Approximation and prediction of the numerical solution of some Burgers problems

Marc Prévost, Denis Vekemans

▶ **To cite this version:**

Marc Prévost, Denis Vekemans. Approximation and prediction of the numerical solution of some Burgers problems. 2022. hal-03663548

HAL Id: hal-03663548
https://hal.science/hal-03663548

Preprint submitted on 10 May 2022

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.

Approximation and prediction of the numerical solution of some Burgers problems

Marc Prévost and Denis Vekemans *

Abstract

The Aitken's Δ^2-prediction of Brezinski has already been used by Morandi Cecchi, Redivo Zaglia and Scenna in order to approximate the solution of a parabolic initial-boundary value problem. The method used consists in two consecutive steps : the approximation with a finite elements method where the solution of system of nonlinear equations is computed by Gauss-Seidel method, followed by a prediction with Aitken's Δ^2-process.

By comparison with this method, we use other methods of prediction and in another way. On the first hand, we not only use the Δ^2-prediction and we can consider a generalization of this method of prediction, the ε-prediction. Moreover, in this paper, we only make use of vector prediction which is more stable than the scalar one. On the other hand, the methods of prediction presented will be used in order to predict the starting vector of the Gauss-Seidel method.

Keywords : approximation, prediction, ε-prediction, vector prediction, Burgers problem

AMS Classification : 65B05, 65N

1 Introduction

From the knowledge of some terms of a sequence, a method of prediction can be used to obtain an approximation or an estimation of the following ones. When the terms of a sequence are difficult to compute exactly or even to approximate, it can be interesting to use methods of prediction. In this paper, the complexity in time to obtain the terms of the

*Université du Littoral Côte d'Opale ; Centre Universitaire de la Mi-Voix ; Laboratoire de mathématiques pures et appliquées Joseph Liouville ; 50, rue Ferdinand Buisson, BP 699 ; 62 228 Calais cedex ; France

sequence is very high and induces a special motivation for the use of methods of prediction. The sequence to be computed in this paper needs the use of Gauss-Seidel method with an arbitrary initial vector (for approximating the solution of a system of nonlinear equations) and it often needs many iterations. We will show here that Gauss-Seidel method with a predicted initial vector is quite performing.

2 Presentation of the Burgers problem

The Burgers problem that will be considered in this paper has already been studied by Morandi Cecchi, Nociforo and Patuzzo Grego [2].

There are several reasons which lead us to look at this equation : this problem has a time evolution and it seems to be natural to consider the sequence of the vectors which define the approximated solution to be predicted as function of time ; it takes a long time to evaluate each component of the approximated solution because a system of nonlinear equations has to be solved and so, it makes sense to use methods of prediction ; as usual solutions of Burgers problem seem to have an exponential behavior, it seems to be *a priori* interesting to use the ε-prediction (see [5, theorem 3, pp.29-30]).

Let $x \in \Omega = [a, b] \subset I\!R$ and $t \in [0, T]$. We consider the partial differential equation

$$u_t(x,t) - \frac{1}{Re} u_{xx}(x,t) + u(x,t)u_x(x,t) = f(x,t),$$

where Re is a real parameter, with the homogeneous boundary conditions

$$u(x,0) = u_0(x) \ ; \ u(a,t) = u(b,t) = 0,$$

where $f(x,t)$ and $u_0(x)$ are given real-valued functions (see [2, (2.1), p44]).

The "weak solution" of this problem u satisfies

$$(u_t, v) + \frac{1}{Re}(u_x, v_x) + (uu_x, v) = (f, v), \ \forall v \in H_0^1(\Omega),$$

where $(.,.)$ is the inner product in $L^2(\Omega)$ [2, (2.2), p44].

The numerical discretizated solution

In the sequel, we take $f \equiv 0$.

