

## Modélisation d'une tour de refroidissement à contact direct dans ThermoOptim

Une tour de refroidissement à contact direct présente la particularité d'être traversée par deux flux séparés : de l'air et de l'eau, qui échangent de la matière et de l'énergie par l'intermédiaire d'une interface. Elle se comporte donc comme un quadripôle recevant deux fluides en entrée, et dont en sortent deux autres.

Pour la représenter dans ThermoOptim, on forme ce quadripôle en associant un mélangeur en entrée et un diviseur en sortie, les deux étant reliés par une transfo-point qui joue un rôle purement passif.

Pour que le modèle soit bien cohérent, on synchronise les calculs effectués par les deux nœuds. Plus précisément le diviseur de sortie prend le contrôle du mélangeur, dont le rôle se limite à effectuer une mise à jour des variables de couplage associées aux flux d'entrée. La structure du modèle est donnée figure 1.

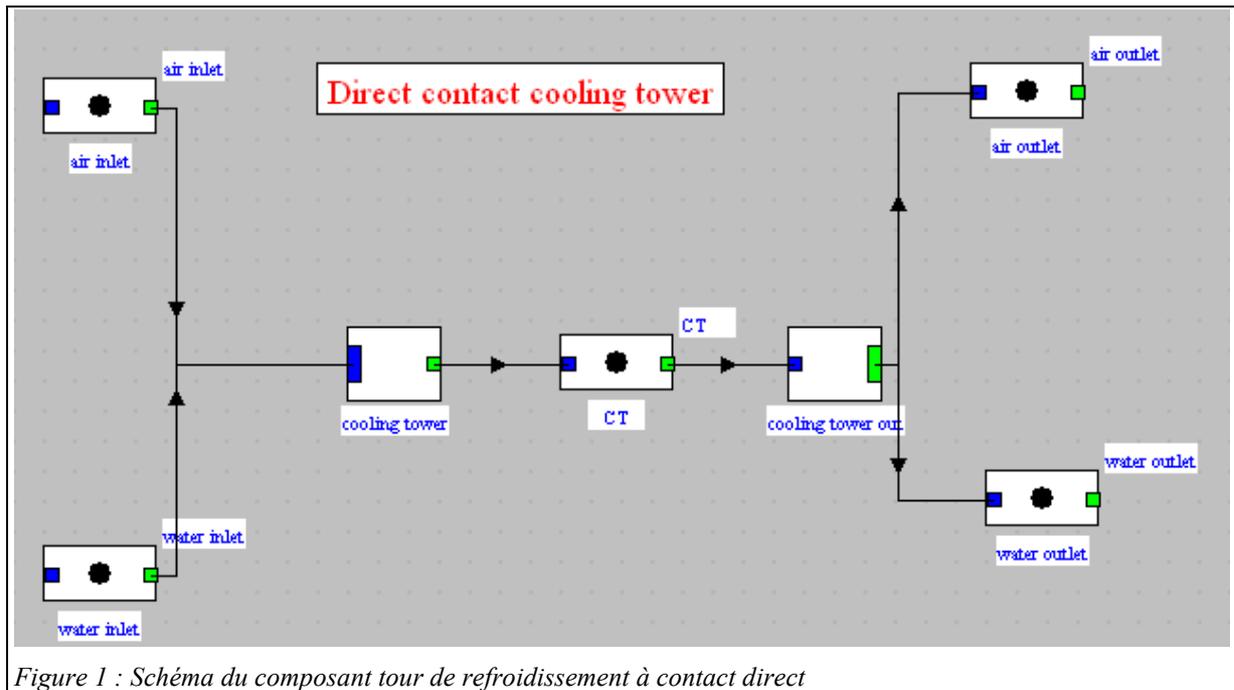


Figure 1 : Schéma du composant tour de refroidissement à contact direct

## Modèle de la tour de refroidissement à contact direct

Le modèle est basé sur un raisonnement enthalpique global où sont supposées connues les conditions d'entrée et de sortie côté air et celles d'entrée côté eau, et les résultats du modèle sont cohérents avec les valeurs expérimentales et celles fournies par les constructeurs.

Les débits des deux flux entrants sont imposés par les conditions en amont du composant et non pas recalculés. Si le débit d'eau est insuffisant pour que son refroidissement (jusqu'à la température humide de l'air entrant) permette de fournir à l'air l'enthalpie requise, un message avertit l'utilisateur.

Les fonctions de calcul des propriétés humides des gaz et des points de ThermoOptim ont été rendues accessibles depuis les classes externes. Nous vous conseillons de vous référer à la note : "Calculs des gaz humides depuis les classes externes"<sup>1</sup> pour une présentation détaillée des méthodes disponibles.

