Mecánica Computacional Vol. XXI, pp. 347–357
S. R. Idelsohn, V. E. Sonzogni and A. Cardona (Eds.)
Santa Fe–Paraná, Argentina, October 2002

AN EFFICIENT THREE LEVEL ALGORITHM FOR THE NUMERICAL SOLUTION OF THE NAVIER-STOKES EQUATIONS

M. Behnia[1], G. de Vahl Davis[1], U. Groisman[2] and M. Wolfshtein[3]

[1] University of New South Wales, Sydney, Australia

[2] Preuniversitario Ciudad de San Felipe, Montevideo, Uruguay
urig@sanfelipe.edu.uy

[3] Faculty of Aerospace Engineering, Technion, Israel Institute of Technology, Haifa, Israel

Key words: Iterative methods, Navier-Stokes Equations, Pseudo-Transient Techniques, Non-Linear Equations.

Abstract. *A three level algorithm (in time) for the solution of parabolic flow problems is discussed. The algorithm is used to solve time independent flow problems. It is demonstrated that this algorithm can generate good results in a small number of time steps and thus reduce the computer time required to solve a given problem. The algorithm is simple and its incorporation in existing codes is straight forward.*

1 INTRODUCTION

Rapid developments in computer technology and numerical methods allow increasingly more complicated problems to be solved numerically. Consequently, Computational Fluid Dynamics (CFD) is becoming more and more attractive as a design tool for problems in which fluid flow is important. This trend has been very significant in various applications, like the design of airplanes, cars and ships, weather forecasting, and various other problems of practical and scientific interest. However, the process of utilizing the new possibilities often requires rewriting of substantial parts of computer codes, as modern numerical technology may be very different from older technology. The fact that the amount of man years invested in existing codes is immense, resulted in a relatively slow pace of penetration of CFD to the design process. Consequently the price performance of computer codes available to the industry is often too bad to allow parametric studies which form an important ingredient of design processes.

In view of this situation it is useful to seek simple improvements to existing algorithms which can improve the computer performance. In this paper we explore such a possibility. The problem in question is that of time independent Navier-Stokes equations. The method to be discussed is a three level time marching algorithm that facilitates an efficient solution of parabolic equations. This algorithm is easy to implement, and can be used to modify existing codes with relatively little effort. Thus it allows an efficient solution but it does not require rewriting of the code.

This three level algorithm was proposed by Israeli and Livne[1] and later applied by Behnia et al.[2] to the problem of free convection in a close cavity. The results of Behnia et al. indicated that a saving of about 30 to 50 percent in computer time is possible with this three level scheme. Yet Behnia et al. did not perform a careful analysis of the various options and it was felt that some numerical experiments are required to explore the potential of the method. The present paper describes such numerical experiments. Indeed, it was found that much higher efficiencies are possible if the parameters of the scheme are carefully selected. In this paper we describe these numerical experiments and draw some conclusions on the usefulness of the three level scheme

2 THE TEST PROBLEM

In order to perform the numerical experiments to determine the best parameter for the numerical scheme we choose to apply the method to the one dimensional Burgers' Equation. It is written in standard form as

$$\frac{\partial u}{\partial t}+\frac{\partial\left(u^2/2\right)}{\partial x}=\nu\frac{\partial^2 u}{\partial x^2} \qquad 1$$

or in quasi-linear form as

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\nu\frac{\partial^2 u}{\partial x^2} \qquad 2$$

