AIAA JOURNAL
Vol. 54, No. 2, February 2016

RK-SIMPLER: Explicit Time-Accurate Algorithm for Incompressible Flows

R. G. Rajagopalan* and Angela D. Lestari†
Iowa State University, Ames, Iowa 50011

DOI: 10.2514/1.J054030

The SIMPLE family of algorithms has popularized the pressure-based schemes for incompressible flows and is the basis of many commercial codes. The influence of pressure on velocity is of primary importance in incompressible flows. The continuity equation implicitly dictates the pressure field, yet pressure is not a variable of the mass conservation equation. The pressure term that appears in the momentum equations is often treated as a source term. Thus, there is no explicit conservation equation for pressure. This predicament, and its remedy well known by the name "pressure–velocity coupling," is resolved in SIMPLE and its variants by obtaining an approximate pressure correction field, which is used iteratively to correct the velocity field and/or the pressure field, seeking an overall satisfaction of the conservation equations. The approximate nature of the pressure correction equation often causes convergence issues. A new algorithm is presented here, eliminating the need for the pressure correction equation, based on the fact that if the pressure field is known, the momentum equations can be solved in any number of ways to obtain the velocity field correctly. An exact equation for the pressure field is obtained by manipulating the discretized mass and momentum equations based on SIMPLER, which is the only nonlinear equation solved iteratively in the new algorithm (RK-SIMPLER). The momentum equations are cast in the form of an ordinary differential equation suitable for time integration using Runge–Kutta stages. Once the pressure field is known, the velocity field is updated explicitly every time step without iteratively solving the momentum equations. This also means that there are no subiterations within a time step. In addition, there are no corrections for the pressure or the velocity field, and hence there is no need for the approximate pressure correction equation. The RK-SIMPLER algorithm proves that, for incompressible flows, the fundamental equation is the pressure–velocity coupled exact equation for pressure and that there is no need for the nonlinear velocity equations to be solved iteratively. Also, the new algorithm presented here uses only exact equations and requires neither underrelaxation for any of the discretized quations nor subiterations for the time integration. The only approximation is that the pressure field is held constant through the Runge–Kutta update of the velocity field. The RK-SIMPLER algorithm converges well and captures the unsteady flow features for the cases analyzed. In contrast, for steady flows, the algorithm is stable but less competitive compared to unsteady flow simulations in terms of CPU time, due to the restrictions on the allowable time step.

Nomenclature

A	=	interface profile assumption
a	=	coefficient of discretized equations
b	=	integrated source term
D	=	diffusion conductance
d	=	coefficient in pressure equation
E, W, N, S, P	=	east, west, north, south, and central primary grids
e, w, n, s	=	east, west, north, and south control volume faces
F	=	mass flux
J	=	total flux
P	=	Peclet number
p	=	pressure
R^i	=	residual function
Re	=	Reynolds number
S	=	source term
t	=	time
u, v	=	Cartesian velocity components
$\hat{u}, \hat{v}$	=	pseudovelocities
V	=	volume
x, y	=	Cartesian coordinates
α	=	constant of time integration scheme
Γ	=	diffusion coefficient
ν	=	kinematic viscosity
ρ	=	density
ϕ	=	scalar transport variable

Received 18 November 2014; revision received 12 March 2015; accepted for publication 7 May 2015; published online 15 December 2015.

*Professor, Department of Aerospace Engineering. Associate Fellow AIAA.

†Graduate Student, Department of Aerospace Engineering. Member AIAA.

