Modélisation dans Thermoptim d'une chambre flash pour solution aqueuse

Nous avons développé une classe externe dans laquelle le **produit à concentrer** est modélisé par un **retard à l'ébullition** proportionnel à la concentration, les autres propriétés thermodynamiques étant celles de l'eau (classe EauSalee).

Lors d'une opération de dessalement, un des procédés consiste à effectuer un flash de l'eau salée, ce qui a pour effet de vaporiser une fraction du débit total et d'augmenter la concentration du soluté. Une chambre de flash se comporte comme un diviseur recevant en entrée le produit à concentrer, et d'où sortent deux fluides, la vapeur d'eau et la solution concentrée. La chambre étant adiabatique, l'enthalpie de vaporisation est prélevée sur la solution aqueuse, dont la température baisse.

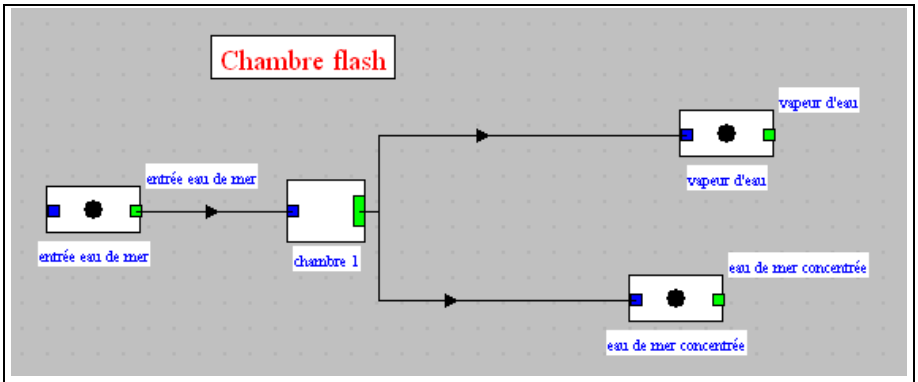

Figure 1 : Schéma du composant

La structure du modèle de chambre est donnée figure 1.

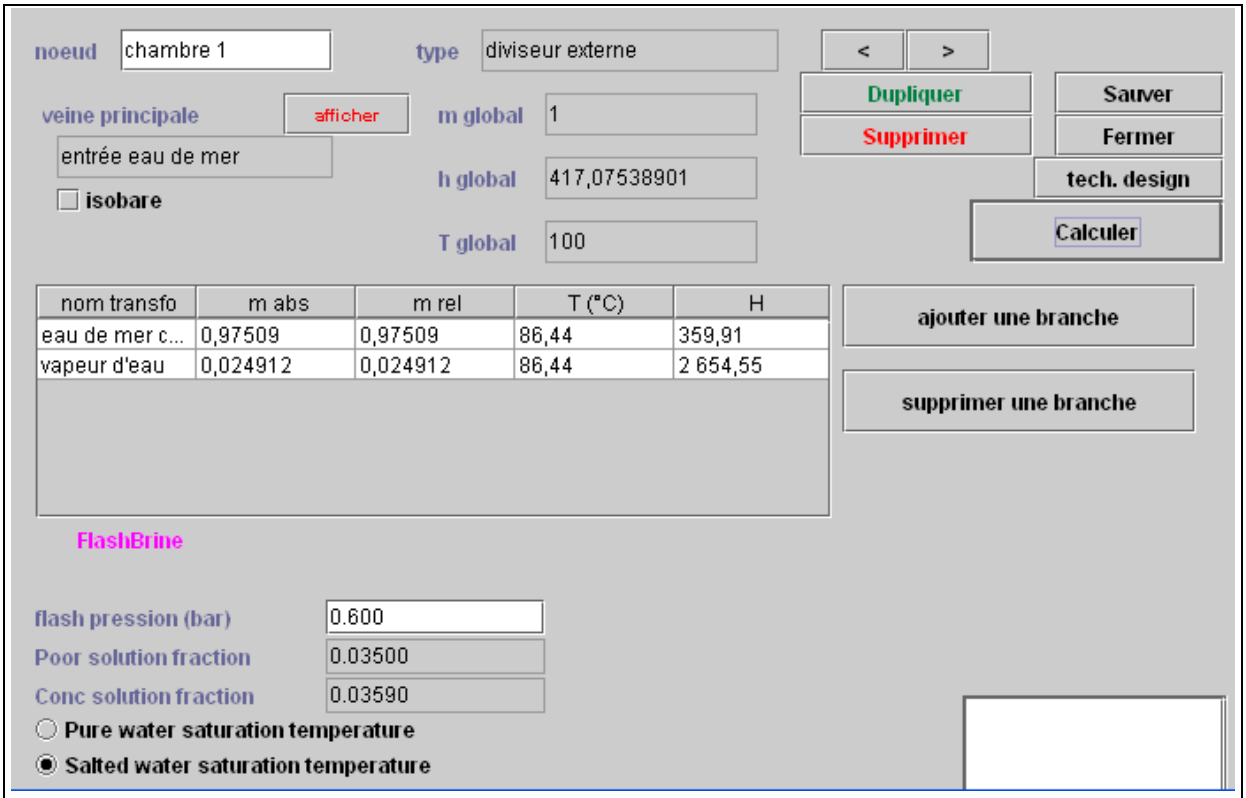

Figure 2 : Ecran du composant

Modèle physique

En appelant x la concentration massique en soluté, les équations qui régissent le comportement de cette unité sont les suivantes :

conservation du débit total : $\dot{m}_1 = \dot{m}_2 + \dot{m}_3$ (1)

conservation du soluté : $x_1 \dot{m}_1 = x_3 \dot{m}_3$ (2)

conservation de l'enthalpie : $h_1 \dot{m}_1 = h_3 \dot{m}_3 + h_2 \dot{m}_2$ (3)

Le modèle ne fait appel qu'à un seul paramètre : la pression P_f à laquelle le flash prend place. L'utilisateur peut choisir entre deux hypothèses :

- soit que la phase liquide de l'eau de mer concentrée est en équilibre avec les buées, et que donc la température qui règne dans la chambre est celle de saturation du produit à la pression P_f ;
- soit que c'est celle de saturation de l'eau à la pression P_f qui doit être prise en compte.