Le modèle que l'on peut retenir est alors le suivant, lorsque l'on connaît la température de sortie de l'air :

- 1) on commence par calculer les propriétés humides de l'air entrant, et on détermine le débit-masse de gaz sec à partir de celui du gaz humide ; l'humidité relative  $\varepsilon$  en entrée de tour est affichée à l'écran
- 2) l'humidité relative  $\varepsilon$  en sortie de tour est imposée égale à 1, et les propriétés humides de l'air de sortie sont calculées, ce qui fournit les enthalpies spécifique et totale à apporter à l'air
- 3) le débit d'eau emporté par l'air est déterminé et la composition de l'air humide en sortie est modifiée
- 4) le bilan enthalpique sur l'eau fournit sa température de sortie, qui doit être supérieure à la température humide de l'air entrant

<sup>1</sup> <http://www.thermooptim.org/sections/base-methodologique/extensions-thermooptim/calculs-gaz-humides>

- 5) les valeurs en aval du nœud sont mises à jour
- 6) le NUT est calculé sur la base de l'écart moyen logarithmique des enthalpies, et l'efficacité s'en déduit à partir de la relation classique pour les échangeurs à contre-courant

La tour est comme on l'a dit représentée par un mélangeur externe connecté à un diviseur externe, les calculs étant effectués par ce dernier. Les classes s'appellent DirectCoolingTowerInlet et DirectCoolingTower.

L'écran du composant tour de refroidissement est donné figure 2. On désire ici refroidir 1kg/s d'eau de 30 °C à 25 °C. Avec un débit de 0,9

kg/s d'air ambiant à 18 °C et d'humidité relative égale à 0,5, la température de l'air sortant est de 20,1 °C et 7,5 g/s d'eau sont évaporés. La puissance de la tour est de 20,9 kW.

The screenshot shows a software interface for a cooling tower component. It includes the following elements:

- noeud:** cooling tower out
- type:** external divider
- veine principale:** CT
- isobare:**
- m global:** 1,9
- h global:** 57,6289
- T global:** 20,09157019
- Table:**

nom transfo	m abs	m rel	T (°C)	H
air outlet	0,90749	0,90749	20,09	-4,83
water outlet	0,99251	0,99251	25	104,9
- direct cooling tower section:**
  - NTU value: 0.499
  - inlet rel. humidity: 0.500
  - $\Delta Q'$ : 20.911
  - water involved: 0.00749
  - approach (°C): 12.868
  - range (°C): 5.000
  - epsilon: 0.354
  - UA: 1.310
  - R: 0.629
  - LMTD: 11.324
  - NTU (integrated): 0.530
  - Twater out known
  - NTU known

Figure 2 : Ecran du composant tour de refroidissement

On voit apparaître sur cet écran deux valeurs de NUT. Celle qui se situe dans le champ éditable est calculée par la formule approchée basée sur les écarts moyens logarithmiques des enthalpies de l'air saturé entre l'entrée et la sortie de la tour, tandis que celle qui est affichée tout en bas est le résultat de l'intégration de l'équation différentielle de Merkel pour une tour à contre-

The screenshot shows the 'point d'entrée' (inlet point) interface. It includes the following elements:

- projet:** cooling tower
- point:** air inlet
- corps:** air\_humide
- external mixture:**
- observée:**
- Systeme ouvert (T,P,h) / Systeme fermé (T,v,u) / Mélanges humides**
- imposer w / imposer epsi / imposer l'humidité du gaz**
- valeurs spécifiques (rapportées à 1 kg de gaz sec):**

w (kg/kg)	0,00639642855	q' (kJ/kg)	34,2469
epsi	0,5	v (m3/kg)	0,8334655
condensats	0	t' (°C)	12,132
p (bar)	1,01325	tr (°C)	7,431
T (°C)	18		
T (K)	291,15		

Figure 3 : Ecran du point d'entrée

courant. Elle est plus précise, mais, comme on peut voir que leurs valeurs sont proches, nous ne l'avons pas retenue dans ce modèle pour le garder plus simple sur le plan thermodynamique. Cette simplification permet de surcroît de bâtir un modèle calculable en mode direct (estimation de NUT) et indirect (NUT connu).

