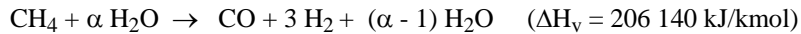


## Modélisation d'une pile SOFC dans Thermoptim

Soit  $\alpha$  le rapport molaire entre l'eau et le méthane ( $\alpha$  doit être supérieur à 1 pour que tout le méthane puisse être transformé).

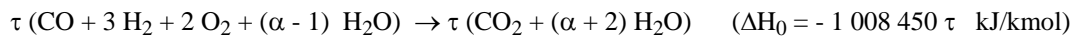
Du côté de l'anode, compte tenu du haut niveau de température de fonctionnement d'une pile SOFC, on peut considérer que tout le combustible est transformé par la réaction de vapocraquage :



On notera que l'un des avantages de cette technologie est qu'elle tolère parfaitement la présence de CO, alors que dans d'autres cas une réaction shift est nécessaire pour le convertir en CO<sub>2</sub>. Le rendement énergétique de l'ensemble est ainsi amélioré.

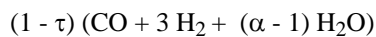
Ensuite, seule une fraction  $\tau$  (taux d'utilisation du combustible) est transformée dans la pile, le reste ressortant. La réaction donnant les espèces en sortie s'écrit globalement :

Pour la partie utilisée :

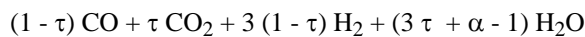


La chaleur de réaction  $\Delta H_0$  est ici calculée en considérant que l'eau produite reste à l'état gazeux compte tenu de la température.

Pour la partie non utilisée :

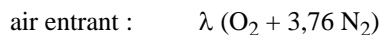


Au total, on a donc en sortie d'anode :



De la quantité utilisée, une fraction  $\varepsilon$  est directement convertie en électricité, et  $(1 - \varepsilon)$  est transformé en chaleur (dont une partie est utilisée pour le vapocraquage).

Par ailleurs,  $\lambda$  étant un paramètre représentatif de l'air entrant, de l'oxygène est prélevé sur l'air comburant du côté de la cathode



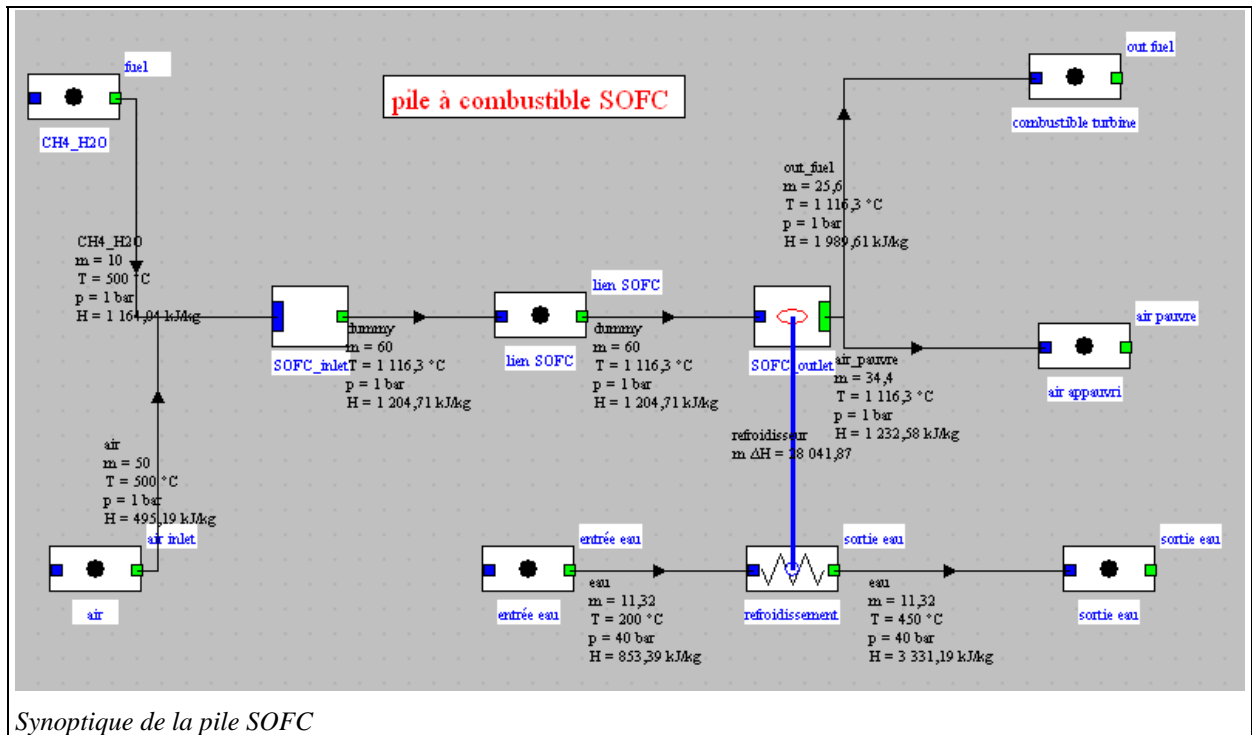
Le modèle que l'on peut retenir est alors le suivant :

