Closiass 32.1 RKP-1000

Karolazar/Mesolvarde/Closiass PERENNOU Kaourant

Décembre 2023

Table des matières

1 Introduction

Le logiciel Closiass est basé sur des fonctions auxiliaires qui calculent chacune des grandeurs importantes de la cinétique du réacteur nucléaire. Ces grandeurs sont ensuite appelées par une dernière fonction d'asservissement pour piloter le réacteur de manière sûre et proche de la réalité. Ce document propose une explication de ces fonctions.

Closiass est écrit en PHP pour permettre son utilisation directe dans un navigateur. Les données d'entrée sont celles du réacteur RKP-1000 de Karolazar/Mesolvarde, très proche du réacteur 900 MW d'EDF. Dans les fonctions, les données considérées comme des données d'entrée de ce type sont notées `$_GET`.

Figure 1: Centrale de Saint-Laurent-des-Eaux dans le Loir-et-Cher fonctionnant avec deux réacteurs d'une puissance unitaire de 900 MW

1.1 Auteurs

Les programmes Closiass sont développés par l'équipe éponyme, notamment par PERENNOU Kaourant et WANG Tuan en tant que fondateurs des projets. Closiass comprend les solutions DEM, Monte-Carlo, ThermoComb, RKP-1000 et Mini-Tool.

L'auteur direct de ce document est PERENNOU Kaourant, responsable de la majorité du code. À titre de crédit, le logiciel DEM utilisé comme élément de preuve a été optimisé par BARRANGER Noé, HUTIN Gérémie et le Docteur NOUGARET Matthieu.

2 Formules des 4 facteurs

Listing 1: Formules des 4 facteurs

```php
function quatres_facteur($eff,$TR){
    $deformation = (293/$TR)**(1/2); //effet thermique sur la section efficace
    $epsilon=$_GET['epsilon']; // facteur correctif de fissions rapides
    $p=$_GET['p']; //facteur antitrappe
    $v5=$_GET['v5']*$deformation;
    $a5=$_GET['a5']*$deformation; // barns
    $a8=$_GET['a8']*$deformation; // barns
    $f5=$_GET['f5']*$deformation; // 582 barns
    $r=$_GET['r']; // rapport moderateur
    $ra=$_GET['ra']; // rapport absorbeur
    $e=$_GET['e']; //%
    $amod=$_GET['amod']; // barns
    $abarre=$_GET['abarre']; // barns
    $fn=($v5*$f5*$e)/($e*$a5+(1-$e)*$a8+$r*1*$amod + $ra*1*$abarre*$eff);
    $k_inf=$p*$epsilon*$fn;
    return $k_inf;
}
```

La fonction ci-dessus permet de calculer le coefficient k_∞, le facteur qui caractérise l'évolution de la population neutronique dans un espace infiniment étendu. Elle utilise la formule des 4 facteurs d'Enrico Fermi, qui implique plusieurs paramètres.

$$k_\infty = \epsilon \rho f \eta \tag{1}$$

Où :

k_∞ : Facteur multiplicatif en milieu infini
ϵ : Facteur de fission rapide
ρ : Facteur antitrape
f : Facteur d'utilisation thermique
η : Facteur de reproduction

avec

$$f\eta = \frac{n\text{Nombre de fissions thermiques}}{\text{Nombre de neutrons thermiques absorbés}} \tag{2}$$

Soit :

$$f\eta = \frac{\nu_5 N_5 \sigma_{f5} \Phi_{comb} V_{comb}}{N_5 \sigma_{a5} \Phi_{comb} V_{comb} + N_8 \sigma_{a8} \Phi_{comb} V_{comb} + N_{mod} \sigma_{amod} \Phi_{mod} V_{mod}} \tag{3}$$

Soit :

$$f\eta = \frac{\nu_5 \sigma_{f5} e}{e\sigma_{a5} + (1-e)\sigma_{a8} + \frac{\Phi_{comb} V_{comb} N_{comb}}{\Phi_{mod} V_{mod} N_{mod}} \sigma_{amod}} \tag{4}$$

S'il s'agit d'un réacteur avec des absorbeurs de neutrons comme ici, on a finalement :

$$f\eta = \frac{\nu_5 \sigma_{f5} e}{e\sigma_{a5} + (1-e)\sigma_{a8} + r_{mod}\xi\sigma_{amod} + r_{abs}\xi\sigma_{aabs}} \tag{5}$$

Où :

$$\xi = \frac{\Phi_{mod}}{\Phi_{comb}} \approx 1$$

r : Les rapports, modérateur et absorbateur

ν : Le nombre de neutrons émis par fission

σ : Les sections efficaces

e : Le taux d'enrichissement

Avec respectivement 5 pour l'uranium 235 et 8 pour l'uranium 238, mod pour le modérateur, abs pour les absorbateurs, a pour l'absorption et f pour la fission.

Sachant

$$\sigma_a = \sigma_f + \sigma_c \tag{6}$$

Les facteurs $\epsilon \approx 1,05$ et $\rho \approx 0,7$ sont fixés par la conception du cœur et ne changent que très peu au cours du temps.

