INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2005; **62**:1559–1578
Published online 11 January 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1250

On a high-order Newton linearization method for solving the incompressible Navier–Stokes equations

Tony W. H. Sheu*, †, ‡ and R. K. Lin§

Department of Engineering Science and Ocean Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan, Republic of China

SUMMARY

The present study aims to accelerate the non-linear convergence to incompressible Navier–Stokes solution by developing a high-order Newton linearization method in non-staggered grids. For the sake of accuracy, the linearized convection–diffusion–reaction finite-difference equation is solved line-by-line using the nodally exact one-dimensional scheme. The matrix size is reduced and, at the same time, the CPU time is considerably saved owing to the reduction of stencil points. This Newton linearization method is computationally efficient and is demonstrated to outperform the classical Newton method through computational exercises. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: incompressible Navier–Stokes solution; high-order Newton linearization; non-staggered grids; convection–diffusion–reaction; nodally exact

1. INTRODUCTION

Numerical simulation of fluid flow encounters difficulties in accurately approximating and properly linearizing the multi-dimensional advection terms in the discretized momentum equations. The chosen linearization methodology can significantly affect the solution convergence sequences [1]. Development of an effective linearization method is, thus, an important task in the area of computational fluid dynamics. This study aims to refine the conventional Newton linearization procedure for accelerating convergence towards the final solution.

Linearization of momentum equations can be achieved straightforwardly by lagging the non-linear term. This simple iteration method (SIM) may result in slow convergence and is hence

*Correspondence to: T. W. H. Sheu, Department of Engineering Science and Ocean Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan, Republic of China.
†E-mail: twhsheu@ntu.edu.tw
‡Professor.
§PhD candidate.

Contract/grant sponsor: National Science Council; contract/grant number: NSC 92-2811-E-002-008

Received 3 March 2003
Revised 7 May 2004
Accepted 17 September 2004

Copyright © 2005 John Wiley & Sons, Ltd.

very time consuming. To improve the convergence, one can apply the potentially attractive Newton linearization method due to its q-quadratic convergence [2]. In Newton's method, the matrix equation $\mathbf{A}\underline{x} = \underline{b}$, where $\underline{x}$ is the state vector and $\mathbf{A}$ is a function of $\underline{x}$, is approximated by a first-order Taylor series to render $\mathbf{J}^k \delta\underline{x}^{k+1} = \underline{R}^k$. Here, k denotes the iteration level and one component in $\mathbf{J}$ is given by $\mathbf{J}_{ij} = -\partial\underline{R}(i)/\partial x(j)$, where $\underline{R}$ ($\equiv \underline{b} - \mathbf{A}\underline{x}$) is the residual vector. Then the matrix equation is solved for $\delta\underline{x}^k$ from $\mathbf{J}^k \delta\underline{x}^{k+1} = \underline{R}^k$, followed by obtaining the updated solution from $\underline{x}^{k+1} = \underline{x}^k + \delta\underline{x}^{k+1}$. Calculations are continued until the residual falls below a user's specified tolerance, with $\mathbf{J}^k$ and $\underline{R}^k$ being calculated from the most updated value of $\underline{x}$.

Newton method has been known as a powerful technique for solving non-linear fluid and heat transfer equations. Implementation of this method requires, however, a massive calculation in the factorization of $\mathbf{J}^k$ [1]. Another drawback in using Newton's method, assuming the required memory is available, is due to its small radius of convergence. Using the variable secant procedure, which is now generally known as the Newton–Raphson method, can overcome these difficulties by replacing the derivative terms with a secant line approximation through two points. However, this potentially attractive linearization needs to factorize the tangent matrix in each iteration [3]. The inexact Newton method [4] was proposed to reduce the computational cost required in the classical Newton–Raphson linearization method. In the modified Newton–Raphson method, the tangent matrix is factorized once for a number of steps. This occasionally updating strategy may lead to poor convergence in the simulation of a highly non-linear equation. As a means of partly circumventing this problem, the asymptotic Newton method was proposed [3]. In this paper a straightforward implementation of high-order Newton linearization procedures is proposed for obtaining non-linear solution at a less computational cost.

The rest of this paper is organized as follows. In Section 2, a high-order Newton linearization method is detailed. This is followed by presenting the proposed scheme that is mostly suited to solve for the resulting linearized momentum equations. In Section 3, the incompressible Navier–Stokes equations are linearized and then they are discretized on non-staggered grids. An assessment of different linearization methods is given in Section 4. Section 5 provides the concluding remarks.

2. LINEARIZATION OF NAVIER–STOKES EQUATIONS

In this paper, the two-dimensional viscous equations for an incompressible fluid flow are considered. Subject to the divergence-free constraint condition given in (1), the steady-state flow equations for the velocity vector $\mathbf{u} = (u, v)$ and the pressure p are as follows:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} + \mathbf{f} \tag{2}$$

Note that Equations (2) is the working equations for $\mathbf{u}$. By virtue of the continuity equation, equation for p can be derived as follows by summing $\partial/\partial x$ (x-momentum) and $\partial/\partial y$ (y-momentum):

$$\nabla^2 p = \nabla\left[\frac{1}{Re}\nabla^2\mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u} + \mathbf{f}\right] \tag{3}$$

Copyright © 2005 John Wiley & Sons, Ltd.

The above equation will be solved subject to the Neumann-type pressure boundary condition

$$\frac{\partial p}{\partial n} = \left[\frac{1}{Re} \nabla^2 \mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u} + \mathbf{f} \right] \cdot \mathbf{n} \tag{4}$$

where $\mathbf{n}$ denotes the outward-directed unit vector normal to the boundary of domain. For simplicity, the dynamic viscosity μ is considered to be uniform.

Newton's method is a powerful technique for solving the system of non-linear equations given by $\mathbf{A}(\mathbf{u})\mathbf{u} = \underline{b}$, where $\mathbf{u} = (u, v)$. After calculating the pressure solution from (3) to (4), one can obtain $\Delta\mathbf{u}^{k+1}$ $(\equiv \mathbf{u}^{k+1} - \mathbf{u}^k)$ from $\mathbf{J}^k \Delta\mathbf{u}^{k+1} = -\underline{r}^k$ using the Newton's method. Here, k denotes the iteration level and $\mathbf{J}^k \equiv \{\mathbf{J}_{ij} = \partial \underline{r}_i^k / \partial \mathbf{u}_j^k\}$ is the Jacobian matrix. In the above method, the execution time is mainly spent on the factorization of $\mathbf{J}^k$. For this reason, it is possible to reduce the computational time by solving two non-linear equations in (2) separately.

The first step towards solving the non-linear equation is to linearize the convective terms shown in (2). We rewrite Equations (2) as $F(u) = 0$ and $G(v) = 0$, respectively, where

$$F(u) = uu_x + vu_y - \frac{1}{Re}(u_{xx} + u_{yy}) + p_x \tag{5}$$

$$G(v) = uv_x + vv_y - \frac{1}{Re}(v_{xx} + v_{yy}) + p_y \tag{6}$$

Note that $F'(u) \neq 0$ and $G'(v) \neq 0$ are two necessary conditions for the quadratically convergent Newton's method. Define $q(u)$ and $r(v)$ by

$$q(u) = F(u)\ell(u) \tag{7}$$

$$r(v) = G(v)h(v) \tag{8}$$

They are differentiated with respect to u and v, respectively, once and twice. These differentiations enable us to derive

$$q'(u) = F'(u)\ell(u) + F(u)\ell'(u) \tag{9}$$

$$r'(v) = G'(v)h(v) + G(v)h'(v) \tag{10}$$

$$q''(u) = F''(u)\ell(u) + 2F'(u)\ell'(u) + F(u)\ell''(u) \tag{11}$$

$$r''(v) = G''(v)h(v) + 2G'(v)h'(v) + G(v)h''(v) \tag{12}$$

In the course of iteratively calculating u and v, we set $F(u) = G(v) = 0$ as the acceleration means to have convergent solutions from the respective non-linear differential equations. As a consequence of this acceleration, $q''(u)$ and $r''(v)$ turn out to be zero owing to $q = F\ell$ and $r = Gh$.