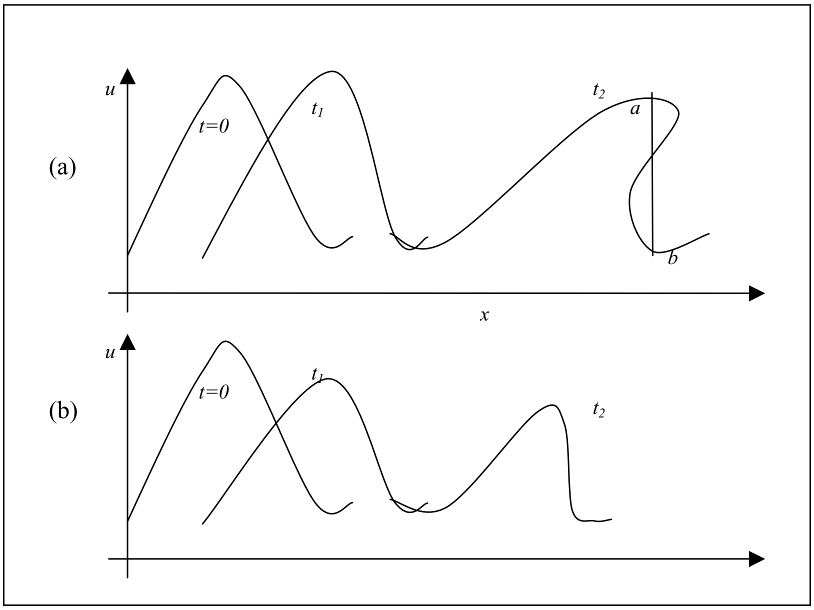
Burgers' equation has the same form of non-linearity that appears in the momentum equations.

2 Physical Behavior

If the ‘viscous’ term is dropped from (2) the result is the inviscid Burgers’ equation.

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=0 \qquad 3$$

The non-linearity in (3) allows discontinuous solutions to develop. The way that this can occur is illustrated schematically fig.1a. A wave is convecting from left to right and solutions



for successive times $t=0$, t_1, t_2 are indicated. Points on the wave with larger values of u convect faster and consequently overtake parts of the wave convecting with smaller values of u. For (3) to have a unique solution (and a physical sensible result) it is necessary to postulate

Fig. 1(a) Formation of multivalued solution of the inviscid Burgers’ equation
(b) Evolution of the solution of Burgers’ equation

a shock (*ab* in fig.1) across which u changes discontinuously.

The comparable wave development for the ‘viscous’ Burgers’ equation (1) is shown in fig.1b. The effect of the viscous term $\nu\,\partial^2 u/\partial x^2$ is twofold. First, it reduces the amplitude of the wave for increasing t. Second, it prevents multivalued solutions from developing.

These features makes Burgers’ equation a very suitable model for testing computational algorithms. This role as a test-bed for computational algorithms is facilitated by the Cole-Hopf[3] transformation which allows exact solutions of Burgers’ equation to be obtained for many combinations of initial and boundary conditions.

When a solution is obtained on a grid of spacing Δx the smallest wavelength that can be resolved is $2\Delta x$. The energy associated with wavelengths shorter that $2\Delta x$ reappears associated with long wavelengths. This phenomenon is called aliasing. Unfortunately the aliased shortwave contribution to the solution distort the true longwave solution and may even cause instability where very long time integrations are made, as in computational weather forecasting.

It may be recalled that if any dissipation, physical or computational, is present it attenuates the amplitude of the short wavelengths very significantly; in this case the errors introduced by aliasing are minimal. This is the situation for most engineering flow problems, which posses physical dissipation, typically.

3 THE NUMERICAL SCHEME

Consider a parabolic partial differential equation of the form

$$\frac{\partial \Phi}{\partial t} = L\Phi + S \qquad 4$$

where *t* is the time or a spatial marching direction, *L* is a elliptic differential operator and *S* is the source term. The usual finite difference substitution for (4) without a source is a two-point formula based on upstream and downstream values of the form

$$\alpha.\Phi^{n+1} + \beta.\Phi^{n} = \Delta t\left(\delta L\left(\Phi^{n+1}\right) + \varepsilon L\left(\Phi^{n}\right)\right) \qquad 5$$

where Δt is the time step size and *n* represent the time and where $\alpha, \beta, \gamma, \delta, \varepsilon$ are dimensionless numerical constants to be chosen. In two point schemes we usually use $\alpha = -\beta = 1$. When $\delta = \varepsilon = 1/2$ we get the second order Crank-Nicolson Scheme; when $\delta = 1$ and $\varepsilon = 0$ the first order implicit scheme is obtained; and $\delta = 0$ and $\varepsilon = 1$ leads to the first order explicit scheme.

