INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:703–717
Published online 20 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1133

An improved (9, 5) higher order compact scheme for the transient two-dimensional convection–diffusion equation

Jiten C. Kalita[1,*,†,§] and Puneet Chhabra[2,‡,¶]

[1]*Department of Mathematics, Indian Institute of Technology, Guwahati 781039, India*
[2]*Department of Mechanical Engineering, Indian Institute of Technology, Guwahati 781039, India*

SUMMARY

In the present study, we propose an implicit, unconditionally stable high order compact (HOC) finite difference scheme for the unsteady two-dimensional (2-D) convection–diffusion equations. The scheme is second-order accurate in time and fourth-order accurate in space. The stencil requires nine points at the nth and five points at the $(n+1)$th time level and is therefore termed a (9, 5) HOC scheme. It efficiently captures both transient and steady solutions of linear and nonlinear convection–diffusion equations with Dirichlet as well as Neumann boundary conditions. It is applied to a linear Gaussian pulse problem, a linear 2-D Schrödinger equation and the lid driven square cavity flow governed by the 2-D incompressible Navier–Stokes (N–S) equations. The results are presented and are compared with established numerical results. Excellent comparison is obtained in all the cases. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: 2-D; unsteady; convection–diffusion; HOC; N–S equations; accuracy

1. INTRODUCTION

The unsteady 2-D convection–diffusion equation with variable coefficients for a transport variable ϕ in some continuous domain with suitable boundary conditions is given by

$$a\frac{\partial \phi}{\partial t} - \nabla^2 \phi + c(x, y, t)\frac{\partial \phi}{\partial x} + d(x, y, t)\frac{\partial \phi}{\partial y} = g(x, y, t) \tag{1}$$

where a is a constant, c and d are the convection coefficients, and g is a forcing function. The equation represents convection–diffusion of many fluid variables such as mass, heat,

*Correspondence to: J. C. Kalita, Department of Mathematics, Indian Institute of Technology, Guwahati 781039, India.
†E-mail: jiten@iitg.ernet.in
‡E-mail: chhabra@iitg.ernet.in
§Assistant Professor.
¶Undergraduate Student.

Received 8 July 2005
Revised 6 October 2005
Accepted 20 October 2005

energy, vorticity, etc. With proper choice of a, c, d and g, it can also represent the complete Navier–Stokes (N–S) equations. As such, it has been an area of intense research in past few years [1–11].

Various schemes have been developed for problems based on Equation (1). Quite recently, HOC finite difference methods [2, 3, 5–9, 11–15] have become quite popular as against the other lower order accurate schemes which require high mesh refinement and hence are computationally inefficient. On the other hand, the higher order accuracy of the HOC methods combined with the compactness of the difference stencil yields highly accurate numerical solutions on relatively coarser grids with greater computational efficiency. Of these, Kalita *et al.* [5] developed a class of HOC schemes for the unsteady 2-D convection–diffusion equation with variable coefficients which requires either nine or five points at the nth and $(n+1)$th time levels. Accordingly, they are termed as $(9,5)$, $(9,9)$ or $(5,9)$ schemes.

Among these schemes, the $(9,5)$ scheme in spite of having fourth-order spatial accuracy and smallest of band-widths of the resultant coefficient matrix amongst all the three schemes, suffered from a low first-order temporal accuracy and conditional stability. On the other hand, the $(9,9)$ scheme was not only temporally second-order accurate but also unconditionally stable. Thus it was able to accurately capture the transient solutions to unsteady 2-D convection–diffusion equation. But a relatively large matrix band-width of the coefficient matrix for this scheme resulted in an increased number of arithmetic operations.

The present work proposes an improved form of $(9,5)$ scheme which is fourth-order accurate in space and second-order accurate in time, is unconditionally stable and still uses the same $(9,5)$ difference stencil of five points at the $(n+1)$th time level and nine points at the nth time level. In the process, the work also intends to extend the idea of the HOC scheme to a more general 2-D second-order partial differential equation given by

$$a\frac{\partial\phi}{\partial t} - \nabla^2\phi + c(x,y,t)\frac{\partial\phi}{\partial x} + d(x,y,t)\frac{\partial\phi}{\partial y} + \varepsilon(x,y,t)\phi = g(x,y,t) \tag{2}$$

where the additional coefficient ε may be termed as the potential function. In addition to the already-mentioned utilities of Equation (1), Equation (2) can also be used to represent transient reaction–diffusion equation. With constant $a = \mathrm{i}$ where $\mathrm{i} = \sqrt{-1}$, it can also be represented as a linear 2D Schrödinger equation.

The proposed scheme is able to solve very accurately and efficiently the unsteady 2-D convection–diffusion problems including 2-D incompressible N–S equations and the linear 2-D Schrödinger equation. The merit of the present scheme for transient problems lies on its temporal accuracy coupled with the CPU time-wise efficiency owing to a more compact difference stencil. To test the robustness, accuracy and efficiency of the scheme, it is applied to three pertinent test cases for which the numerical and/or analytical results are available. We also carry out error analysis wherever analytical solutions are available. Results are compared with those obtained by the $(9,9)$ and $(9,5)$ scheme of Reference [5], and well established results from various other schemes.

The paper has been arranged in four sections. Section 2 deals with discretization and issues related to it, Section 3 with the numerical test cases and finally, Section 4 summarizes the whole work.

