M-DrKIC
19/11/02

Modifications algorithmiques pour une meilleure prise en compte des termes de forces extérieures.

1 Calcul du gradient de pression par gradrc

On se place dans la problématique d'une équation du type :

$$\rho \frac{\partial}{\partial t}\underline{u} = -\underline{\text{grad}}\ P + \underline{f} + \dots \tag{1}$$

On décompose la pression en pression hydrostatique P^h et pression dynamique P'.

$$P = P^h + P' \tag{2}$$

On s'intéresse à l'interpolation de la pression à la face dans le calcul du gradient. On conserve l'interpolation de base, mais uniquement pour la partie dynamique de la pression, la partie hydrostatique étant supposée en équilibre avec le terme source $\underline{f}$ ($\underline{\text{grad}}\ P^h = \underline{f}$).

$$P'_F = \alpha P'_I + (1-\alpha)P'_J + \varepsilon\frac{1}{2}\underline{OF}.(\underline{\text{grad}}\ P'_I + \underline{\text{grad}}\ P'_J) \tag{3}$$

ε vaut 0 si on ne reconstruit pas les non orthogonalités et 1 sinon. On a donc :

$$\begin{aligned} P_F &= P^h_F + P'_F \\ &= P^h_F + \alpha P'_I + (1-\alpha)P'_J + \varepsilon\frac{1}{2}\underline{OF}.(\underline{\text{grad}}\ P'_I + \underline{\text{grad}}\ P'_J) \\ &= P^h_F + \alpha P_I + (1-\alpha)P_J - \alpha P^h_I - (1-\alpha)P^h_J + \varepsilon\frac{1}{2}\underline{OF}.(\underline{\text{grad}}\ P_I - \underline{f}_I + \underline{\text{grad}}\ P_J - \underline{f}_J) \end{aligned} \tag{4}$$

En notant que
$P^h_I = P^h_F + \underline{FI}.\underline{f}_I$,
$P^h_J = P^h_F + \underline{FJ}.\underline{f}_J$,
$\alpha\underline{FI} + (1-\alpha)\underline{FJ} = -\underline{OF}$
on obtient que :

$$\begin{aligned} P_F = \alpha P_I + (1-\alpha)P_J + \varepsilon\frac{1}{2}\underline{OF}.(\underline{\text{grad}}\ P_I + \underline{\text{grad}}\ P_J) \\ - \alpha\underline{FI}.(\underline{f}_I - \varepsilon\underline{f}_F) - (1-\alpha)\underline{FJ}.(\underline{f}_J - \varepsilon\underline{f}_F) \end{aligned} \tag{5}$$

avec $\underline{f}_F = \frac{1}{2}(\underline{f}_I + \underline{f}_J)$

On retrouve la formule usuelle, complétée par des termes correctifs.

Cas des faces de bord

Dans le cas d'une condition de Dirichlet, pas de problème. Dans les autres cas, on met usuellement une condition de flux nul. Dans notre approche, on garde cette condition de flux nul, mais uniquement pour la partie dynamique de la pression.

$$\begin{aligned} P_F &= P_F^h + P_F' \\ &= P_F^h + P_{I'}' \\ &= P_F^h + P_{I'} - P_{I'}^h \end{aligned} \tag{6}$$

Or, on peut écrire qu'à l'ordre 1 on a :

$$P_{I'}^h = P_F^h + \underline{FI'}.\underline{\mathrm{grad}}\ P_I^h = P_F^h + \underline{FI'}.\underline{f}_I \tag{7}$$

On en déduit donc :

$$P_F = P_{I'} - \underline{FI'}.\underline{f}_I \tag{8}$$

Pour combiner les cas Dirichlet ou flux nul, avec ou sans reconstruction, on a donc finalement :

$$P_F = \mathtt{INC} \times A_F + B_F \left[P_{I'} - \underline{FI'}.\underline{f}_I\right] \tag{9}$$

2 Calcul du gradient de pression par gradmc

On reprend ici les notations de la notice consacrée à gradmc. On calcule le gradient $\underline{G}_{c,i}$ au centre de la cellule i à partir d'estimations $\underline{G}_{f,ij} \cdot \underline{d}_{ij}$ et $\underline{G}_{fb,ik} \cdot \underline{d}_{b,ik}$. On prend :

$$\underline{d}_{ij} = \frac{1}{IJ}\underline{IJ} \ \text{ et } \ \underline{d}_{b,ik} = \frac{1}{I'F}\underline{I'F} = \underline{n}_{b,ik} \tag{10}$$

Dans notre approche, on décompose encore la pression en partie hydrostatique et partie dynamique. On connaît par hypothèse d'équilibre le gradient de pression hydrostatique : $\underline{G}_{c,i}^h = \underline{f}_I$. On va appliquer la méthode des moindres carrés pour trouver le gradient de pression dynamique.