Copyright © 2005 John Wiley & Sons, Ltd.

Now, $q(u^{k+1})$ and $r(v^{k+1})$ are expanded with respect to u^k and v^k, respectively, and the Taylor series expansion is terminated up to the third-order term. The superscript $k+1$ shown above represents the updated iteration number. Taking $q''(u^k)$ and $r''(v^k)$ into account, we have

$$u^{k+1} = u^k + \frac{F(u)\ell(u^k)}{F'(u)\ell(u) + F(u)\ell'(u)} \tag{13}$$

$$v^{k+1} = v^k + \frac{G(v)h(v^k)}{G'(v)h(v) + G(v)h'(v)} \tag{14}$$

As Equations (13) and (14) reveal, equations for $\ell(u^k)$ and $h(v^k)$ must be derived to accomplish the linearization of advective terms.

Substitution of $q'' = r'' = 0$ into Equations (11) and (12), respectively, renders

$$F''(u)\ell(u) + 2F'(u)\ell'(u) = 0 \tag{15}$$

$$G''(v)h(v) + 2G'(v)h'(v) = 0 \tag{16}$$

F'' and G'' are then approximated by

$$F''(u) = \frac{F'(u^k) - F'(u^{k-1})}{u^k - u^{k-1}} \tag{17}$$

$$G''(v) = \frac{G'(v^k) - G'(v^{k-1})}{v^k - v^{k-1}} \tag{18}$$

By substituting (17) and (18) into (15) and (16), the updated $\ell(u)$ and $h(v)$ are obtained as follows from the resulting two coupled equations:

$$\ell(u^k) = \frac{2F'(u^k)}{3F'(u^k) - F'(u^{k-1})}\,\ell(u^{k-1}) \tag{19}$$

$$h(v^k) = \frac{2G'(v^k)}{3G'(v^k) - G'(v^{k-1})}\,h(v^{k-1}) \tag{20}$$

By definitions of q and r as given in (7) and (8), Equations (13) and (14) can be rewritten as

$$u^{k+1} = u^k + \frac{q(u^k)}{q'(u^k)} \tag{21}$$

$$v^{k+1} = v^k + \frac{r(v^k)}{r'(v^k)} \tag{22}$$

Copyright © 2005 John Wiley & Sons, Ltd.

The derivation is followed by addition and subtraction of $F(u^k)/F'(u^k)$ and $G(v^k)/G'(v^k)$ in Equations (21) and (22), leading to

$$u^{k+1} = u^k - \frac{F(u^k)}{F'(u^k)} + \frac{F(u^k)}{F'(u^k)} + \frac{q(u^k)}{q'(u^k)} \tag{23}$$

$$v^{k+1} = v^k - \frac{G(v^k)}{G'(v^k)} + \frac{G(v^k)}{G'(v^k)} + \frac{r(v^k)}{r'(v^k)} \tag{24}$$

Now, we can calculate $u^k - (F(u^k)/F'(u^k))$ and $v^k - (G(v^k)/G'(v^k))$ using the conventional Newton linearization method. In other words, the updated velocities in (23) and (24) are rewritten as

$$u^{k+1} = u_{\mathrm{CNL}} + \frac{F(u^k)}{F'(u^k)} + \frac{q(u^k)}{q'(u^k)} \tag{25}$$

$$v^{k+1} = v_{\mathrm{CNL}} + \frac{G(v^k)}{G'(v^k)} + \frac{r(v^k)}{r'(v^k)} \tag{26}$$

where the conventional Newton linearization terms u_{CNL} and v_{CNL} are given by

$$u_{\mathrm{CNL}} = u^k - \frac{F(u^k)}{F'(u^k)} \tag{27}$$

$$v_{\mathrm{CNL}} = v^k - \frac{G(v^k)}{G'(v^k)} \tag{28}$$

Approximate $F'(u^k)$, $G'(v^k)$, $q(u^k)$ and $r(v^k)$ by

$$A'(u^k) = \frac{A(u_{\mathrm{CNL}}) - A(u^{k-1})}{u_{\mathrm{CNL}} - u^{k-1}} \quad (A = F' \text{ or } q) \tag{29}$$

$$B'(v^k) = \frac{B(v_{\mathrm{CNL}}) - B(v^{k-1})}{v_{\mathrm{CNL}} - v^{k-1}} \quad (B = G' \text{ or } r) \tag{30}$$

We then substitute them into (25) and (26) to derive

$$u^{k+1} = (1+\alpha)u_{\mathrm{CNL}} - \alpha u^{k-1} \tag{31}$$

$$v^{k+1} = (1+\beta)v_{\mathrm{CNL}} - \beta v^{k-1} \tag{32}$$

Copyright © 2005 John Wiley & Sons, Ltd.

where α and β are expressed as follows:

$$\alpha = \frac{F(u^k)}{F(u^k) - F(u^{k-1})} + \frac{q(u^k)}{q(u^k) - q(u^{k-1})} \tag{33}$$

$$\beta = \frac{G(v^k)}{G(v^k) - G(v^{k-1})} + \frac{r(v^k)}{r(v^k) - r(v^{k-1})} \tag{34}$$

3. NUMERICAL METHOD

3.1. Flux discretization scheme

Subject to the divergence-free equation (1), Equation (2) can be rewritten as

$$(u^2)_x + (uv)_y = -p_x + \frac{1}{Re}(u_{xx} + u_{yy}) \tag{35}$$

$$(uv)_x + (v^2)_y = -p_y + \frac{1}{Re}(v_{xx} + v_{yy}) \tag{36}$$

An arbitrarily chosen continuous function st is considered and expanded in terms of the Taylor series about its current iteration state. By truncating the resulting series expansion after the first-derivative terms, the expansion for st reads as

$$\begin{aligned} s^{k+1}\ t^{k+1} &= s^k t^k + \left[\frac{\partial}{\partial s}(st)^k\right](s^{k+1} - s^k) + \left[\frac{\partial}{\partial t}(st)^k\right](t^{k+1} - t^k) + \cdots + \text{H.O.T.} \\ &= s^{k+1}t^k + s^k t^{k+1} - s^k t^k + \cdots + \text{H.O.T.} \end{aligned} \tag{37}$$

Terms with the superscripts k and $k+1$ are those evaluated at the previous and the present iteration counters, respectively. According to Equation (37), we can linearize $(u^2)_x^{k+1}$ and $(uv)_y^{k+1}$ as

$$\begin{aligned} (u^2)_x^{k+1} &= (u^{k+1}u^k + u^k u^{k+1} - u^k u^k)_x \\ &= u^k u_x^{k+1} + u^{k+1}u_x^k - u^k u_x^k + \underline{u^{k+1}u_x^k + u^k u_x^{k+1} + u^k u_x^k} \end{aligned} \tag{38}$$

$$\begin{aligned} (uv)_y^{k+1} &= (u^{k+1}v^k + u^k v^{k+1} - u^k v^k)_y \\ &= v^k u_y^{k+1} + v^{k+1}u_y^k - v^k u_y^k + \underline{u^{k+1}v_y^k + u^k v_y^{k+1} + u^k v_y^k} \end{aligned} \tag{39}$$

Note that the terms with the underbar '—' originate from the high-order terms which are introduced to refine the classical linearization method. Substitution of (38) and (39) into

Copyright © 2005 John Wiley & Sons, Ltd.

