WORK WITH M.PARISOT

On a l'algorithme classique de Parareal, avec un solver grossier $\mathcal{G}$ et un solveur fin $\mathcal{F}$, avec une initialisation séquentielle :

$$(0.1) \qquad \begin{cases} U_{j+1}^0 = \mathcal{G}(T_j, T_{j+1}, U_j^0), \quad j = 0, \dots, N-1 \\ U_0^0 = u_0, \end{cases}$$

ainsi que l'itération :

$$(0.2) \qquad U_{j+1}^k = \mathcal{G}(T_j, T_{j+1}, U_j^k) + \mathcal{F}(T_j, T_{j+1}, U_j^{k-1}) - \mathcal{G}(T_j, T_{j+1}, U_j^{k-1}),$$

avec $j = 0, ..., N-1$, $k = 1, ..., N$.
La convergence de l'algorithme vers le résultat fin $\mathcal{F}(T_0, T_N, U_0^0)$ repose sur le fait que à l'itération k, $U_k^k = \mathcal{F}(T_{k-1}, T_k, U_{k-1}^{k-1})$. Pour que ceci soit vérifié, il suffit que $\mathcal{G}(T_{k-1}, T_k, U_{k-1}^k) = \mathcal{G}(T_{k-1}, T_k, U_{k-1}^{k-1})$. Autrement dit, il faut que $U_{k-1}^k = U_{k-1}^{k-1}$ (ce qui sera vérifié par récurrence), et que le grossier soit exactement le même sur ces deux itérations, i.e. : même schéma, même nombre de pas de temps, même temps de départ, etc...

Problème : Dans notre cas on ne veut pas "couper" les pas de temps, surtout ceux du fin, et donc imposer les temps d'arrêt.

1. première idée qui dégrade le grossier mais pas le fin. On propose alors **les modifications suivantes** :

- Lors de l'initialisation, on effectue une réalisation entière du grossier $\mathcal{G}$, et on définit ensuite T_j, $j = 0, .., N$, de manière à avoir autant de pas de temps entre chacun des T_j. (déjà proposé par Gander pour les schémas à pas de temps adaptatifs.)
- Les temps de passage T_j son maintenant fixés pour le reste de la procédure **pour le grossier $\mathcal{G}$ uniquement**.
- Les réalisations du fin $\mathcal{F}$ à l'itération k se feront entre $T_j + \Delta_j^{k-1}$ (arrivée du fin de l'itération précédente à l'intervalle précédent) et $T_{j+1} + \Delta_{j+1}^k$ (on laisse le fin s'arrêter sans "casser" un pas de temps) avec la convention : $\Delta_j^{-1} = T_j$.

Le but étant, d'obtenir à la fin : $U_N^N = \mathcal{F}(T_0, T_N + \Delta_N^N, u_0$ En pratique cela donne, pour $N = 2$:

$$\text{Initialisation : } \begin{cases} U_0^0 = u_0 \\ U_1^0 = \mathcal{G}(T_0, T_1, U_0^0) \\ U_2^0 = \mathcal{G}(T_1, T_2, U_1^0) \end{cases}$$

$$\text{Itération 1 : } \begin{cases} U_0^1 = U_0^0 \\ U_1^1 = \mathcal{G}(T_0, T_1, U_0^1) - \mathcal{G}(T_0, T_1, U_0^0) + \mathcal{F}(T_0, T_1 + \Delta_1^1, U_0^0) \\ U_2^1 = \mathcal{G}(T_1, T_2, U_1^1) - \mathcal{G}(T_1, T_2, U_1^0) + \mathcal{F}(T_1, T_2 + \Delta_2^1, U_1^0) \end{cases}$$

$$\text{Itération 2 : } \begin{cases} U_0^2 = U_0^1 \\ U_1^2 = \mathcal{G}(T_0, T_1, U_0^2) - \mathcal{G}(T_0, T_1, U_0^1) + \mathcal{F}(T_0, T_1 + \Delta_1^2, U_0^1) \\ U_2^2 = \mathcal{G}(T_1, T_2, U_1^2) - \mathcal{G}(T_1, T_2, U_1^1) + \mathcal{F}(T_1 + \Delta_1^1, T_2 + \Delta_2^2, U_1^1) \end{cases}$$