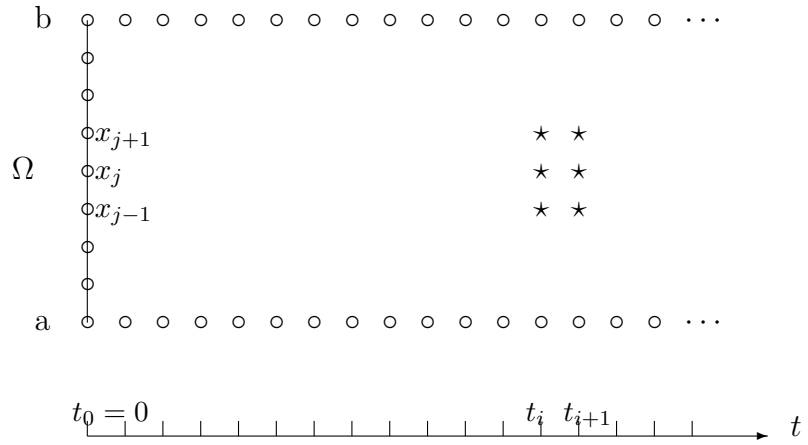
The discretization is homogeneous, i.e. $t_i = i\Delta t, i \in \{0, 1, \ldots, T/\Delta t\}$ (Δt chosen) and $x_j = a + j\Delta x, j \in \{0, 1, \ldots, N\}$ ($\Delta x = (b-a)/N$, N chosen).

We denote by $u_{i,j}$ the approximation of the solution $u(x_j, t_i)$. Then, [2, (4.1), p49] gives the equation

$$\frac{\Delta x}{6\Delta t}[u_{i+1,j+1} + 4u_{i+1,j} + u_{i+1,j-1} - u_{i,j+1} - 4u_{i,j} - u_{i,j-1}]$$

$$
\begin{aligned}
&+\frac{1}{36}[u_{i+1,j+1}^2 + u_{i+1,j+1}u_{i+1,j} - u_{i+1,j}u_{i+1,j-1} - u_{i+1,j-1}^2 \\
&\quad +2u_{i+1,j+1}u_{i,j+1} + u_{i+1,j+1}u_{i,j} + u_{i+1,j}u_{i,j+1} \\
&\quad -u_{i+1,j}u_{i,j-1} - u_{i+1,j-1}u_{i,j} - 2u_{i+1,j-1}u_{i,j-1} \\
&\quad +3u_{i,j+1}^2 + 3u_{i,j+1}u_{i,j} - 3u_{i,j}u_{i,j-1} - 3u_{i,j-1}^2] \\
&+\frac{1}{3Re\Delta x}[-u_{i+1,j+1} + 2u_{i+1,j} - u_{i+1,j-1} - 2u_{i,j+1} + 4u_{i,j} - 2u_{i,j-1}] = 0. \qquad (1)
\end{aligned}
$$

On the following scheme, the $\circ$ represent the quantities of u known as boundary conditions, and the $\star$ represent the six quantities appearing in (1).



To obtain the $(N-1)$ unknowns $u_{1,j}$ for $j = 1, 2, \ldots, N-1$ (i.e. at t_1), we have to solve a system of $(N-1)$ nonlinear equations. To compute the $(N-1)$ unknowns $u_{2,j}$ for $j = 1, 2, \ldots, N-1$ (i.e. at t_1), we have to solve a system of $(N-1)$ nonlinear equations. And so on.

Now, in order to obtain an approximate solution of each of these nonlinear systems, we can use the nonlinear Gauss-Seidel method. If we suppose that, for a fixed i, $u_{i,j},\ j = 1, 2, \ldots, N-1$ are known, $u_{i+1,j},\ j = 1, 2, \ldots, N-1$ can be computed (iteratively from $u_{0,j},\ j = 1, 2, \ldots, N-1$ which is known from the homogeneous boundary conditions) by

1. Input parameters (independent of i)
   - $\alpha \longleftarrow \Delta x/(6\Delta t)$,
   - $\beta \longleftarrow 1/36$,
   - $\gamma \longleftarrow 1/(3Re\Delta x)$.
2. Initializations (i fixed)
   - $$u_{i+1,j}(0) \longleftarrow u_{i,j},\ j = 0, 1, \ldots, N, \qquad (2)$$

- $k \longleftarrow 1$.