Pour piloter le cœur RKP-1000, Closiass utilise donc la section efficace du modérateur par l'injection d'acide borique en début de réaction et le rapport d'absorption des absorbateurs de neutrons en insérant plus ou moins les barres de contrôle.

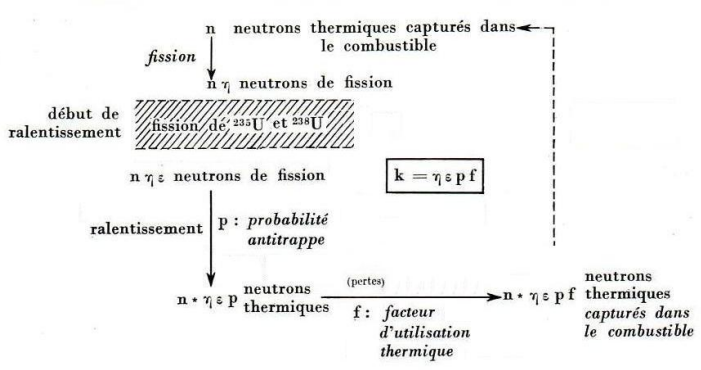

Figure 2: Illustration de la formule des 4 facteur d'Enrico Fermi

3 PNF

Listing 2: PNF

```
function PNF(){
    $L2=$_GET['L2']; // cm2
    $h_coeur=$_GET['h_coeur']; // cm
    $rayon_coeur = $_GET['rayon_coeur']; // cm
    $Bg=(3.14/$h_coeur)**2+(2.405/$rayon_coeur)**2;
    $PNF=1/(1+$L2*$Bg**2);
    return $PNF;
}
```

Pour déterminer le facteur de multiplication efficace, nous devons calculer la probabilité de non fuite (PNF). Dans le modèle Closiass, seuls les réacteurs cylindriques de type RKP-1000 sont pris en charge.

La probabilité de non-fuite est définie telle que :

$$PNF = \frac{1}{1 + L^2 B_g^2} \tag{7}$$

avec B_g^2 représentant le laplacien géométrique :

$$B_g^2 = \left(\frac{\pi}{h}\right)^2 + \left(\frac{2.402}{r}\right)^2 \tag{8}$$

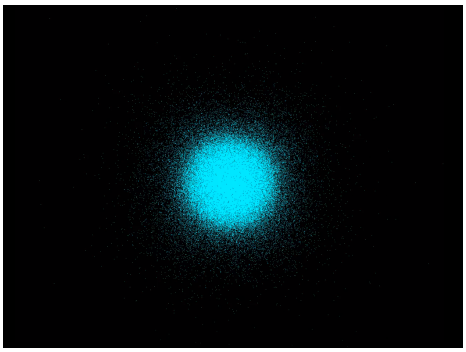

Figure 3: On peut vérifier la PNF et le k_∞ avec Closiass Monte-Carlo

4 Multiplication efficace

Listing 3: Coefficient de multiplication efficace

```
function efficient($k_inf,$PNF,$TR){
    $doppler =
        ($_GET['doppler']+45/1000*$_GET['Bore']+15)*10**-5;
        // -pcm/K
    $k_eff=$k_inf*$PNF;
    $k_eff = 1/(1/$k_eff - $doppler*($TR-293)); //effet
        thermique Doppler
    return $k_eff;
}
```

Une fois le coefficient de multiplication infini calculé et la probabilité de non-fuite obtenue, on peut calculer le coefficient de multiplication efficace. Cependant, il est alors important de prendre en compte l'effet Doppler thermique qui dépend directement d'un facteur constant et de la borication du modérateur.

$$k_{\text{eff}} = k_{\infty} \cdot PNF \tag{9}$$

Les formules pseudo-empiriques sur l'effet Doppler sont issues de REX et du livre Précis de neutronique de Paul REUSS.

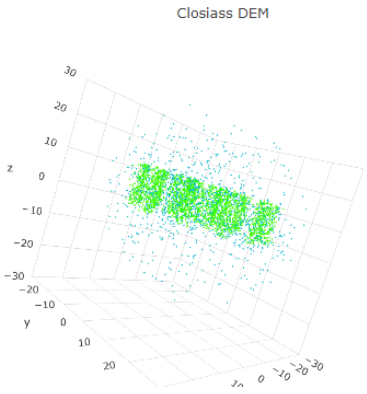

Figure 4: On peut vérifier le résultat avec Closiass DEM

5 Flux Neutronique

L'objectif de cette partie est d'expliquer comment calculer le flux neutronique dans Closiass.

5.1 Bilan neutronique

Si on pose l^* comme le temps de vie moyen lorsque le réacteur est critique, alors $l = k_{eff} l^*$, avec l le temps écoulé en moyenne entre la naissance et la disparition d'un neutron (dans le réacteur).

