Thermoptim model of a flash chamber for aqueous solution

We have developed an external class in which the product to concentrate is modeled by an aqueous solution with a boiling point elevation proportional to the concentration, the other thermodynamic properties being those of water (Class EauSalee).

During an operation of desalination, a process is to perform a flash of salt water, which has the effect of vaporizing a fraction of the total flow-rate and to increase the concentration of the solute.

A flash chamber acts as a divider receiving as input the product to concentrate, and from which exit two fluids: water vapor and the concentrated solution. The chamber being adiabatic, the enthalpy of vaporization is taken from the aqueous solution, whose temperature drops.

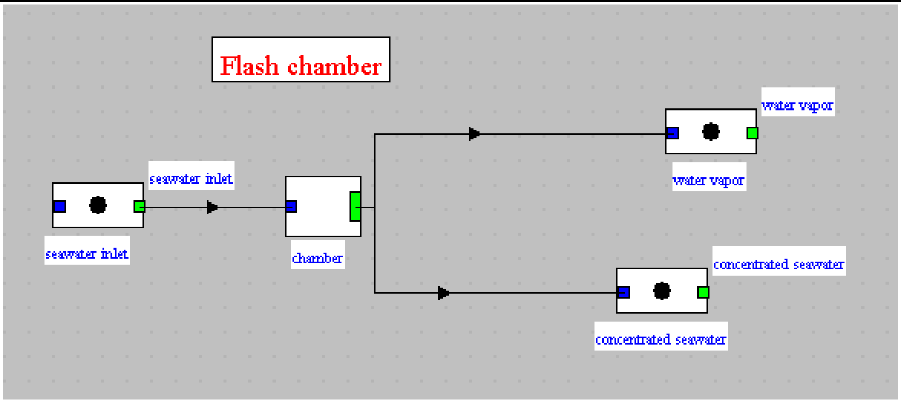

Figure 1: Component diagram

The structure of the model is given in Figure 1.

