Parallélisation temporelle pour la résolution du système de Lorenz

Une approche avec l'algorithme Parareal

Elonm AHOUANYE   **Yanel AÏNA**

École Nationale Supérieure de Génie Mathématique et Modélisation
(ENSGMM Abomey) *Sous la direction de*
Dr. DANDOGBESSI Bruno

26 mars 2025

Introduction

Contexte physique : particules actives guidées par la mémoire

- Les systèmes de particules actives : un domaine de la physique moderne
- Les gouttes "marcheuses" sur un bain liquide vibrant
- Interaction avec les ondes auto-générées
- Mémoire ondulatoire guidant le mouvement

Du système physique au modèle mathématique

- Équation de trajectoire intégro-différentielle :

$$\ddot{x}_d + \dot{x}_d = F_{self} + F_{bias} \tag{1}$$

- Force du champ d'ondes auto-généré :

$$F_{self} = -R \int_{-\infty}^{t} W'(x_d(t) - x_d(s)) e^{-\frac{t-s}{\tau}} ds \tag{2}$$

Émergence du système de Lorenz

- Simplification avec $W(x) = \cos(x)$
- Variables :

$$X = \dot{x}_d \quad \text{(vitesse)}$$
$$Y = F_{self} \quad \text{(force de mémoire)}$$
$$Z = R \int_{-\infty}^{t} \cos(x_d(t) - x_d(s)) e^{-\frac{t-s}{\tau}} ds$$

- Système de Lorenz modifié :

$$\begin{cases} \dot{X} = Y - X \\ \dot{Y} = -\frac{1}{\tau} Y + XZ \\ \dot{Z} = R - \frac{1}{\tau} Z - XY \end{cases}$$

Problématique

- **Défis de la résolution numérique :**
  - Non-linéarité : termes de couplage XZ et XY
  - Sensibilité aux conditions initiales
  - Échelles multiples : paramètre τ
- **Objectifs :**
  - Résolution séquentielle avec RK4
  - Parallélisation temporelle avec l'algorithme Parareal
  - Analyse des performances et de la précision

Méthode RK4

Principe de la méthode RK4

- Méthode classique de résolution numérique des EDO
- Approximation par combinaison de 4 évaluations :

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3}$$

- Coefficients k_i : évaluations à différents points
  - k_1 : pente initiale
  - k_2, k_3 : pentes aux points milieu
  - k_4 : pente finale

Application au système de Lorenz

- Pour un état $\mathrm{u} = (X, Y, Z)$:

$$\frac{d\mathrm{u}}{dt} = \mathrm{f}(\mathrm{u}) = \begin{pmatrix} Y - X \\ -\frac{1}{\tau} Y + XZ \\ R - \frac{1}{\tau} Z - XY \end{pmatrix}$$

- Calcul des coefficients :

$$\begin{aligned} k_1 &= \mathrm{f}(t_n, \mathrm{u}_n) \\ k_2 &= \mathrm{f}(t_n + \frac{h}{2}, \mathrm{u}_n + \frac{h}{2} k_1) \\ k_3 &= \mathrm{f}(t_n + \frac{h}{2}, \mathrm{u}_n + \frac{h}{2} k_2) \\ k_4 &= \mathrm{f}(t_n + h, \mathrm{u}_n + h k_3) \end{aligned}$$

Paramètres de simulation

- **Configuration standard :**
  - Pas de temps : $h = 0.01$
  - Durée totale : $T = 100.0$
  - Amplitude des ondes : $R = 2.5$
  - Conditions initiales : $(X_0, Y_0, Z_0) = (1.0, 0.0, 0.0)$
- **Régimes étudiés :**
  - $\tau = 0.5$: État Non-Marcheur
  - $\tau = 2.0$: Marche Régulière
  - $\tau = 5.0$: Marche Chaotique

Limitations pour la parallélisation

- **Double dépendance séquentielle :**
  1. **Temporelle** :
  $$u_{n+1} = \Phi_{\mathrm{RK4}}(u_n)$$
  2. **Interne** : calcul séquentiel des k_i
  $$k_i = f(k_1, \ldots, k_{i-1})$$

- **Conséquences :**
  - Pas de calcul indépendant des étapes temporelles
  - Impossibilité de parallélisation directe