On minimise alors la fonctionnelle :

$$J = \frac{1}{2}\sum_{j \in Vois(i)} \left[\underline{G}'_{c,i} \cdot \underline{d}_{ij} - \underline{G}'_{f,ij} \cdot \underline{d}_{ij}\right]^2 + \frac{1}{2}\sum_{k \in \gamma_b(i)} \left[\underline{G}'_{c,i} \cdot \underline{d}_{b,ik} - \underline{G}'_{fb,ik} \cdot \underline{d}_{b,ik}\right]^2 \tag{11}$$

Faces internes

Par définition de $\underline{d}_{ij}$, on peut écrire :

$$\begin{aligned} \underline{G}'_{f,ij} \cdot \underline{d}_{ij} &= \frac{1}{IJ}\left[P_J' - P_I'\right] = \frac{1}{IJ}\left[P_J - P_I - P_J^h + P_I^h\right] \\ &= \frac{1}{IJ}\left[P_J - P_I - P_F^h - \underline{FJ}.\underline{f}_J + P_F^h + \underline{FI}.\underline{f}_I\right] \\ &= \frac{1}{IJ}\left[P_J - P_I + \underline{FI}.\underline{f}_I - \underline{FJ}.\underline{f}_J\right] \end{aligned} \tag{12}$$

On retrouve la même forme que dans l'algorithme de base, avec deux termes supplémentaires.

FACES DE BORD

Là encore on considère que les conditions aux limites de pression ne peuvent être que du type Dirichlet ou Neumann homogène. On rappelle qu'aux parois par exemple, la condition "réelle" sur la pression n'est pas un Neumann homogène, mais c'est néanmoins celle qu'on implante en pratique (les algorithmes décrits ici corrigeant cette erreur). Dans gradmc on s'intéresse à la pression dynamique, le Neumann homogène est donc plus justifié (on se débarasse de l'erreur liée à la pression hydrostatique, qui est généralement largement prépondérante). Il n'y a donc pas besoin d'un traitement correctif particulier.

Dans le cas du Neumann homogène ($A_{ik} = 0$ et $B_{ik} = 1$), on a évidemment $\underline{G}'_{f,b,ik} \cdot \underline{d}_{b,ik} = 0$. La formule utilisée dans la version de base de gradmc est donc encore valable :

$$\underline{G}'_{f,b,ik} \cdot \underline{d}_{b,ik} = \frac{A_{ik} + (B_{ik} - 1)\, P_I}{I'F} + \frac{B_{ik} - 1}{I'F} \underline{II'}.\underline{G}'_{c,i} \tag{13}$$

Dans le cas d'une condition de Dirichlet ($B_{ik} = 0$), on a alors :

$$\begin{aligned}
\underline{G}'_{f,b,ik} \cdot \underline{d}_{b,ik} &= \frac{1}{I'F}\left[P'_F - P'_{I'}\right] = \frac{1}{I'F}\left[P_F - P_{I'} - P^h_F + P^h_{I'}\right] \\
&= \frac{1}{I'F}\left[A_{ik} - P_I - \underline{II'}.\underline{G}_{c,i} - \underline{I'F}.\underline{f}_I\right] \\
&= \frac{1}{I'F}\left[A_{ik} - P_I - \underline{II'}.\underline{G}'_{c,i} - \underline{II'}.\underline{f}_I - \underline{I'F}.\underline{f}_I\right] \\
&= \frac{1}{I'F}\left[A_{ik} - P_I - \underline{II'}.\underline{G}'_{c,i} - \underline{IF}.\underline{f}_I\right]
\end{aligned} \tag{14}$$

D'une manière générale, on peut donc écrire la formule suivante, valable pour une condition de Dirichlet ou une condition de Neumann homogène :

$$\underline{G}'_{f,b,ik} \cdot \underline{d}_{b,ik} = \frac{A_{ik} + (B_{ik} - 1)\, P_I}{I'F} + \frac{B_{ik} - 1}{I'F} \underline{II'}.\underline{G}'_{c,i} + \frac{B_{ik} - 1}{I'F} \underline{IF}.\underline{f}_I \tag{15}$$