(35) and (36) results in the following linearized x- and y-momentum equations, respectively

$$u^k u_x^{k+1} + v^k u_y^{k+1} - \frac{1}{Re}(u_{xx}^{k+1} + u_{yy}^{k+1}) + \underline{u_x^k u^{k+1}} = -p_x^{k+1} + \underline{u^k u_x^k + v^k u_y^k - u_y^k v^{k+1}} \tag{40}$$

$$u^k v_x^{k+1} + v^k v_y^{k+1} - \frac{1}{Re}(v_{xx}^{k+1} + v_{yy}^{k+1}) + \underline{v_y^k v^{k+1}} = -p_y^{k+1} + \underline{u^k v_x^k + v^k v_y^k - v_x^k u^{k+1}} \tag{41}$$

In the light of above two equations, it is important to consider the following constant-coefficient convection–diffusion–reaction (CDR) equation:

$$u\phi_x + v\phi_y - k(\phi_{xx} + \phi_{yy}) + c\phi = f \tag{42}$$

where k and c denote the diffusion coefficient and the reaction coefficient, respectively.

The finite difference solution for ϕ is sought consecutively from (42) according to the two steps given below

$$u\phi_x^* - k\phi_{xx}^* + c\phi^* = s_1 \tag{43}$$

$$v\phi_y^{n+1} - k\phi_{yy}^{n+1} + c\phi^{n+1} = s_2 \tag{44}$$

In the above, $s_1 = f - v\phi_y^n - k\phi_{yy}^n$ and $s_2 = f - u\phi_x^* - k\phi_{xx}^*$. Application of splitting technique to multi-dimensional equations typically generates a tri-diagonal system of equations that can be efficiently solved using the Thomas algorithm. This is the motivation for the use of splitting method of Peaceman and Rachford [5].

As Equations (43) and (44) show, it is essential to develop a discretization scheme for the following one-dimensional equation when solving the two-dimensional CDR equation (42)

$$u\phi_x - k\phi_{xx} + c\phi = f \tag{45}$$

For illustrative purposes, f is a known constant throughout. The discretized equation for the above model equation at an interior node i is assumed to take the following form:

$$\left(-\frac{u}{2h} - \frac{\gamma k}{h^2} + \frac{c}{6}\right)\phi_{i-1} + 2\left(\frac{\gamma k}{h^2} + \frac{c}{3}\right)\phi_i + \left(\frac{u}{2h} - \frac{\gamma k}{h^2} + \frac{c}{6}\right)\phi_{i+1} = f \tag{46}$$

where h is the mesh size. The resulting tri-diagonal system of algebraic equations for ϕ can be solved by Thomas algorithm, which requires only $5N - 4$ operations where N is the number of coupled algebraic equations. To determine γ, we take the following general solution into consideration:

$$\phi = c_1 e^{\lambda_1 x} + c_2 e^{\lambda_2 x} + \frac{f}{c} \tag{47}$$

where c_1 and c_2 are constants. Substituting (47) into Equation (45), it is easy to have $\lambda_{1,2} = (u/2k) \pm (\sqrt{u^2 + 4ck}/2k)$. The exact values of ϕ_i and $\phi_{i\pm1}$ are then substituted into (46)

Copyright © 2005 John Wiley & Sons, Ltd.

to derive γ analytically in terms of $R_1 = ch^2/2k$ and $R_2 = uh/2k$ [6]

$$\gamma = \frac{(R_1/3) + (R_1/3)\cosh(\overline{\lambda_1})\cosh(\overline{\lambda_2}) + R_2 \sinh(\overline{\lambda_1})\cosh(\overline{\lambda_2})}{\cosh(\overline{\lambda_1})\cosh(\overline{\lambda_2}) - 1} \tag{48}$$

In the above, $(\overline{\lambda_1},\ \overline{\lambda_2}) = (R_2,\ \sqrt{(R_2)^2 + 2R_1})$. It is now apparent that the numerical error stems solely from the approximation of f.

3.2. Non-staggered incompressible Navier–Stokes calculation

By theory, $-\nabla p$ in (2) should be approximated by a centred scheme. The resulting approximation of $\partial p/\partial x$ and $\partial p/\partial y$ can cause spurious even–odd oscillations to occur on a non-staggered grid [7, 8]. Therefore, in the incompressible flow simulation we have to eliminate the checkerboarding problem in a collocated grid system [9]. In this study the even–odd decoupling is eliminated by calculating F_j $(\equiv h\phi_x)$ and S_j $(\equiv h^2\phi_{xx})$ from the two equations given below

$$\begin{aligned}\alpha_0 F_{j+1} + \beta_0 F_j + \gamma_0 F_{j-1} = {} & a_0(\phi_{j+2} - \phi_{j+1}) + b_0(\phi_{j+1} - \phi_j) \\ & + c_0(\phi_j - \phi_{j-1}) + d_0(\phi_{j-1} - \phi_{j-2})\end{aligned} \tag{49}$$

and

$$\alpha_1 S_{j+1} + \beta_1 S_j + \gamma_1 S_{j-1} = a_1\phi_{j+2} + b_1\phi_{j+1} + c_1\phi_j + d_1\phi_{j-1} + e_1\phi_{j-2} \tag{50}$$

After some straightforward analyses, we are led to have the sixth-order accurate approximations of ϕ_x and ϕ_{xx} provided $(\alpha_0,\ \beta_0,\ \gamma_0,\ a_0,\ b_0,\ c_0,\ d_0) = \left(\frac{1}{5},\ \frac{3}{5},\ \frac{1}{5},\ \frac{1}{60},\ \frac{29}{60},\ \frac{29}{60},\ \frac{1}{60}\right)$ and $(\alpha_1,\ \beta_1,\ \gamma_1,\ a_1,\ b_1,\ c_1,\ d_1,\ e_1) = \left(1,\ \frac{11}{2},\ 1,\ \frac{3}{8},\ 6, -\frac{51}{4},\ 6,\ \frac{3}{8}\right)$.

The implicit equations for F and S at nodes immediately adjacent to the left or right boundary are derived by specifying $d_0 = e_1 = 0$ and $a_0 = a_1 = 0$, respectively. By virtue of the Taylor series expansion, the coefficients can be analytically derived as $(\alpha_0,\ \beta_0,\ \gamma_0,\ a_0,\ b_0,\ c_0,\ d_0) = \left(\frac{3}{10},\ \frac{3}{5},\ \frac{1}{10},\ \frac{1}{30},\ \frac{19}{30},\ \frac{1}{3},\ 0\right)$ and $\left(\frac{1}{10},\ \frac{3}{5},\ \frac{3}{10},\ 0,\ \frac{1}{3},\ \frac{19}{30},\ \frac{1}{30}\right)$ at nodes next to the left and right boundaries, respectively. In addition, coefficients shown in Equation (50) are analytically derived as $(\alpha_1,\ \beta_1,\ \gamma_1,\ a_1,\ b_1,\ c_1,\ d_1,\ e_1) = (1,\ 10,\ 1,\ 0,\ 12,\ -24,\ 12,\ 0)$.

4. NUMERICAL RESULTS

In what follows, the prescribed tolerance, $[1/N\sum(u^{\text{new}} - u^{\text{old}})^2]^{1/2}$, for each non-linear calculation is 10^{-13}. Here, N denotes the number of grid points.

4.1. Non-linear advection–diffusion scalar equation

We are now in a position to validate and then assess the proposed high-order Newton linearization method by solving u from the following non-linear convection–diffusion equation

Copyright © 2005 John Wiley & Sons, Ltd.

