



HAL
open science

Résolution d'EDP par un schéma en temps < pararéel >

Jacques-Louis Lions, Yvon Maday, Gabriel Turinici

► **To cite this version:**

Jacques-Louis Lions, Yvon Maday, Gabriel Turinici. Résolution d'EDP par un schéma en temps < pararéel >. Comptes rendus de l'Académie des sciences. Série I, Mathématique, 2001, 332 (7), pp.661-668. hal-00798372

HAL Id: hal-00798372

<https://hal.science/hal-00798372v1>

Submitted on 11 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Résolution d'EDP par un schéma en temps « pararéel »

Jacques-Louis LIONS ^a, Yvon MADAY ^b, Gabriel TURINICI ^{b,c}

^a Collège de France, 3, rue d'Ulm, 75231, Paris cedex 05, France

^b Laboratoire d'analyse numérique, Université Pierre-et-Marie-Curie, 4, place Jussieu, 75252 Paris cedex 05, France

^c ASCI, UPR 9029, bâtiment 506, Université Paris-Sud, 91405, Orsay cedex, France

(Reçu le 24 novembre 2000, accepté le 4 décembre 2000)

Résumé. On propose dans cette Note un schéma permettant de profiter d'une architecture parallèle pour la discrétisation en temps d'une équation d'évolution aux dérivées partielles. Cette méthode, basée sur un schéma d'Euler, combine des résolutions grossières et des résolutions fines et indépendantes en temps en s'inspirant de ce qui est classique en espace. La parallélisation qui en résulte se fait dans la direction temporelle ce qui est en revanche non classique. Elle a pour principale motivation les problèmes en temps réel, d'où la terminologie proposée de « pararéel ». © 2000 Académie des sciences/Éditions scientifiques et médicales Elsevier SAS

A “parareal” in time discretization of PDE's

Abstract. *The purpose of this Note is to propose a time discretization of a partial differential evolution equation that allows for parallel implementations. The method, based on an Euler scheme, combines coarse resolutions and independent fine resolutions in time in the same spirit as standard spatial approximations. The resulting parallel implementation is done in the non standard time direction. Its main goal concerns real time problems, hence the proposed terminology of “parareal” algorithm. © 2000 Académie des sciences/Éditions scientifiques et médicales Elsevier SAS*

Abridged English version (the numbers are those of the French version)

Let us consider the partial differential evolution equation (1) over the interval $[0, T]$ with initial condition $u(t = 0) = u_0$ and boundary conditions that are not made precise here. The unknown u can be a scalar or a vector function, the PDE can be linear or not...

We propose in this Note a new scheme in time that allows to combine accuracy and parallelism. We decompose the time interval $[0, T]$ into N subintervals $[T^n, T^{n+1}]$ of size $\Delta T = \frac{T}{N}$. Then we introduce

Note présentée par Olivier PIRONNEAU

S0764-4442(00)01793-6/FLA

© 2000 Académie des sciences/Éditions scientifiques et médicales Elsevier SAS. Tous droits réservés.

the functions λ_n for $n = 0, \dots, N-1$ (of course $\lambda_0 = u_0$) and solve over $[T^n, T^{n+1}]$ the equation (2) with the same boundary conditions as in (1).

The collection u_n , for $n = 0, \dots, N-1$, can be computed in parallel. This collection coincides with $\{u_{|[T^n, T^{n+1}]}\}_n$ when $u_n(T^{n+1}, -) (= \lim_{\delta>0, \delta \rightarrow 0} u_n(T^{n+1} - \delta)) = \lambda_{n+1}$ for $n = 0, \dots, N-1$.

We provide in this Note a way to get λ_n at any accuracy.

As illustrated on the simple example of equation (3), the first step consists in defining Y^n by the implicit Euler scheme (4) then solve (exactly or with a very fine approximation based on a small time step δt) the independent (thus parallelizable) problems (5) over each $[T^n, T^{n+1}]$. Iteratively, we then improve the order accuracy of this scheme by deriving from known values Y_k^n and $y_k^n(t)$ over $[T^n, T^{n+1}]$ better approximations Y_{k+1}^n and $y_{k+1}^n(t)$ over $[T^n, T^{n+1}]$ as follows:

- (i) introduce the jumps $S_k^n = y_k^{n-1}(T^n) - Y_k^n$;
- (ii) then propagate the jumps with a coarse resolution of the δ_k^n following (6);
- (iii) finally set $Y_{k+1}^n = y_k^{n-1}(T^n) + \delta_k^n$ and solve (7) (again exactly or with a very fine approximation).

