NEWTON–RELAXATION SCHEMES FOR NONLINEAR FLUID FLOW EQUATIONS

I. Tuwegiar (Tripoli, Libya)

Abstract. The rapid solution of the nonlinear algebraic equations resulting from the discretization of convection-diffusion equations is of interest in the field of computational fluid dynamics. In this paper local linearization schemes based on Newton's method are proposed. The Navier-Stokes equations without the pressure gradient term (Burger's equations) are used as test problems. Unlike full Newton, the proposed schemes are found to be stable for all cases tested, demand less computer storage and can be overrelaxed for faster convergence.

1. Introduction

The Navier-Stokes equations for incompressible fluid flow in Cartesian coordinates are used as starting point.

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{1.1}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-v\Delta u+\frac{\partial p}{\partial x}=0, \tag{1.2}$$

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}-v\Delta v+\frac{\partial p}{\partial y}=0. \tag{1.3}$$

The solution of equations (1.1), (1.2) and (1.3) is difficult due to several problems, namely, the discretization of the convective non-linear terms, the satisfaction of the continuity constraint and the solution of the resulting set of non-linear algebraic equations. All of these three problems are of current interest to the computational fluid dynamics community. In this paper we concentrate on the third problem and attempt to formulate efficient solution procedures based on local Newton linearization. For this aim we remove the pressure term from the momentum equation to obtain the well-known Burger's equations in two dimensions. These equations contain the convective non-linearity typicality of fluid dynamics equations. Furthermore, they possess readily computable exact solutions for many combinations of initial and boundary conditions. For this reason they are appropriate model equations to test various computational techniques. Fletcher [2] used Burger's equation to model various physical phenomena and to test several numerical methods. He also solved the two-dimensional equations using full Newton's method with and without pseudo-transient formulation [3]. We introduce improved schemes based on local Newton linearization and compare our results to those available in [2]. In addition we study and compare the behaviour of the proposed schemes on various grids and for the two test cases given in the Appendix.

2. Discretization

Using central differences for both the convectice and the diffusive terms, the equations 1.2 and 1.3 without the pressure gradient term are discretized on the domain shown in the Figure 1.

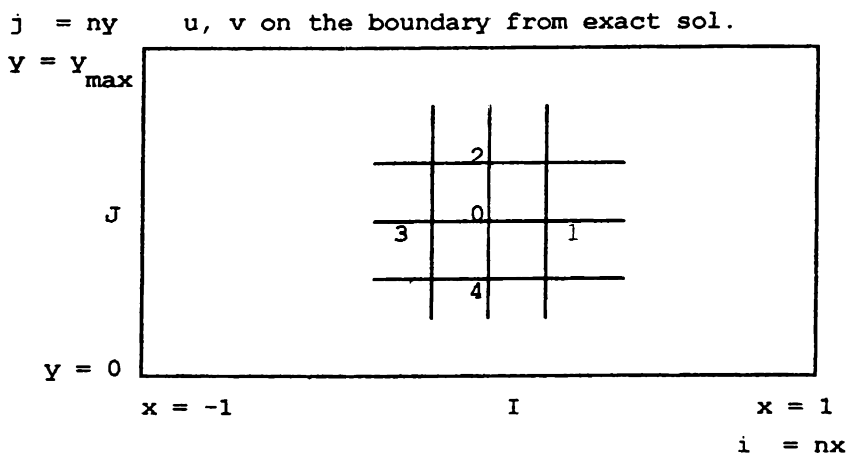


Fig.1. Solution domain

(2.1)
$$R_{u0} = u_0\frac{u_1 - u_3}{2h_x} + v_0\frac{u_2 - u_4}{2h_y} - v\left(\frac{u_1 - 2u_0 + u_3}{h_x^2} + \frac{u_2 - 2u_0 + u_4}{h_y^2}\right) = 0,$$

(2.2)
$$R_{v0} = u_0\frac{v_1 - v_3}{2h_x} + v_0\frac{v_2 - v_4}{2h_y} - v\left(\frac{v_1 - 2v_0 + v_3}{h_x^2} + \frac{v_2 - 2v_0 + v_4}{h_y^2}\right) = 0,$$

where R_{u0} and R_{v0} are the residuals for the equations of u and v. Several solution schemes can be derived for equation (2).