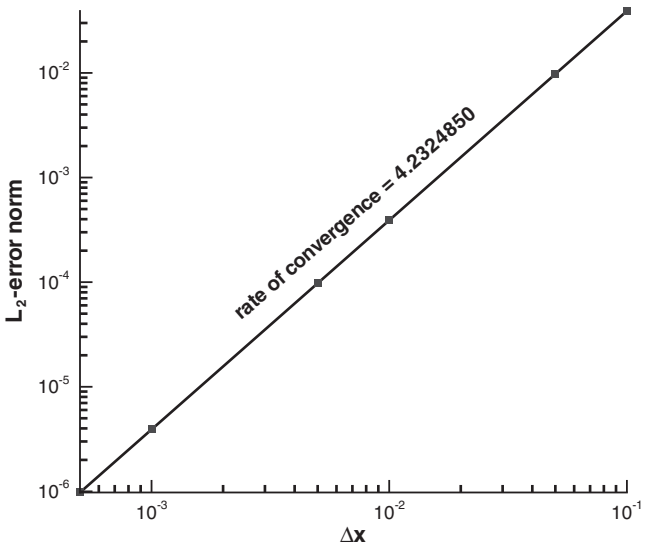

Figure 1. The computed rate of convergence for Equation (51).

in $0 \leqslant x, y \leqslant 1$:

$$uu_x + vu_y - k(u_{xx} + u_{yy}) = f \tag{51}$$

The solution to Equation (51) at $k = 10^{-3}$, $v = (x\cos(2xy) - y\sin(2xy))\exp(x^2 - y^2)$ and $f = -\frac{1}{2}(x^2 + y^2)\exp[2(x^2 - y^2)]$ is exactly derived as

$$u(x, y) = (x\sin(2xy) + y\cos(2xy))\exp(x^2 - y^2) \tag{52}$$

In Figure 1, we plot the rate of convergence for u, which is calculated from

$$C = \frac{\ln ||\text{err}_1|| - \ln ||\text{err}_2||}{\ln |h_1| - \ln |h_2|} \tag{53}$$

The simulated error is cast in its discrete L_2-norm as

$$E = \left[\frac{1}{M}\sum_{i=1}^{M^{1/2}} (\mathbf{u}_i - \mathbf{U}_i)^2\right]^{1/2} \tag{54}$$

where $\mathbf{U}_i$ denotes the exact solution at a nodal point i and $\mathbf{u}_i$ is the corresponding numerical solution.

For the sake of assessment of different linearization methods, the conventional Newton linearization method and the standard relaxation method, namely, $\mathbf{u}^{\text{new}} = w\mathbf{u}^{\text{new}} + (1 - w)\mathbf{u}^{\text{old}}$, where w is the user's specified constant linearzation parameter are considered. As Figure 2 shows, the required non-linear iteration number has been considerably reduced, compared with those needed for other two linearization methods. We also compare in the same figure the

Copyright © 2005 John Wiley & Sons, Ltd.

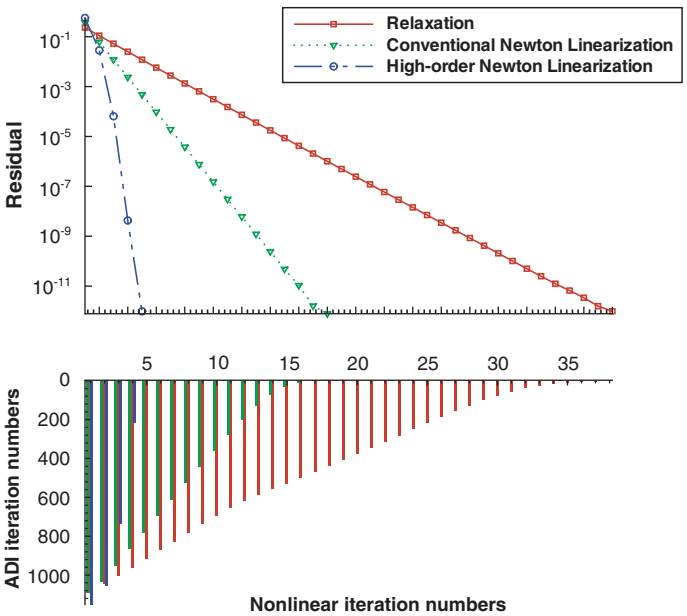

Figure 2. The plots of convergence histories and the needed inner iteration numbers for the three linearization methods used to solve for the non-linear advection–diffusion equation given in (51).

(inner) iteration numbers required to reach the convergent ADI solution at each non-linear (outer) iteration. It is seen that much fewer ADI iterations are required in each non-linear iteration when applying the presently proposed Newton linearization method.

4.2. *Steady-state Navier–Stokes equations*

Now, the Navier–Stokes equations at $\mathbf{f} = 0$ are solved. In the unit square, it is easy to derive the analytic pressure as follows:

$$p = \frac{-2}{(1+x)^2 + (1+y)^2} \tag{55}$$

This pressure is obtained when the boundary velocities u and v are specified according to

$$u = \frac{-2(1+y)}{(1+x)^2 + (1+y)^2} \tag{56}$$

$$v = \frac{2(1+x)}{(1+x)^2 + (1+y)^2} \tag{57}$$

The simulated rates of convergence in Figure 3 show the validity of the method. Similar to the conclusion drawn from the previous scalar problem, much fewer non-linear and ADI iteration numbers are needed to obtain the convergent solutions, as seen from Figure 4.

Copyright © 2005 John Wiley & Sons, Ltd.

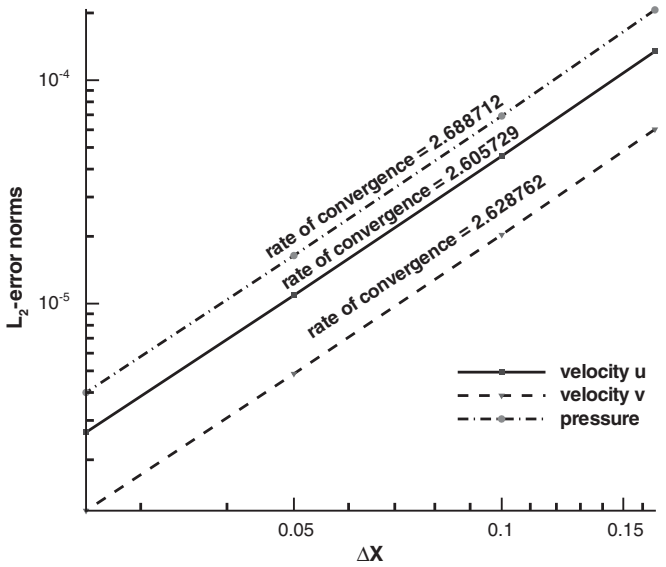

Figure 3. The computed rates of convergence for u, v and p.

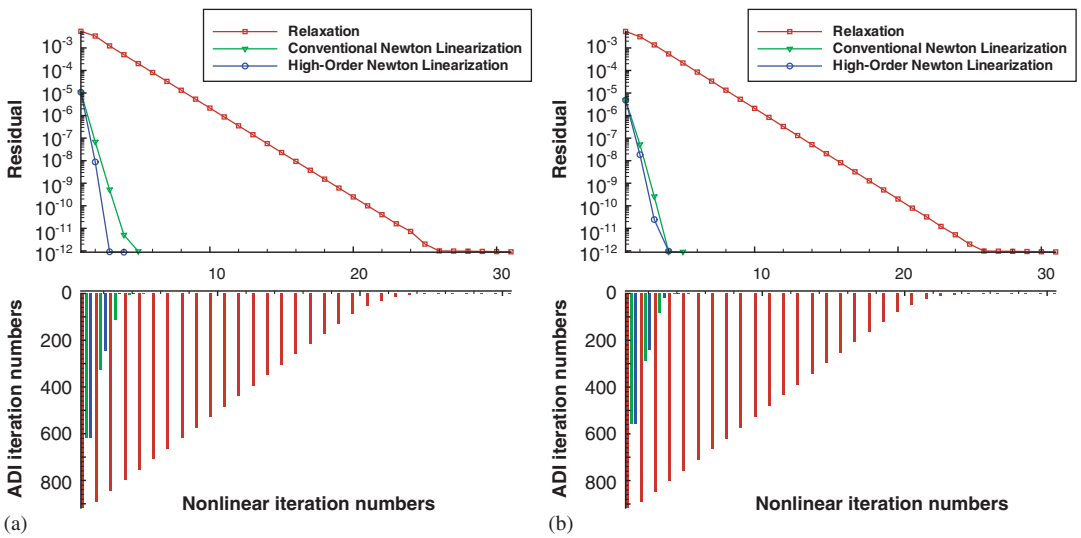

Figure 4. Comparison of the convergence histories and the needed numbers of inner iteration for three investigated linearization methods in the calculation of non-linear Navier–Stokes problem, which has the analytic solutions given in (55)–(57) at $Re = 1000$: (a) convergence histories for u; and (b) convergence histories for v.

