoint;//ici h1
```

Pour avoir accès au point aval, la démarche est tout à fait analogue :

```
String aval=(String)vProp.elementAt(2);//gets the downstream point name
getPointProperties(aval);//direct parsing of point property vector
Taval=Tpoint;
Paval=Ppoint;
double Haval=Hpoint;
```

2) lecture de T_{abs} , T_{gen} et ϵ sur l'écran du composant externe

Le package extThopt fournit un certain nombre de méthodes simples mais robustes pour convertir en double les String affichés dans les champs JTextField utilisés dans l'interface graphique, et réciproquement pour afficher les double dans ces champs. Ils sont implémentés sous forme de méthodes statiques de la classe extThopt.Util :

```
epsiSol=Util.lit_d(epsiSol_value.getText());
```

```
Tabs=Util.lit_d(Tabs_value.getText()+273.15;
Tgen=Util.lit_d(Tgen_value.getText()+273.15;
```

3) inversion de $P_{abs} = P(x_{sr}, T_{abs})$ pour obtenir x_{sr} puis h_{srD}

La première étape consiste à définir les fonctions de calcul des propriétés du couple LiBr-H₂O. Comme indiqué plus haut, on a retenu les formulations proposées par l'ASHRAE [2], exprimées par rapport à X pourcentage massique en LiBr entre 0 et 100. Deux fonctions sont fournies, pour calculer la pression et l'enthalpie du mélange à l'équilibre. Elles sont implémentées dans les méthodes getP(double x, double T) et getH(double x, double T).

Pour pouvoir inverser des fonctions, la classe extThopt.Util propose une méthode par dichotomie appelée dico_T, qui reçoit en arguments :

- la classe appelante, qui doit implémenter l'interface Inversible
- la valeur à atteindre
- un paramètre éventuel
- un String caractérisant la fonction dans la méthode f_dicho() définissant les fonctions inversables
- les valeurs minimales et maximales de l'intervalle de recherche
- la précision demandée

Dans f_dicho(), il faut définir la fonction à inverse par un String, ici "P_LiBr" :

```
if (fonc.equals("P_LiBr"))return getP(x,T);
```

Avec cette signature, l'inversion est obtenue par :

```
xsr=Util.dicho_T (this,Ppoint,Tabs, "P_LiBr",0.3, 0.55,0.00001);
```

qui s'interprète comme : inverser la fonction définie dans f_dicho() par "P_LiBr", la valeur à atteindre étant Ppoint, le paramètre étant Tabs, l'intervalle de recherche étant [0,3 – 0,55] et la précision requise 10⁻⁵.

L'intervalle a été choisi pour correspondre à la validité des équations de l'ASHRAE, données pour X = 100 (1 – x) compris entre 45 % et 70 %.

Etapes 4 à 10

Ces étapes ne posent aucun problème particulier. On notera que le code prévoit l'impression de résultats intermédiaires, essentiellement utiles dans la phase de mise au point.

11) mise à jour et calcul des thermocoupleurs associés

Etant donné que les thermocoupleurs sont des sortes d'échangeurs de chaleur, il est intéressant de les caractériser par des valeurs telles que l'efficacité ϵ , le UA, le NUT ou la DTML, que l'on peut calculer à partir d'équations analogues.

Pour cela, il faut que le composant externe transmette à chacun de ses thermocoupleurs les valeurs d'équivalents débits, température d'entrée et de sortie et énergie thermique transférée qu'ils doivent prendre en compte dans leurs calculs. Des méthodes spécifiques ont été placées dans l'interface à cet effet.

Il faut cependant être conscient que l'analogie avec les échangeurs comporte quelques limites : par exemple, des croisements de température inadmissibles dans un échangeur peuvent se présenter dans un thermocoupleur, conduisant à une absurdité au niveau des calculs si l'on n'y prend pas garde.

On aura donc souvent intérêt à transmettre des valeurs qui ne risquent pas de conduire à ce genre de situation, une solution pouvant être de considérer, pour les calculs des caractéristiques analogues à celles d'un échangeur, que le thermocoupleur est isotherme, conformément au modèle retenu ici pour l'absorbeur et le désorbeur.

La méthode updateThermoCoupler() d'ExtProcess permet d'effectuer cette mise à jour :


```
updateThermoCoupler("absorber", Tabs, Tabs, Qabs, msr);
updateThermoCoupler("desorber", Tgen, Tgen, Qgen, msp);
```

Par exemple, on a ici supposé que l'absorbeur est à la température T_{abs} , qu'il reçoit la charge thermique Q_{abs} et qu'il est traversé par le débit m_{sr} . Si l'on n'avait pas pris la température de l'absorbeur comme référence pour les calculs d'échange, en conservant celles de la vapeur d'eau entrant et sortant de la transfo externe, on aurait abouti à un croisement de température inadmissible.

Sauvegarde et chargement des paramètres du modèle

Il est possible de sauvegarder dans les fichiers de projet normaux de Thermoptim (et ensuite de relire) les paramétrages des composants externes en utilisant deux méthodes spécifiques :

```
public String saveCompParameters()
public void readCompParameters(String ligne_data)
```

La seule contrainte est que l'ensemble du paramétrage du composant externe doit tenir sur une ligne, avec un formatage compatible avec celui qui est utilisé dans le noyau du progiciel : les différents champs de sauvegarde sont séparés par des tabulations.

ExtThopt.Util fournit une méthode générique pour pouvoir associer à la valeur d'un paramètre un code permettant de le repérer :

```
public static String extr_value(String ligne_data, String search)
```

La sauvegarde est effectuée sous la forme "paramètre= valeur", et la recherche se fait sous la forme :

```
valeur=Util.lit_d(Util.extr_value(ligne_data, "paramètre"));
```

Si le paramétrage du composant est trop complexe pour être sauvegardé de la sorte, rien n'empêche un utilisateur d'utiliser ce mécanisme pour sauvegarder le nom d'un fichier spécifique de paramétrage, et ensuite de relire comme il le désire ce fichier.

4) Variantes

Il pourrait être intéressant d'étendre le modèle du cycle à absorption à simple effet au cas d'un cycle à double effet, plus performant.

5) Fichiers de travail

Les fichiers de projet et de schéma des modèles Thermoptim des cycles présentés ci-dessus sont joints au dossier dans l'archive Trigeneration.zip, qui contient aussi les fichiers de la classe externe LiBrAbsorption.