On minimise donc la fonctionnelle :

$$\begin{aligned}
J = \frac{1}{2} \sum_{j \in Vois(i)} & \left[\underline{G}'_{c,i} \cdot \underline{d}_{ij} - \frac{1}{IJ}\left(P_J - P_I + \underline{FI}.\underline{f}_I - \underline{FJ}.\underline{f}_J\right)\right]^2 \\
& + \frac{1}{2} \sum_{k \in \gamma_b(i)} \left[\underline{G}'_{c,i} \cdot \underline{d}_{b,ik} - \frac{A_{ik} + (B_{ik} - 1)\, P_I}{I'F} - \frac{B_{ik} - 1}{I'F} \underline{II'}.\underline{G}'_{c,i} - \frac{B_{ik} - 1}{I'F} \underline{IF}.\underline{f}_I\right]^2
\end{aligned} \tag{16}$$

On pose

$$\underline{d}'_{b,ik} = \underline{d}'_{b,ik} - \frac{B_{ik} - 1}{I'F} \underline{II'} \tag{17}$$

La fonctionnelle qu'on minimise est donc :

$$\begin{aligned}
J = \frac{1}{2} \sum_{j \in Vois(i)} & \left[\underline{G}'_{c,i} \cdot \underline{d}_{ij} - \frac{1}{IJ}\left(P_J - P_I + \underline{FI}.\underline{f}_I - \underline{FJ}.\underline{f}_J\right)\right]^2 \\
& + \frac{1}{2} \sum_{k \in \gamma_b(i)} \left[\underline{G}'_{c,i} \cdot \underline{d}'_{b,ik} - \frac{A_{ik} + (B_{ik} - 1)\, P_I}{I'F} - \frac{B_{ik} - 1}{I'F} \underline{IF}.\underline{f}_I\right]^2
\end{aligned} \tag{18}$$

On a alors pour chaque cellule les trois équations suivantes ($l \in \{1,2,3\}$).

$$\left[\sum_{j\in Vois(i)} (\underline{d}_{ij})_l(\underline{d}_{ij})_m + \sum_{k\in\gamma_b(i)} (\underline{d}_{b,ik})'_l(\underline{d}_{b,ik})'_m\right] G'_{c,i,m} =$$

$$\sum_{j\in Vois(i)} \frac{1}{IJ}\left[P_J - P_I + \underline{FI}.\underline{f}_I - \underline{FJ}.\underline{f}_J\right](\underline{d}_{ij})_l$$

$$+ \sum_{k\in\gamma_b(i)} \left[\frac{A_{ik} + (B_{ik}-1)\,P_I}{I'F} + \frac{B_{ik}-1}{I'F}\underline{IF}.\underline{f}_I\right](\underline{d}_{b,ik})'_l \quad (19)$$

On retrouve la même formule que dans le cas de l'algorithme usuel, avec juste un terme supplémentaire au second membre.

3 Résolution de Navier-Stokes

On cherche à résoudre le système suivant :

$$\begin{cases} \rho\dfrac{\partial}{\partial t}\underline{u} + \mathcal{D}(\underline{u}) = -\underline{\text{grad}}\ P + \underline{\underline{A}}(\underline{u}).\underline{u} + \underline{B}(\underline{u}) + \underline{f} \\ \text{div}(\rho\underline{u}) = 0 \end{cases} \quad (20)$$

où $\mathcal{D}$ est l'opérateur de convection diffusion, $\underline{f}$ représente les forces extérieures (indépendantes de la vitesse : gravité, force centrifuge, force électrique, ...), et $\underline{\underline{A}}(\underline{u}).\underline{u} + \underline{B}(\underline{u})$ représente les autres termes source de l'équation ($\underline{\underline{A}}$ est une matrice diagonale, qui permet d'impliciter une partie des termes).

Dans l'algorithme classique, *Code_Saturne* résout l'équation en deux étapes :

PRÉDICTION

$$\rho^{(n)}\frac{\underline{u}^* - \underline{u}^{(n)}}{\Delta t} + \mathcal{D}(\underline{u}^*) = -\underline{\text{grad}}\ P^{(n)} + \underline{\underline{A}}(\underline{u}^{(n)}).\underline{u}^* + \underline{B}(\underline{u}^{(n)}) + \underline{f}^{(n)} \quad (21)$$

CORRECTION

$$\rho^{(n)}\frac{\underline{u}^{(n+1)} - \underline{u}^*}{\Delta t} = -\underline{\text{grad}}\ \delta P^{(n+1)} \quad (22)$$

avec $\delta P^{(n+1)} = P^{(n+1)} - P^{(n)}$.