3. Newton's method

The equations (2) will be written in the form

$$R_u^{n+1}(u,v) = 0, \qquad R_v^{n+1}(u,v) = 0, \tag{3.1}$$

where n denotes the iteration number. The expansion of residuals over an iteration step yields

$$R_u^{n+1} = R_u^n + \frac{\partial R}{\partial u}\Delta u^{n+1} + \frac{\partial R}{\partial v}\Delta v^{n+1} = 0, \quad \text{where} \quad \Delta u^{n+1} = u^{n+1} - u^n \tag{3.2}$$

and similar expressions for R_v and Δv. The higher-order terms have been neglected. Equation (3.2) is a set of linear algebraic equations for the corrections and can be written in matrix form as follows

$$\begin{bmatrix} \dfrac{\partial R_u^n}{\partial u} & \dfrac{\partial R_u^n}{\partial v} \\ \dfrac{\partial R_v^n}{\partial u} & \dfrac{\partial R_v^n}{\partial v} \end{bmatrix} \begin{bmatrix} \Delta u^{n+1} \\ \Delta v^{n+1} \end{bmatrix} = -\begin{bmatrix} R_u^n \\ R_v^n \end{bmatrix}. \tag{3.3}$$

4.1. Point Newton-Gauss-Seidel

In equation (3.3), if we set the corrections to zero except at point 0 in Figure 1, we get a 2×2 system coupling the two unknowns u and v.

$$\left[\frac{u_1 - u_3}{2h_x} + 2v\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right]\Delta u_0 + \left[\frac{u_2 - u_4}{2h_y}\right]\Delta v_0 = -Ru_0, \tag{4.1}$$

$$\left[\frac{v_1 - v_3}{2h_x}\right]\Delta u_0 + \left[\frac{v_2 - v_4}{2h_y} + 2\nu\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right]\Delta v_0 = -Rv_0. \tag{4.2}$$

This method will be called Block Point Newton-Gauss-Seidel (BPNGS).

4.2. Line Newton-Gauss-Seidel

As we will see later at high grid aspect ratios (discrete anisotropy) point methods are not efficient, because the unknowns are strongly coupled in one direction and they have to be relaxed simultaneously. We call this arrangement Block Line Newton-Gauss-Seidel (BLNGS) scheme. We distinguish XLNGS, in which the linearization is done along the x-axis, and YLNGS, in which only the unknowns along the y-line are included.

4.2.1. The XLNGS scheme

Dropping all corrections except at those points included in the discretization along the x-axis we obtain

$$-\left[\frac{u_0}{2h_x} + \frac{\nu}{h_x^2}\right]\Delta u_3 + \left[\frac{u_1 - u_3}{2h_x} + 2\nu\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right]\Delta u_0 + \left[\frac{u_2 - u_4}{2h_y}\right]\Delta v_0 +$$

$$+\left[\frac{u_0}{2h_x} - \frac{\nu}{h_x^2}\right]\Delta u_1 = -Ru_0, \tag{5.1}$$

$$-\left[\frac{u_0}{2h_x} + \frac{\nu}{h_x^2}\right]\Delta v_3 + \left[\frac{v_2 - v_4}{2h_y} + 2\nu\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)\right]\Delta v_0 + \left[\frac{v_1 - v_3}{2h_x}\right]\Delta u_0 +$$

$$+\left[\frac{u_0}{2h_x} - \frac{\nu}{h_x^2}\right]\Delta v_1 = -Ru_0. \tag{5.2}$$

When equation (5) is written in a 5×5 grid in matrix form, we get

$$\begin{bmatrix} b & c & d & 0 & 0 & 0 \\ e & f & 0 & d & 0 & 0 \\ a & 0 & b & c & d & 0 \\ 0 & a & e & f & 0 & d \\ 0 & 0 & a & 0 & b & c \\ 0 & 0 & 0 & a & e & f \end{bmatrix} \begin{bmatrix} \Delta u_3 \\ \Delta v_3 \\ \Delta u_0 \\ \Delta v_0 \\ \Delta u_1 \\ \Delta v_1 \end{bmatrix} = \begin{bmatrix} -Ru_3 \\ -Rv_3 \\ -Ru_0 \\ -Rv_0 \\ -Ru_1 \\ -Rv_1 \end{bmatrix}. \tag{5.3}$$

The coefficients a, b, c, d, e and f are obvious from equations (5.1)-(5.2).

4.2.2. The YLNGS scheme

Similar steps produce the YLNGS scheme:

$$-\left[\frac{v_0}{2h_y}+\frac{v}{h_y^2}\right]\Delta u_4+\left[\frac{u_1-u_3}{2h_x}+2v\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\right]\Delta u_0+\left[\frac{u_2-u_4}{2h_y}\right]\Delta v_0+$$

$$+\left[\frac{v_0}{2h_y}-\frac{v}{h_y^2}\right]\Delta u_2=-Ru_0, \tag{6.1}$$

$$-\left[\frac{u_0}{2h_y}+\frac{v}{h_y^2}\right]\Delta v_4+\left[\frac{v_1-v_3}{2h_x}\right]\Delta u_0+\left[\frac{v_2-v_4}{2h_y}+2v\left(\frac{1}{h_x^2}+\frac{1}{h_y^2}\right)\right]\Delta v_0+$$