It is easy to show that the Crank-Nicolson scheme is the only two point second order scheme and therefore it is impossible to devise a second order scheme which is more stable than the Crank-Nicolson scheme. Enhancement of stability is possible *only* if additional levels are used. The simplest possibility is the addition of a third level by adding Φ^{n-1} and $L\Phi^{n-1}$ to the left and right sides of (5) respectively. In this work we choose not to add $L\Phi^{n-1}$ because the enhancement of stability is obtained even when this term is not used. Therefore (5) is replaced by

$$\alpha.\Phi^{n+1} + \beta.\Phi^{n} + \gamma\Phi^{n-1} = \Delta t\left(\delta L\left(\Phi^{n+1}\right) + \varepsilon L\left(\Phi^{n}\right)\right) \qquad 6$$

To examine consistency of (6) we substitute a Taylor series expansion of each term around the *n*th step, giving the following equation

$$\alpha\left(\Phi+\Delta t\Phi'+\frac{\Delta t^2}{2}\Phi''+\ldots\right)+\beta\Phi+\gamma\left(\Phi-\Delta t\Phi'+\frac{\Delta t^2}{2}\Phi''-\ldots\right)$$
$$=\Delta t\delta L\left(\Phi+\Delta t\Phi'+\frac{\Delta t^2}{2}\Phi''+\ldots\right)+\Delta t\varepsilon L\Phi \qquad 7$$

where a prime denotes partial differentiation with respect to time. Collection of terms into equal orders of Δt shows that

$$\alpha+\beta+\gamma=0 \qquad 8$$

If

$$\alpha-\gamma=\delta+\varepsilon \qquad 9$$

the truncation error becomes $O(\Delta t^2)$.

For convenience and without loss of generality we may choose $\alpha=1$. It follows that (7) – (9) can be solved for β,ε and δ in terms of γ:

$$\beta=-1-\gamma \qquad \varepsilon=(1-3\gamma)/2 \qquad \delta=(1+\gamma)/2 \qquad 10$$

Substitution of (10) into (5), with the definition

$$1/(1-\gamma)=\omega \qquad 11$$

yields

$$\omega\left(\Phi^{n+1}-\Phi^{n}\right)+(1-\omega)\left(\Phi^{n}-\Phi^{n-1}\right)=\Delta t(\omega-1/2)L\Phi^{n+1}+\Delta t(3/2-\omega)L\Phi^{n} \qquad 12$$

where ω is a free parameter which may be so chosen as to enhance stability.

It appears that the larger the time step, the faster the convergence. However, stability of the numerical procedure often restricts the time step. Moreover, the truncation error resulting from the discretization grow with the time step. Therefore it is not beneficial to use long time steps even if the stability problem can be overcome, unless the truncation error can be reduced. Thus the problem is to select such a value of ω that allows maximum stability while retaining a second order truncation error in time. This can be done analytically for given operators L provided that the operator is linear. However the Navier-Stokes operator is not linear, and therefore we shall try to explore the best value of ω using numerical experimentation. It is obvious that such an approach is restricted, as it does not produce general prescription. However, it can give us some ideas of the potential of the method.

It should be born in mind that the non-linearity of the Navier-Stokes operator means that the discretization requires certain means if second order accuracy in time is only desired (e.g. iterations within the time step). This was not done here as we are interested in the steady state solution. Still linear second order accuracy was used, as it may produce smaller truncation errors than the first order discretization.