Les figures 3 et 4 fournissent les propriétés humides de l'air en entrée et en sortie de la tour, et la figure 5 un synoptique du modèle.

projet   observée

point

corps

mélange externe

valeurs spécifiques (rapportées à 1 kg de gaz sec)

w (kg/kg)	<input type="text" value="0,0147702923"/>	q' (kJ/kg)	<input type="text" value="57,6289"/>
epsi	<input type="text" value="1"/>	v (m3/kg)	<input type="text" value="0,8506372"/>
condensats	<input type="text" value="0"/>	t' (°C)	<input type="text" value="20,0915"/>
p (bar)	<input type="text" value="1,01325"/>	tr (°C)	<input type="text" value="20,0915742278"/>

T (°C)

T (K)

Figure 4 : Ecran du point de sortie

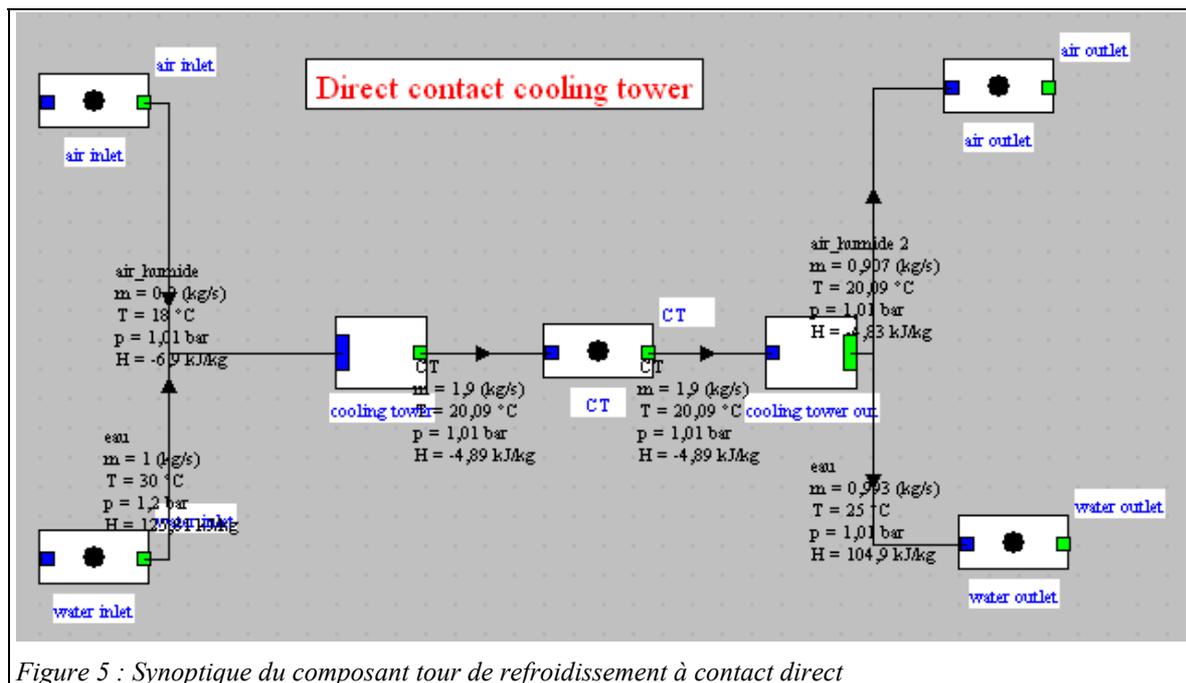


Figure 5 : Synoptique du composant tour de refroidissement à contact direct

### Etude de la classe externe DirectCoolingTower

Pour assurer la cohérence du modèle (éviter que l'on connecte le mélangeur d'entrée à un diviseur de sortie inadéquat), chacun des deux nœuds essaie d'instancier l'autre en recherchant sa classe parmi les composants externes du projet, et vérifie que tous deux sont bien connectés à la même transfo de liaison. Si l'opération échoue, un message avertit l'utilisateur que la construction est incorrecte. Cette vérification est effectuée par les méthodes `setupOutlet()` et `setupInlet()`.

De surcroît, des tests de cohérence de chaque nœud sont effectués par la méthode `checkConsistency()` pour vérifier que les fluides connectés sont les bons : dans ce cas, de l'air humide et de l'eau en entrée et en sortie. 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 `DirectCoolingTower` permet de voir comment le modèle a été implémenté. Comme on peut le voir, dans le cas où la température de l'air en sortie est connue, six étapes suffisent pour effectuer les calculs (dans les autres cas, la démarche est analogue) :