$$+\left[\frac{v_0}{2h_y}-\frac{v}{h_y^2}\right]\Delta v_2=-Rv_0. \tag{6.2}$$

5. Solution procedure

The equations (4,5,6) can be written in the general Newton form

$$J\Delta\Phi=-R, \tag{7.1}$$

where J is the Jacobian, $\Delta\Phi$ is the correction vector of u and v, and R is the residual vector. The corrections obtained from equation (7.1) are applied to the unknowns as follows

$$\begin{aligned} u_0^{n+1} &= u_0^n+\omega_u\partial u_0, \\ v_0^{n+1} &= v_0^n+\omega_v\partial v_0, \end{aligned} \tag{7.2}$$

where n is the iteration number and ω is a relaxation parameter. When $\omega<1$ the corrections are damped or underrelaxed. For $\omega>1$ we have overrelaxation.

In scheme (4) the domain is swept point by point. The sweep direction may affect the rate of convergence [3,4]. In arrengements (5) and (6) the domain is swept in lines, in the X- and Y-direction depending on the nature of the domain and the problem to be solved. It is also possible to combine the two line sweeps in alternating fashion to produce a more robust yet more expensive scheme. We give the procedure we have used to solve (5.3) as a pentadiagonal system.

```
for all (ny-2) j-lines do
    begin
        calculate coefficients a, b, c, d, e, f for line j
        calculate residuals r on line j
        solve the pentadiagonal system: Penta(nx, a, b, c, d, e, f, r, ΔΦ);
        correct the unknowns as in (7.2)
    end;
```

Here nx and ny are the numbers of grid points in the X and Y directions.

6. Results

Schemes 4, 5 and 6 have been implemented to obtain steady-state solution in the computational domain (see Fig.1) $-1 \leq x \leq 1$ and $0 \leq y \leq y_{max}$, where $y_{max} = \pi/6\lambda$ (λ is a constant used to control the behaviour of the exact sol-

Scheme	Iterations	10^6rms	Relax. factors
full Newton*	- 115	diverged 9.87	1.0 0.15
Pseudo-transient Newton*	23	8.32	$\Delta t = 0.01$
BPNGS	23 11	9.70 4.00	1.0 1.2
XLNGS	15 10	7.44 8.30	1.0 1.2
YLNGS**	46	8.6	1.0

Table 1. Comparison of various schemes (test-case 1 on 5×5 grid)
* reference [3]; ** overelaxation ineffective

ution). Dirichlet boundary conditions are obtained from the exact solution (see Appendix). First, the exact solution is used as a starting solution to be able to compare the results to those given in [3]. The comparison produced the results shown in Table 1. Figures 2 and 3 display the convergence rates for the BPNGS and XLNGS schemes for various overrelaxation factors and several grid sizes. Nearly optimal (by numerical experimentation) overrelaxation factors for the two schemes on various grids are quoted in Table 2 and plotted in Figure 4. Convergence rates at optimal relaxation factors are given in Figure 5. To assure comparison on an equal basis, execution times to reach certain levels of accuracy are provided in Figure 6. The effect of steep gradients on the rate of convergence is shown in Figure 7.

nx x ny	BPNGS	XLNGS
5x5	1.2	1.2
9x9	1.5	1.3
17x17	1.7	1.3
33x33	1.85	1.15
65x65	1.9	1.05

Table 2. Optimal overrelaxation factors for various grid dimensions

6. Conclusions

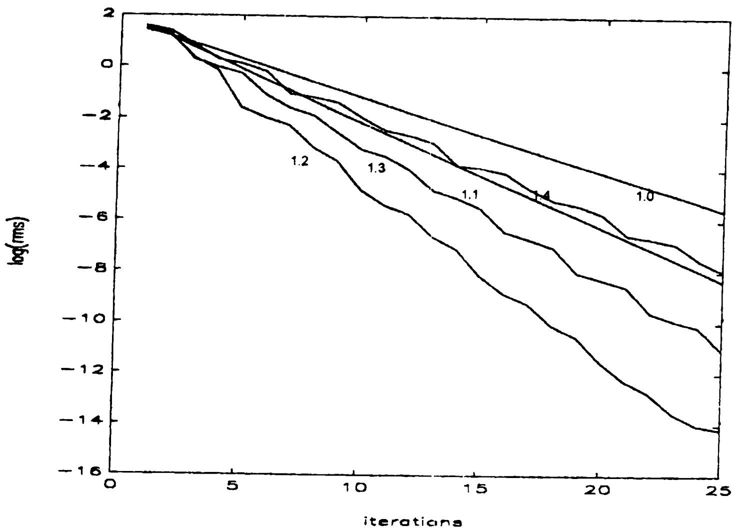
From the results presented in Tables 1 and 2 and Figures 2-7 we draw the following conclusions:

• Local linearization schemes have shown good convergence for all cases and number of grid points considered in contrast to the full Newton's method which has been reported [3] to diverge on a grid of 5×5 for test-case 1.

