MULTILEVEL SCHEMES FOR SOLVING UNSTEADY EQUATIONS

O. GOYON

Laboratoire d'Analyse Numérique, CNRS et Université Paris-Sud, F-91405 Orsay, France

SUMMARY

In this paper we propose and study various multilevel schemes for solving unsteady equations. Numerical comparisons between the choices of discretization and discussions are made on the Burgers equation in one and two dimensions. In particular we prove the advantage of using a hierarchical ADI solver.

KEY WORDS: incremental unknowns; finite difference; multilevel discretization

1. INTRODUCTION

A number of new proposals have been advanced during the last decade in dynamical systems theory about inertial manifolds and approximate inertial manifolds by Foias *et al.*[1,2] and Jolly *et al.*[3] respectively. The associated numerical methods are called non-linear Galerkin methods (as opposed to usual Galerkin methods which project equations on linear manifolds). These methods are based on a distinction between some quantities thanks to the notion of small and large scales. In spectral analysis these sizes identify themselves with first and last modes of the decomposition. In finite difference (Incremental Unknowns) and finite element (hierarchical basis) one would rather speak about nodal values and corrections on multilevel meshes and associate large scales with coarse grid unknowns. The application of these new concepts must allow us to carry out long-time scientific computations (especially in fluid mechanics) with several million unknowns, computations henceforth possible thanks to new generations of computers. We also show that those new concepts are especially suitable for the simulation of turbulent flows.

Some previous studies have already been made in spectral analysis and stability improvement has been observed by Debussche *et al.*[4] Unfortunately, it seems that small scales do not vanish enough in finite difference. Hence we have to define particular algorithms and make a first extensive study of multilevel schemes.

There exist many possibilities for solving unsteady equations numerically. A differentiation between the linear part and the non-linear part allows us (for some parabolic equations) to use semi-implicit time differencing for the heat operator (a Crank–Nicolson scheme) and second-order explicit Adams–Bashforth time differencing for the non-linear (convection) terms. From this point of view, the first requirement will be to define efficient schemes, adapted to vectorial computers, for the heat operator.

In this work we will begin by resolving the one-dimensional case. We will propose for the Burgers equation a scheme defining a multilevel resolution, which is assimilated with a direct solver when the number of unknowns plus one is a power of two. In the two-dimensional case we will propose several schemes, some classical, some defined with Incremental Unknowns and some extensions of the one-dimensional case, thanks to ADI formulations. For each scheme we will study computational aspects

CCC 0271–2091/96/100937–23

Received April 1995
Revised August 1995

and performances on vectorial computers. In fact, we will propose here some new treatments of the discretized algebraic equations, allowing us to define, in the near future, non-linear Galerkin methods in finite difference.

The paper is organized as follows. In Section 2 we introduce the Incremental Unknowns concept. In Section 3 we deal with the one-dimensional case of the Burgers equation. We also present a discussion about variations in Incremental Unknowns and their connection with a spectral analysis of the solutions. The two-dimensional case is presented in Section 4. The last section is devoted to concluding remarks.

2. THE INCREMENTAL UNKNOWNS CONCEPT

Incremental Unknowns were introduced by Chen and Temam[5] as a finite difference multilevel method to solve linear elliptic problems. Some first theoretical results were given by the authors.[6]

The prime aspect of Incremental Unknowns schemes is the fact that they give better-conditioned algebraic systems in elliptic-like problems. Several numerical studies have been made in that sense: we can cite Chehab and Temam[7] and Chehab[8] for solving a bifurcation problem, and Poullet[9] and O. Goyon[10,11] for solving non-linear steady problems.

The Incremental Unknowns concept is based on a multilevel discretization in finite difference. The one-dimensional case is especially propitious: the Incremental Unknowns convert the inversion of the linear system on a fine grid (FG) into an inversion on a coarse grid (CG) (as a cyclic reduction). In two dimensions this decoupling does not operate but the new schemes generate better-conditioned linear systems.

We denote by U an unknown of the initial system, by Yan unknown of CG and by Z an unknown of FG, where $Y$and Z are called Incremental Unknowns. We can now define formulae for $Y$and Z. In one dimension we obtain

$$Y_{2i} = U_{2i}, \tag{1}$$

$$Z_{2i+1} = U_{2i+1} - \tfrac{1}{2}(U_{2i} + U_{2i+2}); \tag{2}$$

in two dimensions we obtain

$$\begin{aligned} Y_{2i,2j} &= U_{2i,2j}, \\ Z_{2i+1,2j} &= U_{2i+1,2j} - \tfrac{1}{2}(U_{2i,2j} + U_{2i+2,2j}), \\ Z_{2i,2j+1} &= U_{2i,2j+1} - \tfrac{1}{2}(U_{2i,2j} + U_{2i,2j+2}), \\ Z_{2i+1,2j+1} &= U_{2i,2j} - \tfrac{1}{4}(U_{2i,2j} + U_{2i,2j+2} + U_{2i+2,2j} + U_{2i+2,2j+2}). \end{aligned} \tag{2}$$

A study of the algebra of Incremental Unknowns in three dimensions can be found in Reference 10. Using these formulae recursively on several grids, we are able to define multilevel schemes in Incremental Unknowns.

Let A be the matrix of the linear system where unknowns are set grid by grid and let S be the transfer matrix of (1) or (2). Then the initial system in nodal basis (on one grid) is rewritten as

$$S^{\mathrm{T}}AS\bar{u} = S^{\mathrm{T}}b, \tag{3}$$

where $\bar{u}$ is the reordered vector of Incremental Unknowns and b is the reordered vector of the right member.

Unfortunately, we will see later that, as a general rule, $S^{\mathrm{T}}AS$ is not well conditioned if A is the matrix of the heat operator. Therefore we will be obliged to propose new multilevel formulations.

3. RESOLUTIONS OF THE ONE-DIMENSIONAL BURGERS EQUATION

In a monograph, Burgers[12] described his study (to attempt to define a statistical theory of turbulent fluid motion) of the interactions between dissipative and non-linear terms in an extremely simplified non-linear diffusion equation (now called the Burgers equation). Many problems can be modelled by the Burgers equation. A detailed study has been made by Fletcher.[13] Independently, the Burgers equation can be considered as an approach to the Navier–Stokes equations.

The Burgers equation is similar to the usual transport equation, except that the convective term is non-linear. If the viscous (dissipative) term is dropped, the result is the inviscid Burgers equation. The non-linearity allows discontinuous solutions (shocks) to develop. Several formulations of the equation exist. In a conservative form the Cauchy problem to solve is

$$\begin{gathered}\frac{\partial u(x,t)}{\partial t}+\frac{1}{2}\frac{\partial u^2(x,t)}{\partial x}=0, \quad 0<t<T, \quad x\in\mathbb{R}, \\ u(x,0)=f(x).\end{gathered} \tag{4}$$

The numerical treatment of solutions with shocks goes beyond the purpose of our study. Nevertheless, we can obtain smooth solutions (in a sense to be defined) approaching discontinuous solutions of (4). These are the solutions of the dissipative Burgers equation

$$\begin{gathered}\frac{\partial u(x,t)}{\partial t}+\frac{1}{2}\frac{\partial u^2(x,t)}{\partial x}=\nu\frac{\partial^2 u(x,t)}{\partial x^2}, \quad \nu>0, \quad 0<t<T, \quad x\in\mathbb{R}, \\ u(x,0)=f(x),\end{gathered} \tag{5}$$

where u will be called thereinafter the velocity and f is the initial condition.

The effect of the viscous term in (5) prevents multivalued solutions but permits severe gradients. Hence the dissipative Burgers equation is very suitable for testing computational algorithms for flows. Another reason for the success of this equation is the possibility of obtaining explicitly exact solutions of (5) for many initial conditions thanks to the Cole– Hopf[14,15] transformation. An extensive study of these exact solutions has been made by Benton and Platzman.[16]

It is also possible to solve exactly the general problem with arbitrary, time- dependent boundary values on u: $u(x,t)=1-2\sqrt{\nu}\tanh[(x-t)/\sqrt{\nu}]$ is one of the known solutions. This last solution has been used to verify the accuracy of our schemes (see Figure 1 in Section 3.2).

3.1. Discretizations of the equation

We have chosen to solve (5) on a bounded domain $[a, b]$ and on a uniform mesh in finite difference. In this subsection we present several discretizations of the equation and we will justify the choice of multilevel schemes.