This procedure provides at step k an order k approximation as is stated in (8). The important remark is that step (ii) consists in N independent problems that can thus be solved in parallel.

We first analyse the stability diagram of our scheme, both in the case where the basic scheme is the implicit (plot 1) and the explicit Euler scheme (plot 2). The conclusion is that our extension does not pollute much the basic stability diagrams.

It is simple to extend this scheme to a linear PDE, as (9) after a spatial discretization. Table 1 presents the convergence history of the scheme for a fixed spacial discretization. We can compare to the standard Euler scheme with δt which provides an error estimated to $0.6 \cdot 10^{-2}$. With our scheme, this error is recovered at step $k = 3$. Our scheme requires three coarse resolutions and two fine parallel resolutions over each $[T^n, T^{n+1}]$. This improves the efficiency by a factor of 8 if parallel implementation is done.

The scheme can also be extended to nonlinear nonsymmetric equations as in (10) by linearizing the equation at each available state: first, solve (11) that provides a coarse solution U_1^n , then use these values to solve (12) with a fine time step over each $[T^n, T^{n+1}]$ and in parallel. The iteration then proceeds as follows, assuming U_k^n and $u_k^n(t)$ are known

- (i) introduce $S_k^n = u_k^{n-1}(T^n) - U_k^n$;
- (ii) then propagate the jumps δ_k^n with (13);
- (iii) then set $U_{k+1}^n = u_k^{n-1}(T^n) + \delta_k^n$ and solve (14) over each $[T^n, T^{n+1}]$ and in parallel.

Table 2 presents the convergence history in this case. Here the two time steps have been chosen in a more coherent manner $\frac{T}{\Delta T} \simeq \frac{\Delta T}{\delta t}$ and the standard Euler scheme's error is recovered with four iterations of our scheme. A factor of 18 can be gained through parallel implementation.

Remarks. – This scheme combines coarse and fine resolutions in time in the same spirit as what is done in space for the domain decomposition methods (cf. e.g. [6]).

Runge–Kutta or predictor-corrector scheme are other examples of the fact that the right combination of steps similar to a first order scheme may lead to higher order discretization schemes. What is new here is the potentiality to implement the resulting scheme in parallel.

Our scheme is not the first attempt to perform parallel in time discretization, we can quote e.g. [1] and [2] for small systems. Nevertheless, the extension of their ideas based on Newton scheme to PDE is not obvious.

1. Introduction

On considère une équation d'évolution aux dérivées partielles

$$\frac{\partial u}{\partial t} + Au = f \text{ dans un intervalle de temps } [0, T], \quad (1)$$

avec des conditions initiales $u(t = 0) = u_0$ et des conditions aux limites qu'il est inutile de préciser ici. L'inconnue u peut être scalaire ou vectorielle, l'EDP linéaire ou non linéaire...

Nous proposons dans cette Note un schéma en temps compatible avec une résolution en parallèle et avec une précision souhaitée. Pour cela, on choisit un entier N représentant le nombre de pas de temps et on note $\Delta T = \frac{T}{N}$ le pas de temps et $T^n = n\Delta T$ pour $n = 0, \dots, N$. On introduit ensuite à ces instants des fonctions λ_n pour $n = 0, \dots, N-1$ (bien sûr $\lambda_0 = u_0$) et on résout, dans l'intervalle $[T^n, T^{n+1}]$ l'équation

$$\frac{\partial u_n}{\partial t} + Au_n = f_n, \quad f_n = f|_{[T^n, T^{n+1}]}, \quad (2)$$

avec pour condition initiale $u_n(t = T^n) = \lambda_n$ et les mêmes conditions aux limites que pour u .