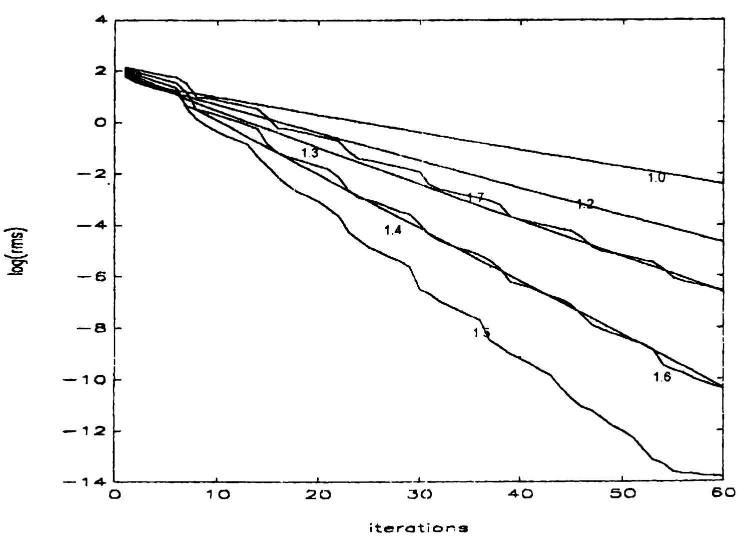
• Overrelaxation was found to improve the rate of convergence of the BPNGS and XLNGS schemes especially on small number of grid points.

• Point relaxation schemes on grids with small number of points perform as good as line relaxation (see Figure 4 and 5). On large grids, however, they perform very poor.

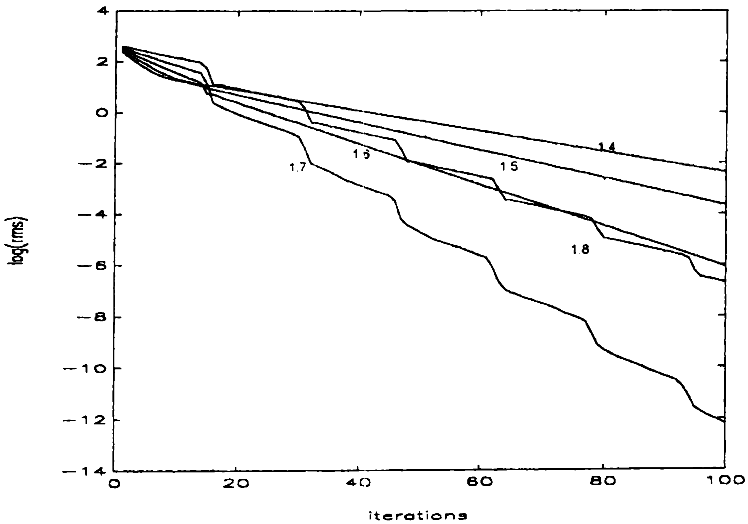
• On larger grids considerable time saving is achieved by the use of line relaxation instead of point relaxation for the same levels of accuracy (see Figure 5).