Copyright © 2005 John Wiley & Sons, Ltd.

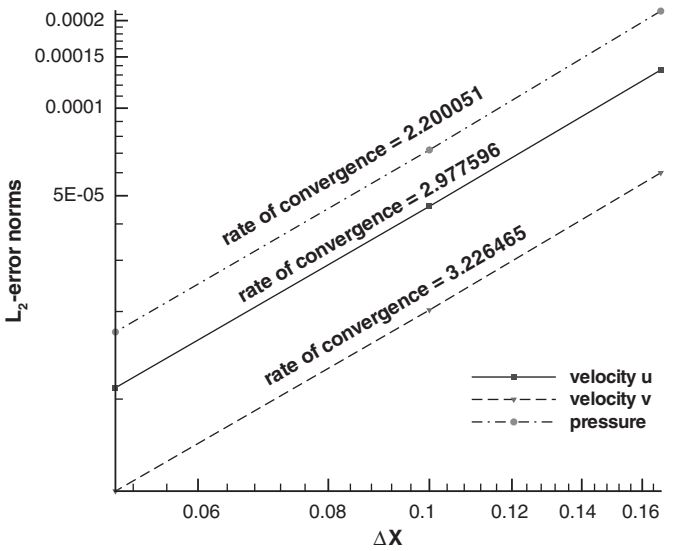

Figure 5. The computed rates of convergence for u, v and p.

Then the classical Kovasznay flow problem is investigated [10], which is amenable to the following analytic solutions:

$$u = 1 - \mathrm{e}^{\lambda x}\cos(2\pi y) \tag{58}$$

$$v = \frac{\lambda}{2\pi}\mathrm{e}^{\lambda x}\sin(2\pi y) \tag{59}$$

$$p = \tfrac{1}{2}(1 - e^{2\lambda x}) \tag{60}$$

where $\lambda = Re/2 - ((Re^2/4) + 4\pi^2)^{1/2}$. Numerical calculations have been carried out in a square which is covered with uniform grids. At $Re = 1000$, it is observed from Figure 5 the simulated high rates of convergence for pressure and velocity fields. The residual reduction plots, shown in Figure 6, indicate that the required non-linear iteration number has been considerably reduced. In Figure 6, the inner ADI iteration number in each outer non-linear iteration is also seen to have been considerably reduced, thus clearly demonstrating the advantage of using the proposed high-order Newton linearization method.

We also consider the analytic lid-driven cavity flow problem [11] in a square domain. The Navier–Stokes equations are solved subject to the following Dirichlet-type velocity boundary conditions at $x = 0,\ 1$ and $y = 0,\ 1$:

$$u = 8(x^4 - 2x^3 + x^2)(4y^3 - 2y) \tag{61}$$

$$v = -8(4x^3 - 6x^2 + 2x)(y^4 - y^2) \tag{62}$$

Copyright © 2005 John Wiley & Sons, Ltd.

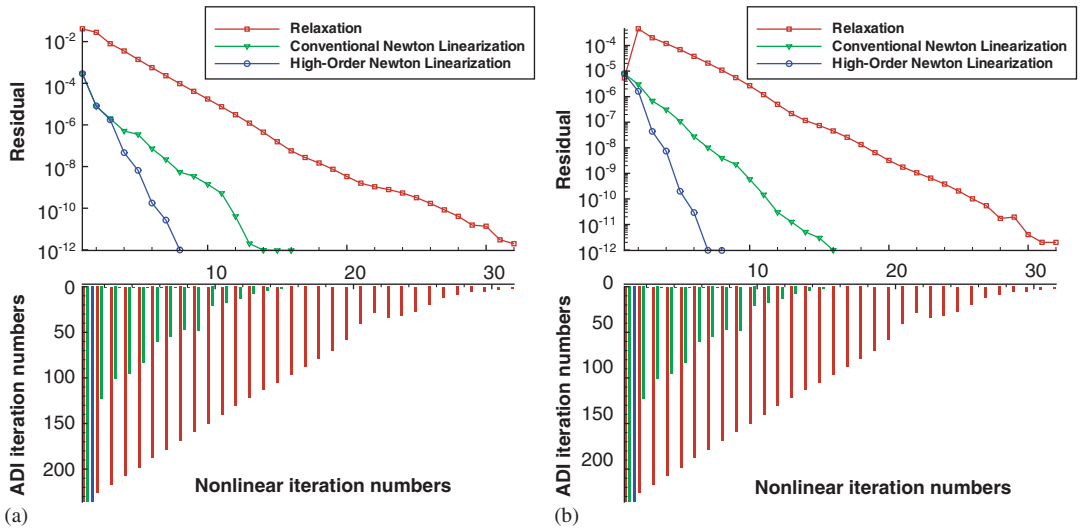

Figure 6. Comparison of the convergence histories and the needed numbers of inner iteration for three investigated linearization methods in the calculation of non-linear Navier–Stokes problem, which has the analytic solutions given in (58)–(60) at $Re = 1000$: (a) convergence histories for u; and (b) convergence histories for v.

If the body force $\mathbf{f} = (f_1, f_2)$ is given by

$$f_1 = 0 \tag{63}$$

$$f_2 = \frac{8}{Re}[24J_1(x) + 2I_1'(x)I_2''(y) + I_1'''(x)I_2(y)] + 64[J_3(x)J_4(y) - I_2(y)I_2'(y)J_2(x)] \tag{64}$$

the exact pressure takes the following form:

$$p = \frac{8}{Re}[J_1(x)I_2'''(y) + I_1'(x)I_2'(y)] + 64J_3(x)[I_2(y)I_2''(y) - (I_2'(y))^2] \tag{65}$$

where $I_1(x) = x^4 - 2x^3 + x^2$, $I_2(y) = y^4 - y^2$, $J_1(x) = \frac{1}{5}x^5 - \frac{1}{2}x^4 + \frac{1}{3}x^3$, $J_2(x) = -4x^6 + 12x^5 - 14x^4 + 8x^3 - 2x^2$, $J_3(x) = \frac{1}{2}(x^4 - 2x^3 + x^2)^2$ and $J_4(y) = -24y^5 + 8y^3 - 4y$.

For the case considered at Re=1000, our proposed Newton linearization method can render a much faster convergent solution (Figure 7). From Figure 8, it is seen that much of the CPU time has been saved. In addition, the proposed model can offer good accuracy without deteriorating convergence, as seen in the simulated streamlines, pressure contours and the mid-sectional profiles for u and v plotted in Figure 9.

In this paper the incompressible Navier–Stokes fluid flow in the cavity, which is subjected to a constant lid velocity u_{lid}, is also considered. The geometrical simplicity and physical complexity have made this problem attractive to benchmark the incompressible Navier–Stokes model. With L as the characteristic length, u_{lid} the characteristic velocity, the Reynolds number under investigation is taken as 5000.

Copyright © 2005 John Wiley & Sons, Ltd.