Notation. N is the number of interior nodes (without boundaries), $h=(b-a)/(N+1)$ is the space step and δt is the time step. We denote by $U_i^n=u(x_i,t_n)$ the solution at node $x_i=a+i\times h$ time $t_n=n\times\delta t$. We denote by $\mathscr{L}$ the linear heat operator of (5) and by $\mathscr{B}$ the non-linear operator. We can thus reformulate (5) as

$$\begin{gathered}\mathscr{L}u+\mathscr{B}(u)=0, \quad 0<t<T, \quad x\in]a,b[, \\ u(a,t)=u_l(t), \qquad u(b,t)=u_r(t), \qquad u(x,0)=f(x).\end{gathered} \tag{6}$$

Discretization in nodal basis. If we choose to use a Crank–Nicolson time differencing for $\mathscr{L}$, second-order explicit Adams–Bashforth time differencing for $\mathscr{B}$ and second-order centred schemes for all derivatives, the discretization gives

$$\begin{aligned}
&\left(1+\frac{\nu\delta t}{h^2}\right)U_i^n-\frac{\nu\delta t}{2h^2}(U_{i-1}^n+U_{i+1}^n)\\
&\quad=\left(1-\frac{\nu\delta t}{h^2}\right)U_i^{n-1}+\frac{\nu\delta t}{2h^2}(U_{i-1}^{n-1}+U_{i+1}^{n-1})\\
&\quad-\left(\frac{3\delta t}{8h}(U_{i-1}^{n-1}+U_{i+1}^{n-1})(U_{i+1}^{n-1}-U_{i-1}^{n-1})\right.\\
&\quad\left.-\frac{\delta t}{8h}(U_{i-1}^{n-2}+U_{i+1}^{n-2})(U_{i+1}^{n-2}-U_{i-1}^{n-2})\right),\qquad 1\leqslant i\leqslant N,\\
&\quad U_0^n=u_1(n\times\delta t),\qquad U_{N+1}^n=u_r(n\times\delta t).
\end{aligned}\tag{7}$$

One can easily prove that system (7) is of second order in space and time. Let A be the Laplacian matrix, I the identity matrix and $NL(\ldots)$ the discretization of the non-linear term. A matrix formulation of (7) leads us to

$$\left(I+\frac{\nu\delta t}{2}A\right)U^n=\left(I-\frac{\nu\delta t}{2}A\right)U^{n-1}-NL(U^{n-1},U^{n-2}).\tag{8}$$

A two-level discretization. This discretization is the result of substitutions between equations. Rewriting system (7) at nodes $2i$, $2i-1$ and $2i+1$, we obtain

$$\alpha U_{2i}^n-\beta(U_{2i+1}^n+U_{2i-1}^n)=f_{2i},\tag{9a}$$

$$\alpha U_{2i-1}^n-\beta(U_{2i-2}^n+U_{2i}^n)=f_{2i-1},\tag{9b}$$

$$\alpha U_{2i+1}^n-\beta(U_{2i}^n+U_{2i+2}^n)=f_{2i+1},\tag{9c}$$

with

$$\begin{aligned}
&\alpha=1+\frac{\nu\delta t}{h^2},\qquad\beta=\frac{\nu\delta t}{2h^2},\\
&f_i=\left(1-\frac{\nu\delta t}{h^2}\right)U_i^{n-1}+\frac{\nu\delta t}{2h^2}(U_{i-1}^{n-1}+U_{i+1}^{n-1})\\
&\quad-\left(\frac{3\delta t}{8h}(U_{i+1}^{n-1}+U_{i-1}^{n-1})(U_{i+1}^{n-1}-U_{i-1}^{n-1})-\frac{\delta t}{8h}(U_{i+1}^{n-2}+U_{i-1}^{n-2})(U_{i+1}^{n-2}-U_{i-1}^{n-2})\right).
\end{aligned}\tag{10}$$

Eliminating U_{2i-1}^n and U_{2i+1}^n in (9a) and (9c), we get a new formulation of (9):

$$(\alpha^2-2\beta^2)U_{2i}^n-\beta^2(U_{2i-2}^n+U_{2i+2}^n)=\alpha f_{2i}+\beta(f_{2i-1}+f_{2i+1}),\tag{11a}$$

$$U_{2i-1}^n=\frac{\beta}{\alpha}(U_{2i-2}^n+U_{2i}^n)+\frac{1}{\alpha}f_{2i-1},\tag{11b}$$

$$U_{2i+1}^n=\frac{\beta}{\alpha}(U_{2i}^n+U_{2i+2}^n)+\frac{1}{\alpha}f_{2i+1}.\tag{11c}$$

System (11) consists of a linear system to solve only on a coarse grid (with $(N-1)/2$ unknowns when system (7) has N unknowns).

A multilevel discretization. Defining

$$\alpha_1 = \alpha, \qquad \beta_1 = \beta, \tag{12}$$

$$\alpha_2 = (\alpha_1)^2 - 2(\beta_1)^2, \qquad \beta_2 = (\beta_1)^2, \qquad f^2_{2i} = \alpha_1 f_{2i} + \beta_1 (f_{2i-1} + f_{2i+1}), \tag{13}$$

we can rewrite (11a) as

$$\alpha_2 U^n_{2i} - \beta_2 (U^n_{2i-2} + U^n_{2i+2}) = f^2_{2i}. \tag{14}$$

Thus we can repeat the substitution procedure from the old grid. If $N = 2^L - 1$, we obtain on k levels (for $k < L$) a linear system to solve on a coarse grid,

$$\alpha_k U^n_{2^{k-1}i} - \beta_k (U^n_{2^{k-1}i-2^{k-1}} + U^n_{2^{k-1}i+2^{k-1}}) = f^k_{2^{k-2}i}, \tag{15}$$

and solutions are obtained recursively on fine grids for $l = k-1, \dots, 1$,

$$\begin{aligned} U^n_{2^l i-2^{l-1}} &= \frac{1}{\alpha_l} f^l_{2^l i-2^{l-1}} + \frac{\beta_l}{\alpha_l}(U^n_{2^l i-2^l} + U^n_{2^l i}), \\ U^n_{2^l i+2^{l-1}} &= \frac{1}{\alpha_l} f^l_{2^l i+2^{l-1}} + \frac{\beta_l}{\alpha_l}(U^n_{2^l i} + U^n_{2^l i+2^l}), \end{aligned} \tag{16}$$

with

$$\begin{aligned} \alpha_l = (\alpha_{l-1})^2 - 2(\beta_{l-1})^2, &\qquad \beta_l = (\beta_{l-1})^2, \\ f^l_{2^{l-1}i} = \alpha_{l-1} f^{l-1}_{2^{l-1}i} + \beta_{l-1}(f^{l-1}_{2^{l-1}i-2^{l-2}} &+ f^{l-1}_{2^{l-1}i+2^{l-2}}). \end{aligned} \tag{17}$$

If $k = L$, there is no linear system like (15) to solve; all unknowns are explicitly obtained.

Multilevel discretizations using Incremental Unknowns. If we choose to apply Incremental Unknowns to system (8), we obtain (with the notation of Section 2)

$$S^{\mathrm{T}}\left(I + \frac{\nu \delta t}{2} A\right) S \bar{U}^n = S^{\mathrm{T}}\left[\left(I - \frac{\nu \delta t}{2} A\right) U^{n-1} - NL(U^{n-1}, U^{n-2})\right]. \tag{18}$$

We also can apply Incremental Unknowns formulae (1) to (9). It is straightforward to see that the resulting systems are

$$\alpha_k Y^n_{2^{k-1}i} - \beta_k (Y^n_{2^{k-1}i-2^{k-1}} + Y^n_{2^{k-1}i+2^{k-1}}) = f^k_{2^{k-2}i}, \tag{19}$$

$$\begin{aligned} Z^n_{2^l i-2^{l-1}} &= \frac{1}{\alpha_l} f^l_{2^l i-2^{l-1}} + \left(\frac{\beta_l}{\alpha_l} - \frac{1}{2}\right)(Y^n_{2^l i-2^l} + Y^n_{2^l i}), \\ Z^n_{2^l i+2^{l-1}} &= \frac{1}{\alpha_l} f^l_{2^l i+2^{l-1}} + \left(\frac{\beta_l}{\alpha_1} - \frac{1}{2}\right)(Y^n_{2^l i} + Y^n_{2^l i+2^l}). \end{aligned} \tag{20}$$

To avoid numerical overflows, we replace relations (17) by

$$\begin{aligned} \alpha_l = \alpha_{l-1} - 2\frac{(\beta_{l-1})^2}{\alpha_{l-1}}, &\qquad \beta_l = \frac{(\beta_{l-1})^2}{\alpha_{l-1}}, \\ f^l_{2^{l-1}i} = f^{l-1}_{2^{l-1}i} + \frac{\beta_{l-1}}{\alpha_{l-1}}(f^{l-1}_{2^{l-1}i-2^{l-2}} &+ f^{l-1}_{2^{l-1}i+2^{l-2}}). \end{aligned} \tag{21}$$

A study of the sequence $\gamma_l = \beta_l / \alpha_l$ exhibits the convergence of α_l to α and of β_l to zero in (21).

3.2. Numerical results

In this subsection we begin by considering an exact analytical solution of the Burgers equation (5) with Dirichlet boundary conditions. This solution is useful to verify the accuracy of the schemes and to validate the Fortran programs. We present in Figure 1 the time evolution of this solution.

We now consider a sine as an initial condition, with the computational domain [0, 1]:

$$u(x, t = 0) = \sin(2\pi x). \tag{22}$$

This is quite an interesting numerical test thanks to the growth and control (with the viscosity) of a severe gradient near $x = 0{\cdot}5$ (see Figure 3 in Section 3.3). On the other hand, the spatial periodicity of the solution allows us a spectral study of the results (see Section 3.4). The Reynolds number is given by $Re = 1/\nu$. This last example has been solved by the numerical schemes proposed in the previous subsection.

To make a comparison of CPU times between the previous schemes, we have chosen to set the final time of computation at $t = 1{\cdot}00$ (100 time steps). All results are given in Table I. These tests have been done on a SUNsparc 10.