On prouve par récurrence qu'on a bien $U_k^k = \mathcal{F}(T_0, T_k + \Delta_k^k, u_0)$:

$$U_1^1 = \mathcal{G}(T_0, T_1, U_0^1) - \mathcal{G}(T_0, T_1, U_0^0) + \mathcal{F}(T_0, T_1 + \Delta_1^1, U_0^0), \tag{1.1}$$

or $U_0^1 = U_0^0 = u_0$ donc $\mathcal{G}(T_0, T_1, U_0^1) = \mathcal{G}(T_0, T_1, U_0^0)$ et $U_1^1 = \mathcal{F}(T_0, T_1 + \Delta_1^1, U_0^0 = u_0)$. Pareil pour :

$$U_2^2 = \mathcal{G}(T_1, T_2, U_1^2) - \mathcal{G}(T_1, T_2, U_1^1) + \mathcal{F}(T_1 + \Delta_1^1, T_2 + \Delta_2^2, U_1^1). \tag{1.2}$$

On a $U_1^2 = U_1^1$ donc $\mathcal{G}(T_1, T_2, U_1^2) = \mathcal{G}(T_1, T_2, U_1^1)$ et

$$U_2^2 = \mathcal{F}(T_1 + \Delta_1^1, T_2 + \Delta_2^2, U_1^1) = \mathcal{F}(T_1 + \Delta_1^1, T_2 + \Delta_2^2, \mathcal{F}(T_0, T_1 + \Delta_1^1, u_0)), \tag{1.3}$$

donc $U_2^2 = \mathcal{F}(T_0, T_2 + \Delta_2^2, u_0)$. Avec le même raisonnement, si l'on suppose $U_k^k = \mathcal{F}(T_0, T_k + \Delta_k^k, u_0)$, alors :

$$U_{k+1}^{k+1} = \mathcal{G}(T_k, T_{k+1}, U_k^{k+1}) - \mathcal{G}(T_k, T_{k+1}, U_k^k) + \mathcal{F}(T_k + \Delta_k^k, T_{k+1} + \Delta k + 1_{k+1}, U_k^k). \tag{1.4}$$

Comme $U_k^{k+1} = U_k^k$ (a vérifier ?), on a alors bien :

$$U_{k+1}^{k+1} = \mathcal{F}(T_k + \Delta_k^k, T_{k+1} + \Delta_{k+1}^{k+1}, U_1^1) = \mathcal{F}(T_k + \Delta_k^k, T_{k+1} + \Delta_{k+1}^{k+1}, \mathcal{F}(T_0, T_k + \Delta_k^k, u_0)), \tag{1.5}$$

et donc : $U_{k+1}^{k+1} = \mathcal{F}(T_0, T_{k+1} + \Delta_{k+1}^{k+1}, u_0)$.

2. Seconde idée qui ne dégrade ni le grossier ni le fin.

$$\text{Initialisation : } \begin{cases} U_0^0 = u_0 \\ U_1^0 = \mathcal{G}(T_0, (n_{\Delta t})_1^0, U_0^0) \\ U_2^0 = \mathcal{G}(t((n_{\Delta t})_1^0), (n_{\Delta t})_2^0, U_1^0) \end{cases}$$

$$\text{Itération 1 : } \begin{cases} U_0^1 = U_0^0 \\ U_1^1 = \mathcal{G}(T_0, (n_{\Delta t})_1^1, U_0^1) - \mathcal{G}(T_0, (n_{\Delta t})_1^0, U_0^0) + \mathcal{F}(T_0, (n_{\delta t})_1^1, U_0^0) \\ U_2^1 = \mathcal{G}(t((n_{\Delta t})_1^1), (n_{\Delta t})_2^1, U_0^1) - \mathcal{G}(t((n_{\Delta t})_1^0), (n_{\Delta t})_2^0, U_1^0) + \mathcal{F}(t((n_{\delta t})_1^0), (n_{\delta t})_2^1, U_1^0) \end{cases}$$

$$\text{Itération 2 : } \begin{cases} U_0^2 = U_0^1 \\ U_1^2 = \mathcal{G}(T_0, T_1, U_0^2) - \mathcal{G}(T_0, T_1, U_0^1) + \mathcal{F}(T_0, T_1 + \Delta_1^2, U_0^1) \\ U_2^2 = \mathcal{G}(T_1, T_2, U_1^2) - \mathcal{G}(T_1, T_2, U_1^1) + \mathcal{F}(T_1 + \Delta_1^1, T_2 + \Delta_2^2, U_1^1) \end{cases}$$