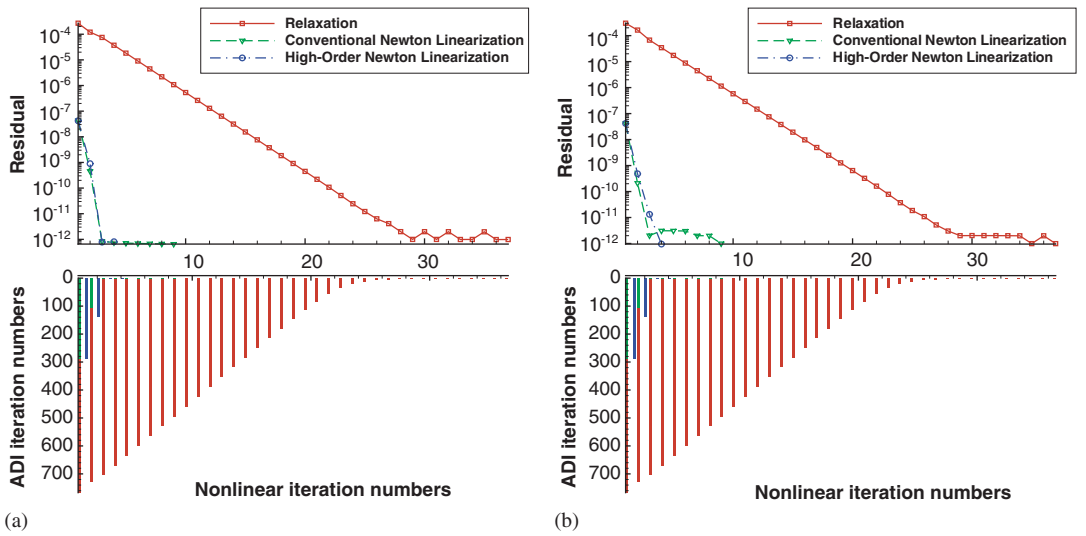

Figure 7. Comparison of the convergence histories and the needed numbers of inner iteration for three investigated linearization methods in the calculation of non-linear Navier–Stokes problem, which has the analytic solutions given in (61)–(65) at $Re = 1000$: (a) convergence histories for u; and (b) convergence histories for v.

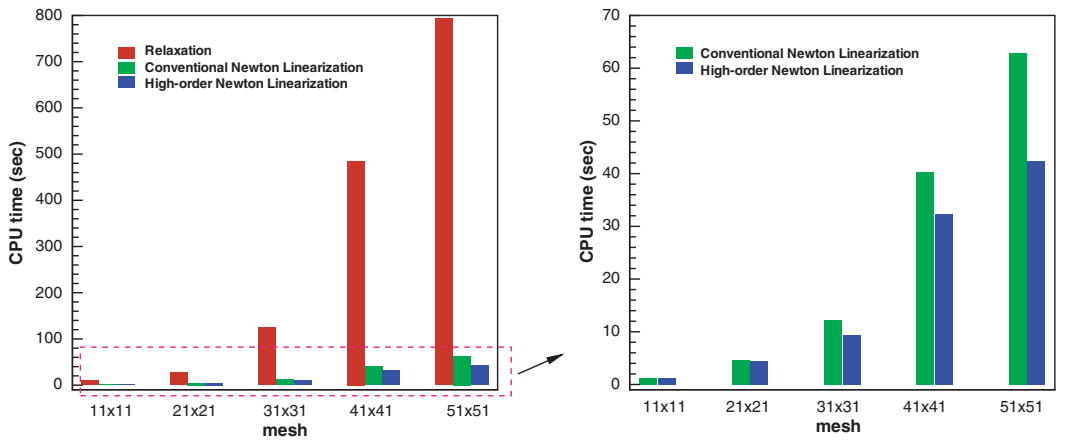

Figure 8. Comparison of the CPU time required to obtain the convergent solutions using the proposed nodally exact finite difference scheme.

As the mesh is sufficiently refined to obtain a grid-independent solution, we plot the mid-plane velocity profiles $u\ (0.5, y)$ and $v\ (x, 0.5)$ in Figure 10. These simulated profiles are compared with the steady-state benchmark solution obtained by Ghia [12]. The excellent agreement between the simulated and exact solutions and the much improved convergence histories seen in Figure 11 further confirm the suitability of employing the proposed linearization method.

Copyright © 2005 John Wiley & Sons, Ltd.

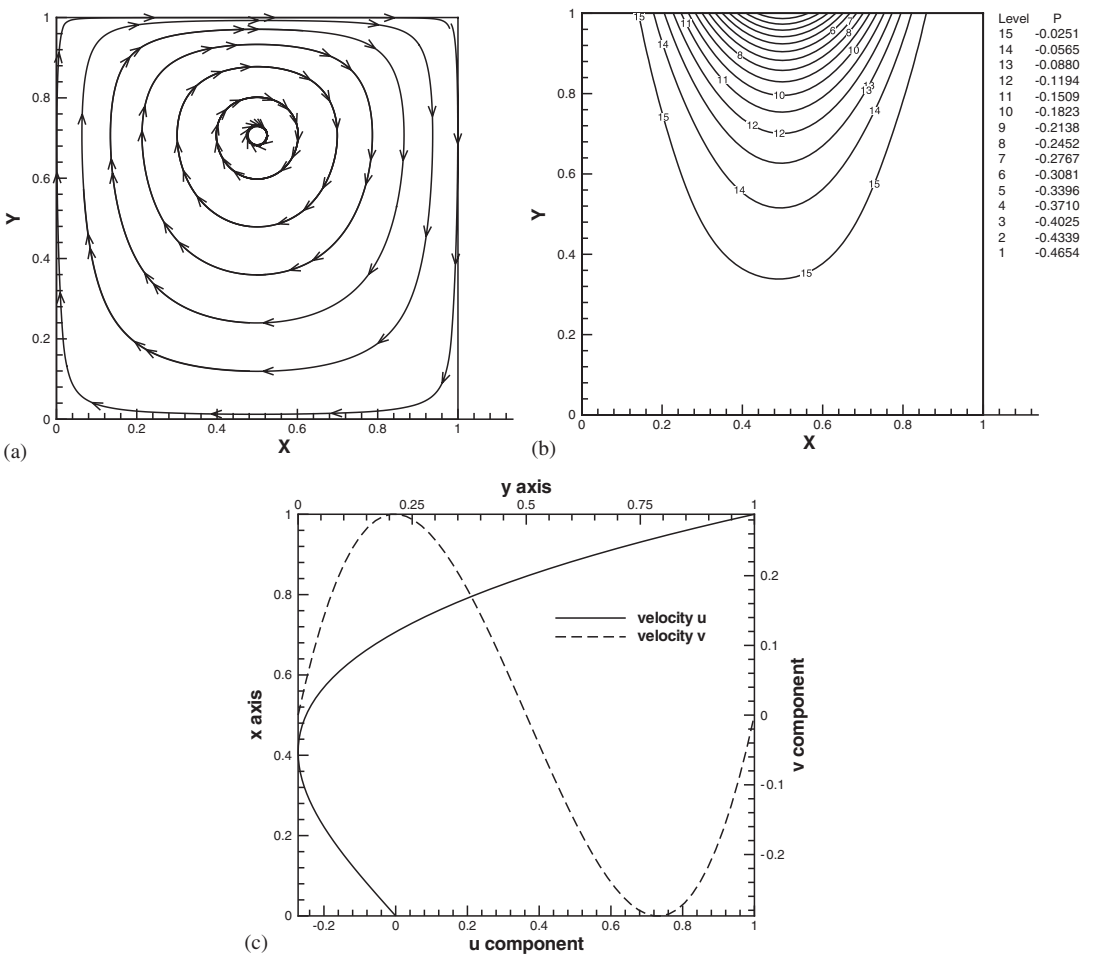

Figure 9. The simulated solutions for the analytic lid-driven cavity flow problem given in Section 4.2: (a) streamlines; (b) pressure contours; and (c) mid-sectional velocity profiles for u and v.

4.3. Unsteady Navier–Stokes equations

Encouraged by the above success in validating the proposed transport scheme for the steady-state problems, the transient Navier–Stokes equations are solved in this section. The method presented in Section 3 can be directly applied to the unsteady Navier–Stokes problem. Take the linearized model equation, namely $\phi_t + a\phi_x + b\phi_y - k(\phi_{xx} + \phi_{yy}) + c\phi = f$, as an example, we can approximate ϕ_t by $\phi_t = (\phi^{n+1} - \phi_n)/\Delta t$. The resulting equation involving only the spatial derivative terms takes the convection–diffusion–reaction form given by

$$\overline{a}\phi_x^{n+1} + \overline{b}\phi_y^{n+1} - \overline{k}(\phi_{xx}^{n+1} + \phi_{yy}^{n+1}) + \overline{c}\phi^{n+1} = \overline{f} \tag{66}$$

Copyright © 2005 John Wiley & Sons, Ltd.