By examining Table II, one observes that the condition number of

$$M = S^{\mathrm{T}}\left(I + \frac{\nu\delta t}{2}A\right)S$$

decreases as the number of grids increases (as a general rule this is not true). Since we solve the linear system by a conjugate gradient method, the number of iterations is a function of the condition number of the matrix. This result explains the decrease in CPU time when the number of grids increases with (18). When the scheme (19), (20) is used, we reduce the size of the linear system by increasing the number of grids. When 11 grids are chosen, there is no linear system to solve, so the resolution is very fast (1·5 s).

Theoretically we do not know how to choose a valid time step because of the non-linear term. Experimentally we observe on curves the appearance of small oscillations of the velocity when we increase the time step at a fixed number of nodes. For $N = 2049$ this leads us to choose $\delta t = 0{\cdot}01$ at

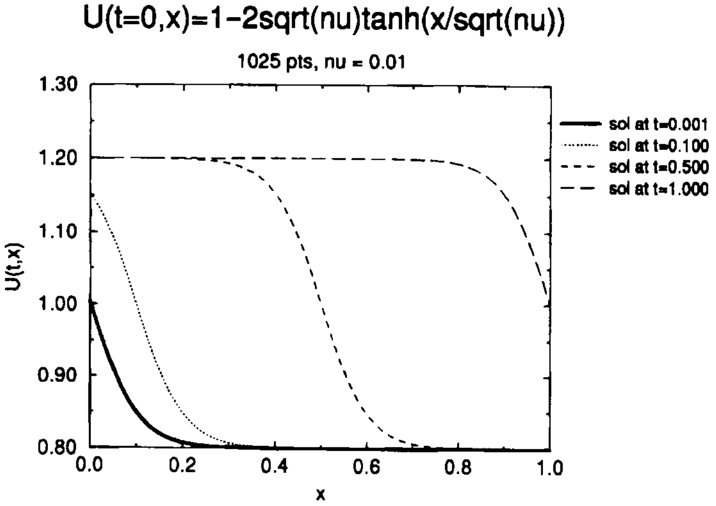


Figure 1. Time evolution of an exact analytical solution

Table I. CPU time comparison: $\delta t = 0{\cdot}01$, $Re = 100$, $t = 1{\cdot}00$

Scheme	(8)	(18)	(18)	(18)	(19), (20)	(19), (20)	(19), (20)
Number of grids	1	1	2	11	1	2	11
Number of nodes	2049	2049	2049	2049	2049	2049	2049
CPU time (s)	64	73	50	37	71	18	1·5

Table II. Condition number of heat operator in Incremental Unknowns

Scheme	(18)	(18)	(18)
Number of grids	1	2	11
Condition number $\kappa(M)$	824	209	155

$Re = 100$, $\delta t = 0{\cdot}005$ at $Re = 200$ and $\delta t = 0{\cdot}002$ at $Re = 500$. We clearly see that the time step depends on the viscosity. Classically we also consider that the stability condition is of CFL type.

3.3. *A study of Incremental Unknowns*

In this subsection we observe the numerical behaviour of Incremental Unknowns. The study is devoted to (i) the time evolution of $|Z_l|/|U|$, (ii) the time evolution of the solution in Incremental Unknowns and (iii) the effect of viscosity.

Figure 2 shows the time evolution of relative norms of Z_l with respect to U. The coarse grid has 63^2 nodes.

The level-by-level evolution is visible thanks to the logarithmic scale. We observe different orders of magnitude. We also observe a general growth (by a factor of 100 for $Re = 500$) of norms from $t = 0{\cdot}1$ to

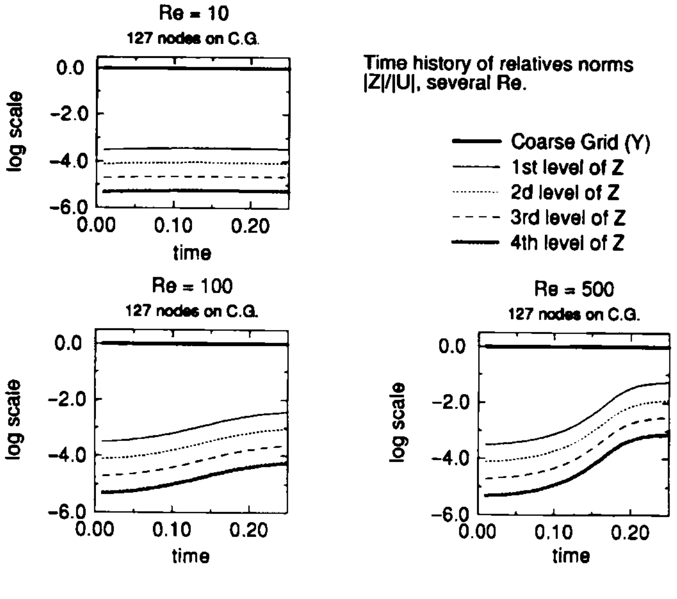


Figure 2. Time history of relative norms; 2049 nodes, several *Re*

0·3. This last result can easily be explained from relation (1) and Figure 3: the value of Z must be larger than expected in the region of severe gradients and more generally in the region of non-uniform severe gradients. The effect of viscosity reduces the gradients and we finally obtain a smooth solution. We obtain our first result: we can observe the existence of localized gradients thanks to variations in relatives norms of $|Z_l|/|U|$ (i.e. gradients generate comparatively high values of Incremental Unknowns).

To underline this effect, in Figure 4 we compare values of Incremental Unknowns at $t = 0{\cdot}25$ for several *Re*. When *Re* increases, the black band indicates that some Incremental Unknowns are not small around $x = 0{\cdot}5$. For $Re = 500$, close to $x = 0{\cdot}5$, fine grids are locally of the order of the coarse grid (this is not true for $Re = 10$).

We are also interested in an analysis of the time evolution of Incremental Unknowns. On this subject we plot in Figures 5 and 6 the absolute values of variations in the solution (in Incremental Unknowns) between two time steps ($t = 0{\cdot}248$ and 0·25) in accordance with the grid level and *Re*. In Figure 6 we divide each spatial variation by the nodal solution. We observe that, at particular nodes, variations are not connected to a grid level (see e.g. $Re = 500$ and $x \in [0{\cdot}45,\ 0{\cdot}55]$): we cannot distinguish a hierarchy of variations as for $Re = 10$. Thus we obtain our second result: time variations of Incremental Unknowns may not be as small as expected.

In short, even if global variations in Incremental Unknowns are small, locally they may be significant. This result causes problems when we want to neglect or estimate these quantities (in the spirit of non-linear Galerkin methods). A solution is to practise a strategy of local mesh refinement; results in this direction will be given in the near future.

3.4. Some spectral aspects

In this subsection we have two different aims: first, to justify by an *a posteriori* analysis our discretization choices, and equally, to connect the previous study of Incremental Unknowns with a spectral analysis.