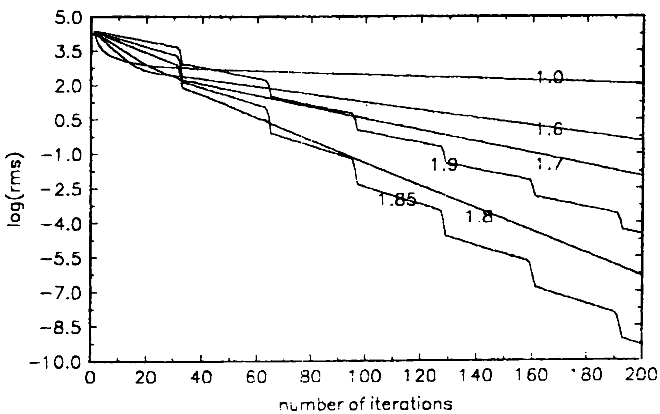
2a. 5×5 grid



2b. 9×9 grid

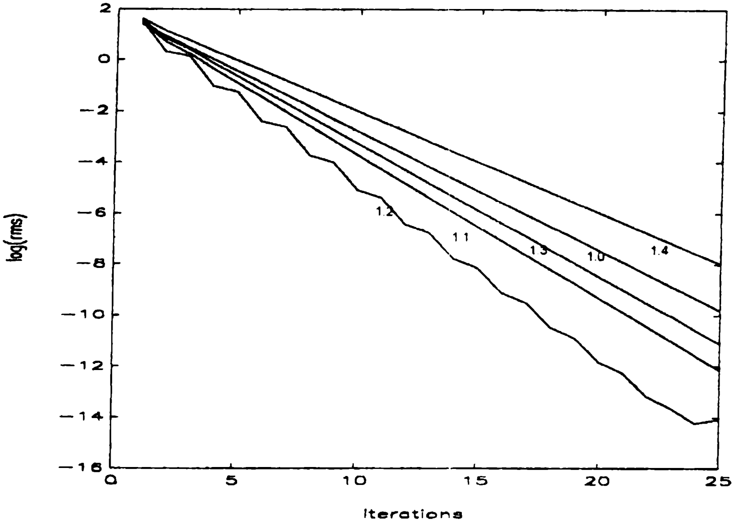


2c. 17 × 17 grid

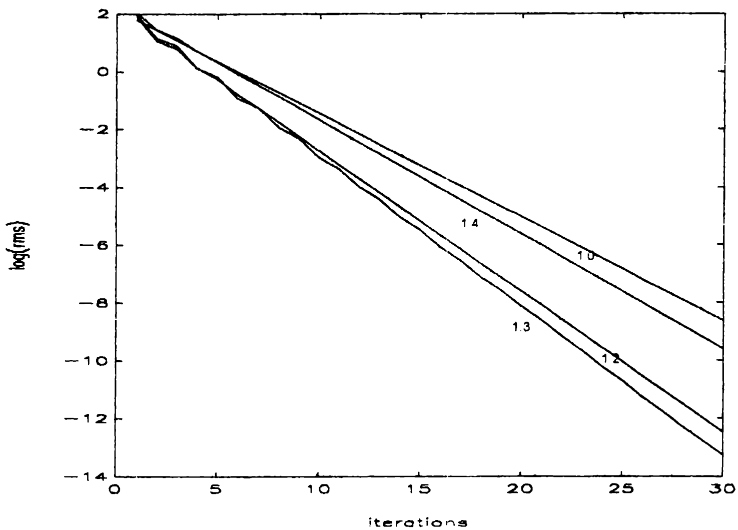


2d. 33 × 33 grid

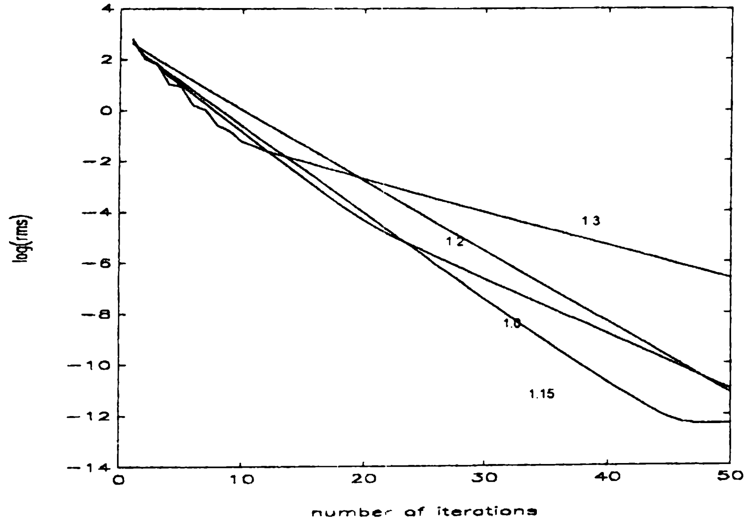
Fig.2. Convergence rates for BPNGS for various overrelaxation factors