Algorithme Parareal

Principe fondamental

- Solution à la barrière de séquentialité temporelle
- **Éléments clés :**
  - Décomposition du domaine temporel
  - Deux niveaux de propagateurs
  - Processus itératif de correction
- **Avantages :**
  - Calcul parallèle sur différents intervalles
  - Maintien de la précision

Architecture à deux niveaux

- **Propagateur grossier $\mathcal{G}$**
  - Rapide mais approximatif
  - Basé sur RK2 ou méthode d'Euler
  - Prédiction initiale
- **Propagateur fin $\mathcal{F}$**
  - Précis mais coûteux
  - Basé sur RK4
  - Exécution parallèle

Processus itératif

1. **Initialisation**
   - Division de $[0, T]$ en N sous-intervalles
   - $U_n^0 = u_0$ pour $n = 0$
   - Prédiction grossière : $U_{n+1}^0 = \mathcal{G}(T_n, U_n^0, \Delta T)$
2. **Calcul parallèle**
   - Calcul fin : $\mathcal{F}(T_n, U_n^k, \Delta T)$
   - Formule de correction :

$$U_{n+1}^{k+1} = \mathcal{G}(T_n, U_n^{k+1}, \Delta T) + \mathcal{F}(T_n, U_n^k, \Delta T) - \mathcal{G}(T_n, U_n^k, \Delta T)$$

Application au système de Lorenz

- **Adaptation des propagateurs**
  - Propagateur grossier : RK2 avec pas adaptatif
  - $\Delta t_{\mathcal{G}} = \min(\alpha\tau, \Delta T)$
  - Propagateur fin : RK4 avec pas fin
  - $\Delta t_{\mathcal{F}} = \frac{\Delta t_{\mathcal{G}}}{m}$
- **Considérations particulières**
  - Gestion des différents régimes dynamiques
  - Adaptation aux échelles de temps du système
  - Conservation des invariants physiques

Stratégies de convergence

- **Critères adaptatifs**

$$\varepsilon_k = \max\left\{ \frac{\|U_n^{k+1} - U_n^k\|}{\|U_n^k\|}, \frac{|E_k - E_{k-1}|}{|E_k|} \right\} < \mathrm{tol}$$

- **Optimisations**
  - Prédiction améliorée avec extrapolation
  - Stockage intelligent des états intermédiaires
  - Équilibrage de charge dynamique

Implémentation

Architecture logicielle

- **Architecture modulaire à trois niveaux**
  1. Programme principal (main.f90)
  2. Modules principaux
     - Solveurs
     - Parareal
     - Décomposition
  3. Sous-modules spécialisés
- **Séparation claire des responsabilités**
  - Logique de calcul
  - Parallélisation
  - Gestion des données

Composants principaux

Programme principal

- Gestion des arguments
- Initialisation MPI
- Distribution des tâches
- Mesure des performances

Module de solveurs

- RK4 (haute précision)
- AB2/AB3 (intermédiaire)
- Euler (rapide)

Adaptation aux régimes dynamiques

- **Ajustement automatique des paramètres**

```
! Ajustement selon le regime
if (tau < 1.0) then  ! Non-marcheur
    h_coarse = min(h_coarse, tau/20.0)
    h_fine = min(h_fine, tau/200.0)
elsif (tau < 3.0) then  ! Marche reguliere
    h_coarse = min(h_coarse, tau/10.0)
    h_fine = min(h_fine, tau/100.0)
else  ! Regimes chaotiques
    h_coarse = min(h_coarse, 0.1)
    h_fine = min(h_fine, 0.01)
end if
```

Mécanismes de protection

- **Gestion des instabilités**
  - Détection des valeurs aberrantes
  - Limitation des amplitudes
  - Compteur d'erreurs
- **Circuit breaker**
  - Arrêt en cas d'instabilité répétée
  - Repli sur des paramètres plus conservateurs
  - Journalisation des événements

Parallélisation et équilibrage

- **Distribution des intervalles**
  - Répartition équitable entre processus
  - Gestion du reste de la division
  - Adaptation à la charge
- **Communications MPI**
  - Échanges aux points de synchronisation
  - Réduction des données transférées
  - Optimisation des communications collectives