2. DISCRETIZATION AND RELATED ISSUES

Assuming the problem domain to be rectangular and constructing on it a uniform rectangular mesh of steps h and k in the x- and y-directions respectively, the standard forward-time centred space (FTCS) approximation to Equation (2) at the (i,j)th node is given by

$$(a\delta_t^+ - \delta_x^2 - \delta_y^2 + c\delta_x + d\delta_y + \varepsilon)\phi_{ij} - \zeta_{ij} = g_{ij} \tag{3}$$

where ϕ_{ij} denotes $\phi(x_i, y_j)$; δ_x, δ_x^2 and δ_y, δ_y^2 are the first and second-order central difference operators along x- and y-directions respectively, and δ_t^+ is the first-order forward difference operator for time. The truncation error ζ_{ij} with a uniform time step Δt is given by

$$\zeta_{ij} = \left[a\frac{\Delta t}{2}\frac{\partial^2\phi}{\partial t^2} + \frac{h^2}{12}\left(2c\frac{\partial^3\phi}{\partial x^3} - \frac{\partial^4\phi}{\partial x^4}\right) + \frac{k^2}{12}\left(2d\frac{\partial^3\phi}{\partial y^3} - \frac{\partial^4\phi}{\partial y^4}\right)\right]_{ij} + O(\Delta t^2, h^4, k^4) \tag{4}$$

To obtain a second-order time accurate and spatially fourth-order compact formulation for (3), each of the derivatives of the leading term of (4) are compactly approximated [15, 16] to $O(\Delta t, h^2, k^2)$. In order to accomplish this, the original PDE of Equation (2) is treated as an auxiliary relation that can be differentiated to yield expressions for higher derivatives. For example, using forward temporal difference for the transport variable ϕ and backward difference for c, d, ε and g (see the note at the end of the section), the derivative in the first term on the right-hand side of (4) can be written as

$$a\left.\frac{\partial^2\phi}{\partial t^2}\right|_{ij} = (\delta_x^2 + \delta_y^2 - c_{ij}\delta_x - d_{ij}\delta_y - \varepsilon_{ij})\delta_t^+\phi_{ij} - (\delta_t^- c_{ij}\delta_x + \delta_t^- d_{ij}\delta_y + \delta_t^-\varepsilon_{ij})\phi_{ij}$$
$$+ \delta_t^- g_{ij} + O(\Delta t, h^2, k^2) \tag{5}$$

where δ_t^- is the first-order temporal backward difference operator. Similar approximations can be constructed for the spatial derivatives as well. Thus replacing the derivatives in Equation (4) with approximations such as Equation (5) and subsequent substitution for ζ_{ij} in Equation (3) yields an $O(\Delta t^2, h^4, k^4)$ approximation for Equation (2) on a (9,5) stencil (see Figure 1) as

$$a\left[1 + \left(\frac{h^2}{12} - \frac{\Delta t}{2a}\right)(\delta_x^2 - c_{ij}\delta_x) + \left(\frac{k^2}{12} - \frac{\Delta t}{2a}\right)(\delta_y^2 - d_{ij}\delta_y) + \frac{\Delta t}{2a}\varepsilon_{ij}\right]\delta_t^+\phi_{ij}$$
$$+ (-\alpha_{ij}\delta_x^2 - \beta_{ij}\delta_y^2 + C_{ij}\delta_x + D_{ij}\delta_y + E_{ij})\phi_{ij}$$
$$- \frac{(h^2 + k^2)}{12}(\delta_x^2\delta_y^2 - c_{ij}\delta_x\delta_y^2 - d_{ij}\delta_x^2\delta_y - \gamma_{ij}\delta_x\delta_y)\phi_{ij} = G_{ij} \tag{6}$$

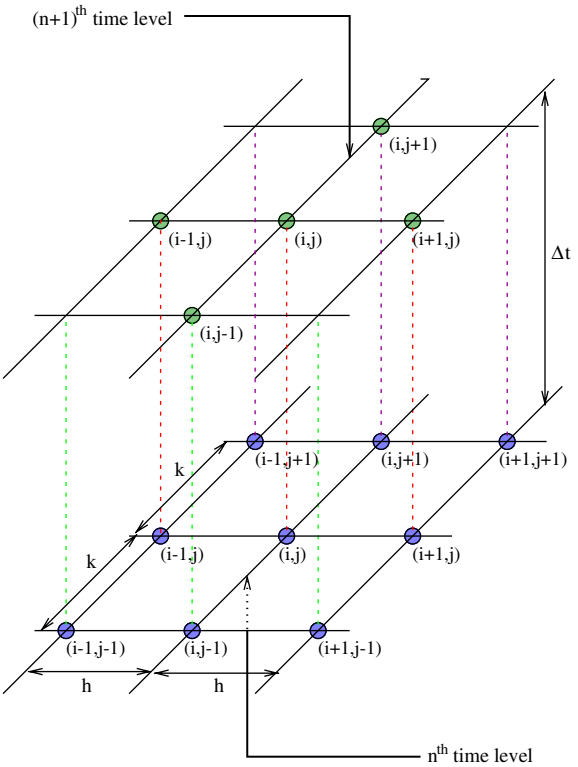


Figure 1. Unsteady $(9,5)$ HOC stencil.

where the coefficients α_{ij}, β_{ij}, C_{ij}, D_{ij}, E_{ij}, G_{ij} and γ_{ij} are as follows:

$$\alpha_{ij} = 1 + \frac{h^2}{12}(c_{ij}^2 - 2\delta_x c_{ij} - \varepsilon_{ij})$$

$$\beta_{ij} = 1 + \frac{k^2}{12}(d_{ij}^2 - 2\delta_y d_{ij} - \varepsilon_{ij})$$

$$C_{ij} = \left[1 + \frac{h^2}{12}(\delta_x^2 - c_{ij}\delta_x) + \frac{k^2}{12}(\delta_y^2 - d_{ij}\delta_y) + \frac{\Delta t}{2}\delta_t^-\right] c_{ij} + \frac{h^2}{12}(2\delta_x - c_{ij})\varepsilon_{ij}$$

$$D_{ij} = \left[1 + \frac{h^2}{12}(\delta_x^2 - c_{ij}\delta_x) + \frac{k^2}{12}(\delta_y^2 - d_{ij}\delta_y) + \frac{\Delta t}{2}\delta_t^-\right] d_{ij} + \frac{k^2}{12}(2\delta_y - d_{ij})\varepsilon_{ij}$$

$$E_{ij} = \left[1 + \frac{h^2}{12}(\delta_x^2 - c_{ij}\delta_x) + \frac{k^2}{12}(\delta_y^2 - d_{ij}\delta_y) + \frac{\Delta t}{2}\delta_t^-\right] \varepsilon_{ij}$$

$$G_{ij} = \left[1 + \frac{h^2}{12}(\delta_x^2 - c_{ij}\delta_x) + \frac{k^2}{12}(\delta_y^2 - d_{ij}\delta_y) - \frac{\Delta t}{2}\delta_t^-\right] g_{ij}$$

$$\gamma_{ij} = \frac{2}{h^2 + k^2}(h^2\delta_x d_{ij} - k^2\delta_y c_{ij}) - c_{ij}d_{ij}$$