3. Recursive rule (the stopping criterium is

$$
\begin{aligned}
&|\alpha[u_{i+1,j+1}(k) + 4u_{i+1,j}(k) + u_{i+1,j-1}(k) - u_{i,j+1} - 4u_{i,j} - u_{i,j-1}] \\
&+\beta[u_{i+1,j+1}(k)^2 + u_{i+1,j+1}(k)u_{i+1,j}(k) - u_{i+1,j}(k)u_{i+1,j-1}(k) - u_{i+1,j-1}(k)^2 \\
&+2u_{i+1,j+1}(k)u_{i,j+1} + u_{i+1,j+1}(k)u_{i,j} + u_{i+1,j}(k)u_{i,j+1} \\
&-u_{i+1,j}(k)u_{i,j-1} - u_{i+1,j-1}(k)u_{i,j} - 2u_{i+1,j-1}(k)u_{i,j-1} \\
&+3u_{i,j+1}^2 + 3u_{i,j+1}u_{i,j} - 3u_{i,j}u_{i,j-1} - 3u_{i,j-1}^2] \\
&+\gamma[-u_{i+1,j+1}(k) + 2u_{i+1,j}(k) - u_{i+1,j-1}(k) - 2u_{i,j+1} + 4u_{i,j} - 2u_{i,j-1}]| \leq \xi, \qquad (3)
\end{aligned}
$$

   where ξ (the precision of the solution) has an arbitrary value)

   - $u_{i+1,0}(k) \longleftarrow 0$,
   - $u_{i+1,N}(k) \longleftarrow 0$,
   - $u_{i+1,j}(k) \longleftarrow -\alpha[u_{i+1,j+1}(k-1) + u_{i+1,j-1}(k) - u_{i,j+1} - 4u_{i,j} - u_{i,j-1}]$
     $+\beta[u_{i+1,j+1}(k-1)^2 - u_{i+1,j-1}(k)^2 + 2u_{i+1,j+1}(k-1)u_{i,j+1} + u_{i+1,j+1}(k-1)u_{i,j}$
     $-u_{i+1,j-1}(k)u_{i,j} - 2u_{i+1,j-1}(k)u_{i,j-1} + 3u_{i,j+1}^2 + 3u_{i,j+1}u_{i,j} - 3u_{i,j}u_{i,j-1} - 3u_{i,j-1}^2]$
     $/\left\{4\alpha + \beta[u_{i+1,j+1}(k-1) - u_{i+1,j-1}(k) + u_{i,j+1} - u_{i,j-1}] + 2\gamma\right\}$,
     $j = 1, 2, \ldots, N-1$,
   - $k \longleftarrow k+1$ (k will be increased as long as the stopping criterium (3) is not satisfied).

4. Output values

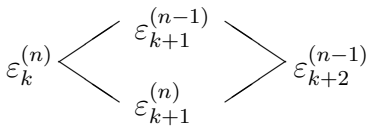
   - $u_{i+1,j} \longleftarrow u_{i+1,j}(k-1)$, $j = 0, 1, \ldots, N$.

3 Using the ε-prediction

The problem is that many iterations of the Gauss-Seidel method are generally needed in the previous algorithm. In order to reduce the complexity in time of this algorithm, we have to try to change the initialization (2) in order to start the iterations of the Gauss-Seidel method nearer the solution. In that way, we use the ε-prediction (introduced by Vekemans [6]) to predict the approximated solution. The ε-prediction is a very simple (i.e. it does not use special properties of the sequence to predict) method of prediction which is based on the ε-algorithm (the ε-algorithm of Wynn [7] is used to implement the Shanks' tranformation, a transformation for accelerating the convergence [1]).

In this paper, only the ε-prediction is used. For details, the interested reader is refered to [5]. This method of prediction is based on the ε-algorithm which uses the rhumbus rule that is : $\varepsilon_{k+2}^{(n-1)} = \varepsilon_k^{(n)} + 1/(\varepsilon_{k+1}^{(n)} - \varepsilon_{k+1}^{(n-1)}), \forall n \geq 1, \forall k \geq -1$, with $\varepsilon_{-1}^{(n)} = 0$, $\forall n$ and

$\varepsilon_0^{(n)} = S_n,\ \forall n$, where $(S_n)_n$ is the sequence to be accelerated.