On peut ainsi résoudre en parallèle la collection des u_n , pour $n = 0, \dots, N-1$. Cette collection coïncide avec $\{u|_{[T^n, T^{n+1}]}\}_n$ quand $u_n(T^{n+1}, -) (= \lim_{\delta > 0, \delta \rightarrow 0} u_n(T^{n+1} - \delta)) = \lambda_{n+1}$ pour $n = 0, \dots, N-1$. Nous montrons dans cette Note comment trouver ces valeurs de façon itérative.

Remarque. – On peut considérer, comme dans [4], que les λ_n jouent le rôle de « contrôles virtuels ». Il est alors naturel d'introduire la fonction de coût virtuel $J(\lambda) = \sum_{n=1}^{N-1} \|u_n(T^n, -) - \lambda_n\|^2$ et de proposer une méthode de gradient pour trouver les λ_n . La procédure qui en résulte n'est pas parallélisable directement et nous proposons ici une sorte de loi de feedback pour se faire une intuition les bonnes valeurs. On développera cette idée dans une prochaine note sur l'application de ce schéma pour le contrôle.

Pour commencer, on expose l'idée sur l'exemple simple d'une équation différentielle linéaire

$$\begin{cases} \frac{dy}{dt}(t) = -ay(t) & \text{sur } [0, T], \\ y(t = 0) = y_0. \end{cases} \quad (3)$$

On considère le schéma d'Euler implicite

$$\begin{cases} \frac{Y^{n+1} - Y^n}{\Delta T} + aY^{n+1} = 0, \\ Y^0 = y_0, \end{cases} \quad (4)$$

puis on utilise les valeurs précédemment calculées pour résoudre de façon exacte, sur chaque intervalle de temps $[T^n, T^{n+1}]$

$$\begin{cases} \frac{dy^n}{dt}(t) = -ay^n(t) & \text{sur } [T^n, T^{n+1}], \\ y^n(t = T^n) = Y^n. \end{cases} \quad (5)$$

On propose maintenant une procédure itérative pour améliorer la précision de ce schéma.

On pose donc $Y_1^n = Y^n$ et, défini sur $[T^n, T^{n+1}]$, $y_1^n(t) = y^n(t)$. Puis, supposant connus Y_k^n et $y_k^n(t)$ sur $[T^n, T^{n+1}]$

(i) on introduit les sauts $S_k^n = y_k^{n-1}(T^n) - Y_k^n$;

(ii) puis on propage les sauts

$$\begin{cases} \frac{\delta_k^{n+1} - \delta_k^n}{\Delta T} + a\delta_k^{n+1} = \frac{S_k^n}{\Delta T}, \\ \delta_k^0 = 0, \end{cases} \quad (6)$$

(iii) ensuite on pose $Y_{k+1}^n = y_k^{n-1}(T^n) + \delta_k^n$ et on résout de façon exacte, et en parallèle

$$\begin{cases} \frac{dy_{k+1}^n}{dt}(t) = -ay_{k+1}^n(t) & \text{sur } [T^n, T^{n+1}], \\ y_{k+1}^n(t = T^n) = Y_{k+1}^n. \end{cases} \quad (7)$$

C'est alors un exercice que de montrer la :

Proposition 1. – Le schéma précédent est d'ordre k au sens où il existe une constante c_k telle que

$$\forall n, 0 \leq n \leq N-1, \quad |Y_k^n - y(T^n)| + \max_{t \in [T^n, T^{n+1}]} |y_k^n(t) - y(t)| \leq c_k \Delta T^k. \quad (8)$$