Il est alors possible d'écrire le bilan suivant. Soit, entre t et $dt + t$:

$$\begin{aligned} \text{Disparition : } & \frac{dt}{l} n(t) \\ \text{Apparitions : } & k_{eff}(1-\beta)\frac{dt}{l} n(t) + \Sigma \lambda_i C_i(t) dt \\ \text{Bilan : } & \frac{dn(t)}{dt} = \frac{k_{eff}(1-\beta) - 1}{l} n(t) + \Sigma \lambda_i C_i(t) \end{aligned}$$

Avec :

$\beta \approx 679$ pcm, Somme des rapports des groupes des précurseurs

$k_{eff}(1-\beta)\frac{dt}{l} n(t)$: Les neutrons prompts

$\Sigma \lambda_i C_i(t)$: Les neutrons retardés

Résoudre cette équation est relativement simple si on fait l'hypothèse d'une réactivité constante, on peut alors poser :

$$n(t) = n_0 e^{\omega t} \tag{10}$$

5.2 Calcule du flux

Si on fait une nouvelle hypothèse, celle de neutrons tous de même vitesse, on peut alors écrire :

$$\Phi(t) = v n(t) = v n_0 e^{\omega t} \tag{11}$$

Soit :

$$\Phi(t + \Delta t) = \Phi(t) e^{\omega \Delta t} \tag{12}$$

Avec un Δt tel que k_{eff} et v puissent être considérés comme constants.

Si on pose $\tilde{l} = l + \beta \bar{\tau}$, avec $\bar{\tau}$ la moyenne de l'inverse des constantes radioactives des précurseurs et admettant $\rho \approx k_{eff} - 1$ et $l^* \approx l$, on a :

$$\omega = \frac{\tilde{l}}{k_{eff} - 1} \tag{13}$$

5.3 Usage

Par commodité, on introduit souvent 2 grandeurs, la réactivité ρ et l'octave par minute Ω tels que :

$$\rho = \frac{k_{eff} - 1}{k_{eff}}$$
$$\Omega = \frac{60 \times \omega}{\ln(2)}$$

Ces deux paramètres permettent conjointement de déterminer et de mesurer la dynamique dans le réacteur.

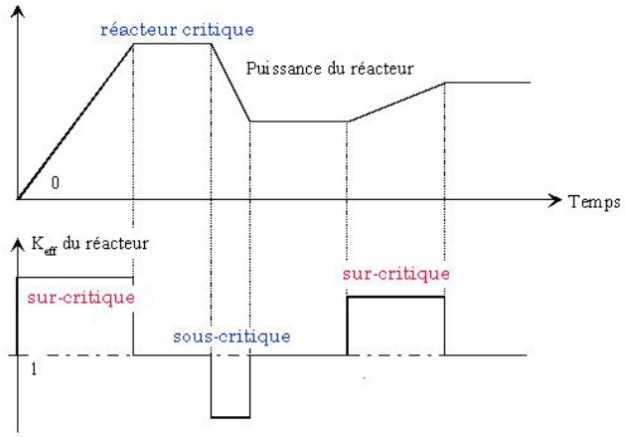

Figure 5: Illustration de la réactivité

5.4 Puissance sous-jacente

On peut alors calculer la puissance volumique du combustible si l'on connaît le flux neutronique surfacique. En effet, comme toutes les fissions sont induites (ce qui est une approximation valide pour les réacteurs commerciaux), on peut déterminer la section efficace macroscopique de fission. Ainsi, on peut calculer le taux de fission en $fissions.s^{-1}.cm^{-3}$ et en le multipliant par l'énergie de chaque fission, on obtient la puissance volumique du combustible en $J.s^{-1}.cm^{-3}$, soit $W.cm^{-3}$:

$$\tau = (N \times \sigma)(n \times v) = \Sigma\Phi$$
$$p_v = E_{\text{fission}} \times \tau$$

Avec Σ la section efficace macroscopique, définie comme la probabilité pour un neutron d'interagir avec une cible par unité de longueur, et τ, le taux de fissions (induites).

Cela conduit Closiass à ces fonctions :

Listing 4: Calcule de la puissance du coeur

```
function octave($k_eff){
    $tau=$_GET['tau']; //s
    $beta=$_GET['beta']; //sans dimension
    $l=$_GET['l']; //s
    $l_prec=$l+$beta*$tau;
    return ($k_eff-1)/$l_prec;
}

function reactivite($k_eff){
    $p=($k_eff-1)/$k_eff;
    return $p;
}

function phi($phi,$octave){
    $t=$_GET['t'];
    $phi=$phi*exp($octave*$t);
    return $phi;
}

function puissance_volumique($phi){
    $somme_f=$_GET['somme_f']; // cm-1 section efficace
        macroscopique
    $energie_de_fission=3.3*10**(-11);// Energie degagee
        par une fission en J
    return $phi*$somme_f*$energie_de_fission;
}
```