With this (6) yields an implicit finite difference scheme of accuracy $O(\Delta t^2, h^4, k^4)$. It may be mentioned that in the earlier HOC schemes developed in Reference [5], time-discretization was performed by considering the time derivative as a part of the source function of the steady-state convection–diffusion equation. Then a weighted time averaging strategy was employed to improve the temporal accuracy, which roped in some mixed space-derivative terms. As such, the corner points also came into the picture resulting in a nine point stencil at the $(n+1)$th time level. On the other hand, from Equations (4) and (5), we see that process of improving the temporal accuracy for the present scheme involves no mixed space-derivatives. Therefore, we are left with to deal with only five points at the $(n+1)$th time level. As the stencil requires only the (i,j)th and its neighbouring four points (Figure 1) at the $(n+1)$th time level, this scheme has also the potential to be extended as an alternate-direction-implicit (ADI) scheme which will be discussed in a separate paper. Equation (6) can now be evaluated in terms of system of equations that can be written in a matrix form as

$$A\boldsymbol{\phi}^{(n+1)} = f(\boldsymbol{\phi}^{(n)}) \tag{7}$$

where the coefficient matrix A is an asymmetric pentadiagonal sparse matrix. To solve the system of Equation (7) biconjugate gradient stabilized method (BiCGStab) [17, 18] has been employed without any preconditioning.

Note: If the closed forms of c, d, v and g are not known as in the case of N–S equations, it is easier to approximate their time-derivatives with backward (than forward) difference with the current and previous time-level values already known. With some suitable initialization strategy, this can be easily accomplished (see Section 3.3).

3. NUMERICAL TEST CASES

In order to study the validity and effectiveness of the proposed scheme, it is applied to three unsteady 2-D problems. These are: (i) convection–diffusion of a Gaussian pulse; (ii) the linear 2-D Schrödinger equation; and (iii) the lid-driven cavity flow. Problems (i) and (ii) have analytical solutions, so Dirichlet boundary conditions are used for them while for the cavity problem, both Dirichlet and Neumann boundary conditions are applied. All computations are carried on a PC with Pentium 4 processor and 512 mB RAM.

3.1. Problem 1

Consider Equation (2) with $\varepsilon = g = 0$ and constant convective coefficients in the square $0 \leqslant x, y \leqslant 2$ with initial condition

$$\phi(x, y, 0) = \exp[-a((x-0.5)^2 + (y-0.5)^2)]$$

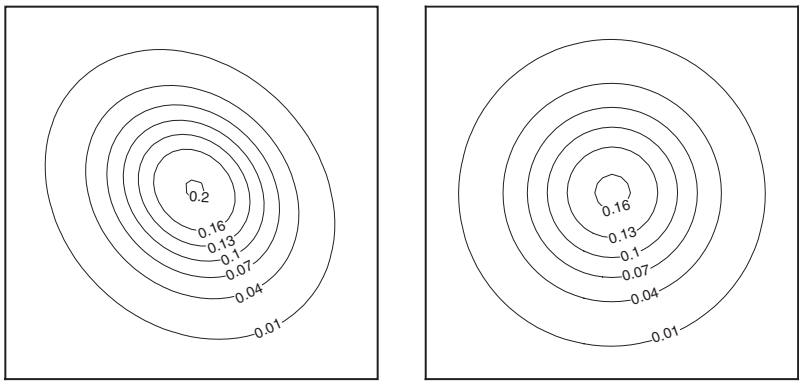
An analytical solution to this problem is

$$\phi(x, y, t) = \frac{1}{(4t+1)} \exp\left[-\frac{a}{(4t+1)}((x-ct-0.5)^2 + (y-dt-0.5)^2)\right]$$

The initial condition is a Gaussian pulse centred at $(0.5, 0.5)$ having height 1. The boundary conditions have been taken from the analytical solution. For the sake of comparison of the

Table I. Error and CPU time ratios for problem 1 at $t = 1.25$ with $\Delta t = 0.0125$ and $h = k = 0.025$.

Method	Average|error|	Maximum|error|	CPU time ratios with FTCS
FTCS	3.94×10^{-3}	1.21×10^{-1}	1
Upwind	2.65×10^{-3}	6.63×10^{-2}	2
Noye and Tan [7]	1.43×10^{-5}	4.84×10^{-4}	447.12
(9, 5) [5]	1.49×10^{-3}	3.74×10^{-2}	9.78
(5, 9) [5]	1.02×10^{-3}	2.25×10^{-2}	7.03
(9, 9) [5]	5.24×10^{-5}	1.19×10^{-3}	5.26
Present (9, 5)	7.77×10^{-5}	1.69×10^{-3}	3.11



(a) (b)

Figure 2. Contour plots of the pulse in the subregion $1 \leqslant x,\ y \leqslant 2$ at $t = 1.25$ for: (a) (9, 5) scheme in Reference [5]; and (b) the present (9, 5) scheme with $\Delta t = 0.00625$.

results with those of References [5, 7], we choose $a = 100$, $c = d = 80$ and $t = 1.25$ at which time the pulse moves to a position centred at (1.5, 1.5), now with height $\frac{1}{6}$.