Démonstration. – La proposition est évidente au rang $k = 1$, on la démontre par récurrence. On rappelle tout d'abord que $y_k^{n-1}(T^n) = e^{-a\Delta T} Y_k^{n-1}$ et que donc $S_k^n = e^{-a\Delta T} Y_k^{n-1} - Y_k^n$. Par ailleurs, la solution de (6) est

$$\delta_k^n = \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} S_k^p = \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} Y_k^{p-1} - Y_k^p),$$

et ainsi

$$Y_{k+1}^n = e^{-a\Delta T} Y_k^{n-1} + \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} Y_k^{p-1} - Y_k^p)$$

en rappelant que la solution exacte $y(T^n) = e^{-na\Delta T} y_0$ on déduit que

$$\begin{aligned} Y_{k+1}^n - e^{-na\Delta T} y_0 &= e^{-a\Delta T} [Y_k^{n-1} - e^{-(n-1)a\Delta T} y_0] \\ &+ \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} \left(e^{-a\Delta T} [Y_k^{p-1} - e^{-(p-1)a\Delta T} y_0] - [Y_k^p - e^{-pa\Delta T} y_0] \right), \end{aligned}$$

ou encore, en introduisant $\varepsilon_k^n = Y_k^n - y(T^n)$,

$$\begin{aligned} \varepsilon_{k+1}^n &= e^{-a\Delta T} \varepsilon_k^{n-1} + \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} \varepsilon_k^{p-1} - \varepsilon_k^p) \\ &= e^{-a\Delta T} \varepsilon_k^{n-1} + \sum_{p=0}^{n-2} (1 + a\Delta T)^{1+p-n} (e^{-a\Delta T} \varepsilon_k^p) - \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (\varepsilon_k^p) \\ &= \varepsilon_k^{n-1} (e^{-a\Delta T} - (1 + a\Delta T)^{-1}) + \sum_{p=1}^{n-2} (1 + a\Delta T)^{p-n} ((1 + a\Delta T) e^{-a\Delta T} - 1) \varepsilon_k^p, \end{aligned}$$

de sorte que si, à l'ordre k il existe c_k indépendante de ΔT telle que, pour tout n , $|\varepsilon_k^n| \leq c_k \Delta T^k$, alors

$$\varepsilon_{k+1}^n \leq c_k \Delta T^k (e^{-a\Delta T} - (1 + a\Delta T)^{-1}) + c_k \left[\sum_{p=1}^{n-2} ((1 + a\Delta T) e^{-a\Delta T} - 1) \right] \Delta T^k \leq \alpha_N c_k \left(\frac{Ta^2}{2} \right) \Delta T^{k+1},$$

avec une constante α_N indépendante de k (et de l'ordre de 2) ce qui montre la proposition avec $c_{k+1} = \alpha_N c_k (Ta^2/2)$.

- Après cet exercice, quelques remarques s'imposent.
- On aurait pu choisir un schéma d'Euler explicite et la conclusion aurait été la même.
 - Il n'y a rien d'étonnant à ce qu'à partir d'un schéma du premier ordre, bien combiné, on arrive à un schéma d'ordre supérieur. C'est par exemple le cas des schémas de Runge–Kutta ou des schémas de type prédictor–correcteur. Néanmoins, le schéma précédent a la particularité de corriger l'ordre de la solution et d'être naturellement parallélisable.
 - Pour que ce qui vient d'être donné soit effectivement un schéma, il faut bien évidemment remplacer la résolution exacte entre T^n et T^{n+1} par une résolution précise, avec n'importe quel schéma classique utilisant un pas de temps δt petit par rapport à ΔT . L'intérêt de ce schéma réside dans le fait que les résolutions sur les intervalles $[T^n, T^{n+1}]$ peuvent se faire en parallèle.
 - La grille grossière (pas de temps ΔT) permet de propager rapidement l'information sur tout l'intervalle $[0, T]$, la grille fine (pas de temps δt) permet d'obtenir la précision du schéma. Cette combinaison de calculs sur deux grilles généralise un concept bien connu dans les méthodes de décomposition de domaine pour aborder le parallélisme (cf. [6] par exemple).

2. Quelques notions sur la stabilité

Il est intéressant de vérifier que la stabilité de notre schéma de base n'est pas trop dégradée par la multiplication des étapes d'amélioration de l'ordre. C'est ce que l'on a vérifié en recherchant numériquement les diagrammes de stabilité pour les différents ordres en partant d'un schéma de Euler implicite ou explicite. Les diagrammes correspondants sont donnés dans les figures suivantes où l'on vérifie que pour des ordres raisonnables (≤ 4) les diagrammes de stabilité sont comparables. C'est particulièrement intéressant pour le schéma implicite où l'on vérifie que le demi espace des réels négatifs est largement dans la zone de stabilité.