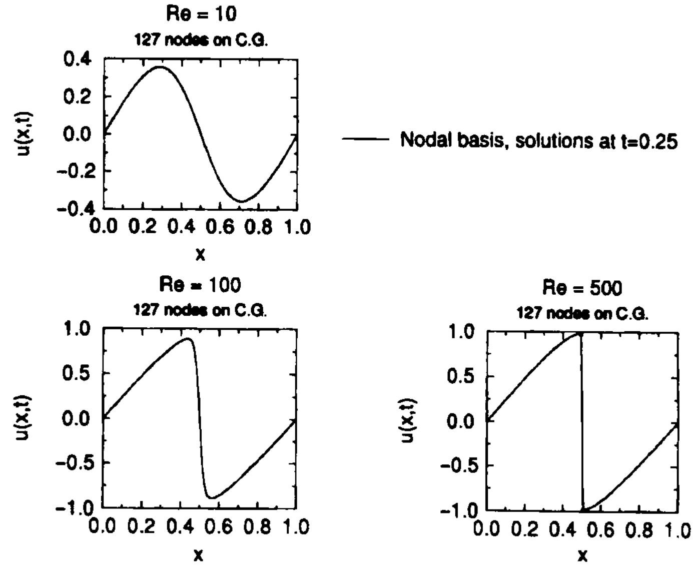


Figure 3. Solutions at $t = 0{\cdot}25$; 2049 nodes, several *Re*

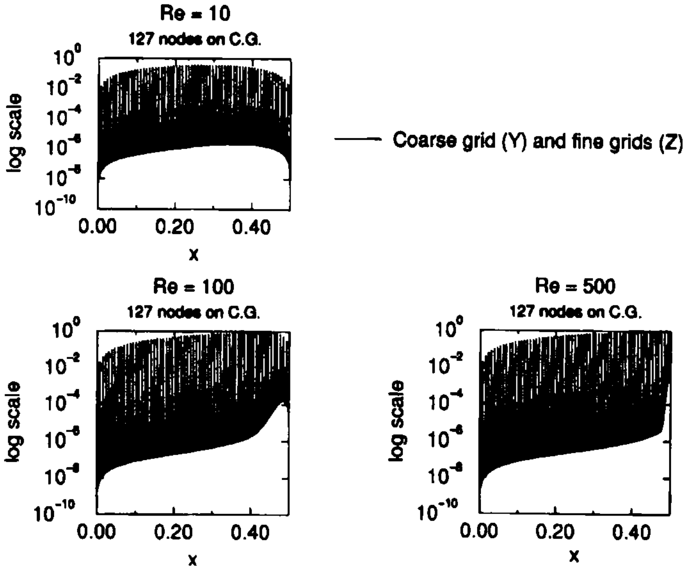


Figure 4. Solutions at $t=0{\cdot}25$; 127 nodes on coarse grid, several *Re*

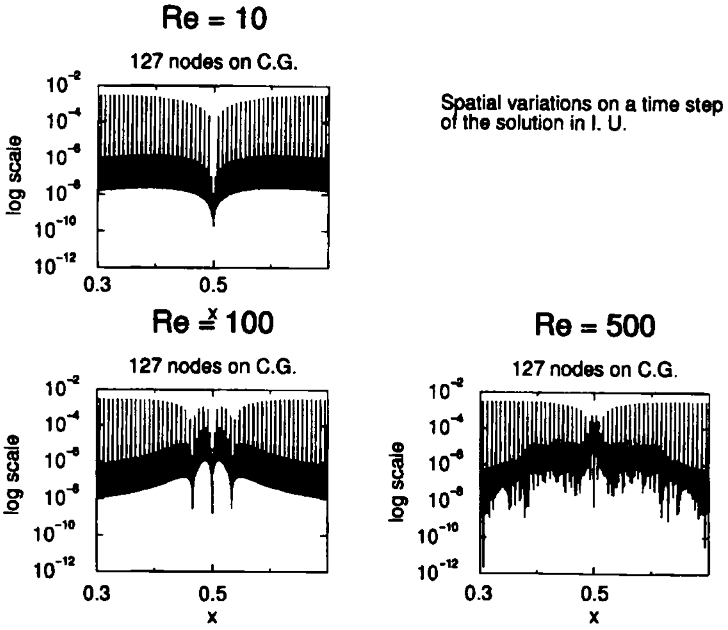
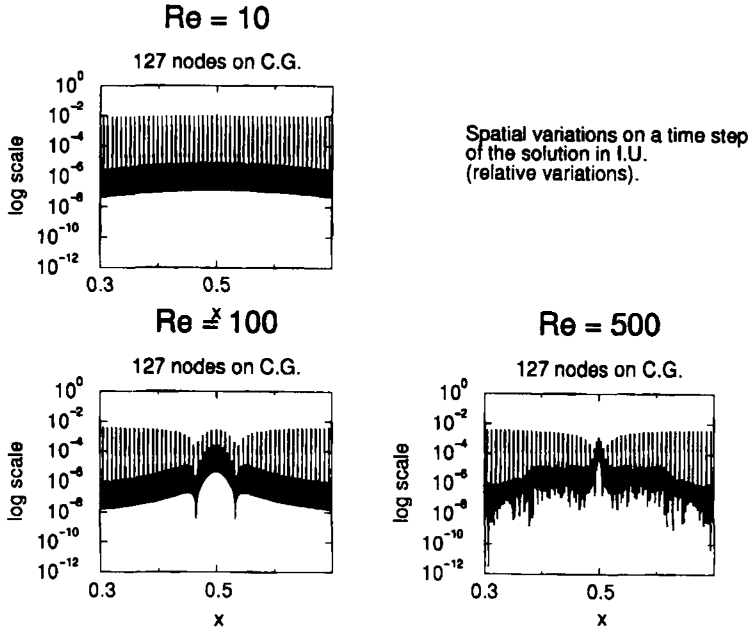


Figure 5. Spatial variations during a time step; 2049 nodes, several *Re*



Spatial variations on a time step of the solution in I.U. (relative variations).

Figure 6. Relative spatial variations during a time step; 2049 nodes, several *Re*

The next figures present results of the energy spectrum for several numbers of nodes and several values of *Re*. We define by E the energy spectrum

$$E = \sum_{k=1}^{N/2} E_k,$$

where $E_k = \hat{u}^2(k) + \hat{u}^2(-k)$ (E_k is the kth mode of the spectrum). Figure 7 explains why we have decided to take at least 1025 nodes at $t = 0{\cdot}25$ and $Re = 500$. By varying the mesh size (by adding new grids of Incremental Unknowns), we observe variations in the spectrum size until $N = 1025$ (500 modes). The mesh size is a function of *Re*: the spectrum size increases with *Re* (see Figure 8).

What is the time behaviour of the energy spectrum? We will try to answer this question by solving the equation with a random initial condition:

$$u(x, t = 0) = \sum_{l=1}^{n_l} \alpha_i \sin(2\pi l x). \tag{23}$$

Thus we generate a solution with n_l modes (we have chosen $n_l = 255$ and $Re = 100$); α_l are random numbers. We choose to take $N = 1025$ (a higher N gives the same results).

In Figure 9 we present the time variation of the energy spectrum. We observe a growth of the bandwidth between $t = 0{\cdot}0$ and $0{\cdot}01$. Then the dissipative effects of the viscosity reduce it. Thus we choose in Figure 10 two mesh sizes and compare solutions at $t = 0{\cdot}01$. There are obvious differences between the solutions (4097 nodes would have given quasi- identical results as 1025 nodes): the spectrum climb before $t = 0{\cdot}01$ seems not to be artificial and a fine discretization is essential even if the initial condition is well represented by a coarse discretization. We observe here effects of the non-linear term. How do Incremental Unknowns behave towards this type of solution? An answer is given in

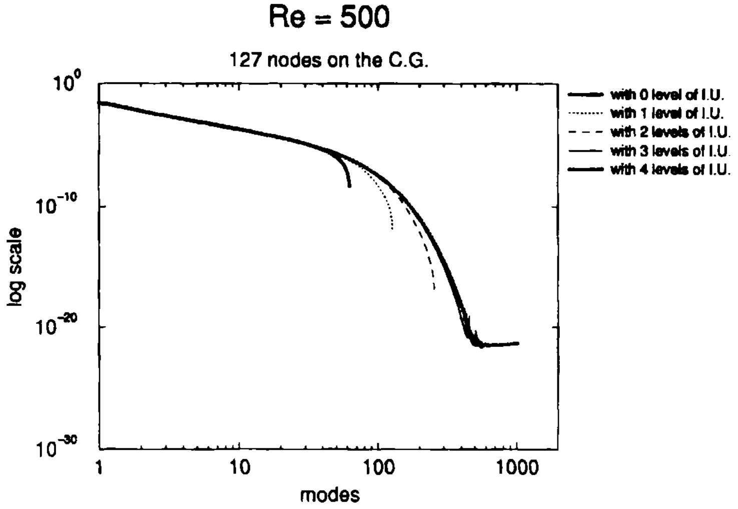


Figure 7. Energy spectrum; several numbers of nodes, $Re = 500$, $t = 0{\cdot}25$

Figure 11: we clearly observe first the non-linear effect (between $t = 0{\cdot}0$ and $0{\cdot}04$) and then the viscous effect. During the first part of the time evolution we cannot distinguish any hierarchy of the unknowns; after dissipative (smoothing) effects we find henceforth usual results (see Figure 2).

In short, conjugated effects of viscous and convective terms in the one- dimensional Burgers equation allow solutions to have severe gradients during some time steps and smooth oscillations during other time steps. Incremental Unknowns appear to be a new manner to study this type of solution.