The average and the maximum absolute errors of different schemes at time $t = 1.25$ captured with $\Delta t = 0.0125$ and $h = k = 0.025$ including the present one along with the ratio of their CPU times with that of the explicit FTCS scheme have been presented in Table I. It can be seen from the table that the accuracy of the present (9, 5) scheme is comparable with the (9, 9) [5] scheme and it is far superior than the earlier (9, 5) [5] scheme. This is also exemplified in Figure 2 where the pulse height contours are presented for the old and present (9, 5) scheme at $t = 1.25$ with $\Delta t = 0.00625$ and $h = k = 0.025$. Whereas the present scheme yields an accurate solution, the earlier (9, 5) scheme suffers from the presence of in-built numerical anti-diffusion resulting in the generation of elliptic contours (with pulse height 0.202492) which can be seen in Figure 2(a). On the other hand, the present (9, 5) scheme gives a remarkably accurate solution (with pulse height 0.166325 against the analytical value of 0.166667), free from diffusion or anti-diffusion, yielding a pulse which is indistinguishable from the exact one as seen from Figure 2(b). It will be worthwhile to compare the CPU time-wise efficiency of the present (9, 5) scheme with the (9, 9) [5] scheme and the nine point scheme of Noye and Tan [6, 7] as all of them are implicit and second-order accurate in

Table II. Convergence rate of the present scheme: maximum |error| comparisons on different grid sizes for problem 1.

Grid	Maximum|error| (at $t=0.625$)	Rate	Maximum|error| (at $t=1.25$)	Rate
41×41	8.586×10^{-3}		3.512×10^{-3}	
		3.777		3.848
81×81	6.261×10^{-4}		2.438×10^{-4}	
		4.001		3.952
161×161	3.910×10^{-5}		1.575×10^{-5}	

time. The CPU time ratio of the Noye and Tan scheme and (9,9) [5] scheme to that of FTCS scheme is 447.12 and 5.257, respectively. This ratio for the present scheme is 3.11, which clearly shows the superior CPU time-wise efficiency of the present scheme. This is because at the $(n+1)$th time level the present scheme uses a smaller stencil of five points instead of nine. In Table II, we present the maximum errors at two time stations $t=0.625$ and 1.25 captured with a time step $\Delta t=6.25\times 10^{-5}$ on three different grids. It is seen from the table that the error decays with $O(h^4)$ as expected.

3.2. Problem 2

Equation (2) is now considered here with $c=d=0$ and $a=\mathrm{i}\ (=\sqrt{-1})$ and potential function $\varepsilon=\varepsilon(x,y)=-1+(1/x^2)+(1/y^2)$ [19] in the region $0\leqslant x,\ y\leqslant 1$ such that it leads to the unsteady 2-D Schrödinger equation for wave function ϕ given by

$$\mathrm{i}\frac{\partial\phi}{\partial t}+\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}+\varepsilon(x,y)\phi=0 \tag{8}$$

The analytical solution of the equation is

$$\phi(x,y,t)=x^2y^2\mathrm{e}^{\mathrm{i}t}$$

The initial and the boundary conditions can be found from the analytical solution as

$$\phi(x,y,0)=x^2y^2 \text{ and}$$

$$\phi(x,0,t)=\phi(0,y,t)=0,\quad \phi(x,1,t)=x^2\mathrm{e}^{\mathrm{i}t},\quad \phi(1,y,t)=y^2\mathrm{e}^{\mathrm{i}t}$$

Since in this case the closed form of $\varepsilon(x,y)$ is known, we replace the first and second-order spatial difference operators of $\varepsilon_{i,j}$ in Equation (6) by the analytical form of first and second-order partial derivatives of $\varepsilon(x,y)$. For example $\delta_x\varepsilon_{i,j}=-4/x_i^3,\ \delta_x^2\varepsilon_{i,j}=12/x_i^4$, etc.

To compare our results with that of Reference [19], we first obtain result for ϕ at $t=1.0$ with $h=k=0.1$ and $\Delta t=0.00007$. Comparison of errors from different schemes with this time step is given in Table III. Here the second and third columns respectively represents the real and imaginary parts of the errors from the FTCS scheme, column four and five represent the same for Noye–Hayman [8] (5,5) implicit scheme, columns six and seven the (3,3) Paceman–Ratcford ADI [20–22] scheme and the last four columns represent the same

Table III. Comparison of errors at diagonal locations with different schemes at time $t = 1.0$ with $h = k = 0.1$ and $\Delta t = 0.00007$ for problem 2.

Location	Absolute error									
	(1,5 Explicit)		(5,5 N–H)		(3,3 P–R ADI)		(9,9) [5]		Present(9,5)	
(x, y)	Real	Imaginary	Real	Imaginary	Real	Imaginary	Real	Imaginary	Real	Imaginary
(0.1, 0.1)	1.3e − 1	7.4e − 2	5.9e − 2	2.3e − 3	1.1e − 2	5.4e − 3	4.65e − 12	2.59e − 11	1.04e − 11	4.27e − 12
(0.2, 0.2)	5.4e − 2	4.0e − 2	5.4e − 3	3.5e − 4	4.3e − 3	1.8e − 3	7.81e − 11	2.54e − 10	2.34e − 10	9.59e − 11
(0.3, 0.3)	2.0e − 2	2.6e − 2	7.6e − 4	2.6e − 4	7.9e − 4	1.9e − 4	2.33e − 10	7.90e − 10	7.40e − 10	2.00e − 10
(0.4, 0.4)	2.1e − 2	2.4e − 2	9.3e − 4	8.7e − 3	7.0e − 4	7.2e − 4	1.51e − 10	1.74e − 09	1.68e − 09	1.57e − 10
(0.5, 0.5)	3.7e − 2	2.3e − 2	4.5e − 4	6.9e − 3	2.7e − 3	1.5e − 3	1.12e − 09	2.63e − 09	2.52e − 09	1.07e − 09
(0.6, 0.6)	5.4e − 2	1.5e − 2	5.7e − 3	4.4e − 4	4.6e − 3	4.1e − 3	1.51e − 09	3.11e − 09	3.00e − 09	1.45e − 09
(0.7, 0.7)	5.3e − 2	5.9e − 3	6.6e − 3	5.9e − 4	4.2e − 3	6.4e − 3	1.31e − 09	3.01e − 09	2.89e − 09	1.30e − 09
(0.8, 0.8)	3.3e − 2	2.0e − 2	6.2e − 3	5.2e − 3	3.7e − 3	5.0e − 3	1.04e − 09	2.23e − 09	2.15e − 09	1.04e − 09
(0.9, 0.9)	1.0e − 2	7.6e − 3	3.9e − 3	4.5e − 3	1.5e − 3	2.2e − 3	5.16e − 10	1.01e − 09	9.53e − 10	4.99e − 10

Table IV. Comparison of the analytical and the numerical solutions with the present scheme at diagonal locations along with the absolute errors at time $t = 1.0$ with $h = k = 0.1$ and $\Delta t = 0.05$ for problem 2.