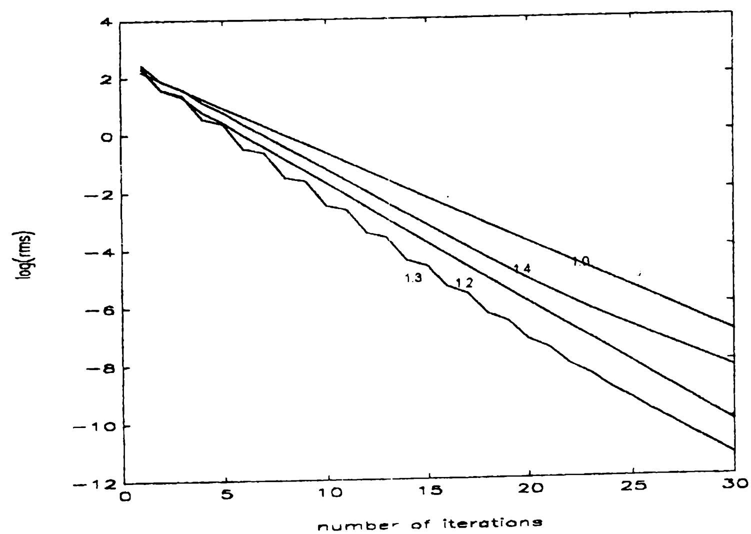
3a. 5 × 5 grid



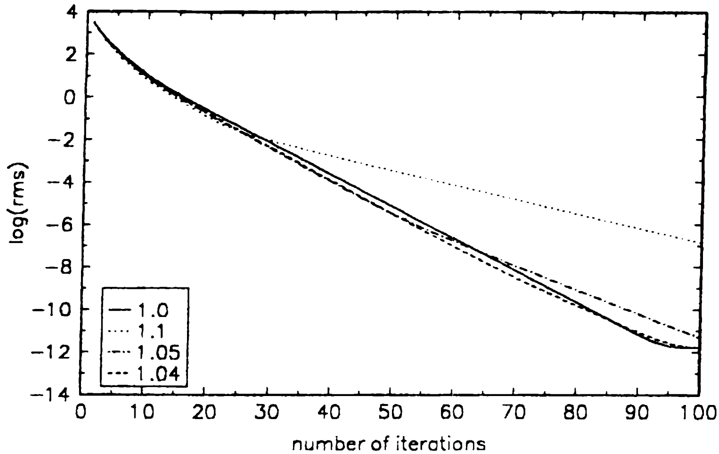
3b. 9 × 9 grid



3c. 17 × 17 grid



3d. 33 × 33 grid



3e. 65 × 65 grid

Fig.3. Convergence rates for XLNGS for various grids and overrelaxation factors

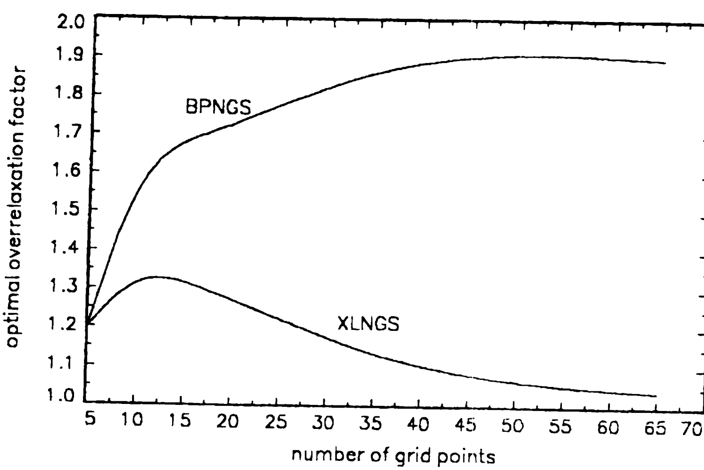


Fig.4. Optimal overrelaxation factors

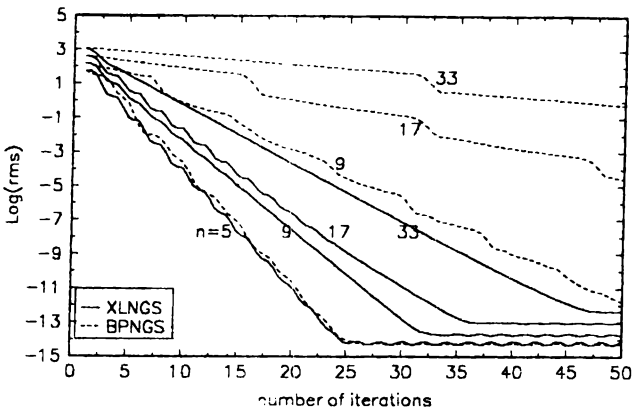
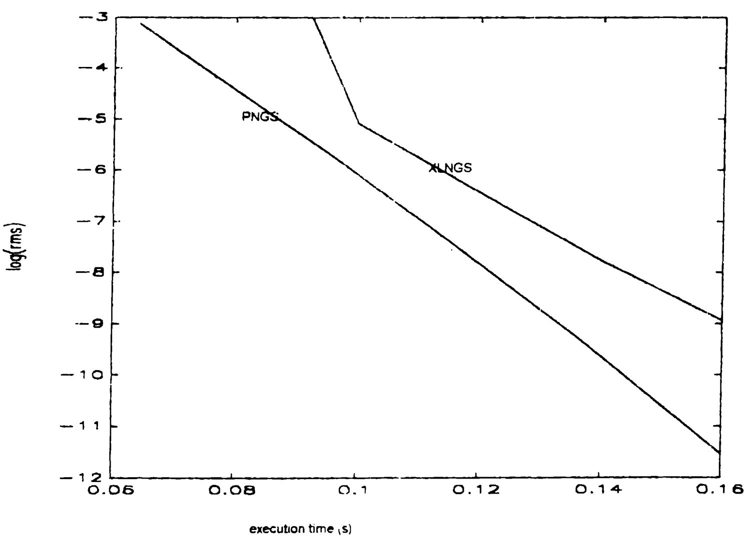
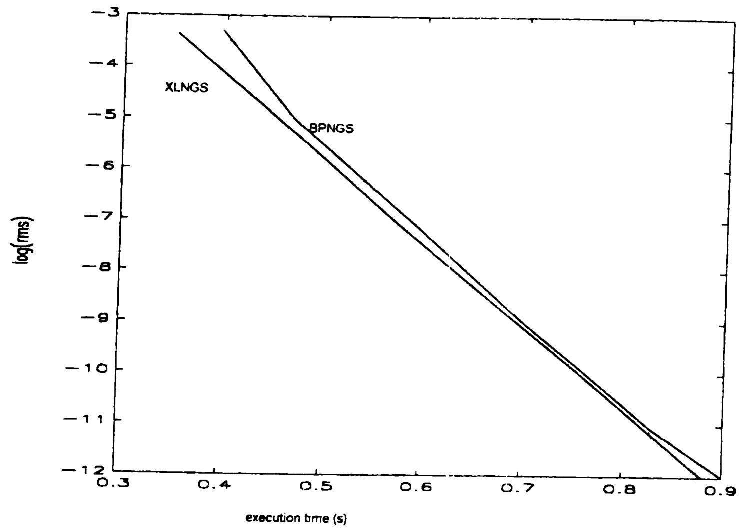