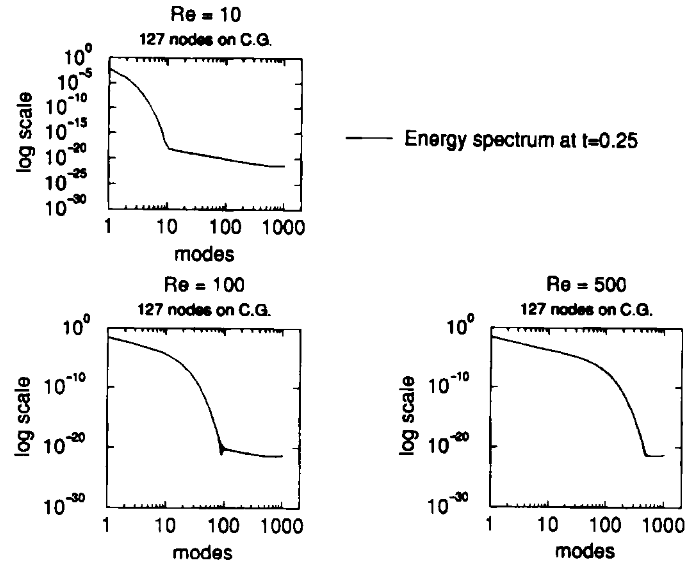


Figure 8. Variation in spectrum size; several Re, $t = 0{\cdot}25$

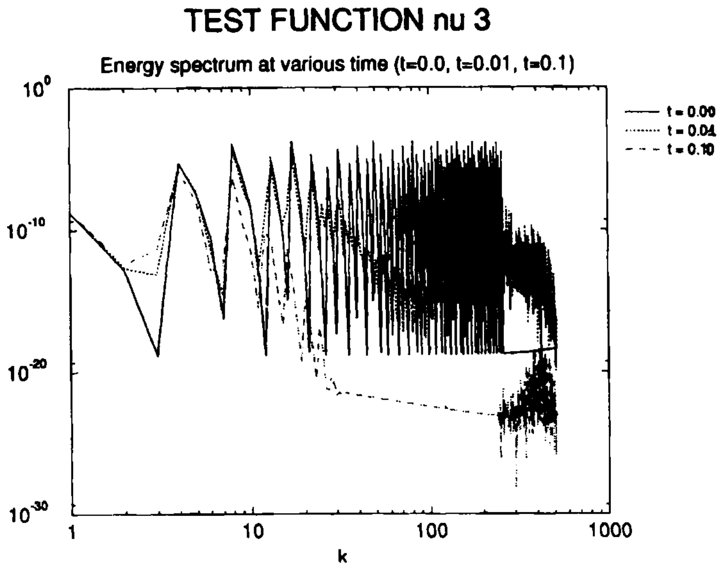


Figure 9. Time variation of energy spectrum

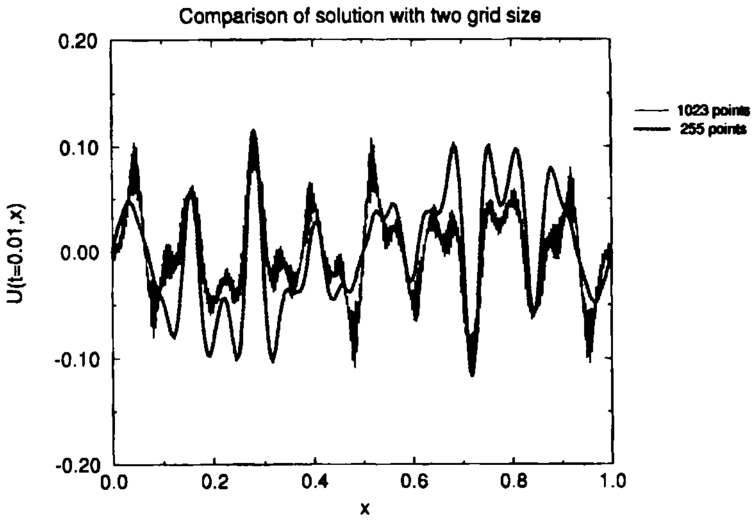


Figure 10. Comparison of two mesh sizes

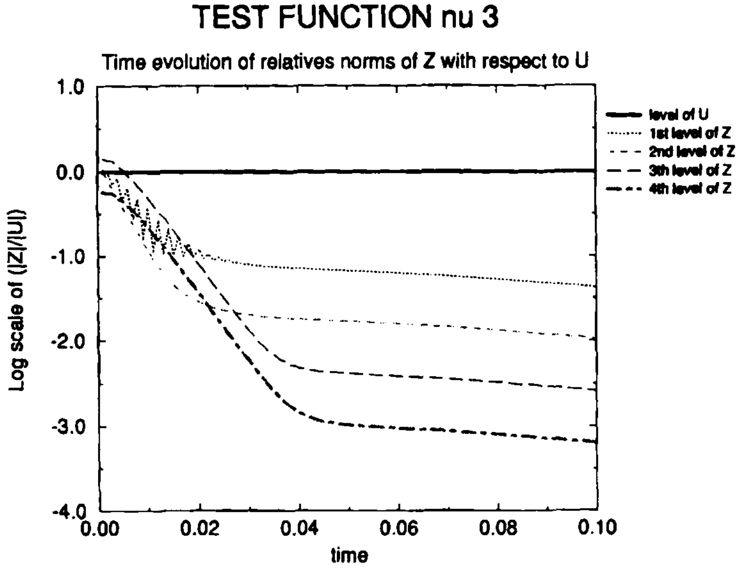


Figure 11. Time history of relative norms for third test function

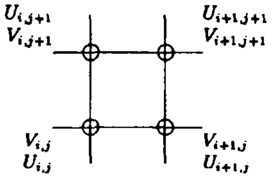
4. THE TWO-DIMENSIONAL CASE

The one-dimensional Burgers equation can be extended to multiple dimensions. The two-dimensional Burgers equations are

$$\begin{aligned}\frac{\partial u(x,y,t)}{\partial t}-\frac{1}{Re}\nabla^2u(x,y,t)+u(x,y,t)\frac{\partial u(x,y,t)}{\partial x}+v(x,y,t)\frac{\partial u(x,y,t)}{\partial y}&=f(x,y,t),\\ \frac{\partial v(x,y,t)}{\partial t}-\frac{1}{Re}\nabla^2v(x,y,t)+u(x,y,t)\frac{\partial v(x,y,t)}{\partial x}+v(x,y,t)\frac{\partial v(x,y,t)}{\partial y}&=(x,y,t),\end{aligned}\tag{24}$$

where *Re* is a strictly positive parameter. These equations coincide with the two-dimensional Navier–Stokes equations if the pressure and the divergence-free equation are neglected. As in the one-dimensional case, exact solutions can be constructed by using an extension of the Cole–Hopf transformation. In two dimensions, only steady solutions are given by Fletcher[17] in the literature.

We have decided to discretize velocity unknowns (u and v) on each mesh node, so we obtain



In this section we carry out a study and comparison of two families of numerical methods. The first one assumes a discretization of non-linear terms by a second-order Adams–Bashforth scheme (AB2) and of linear terms by a Crank–Nicolson scheme (CN). The second family uses the concept of alternating directions (ADI) that we will develop later. In both cases we present and discuss the results for nodal basis with those of multilevel schemes. We are also interested in computational efficiency.

4.1. *A first family of schemes*

We have chosen to solve (24) on a rectangular domain $[0, a] \times [0, b]$ with the following notation: h_x and h_y are the space steps and δt is the time step. We denote by $U^n_{i,j}$ the solution at node $(x_i = (i-1) \times h_x, y_j = (j-1) \times h_y)$ and time $t_n = n \times \delta t$.

Discretization in nodal basis. Our choice of discretization (CN + AB2) gives

$$\begin{aligned}\frac{U^n_{i,j} - U^{n-1}_{i,j}}{\delta t} = &-\frac{1}{2Re \cdot h_x^2}(2U^n_{i,j} - U^n_{i-1,j} - U^n_{i+1,j}) - \frac{1}{2Re \cdot h_y^2}(2U^n_{i,j} - U^n_{i,j-1} - U^n_{i,j+1})\\
&-\frac{1}{2Re \cdot h_x^2}(2U^{n-1}_{i,j} - U^{n-1}_{i-1,j} - U^{n-1}_{i+1,j}) - \frac{1}{2Re \cdot h_y^2}(2U^{n-1}_{i,j} - U^{n-1}_{i,j+1})\\
&-\frac{3}{8h_x}(U^{n-1}_{i+1,j} + U^{n-1}_{i-1,j})(U^{n-1}_{i+1,j} - U^{n-1}_{i-1,j}) + \frac{1}{8h_x}(U^{n-2}_{i+1,j} + U^{n-2}_{i-1,j})(U^{n-2}_{i+1,j} - U^{n-2}_{i-1,j})\\
&-\frac{3}{8h_y}(V^{n-1}_{i,j+1} + V^{n-1}_{i,j-1})(U^{n-1}_{i,j+1} - U^{n-1}_{i,j-1}) + \frac{1}{8h_y}(V^{n-2}_{i,j+1} + V^{n-2}_{i,j-1})(U^{n-2}_{i,j+1} - U^{n-2}_{i,j-1})\\
&+\tfrac{1}{2}(f^n_{i,j} + f^{n-1}_{i,j})\end{aligned}$$

for the first component (u) of the velocity and a similar linear system for the second component (v) of the velocity. These discretizations are of second order in space and time. Using the same matrix notation as in one dimension, we can rewrite the last equations as

$$\begin{aligned}\left(I + \frac{\delta t}{2Re}A\right)U^n &= \left(I - \frac{\delta t}{2Re}A\right)U^{n-1} + \delta t\left(\frac{f^n + f^{n-1}}{2} - NL_1(U, V)\right),\\
\left(I + \frac{\delta t}{2Re}A\right)V^n &= \left(I - \frac{\delta t}{2Re}A\right)V^{n-1} + \delta t\left(\frac{g^n + g^{n-1}}{2} - NL_2(U, V)\right).\end{aligned} \tag{26}$$

As in the one-dimensional case, the explicit discretization of the non-linear terms implies (numerically confirmed) a stability condition on the time step.

Discretizations using Incremental Unknowns. In this subsection we are interested in applying relations (2) to system (26). Unlike in the one- dimensional case, there is no decoupling between the coarse grid and the fine grid. Using Incremental Unknowns notation, the classical matrix formulation is

$$\begin{aligned}S^{\mathrm{T}}\left(I + \frac{\delta t}{2Re}A\right)S\bar{U}^n &= S^{\mathrm{T}}\left[\left(I - \frac{\delta t}{2Re}A\right)U^{n-1} + \delta t\left(\frac{f^n - f^{n-1}}{2} - NL_1(U, V)\right)\right],\\
S^{\mathrm{T}}\left(I + \frac{\delta t}{2Re}A\right)S\bar{V}^n &= S^{\mathrm{T}}\left[\left(I - \frac{\delta t}{2Re}A\right)V^{n-1} + \delta t\left(\frac{g^n + g^{n-1}}{2} - NL_2(U, V)\right)\right].\end{aligned} \tag{27}$$

One can also try to solve a non-symmetric version of (27):

$$\begin{aligned}\left(I + \frac{\delta t}{2Re}S^{-1}AS\right)\bar{U}^n &= S^{-1}\left[\left(I - \frac{\delta t}{2Re}A\right)U^{n-1} + \delta t\left(\frac{f^n + f^{n-1}}{2} - NL_1(U, V)\right)\right],\\
\left(I + \frac{\delta t}{2Re}S^{-1}AS\right)\bar{V}^n &= S^{-1}\left[\left(I - \frac{\delta t}{2Re}A\right)V^{n-1} + \delta t\left(\frac{g^n + g^{n-1}}{2} - NL_2(U, V)\right)\right].\end{aligned} \tag{28}$$

Remark. In system (28) the explicit discretization of the non-linear terms is hierarchic thanks to the matrix S^{-1}. This allows multilevel estimations of linear and non-linear quantities and future new schemes.