I. Introduction

INCOMPRESSIBLE flows play an important role in engineering applications. The effect of pressure on velocity is of primary importance in incompressible flows. However, there is no direct conservation equation for pressure. The continuity equation implicitly dictates the pressure field, yet pressure is not a variable of the mass conservation equation. Hence, the incompressible Navier–Stokes equations are complex and require special algorithms to solve the nonlinear coupled set of partial differential equations. Current techniques for the solution of the incompressible Navier–Stokes Equations can be categorized as 1) vorticity/stream-function methods [1], 2) artificial compressibility methods [2], and 3) projection methods [3]. The vorticity/stream-function method [1] solves the vorticity transport equation, which is constructed by taking the curl of the momentum equation and eliminating the pressure terms. The method requires the use of vorticity boundary conditions, which are difficult to specify at times, and an additional calculation is required if the pressure field is desired. On the other hand, the artificial compressibility methods and the projection methods generally lead to indirect solutions of the discretized continuity equation. This is typically done either by adding "artificial compressibility" to the continuity equation or by indirectly satisfying the continuity equation with the pressure Poisson equation. More specifically, an artificial equation of state, $P = \delta\rho$, is the basis of the method of artificial compressibility [2]. This method differs from the projection method

in that the continuity equation is not satisfied until a steady state is reached. Chorin [2] uses central differences in space and time to arrive at a scheme designed for steady-state solutions. Methods for unsteady solutions using artificial compressibility have also been developed by later researchers (Soh and Goodrich [4]). The projection method [3] is a fractional step method in which intermediate velocity and pressure fields are calculated. The intermediate pressure and velocity are then corrected sequentially by the divergence of the intermediate velocity (continuity equation) and the pressure gradient, respectively. New values for pressure and velocity are obtained until the divergence of the velocity vanishes. The SIMPLE and SIMPLER family of methods [5] fall in this class. The pressure correction equation plays a very important role in these methods, the derivation of which requires the use of the approximate forms of the momentum equations and the continuity equation. The SIMPLE [5] family of algorithms has popularized the pressure-based schemes for incompressible flows. The pressure velocity coupling (absence of pressure as a variable in continuity) is resolved in SIMPLE and its variants (SIMPLER [5], SIMPLEC [6], MSIMPLER [7], CLEAR [8], CLEARER [9]) by obtaining an approximate pressure correction field, which is used iteratively to correct the velocity field and/or the pressure field seeking an overall satisfaction of the conservation equations. Another SIMPLE variant (IDEAL [10]) removes any need for a pressure correction by employing two inner iterations in an attempt to more closely couple pressure and velocity without introducing approximate equations. All of the aforementioned methods use some form of underrelaxation for the velocities. The approximate nature of the pressure correction equation and/or the velocity underrelaxation is often attributed to convergence issues. This difficulty for incompressible flow calculations can again be traced to the continuity equation, which is given not in a time evolution form but in the form of a divergence-free constraint. This constraint, which is the continuity equation, prohibits time integration of the incompressible flow equations in a straightforward manner [4,11].

In this paper, a new algorithm (RK-SIMPLER) is proposed for incompressible flows. The RK-SIMPLER algorithm does not involve any approximate equation or require underrelaxation. The method stems from the fact that pressure and velocity are weakly coupled in incompressible flow, but if the pressure field is known exactly, the velocity field can be solved directly as a nonlinear coupled scalar field. To achieve this, the momentum conservation equations are spatially discretized first to yield a set of first-order ordinary differential equations (ODEs) in time. To the extent possible, the development of the algorithm follows the notation used in [5]. An explicit four-stage Runge–Kutta algorithm [12,13] of second-order temporal accuracy, is used to integrate the momentum conservation equations. The spatial and temporal discretization methods used are strictly arbitrary and can be easily substituted. For example, Purohit [14] exploits the fully implicit temporal discretization for deriving the exact pressure equation, whereas in the current development, the second-order Crank–Nicholson [15,16] is employed. The development is shown in Cartesian coordinates for simplicity, but the results presented are from a two-dimensional cylindrical $(r-\theta)$ system solver. The RK-SIMPLER algorithm works well for structured [14] as well as unstructured grid systems as presented by Lestari [17]. In passing, it is also noted that [18] has exploited the implicit Runge–Kutta algorithm in conjunction with the SIMPLE algorithm to obtain time-accurate calculations but primarily relies on SIMPLE to address the pressure velocity coupling and inherently has the same limitations as the SIMPLE algorithm. Also, Sanderse and Koren [19] and Cabuk et al. [20] present alternate use of explicit Runge–Kutta methods for the solution of incompressible Navier–Stokes equations and time integration. The RK-SIMPLER algorithm presented in this article is of a generic nature and offers a completely new way to solve the incompressible Navier–Stokes equations. In developing this algorithm, the general discretization techniques used by [5,16] are used in contrast to [19,20] minimizing code modifications. In principle, the only constraint on this algorithm is that an exact equation for pressure be available at the beginning of the iteration that can be solved from a guessed velocity field. The explicit RK-SIMPLER algorithm works well on staggered and collocated grid systems [17].