L'écran du composant est donné figure 2, le produit étant flashé de 1 à 0,6 bar. 2,5 % du débit est alors vaporisé à 86,4 °C, et le synoptique correspond à la figure 3.

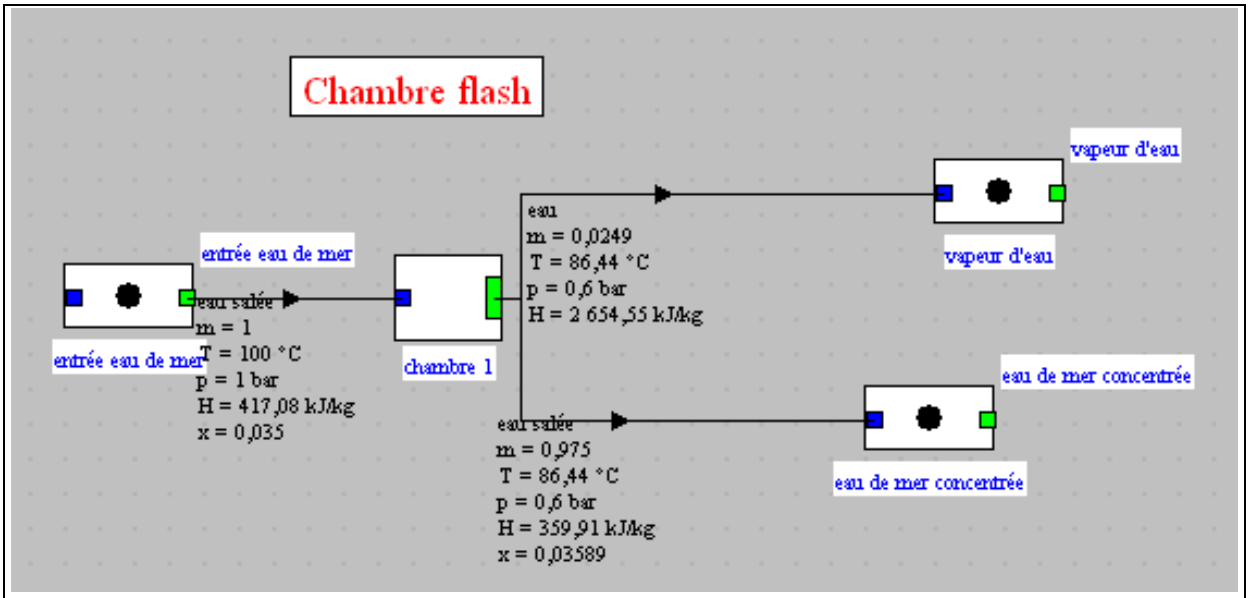

Figure 3 : Synoptique du composant

Etude de la classe externe FlashBrine

Des tests de cohérence sur la construction du diviseur externe sont effectués par la méthode checkConsistency() pour vérifier que les fluides connectés sont les bons : dans ce cas, un produit en entrée et le produit et de l'eau et en sortie. Les tests ici implémentés sont très sommaires et pourraient être améliorés. On se reportera au tome 3 du manuel de référence pour les explications sur ce point, valables pour tous les nœuds externes.

L'étude de la classe externe FlashBrine permet de voir comment le modèle a été implémenté. Deux étapes suffisent pour effectuer les calculs :

1) on cherche par dichotomiela concentration de sortie qui permet de résoudre le jeu d'équations retenu, et détermine la température de sortie des fluides (produit concentré et vapeur d'eau) ainsi que leurs débits-masses

```
if(isBuilt){
    Pflash=Util.lit_d(Pflash_value.getText());
    double Xmax=1.;
    Xconc=Util.dicho_T(this,0,1, "X_conc",Xpoor, Xmax,0.00001);//recherche de la concentration de sortie
```

La recherche est faite en annulant le résidu de l'équation (3) :

```
double getOutletConcentration(double T, double X){

    if(jcheckTsatEauSalee.isSelected())Tflash=produit.getSatTemperature(Pflash,X);
    else Tflash=eau.getSatTemperature(Pflash,X);

    Tbuees=Tflash;
    eau.CalcPropCorps(Tbuees,Pflash,1);
    Hbuees=eau.getState()[5];

    mconc=mpoor*Xpoor/X;//concentration en sortie
    mbuees=mpoor-mconc;//débit de vapeur

    produit.CalcPropCorps(Tflash,Pflash,X) ;
    getSubstProperties(nomCorps);
    Hconc=Hsubst;

    double z=mconc*Hconc+mbuees*Hbuees-mpoor*Hpoor;
    return z;
}
```

2) le nœud est ensuite mis à jour en utilisant les méthodes génériques décrites dans le manuel de référence

```
            vTransfo= new Vector[nBranches+1];
            vPoints= new Vector[nBranches+1];
            setupPoorSolution(mpoor,Tpoor,Ppoor,Xpoor);
            setupConcSolution(mconc,Tflash,Pflash,Xconc);
            setupBuees(mbuees,Tbuees,Pflash,1);
            updateDivider(vTransfo,vPoints,Tpoor,Hpoor);

            Xpoor_value.setText(Util.aff_d(Xpoor,5));
            Xconc_value.setText(Util.aff_d(Xconc,5));
        }
```