Here, only the vector ε-prediction (the interested reader can be refered to [5] for the scalar ε-prediction) is used (more precisely, the ε_2-prediction (i.e. the Aitken's Δ^2-prediction), the ε_4-prediction and the ε_6-prediction where the subscript denotes the number of columns used for the computation). Effectively, on numerical examples, it has been observed that the use of vector prediction is better than the scalar one because it is more stable. In fact, we can introduce the vector ε-prediction, in order to generalize the scalar ε-algorithm to the vector case : the division in the ε-algorithm has just to be replaced by the pseudo-inverse of a vector in the vector ε-algorithm (i.e. if ϖ is a vector of $I\!R^n$, its pseudo-inverse $\varpi^\dagger$ is given by $\varpi^\dagger = \varpi/||\varpi||_2^2$ and so, the rhumbus rule becomes $\varepsilon_{k+2}^{(n-1)} = \varepsilon_k^{(n)} + (\varepsilon_{k+1}^{(n)} - \varepsilon_{k+1}^{(n-1)})^\dagger, \forall n \geq 1, \forall k \geq -1$, with $\varepsilon_{-1}^{(n)} = 0,\ \forall n$, with $\varepsilon_{-1}^{(n)} = 0,\ \forall n$ and $\varepsilon_0^{(n)} = S_n,\ \forall n$, where $(S_n)_n$ is the sequence of vectors to be accelerated).

How did we use the ε-prediction ?

The method of Morandi Cecchi, Redivo Zaglia and Scenna [4] can be summed up as an approximation (i.e. with nonlinear Gauss-Seidel method) of the solution at t_1 from the knowledge of the solution at t_0, then an approximation of the solution at t_2 from the approximated solution at t_1, ..., then an approximation of the solution at t_{l+2} ($l \in I\!N$) from the approximated solution at t_{l+1}, and only now a prediction (i.e. by mean of Aitken's Δ^2-prediction) of the solution at t_L ($L \in I\!N$) ($L > l+2$), from the approximated solution at t_l, t_{l+1} and t_{l+2}. This is an approximation followed by a method of prediction.

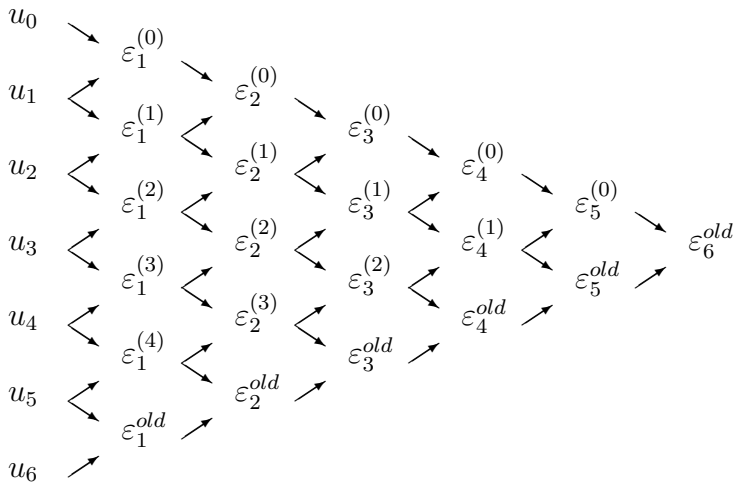
Our method (so called prediction/approximation) is based on the use of the ε-prediction to evaluate a *good* starting vector before using Gauss-Seidel method : when looking at the Gauss-Seidel method for the initialization step, we can see $u_{i+1,j}(0) \longleftarrow u_{i,j}$, $j = 0, 1, \ldots, N$ (see (2)) ; the idea is now to replace $u_{i+1,j}(0) \longleftarrow u_{i,j}$, $j = 0, 1, \ldots, N$, by $u_{i+1,j}(0) \longleftarrow \overline{u_{i+1,j}}$, $j = 0, 1, \ldots, N$, where $\overline{u_{i+1,j}}$ is a vector predicted from $u_{0,j}$, $u_{1,j}$, $u_{2,j}$, ... $u_{i,j}$, $j = 0, 1, \ldots, N$.

The vector ε_2-prediction (respectively ε_4-prediction, then ε_6-prediction)