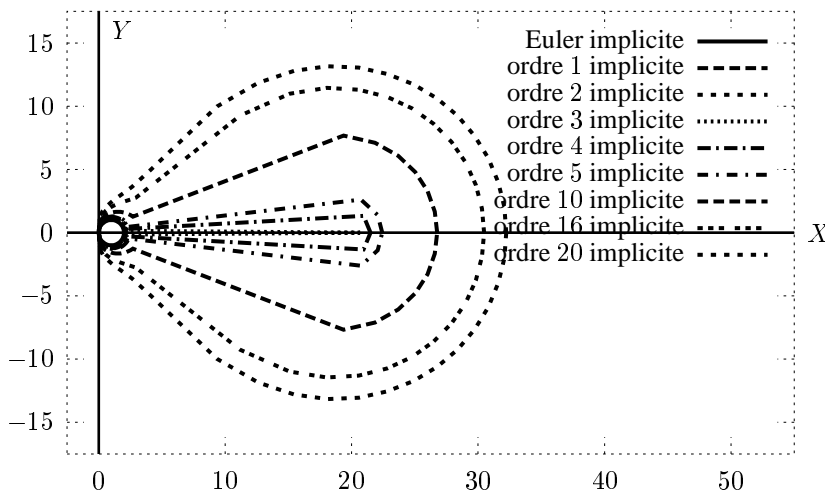


Figure 1. – La région de stabilité pour le schéma implicite est l'extérieur de la courbe.
 Figure 1. – The region of stability for the implicit scheme is the exterior of the curve.

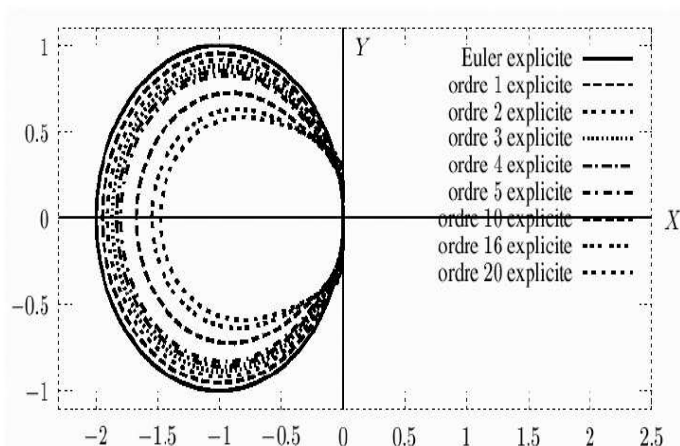


Figure 2. – La région de stabilité pour le schéma explicite est l’intérieur de la courbe.
 Figure 2. – The region of stability for the explicit scheme is the interior of the curve.

3. Application à la résolution d’EDP linéaires en parallèle

Pour appliquer notre schéma à une EDP parabolique, avec un opérateur en espace symétrique, on discrétise en espace ce qui conduit à un système différentiel en temps. La proposition 1 reste vraie comme on peut facilement s’en convaincre en travaillant dans une base propre de l’opérateur spatial.

Pour vérifier le bon comportement de ce schéma sur une EDP elliptique, on a considéré le problème de la chaleur

$$\frac{\partial u}{\partial t} - \Delta u = f = A \cdot \sin(\alpha \cdot t), \quad u(t = 0) = \tan(x^2 + y^2 - 1), \quad (9)$$

dans le disque unité et avec des conditions aux limites homogènes. Les résultats sont dans le tableau suivant.