6 Pilotage du coeur

Listing 5: Pilotage

```
function inverse($PNF,$TR){
    $deformation = (293/$TR)**(1/2); //effet thermique sur
        la section efficace
    $epsilon=$_GET['epsilon']; // facteur correctif de
        fissions rapides
    $doppler =
        ($_GET['doppler']+45/1000*$_GET['Bore']+15)*10**-5;
        // -pcm/K
    $p=$_GET['p']; //facteur antitrappe
    $v5=$_GET['v5']*$deformation;
    $a5=$_GET['a5']*$deformation; // barns
    $a8=$_GET['a8']*$deformation; // barns
    $f5=$_GET['f5']*$deformation; // 582 barns
    $r=$_GET['r']; // rapport moderateur
    $ra=$_GET['ra']; // rapport absorbeur
    $e=$_GET['e']; //%
    $amod=$_GET['amod']; // barns
    $abarre=$_GET['abarre']; // barns

    $vrai_1 = 1/(1 + $doppler*($TR-293));
    $iv=$vrai_1/$PNF;
    $iv=$iv/($p*$epsilon);
    $eff=(($v5*$f5*$e)/$iv -
        ($e*$a5+(1-$e)*$a8+$r*1*$amod))/($ra*1*$abarre);
    return $eff;
}
```

Pour piloter le cœur, Clossias doit être en capacité de déterminer par le calcul l'enfoncement des barres à appliquer pour rendre le réacteur critique à tout instant. Pour ce faire, la fonction *inverse* répond à cela en inversant (d'où son nom) les formules présentées ci-dessus.

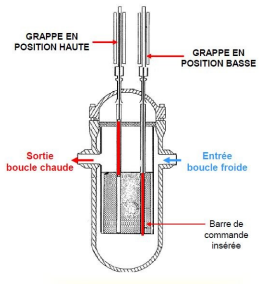

Figure 6: Illustration des barres

7 Thermodynamique

7.1 Introduction

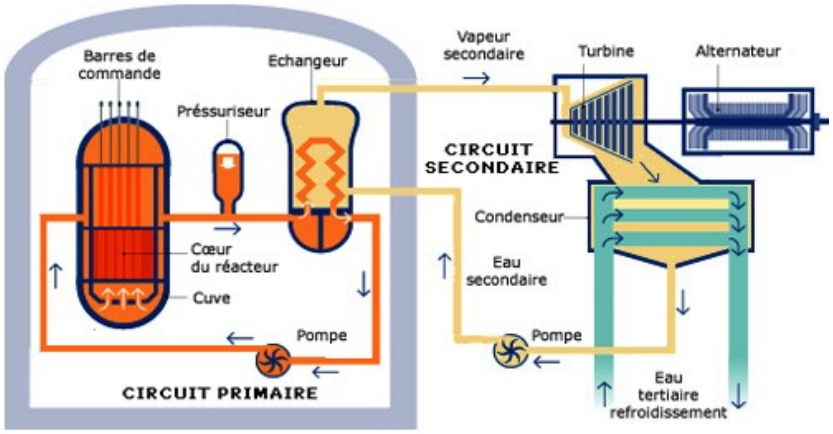

Figure 7: Illustration de la centrale

Nous avons calculé la puissance dite "thermique" du cœur du réacteur, mais le RKP-1000 a pour objectif de produire du courant à travers une turbine à vapeur (d'eau), et pour en calculer les performances, il faut modéliser les deux circuits principaux. Le circuit primaire comprend la cuve, le pressuriseur, 3 échangeurs et 3 pompes ; c'est lui qui chauffe. Le circuit secondaire, quant à lui, comprend 3 échangeurs, une pompe, la turbine (ou les dépendamment du maître d'ouvrage) et le condenseur.

Nous n'avons en liberté (pilotable) que la puissance du cœur, comme vu précédemment, et la puissance dissipée au niveau du condenseur.

Nous adopterons une approche systèmatique pour éviter de trop longue explication.

7.2 Cœur du réacteur

Système : cuve avec de l'eau du primaire à T_{in} en entrée et à T_{out} en sortie.

1er principe :

$$\Delta h + \Delta e = w + q \tag{14}$$

L'énergie mécanique est négligeable ici et les forces de pression ne travaillent pas du fait du pressuriseur. On a alors :

$$\begin{aligned} \Delta h &= q \\ D_m \Delta h &= P_{th} \\ D_m c_p \Delta T &= p_v V \end{aligned}$$

Soit :

$$\Delta T_{coeur} = \frac{p_v V}{D_m c_p} \tag{15}$$

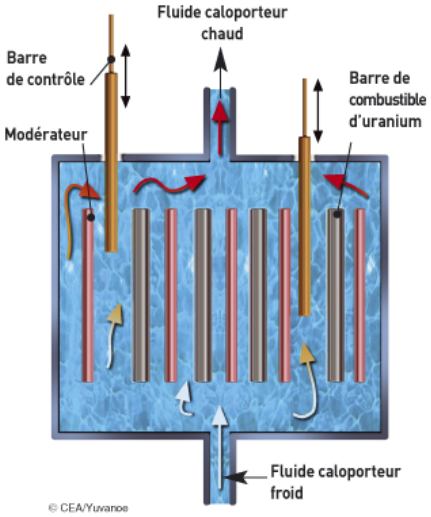

Figure 8: Schema simplifié du coeur

7.3 Échangeur côté primaire

Système : échangeur avec de l'eau du primaire à T_{in} en entrée et à T_{out} en sortie. On suppose que sur le moment T_{gv} est constant.