	Exact		Numerical (present)		Absolute error	
(x, y)	Real	Imaginary	Real	Imaginary	Real	Imaginary
(0.1 0.1)	5.403023e − 5	8.414710e − 5	5.416079e − 5	8.354585e − 5	1.305584e − 7	6.012446e − 7
(0.2 0.2)	8.644837e − 4	1.346354e − 3	8.653134e − 4	1.346018e − 3	8.297390e − 7	3.358712e − 7
(0.3 0.3)	4.376449e − 3	6.815915e − 3	4.377099e − 3	6.816368e − 3	6.503057e − 7	4.533907e − 7
(0.4 0.4)	1.383174e − 2	2.154166e − 2	1.383186e − 2	2.154218e − 2	1.250721e − 7	5.262246e − 7
(0.5 0.5)	3.376889e − 2	5.259194e − 2	3.376922e − 2	5.259211e − 2	3.280010e − 7	1.750155e − 7
(0.6 0.6)	7.002318e − 2	1.090546e − 1	7.002407e − 2	1.090544e − 1	8.954658e − 7	2.174041e − 7
(0.7 0.7)	1.297266e − 1	2.020372e − 1	1.297277e − 1	2.020365e − 1	1.142318e − 6	6.420636e − 7
(0.8 0.8)	2.213078e − 1	3.446665e − 1	2.213087e − 1	3.446655e − 1	8.835922e − 7	1.010154e − 6
(0.9 0.9)	3.544923e − 1	5.520891e − 1	3.544926e − 1	5.520883e − 1	2.526085e − 7	8.366844e − 7

from (9,9) [5] and the present (9,5) scheme. The earlier (9,5) scheme is not stable for this time step. From the table it is clear that the (9,9) and present (9,5) schemes are far superior to the others. But when we bring the CPU times in the picture it is found that the present (9,5) scheme takes only two-thirds the time taken by the (9,9) scheme. This clearly shows the higher computational efficiency of the present scheme. The fact that the present scheme is able to give a remarkably accurate solution even with time steps as high as $\Delta t = 0.05$ and $h = k = 0.1$, as shown in Table IV further shows the robustness of the scheme.

3.3. *Problem 3: the lid-driven square cavity flow*

The last test case is the problem of 2-D lid-driven square cavity flow which is extensively used as a benchmark for code validation of the incompressible N–S equations. In nondimensional form, they can be written as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{9}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\nabla^2 u \tag{10}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\nabla^2 v \tag{11}$$

Here u and v are the velocities along x- and y-directions, Re is the Reynolds number and p is the pressure. Introducing streamfunction ψ and vorticity ω, the above equations can be written as

$$\omega_t - \frac{1}{Re}(\omega_{xx} + \omega_{yy}) + (u\omega_x + v\omega_y) = 0 \tag{12}$$

$$\psi_{xx} + \psi_{yy} = -\omega(x, y) \tag{13}$$

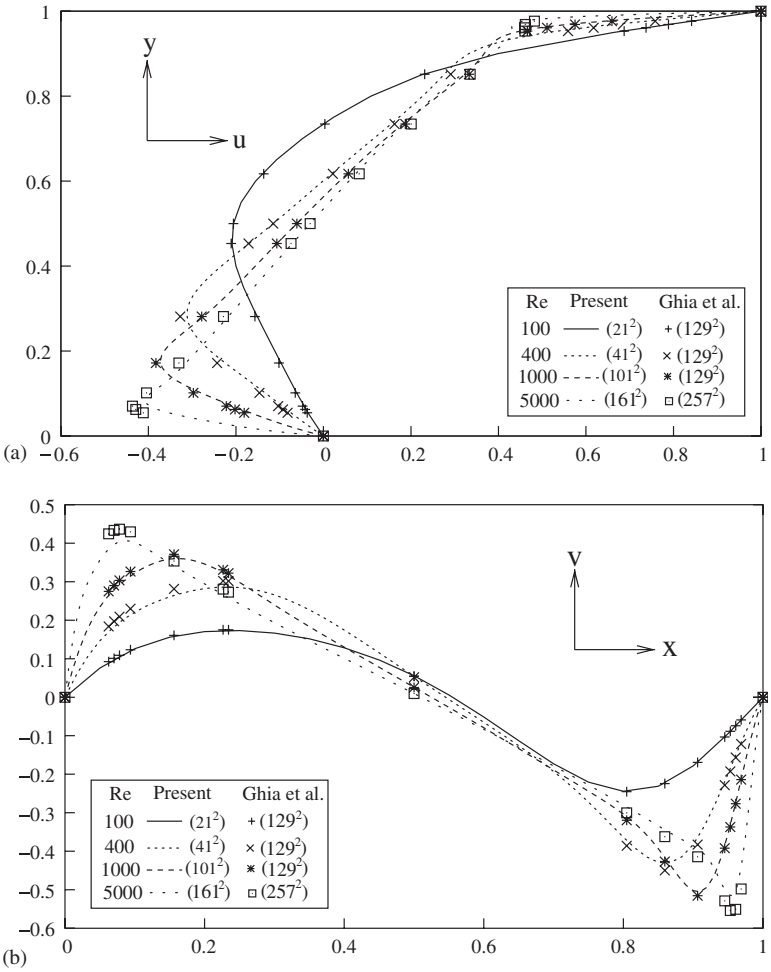


Figure 3. Comparisons of steady state: (a) horizontal velocity along the vertical centreline; and (b) vertical velocity along the horizontal centreline for the lid-driven square cavity flows from $Re = 100$ to 5000.