4.2. A second family of schemes: ADI schemes

The alternating direction implicit method was developed originally by Peaceman and Rachford[18] and Douglas[19] for the resolution of multidimensional problems. The main idea is to split a multidimensional operator (in our case the two-dimensional heat operator) into several monodimensional operators which are easier to solve numerically. Many applications have been made with ADI schemes, especially for elliptic and hyperbolic problems, e.g. by Douglas and Gunn.[20]

Multilevel discretizations. By using our multilevel schemes (detailed in Section 3) on each monodimensional problem, we are able to define a solver reducing to a maximum the size of linear systems to invert. Let us present a two-grid discretization; it is straightforward with (21) to extend formulae on any number of grids. During the first half-time step and for each j, solve

$$\begin{aligned}(\alpha_x^2 - 2\beta_x^2)U_{2i,j}^{n+1/2} - \beta_x^2(U_{2i-2,j}^{n+1/2} + U_{2i+2,j}^{n+1/2}) &= \alpha_x F_{2i,j}^{n,n-1} + \beta_x(F_{2i-1,j}^{n,n-1} + F_{2i+1,j}^{n,n-1}),\\ U_{2i+1,j}^{n+1/2} &= \frac{1}{\alpha_x}F_{2i+1,j}^{n,n-1} + \frac{\beta_x}{\alpha_x}(U_{2i,j}^{n+1/2} + U_{2i+2,j}^{n+1/2}),\\ U_{2i-1,j}^{n+1/2} &= \frac{1}{\alpha_x}F_{2i-1,j}^{n,n-1} + \frac{\beta_x}{\alpha_x}(U_{2i,j}^{n+1/2} + U_{2i-2,j}^{n+1/2}),\end{aligned} \tag{29}$$

with

$$\alpha_x = 1 + \frac{\delta t}{Re \cdot h_x^2}, \qquad \beta_x = \frac{\delta t}{2Re \cdot h_x^2}$$

and right members of the kind

$$\begin{aligned}F_{i,j}^{n,n-1} = {}&U_{i,j}^n - \frac{\delta t}{2Re \cdot h_y^2}(2U_{i,j}^n - U_{i,j-1}^n - U_{i,j+1}^n) - \alpha\biggl(\frac{3\delta t}{8h_x}(U_{i+1,j}^n + U_{i-1,j}^n)(U_{i+1,j}^n - U_{i-1,j}^n)\\ &+ \frac{\delta t}{8h_x}(U_{i+1,j}^{n-1} + U_{i-1,j}^{n-1})(U_{i+1,j}^{n-1} - U_{i-1,j}^{n-1}) - \frac{3\delta t}{8h_y}(V_{i,j+1}^n + V_{i,j-1}^n)(U_{i,j+1}^n - U_{i,j-1}^n)\\ &+ \frac{\delta t}{8h_y}(V_{i,j+1}^{n-1} + V_{i,j-1}^{n-1})(U_{i,j+1}^{n-1} - U_{i,j-1}^{n-1}) + \delta t f_{i,j}^{n-1}\biggr).\end{aligned} \tag{30}$$

During the second half-time step and for each i, solve

$$\begin{aligned}(\alpha_y^2 - 2\beta_y^2)U_{i,2j}^{n+1} - \beta_y^2(U_{i,2j-2}^{n+1} + U_{i,2j+2}^{n+1}) &= \alpha_y G_{i,2j}^{n+1/2,n,n-1} + \beta_y(G_{i,2j-1}^{n+1/2,n,n-1} + G_{i,2j+1}^{n+1/2,n,n-1}),\\ U_{i,2j+1}^{n+1} &= \frac{1}{\alpha_y}G_{i,2j+1}^{n+1/2,n,n-1} + \frac{\beta_y}{\alpha_y}(U_{i,2j}^{n+1} + U_{i,2j+2}^{n+1}),\\ U_{i,2j-1}^{n+1} &= \frac{1}{\alpha_y}G_{i,2j-1}^{n+1/2,n,n-1} + \frac{\beta_y}{\alpha_y}(U_{i,2j}^{n+1} + U_{i,2j-2}^{n+1}),\end{aligned} \tag{31}$$

with

$$\alpha_y = 1 + \frac{\delta t}{Re \cdot h_y^2}, \qquad \beta_y = \frac{\delta t}{2Re \cdot h_y^2}$$

and right members of the kind

$$G_{i,j}^{n+1/2,n,n-1} = U_{i,j}^{n+1/2} - \frac{\delta t}{2Re\cdot h_x^2}(2U_{i,j}^{n+1/2} - U_{i-1,j}^{n+1/2} - U_{i+1,j}^{n+1/2})$$
$$-(1-\alpha)\left(\frac{3\delta t}{8h_x}(U_{i+1,j}^{n} + U_{i-1,j}^{n})(U_{i+1,j}^{n} - U_{i-1,j}^{n})\right.$$
$$+\frac{\delta t}{8h_x}(U_{i+1,j}^{n-1} + U_{i-1,j}^{n-1})(U_{i+1,j}^{n-1} - U_{i-1,j}^{n-1}) - \frac{3\delta t}{8h_y}(V_{i,j+1}^{n} + V_{i,j-1}^{n})(U_{i,j+1}^{n} - U_{i,j-1}^{n})$$
$$\left.+\frac{\delta t}{8h_y}(V_{i,j+1}^{n-1} + V_{i,j-1}^{n-1})(U_{i,j+1}^{n-1} - U_{i,j-1}^{n-1}) + \delta t f_{i,j}^{n-1}\right). \tag{32}$$

Remark. To obtain a second-order-accurate time scheme, we have to choose $\alpha = 0{\cdot}5$.

We can also define multilevel schemes using Incremental Unknowns. During the first half-time step, formulae linking Y, Z and U for each j are

$$\begin{aligned} Y_{2i,j}^{k+1/2} &= U_{2i,j}^{k+1/2}, \\ Z_{2i+1,j}^{k+1/2} &= U_{2i+1,j}^{k+1/2} - \tfrac{1}{2}(U_{2i,j}^{k+1/2} + U_{2i+2,j}^{k+1/2}). \end{aligned} \tag{33}$$

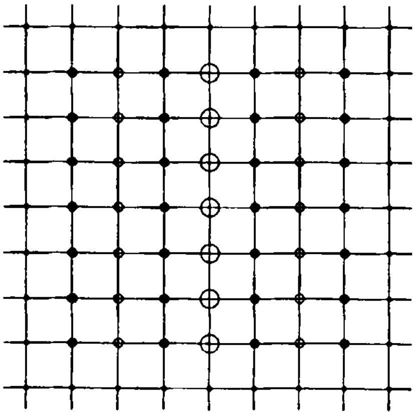
During the second half-time step, these formulae for each i become

$$\begin{aligned} Y_{i,2j}^{k+1} &= U_{i,2j}^{k+1}, \\ Z_{i,2j+1}^{k+1} &= U_{i,2j+1}^{k+1} - \tfrac{1}{2}(U_{i,2j}^{k+1} - U_{i,2j+2}^{k+1}). \end{aligned} \tag{34}$$

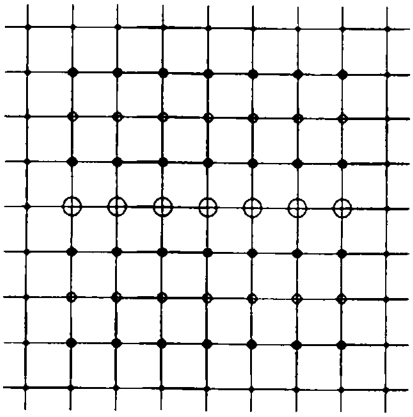
The discretization using Incremental Unknowns is obtained by applying relations (33) and (34) to systems (29) and (31).

Let us show the pattern of three grid meshes in Incremental Unknowns: for the first half-step we have

Boundaries : ·
Coarse grid : ○
First level of Incremental Unknowns : ∘
Second level of Incremental Unknowns (fine grid) : ●



and for the second half-step we have



4.3. Numerical results

All numerical results are obtained by the use of a steady solution already suggested by Jain and Holla.[21] The initial conditions are

$$\begin{aligned} u(t=0,x,y) &= \sin(\pi x)+\cos(\pi y) \quad \text{in } [0,0{\cdot}5]^2, \\ v(t=0,x,y) &= x+y \quad \text{in } [0,0{\cdot}5]^2. \end{aligned} \tag{35}$$

The boundary conditions follow from the initial conditions.

This solution is interesting because of the development of gradients along boundaries (see Figures 12 and 13). We can control the importance of these gradients thanks to the Re value. Consequently, this solution seems to be a good test to compare our schemes. Comparisons are made with $Re=100$ and two mesh sizes: 513^2 nodes and 1025^2 nodes. The latter mesh size is comparatively fine and permits higher values of Re. As far as the CPU time is concerned, $t=0{\cdot}01$. The numerical stability condition allows us time steps $\delta t=0{\cdot}0005$ and $0{\cdot}00025$ (the use of our multilevel schemes does not allow stability improvement and therefore all schemes have the same time step). We have chosen to solve equations with the CRAY YMP/EL of the CRI (Orsay University) and with the CRAY C98 of the IDRIS CNRS centre respectively.