This algorithm uses the last three (respectively five, then seven) consecutive values of the finite sequence to predict. It will define the vector $\overline{u_{i+1,\cdot}}$. For example, for i fixed, starting from a finite sequence $(u_{k,\cdot})_{k\in\{0,1,\ldots,i\}}$, the ε_2-prediction (respectively ε_4-prediction, then ε_6-prediction) only uses $u_{i-2,\cdot}$, $u_{i-1,\cdot}$ and $u_{i,\cdot}$ (respectively $u_{i-4,\cdot}$, $u_{i-3,\cdot}$, $u_{i-2,\cdot}$, $u_{i-1,\cdot}$ and $u_{i,\cdot}$, then $u_{i-6,\cdot}$, $u_{i-5,\cdot}$, $u_{i-4,\cdot}$, $u_{i-3,\cdot}$, $u_{i-2,\cdot}$, $u_{i-1,\cdot}$ and $u_{i,\cdot}$) and then it predicts the terms $u_{i+k,\cdot}$, for $k = 1, 2, \ldots$ of the sequence (here, it is just for $k = 1$ as we only need to predict one vector) which can be computed as we can see for the vector ε_6-prediction (one can imagine for the vector ε_2-prediction or the ε_4-prediction) by

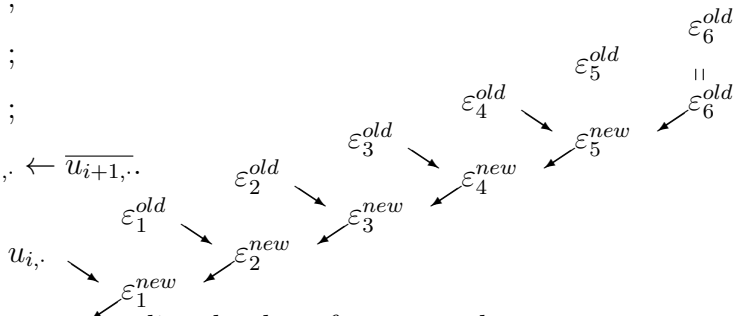
1. Initialization ($u_{0,\cdot}$, $u_{1,\cdot}$, …, $u_{6,\cdot}$ are supposed to be approximated)

   - $\varepsilon_0^{(l)} \leftarrow u_{l,\cdot}$, $l = 0, 1, \ldots, 6$;
   - $\varepsilon_1^{(l)} \leftarrow (\varepsilon_0^{(l+1)} - \varepsilon_0^{(l)})^\dagger$, $l = 0, 1, \ldots, 5$;
   - $\varepsilon_m^{(l)} \leftarrow \varepsilon_{m-2}^{(l+1)} + (\varepsilon_{m-1}^{(l+1)} - \varepsilon_{m-1}^{(l)})^\dagger$, $l = 0, 1, \ldots, 6-m; m = 2, 3, \ldots, 6$;
   - Storage of the subdiagonal $\varepsilon_m^{old} \leftarrow \varepsilon_m^{(6-m)}$, $m = 1, 2, \ldots, 6$.



2. Recursive rule for $i = 6, 7, \ldots, T/\Delta t - 1$

   - $\varepsilon_5^{new} \leftarrow \varepsilon_5^{old} + (\varepsilon_6^{old} - \varepsilon_4^{old})^\dagger$;
   - $\varepsilon_4^{new} \leftarrow \varepsilon_4^{old} + (\varepsilon_5^{new} - \varepsilon_3^{old})^\dagger$;
   - $\varepsilon_3^{new} \leftarrow \varepsilon_3^{old} + (\varepsilon_4^{new} - \varepsilon_2^{old})^\dagger$;
   - $\varepsilon_2^{new} \leftarrow \varepsilon_2^{old} + (\varepsilon_3^{new} - \varepsilon_1^{old})^\dagger$;
   - $\varepsilon_1^{new} \leftarrow \varepsilon_1^{old} + (\varepsilon_2^{new} - \varepsilon_0^{old})^\dagger$;
   - $\overline{u_{i+1,\cdot}} \leftarrow u_{i,\cdot} + (\varepsilon_1^{new})^\dagger$; $u_{i+1,\cdot} \leftarrow \overline{u_{i+1,\cdot}}$.



$\overline{u_{i+1,\cdot}}$, predicted value of $u_{i+1,\cdot}$ used to start the Gauss-Seidel procedure.