Le problème de cette méthode est que le terme $\underline{f}^{(n)}$ a été mis à jour en début de pas de temps (mise à jour de $\rho^{(n)}$ par exemple) et qu'il n'est donc plus équilibré avec $P^{(n)}$ (qui est équilibré avec $\underline{f}^{(n-1)}$). Ceci est contraire à l'hypothèse de base de notre algorithme modifié de calcul du gradient. Nous modifions donc le schéma de la façon suivante :

PRÉDICTION

$$\rho^{(n)}\frac{\underline{u}^* - \underline{u}^{(n)}}{\Delta t} + \mathcal{D}(\underline{u}^*) = -\underline{\text{grad}}\ P^{(n)} + \underline{\underline{A}}(\underline{u}^{(n)}).\underline{u}^* + \underline{B}(\underline{u}^{(n)}) + \underline{f}^{(n-1)} \quad (23)$$

Correction

$$\rho^{(n)}\frac{\underline{u}^{(n+1)}-\underline{u}^*}{\Delta t} = -\underline{\mathrm{grad}}\ \delta P^{(n+1)} + \delta \underline{f}^{(n)} \tag{24}$$

avec $\delta \underline{f}^{(n)} = \underline{f}^{(n)} - \underline{f}^{(n-1)}$.

Au premier pas de temps, on pose $\underline{f}^0 = \underline{0}$, ce qui est cohérent avec l'initialisation uniforme de la pression ($P = P_0$).

Le traitement de la phase de prédiction ne pose pas de problème particulier. On utilise simplement le nouveau sous-programme de calcul de gradient de pression, pour prendre en compte correctement la composante hydrostatique. Il pourra être nécessaire de modifier la position de l'appel du sous-programme `ustsns`, pour pouvoir inclure des termes source utilisateur dans le vecteur $\underline{f}$ et les prendre en compte dans le gradient de pression.

La phase de correction demande une attention plus particulière. Nous nous y attardons au paragraphe suivant.

4 Traitement de la phase de correction de pression

4.1 Méthode générale

L'équation (24) est résolue en en prenant la divergence, et en prenant en compte le fait que $\mathrm{div}(\rho^{(n)}\underline{u}^{(n+1)}) = 0$. En ajoutant la partie Rhie & Chow, on obtient l'équation suivante :

$$\mathrm{div}(\Delta t \underline{\mathrm{grad}}\ _f \delta P^{(n+1)}) = \mathrm{div}\left(\rho \underline{u}^* + \Delta t \delta \underline{f}^{(n)} + Arak\left[\Delta t \underline{\mathrm{grad}}\ _c P^{(n)} - \Delta t \underline{\mathrm{grad}}\ _f P^{(n)}\right]\right) \tag{25}$$

en distinguant dans les notations les parties qui seront traitées avec des gradients "facette" ($\underline{\mathrm{grad}}\ _f$) ou "cellule" ($\underline{\mathrm{grad}}\ _c$).

En passant en discrétisation volumes finis classique, avec les notations usuelles, on a :

$$\begin{aligned}\sum_{j\in Vois(i)} \Delta t_{ij}\underline{\mathrm{grad}}\ _f \delta P_{ij}^{(n+1)}.\underline{S}_{ij} &= \sum_{j\in Vois(i)} (\rho\underline{u}^*)_{ij}.\underline{S}_{ij} + \sum_{j\in Vois(i)} \Delta t_{ij}\delta \underline{f}_{ij}^{(n)}.\underline{S}_{ij} \\ &\quad + Arak \sum_{j\in Vois(i)} \left[(\Delta t \underline{\mathrm{grad}}\ _c P^{(n)})_{ij} - \Delta t_{ij}\underline{\mathrm{grad}}\ _f P_{ij}^{(n)}\right].\underline{S}_{ij}\end{aligned} \tag{26}$$

Le membre de gauche est calculé par `itrmas` ou directement par la matrice. On choisit de ne rien changer à la façon dont ces termes sont calculés, ni aux conditions aux limites "standard" sur la pression (en particulier on met toujours des A_{ik} et B_{ik} qui correspondent à un flux nul en paroi). Par contre, on agira sur le membre de droite pour ajouter les termes correctifs nécessaires.