- 1) la composition des espèces est donnée par la résolution des équations ci-dessus : on détermine les débits molaires de combustible humidifié et d'air en entrée, ce qui fournit les valeurs de  $\alpha$  et de  $\lambda$ , dont on déduit les débits molaires en sortie, les valeurs de  $\tau$  et de  $\varepsilon$  étant lues à l'écran ;
- 2) la chaleur libérée par la fraction  $\tau(1 - \varepsilon)$  du combustible sert à fournir l'énergie nécessaire au vapocraquage et à l'échauffement des gaz

L'enthalpie libérée est égale à  $\tau \Delta H_0$ . Elle se répartit entre de l'électricité ( $\varepsilon \tau \Delta H_0$ ), et de la chaleur pour le vapocraquage ( $\Delta H_v$ ) et l'échauffement des gaz ( $\tau(1 - \varepsilon) \Delta H_0 - \Delta H_v$ ).

Dans Thermoptim, la pile est représentée par un mélangeur externe connecté à un diviseur externe, les calculs étant effectués par ce dernier. Les classes s'appellent SOFCCH4inlet et SOFCCH4outlet.

La figure ci-dessous montre le synoptique de cette pile à combustible. Le paramétrage retenu est proche de celui du modèle de pile fonctionnant à l'hydrogène : température d'entrée des gaz 500 °C, débit 10 g/s pour le combustible, et 80 g/s pour l'air.



Synoptique de la pile SOFC

La figure ci-dessous montre l'écran du mélangeur amont, où apparaissent les paramétrages du modèle électrique, du taux d'utilisation du combustible et de la fraction de la puissance thermique extraite par le thermocoupleur.

noeud:  type:

veine principale:

isobare

| nom transfo | m abs | T (°C) | H        |
|-------------|-------|--------|----------|
| air         | 50    | 500    | 495,19   |
| CH4_H2O     | 10    | 500    | 1 164,04 |

**SOFC CH4 inlet**

b parameter:   **Ocooling fract.**

E parameter:

Jd parameter:

delta parameter:

tau/Jd parameter:

Ecran du mélangeur amont de la pile SOFC

L'écran du composant pile SOFC est présenté ci-dessous. On a pris un taux d'utilisation du combustible de près de 85%, correspondant au chiffre annoncé par Siemens et Westinghouse pour ce type de pile fonctionnant au gaz naturel.

noeud  type

veine principale  m global

lien SOFC  h global

isobare T global

| nom transfo      | m abs   | m rel   | T (°C)  | H        |
|------------------|---------|---------|---------|----------|
| air appauvri     | 34,4016 | 34,4016 | 1 116,3 | 1 232,58 |
| combustible t... | 25,5984 | 25,5984 | 1 116,3 | 1 989,61 |

SOFC CH4 outlet SOFC cooler

current intensity (A)  Number of cells

active surface (cm2)  fuel use rate : 0.867

heat released (W)  conversion efficiency : 0.548

electric power generated  voltage : 158.212

outlet temperature (K)

*Ecran du composant pile SOFC*

Les compositions des gaz que l'on obtient sont données ci-dessous :

| nom du composant | fraction molaire | fraction massique |
|------------------|------------------|-------------------|
| CH4 ` méthane    | 0,48             | 0,4511549         |
| H2O              | 0,52             | 0,5488451         |

*fuel, débit 10 g/s, PCI : 22 562 kJ/kg*

| nom du composant | fraction molaire | fraction massique |
|------------------|------------------|-------------------|
| N2               | 0,781            | 0,7555302         |
| Ar               | 0,009            | 0,01241636        |
| O2               | 0,21             | 0,2320534         |