1er principe :

$$dh + de = \delta w + \delta q \tag{16}$$

L'énergie mécanique est négligeable ici et les forces de pression ne travaillent pas du fait du préssuriseur. On a alors :

$$\begin{aligned} dh &= \delta q \\ c_p dT &= -\frac{\Phi(T)}{m} dt \\ \frac{dT}{dt} &= \frac{T_{gv} - T}{m c_p R_{th}} \end{aligned}$$

On pose $\alpha = \frac{1}{m c_p R_{th}}$ et $R_{th} = \frac{e}{\lambda S}$.
Intégration :

$$T(t) = (T_{in} - T_{gv}) e^{-\alpha t} + T_{gv} \tag{17}$$

Sachant que l'eau a un temps de séjour $t_f = \frac{m}{D_m}$, on a :

$$T_{out} = (T_{in} - T_{gv}) e^{-\alpha t_f} + T_{gv} \tag{18}$$

Et donc (on choisit le signe positif volontairement) :

$$\Delta T_{gv} = T_{in} - T_{out} \tag{19}$$

7.4 Circuit primaire

Système : Primaire, et eau du primaire (hypothèse : sur le moment, les variables sont constantes)

L'eau subit un gain ΔT_{coeur} et perd ΔT_{gv} au cours d'une traverse de la boucle, qu'elle fait en $t_f = \frac{m}{D_m}$. On peut alors introduire :

$$\frac{\Delta T}{\Delta t} = k_T = \frac{\Delta T_{coeur} - \Delta T_{gv}}{\frac{m}{D_m}} \tag{20}$$

Soit :

$$\frac{\Delta T}{\Delta t} = k_T = \frac{D_m(\Delta T_{coeur} - \Delta T_{gv})}{m} \tag{21}$$

Compte tenu de nos hypothèses, on a :

$$\Delta T = k_T \Delta t \tag{22}$$

Figure 9: Illustration du circuit primaire

7.5 Circuit secondaire

Système : eau dans le secondaire, du condenseur au générateur de vapeur. (hypothèse : sur le moment, les variables sont constantes)

1er principe :

$$\Delta h + \Delta e = w + q \tag{23}$$

L'énergie mécanique est négligeable ici, et les forces de pression ne travaillent que dans la turbine. On a alors :

$$\begin{aligned} \Delta h &= q \\ D_m \Delta h &= P_{th} - P_{comd} \\ \Delta T &= \frac{P_{th} - P_{comd}}{D_m c_p} \end{aligned}$$

Soit :

$$\Delta T_{sec} = \frac{P_{th} - P_{comd}}{D_m c_p} \tag{24}$$

L'eau subit un gain ΔT_{sec} au cours d'une traverse de la boucle, qu'elle fait en $t_f = \frac{m}{D_m}$. On peut alors introduire :

$$\frac{\Delta T}{\Delta t} = k_T = \frac{\Delta T_{sec}}{\frac{m}{D_m}} \tag{25}$$

Soit :

$$\frac{\Delta T}{\Delta t} = k_T = \frac{D_m \Delta T_{sec}}{m} \tag{26}$$

Compte tenu de nos hypothèses, on a :

$$\Delta T = k_T \Delta t \tag{27}$$

⚠Remarque : La variable P_{comd} malgré son nom, représente la puissance dissipée par l'ensemble du circuit secondaire. Elle peut être interprétée comme la puissance du générateur de vapeur lorsque le condenseur est considéré comme négligeable, et inversement.

7.6 Combustible

Système : Le combustible

Si l'on considère les pastilles de combustible comme des cylindres parfaits et que la puissance volumique n'est due qu'aux réactions nucléaires, nous pouvons écrire les équations suivantes en utilisant le premier principe de la thermodynamique :

$$\begin{aligned} p_v V &= \Phi = J_Q S \\ J_Q &= p_v \frac{V}{S} = p_v \frac{r}{2} \\ P &= J_Q S \end{aligned}$$

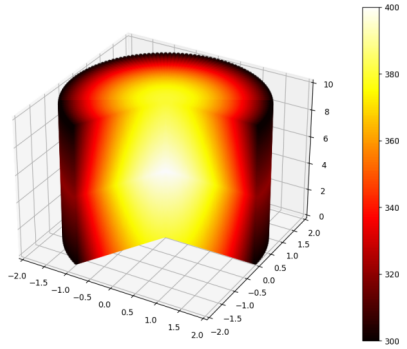

Figure 10: Calcule des hypothèses prises sur le combustible, Closiass Thermo-Comb

(Cela peut paraître inutile, mais cette réécriture est utile pour comprendre certaines fonctions de Closiass)

