DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

Utilisation d'Alien pour un solveur de diffusion sur des mailles mixtes dans le code A3

CEA/DAM/DIF

Rémi Chauvin

Rencontres Arcane, jeudi 5 mars 2020

Plan de la présentation

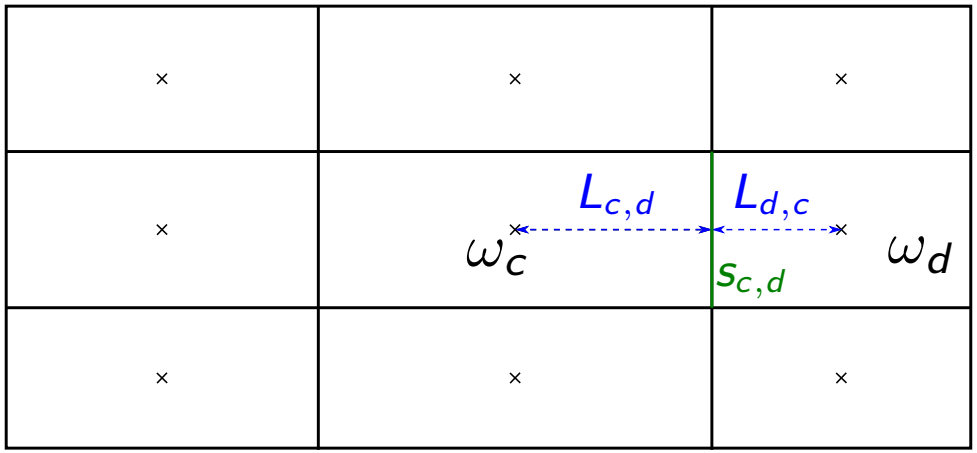
1. Modèles et méthodes numériques
2. Branchement à Alien pour l'inversion de la matrice Jacobienne
3. Conclusion

Plan de la présentation

Modèle de diffusion radiative simplifié

$$\rho \frac{d}{dt}\left(\frac{E}{\rho}\right) + \boldsymbol{\nabla} \cdot \boldsymbol{F} = S(T), \tag{1}$$

- ρ : densité
- T : température radiative
- $E = aT^4$: énergie radiative volumique
- a : constante de Stefan
- $\boldsymbol{F} = -\lambda(T)\boldsymbol{\nabla} T^{\eta}$: flux radiatif
- η : $17/2$
- $\lambda(T)$: coefficient de diffusion (fonction non linéaire de T)
- $S(T)$: terme source (fonction non linéaire de T)



Discrétisation volumes finis + schéma Euler implicite

$$\rho_c^{n+1}\left(\frac{a\left(T_c^{n+1}\right)^4}{\rho_c^{n+1}}-\frac{a\left(T_c^{n}\right)^4}{\rho_c^{n}}\right)-\frac{1}{|\omega_c|}\sum_{d\in\mathcal{C}(c)}\frac{\left(T_c^{n+1}\right)^\eta-\left(T_d^{n+1}\right)^\eta}{\frac{L_{c,d}}{\lambda\left(T_c^{n+1}\right)}+\frac{L_{d,c}}{\lambda\left(T_d^{n+1}\right)}}s_{c,d}=S\left(T_c^{n+1}\right)$$

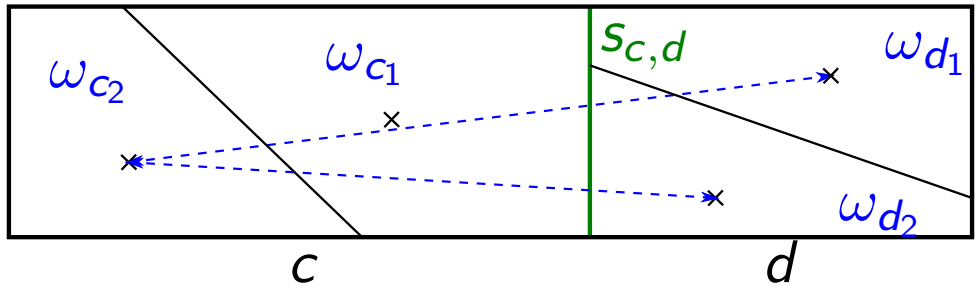


Schéma pour les mailles mixtes

- Échange entre une `envcell` et les `envcells` des mailles voisines au prorata des volumes partiels
- Pas d'échange entre les `envcells` d'une même maille

$$\rho_{c_\alpha}^{n+1}\left(\frac{a\left(T_{c_\alpha}^{n+1}\right)^4}{\rho_{c_\alpha}^{n+1}}-\frac{a\left(T_{c_\alpha}^{n}\right)^4}{\rho_{c_\alpha}^{n}}\right)-\frac{1}{|\omega_{c_\alpha}|}\sum_{d\in\mathcal{C}(c)}\sum_{\beta\in\mathcal{E}(d)}\frac{|\omega_{c_\alpha}|}{|\omega_c|}\frac{|\omega_{d_\beta}|}{|\omega_d|}\frac{\left(T_{c_\alpha}^{n+1}\right)^\eta-\left(T_{d_\beta}^{n+1}\right)^\eta}{\frac{L_{c,d}}{\lambda\left(T_{c_\alpha}^{n+1}\right)}+\frac{L_{d,c}}{\lambda\left(T_{d_\beta}^{n+1}\right)}}s_{c,d}=S\left(T_{c_\alpha}^{n+1}\right)$$

- Algorithme de résolution : méthode de Newton

↳ Nécessité d'assembler une matrice Jacobienne de dimension $N_{\text{envcells}} \times N_{\text{envcells}}$