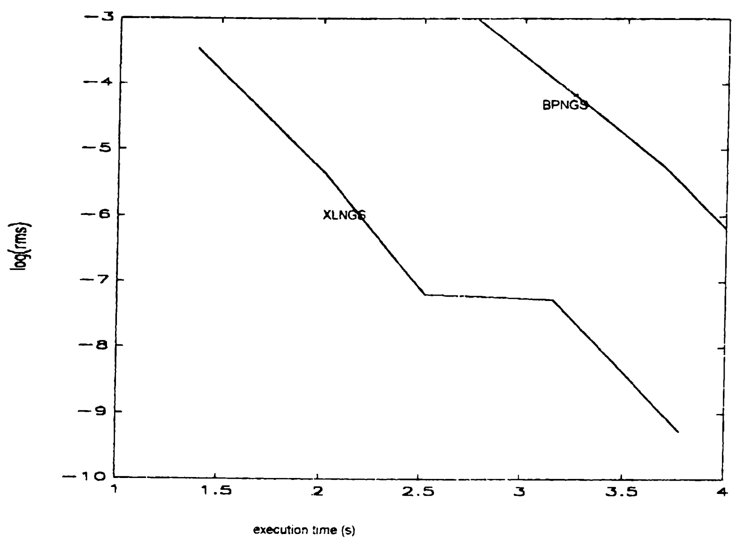
Fig.5. Convergence rates at optimal overrelaxation factors on various grids