Table III presents a comparison of execution times for various schemes. If we are only interested in CPU time (otherwise the scheme (28) may be a good choice, thanks to a previous remark), we clearly observe in Table III that the best choice is the multilevel ADI scheme.

Remark. In tables, when the system solved is called ADI, it is the classical version on one grid.

If we denote by $\kappa(M)$ the condition number of the matrix of (26) ($M=I+(\delta t/2Re)A$), then by the following equation (where $\mu_{\max}$ and $\mu_{\min}$ are the extreme eigenvalues of the matrix A) it is easy to verify that the matrix M is well conditioned for high values of Re:

$$\kappa(M)=\left(1+\frac{\delta t\mu_{\max}}{2Re}\right)\bigg/\left(1+\frac{\delta t\mu_{\min}}{2Re}\right). \tag{36}$$

In Table IV we notice (thanks to a numerical determination of condition numbers) that the matrices of system (27) are not well conditioned when the number of grids increases, except for low values of Re. Noting that $S^{\mathrm{T}}[I+(\delta t/2Re)A]S=S^TS+(\delta t/2Re)S^TAS$ and the fact that $\delta t/2Re$ is generally very small (of the order of 10^{-6} in our examples), $\kappa(S^{\mathrm{T}}[I+(\delta t/2Re)A]S)$ is very close to $\kappa(S^{\mathrm{T}}S)$. We observe this result in Table V (by comparison with Table IV, $Re=100$).

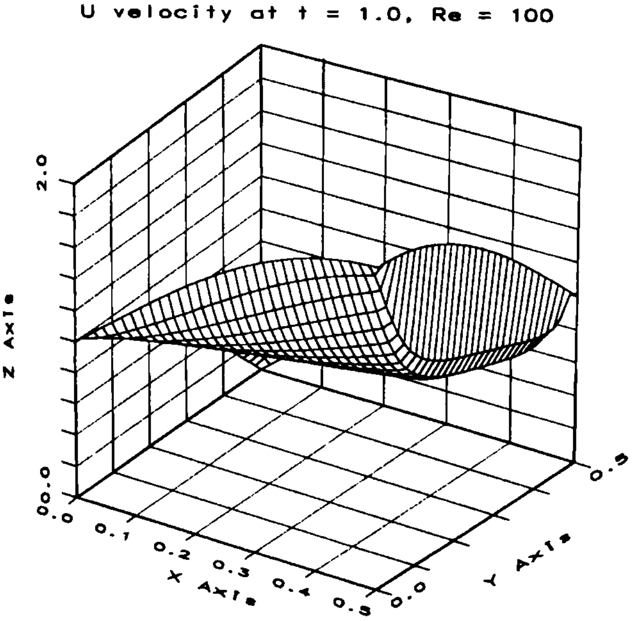


Figure 12. First component of velocity; $t = 1{\cdot}0$, 513^2 nodes, $Re = 100$

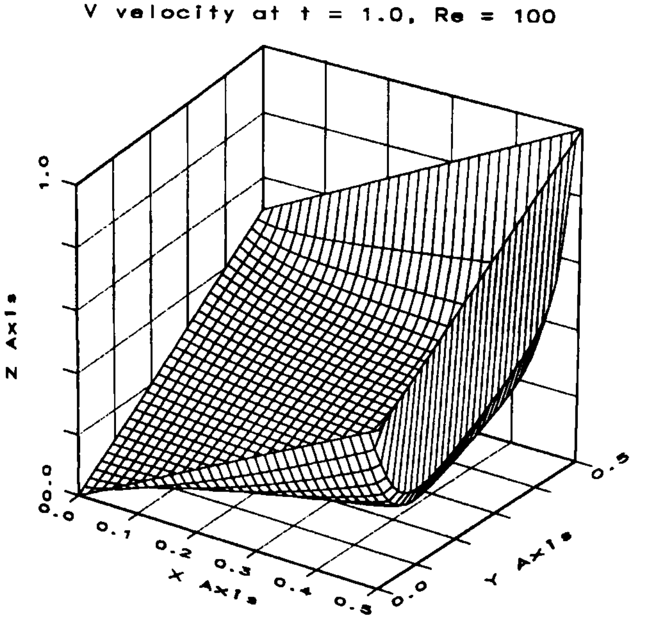


Figure 13. Second component of velocity; $t = 1{\cdot}0$, 513^2 nodes, $Re = 100$

Table III. Comparison of CPU times: 513^2 and 1025^2 nodes, $Re=100$

513^2 nodes, 200 time steps						
Number of grids Type of addressing System solved	1 Nodal (26)	2 Multilevel (27)	1 Multilevel (28)	9 Multilevel (28)	1 Nodal ADI	9 Multilevel (29)–(31)
CPU time (s)	1450	2175	4183	4774	1953	535
1025^2 nodes, 400 time steps						
Number of grids Type of addressing System solved	1 Nodal (26)	2 Multilevel (27)	1 Multilevel (28)	10 Multilevel (28)	1 Nodal ADI	10 Multilevel (29)–(31)
CPU time (s)	1116	1238	2719	3153	1270	493

To complete our study and explain more carefully the results in Table III, we also desire to make a qualitative analysis of CPU times of each scheme. To do this, we define a relative cost per time step,

$$\begin{aligned}\mathscr{C}_t = \frac{1}{\mathscr{C}_0} &\times \text{(number of iterations of gradient method per time step)}\\ &\times \text{(number of matrixvector products per iteration)}\\ &\times \text{(relative cost of a matrixvector product)},\end{aligned} \tag{37}$$

where $\mathscr{C}_0$ is the cost of resolution of system (26). All results summarized in Table VI.

Remark. (i) The Bi-CGSTAB gradient method (developed by Van der Vorst[22]) used to solve the non-symmetric system (28) prescribes two matrix–vector products per iteration, while the conjugate gradient (CG) method prescribes one matrix–vector product per iteration. (ii) By our choice of number

Table IV. Variations in condition number of matrices of (27): $\delta t=0{\cdot}004$, 257^3 nodes

Number of grids	1	2	3	4	5	6	7	8
$\kappa\left(S^{\mathrm{T}}\left(I+\frac{\delta t}{2Re}A\right)S\right)$ with $Re=1$	774	205	56	21	25	38	148	358
$\kappa\left(S^{\mathrm{T}}\left(I+\frac{\delta t}{2Re}A\right)S\right)$ with $Re=10$	100	23	12	19	67	291	967	2546
$\kappa\left(S^{\mathrm{T}}\left(I+\frac{\delta t}{2Re}A\right)S\right)$ with $Re=100$	11	7	19	96	408	1525	4604	7253

Table V. Condition number of $S^{\mathrm{T}}S$: 257^2 nodes

Numer of grids	1	2	3	4	5	6	7	8
$\kappa(S^{\mathrm{T}}S)$	1	22	109	458	1806	5356	8209	9323

Table VI. Estimations of CPU time

Number of grids	1	2	2	9	1
Type of addressing	Nodal	Multilevel	Multilevel	Multilevel	Nodal
System solved	(26)	(27)	(28)	(28)	ADI
Gradient method	CG	CG	Bi-CGSTAB	Bi-CGSTAB	CG
Number of gradient iterations	53	32	31	29	41 + 34
Number of matrix–vector products per iteration	1	1	2	2	1
Relative cost of a matrix–vector product	1	3	3	3	1
Relative cost $\mathscr{C}_t$ per time step	1	1·8	3·5	3·3	1·4

of nodes the multilevel ADI scheme becomes a direct solver at each time step: there is no use of a gradient method and then we are not able to estimate the relative cost $\mathscr{C}_t$.

If we compare the estimations in Table VI with CPU times actually obtained (see Table III), we find that all results are very close, so we can suppose that all these schemes have a similar computational optimization. This aspect will be developed in the next subsection. Nevertheless, when Incremental Unknowns are used, estimated times are always too large. There exists a simple explanation: the relative cost of matrix–vector products in Incremental Unknowns is overestimated. In fact, this cost is between 1·3 and 2·8 depending on the number of grids.

By referring to Table VI, we also notice that system (27) with two grids is better conditioned than system (26) (nodal basis). This is also true for system (28), but, in addition, this last choice of discretization leads to new results: any number of grids (any level of discretization) improves the condition number of the matrix to invert (see the number of gradient iterations).

Figures 14 and 15 exhibit relatively high values of Incremental Unknowns near boundaries. This result is of course an extension of our one-dimensional observations.