*air, débit 80 g/s*

| nom du composant | fraction molaire | fraction massique |
|------------------|------------------|-------------------|
| CO               | 0,03265306       | 0,0410301         |
| CO2              | 0,2122449        | 0,4190309         |
| H2               | 0,09795918       | 0,008858684       |
| H2O              | 0,6571429        | 0,5310803         |

*combustible en sortie, débit 25,6 g/s, PCI : 1 477 kJ/kg*

| nom du composant | fraction molaire | fraction massique |
|------------------|------------------|-------------------|
| N2               | 0,9483251        | 0,9385143         |
| Ar               | 0,0109282        | 0,01542351        |
| O2               | 0,04074674       | 0,04606218        |

*air appauvri en O2, débit 64,4 g/s*

## Implémentation du modèle

Les principales modifications à apporter au modèle de pile sont affichées ci-dessous : il s'agit essentiellement du calcul de la composition du gaz en sortie d'anode et de celui de la puissance thermique dégagée, qui est diminuée de la chaleur de réaction nécessaire au vapocraquage.

```
//nombre de moles du fuel en sortie
double xCO2=tau;
double xH2O=3.*tau + alpha -1;
double xCO=1-tau;
double xH2=3*(1-tau);
double sigma=xCO2+xH2O+xCO+xH2;

Util.updateMolarComp(fuelComp, gasComp[0], xCO/sigma);
Util.updateMolarComp(fuelComp, gasComp[1], xCO2/sigma);
Util.updateMolarComp(fuelComp, gasComp[2], xH2/sigma);
Util.updateMolarComp(fuelComp, gasComp[3], xH2O/sigma);

double current = -Vtot*intens;
double Qlib= (-DHO*tau-DHvapo)*molFlowCH4+current;

double fractQthermo=Util.lit_d(inletSOFC.Qex_value.getText());//

double Qcool=Qlib*fractQthermo;
Qlib=Qlib-Qcool;
double mcool=inletSOFC.fuelFlow+inletSOFC.airFlow;

epsi = -current/(-DHO*tau-DHvapo)/molFlowCH4;
JLabel11.setText("conversion efficiency : "+Util.aff_d(epsi,3));
```

## Modélisation de la pile avec recyclage

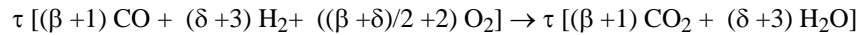
On a en entrée d'anode :



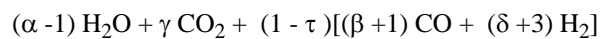
Le vapocraquage étant supposé complet (avec la condition  $\alpha \geq 1$ ), la composition après vapocraquage est :



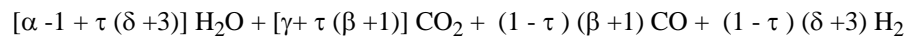
La partie réagissant a pour équation :



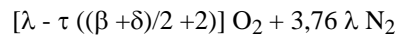
La partie ne réagissant pas est :



On en déduit la composition du gaz en sortie d'anode



En sortie de cathode, avec les notations habituelles, on a :



## Prise en compte de l'influence de la pression sur les performances de la pile

Les essais réalisés sur les cellules Siemens Westinghouse montrent que l'influence de la pression se fait essentiellement sentir au niveau de la tension en circuit ouvert  $E$ .