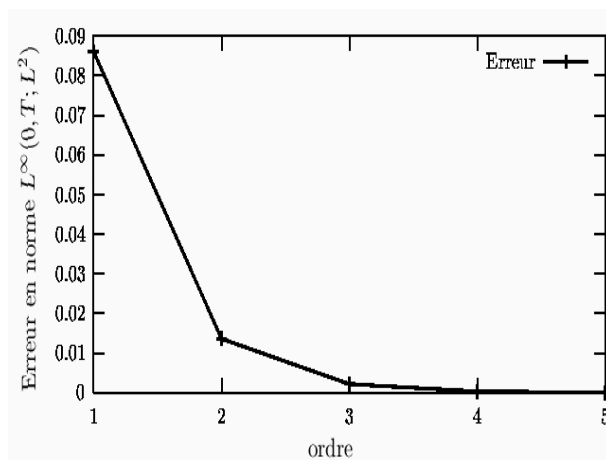


Tableau 1. – Convergence du schéma pour $T = 50$, $\Delta T = 0.1$, $\delta t = 0.004$, $A = 5$, $\alpha = 10$.
 Table 1. – Convergence of the scheme for $T = 50$, $\Delta T = 0.1$, $\delta t = 0.004$, $A = 5$, $\alpha = 10$.

L’erreur pour une résolution classique avec le pas de temps de δt est de $0.6 \cdot 10^{-2}$. On retrouve cette erreur avec notre schémas à l’ordre 3 qui nécessite donc trois résolutions grossières (avec un $\Delta T = 0.1$)

et deux résolutions fines, mais en parallèle, sur chaque $[T^n, T^{n+1}]$. Ceci donne un gain en efficacité d'un facteur 8 si les résolutions fines sont effectuées en parallèle. On précise aussi que l'erreur à chaque ordre est divisée par un facteur 6.

4. Applications dans le cas non linéaire

Pour la généralisation aux cas d'EDP non linéaires, l'argument est d'utiliser la linéarisation de l'équation autour de l'état courant. Prenons par exemple le problème suivant (non linéaire et non symétrique)

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} - \nu \Delta u + \beta u^3 = f = A \cdot \sin(\alpha \cdot t), \quad \nu > 0, \quad \beta > 0, \quad (10)$$

dans le disque unité Ω , avec la même condition initiale que pour (9) et des conditions aux limites homogènes. On propose alors le schéma

$$\begin{cases} \frac{U_1^{n+1} - U_1^n}{\Delta T} + \frac{\partial U_1^{n+1}}{\partial x} - \nu \Delta U_1^{n+1} + \beta (U_1^n)^3 = f^n, \\ U_1^0 = u(t=0). \end{cases} \quad (11)$$

On utilise les valeurs précédentes pour résoudre en parallèle sur chaque intervalle $[T^n, T^{n+1}]$ et de façon fine (en temps)

$$\begin{cases} \frac{\partial u_1^n}{\partial t} + \frac{\partial u_1^n}{\partial x} - \nu \Delta u_1^n + \beta (u_1^n)^3 = f \quad \text{sur } [T^n, T^{n+1}], \\ u_1^n(t = t^n) = U_1^n, \end{cases} \quad (12)$$

puis, supposant connus U_k^n et $u_k^n(t)$ sur $[T^n, T^{n+1}]$

- (i) on introduit $S_k^n = u_k^{n-1}(T^n) - U_k^n$;
- (ii) puis on propage les sauts

$$\begin{cases} \frac{\delta_k^{n+1} - \delta_k^n}{\Delta T} + \frac{\partial \delta_k^{n+1}}{\partial x} - \nu \Delta \delta_k^{n+1} + 3\beta (u_k^{n+1}(T^n))^2 \delta_k^{n+1} = \frac{S_k^n}{\Delta T}, \\ \delta_k^0 = 0; \end{cases} \quad (13)$$

- (iii) ensuite on pose $U_{k+1}^n = u_k^{n-1}(T^n) + \delta_k^n$ et on résout de façon fine, et en parallèle

$$\begin{cases} \frac{d u_{k+1}^n}{dt}(t) + \frac{\partial u_{k+1}^n}{\partial x}(t) - \nu \Delta u_{k+1}^n(t) + \beta (u_{k+1}^n)^3 = f \quad \text{sur } [T^n, T^{n+1}], \\ u_{k+1}^n(t = t^n) = U_{k+1}^n. \end{cases} \quad (14)$$

On vérifie sur le tableau suivant que les ordres augmentent bien avec k . Ici le gain est encore plus important que pour le cas linéaire car les paramètres ont été choisis de façon plus compatible. On a en effet $\frac{T}{\Delta T} \simeq \frac{\Delta T}{\delta t}$ et l'on remarque que l'erreur d'un schéma classique (avec δt), de l'ordre de $0.3 \cdot 10^{-3}$, est retrouvée avec notre schéma à l'ordre 4, nécessitant quatre résolutions grossières et trois résolutions fines, sur chaque $[T^n, T^{n+1}]$, en parallèle, donnant un gain d'un facteur $\frac{\Delta T}{(4+3) \times \delta t} \simeq 18$.