4.2 Projection aux faces du terme source sur les faces internes

L'interpolation effectuée dans itrmas pour les faces internes est :

$$\Delta t_{ij}\underline{\text{grad}}\ _f\delta P_{ij}.\underline{S}_{ij} = \Delta t_{ij}S_{ij}\frac{\delta P_J - \delta P_I}{I'J'} + \frac{\varepsilon S_{ij}}{2I'J'}(\underline{JJ'} - \underline{II'}).(\Delta t_I\underline{\text{grad}}\ _c\delta P_I + \Delta t_J\underline{\text{grad}}\ _c\delta P_J) \quad (27)$$

Pour avoir un traitement correct de la pression, il faut que la valeur à la face $\delta\underline{f}_{ij}$ soit telle que

$$\Delta t_{ij}\underline{\text{grad}}\ _f\delta P^h_{ij}.\underline{S}_{ij} = \Delta t_{ij}\delta\underline{f}_{ij}.\underline{S}_{ij} \quad (28)$$

(pour alléger les notations on ne note plus les exposants $(n+1)$ sur δP^h et (n) sur $\delta\underline{f}$).

On doit donc avoir :

$$\begin{aligned}\Delta t_{ij}\delta\underline{f}_{ij}.\underline{S}_{ij} &= \frac{S_{ij}}{I'J'}\left[\Delta t_{ij}(\delta P^h_J - \delta P^h_I) + \frac{\varepsilon}{2}(\underline{JJ'} - \underline{II'}).(\Delta t_I\underline{\text{grad}}\ _c\delta P^h_I + \Delta t_J\underline{\text{grad}}\ _c\delta P^h_J)\right]\\ &= \frac{S_{ij}}{I'J'}\left[\Delta t_{ij}(\delta P^h_{ij} + \underline{FJ}.\delta\underline{f}_J - \delta P^h_{ij} - \underline{FI}.\delta\underline{f}_I) + \frac{\varepsilon}{2}(\underline{JJ'} - \underline{II'}).(\Delta t_I\delta\underline{f}_I + \Delta t_J\delta\underline{f}_J)\right]\end{aligned}$$

et donc, finalement :

$$\Delta t_{ij}\delta\underline{f}_{ij}.\underline{S}_{ij} = \frac{S_{ij}}{I'J'}\left[\Delta t_{ij}(\underline{FJ}.\delta\underline{f}_J - \underline{FI}.\delta\underline{f}_I) + \frac{\varepsilon}{2}(\underline{JJ'} - \underline{II'}).(\Delta t_I\delta\underline{f}_I + \Delta t_J\delta\underline{f}_J)\right] \quad (29)$$

<u>REMARQUES</u>

- On pourrait encore transformer l'équation (29) en faisant apparaître des termes en $\frac{S_{ij}}{I'J'}\underline{FJ'} = \alpha\underline{S_{ij}}$, mais il subsiste toujours des termes en $\underline{II'}$, qui dans le code doivent être recalculés à partir de $\underline{FI}$. On ne gagne donc pas en temps de calcul.
- L'équation (29) est assez complexe, mais résulte uniquement de la façon pas forcément très naturelle de traiter le pas de temps dans itrmas. Tout changement dans ce dernier sous-programme affectera évidemment directement la façon d'interpoler le terme source.

4.3 Projection aux faces du terme source sur les faces de bord

Pour les faces de bord, on a assez naturellement : $\Delta t_{ik}\delta\underline{f}_{ik}.\underline{S}_{ik} = \Delta t_I\delta\underline{f}_I.\underline{S}_{ik}$
Mais la manière dont est traité le gradient de pression aux faces de bord par itrmas ou dans la matrice du système génère en fait une erreur. Comme on ne peut ou ne souhaite pas agir sur ces derniers, on va modifier la façon de projeter le terme source, pour corriger cet effet.

L'interpolation effectuée par itrmas est la suivante :

$$\Delta t_{ik}\underline{\text{grad}}\ _f\delta P_{ik}.\underline{S}_{ik} = \Delta t_I S_{ik}\frac{\texttt{INC}\times A_{ik} + (B_{ik}-1)\delta P_{I'}}{I'F} \quad (30)$$

(dans le cas ci-dessus, INC $= 0$ puisqu'on traite δP, mais itrmas est aussi utilisé pour une partie du filtre Rhie & Chow, cette fois-ci avec la pression complète ; le raisonnement est donc fait ici avec `INC` quelconque).

Dans le cas où on a une condition de Dirichlet sur la pression ($B_{ik} = 0$), cette formule est correcte, et aucun problème ne se pose. La projection du terme source à la face est simplement : $\Delta t_{ik}\delta \underline{f}_{ik}.\underline{S}_{ik} = \Delta t_I \delta \underline{f}_I.\underline{S}_{ik}$.