Visualisation et analyse

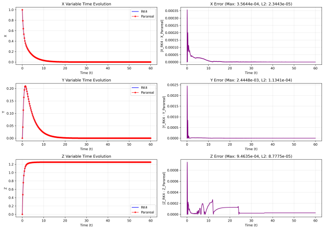
- **Densité des trajectoires selon τ**
  - 50 points/intervalle pour $\tau < 2.0$
  - 75 points/intervalle pour $2.0 \leq \tau < 5.0$
  - 100 points/intervalle pour $\tau \geq 5.0$
- **Outils d'analyse**
  - Scripts de post-traitement
  - Validation automatisée
  - Génération de rapports

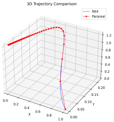
Résultats et analyse comparative

Méthodologie de comparaison

- **Comparaison systématique RK4 vs Parareal**
- **Aspects analysés :**
  - Évolution temporelle (X, Y, Z)
  - Erreur absolue entre solutions
  - Convergence vers états stationnaires/attracteurs
  - Portraits de phase
- **Régimes étudiés :** $\tau = 0.5,\ 2.0,\ 5.0$

Régime non-marcheur ($\tau = 0.5$)

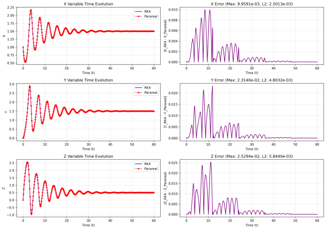


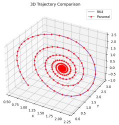


Observations :

- Excellente concordance
- Erreurs $< 10^{-6}$
- Stabilité numérique

Marche régulière ($\tau = 2.0$)

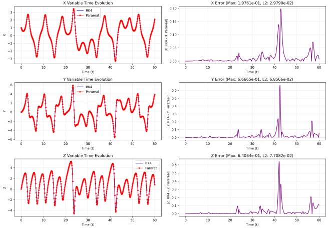


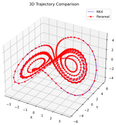


Résultats :

- États stationnaires fidèles
- Stabilisation rapide
- Précision maintenue

Régime chaotique ($\tau = 5.0$)

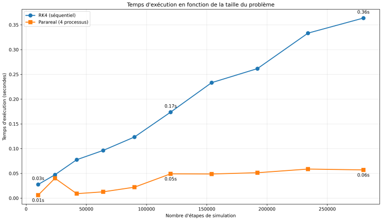




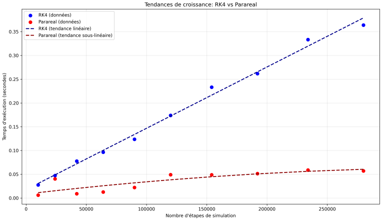
Analyse :

- Cohérence des trajectoires
- Pics d'erreur aux transitions
- Structure d'attracteur préservée

Performances temporelles



Temps d'exécution



Tendances de croissance

- Gain significatif en temps de calcul
- Efficacité $>$ 50% à grande échelle
- Scalabilité quasi-linéaire jusqu'à 250000 itérations

Synthèse des performances

- **Précision**
  - Erreurs relatives $< 10^{-4}$ (régimes stables)
  - Conservation des caractéristiques qualitatives
  - Préservation des invariants
- **Stabilité**
  - Robustesse dans tous les régimes
  - Gestion efficace des transitions
  - Contrôle des erreurs numériques

Conclusion et perspectives

Synthèse des résultats

- **Objectifs atteints**
  - Parallélisation temporelle efficace
  - Précision comparable à RK4 séquentiel
  - Convergence robuste en régime chaotique
- **Gains de performance**
  - Accélération significative
  - Maintien de la qualité numérique
  - Scalabilité démontrée

Limitations actuelles

- **Contraintes identifiées**
  - Sensibilité à la taille des intervalles
  - Divergence possible avec Euler comme solveur grossier
  - Scalabilité limitée à 6 processeurs dans notre étude
- **Points d'amélioration**
  - Adaptation dynamique des intervalles
  - Optimisation des communications MPI
  - Extension à plus de processeurs