On prouve par récurrence qu'on a bien $U_n^n = \mathcal{F}(T_0, (n_{\delta t})_n^n, u_0)$:

$$U_1^1 = \mathcal{G}(T_0, (n_{\Delta t})_1^1, U_0^1) - \mathcal{G}(T_0, (n_{\Delta t})_1^0, U_0^0) + \mathcal{F}(T_0, (n_{\delta t})_1^1, U_0^0) \tag{2.1}$$

or $\mathcal{G}(T_0, (n_{\Delta t})_1^1, U_0^1) = \mathcal{G}(T_0, (n_{\Delta t})_1^0, U_0^0)$ car ils ont tous les deux même temps de départ et même donnée initiale.

Questions :

- Quelle erreur regarder ?
- erreur relative à l'échelle ?
- Pertinent de comparer $\|s_{Para} - s_{GN}\|$ et $\|s_{Para} - s_{SW}\|$?
- Modèle plus complexe ? 2D ? avec batimétrie ? îlot ? maillage à trous ?
- Vu qu'on veut un parareal "non-intrusif", il me "suffit" de deux fonctions @ShallowWater(u0,T0,Tf) et @GreenNagdhi(u0,T0,Tf) (si possible avec option ou non de tronquer le dernier pas de temps).

$$\begin{pmatrix} h \\ hu \end{pmatrix}_j^k = \mathcal{G}\left(\begin{pmatrix} h \\ hu \end{pmatrix}_{j-1}^k\right) - \mathcal{G}\left(\begin{pmatrix} h \\ hu \end{pmatrix}_{j-1}^{k-1}\right) + \mathcal{F}\left(\Pi_{GN}\left[\begin{pmatrix} h \\ hu \\ 0 \\ 0 \end{pmatrix}_{j-1}^{k-1}\right]\right) \tag{2.2}$$

ou bien :

$$\begin{pmatrix} h \\ hu \\ hw \\ h\sigma \end{pmatrix}_j^k = \Pi_{GN}\left[\mathcal{G}\left(\begin{pmatrix} h \\ hu \\ 0 \\ 0 \end{pmatrix}_{j-1}^k\right)\right] - \Pi_{GN}\left[\mathcal{G}\left(\begin{pmatrix} h \\ hu \\ 0 \\ 0 \end{pmatrix}_{j-1}^{k-1}\right)\right] + \mathcal{F}\left(\begin{pmatrix} h \\ hu \\ hw \\ h\sigma \end{pmatrix}_{j-1}^{k-1}\right) \tag{2.3}$$

$$\Pi_{GN} : \begin{pmatrix} h \\ u \\ w \\ \sigma \end{pmatrix} = \begin{pmatrix} h \\ U \end{pmatrix} \mapsto \begin{pmatrix} h \\ \tilde{U} \end{pmatrix}, \quad \text{où } \tilde{U} = \begin{pmatrix} \tilde{u} \\ \tilde{w} \\ \tilde{\sigma} \end{pmatrix} = \texttt{getws}\begin{pmatrix} h \\ A^{-1}\beta \end{pmatrix} \tag{2.4}$$

$A^{-1}\beta$ la solution du système linéaire $Ax = \beta$ (avec les defs de A et β que tu m'avais donné).
Selon moi pour projeter, il faut avant diviser $hu, hw, h\sigma$ par h, puis projeter, puis re-multiplier par h.

3. 06/09/23. Ce qui a été proposé par Martin avant l'été:

$$\begin{pmatrix} h \\ hu \\ hw \\ h\sigma \end{pmatrix}_j^k = G\left[\begin{pmatrix} h \\ hu \\ hw \\ h\sigma \end{pmatrix}_{j-1}^k\right] - G\left[\begin{pmatrix} h \\ hu \\ hw \\ h\sigma \end{pmatrix}_{j-1}^{k-1}\right] + F\left[h_{j-1}^{k-1} \cdot \Pi_{h_{j-1}^{k-1}}\left[\begin{pmatrix} u \\ w \\ \sigma \end{pmatrix}_{j-1}^k\right]\right] \tag{3.1}$$