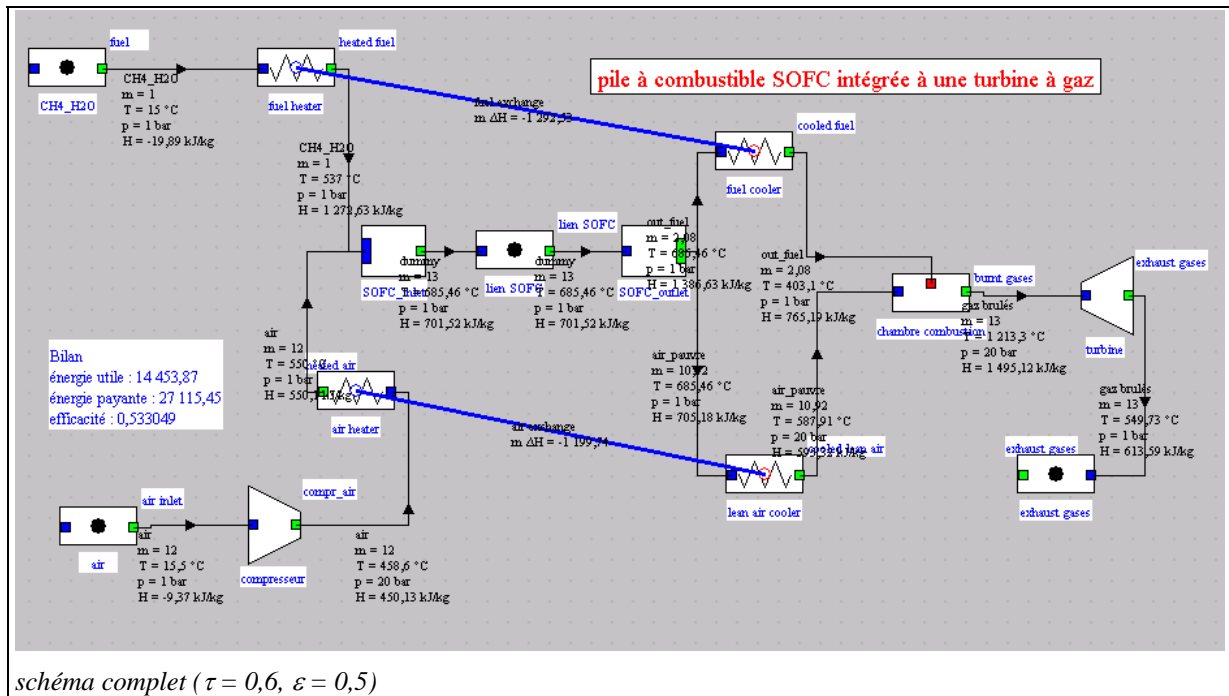
Nous avons retenu un modèle exponentiel par rapport à la pression :

$$E = E_0 + E_1 e^{-E_2 P}$$

Les paramètres de cette équation sont définis dans le mélangeur d'entrée.

## Modélisation de la pile associée à la turbine à gaz

Le modèle de pile associée à la turbine à gaz conduit aux résultats suivants :



Les gaz sortant de la turbine étant à 550 °C environ, il y a moyen de prévoir un cycle combiné qui pourrait augmenter le rendement global de 15 points environ, atteignant ainsi près de 70 %.

noeud : SOFC\_outlet      type : external divider

veine principale : lien SOFC      m global : 13      h global : 814,20870704      T global : 685,46122456

isobare

| nom transfo     | m abs   | m rel | T (°C) | H        |
|-----------------|---------|-------|--------|----------|
| lean air cooler | 10,9201 | 1     | 685,46 | 705,18   |
| fuel cooler     | 2,0799  | 1     | 685,46 | 1 386,63 |

ajouter une branche

supprimer une branche

**SOFC outlet**

fuel use rate : 0.6

conversion efficiency : 0.500

heat released : 2710.83

electric power generated : -8507.89

outlet temperature (K) : 958.62

*Ecran du composant pile SOFC*

## Amélioration du modèle

Pour améliorer la modélisation, il faudrait affiner les choses, en introduisant :

- l'humidification du combustible jusqu'à la haute température, le prélèvement étant fait sur la pile
- la récupération de l'eau contenue dans les fumées pour humidifier le combustible
- le vapocraquage comme composant séparé, sans doute isotherme avec thermocoupleur connecté à la pile par l'intermédiaire d'un échangeur (pas simple, la solution étant peut-être de prévoir un thermocoupleur partant de la pile, réchauffant un caloporteur dans une transfo échange, le caloporteur étant ensuite refroidi par le thermocoupleur du vapocraqueur, puis par l'humidification, et la pile "récupérant" par un autre thermocoupleur la chaleur excédentaire)
- un modèle de pile recevant du combustible vapocraqué