4 NUMERICAL EXPERIMENTS

We do the following procedure in order to study the Super-Stable (SS) algorithm: we start solving Burgers' equation giving to ω such values as to obtain some classic schemes like Richardson explicit, Crank-Nicolson, Leap-Frog and found the Δt which gives a minimum iteration number. This happens just before the different schemes became unstable. After that start giving ω grater values, and seek for minimum iteration number changing Δt as shown in figures 1 and 2. We did so for several Re numbers and mesh sizes.

4 Results

Results are shown in the following figures. Firstly we found an excellent agreement between the exact and computed solutions, even for very high Re number. We check for agreement of the solutions from Re=0.01, to Re=1e+5, being aware that there's no physical interest in the solutions obtained for Re>1000. On the other hand the agreement we obtained between the two solutions when we arrived to Re>1000 talks about the robustness of the SS algorithm.

Our aim was to find a SS parameter that minimize the number of iterations to convergence, so we change ω and Δt in order to achieve this. We found two different behaviors of the solution. When we gave ω small values and start growing in Δt, the number of iterations decrease until the scheme became unstable. On the other hand when we gave ω higher values we found an asymptotic behavior. This property of the scheme is summarized on table I.

We also found that the iteration number to convergence is independent of the mesh size. This fact is very interesting because this property is found in much more complicated and cumbersome numerical methods.

5 CONCLUSIONS

The major conclusions of this paper are:

a.- The optimal number of iterations of the classical methods appears to be on a continuation of the SS results.

b.- The classical methods are inferior to the SS ones not only in the number of iterations but also in the accuracy.

c.- The dip in the number of iterations is rather narrow. Therefore it is necessary to have a pretty good estimate of the optimal ω and Δt so it is safer to use such omegas where an asymptotic constant value of Δt exists.

d.- The location of the smallest number of iterations and the highest accuracy are very close.

e.- A good estimate of the optimal omega is required, probably by using stability analysis.

6 REFERENCES

[1] M. Israeli and A. Livne, Development of a super stable scheme of second order in the flow

direction and fourth order in the cross flow direction, for boundary layer equations, Aerodynamic Lab. Report Nº 160-073, Technion, Israel, 1990

[2] M. Behnia, G. de Vahl Davis and M. Wolfshtein, A stable fast marching scheme for computational fluid mechanics, International Journal for Numerical Methods in Fluids, Vol. 10, pp. 607-621, 1991

[3] Cole, J. D., Q. Applied Mathematics 9, pp. 225-236, 1951

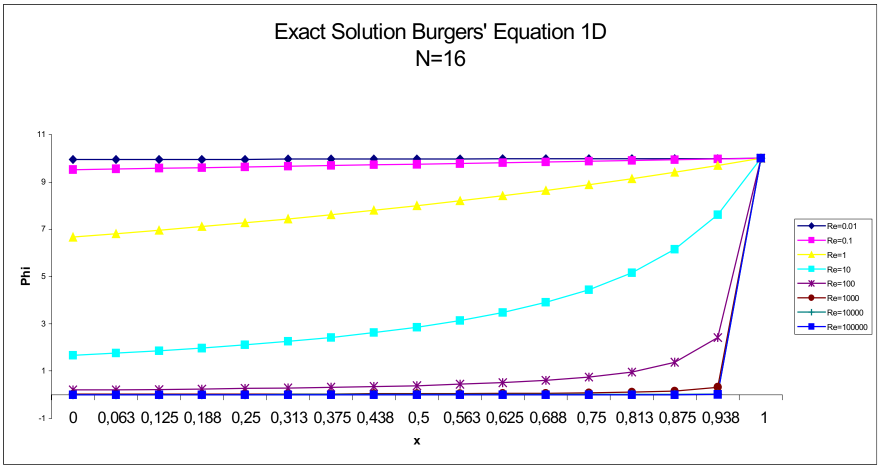


Fig. 2 Exact solution of Burgers' equation with Dirichlet boundary conditions for different Re number