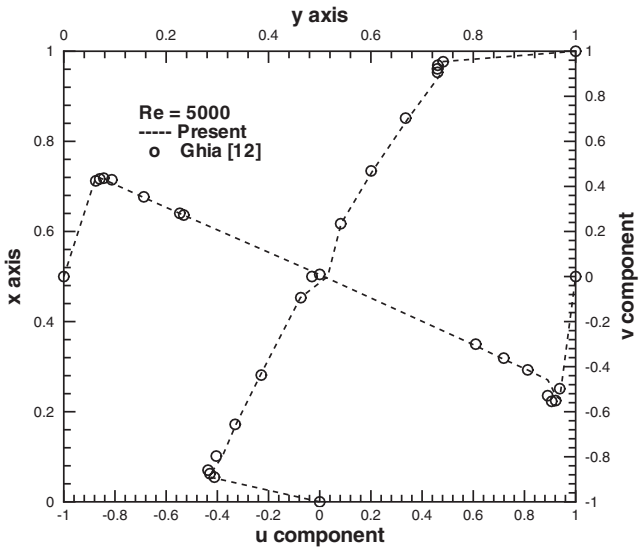

Figure 10. A comparison of the computed and Ghia's velocity profiles $u(x, 0.5)$ and $v(0.5, y)$.

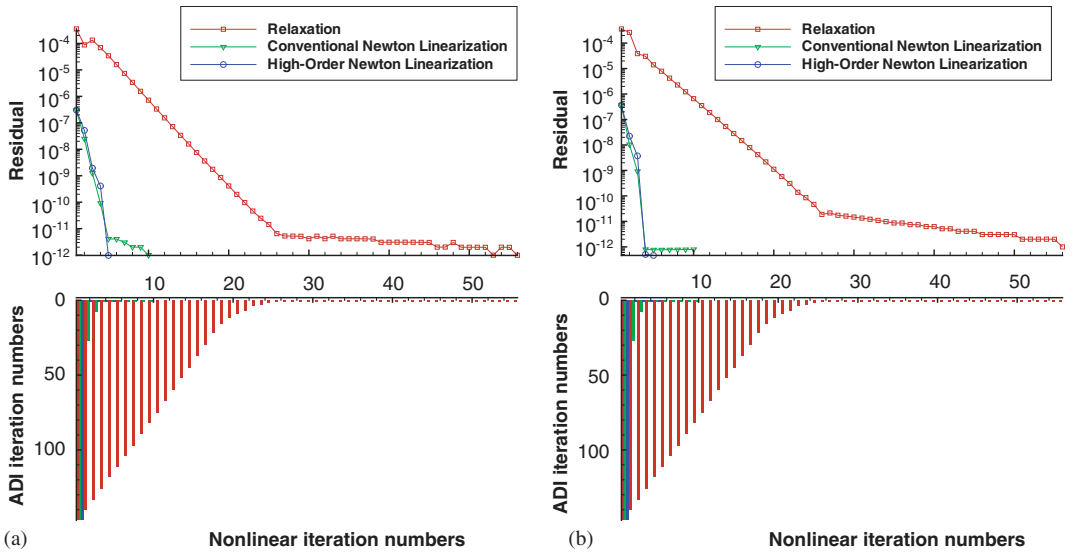

Figure 11. Comparison of the convergence histories and the needed numbers of inner iteration for three investigated linearization methods in the calculation lid-driven cavity flow problem at $Re = 5000$: (a) convergence histories for u; and (b) convergence histories for v.

The coefficients $(\overline{a}, \overline{b})$, $\overline{k}$, $\overline{c}$ and $\overline{f}$ are as follows: $(\overline{a}, \overline{b}) = (a\Delta t, b\Delta t)$, $\overline{k} = k\Delta t$, $\overline{c} = c\Delta t + 1$ and $\overline{f} = f\Delta t + \phi^n$. We can then apply the CDR scheme presented earlier to solve for the resulting Newton linearized momentum equations. In a square of unit length, the problem under

Copyright © 2005 John Wiley & Sons, Ltd.

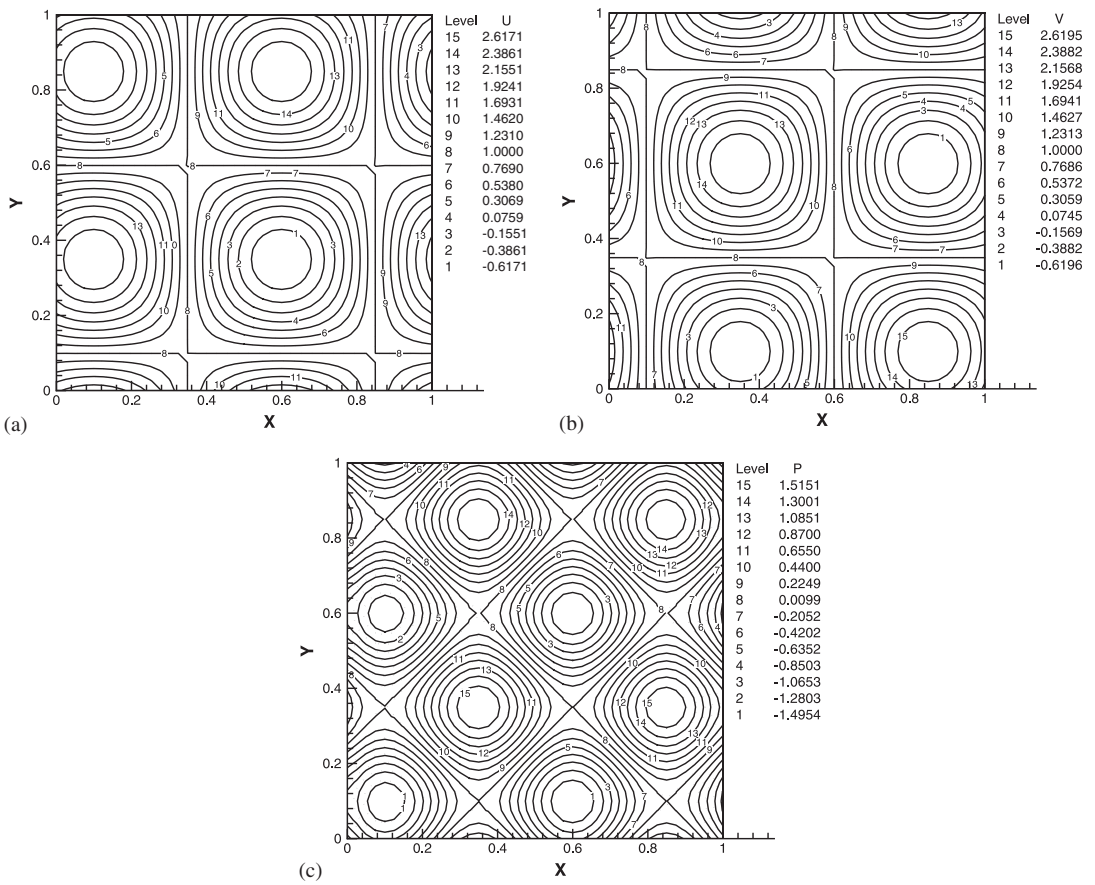

Figure 12. The simulated contours for the unsteady Navier–Stokes equations at $t = 1$: (a) u; (b) v; and (c) p.

investigation has the following exact solutions:

$$u = 1 + 2\cos[2\pi(x - t)]\sin[2\pi(y - t)]\mathrm{e}^{-8\pi^2 \nu t} \tag{67}$$

$$v = 1 - 2\sin[2\pi(x - t)]\cos[2\pi(y - t)]\mathrm{e}^{-8\pi^2 \nu t} \tag{68}$$

$$p = c_2 - \{\cos[4\pi(x - t)] + \cos[4\pi(y - t)]\}\mathrm{e}^{-16\pi^2 \nu t} \tag{69}$$

All the solutions are obtained in $0 \leqslant x, y \leqslant 1$. In Figure 12, we plot the simulated contours for u, v and p at $t = 1$ under the specified conditions $\nu = 10^{-3}$, $\Delta x = \Delta y = \frac{1}{20}$ and $\Delta t = 10^{-2}$. Computations are also performed on a range of mesh sizes $h = 1/2^n$, where $n = 4$, 5, 6, 7, at $\nu = 10^{-3}$ and $\Delta t = \frac{1}{100}$ for the sake of completeness. In view of the simulated L_2-norm errors in Figure 13, the proposed method can be applied to unsteady flow simulation. The save in

Copyright © 2005 John Wiley & Sons, Ltd.