6a. 5×5 grid

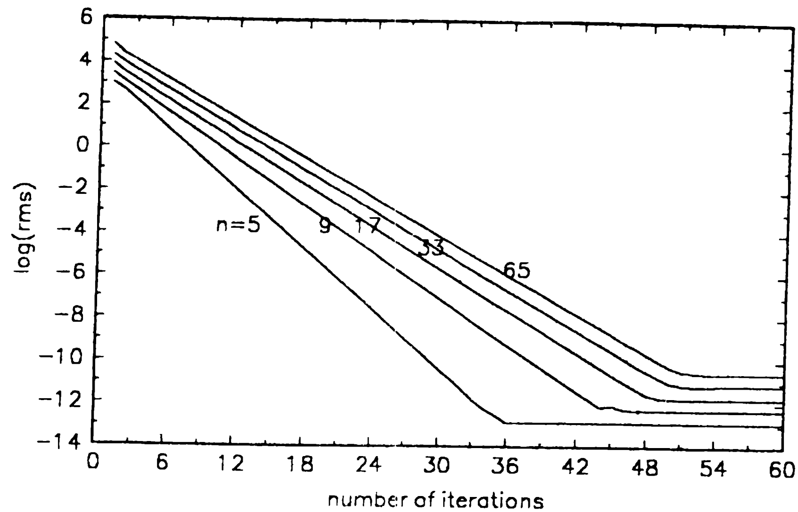


6b. 9 × 9 grid

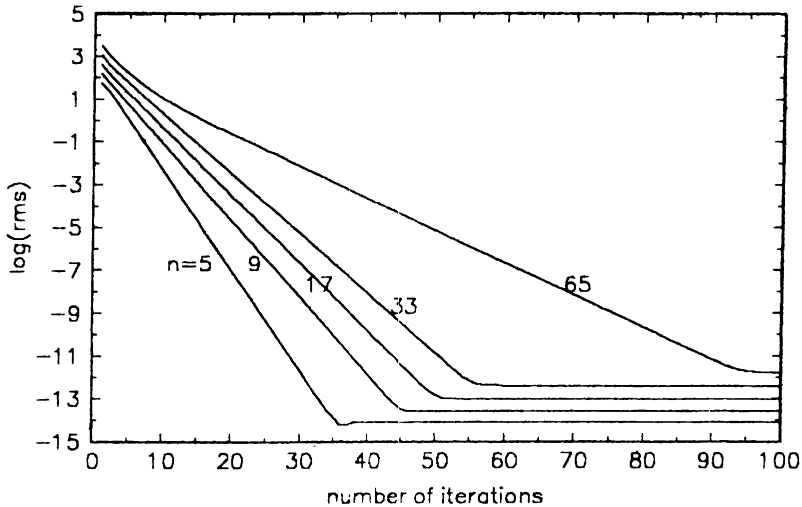


6c. 17 × 17 grid

Fig.6. Comparison of execution times for BPNGS and XLNGS on various grids



7a. Case 2



7b. Case 1

Fig. 7. Convergence rates for XLNGS for various grids for the two cases

• From the last two conclusions it follows that point relaxation may be cheaper if one uses as a solver on the coarsest grid in a multigrid scheme.

• The trends of the optimal overrelaxation factors shown in Figure 3 promote further study using various problems and boundary conditions. For the studied cases the optimal overrelaxation factor tends to 1.0 as the number of grid points increases in the case of the XLNGS scheme. In the case of BPNGS it is interesting to note the increase of the optimal overrelaxation factor with the increase of number of grid points. Further study is needed before drawing any general conclusions.

• The YLNGS scheme performed is rather poor even on a 5×5 grid (see Table 1). Overrelaxation was ineffective for this scheme. This emphasizes the importance of sweeping in the proper direction when using line relaxation methods (this is the down-stream direction, see [1,4,6]).

Acknowledgement. The author is very grateful for the support and guidance of Prof. G.Stoyan of the Eötvös Loránd University, Department of Numerical Analysis, Budapest.

References

[1] **Brandt A.**, Guide to multigrid development, *Multigrid methods,* eds. W.Hachbusch and U.Trottenberg, Lecture Notes in Mathematics **960**, Springer, 1981, 220-312.

[2] **Fletcher C.A.J.**, Burger's equation: A model for all reason, *Numerical solution of differential equations,* ed. J.Noye, North Holland, 1983, 355-475.

[3] **Fletcher C.A.J.**, *Computation techniques for fluid dynamics,* Springer, 1988.

[4] **Hackbusch W.**, *Multigrid methods and applications,* Springer, 1985.

[5] **Patankar S.V.**, *Numerical heat transfer and fluid flow,* Series in Computational Methods in Mechanics and Thermal Sciences, McGraw Hill, 1980.

[6] **Wesseling P.**, *An introduction to multigrid methods,* Wiley-Interscience, 1992.

Appendix

1. Exact solution of Burger's equations

Using the Cole-Hopf transformation [3]

$$(1)\qquad \overline{u} = \frac{-2v\dfrac{\partial\overline{\Phi}}{\partial x}}{\overline{\Phi}}, \qquad \overline{v} = \frac{-2v\dfrac{\partial\overline{\Phi}}{\partial y}}{\overline{\Phi}}.$$

Burger's equations are transformed to

$$(2)\qquad \frac{\partial^2\overline{\Phi}}{\partial x^2} + \frac{\partial^2\overline{\Phi}}{\partial y^2} = 0.$$

Equation (2) has the exact solution

$$\overline{\Phi} = a_1 + a_2 x + a_3 y + a_4 xy + a_5\left[e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)}\right]\cos(\lambda y),$$

$a_1, \ldots, a_5, \lambda$ and x_0 are parameters chosen to give appropriate features to the exact solution.

The corresponding exact solutions for u and v are

$$(3.1)\qquad \overline{u} = \frac{-2v\left\{a_2 + a_4 y + \lambda a_5\left[e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)}\right]\cos(\lambda y)\right\}}{a_1 + a_2 x + a_3 y + a_4 xy + a_5\left[e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)}\right]\cos(\lambda y)},$$

$$(3.2)\qquad \overline{v} = \frac{-2v\left\{a_3 + a_4 x - \lambda a_5\left[e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)}\right]\sin(\lambda y)\right\}}{a_1 + a_2 x + a_3 y + a_4 xy + a_5\left[e^{\lambda(x-x_0)} + e^{-\lambda(x-x_0)}\right]\cos(\lambda y)}.$$

2. Test cases

Two test cases are considered.

2.1. moderate internal gradient

$a_1 = a_2 = 110, \quad a_3 = a_4 = 0, \quad a_5 = 1.0, \quad x_0 = 1.0, \quad \lambda = 5$ and $v = 0.1$.

2.2. sever internal gradient

$a_1 = a_2 = 1.3\times 10^{13}, \; a_3 = a_4 = 0, \; a_5 = 1.0, \; x_0 = 1.0, \; l = 25$ and $n = 0.04$.

(Received January 17, 1995)

I. Tuwegiar
Hungarian Academy of Sciences
Szentmihályi u. 26.
H-1144 Budapest, Hungary