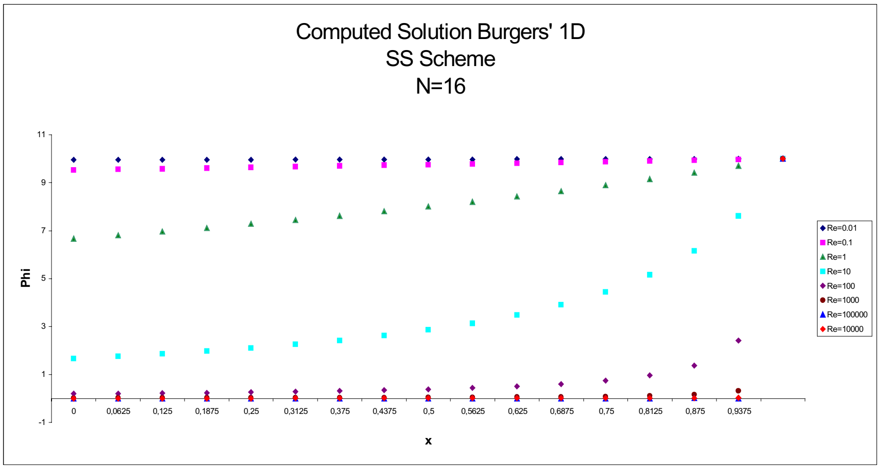


Fig. 3 Computed solution of Burgers' equation using super-stable algorithm for various Re number