7.7 Cycle vapeur

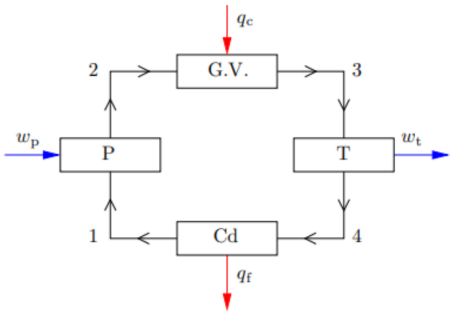

Figure 11: Cycle du CNPE

1er principe au générateur de vapeur

$$\Delta h + \Delta e = q_c + w \tag{28}$$

Il n'y a pas de travail dans le générateur de vapeur et on néglige les variations d'énergie mécanique.

$$\begin{aligned} h_3 - h_2 &= q_c \\ D_m(h_3 - h_2) &= P_{gv} \\ h_3 &= \frac{P_{gv}}{D_m} + h_2 \end{aligned}$$

1er principe à la turbine

$$\Delta h + \Delta e = q + w_t \tag{29}$$

Il n'y a pas de flux thermique dans la turbine et on néglige les variations d'énergie mécanique.

$$\begin{aligned} h_4 - h_3 &= w_t \\ P_t &= D_m(h_4 - h_3) \\ P_t &= D_m(h_4 - h_2) - P_{gv} \end{aligned}$$

Ces différentes équations justifient les approches simplifiées de Closiass dans la simulation du cœur du réacteur RKP-1000. Elles sont calculées par les fonctions de la manière suivante :

7.8 Programmes

Listing 6: Puissance primaire

```
function puissanceTh($JQ,$surface_combustible){
    return $JQ*$surface_combustible;
}
```

Listing 7: Puissance secondaire

```
function puissanceGV($TR,$Tgv,$puissance_volumique){
    $h=$_GET['h_coeur']*10**-2; // m
    $rayon = $_GET['rayon_coeur']*10**-2; //m
    $volume = 3.14*$h*$rayon**2; // m3 volume du cylindre
    $thermostat = $_GET['thermostat']; // K (20 C)
    $dt = $_GET['t']; // s
    $cp_vap = $_GET['Cp_vap']; // j/kg/K
    $m_secondaire = $_GET['masse_secondaire']; //kg
    $Pcomd_lim = $_GET['Plim']*10**6; // MW

    // Automate
    $dT_GV = $_GET['gradiant_GV'] * ($TR-$thermostat) /
        (591-$thermostat); //K
    $Tgv_consigne = $TR - $dT_GV; //K
    $Pcomd = $volume * $puissance_volumique -
        ($Tgv_consigne - $Tgv)/$dt * $m_secondaire * $cp_vap;
    //fin Automate

    if($Pcomd<0){ // hypothese : Le cricuit de
        refroidissement ne peut pas chauffer le GV
        return 0;
    }elseif($Pcomd > $Pcomd_lim){ // Le circuit de
        refroidissement a une puissance maximale qui, meme
        si l'automate le desire, ne peut etre depassee
        return $Pcomd_lim;
    }else{
        return $Pcomd;
    }
}
```

Listing 8: Puissance de la turbine

```
function puissanceTurbine($puissanceGV){
    $debit = $_GET['debit_massique'];
    $h2=$_GET['h2']; //kj/kg
    $h4=$_GET['h4']; //kj/kg
    $h3=$puissanceGV/$debit*10**-3+$h2; //kj/kg
    $wt=$h4-$h3; //kj/kg
    if($wt*$debit*10**3<0){
        return $wt*$debit*10**3; //W
    }else{
        return 0;
    }
}
```

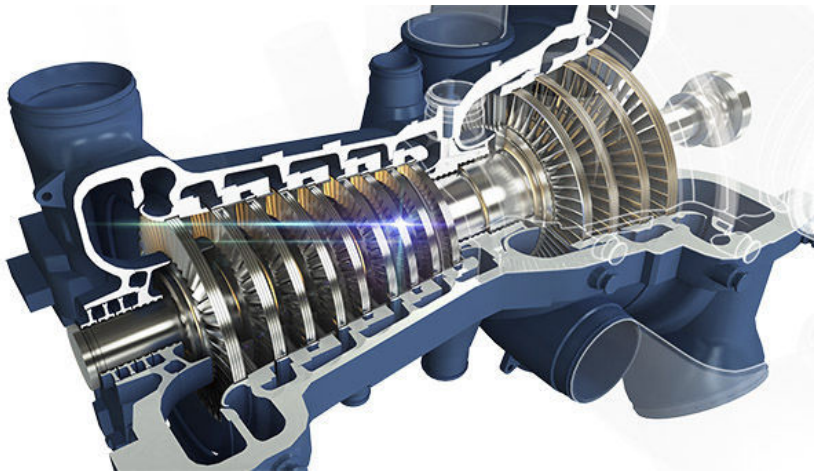

Figure 12: Arabelle, la tubine EDF

Listing 9: Températeur secondaire

```
function Tgv($Pcomd,$Tgv,$puissance_volumique){
    $h=$_GET['h_coeur']*10**-2; // m
    $rayon = $_GET['rayon_coeur']*10**-2; //m
    $volume = 3.14*$h*$rayon**2; // m3 volume du cylindre
    $dt = $_GET['t']; // s
    $cp_vap = $_GET['Cp_vap']; // j/kg/K
    $m_secondaire = $_GET['masse_secondaire']; //kg

    $Tgv =  $Tgv + $dt * ($volume * $puissance_volumique -
        $Pcomd)/($m_secondaire * $cp_vap);
    return $Tgv;
}
```