with

$$u = \psi_y \quad \text{and} \quad v = -\psi_x \tag{14}$$

and

$$\omega = u_y - v_x \tag{15}$$

We use this stream-function-vorticity ($\psi - \omega$) formulation and a time-marching strategy to reach the steady state. The cavity is defined as the square $0 \leqslant x,\ y \leqslant 1$. The fluid motion is generated by the sliding motion of the top wall of the cavity ($y = 1$) in its own plane from left to right. Boundary conditions on the top wall are given as $u = 1$, $v = 0$. On all other walls

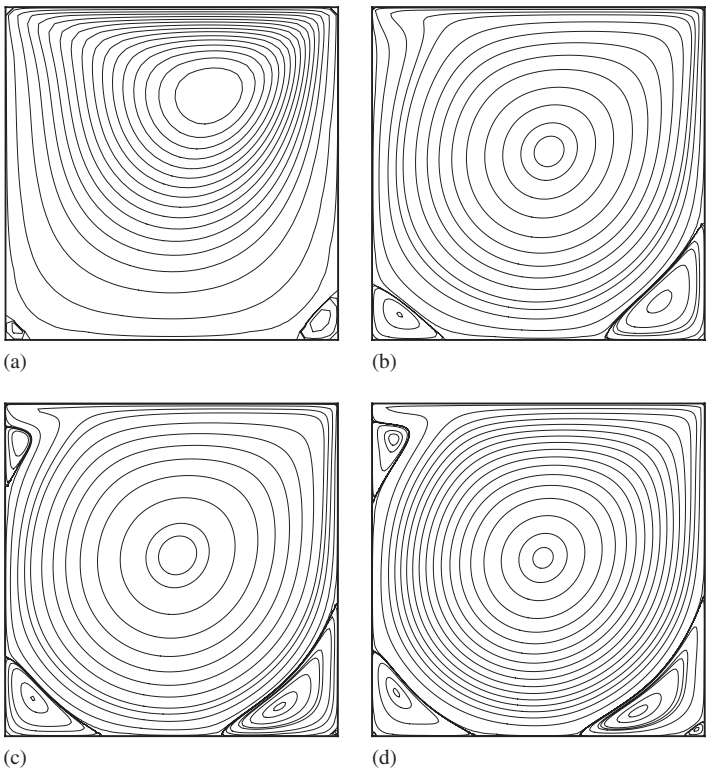


Figure 4. Steady-state streamfunction contours for the lid-driven square cavity flows for: (a) $Re = 100$ (41×41); (b) $Re = 1000$ (101×101); (c) $Re = 3200$ (121×121); and (d) $Re = 5000$ (161×161).

of the cavity the velocities are zero ($u = v = 0$). Further the streamfunction values on all four walls are zero ($\psi = 0$). A transient fourth-order compact formula [5] has been used for the Neumann boundary conditions for vorticity at the walls.

To solve the N–S equations using the proposed scheme, we employ an outer–inner iteration procedure. In a typical outer temporal cycle, we discretize Equation (12) using (6) with $a = Re$, $c = Re\,u$, $d = Re\,v$ and $g = \varepsilon = 0$. Once ω is obtained, we compute ψ by discretizing (13) with the steady-state version of (6). Thus on setting $\phi = \psi$, $c = d = \varepsilon = 0 = \delta_t^+ \phi_{ij}$ and $g = -\omega$, the fourth-order compact approximation of the Poisson equation (13) becomes

$$\left[\delta_x^2 + \delta_y^2 + \frac{1}{12}(h^2 + k^2)\delta_x^2\delta_y^2\right]\psi_{ij} = \left[1 + \frac{h^2}{12}\delta_x^2 + \frac{k^2}{12}\delta_y^2\right]\omega_{ij} \tag{16}$$

For both the vorticity and streamfunction equation, the resulting matrix equations are solved using BiCGStab [17, 18], which constitutes the inner iterations. Once (13) is solved, u and v in (14) are calculated using a fourth-order compact formula (see Reference [23]). This constitutes one outer iteration cycle. For the inner iterations, the computations were stopped when the maximum ϕ-error (ϕ being either ω or ψ) between two successive iteration steps

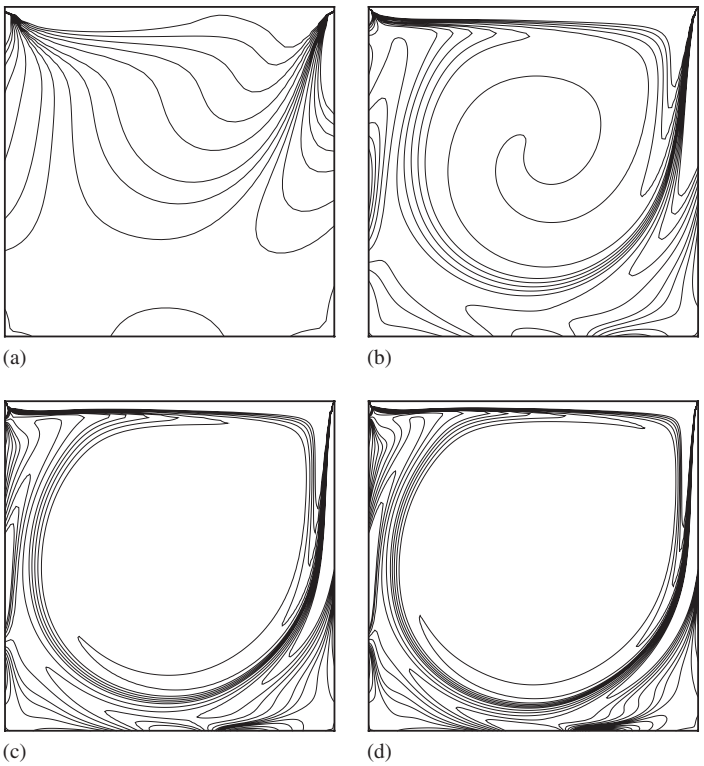


Figure 5. Steady-state vorticity contours for the lid-driven square cavity flows for: (a) $Re = 100$ (41×41); (b) $Re = 1000$ (101×101); (c) $Re = 3200$ (121×121); and (d) $Re = 5000$ (161×161).

was smaller than 0.5×10^{-6}. Steady state was assumed to reach when the maximum ω-error between two successive outer temporal iteration steps was smaller than the same tolerance limit 0.5×10^{-6}. In all the computations for this problem, zero initial data was used.