Remarque. – Ce schéma n'est pas la première tentative de parallélisation en temps, on pourra aussi consulter [1] et [2] pour des systèmes différentiels non linéaires de petite dimension. Leur approche est basée sur des itérations de Newton et ne semble pas pouvoir être étendue à des systèmes de grande dimension. À notre connaissance, notre schéma est une première proposition qui permette de traiter des EDP (non linéaires).

Remarque. – Les potentialités de ce schéma peuvent encore être améliorées en utilisant des discrétisations en espace différentes pour les résolutions avec les pas de temps ΔT et δt . Dans le même sens que [3], on peut en effet proposer d'utiliser un maillage H grossier pour propager les sauts entre $u_n(T^{n,-})$ et

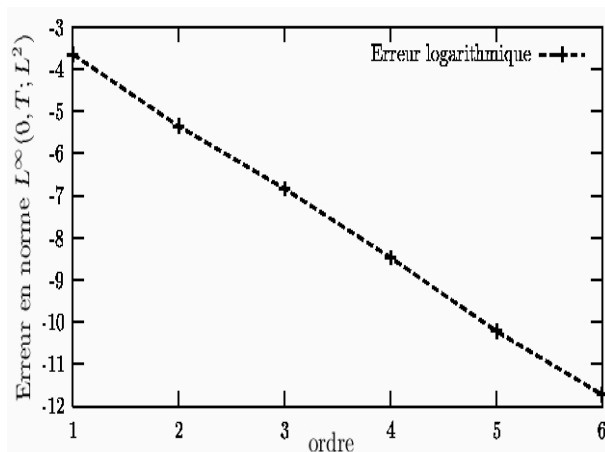


Tableau 2. – Taux de convergence pour $\nu = 1, \alpha = 2, \beta = 5, T = 10, A = 5, \Delta T = 0.1, \delta t = 0.0008$.

Table 2. – Rate of convergence for $\nu = 1, \alpha = 2, \beta = 5, T = 10, A = 5, \Delta T = 0.1, \delta t = 0.0008$.

$u_{n+1}(T^{n,+})$ et par ailleurs un maillage h fin pour les résolutions en parallèle sur chaque $[T^n, T^{n+1}]$. Le schéma de base étant implicite, donc inconditionnellement stable, nous n’avons pas eu à nous préoccuper ici d’une éventuelle condition de type CFL entre les paramètres de discrétisation en espace et en temps. Les gains obtenus nous semblent être un pas supplémentaire vers le temps réel pour des problèmes dépendants du temps.

Remerciements. Pour ce travail nous avons utilisé le code FreeFem+ [5] et bénéficié des conseils judicieux de Frédéric Hecht et Olivier Pironneau. Qu’ils en soient remerciés.

Références bibliographiques

- [1] Chartier P., Philippe B., A parallel shooting technique for solving dissipative ODE’s, Computing 51 (1993) 209–236.
- [2] Erhel J., Rault S., Algorithme parallèle pour le calcul d’orbites, Techniques et Sciences Informatiques 19 (2000) 649–673.
- [3] Girault V., Lions J.-L., Two-grids finite element schemes for the steady Navier–Stokes problem in polyhedra, Port. Math. (2000).
- [4] Lions J.-L., Virtual and effective control for distributed systems and decomposition of everything, J. d’Anal. Math. 80 (2000) 257–297.
- [5] Pironneau O., Hecht F., Bernardi D., Ohtsuka K., FreeFem+, <http://www-rocq.inria.fr/Frederic.Hecht/FreeFemPlus.htm>
- [6] Quarteroni A., Valli A., Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, Oxford, 1999.