S'il fallait modéliser une pile PEM, le mécanisme de migration des espèces chimiques serait différent. A la cathode, des ions  $H^+$  se combindraient à l'oxygène de l'air pour former  $H_2O$ .

```
epsi=getEpsilon(inletSOFC);//méthode de calcul de epsilon
tau=Util.lit_d(fuelUseRate_value.getText());
double alpha=inletSOFC.fractH2O/(1-inletSOFC.fractH2O);

//débit molaire de CH4
double molFlowCH4=inletSOFC.fuelFlow/inletSOFC.fuelM*(1-inletSOFC.fractH2O);

//nombre de moles d'oxygène initial
double xO2=inletSOFC.fractO2;
double airMolFlow=inletSOFC.airFlow/inletSOFC.airM;

double O2MolFlow=airMolFlow*xO2;
double lambda=O2MolFlow/molFlowCH4;
System.out.println("molFlowCH4 : "+molFlowCH4+" airMolFlow : "+airMolFlow+" O2MolFlow : "+O2MolFlow);

//nombre de moles du gaz vapocraqué
/* double xCO=1.;
double xH2=3.;
double xH2O=(alpha-1);*/

//nombre de moles du fuel en sortie
double xCO2=tau;
double xH2O=3.*tau + alpha -1;
double xCO=1-tau;
double xH2=3*(1-tau);
double sigma=xCO2+xH2O+xCO+xH2;

//débit molaire de CO + 3 H2 utilisé
double molFlow=molFlowCH4*tau;

//débit molaire d'O2 consommé
double O2conso=molFlow*2.;

//débit restant d'O2
double O2restant=O2MolFlow-O2conso;
System.out.println("O2MolFlow : "+O2MolFlow+" O2conso : "+O2conso+" O2restant : "+O2restant);

//fraction du O2 initial utilisé
double epsiO2=(xO2-O2restant/airMolFlow)/xO2;
System.out.println("epsiO2 : "+epsiO2);
for(int i=0;i<inletSOFC.inletAirFractmol.length;i++){//mise à jour des fractions molaire de so
double x_out=inletSOFC.inletAirFractmol[i]/(1-epsiO2*xO2);
Util.updateMolarComp(airComp,inletSOFC.inletAirComp[i], x_out);
}
double xO2_out=(1-epsiO2)*xO2/(1-epsiO2*xO2);//fraction molaire de sortie en CO2
System.out.println("xO2_out : "+xO2_out);
Util.updateMolarComp(airComp,"O2", xO2_out);
outletAirSubstance.updateGasComposition(airComp);
```

```

flowOutFuel=volFlowOutFuel;//*outFuelM;

double totalFlow=flowAirPauvre+flowOutFuel;

double DHO=1008450;//kJ/kmol CH4
double DHvapo=206140;//kJ/kmol CH4

double Qlib=(tau*DHO*(1-epsi)-DHvapo)*molFlowCH4;
double current=-tau*DHO*epsi*molFlowCH4;
System.out.println("Qlib : "+Qlib+" current : "+current);
Q_value.setText(Util.aff_d(Qlib,2));
Current_value.setText(Util.aff_d(current,2));

double Htot=(inletSOFC.Htot+Qlib);///totalFlow;
System.out.println("Htot : "+Htot);
Taval=Util.dicho_T(this, 0, Htot, "outletTemp", inletSOFC.TamontAir, 2600, 0.01);
Tout_value.setText(Util.aff_d(Taval,2));

//mise à jour du noeud en utilisant les méthodes génériques
vTransfo= new Vector[nBranches+1];
vPoints= new Vector[nBranches+1];
setupVector(outletFuelProcess, outletFuelPoint, 0, flowOutFuel, Taval, 1, 0);
setupVector(outletAirProcess, outletAirPoint, 1, flowAirPauvre, Taval, inletSOFC.PamontAir,
setupVector(trashProcess, trashPoint, 2, inletSOFC.inletFuelFlow, inletSOFC.Tamont, 1, 0);
setupVector(linkProcess, linkPoint, 2, totalFlow, Taval, inletSOFC.PamontAir, 0);
updateDivider(vTransfo,vPoints,Taval,Htot/(inletSOFC.airFlow+inletSOFC.fuelFlow));
de.updateProcess(setEnergyTypes(linkProcess,current,DHO*epsi*molFlowCH4,0));

public double f_dicho(double T, double Htot,String fonc){

    if (fonc.equals("outletTemp")){
        double diff;
        Tpoint=T;
        outletFuelSubstance.CalcPropCorps(T,Ppoint, 1); //recalcul de l'état de sortie (fournit Hs
        getSubstProperties(outletFuelSubstanceName);
        double Hfuel=Hsubst;
        outletAirSubstance.CalcPropCorps(T,Ppoint, 1); //recalcul de l'état de sortie (fournit Hsu
        getSubstProperties(outletAirSubstanceName);
        // System.out.println("T "+T+" flowOutFuel*Hfuel : "+flowOutFuel*Hfuel+" flowAirPauvre*H
        diff=Htot-flowOutFuel*Hfuel-flowAirPauvre*Hsubst;
        return diff;
    }

    return 0;
}

```