Par contre, dans le cas où on met une condition de flux nul sur la pression ($A_{ik} = 0$ et $B_{ik} = 1$), on commet une erreur, car il est certes plus ou moins correct de mettre une telle condition sur la partie dynamique de la pression, mais on a toujours un gradient non nul de pression hydrostatique (si on a un terme source bien orienté, bien sûr). Une condition à la limite plus correcte est donc d'écrire : $\dfrac{\partial \delta P'}{\partial n} S_{ik} = 0$ et $\dfrac{\partial \delta P^h}{\partial n} S_{ik} = \delta \underline{f}_I.\underline{S}_{ik}$.

Mais ce qui importe, c'est que le terme source à la face équilibre le gradient de pression hydrostatique calculé par le code (dans `itrmas` ou dans la matrice), qui est nul. Le terme source projeté à une face de bord avec condition aux limites de type flux nul doit donc être nul.

Au final, on a donc, pour tout type de face de bord :

$$\Delta t_{ik}\delta \underline{f}_{ik}.\underline{S}_{ik} = (1 - B_{ik})\Delta t_I \delta \underline{f}_I.\underline{S}_{ik} \tag{31}$$

où B_{ik} est le coefficient des conditions aux limites de la pression.

<u>REMARQUE</u>
Dans le cas de faces à condition de flux nul de pression, la valeur de la pression donnée par les conditions aux limites n'est donc pas la "vraie" pression à la face. Il faut lui rajouter la composante hydrostatique. Ceci est à garder en tête si l'on cherche par exemple à calculer des efforts sur une paroi.

4.4 Cas des conditions aux limites de sortie libre

Les conditions standards implantées dans *Code_Saturne* pour les faces de sortie sont du type suivant : en début de pas de temps, on calcule le gradient de pression (par `grdcel`) et on l'utilise pour extrapoler la pression aux faces de bord à partir de la valeur aux centre des cellules de bord ($P_F = P_I + \underline{IF}.\underline{\text{grad}}\ P_I$) ; ensuite, on décale la pression sur toutes les faces de bord d'une constante, afin que la pression sur la face de référence soit P_0, et on utilise ces valeurs en conditions de Dirichlet pour le pas de temps à venir.

Ces conditions aux limites étant calculées en début de pas de temps, on peut avoir un problème si la pression varie fortement dans le pas de temps. Dans les cas usuels traités jusqu'à présent, les variations de masse volumique (ou plus généralement de terme source) sur un pas de temps sont assez faible, et on n'a donc pas détecté de réel problème. Dans le cadre des applications visées par les modifications proposées, ces variations peuvent être plus fortes, notamment au premier pas de temps (initialisation uniforme de P, donc pression uniforme sur les faces de sortie, alors qu'en fin de pas de temps on doit obtenir une pression avec une composante hydrostatique non nulle). Nous avons donc été amenés à modifier le traitement des conditions aux limites de sortie.

PHASE DE PRÉDICTION
Dans cette phase, on garde le traitement initial, la pression n'intervenant que de manière explicite, donc totalement conforme aux conditions aux limites.

PHASE DE CORRECTION
Avant de résoudre le système qui va donner l'incrément de pression, on met à jour les conditiosn de Dirichlet aux faces de sortie. Comme on ne connaît évidemment pas la valeur δP_F de la variation de pression à la face (puisqu'on cherche justement à calculer δP), on va supposer que l'essentiel des variations vient d'un changement des termes sources $\delta \underline{f}$ et de la pression hydrostatique associée. On résout donc une équation en $\text{div}(\underline{\text{grad}}\ (\delta \tilde{P}^h)) = \text{div}(\delta \underline{f})$, avec des conditions de flux nul sur toutes les frontières. Puis on extrapole la valeur de cette pseudo-pression hydrostatique aux faces de sortie (car on connaît son gradient) : $\delta \tilde{P}^h_F = \delta \tilde{P}^h_I + \underline{IF}.\delta \underline{f}_I$. Enfin, on recale cet incrément de pression de manière à ce qu'il soit nul sur la face de référence.

REMARQUES
• Dans la phase de correction, les conditions aux limites sont utilisées avec $\texttt{INC} = 0$ pour annuler les Dirichlets. Dans le cas des faces de sortie, on veut justement garder ces Dirichlets. Dans toute la phase de correction, on n'utilise donc pas le coefficient A_{ik} de base, mais un coefficient $\tilde{A}_{ik}$, initialisé à 0 pour toutes les faces et mis à $\delta \tilde{P}^h_F$ pour les faces de sortie. On passe alors dans les différentes routines avec $\texttt{INC} = 1$.