   - Gauss-Seidel procedure with (2 bis) : $u_{i+1,j}(0) \longleftarrow \overline{u_{i+1,j}}$, $j = 0, 1, \ldots, N$, instead of (2) : $u_{i+1,j}(0) \longleftarrow u_{i,j}$, $j = 0, 1, \ldots, N$.
     And a *restorage of the subdiagonal*
     (a) $A \leftarrow \varepsilon_1^{old}$, $\varepsilon_1^{old} \leftarrow (\varepsilon_0^{(new)} - \varepsilon_0^{(old)})^\dagger$;
     (b) $B \leftarrow A$, $A \leftarrow \varepsilon_2^{old}$, $\varepsilon_2^{old} \leftarrow u_{i,\cdot} + (\varepsilon_1^{(old)} - B)^\dagger$;
     (c) $C \leftarrow B$, $B \leftarrow A$, $A \leftarrow \varepsilon_m^{old}$, $\varepsilon_m^{old} \leftarrow C + (\varepsilon_{m-1}^{(old)} - B)^\dagger$, $m = 3, 4, \ldots, 6$.

3.1 Remark

Of course, these three methods of prediction (ε_2-prediction, ε_4-prediction and ε_6-prediction) can be generalized to an arbitrary ε_{2k}-prediction for each integer k. However, there is no need to use expensive methods of prediction (i.e. with ε_{2k}-prediction where $k \geq 4$) because our main goal is to reduce the complexity in time when approximating the numerical solution of Burgers problems. Moreover, some problem can occur about numerical stability.

4 Numerical complexity of the algorithms

	Divisions	Multiplications	Additions
vector ε_2-prediction	$3+8(\frac{T}{\Delta t}-2)$	$(N-1)\cdot(3+8(\frac{T}{\Delta t}-2))$	$(N-1)\cdot(4+8(\frac{T}{\Delta t}-2))$
vector ε_4-prediction	$10+16(\frac{T}{\Delta t}-4)$	$(N-1)\cdot(10+16(\frac{T}{\Delta t}-4))$	$(N-1)\cdot(16+16(\frac{T}{\Delta t}-4))$
vector ε_6-prediction	$21+24(\frac{T}{\Delta t}-6)$	$(N-1)\cdot(21+24(\frac{T}{\Delta t}-6))$	$(N-1)\cdot(36+24(\frac{T}{\Delta t}-6))$
Gauss-Seidel	$(N-1)\cdot\sum_{i=1}^{\frac{T}{\Delta t}}\eta_i$	$58(N-1)\cdot\sum_{i=1}^{\frac{T}{\Delta t}}\eta_i$	$49(N-1)\cdot\sum_{i=1}^{\frac{T}{\Delta t}}\eta_i$

In the previous array, η_i is the number of iterations of the Gauss-Seidel procedure (depending on ξ) for computing $u_{i,j}$ (moreover, it is self-evident that η_i depends also on the method of prediction used). Consequently, $(N-1)\eta_i$ is the number of iterations of the Gauss-Seidel procedure for computing $u_{i,.}$, and $(N-1)\cdot\sum_{i=1}^{T/\Delta t}\eta_i$ is the number of iterations of the Gauss-Seidel procedure for computing $u_{i,.}$, $i=1,2,\ldots,T/\Delta t$.

5 Numerical examples

The results are given for 500 terms (i.e. $T/\Delta t = 500$).

Three graphs are given (for each example) which represent the difference between the approximated solution and the computed solution by the following methods

1. u_0 is given. For $i=1,2$, u_i is approximated by using Gauss-Seidel method with $u_{i,j}(0) \longleftarrow u_{i-1,j}$, $j=0,1,\ldots,N$ (2). For $i=3,4,\ldots,500$, u_i is approximated by using Gauss-Seidel method with $u_{i,j}(0) \longleftarrow \overline{u_{i,j}}$, $j=0,1,\ldots,N$ where $\overline{u_{i,j}}$ is predicted by the vector ε_2-prediction based on u_{i-3}, u_{i-2} and u_{i-1} (see Figures 2 and 6).

2. u_0 is given. For $i = 1, 2, 3, 4$, u_i is approximated by using Gauss-Seidel method with $u_{i,j}(0) \longleftarrow u_{i-1,j}$, $j = 0, 1, \ldots, N$ (2). For $i = 5, 6, \ldots, 500$, u_i is approximated by using Gauss-Seidel method with $u_{i,j}(0) \longleftarrow \overline{u_{i,j}}$, $j = 0, 1, \ldots, N$ where $\overline{u_{i,j}}$ is predicted by the vector ε_4-prediction based on u_{i-5}, u_{i-4},u_{i-3}, u_{i-2} and u_{i-1} (see Figures 3 and 7).