As described in Section 2, the closed form of c and d in Equation (1) are not known here. As such, for the first time level, $\delta_t^- c_{ij}$ and $\delta_t^- d_{ij}$ which are required to calculate C_{ij} and D_{ij} appearing in (6) were set as zeros. For the subsequent time levels, they can be computed easily as c_{ij} and d_{ij} are now known both at the current and previous time levels.

The details of applications of the present scheme to the primitive variable form (Equations (9)–(11)) of the N–S equations and the extension of it to three-dimensional problems will be discussed in separate papers in near future.

We now present our computational results for this problem in Figures 3–6 and Table V. Figure 3 exhibits comparisons of the horizontal velocities on the vertical centreline and the vertical velocities on the horizontal centreline of the square cavity for Reynolds numbers ranging from 100 to 5000 and compare our data with that from Ghia *et al.* [24]. In each case, our velocity profiles obtained on relatively coarser grids match very well with Ghia's results. It is worthwhile mentioning that the graph for $Re = 100$ was obtained on a grid of size 21×21 only and corresponding Δt was as high as 0.1.

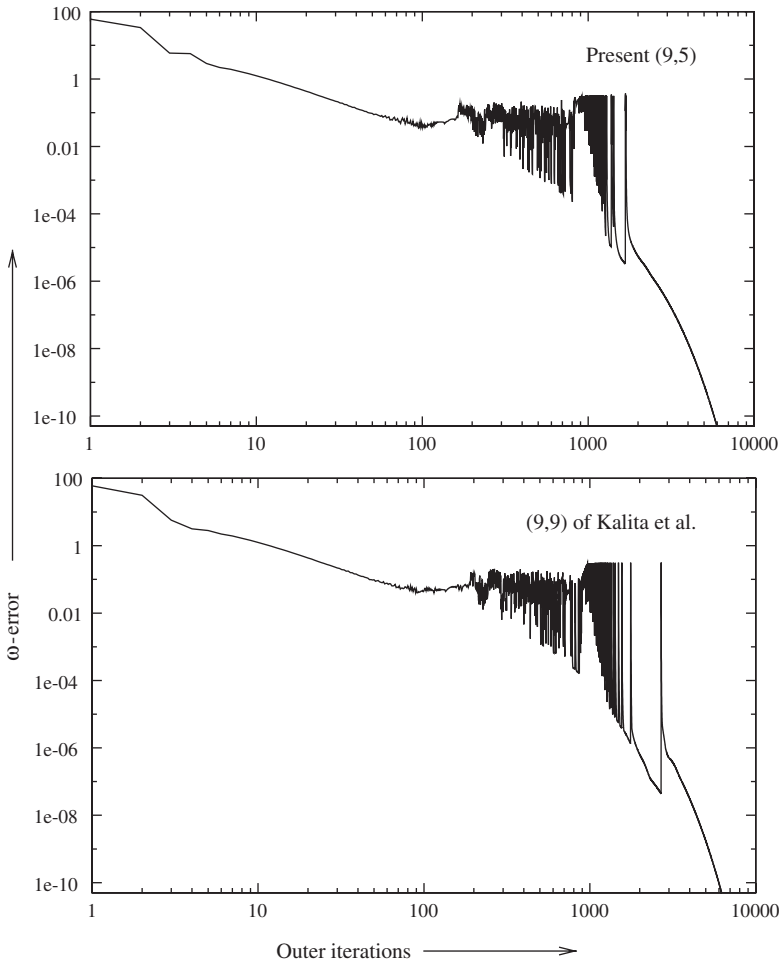


Figure 6. Comparison of the convergence history of the ω-error of the present $(9,5)$ scheme with the $(9,9)$ scheme of Reference [5] for the lid-driven cavity flow on 41×41 grid for $Re = 100$ ($\Delta t = 0.01$).

Table V. Convergence data till steady-state for the lid-driven cavity problem for $Re = 100$ and 1000.

	100 (41×41)		1000 (61×61)	
Re	Present $(9,5)$	$(9,9)$ [5]	Present $(9,5)$	$(9,9)$ [5]
Iterations	2016	2250	10 211	11 106
CPU (s)	4.617	7.457	138.216	187.328

In Figure 4, we present the well-known streamfunction contours for $100 \leqslant Re \leqslant 5000$ while Figure 5 shows the corresponding vorticity contours for the same Reynolds numbers. All these graphs exhibit the typical separations and secondary vortices at the bottom corners of

the cavity as well as at the top left (which is visible here from $Re = 3200$ onwards). We can also see evolution of tertiary vortices quite visibly at $Re = 5000$. Both the streamfunction and vorticity profiles match with the benchmark results of Ghia *et al.* [24] and other established results [25–28] thereby confirming that our formulation yields quantitatively accurate solutions.

In Figure 6 and Table V, we compare the convergence behaviour of the present $(9,5)$ scheme with that of the $(9,9)$ scheme [5]. Figure 6 shows the convergence history of the ω-error for $Re = 100$. Both Figures 6(a) and (b) show quite similar pattern of convergence history. This is also clear from Table V which show the number of outer iterations (one outer iteration here is equivalent to an increment $\Delta t = 0.01$ in time) along with the CPU time to reach the steady-state tolerance limit of 0.5×10^{-6}. From here, we see that though the number of outer iterations taken by the $(9,9)$ scheme is just slightly more than that taken by the $(9,5)$ scheme, CPU time-wise, the present scheme is more efficient than the $(9,9)$ scheme.