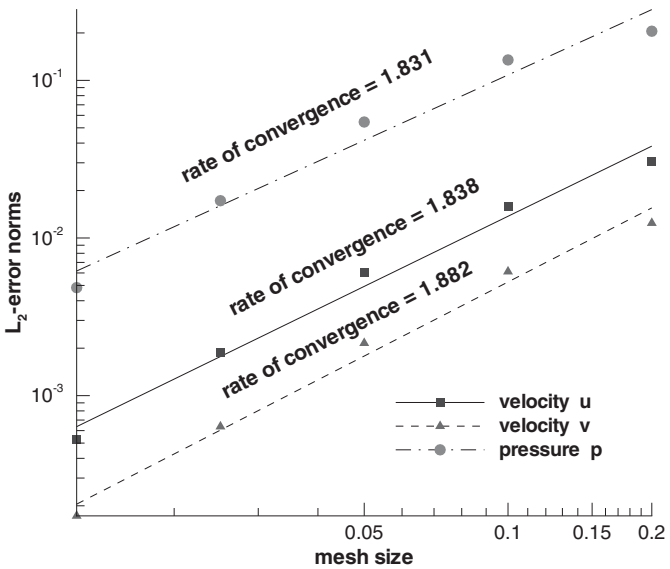

Figure 13. The simulated L_2-error norms and the resulting rates of convergence for the investigated unsteady Navier–Stokes equations based on the solutions obtained at $t = 1$.

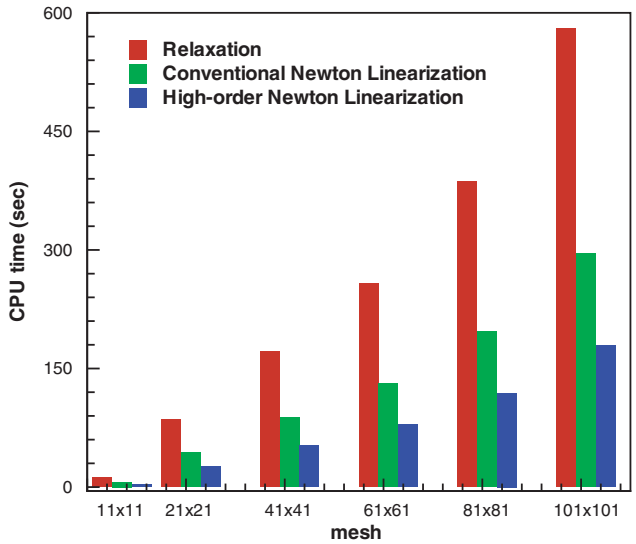

Figure 14. The CPU times for three chosen linearization methods applied to solve for the unsteady Navier–Stokes equations at $t = 1$ at different meshes $N \times N$, where $N = 10, 20, 40, 60, 80, 100$.

CPU time is clearly seen in Figure 14. The non-linear iteration numbers against time are also plotted in Figure 15 to show the advantage of applying the proposed high-order Newton linearization method to simulate the unsteady Navier–Stokes equations.

Copyright © 2005 John Wiley & Sons, Ltd.

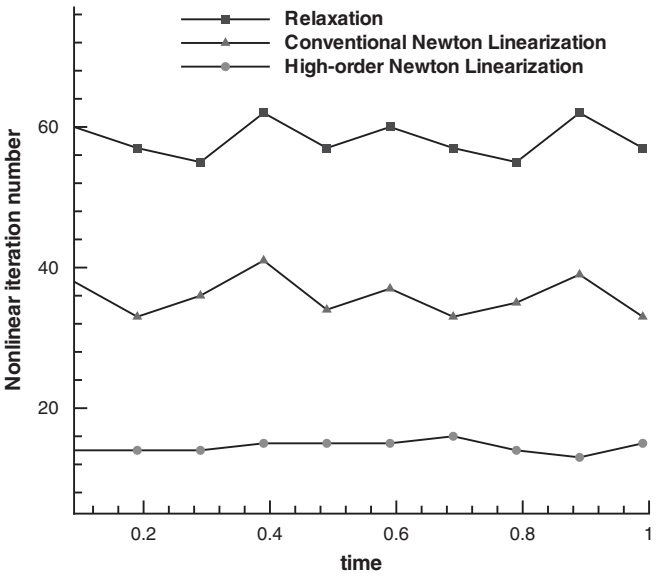

Figure 15. The plots of needed numbers of outer iteration against time ($0 \leqslant t \leqslant 1$) in the assessment of three linearization methods chosen for solving the unsteady Navier–Stokes equations at 40×40 mesh system.

5. CONCLUDING REMARKS

This study aims to accelerate non-linear convergence to the incompressible Navier–Stokes solutions using the proposed high-order Newton linearization method. For all the linearized equations, they are solved on a non-staggered grid system using the computationally very accurate and efficient CDR scheme. By virtue of the present assessment studies for steady as well as unsteady problems, much faster convergence to the convergent solutions can be obtained for all test problems.

ACKNOWLEDGEMENTS

We would like to acknowledge the financial support from National Science Council under NSC 92-2811-E-002-008. This work was partially accomplished in the course of the first author's sabbatical leave in University of Paris 6. Excellent facilities provided by Professors Olivier Pironneau and Yvon Maday are highly appreciated.

REFERENCES

1. Galpin PF, Raithby GD. Treatment of non-linearities in the numerical solution of the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1986; **6**:409–426.
2. Dennis JE Jr., Schmabel RB. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall: Englewood Cliffs, NJ, 1983.
3. Hadji S, Dhatt G. Asymptotic-Newton method for solving incompressible flows. *International Journal for Numerical Methods in Fluids* 1997; **25**:861–878.

Copyright © 2005 John Wiley & Sons, Ltd.

4. Dembo RS, Eisenstat SC, Steihung T. Inexact Newton methods. *International Journal for Numerical Methods in Fluids* 1982; **19**:400–408.
5. Peaceman DW, Rachford HH. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics* 1955; **3**:28–41.
6. Sheu TWH, Wang SK, Lin RK. An implicit scheme for solving the convection–diffusion–reaction equation in two dimensions. *Journal of Computational Physics* 2000; **164**:123–142.
7. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: New York, 1990.
8. Harlow FW, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surfaces. *The Physics of Fluids* 1965; **8**:2182–2189.
9. Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 1983; **21**:1525–1532.
10. Kovasznay LIG. Laminar flow behind a two-dimensional grid. *Proceedings of Cambridge Philosophical Society* 1948; **44**.
11. Shih TM, Tan CH, Hwang BC. Effects of grid staggering on numerical schemes. *International Journal for Numerical Methods in Fluids* 1989; **8**:193–212.
12. Ghia U, Ghia KN, Shin CT. High-*Re* solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.

Copyright © 2005 John Wiley & Sons, Ltd.