Listing 10: Températeur primaire

```
function TR($TR,$Tgv,$puissance_volumique) {
    $h=$_GET['h_coeur']*10**-2; // m
    $rayon = $_GET['rayon_coeur']*10**-2; //m
    $volume = 3.14*$h*$rayon**2; // m3 volume du cylindre
    $thermostat = $_GET['thermostat']; // K (20 C)
    $cp = $_GET['Cp']; //5829 j/kg/K
    $Debit_massique = $_GET['Dm']; // kg/s
    $masse = $_GET['masse_primaire']; // kg
    $dt = $_GET['t']; // s

    $lambda =  $_GET['lambda'];
    $surface_gv = $_GET['surface_gv']; // m2
    $e = $_GET['ep']; // m
    $Debit_massique_gv = $_GET['Dm_gv']; //kg/s

    $Tf = ($TR - $Tgv) * exp(-($lambda*$surface_gv) /
        ($cp*$Debit_massique_gv*$e)) + $Tgv;
    $dT_GV = $TR - $Tf;

    $thermo_forcage = ($puissance_volumique * $volume)/($cp
        * $masse) - $Debit_massique * $dT_GV / $masse;
    return $TR + $thermo_forcage*$dt;
}
```

8 Asservissement

8.1 Fonction d'asservissement

Comme tout réacteur commercial, le RKP-1000 doit être asservi pour répondre à la demande en électricité de manière sûre. Pour cela, le logiciel Closiass fait appel aux différentes fonctions vues précédemment et asservit en mode linéaire le réacteur vers une puissance volumique déterminée à l'avance et une température cible pour le bon fonctionnement de la centrale. Closiass a le droit d'utiliser une amplitude dite "limite" autour de la criticité pour assurer un état sûr, contrôlable, et dans les domaines de validité de nos hypothèses.

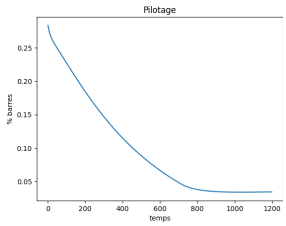

Figure 13: Illustration de l'asservissement

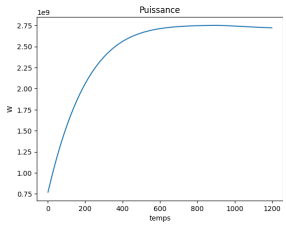

Figure 14: Illustration de l'évolution de la puissance suivant l'asservissement

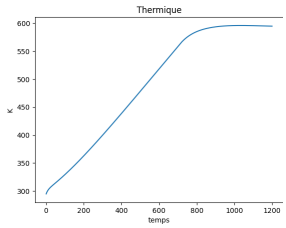

Figure 15: Illustration de l'évolution de la température suivant l'asservissement

Listing 11: Asservissement

```
while($i<$total){
    $k_inf=quatres_facteur($eff,$TR);
    $k_eff=efficient($k_inf,$PNF,$TR);
    $octave=octave($k_eff);
    $p=reactivite($k_eff);
    $phi=phi($phi,$octave);
    $pv=puissance_volumique($phi);
    $JQ=JQ($pv*10**6);
    $TR=TR($TR,$Tgv,$pv*10**6);
    $Pgv=puissanceGV($TR,$Tgv,$pv*10**6);
    $Tgv=Tgv($Pgv,$Tgv,$pv*10**6);
    $Pth=puissanceTh($JQ,$surface_combustible);
    $Pt=puissanceTurbine($Pgv);
    //Asservissement
    if($pv<$pv_consigne and $TR<$T_consigne){
        if($securite>0){
            if($eff-$amplitude*($pv_consigne-$pv)/$pv>0){
                $eff_ad=$eff_ad -
                    $amplitude*($pv_consigne-$pv)/$pv;
                $eff = inverse($PNF,$TR) + $eff_ad;
                $eff_pu = 0;
            }else{
                $eff=0;
            }
        }else{
            $eff=inverse($PNF,$TR) -
                $amplitude*($pv_consigne-$pv)/$pv;
        }
    }elseif ($k_eff>0 or $phi>10**15) {
        if($securite>0){
            if($eff+$amplitude*($pv-$pv_consigne)/$pv<100){
                $eff_pu=$eff_pu +
                    $amplitude*($pv-$pv_consigne)/$pv;
                $eff = inverse($PNF,$TR) + $eff_pu;
                $eff_ad = 0;
            }else{
                $eff=100;
            }
        }else{
            $eff=inverse($PNF,$TR) + $amplitude *
                ($TR-$T_consigne)/$TR;
        }
    }

    $i=$i+1;
}
```