Plan de la présentation

1. Modèles et méthodes numériques

2. Branchement à Alien pour l'inversion de la matrice Jacobienne
   - Numérotation des mailles mixtes
   - Allocation des structures Alien pour l'indexeur
   - Remplissage et inversion

3. Conclusion

Numérotation des mailles mixtes

- Par défaut : la matrice Alien est construite sur les celulles du maillage
- Utilisation d'un indexeur *ad-hoc* pour le problème défini aux `envcells`
- Difficulté : identifier chaque `envcell` avec un **identifiant unique**
  - Première solution : utilisation du `componentUniqueId()` d'Arcane
  - ↳ Problème : allocation d'un tableau d'indexage aussi grand que le plus grand `componentUniqueId` ⇒ consommation inutile de la mémoire
  - Solution retenue : algorithme de numérotation dans A3

e	`envcell`
p	processeur (sous-domaine Arcane)
I_e	Indice global de l'`envcell` e
I_e^p	Indice local de l'`envcell` e pour le processor p
$N_{\max}$	Nombre maximal d'`envcells` par processeur

$$I_e = p \times N_{\max} + I_e^p \tag{2}$$

```
Int32 nb_ids(envcell_vector.size());
m_max_envcells_per_proc = parallel_mng->reduce(Parallel::ReduceMax, nb_ids);

Int32 sub_domain_id(subDomain()->subDomainId());
Integer cpt(0);
Arcane::Int64UniqueArray ids;
for (auto&& envcell : envcell_vector) {
  Int64 unique_id = sub_domain_id * m_max_envcells_per_proc + cpt;
  (*m_a3_indices)[envcell] = unique_id;
  ids.add(unique_id);
  cpt++;
}
```

Allocation des structures Alien pour l'indexeur

```
m_simple_abstract_family = new Alien::SimpleAbstractFamily(ids, m_index_mng);
Alien::IItemIndexManager::ScalarIndexSet index_set = m_index_mng->buildScalarIndexSet(
    index_name, *m_simple_abstract_family);
m_index_mng->prepare();
m_indexes = m_index_mng->getIndexes(index_set);
m_space = new Alien::Space(m_index_mng);

Int32UniqueArray all_local_ids(m_simple_abstract_family->size());
m_simple_abstract_family->allLocalIds(all_local_ids);

Int64UniqueArray all_unique_ids(all_local_ids.size());
m_simple_abstract_family->uids(all_local_ids, all_unique_ids);

Integer uids_size(all_unique_ids.size());
for (Integer ii(0) ; ii < uids_size ; ++ii)
  m_unique_id_to_local[all_unique_ids[ii]] = ii;
```

- `m_a3_indices` : tableaux d'entiers indicés par des `envcells` qui donne l'indice global de chaque `envcell` pour A3
- `m_unique_id_to_local` : tableaux d'entiers indicés par des entiers qui convertit l'indice global A3 d'une `envcell` en indice local pour Alien
- `m_indexes` : tableaux d'entiers indicés par des entiers qui convertit l'indice local pour Alien en indice global pour Alien

↳ Indice global pour Alien:
`m_indexes[m_unique_id_to_local[(*m_a3_indices)[envcell]]];`

Remplissage et inversion

$$\rho_{c_\alpha}^{n+1}\left(\frac{a\left(T_{c_\alpha}^{n+1}\right)^4}{\rho_{c_\alpha}^{n+1}}-\frac{a\left(T_{c_\alpha}^{n}\right)^4}{\rho_{c_\alpha}^{n}}\right)-\frac{1}{|\omega_{c_\alpha}|}\sum_{d\in\mathcal{C}(c)}\sum_{\beta\in\mathcal{E}(d)}\frac{|\omega_{c_\alpha}|}{|\omega_c|}\frac{|\omega_{d_\beta}|}{|\omega_d|}\frac{\left(T_{c_\alpha}^{n+1}\right)^\eta-\left(T_{d_\beta}^{n+1}\right)^\eta}{\frac{L_{c,d}}{\lambda\left(T_{c_\alpha}^{n+1}\right)}+\frac{L_{d,c}}{\lambda\left(T_{d_\beta}^{n+1}\right)}}s_{c,d}=S\left(T_{c_\alpha}^{n+1}\right)$$

Pour un couple d'`envcells` (c_α, d_β)

- `i_l` : indice local de c_α pour Alien
- `i_g` : indice global de c_α pour Alien
- `j_g` : indice global de d_β pour Alien
- `m_builder` : `Alien::DirectMatrixBuilder`
- `m_matrix_data` : `Alien::MatrixData`
- `rhs` : `std::vector<Real>`

Remplissage de la matrice et du second membre

- `(*m_builder)(i_g, j_g) = ... ; //` (ou `+=`)
- `rhs[i_l] = ...;`

Inversion du système linéaire

```
getLinearSolverStrategy()->solve(m_matrix_data, rhs, sol);
```

Plan de la présentation

1. Modèles et méthodes numériques
2. Branchement à Alien pour l'inversion de la matrice Jacobienne
3. Conclusion

Conclusion

Indexeur particulier pour les mailles mixtes

- Nécessité de remplir l'indexeur `envcell` par `envcell`

↳ Possibilité de travailler sur une partie du maillage

- Recalcul de l'indexeur si modification de la topologie des mailles mixtes

Travaux futurs

- Amélioration de l'algorithme d'indiçage A3 (sans utiliser le maximum des envcells par processeur)
- Ré-allocation des structures Alien uniquement quand cela est nécessaire