• Pour plus de précision, on pourrait penser à mettre à jour les conditions de Dirichlet à chaque sous-sweep de résolution en pression. Des tests ont été faits dans ce sens, qui n'ont pas apporté grand chose, mais ont plutôt destabilisé le système.

• La phase de calcul de la pseudo-pression hydrostatique étant assez lourde, on ne la met en œuvre que quand il y a des faces de sortie, et si le terme en $\delta \underline{f}$ sur ces faces est supérieur à une valeur limite (ce pourra être le cas notamment au premier pas de temps, où toute la gravité est prise dans le terme en $\delta \underline{f}$). Empiriquement, on a choisi de ne pas calculer la pression hydrostatique si $||\delta \underline{f}|| < 10^{-6}||\underline{f}||$ ou si $||\delta \underline{f}|| < 10^{-10}$.

4.5 Filtre Rhie & Chow

Pour éviter des problèmes d'interpolation de la pression hydrostatique aux faces, on ne va appliquer ce flitre qu'à la pression dynamique :

$$\Phi = (\Delta t \underline{\text{grad}}\ {}_c P')_{ij} - \Delta t_{ij} \underline{\text{grad}}\ {}_f P'_{ij} = \left(\Delta t(\underline{\text{grad}}\ {}_c P - \underline{f})\right)_{ij} - \left(\Delta t_{ij} \underline{\text{grad}}\ {}_f P_{ij} - \Delta t_{ij} \underline{f}_{ij}\right) \quad (32)$$

Le premier terme ne pose pas de problème : on calcule le gradient de pression au centre des cellules, on retranche le terme source, et on projette aux faces. Pour le deuxième terme, le gradient facette est calculé par `itrmas`. La projection du terme source $\underline{f}$ doit donc s'effectuer de manière compatible, c'est-à-dire qu'on doit utiliser la formule (29).

Pour la projection du premier terme ou le calcul du second, on utilise des conditions aux limites particulières, qui assurent uen valeur nulle de Φ sur les faces de bord (condition de Dirichlet homogène pour la projection du gradient cellule, condition de flux nul pour le calcul du gradient facette).

4.6 Mise en œuvre pratique des modifications

4.6.1 Dans preduv

On suppose qu'on entre dans preduv avec les composantes de $\underline{f}^{(n-1)}$ stockées dans les tableaux RDEVEL(.), RDEVEL(NCEL+.) et RDEVEL(2*NCEL+.).

Avant le calcul du gradient de pression, on calcule le terme source exact $(\rho^{(n)} - \rho_0)\underline{g}$, qui correspond à $\underline{f}^{(n)}$, et on stocke $\delta\underline{f}^{(n)}$ dans les tableaux RDEVEL(3*NCEL+.), RDEVEL(4*NCEL+.) et RDEVEL(5*NCEL+.).

On appelle ensuite le sous-programme de gradient cellule modifié, grdcep (qui appelle gradrp), avec le terme source $\underline{f}^{(n-1)}$, pour obtenir le gradient de pression exact.

Le reste de preduv n'est pas modifié.

REMARQUE

On pourrait intégrer dans $\delta\underline{f}^{(n)}$ des termes sources issus du sous-programme ustsns. Il faudrait alors pour cela changer un peu l'ordre des appels des différentes routines, mais cela ne constituerait pas une modification majeure.

4.6.2 Dans resolp

• Calcul du résidu de normalisation

Pas de changement.

• Calcul de la pression hydrostatique

On appelle une routine spécifique, calhyd, qui n'est autre qu'une version simplifiée de resolp. Les conditions aux limites de flux nul dont passées *via* les tableaux RDEVEL(6*NCEL+.) et RDEVEL(6*NCEL+NFABOR+.). L'incrément de pression hydrostatique est stocké dans RTP(.,IPR(IPHAS)), qui n'a pas encore été initialisé.

• Remplissage de la matrice

Pas de changement.

• Initialisation du flux de masse

Premier appel à `inimas`, avec $\underline{u}^*$, et les conditions aux limites de vitesse. FLUMAS et FLUMAB contiennent alors $(\rho \underline{u}^*)_{ij}$.

On appelle ensuite le sous-programme de calcul de gradient de pression, `grdecp` pour calculer le gradient cellule de $P^{(n)}$, pour le filtre Rhie & Chow. C'est $\underline{f}^{(n-1)}$ qui est passé en argument des forces extérieures. On retire aussi $\underline{f}^{(n-1)}$ du gradient de pression, pour ne garder que le gradient de pression dynamique. Le terme en $Arak$ ($\underline{\text{grad}}\ _c P^{(n)} - \underline{f}^{(n-1)}$) est ensuite projeté aux faces par `inimas`, en prenant des conditions aux limites de Dirichlet nul à toutes les faces de bord.