8.2 Sûreté

À cela s'ajoutent quelques consignes évidentes de bon fonctionnement

Listing 12: Sûreté

```
1     if($securite<2){
2         if($TR>900 or $k_eff>1.7 or -$Pt*10**-6>2000 or
              $phi>10**16 or $SCRAM==1){
3             $eff=inverse($PNF,$TR)*2;
4             $SCRAM=1;
5         }
6     }
```

La variable "sécurité" définit le profil de sécurité que l'opérateur a enregistré dans le modèle. Cela permet de désactiver les mesures de sûreté à des fins de tests lors de la simulation (uniquement bien entendu).

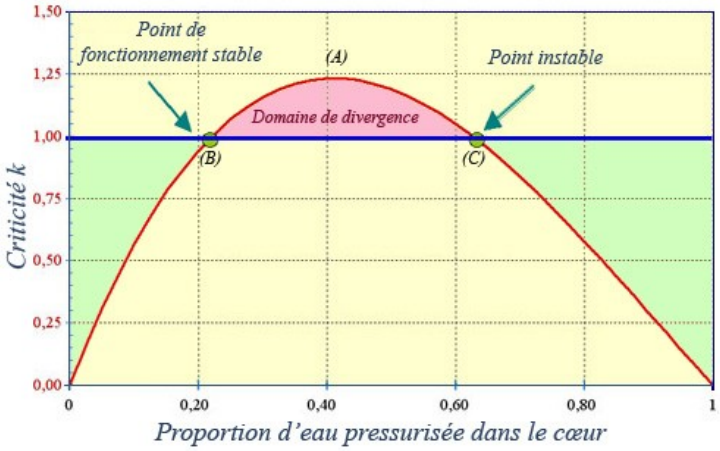

Figure 16: Domaine de sûreté du réacteur

9 Utilisation

Le logiciel Closiass est développé sur le web et possède donc une interface accessible par ce dernier. Elle se décompose en deux parties : un tableau d'entrée pour le modèle et le système de résolution en lui-même. Il est constitué de cinq sous-fenêtres et d'un tableau. Respectivement, ces fenêtres montrent la puissance, la température, un moniteur SCADA, l'enfoncement des grappes, une distribution de puissance, l'octavemètre et la réactivité.

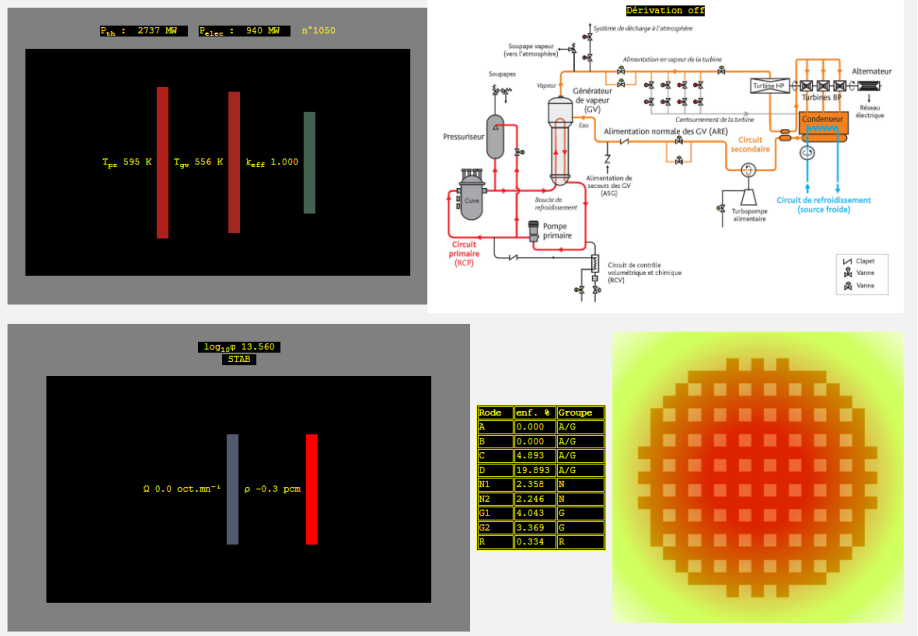

Figure 17: Illustration de Closiass 32.1

10 Conclusion

Le programme Closiass permet de simuler en temps réel des réacteurs, dont le RKP-1000 pour lequel il a été développé de manière approchée, offrant ainsi une vision rapide de la viabilité d'un modèle de réacteur. Cela en fait un outil éducatif et utile, tant en termes de suivi que de gestion

11 Sources

Références

[1] REUSS Paul, Précis de neutronique

[2] MAYET Frédéric, Physique nucléaire appliquée, Thermodynamique appliquée à l'énergétique

[3] SOLDEVILA Michel, Neutronique