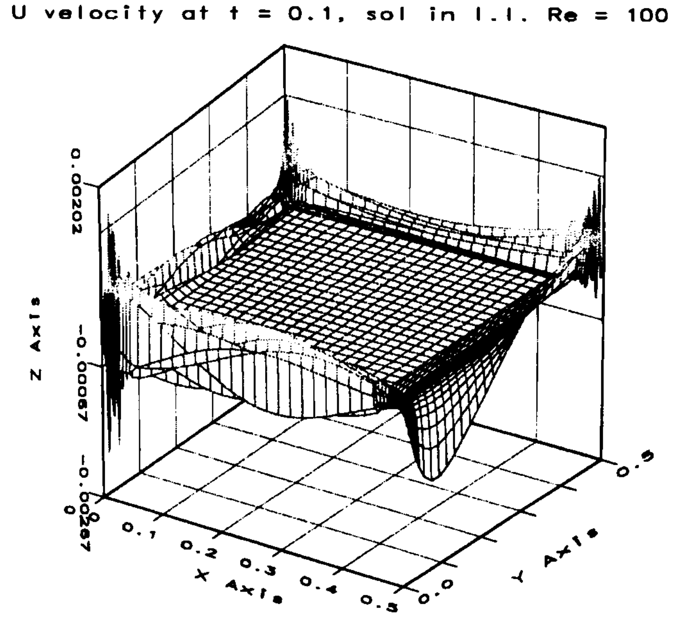


Figure 14. First component in Incremental Unknowns of velocity (first level); $t = 0{\cdot}1$, 513^2 nodes, $Re = 100$

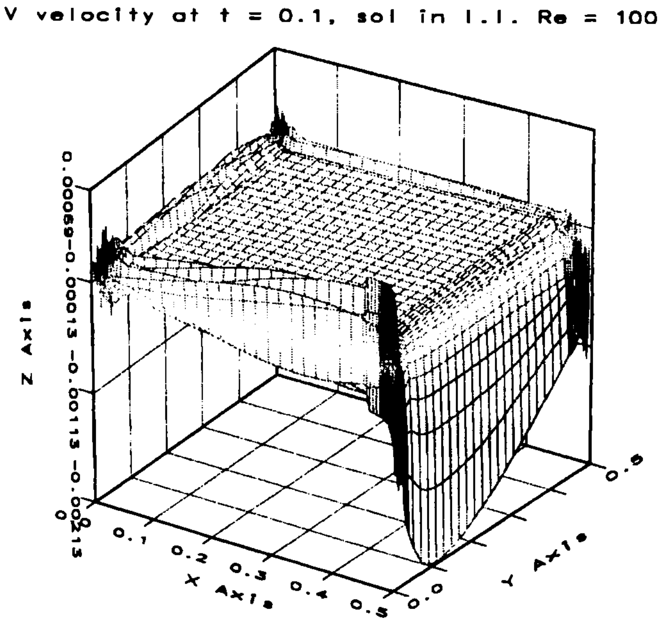


Figure 15. Second component in Incremental Unknowns of velocity (first level); $t=0{\cdot}1$, 513^2 nodes, $Re=100$

4.4. Computational aspects

This subsection is devoted to the study of vectorial and optimization aspects of our schemes, especially on CRAY computers (the CRAY YMP/EL and the CRAY C98) with the High Performance Monitor (HPM) CRAY tool.

By referring to Table VII, we notice that the Incremental Unknowns addressing (except for the multilevel ADI scheme) only damages the vectorial performance a little (even with a maximal number of grids). The lost of vectorial performance is the result of indirect references and recursive parts in programs.

As is made clear by Table VIII, the multilevel ADI scheme is the fastest method in scalar mode: 10^8 instructions compared with $1{\cdot}5 \times 10^9$ instructions (vectorial CPU times are not in this ratio because of

Table VII. Mflops performance of several schemes

Number of grids Type of addressing System solved	1 Nodal (26)	2 Multilevel (27)	9 Multilevel (28)	1 Nodal ADI	9 Multilevel (29)–(31)
Mflops	575	395	349	573	132

Table VIII. Number of instructions for several schemes

Number of grids Type of addressing System solved	1 Nodal (26)	2 Multilevel (27)	9 Multilevel (28)	1 Nodal ADI	9 Multilevel (29)–(31)
Number of instructions ($\times 10^9$)	1·5	0·9	2·6	1·9	0·1

Table IX. Average conflicts per reference for several schemes

Number of grids	1	2	9	1	9
Type of addressing	Nodal	Multilevel	Multilevel	Nodal	Multilevel
System solved	(26)	(27)	(28)	ADI	(29)–(31)
Average conflicts/reference	0·09	0·23	0·37	0·10	1·67

the difference in Mflops performances). We also see that system (27) on two grids presents fewer instructions than system (26).

Table IX exhibits the average conflicts per reference. If we detail reasons for the conflicts, we see that the main one comes from memory bank conflicts and is a consequence of multilevel addressing; conflicts increase with the number of grids.

By its nature the multilevel ADI scheme is designed for parallelization (an equation to solve by row or by column of the mesh). At each half-time step we solve several monodimensional problems each involving a small number of nodes and with only recursive references (see (29) and (31)). Thus operations are not really designed for vectorization. This is an explanation of the poor result for the multilevel ADI scheme in term of conflicts per reference.

5. CONCLUDING REMARKS

In this work, several multilevel schemes are proposed for solving unsteady equations. The one-dimensional case allows us to define schemes restricting to a maximum the size of linear systems to solve. In these schemes the Incremental Unknowns concept appears to be a new manner to study solutions. In the two-dimensional case we extend our schemes and propose a fast 'direct' solver of the Burgers equations, thanks to an ADI formulation. This last algorithm is especially designed for parallelization. We also propose a non-symmetric formulation of matrices which define a hierarchy of the non-linear terms and well condition the systems to solve.

In short, we have here defined some multilevel formulations of a new kind, allowing fast (vectorial and parallel) numerical resolutions of unsteady equations in one and two dimensions. The hierarchy of equations possesses numerous advantages: a simplified study of physical phenomena (boundary layers are on fine grids) and experimental setting of new numerical analysis concepts (scale distinction, linear and non-linear coupling of equations on various grids).

ACKNOWLEDGEMENTS

The author is pleased to acknowledge Professor R. Temam for his suggestions and interest, and the CRI and IDRIS centres where computational tests have been performed.

REFERENCES

1. C. Foias, G. Sell and R. Temam, 'Inertial manifolds for nonlinear evolutionary equations', *J. Diff. Eq.*, **73**, 309–353 (1988).
2. C. Foias, O. Manley and R. Temam, *C.R. Acad. Sci. Paris, Sér. I*, **305**, 497–500 (1987); *Math. Model. Numer. Anal.*, **22**, 93–114 (1988).
3. M. S. Jolly, I. G. Kevrekidis and E. S. Titi, 'Approximate inertial manifolds for the Kuramoto–Sivashinsky equation: analysis and computations', *Physica D*, **44**, 38–60 (1990).
4. A. Debussche, T. Dubois and R. Temam, 'The nonlinear Galerkin method: a multiscale method appied to the simulation of homogeneous turbulent flows', *Theor. Comput. Fluid Dyn.*, (1994).
5. M. Chen and R. Temam, 'Incremental Unknowns for solving partial differential equations', *Numer. Math.*, **59**, 255–271 (1991).

6. M. Chen and R. Temam, 'Incremental Unknowns in finite differences: condition number of the matrix', *SIAM J. Matrix Anal. Appl.*, **14**, 432–455 (1993).
7. J. P. Chehab and R. Temam, 'Incremental Unknowns for solving nonlinear eigenvalue problems. New multiresolution methods', *Numer. Methods Partial Diff. Eq.*, (1995).
8. J. P. Chehab, 'A nonlinear adaptive multiresolution method in finite differences with Incremental Unknowns', *M2AN*, (1995).
9. P. Poullet, *Thesis*, University of Orsay, 1996.
10. O. Goyon, 'Résolution numérique de problèmes stationnaires et évolutifs non linéaires par la méthode des Inconnues Incrémentales', *Thesis*, University of Orsay, 1994.
11. O. Goyon, 'High Reynolds number solution of Navier–Stokes equations using Incremental Unknowns', in preparation.
12. J. M. Burgers, *The Nonlinear Diffusion Equation*, Reidel, Dordrecht, 1974.
13. C. A. J. Fletcher, 'Burgers equation: a model for all reasons', in J. Noye (ed.), *Numerical Solutions of Partial Differential Equations*, North-Holland, Amsterdam, 1982.
14. J. D. Cole, 'On a quasi-linear parabolic equation occurring in aerodynamics', *Q. Appl. Math.*, **9**, 225–236 (1951).
15. E. Hopf, 'The partial differential equation $u_t + uu_x = \mu u_{xx}$', *Commun. Pure Appl. Math.*, **3**, 201–230 (1950).
16. E. Benton and G. W. Platzman, 'A table of solutions of the one- dimensional Burgers equation', *Q. Appl. Math.*, **30**, 195–212 (1972).
17. C. A. J. Fletcher, 'Generating exact solutions of the two-dimensional Burgers equations', *Int. j. numer. methods fluids*, **3**, 213–216 (1983).
18. D. W. Peaceman and H. H. Rachford Jr., 'The numerical solution of parabolic and elliptic differential equations', *J. Soc. Ind. Appl. Math.*, **3**, 29–41 (1955).
19. J. Douglas Jr., 'On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods', *J. Soc. Ind. Appl. Math.*, **3**, 42–65 (1955).
20. J. Douglas Jr. and J. E. Gunn, 'A general formulation of alternating direction methods', *Numer. Math.*, **6**, 428–453 (1964).
21. P. C. Jain and D. N. Holla, 'Numerical solutions of coupled Burgers equations', *Int. J. Nonlin. Mech.*, **13**, 213–222 (1978).
22. H. A. Van der Vorst, 'Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **13**, 631–644 (1992).