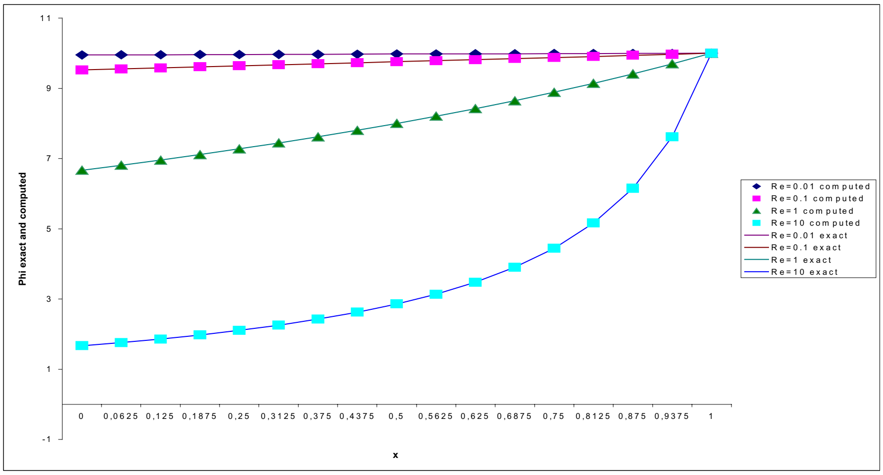


Fig. 4 Agreement between computed and exact solution for small Re numbers

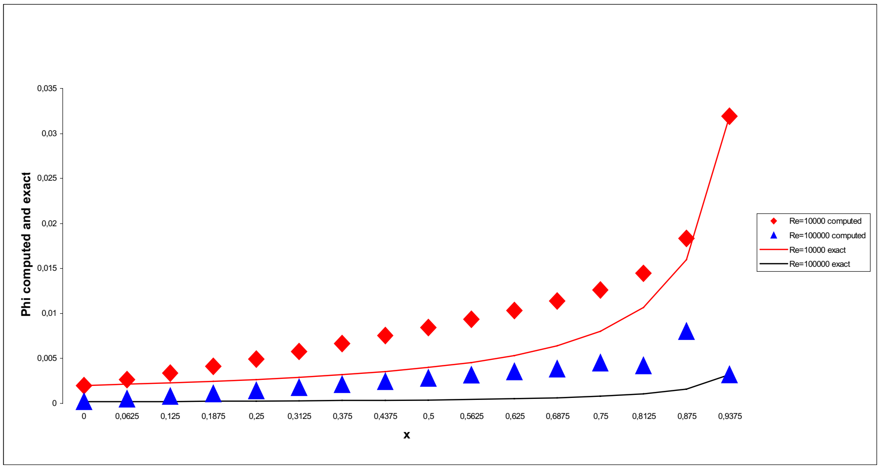


Fig. 5 Agreement between computed and exact solution for small Re numbers

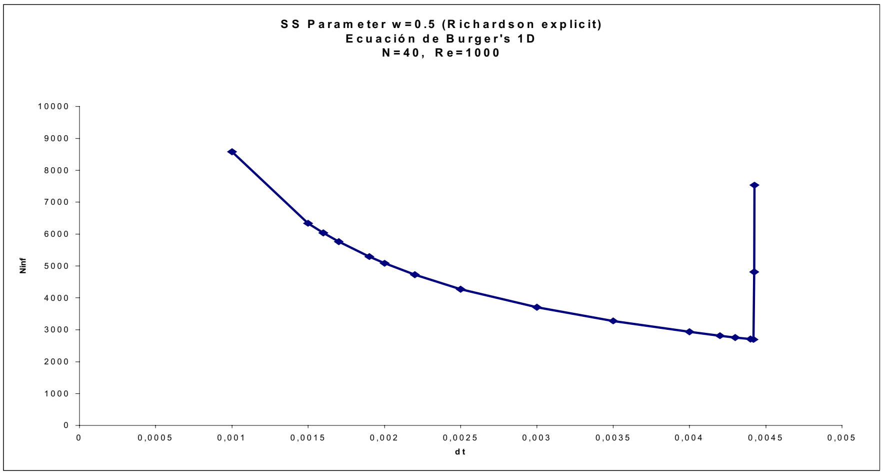


Fig. 6 Small omega behavior of the super-stable scheme

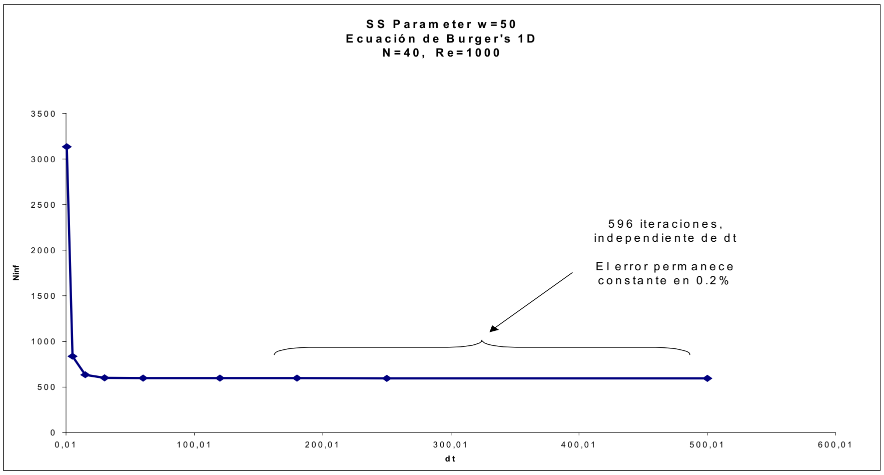


Fig. 7 High omega behavior of the super-stable scheme

iters	dt	dy	Re	omega	difer(%)			
2370	0,012	0,025	1000		1,1165	1st order fully implicit		
2706	0,0044	0,025	1000	0,5	1,52E+00	Richardson		
2738	0,01	0,025	1000	1	1,34E+00	Crank-Nicolson		
2423	0,0152	0,025	1000	1,309	1,15E+00	Behnia, de Vahl Davis, Wolfshtein		
2290	0,0187	0,025	1000	1,5	1,08E+00	Leap-Frog		
1645	0,0465	0,025	1000	2,5	7,24E-01			
1113	0,105	0,025	1000	3,5	4,51E-01			
633	0,271	0,025	1000	4,5	2,28E-01			
272	1	0,025	1000	5,5	8,15E-02			
100	8,2	0,025	1000	6,5	2,25E-02			
91	12,5	0,025	1000	6,6	2,04E-02	minimum error		
89	15,2	0,025	1000	6,7	2,05E-02			
86	21,5	0,025	1000	6,8	2,22E-02			
85	37	0,025	1000	6,9	2,59E-02	minimum iterations		
85	40	0,025	1000	6,91	2,59E-02	minimum iterations		