- 1) on commence par calculer les propriétés humides de l'air entrant, grâce à la méthode générique `updatepoint()`, puis on initialise le débit d'air sec et l'enthalpie spécifique amont :

```
//imposition de w et calcul des propriétés humides
updatepoint(amont, false, 0, //T
            false, 0, false, 0, //P,x
            true, "setW and calculate all", inlet_w);

getPointProperties(amont);

double TprimeAmont=Tprimepoint+273.15;
Double f=(Double)vProp.elementAt(3);
double flow=f.doubleValue();//débit massique de gaz humide
flow_as=flow/(1+Wpoint);//débit massique de gaz sec
JLabel1.setText("inlet rel. humidity : "+Util.aff_d(Epsipoint,3));
qPrimeAmont=QPrimepoint;//enthalpie spécifique de l'air entrant
```

- 2) on calcule ensuite les propriétés de l'air humide sortant et on en déduit l'enthalpie totale mise en jeu

```
//Propriétés humides de l'air sortant
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=airProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
aval=(String)vProp.elementAt(1);//gets the upstream point name
getPointProperties(aval);
outletT=Tpoint;

epsi=1;
updatepoint(aval, false, 0, //T
            false, 0, false, 0, //P,x
            true, "setEpsi and calculate", epsi);
getPointProperties(aval);

qPrimeAval=QPrimepoint;//enthalpie spécifique de l'air sortant

DeltaQprime=flow_as*(qPrimeAval-qPrimeAmont);//enthalpie totale acquise par l'air
JLabel3.setText("\u0394Q' : "+Util.aff_d(DeltaQprime,3));
```

- 3) on modifie la composition de l'air humide en sortie, et on calcule les débits-masses sortant :

```
//modification de la composition du gaz
updatepoint(aval, false, 0, //T
            false, 0, false, 0, //P,x
            true, "modHum", 0);

//Bilans massiques air et eau
outletFlow=flow_as*(1+Wpoint);

waterFlow=dcti.waterFlow - (Wpoint-inlet_w)*flow_as;
JLabel4.setText("water involved : "+Util.aff_d((Wpoint-inlet_w)*flow_as,5));
```

- 4) le bilan enthalpique sur l'eau fournit alors la température de l'eau sortante

```

//Bilans enthalpique sur l'eau
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=waterProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
waterOut=(String)vProp.elementAt(1);//gets the upstream point name
getPointProperties(waterOut);
hAval=dcti.waterH-DeltaQprime/dcti.waterFlow;
getSubstProperties(nomCorps);

waterOutT=lecorps.getT_from_hP(hAval,Ppoint);//température de sortie de l'eau

```

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

```

//mise à jour du lien entre les noeuds externes
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=mainProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
String dummy=(String)vProp.elementAt(2);// point name
updatepoint(dummy, true, outletT, //T
            false, 0, false, 0, //P,x
            false, "dummy", 1);

//mise à jour du noeud en utilisant les méthodes génériques
vTransfo= new Vector[nBranches+1];
vPoints= new Vector[nBranches+1];
setupVector(airProcess, aval, 0, outletFlow, outletT, airP, 0);
setupVector(waterProcess, waterPoint, 1, waterFlow, waterOutT, airP, 0);
setupVector(mainProcess, aval, 2, outletFlow+waterFlow, outletT, airP, 0);
updateDivider(vTransfo,vPoints,outletT,qPrime&aval);

```

6) on estime enfin le NUT et les caractéristiques globales de l'échangeur équivalent

```

double DqPrimeML=(deltaQprime-deltaQprime0)/(Math.log(deltaQprime/deltaQprime0));
double NUT_ML=4.18*(dcti.waterT-waterOutT)/DqPrimeML;

double mCpc= DeltaQprime/(dcti.waterT-waterOutT);
double mCpf= DeltaQprime/(outletT-Tprime&mont);
double Tfe=Tprime&mont;
double Tfs=outletT;
double Tce=dcti.waterT;
double Tcs=waterOutT;
double Tfmin=Tfe, Tfmax=Tfs;
if (mCpc>mCpf){
    R=mCpf/mCpc;
    epsilon=epsi_NUT(NUT_ML,R);
    UA=mCpf*NUT_ML;
}
else{
    R=mCpc/mCpf;
    epsilon=epsi_NUT(NUT_ML,R);
    UA=mCpc*NUT_ML;
}

double Delta=(Tce-Tfs)*(Tce-Tfs)-(Tcs-Tfe)*(Tcs-Tfe);
if(Delta<0)Delta=-Delta;//modRG 04/12
if(Delta>1e-6)DTML=(Tce-Tfs+Tfe-Tcs)/(Math.log((Tce-Tfs)/(Tcs-Tfe)));
else DTML=Tce-Tfs;
if(DTML<0)DTML=-DTML;

```