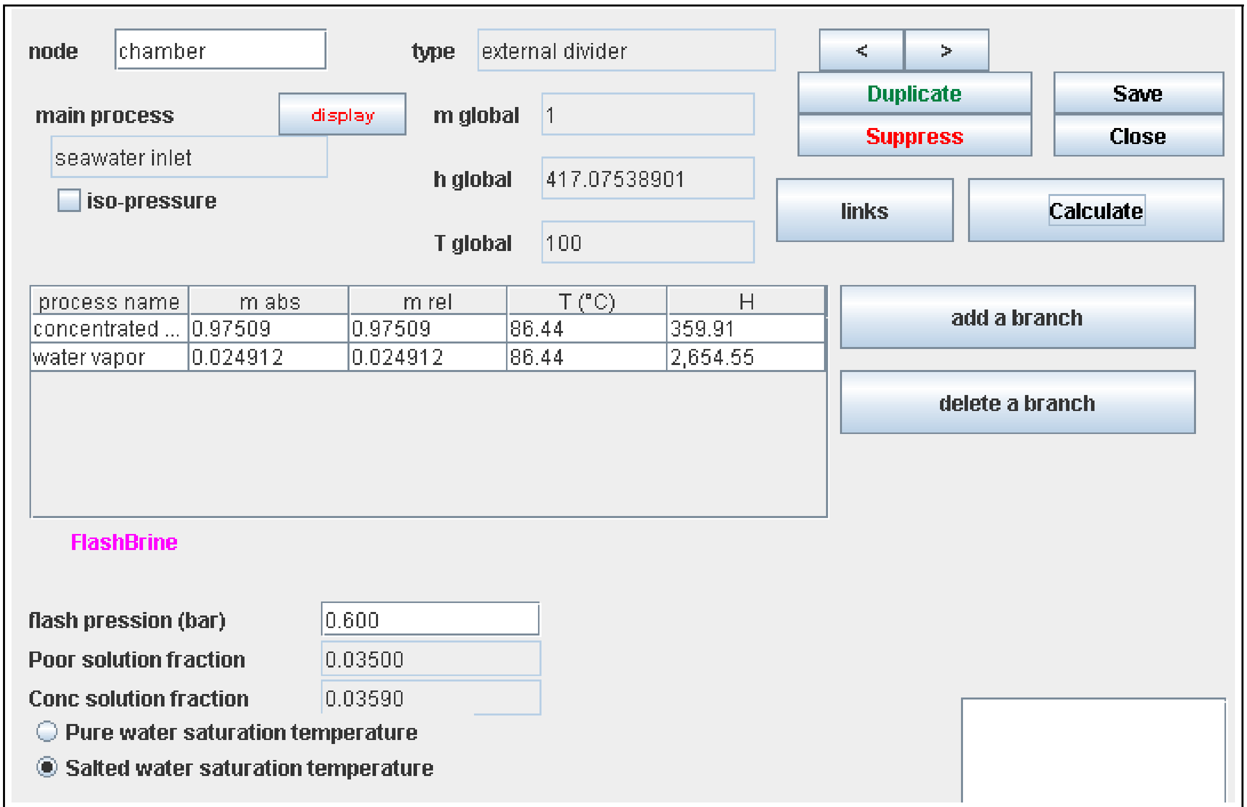

Figure 2: Component screen

Physical model

Calling x the mass concentration of solute, the equations governing the behavior of this unit are:

conservation of the total flow: $\dot{m}_1 = \dot{m}_2 + \dot{m}_3$ (1)

conservation of the solute: $x_1 \dot{m}_1 = x_3 \dot{m}_3$ (2)

conservation of enthalpy: $h_1 \dot{m}_1 = h_3 \dot{m}_3 + h_2 \dot{m}_2$ (3)

The model uses only a single parameter: the pressure P_f at which the flash takes place. The user can choose between two hypotheses:

- Either the liquid phase of the concentrated seawater is in equilibrium with the vapor, and therefore the temperature prevailing in the chamber is the product saturation temperature at pressure P_f;
- Or the water saturation temperature at pressure P_f must be taken into account.

The component screen is given in Figure 2, the product being flashed from 1 to 0.6 bar. 2.5% of the flow-rate is then vaporized at 86.4 ° C, and the synoptic view is given Figure 3.

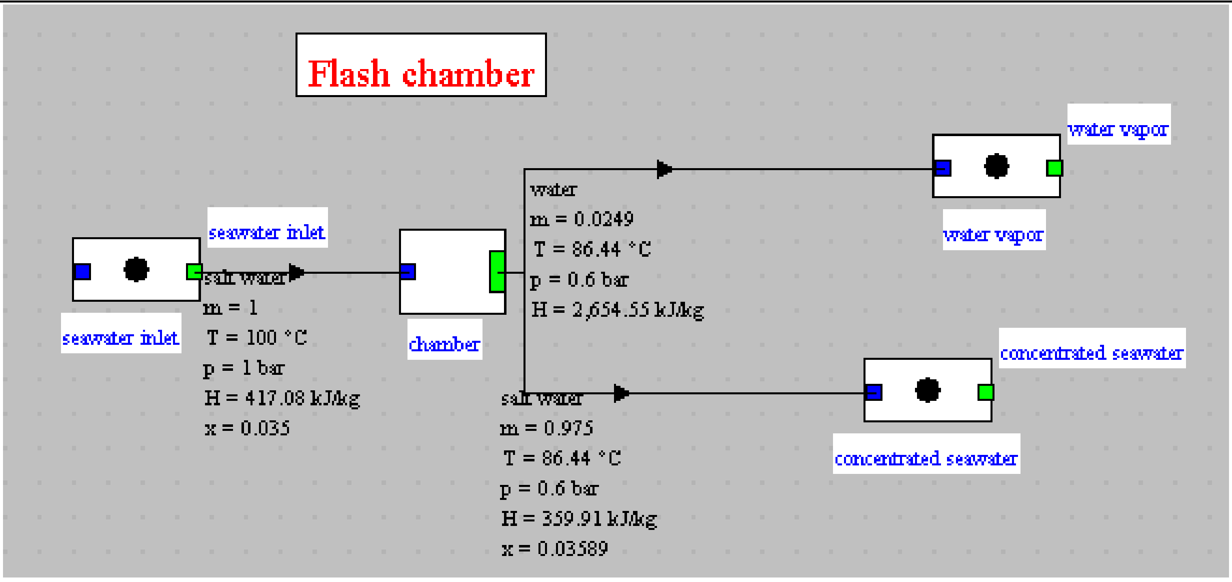

Figure 3: Synoptic view of the component

Study of the external class FlashBrine

Consistency tests on the construction of the external divider are made by the method checkConsistency() to check the fluids are well connected: in this case, a product as input and the product and water as output. The tests implemented here are very basic and could be improved. Please refer to Volume 3 of the reference manual for explanations on this point, valid for all external nodes.

The study of the external class FlashBrine shows how the model has been implemented. Two steps are enough to make the calculations:

1) we seek by dichotomy the output concentration that solves the set of equations selected, and determines the outlet temperature of fluids (concentrate and water vapor) and their mass flow rates

```
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
```

Research is done by canceling the residue of the equation (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) the node is then updated using the generic methods described in the reference manual

```
        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));
    }
```