3. u_0 is given. For $i = 1, 2, 3, 4, 5, 6$, u_i is approximated by using Gauss-Seidel method with $u_{i,j}(0) \longleftarrow u_{i-1,j}$, $j = 0, 1, \ldots, N$ (2). For $i = 7, 8, \ldots, 500$, u_i is approximated by using Gauss-Seidel method with $u_{i,j}(0) \longleftarrow \overline{u_{i,j}}$, $j = 0, 1, \ldots, N$ where $\overline{u_{i,j}}$ is predicted by the vector ε_6-prediction based on u_{i-7}, u_{i-6}, u_{i-5}, u_{i-4},u_{i-3}, u_{i-2} and u_{i-1} (see Figures 4 and 8).

In the figures, we give the difference $u_{500,j} - \check{u}_{500,j}$, $j = 0, 1, \ldots, N$ (where u denotes the approximated value and $\check{u}$ denotes the predicted/approximated value). In the captions, GS indicates the number of iterations in the Gauss-Seidel method (i.e. $GS = \sum_{i=0}^{T/\Delta t} \eta_i$). In order to take into account the complexity in time for the different methods, we will give the value CT representing the equivalent number of multiplications of the algorithm, which is calculated with the classical rules : for one multiplication or one division, $CT = 1$ and for one addition, $CT = 0$.

5.1 The first example

Parameters of the Burgers problem [2, example 1, p54] : $Re = 100$, $a = 0$, $b = 2$, $\Delta x = 0.05$, $T = 5$, $\Delta t = 0.01$,

$$u_0(x) = \frac{2\pi}{Re} \frac{\frac{1}{4}\sin(\pi x) + \sin(2\pi x)}{1 + \frac{1}{4}\cos(\pi x) + \frac{1}{2}\cos(2\pi x)},$$

$$f(x, t) = 0.$$

Parameter of the method of Gauss-Seidel : $\xi = 1.10^{-10}$.

For these special values of parameters, we can explicitly give the exact solution

$$u_{exact}(x, t) = \frac{2\pi}{Re} \frac{\frac{1}{4}\sin(\pi x)e^{-\pi^2 t/Re} + \sin(2\pi x)e^{-4\pi^2 t/Re}}{1 + \frac{1}{4}\cos(\pi x)e^{-\pi^2 t/Re} + \frac{1}{2}\cos(2\pi x)e^{-4\pi^2 t/Re}}.$$

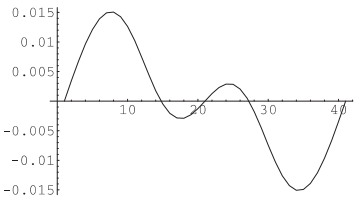


Figure 1: $GS = 4809$, $CT = 11.06 \cdot 10^6$

The Figure 1 represents the approximated solution $u(x,t)$ at $t = T$. The graph of $u_{exact}(x,t)$ at $t = T$ is nearly the same.

5.1.1 Results

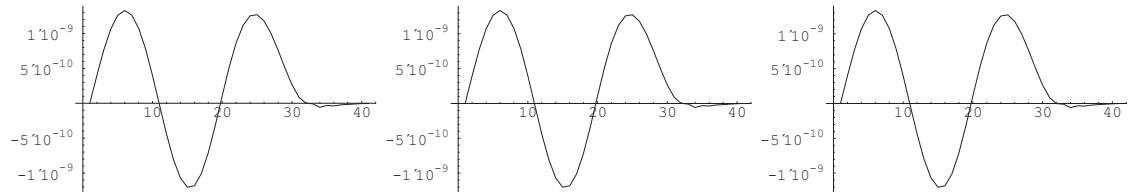


Figure 2: (1) $GS = 2857$, $CT = 6.73 \cdot 10^6$ (2) $GS = 1012$, $CT = 2.64 \cdot 10^6$ (3) $GS = 857$, $CT = 2.45 \cdot 10^6$

5.2 The second example

Parameters of the Burgers problem [2, example 2), p55] : $Re = 100$, $a = 0$, $b = 2$, $\Delta x = 0.05$, $T = 5$, $\Delta t = 0.01$,

$$u_0(x) = \begin{cases} 0 \text{ if } x = 0 \\ 1 \text{ if } 0 < x < 2 \\ 0 \text{ if } x = 2 \end{cases},$$

$$f(x,t) = 0.$$

Parameter of the method of Gauss-Seidel : $\xi = 1.10^{-10}$.

