

Utilisation d'OpenFOAM pour la réalisation d'un code de combus- tion 3D

Aymeric LAMORLETTE
Dominique MORVAN



Introduction

- ⇒ Pourquoi utiliser OpenFOAM ?
 - Aspect programmation
 - Aspect économique
 - Aspect scientifique
- ⇒ Applications
- ⇒ Illustration : propagation sans vent



Aspect programmation

- ⇒ CFD « classique » :
 - GUI, structure algorithmique figée, modifications limitées des systèmes à résoudre

- ⇒ OpenFOAM :
 - Bibliothèque d'objet pour la résolution discrète de systèmes d'EDP, suggestion de structures algorithmiques

- ⇒ Écriture (contrôle) de l'algorithmique
 - Lecture de données, discrétisation, résolution, mise à jour...



```

J().internalField() = radiation->J_() ;

// Calcul des termes d'Ächanges
RAD = LAD * ( J - 4. * Foam::constant::physicoChemical::sigma * pow4(Tsolide) ) / 4. ;

Qconv = ((sign(LAD-dimensionedScalar("small",dimensionSet(0,-1,0,0,0,0,SMALL))+1)/2) * ( thermo.T() - Tsolide ) * rho * dimensionedScalar("Cp_air_film",dimensionSet(0,
* LAD / ( pow( mag(U) * D_0 / dimensionedScalar("nu_air_film",dimensionSet(0,2,-1,0,0,0,0),0.00002) , ( 1.0 - m_chi ) ) + SMALL ) ) ;

volScalarField Wpyr = ((sign(LAD-dimensionedScalar("small",dimensionSet(0,-1,0,0,0,0,SMALL))+1)/2) * ((sign(Ys_sec-SMALL)+1)/2) * rho_solide * LAD / ( surf_spe + dime
volScalarField Wchar = ((sign(LAD-dimensionedScalar("small",dimensionSet(0,-1,0,0,0,0,SMALL))+1)/2) * ((sign(Ys_char-SMALL)+1)/2) * 0.5 * LAD * rho * Y[0] * dimensionior
volScalarField Wvap = pos( Qconv + RAD ) * ( ( sign( Ys_hu - SMALL ) + 1 ) / 2 )*( ( sign ( Tsolide - dimensionedScalar("T100C",dimensionSet(0,0,0,1,0,0,0),373.15) ) +

// Calcul prealable de Cp solide d'apres Cpsec, Cpeau
volScalarField CP_solide =
pos(Ys_sec) * ( Ys_sec + SMALL ) * dimensionedScalar("Cp_bois",dimensionSet(0,2,-2,-1,0,0,0),CP_bois) +
pos(Ys_hu) * ( Ys_hu + SMALL ) * dimensionedScalar("Cp_eau",dimensionSet(0,2,-2,-1,0,0,0),CP_eau) +
pos(Ys_char)* ( Ys_char + SMALL ) * dimensionedScalar("Cp_char",dimensionSet(0,2,-2,-1,0,0,0),CP_char) +
pos(Ys_sol) * ( Ys_sol + SMALL ) * dimensionedScalar("Cp_sol",dimensionSet(0,2,-2,-1,0,0,0),CP_sol);

// Resolution explicite de Tsolide (02)

Tsolide = Tsolide * ( 1. - runTime.deltaT()*runTime.deltaT()/runTime.deltaT0()/runTime.deltaT0() )

+ runTime.deltaT()*runTime.deltaT()/runTime.deltaT0()/runTime.deltaT0() * Tsolide.oldTime()

+ runTime.deltaT() * ( 1. + runTime.deltaT()/runTime.deltaT0() ) * ( surf_spe / ( ( LAD + dimensionedScalar("small",dimensionSet(0,-1,0,0,0,0,SMALL) ) * rho_solide *

(

// terme d'echange convectif entre phases
Qconv

// le termer Grad
+ RAD

// terme puit du Ä la pyrolyse
- Wpyr * dimensionedScalar("Delta_h_pyr",dimensionSet(0,2,-2,0,0,0,0),Delta_h_pyrolyse)

// terme puit du au charring
- Wchar * dimensionedScalar("Delta_h_char",dimensionSet(0,2,-2,0,0,0,0),Delta_h_charring)

// terme puit du Ä la vaporisation
- Wvap * dimensionedScalar("Delta_h_vap",dimensionSet(0,2,-2,0,0,0,0),Delta_h_vapeur)

// terme source d'ignition
+ Source_Ts

);

Tsolide.correctBoundaryConditions();

Info << "Solving for Tsolide (explicit)" << endl;

```

Intérêt économique : Codes de combustion = nécessité de grandes ressources informatiques

- ⇒ parallélisation, serveurs de calculs
- ⇒ OpenFOAM
 - licence GNU, plateforme LINUX
 - Bonne scalabilité
- ⇒ Dépendance financière limitée pour le maintien du code dans le temps



Intérêt scientifique

- ⇒ Méthodes numériques disponibles récentes et robustes
 - Ré-écriture des codes préférable à la mise à jour d'anciens codes laboratoires
- ⇒ Possibilité de calculs 3D
 - Études de phénomènes physiques impliquant des effets 3D (contre-feux, vent et pentes non colinéaires, effets locaux à rayon de courbure faible...)
- ⇒ Études de sensibilité et prévention



Application : ajustement de codes grandes échelles

- ➔ Codes grandes échelles : existence de paramètres empiriques (corrélations)
- ➔ Ajustement des corrélations à partir d'études paramétriques effectuées sur le code de combustion
- ➔ Application à un modèle obtenu par bilan intégral



Propagation sans vent