4. CONCLUSION

In this paper, we develop an implicit unconditionally stable HOC scheme for the unsteady 2-D convection–diffusion equations with variable coefficients. This $(9,5)$ scheme requires nine points at the nth and only five points at the $(n+1)$th time level. As against an earlier first-order time-accurate $(9,5)$ HOC scheme, the present scheme is temporally second and spatially fourth-order accurate. Both Dirichlet and Neumann boundary conditions can easily be incorporated into the scheme. It is easy to implement and the use of BiCGStab algorithm for solving the algebraic systems arising at every time level, makes the implicit procedure computationally efficient even in capturing transient solutions. To bring out different aspects of the scheme, we employed it to compute the transient solutions of two 2-D linear convection–diffusion problems and the time marching steady solution of the 2-D lid-driven cavity flow problem. The robustness of the scheme is illustrated by its applicability to a wide range of problems of varying physical complexities, represented among others, by Reynolds numbers ranging from 100 to 5000. Computational efficiency of the present scheme is reflected in very low demand on CPU time. The results obtained in all the three test cases with coarser grids are in excellent agreement with the analytical as well as established numerical results, underlining the high accuracy of the scheme. The implicit nature of the scheme is fully exploited in arriving at the steady-state results for the lid-driven cavity problem, where time steps as high as 0.1 (in conjunction with $h = k = 0.05$) has been employed for some of the computations. Because of second-order temporal accuracy, higher order spatial accuracy with a much smaller stencil and computational efficiency, the scheme has a good potential for efficient application to many problems of incompressible viscous flows. Currently, we are working on the application of the scheme to the primitive variable formulation of the N–S equations along with its ADI version.

ACKNOWLEDGEMENTS

The first author would like to thank Dr A.K. Dass, Department of Mechanical Engineering, Indian Institute of Technology Guwahati, India for his valuable suggestions.

REFERENCES

1. Rai MM, Moin P. Direct simulation of turbulent flows using finite difference methods. *Journal of Computational Physics* 1991; **96**:15–53.
2. Lele SK. Compact finite difference schemes with spectral like resolution. *Journal of Computational Physics* 1992; **103**:16–42.
3. Abarbanel S, Kumar A. Compact higher-order schemes for the Euler equations. *Journal of Scientific Computing* 1988; **3**:275–288.
4. Balzano A. Mosquito: an efficient finite difference scheme for numerical simulation of 2D advection. *International Journal for Numerical Methods in Fluids* 1999; **31**:481–496.
5. Kalita JC, Dalal DC, Dass AK. A class of higher order compact schemes for the unsteady two-dimensional convection–diffusion equation with variable convection coefficients. *International Journal for Numerical Methods in Fluids* 2002; **38**:1111–1131.
6. Noye BJ, Tan HH. A third order semi implicit method for solving the one-dimensional convection–diffusion equation. *International Journal for Numerical Methods in Engineering* 1988; **26**:1615–1629.
7. Noye BJ, Tan HH. Finite difference methods for solving two-dimensional advection diffusion equation. *International Journal for Numerical Methods in Fluids* 1989; **9**:75–98.
8. Noye BJ, Hayman KJ. Implicit two-level finite-difference methods for the two-dimensional diffusion equation. *International Journal of Computer Mathematics* 1993; **48**:219–228.
9. Karaa S, Zhang J. High order ADI method for solving unsteady convection–diffusion problems. *Journal of Computational Physics* 2004; **198**:1–9.
10. Chorin AJ. Numerical solution of the Navier–Stokes equation. *Mathematics of Computation* 1968; **22**:747–762.
11. Strikwerda JC. High-order accurate schemes for incompressible viscous flow. *International Journal for Numerical Methods in Fluids* 1997; **24**:715–734.
12. Hirsh RS. Higher order accurate solution of fluid mechanics problems by a compact differencing technique. *Journal of Computational Physics* 1975; **9**:90–109.
13. Gupta MM, Monohar RM, Stephenson JH. A single cell high order scheme for the convection–diffusion equation with variable coefficients. *International Journal for Numerical Methods in Fluids* 1984; **4**:641–651.
14. Kalita JC, Dass AK, Dalal DC. A transformation-free HOC for steady convection–diffusion on nonuniform grids. *International Journal for Numerical Methods in Fluids* 2004; **44**:33–53.
15. Mackinnon RJ, Johnson RW. Differential equation based representation of truncation errors for accurate numerical solution. *International Journal for Numerical Methods in Fluids* 1991; **13**:739–757.
16. Spotz WF, Carey GF. High-order compact scheme for the steady stream-function vorticity equations. *International Journal for Numerical Methods in Engineering* 1995; **38**:3497–3512.
17. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. SIAM: Philadelphia, PA, 1995.
18. Sleijpen GLG, Vendervorst HA. Hybrid bi-conjugate gradient methods for CFD problems. *Computational Fluid Dynamics Review* 1995; 457–476.
19. Subasi M. On the finite-difference schemes for the numerical solution of two dimensional Schrödinger equation. *Numerical Methods for Partial Differential Equation* 2002; **18**:752–758.
20. Anderson DA, Tannehil JC, Pletcher RH. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Corporation: New York, 1984.
21. Anderson Jr JD. *Computational Fluid Dynamics*. McGraw-Hill: New York, 1995.
22. Smith GD. *Numerical Solutions of Partial Differential Equations*: *Finite Difference Methods*. Clarendon Press: Oxford, 1984.
23. Kalita JC, Dalal DC, Dass AK. Fully compact higher order computation of steady state natural convection in a square cavity. *Physical Review E* 2001; **64**(6):066703(1–13).
24. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equation and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
25. Gupta MM, Kalita JC. A new paradigm for solving Navier–Stokes equations: streamfunction-velocity formulation. *Journal of Computational Physics* 2005; **207**(1):52–68.
26. Barragy E, Carey GF. Stream function-vorticity driven cavity solution using p finite elements. *Computers and Fluids* 1997; **26**:453–468.
27. Botella O, Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers and Fluids* 1998; **27**: 421–433.
28. Bruneau C-H, Jouron C. An efficient scheme for solving steady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1990; **89**:389–413.