On projette ensuite le terme source $\Delta t \delta \underline{f}^{(n)}$ aux faces, selon les équations (29) et (31). Le sous -programme `projts` opère cette projection.

Enfin, on ajoute la deuxième partie du filtre Rhie & Chow, en gradient facette, par `itrmas`. $\underline{f}^{(n-1)}$ est passé en argument de `itrmas`, car la reconstruction des non orthogonalités doit se faire avec le gradient "exact", calculé par `grdcep`. Là encore, on met une condition de Dirichlet nul à toutes les faces de bord (c'est-à-dire une condition de flux nul pour `itrmas`). On repasse ensuite dans `projts` pour ôter la composante hydrostatique du gradient facette de pression.

• Mise à jour des conditions aux limites

On remplit le tableau RDEVEL(6*NCEL+.) qui va contenir le coefficient de condition aux limites $\tilde{A}_{ik}$. Tous les $\tilde{A}_{ik}$ sont d'abord mis à zéro, puis on met à jour ceux qui correspondent à des faces de sortie, en ajoutant l'incrément de pression hydrostatique recalé.

• Résolution du système

Rien de particulier à changer dans cette phase. Il faut juste modifier les appels à `itrmas` et `itrgrp` en passant $\delta \underline{f}^{(n)}$ en argument de terme source, pour les gradients de reconstruction des non orthogonalités. En effet, c'est à chaque fois le gradient de $(\delta P^{(n+1)})^k$ qu'on calcule, sauf dans un appel à `itrmas`, en cas de non convergence à la fin du nombre maximal de sous-itérations. Dans ce cas, il y a en effet un deuxième appel à `itrmas`, pour $\delta(\delta P^{(n+1)})^{k+1}$, qui n'est pas vraiment équilibré avec $\delta \underline{f}^{(n)}$. Mais comme ce passage dans `itrmas` se fait sans reconstruction, le problème ne se pose pas.

4.6.3 Dans navsto

• Mise à jour de la vitesse

On appelle `grdcep` avec le terme source $\delta \underline{f}^{(n)}$ pour calculer le gradient d'incrément de pression total, puis on met à jour la vitesse selon l'équation (24).

• Mise à jour du terme source

On met à jour les tableaux RDEVEL(.), RDEVEL(NCEL+.) et RDEVEL(2*NCEL+.), en y stockant

$\underline{f}^{(n)} = \underline{f}^{(n-1)} + \delta \underline{f}^{(n)}$. On met aussi à jour le tableau COEFA(.,ICLIPR) des conditions aux limites de pression, en y ajoutant l'incrément de pression hydrostatique recalé. En effet, on aura éventuellement besoin de $\underline{f}^{(n)}$ et de conditions aux limites sur $P^{(n+1)}$ dans typecl pour le calcul des conditions aux limites du pas de temps suivant, soit bien avant le prochain passage dans navsto. La mise à jour doit donc se faire en fin de pas de temps.

4.6.4 Dans typecl

Pour les conditions de sortie libre, on a besoin de calculer un gradient de pression. Au début du pas de temps n, il s'agit donc du gradient de $P^{(n)}$, qui est en équilibre avec $\underline{f}^{(n-1)}$. On appelle donc grdcep avec les tableaux RDEVEL(.), RDEVEL(NCEL+.) et RDEVEL(2*NCEL+.) qui ont été mis à jour en fin de pas de temps précédent. Les conditions aux limites de pression ont elles aussi été mises à jour en fin de pas de temps précédent.

Lors du traitement des faces de sortie (type 09 ou 10), on repère ces dernières en remplissant le tableau IDEVEL. Valeur 1 pour une face de sortie, valeur 0 sinon. On stocke aussi dans IDEVEL(NFABOR+1) le numéro de la face de bord de référence.

REMARQUE

Attention : l'indicateur IDEVEL ne doit identifier que les faces de sortie sur lesquelles on impose la condition de pression standard. Pour une face de sortie sur laquelle on imposerait un Dirichlet différent (type $P_F = P0$), il n'y a pas à faire de traitement particulier (pas de mise à jour des conditions aux limites par l'incrément de pression hydrostatique). Dans typecl, on ne met donc IDEVEL à 1 que lorsqu'on a une face de sortie sur laquelle l'utilisateur n'a rien spécifié (ICODCL(1,IPR(IPHAS))=0).