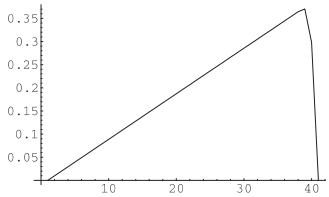


Figure 3: $GS = 6852$, $CT = 15.76 \cdot 10^6$

The Figure 3 represents the approximated solution at $t = T$

5.2.1 Results

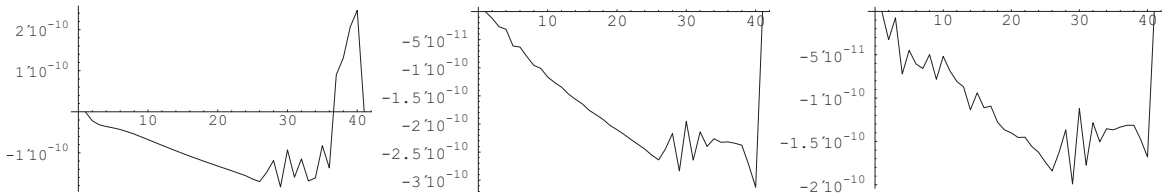


Figure 4: (1) $GS = 3718$, $CT = 8.71 \cdot 10^6$ (2) $GS = 2058$, $CT = 5.05 \cdot 10^6$ (3) $GS = 1535$, $CT = 4.01 \cdot 10^6$

5.3 Remarks

1. It seems that we have obtained the precision expected by the chosen parameter of the method of Gauss-Seidel : $\xi = 1.10^{-10}$ (even for the second example which presents a discontinuity of u_0 at $x = 0$ and at $x = 2$).

2. When we consider the values of CT, in the following table

without prediction	with the vector ε_2-prediction	with the vector ε_4-prediction	with the vector ε_6-prediction	
$11.06 \cdot 10^6$	$6.73 \cdot 10^6$	$2.64 \cdot 10^6$	$2.45 \cdot 10^6$	First example
$15.76 \cdot 10^6$	$8.71 \cdot 10^6$	$5.05 \cdot 10^6$	$4.01 \cdot 10^6$	Second example

   it appears that there surely exists a method of prediction which is better than the others. For example, for the two examples, the use of the vector ε_4-prediction has reduced the complexity in time approximatively to the third of the method without prediction. In fact, there exist two inverse effects :

   (a) the more the method of prediction uses terms, the more the complexity in time for the part of prediction is important ;

   (b) the more the method of prediction uses terms, the better is the method of prediction and the more it reduces the complexity in time for the part of approximation.

   The combination of these effects is often very performing when using methods of prediction as the vector ε_2-prediction or the vector ε_4-prediction.

6 Conclusion

Mixing approximation and prediction is certainly the good way of using methods of prediction. As used here, the method of prediction has no effect on the precision obtained. Moreover, it is very performing when using very simple methods of prediction (vector ε_2-prediction or vector ε_4-prediction).

Theoretical properties of consistency in column and in diagonal for vector ε-prediction are under consideration.

References

[1] C. Brezinski, M. Redivo Zaglia, *Extrapolation Methods. Theory and Practice*, North Holland, Amsterdam, 1991.

[2] M. Morandi Cecchi, R. Nociforo, P. Patuzzo Grego, Space-time finite elements numerical solution of Burgers problems, *Le Matematiche*, Vol. LI, Fasc. I, pp. 43–57, 1996.

[3] M. Morandi Cecchi, R. Nociforo, P. Patuzzo Grego, Burgers problems, Theoretical results, *Italian Journal of Pure and Applied Mathematics*, N. 2, pp. 159–174, 1997.

[4] M. Morandi Cecchi, M. Redivo Zaglia, G. Scenna, Approximation of the numerical solution of parabolic problems, *Computational and Applied Mathematics I*, 71–80, 1992.

[5] M. Prévost, D. Vekemans, Partial Padé prediction, *Numer. Algorithms* 20, pp.23–50, 1999.

[6] D. Vekemans, Algorithmes pour méthodes de prédiction, *Thèse*, Université des Sciences et Technologies de Lille, 1995.

[7] P. Wynn, On a device for computing the $e_m(S_n)$ transformation. *MTAC*, 10:91–96,1956.