87	40	0,025	1000	7	2,34E-02			
93	40	0,025	1000	7,5	2,85E-02			
120	40	0,025	1000	9,5	3,48E-02			
106	40	0,025	1000	8,5	3,41E-02			
189	40	0,025	1000	15	6,51E-02			
596	40	0,025	1000	50	2,35E-01			
1140	40	0,025	1000	100	0,4773			

Table I Numerical experiments in order to find optimal super-stable parameter omega

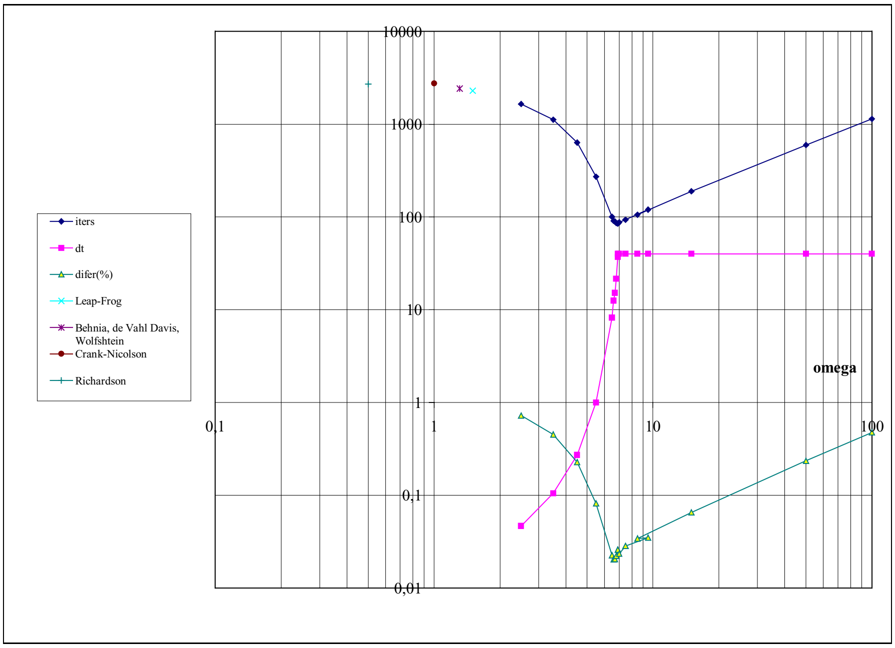


Fig. 8 Number of iterations, time step and error between calculated and exact solution of Burgers' equation as function of the super-stable parameter omega

iters	dt	dy	Re	omega	difer(%)
3624	0,05	0,0625	1000	7	1,89E+00
570	0,5	0,0625	1000	7	1,99E-01
2075	0,1	0,0625	1000	7	9,54E-01
574	0,5	0,025	1000	7	1,99E-01
130	5	0,025	1000	7	3,02E-02
94	15	0,025	1000	7	2,00E-02
87	45	0,025	1000	7	2,31E-02
87	45	0,0156	1000	7	2,36E-02
87	90	0,0156	1000	7	2,29E-02
86	120	0,0156	1000	7	2,70E-02
94	15	0,0156	1000	7	2,00E-02
130	5	0,0156	1000	7	3,03E-02
130	5	0,0078	1000	7	3,04E-02
94	15	0,0078	1000	7	2,00E-02
87	45	0,0078	1000	7	2,40E-02
87	45	0,0052	1000	7	2,41E-02
94	15	0,0052	1000	7	2,00E-02

Table II Independency of number of iterations from mesh size