II. Explicit Four-Stage Runge–Kutta for Navier–Stokes Equations

For the sake of simplicity, all the equations are developed in two-dimensional Cartesian coordinates. However, the algorithm and the underlying principle are general and can be applied to all structured (staggered or collocated) or unstructured grid systems. In this development, wherever possible, the nomenclature closely follows [5]. The continuity equation in two-dimensional Cartesian coordinates can be written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{1}$$

For incompressible flows (ρ being a constant), the previous equation can be integrated over the control volume shown in Fig. 1 to yield

$$(F_e - F_w + F_n - F_s) = 0 \tag{2}$$

where F_e, F_w, F_n, and F_s are the mass flow rates through the faces of the control volume given by

$$F_e = (\rho u)_e \Delta y$$
$$F_w = (\rho u)_w \Delta y$$
$$F_n = (\rho v)_n \Delta x$$
$$F_s = (\rho v)_s \Delta x$$

The x and y momentum equations are given by

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial J_{u-x}}{\partial x} + \frac{\partial J_{u-y}}{\partial y} = S_u - \frac{\partial p}{\partial x} \tag{3}$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial J_{v-x}}{\partial x} + \frac{\partial J_{v-y}}{\partial y} = S_v - \frac{\partial p}{\partial y} \tag{4}$$

where S_u and S_v are the source terms, and J_{u-x}, J_{u-y}, J_{v-x}, and J_{v-y} are the sum of convection and diffusion fluxes defined by

$$J_{u-x} = \rho u u - \Gamma \frac{\partial u}{\partial x} \qquad J_{u-y} = \rho u v - \Gamma \frac{\partial u}{\partial y}$$
$$J_{v-x} = \rho u v - \Gamma \frac{\partial v}{\partial x} \qquad J_{v-y} = \rho v v - \Gamma \frac{\partial v}{\partial y}$$

Performing only spatial integration on the x momentum equation, over the control volume presented in Fig. 1, leads to the following discrete equation:

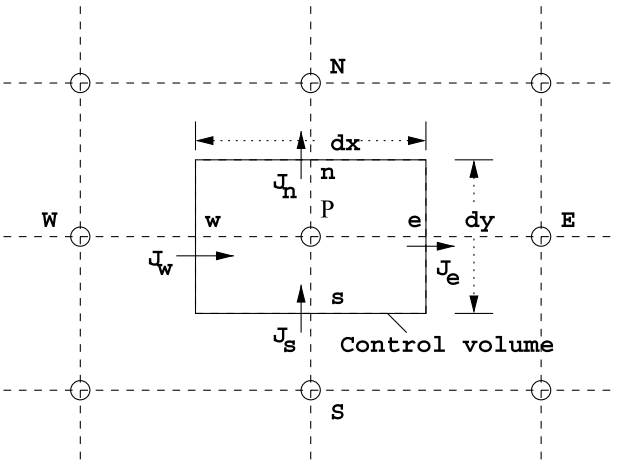

Fig. 1 Control volume for the two-dimensional situation.

$$\frac{\mathrm{d}(\rho u)}{\mathrm{d}t}\Delta x\Delta y + J_{u-e} - J_{u-w} + J_{u-n} - J_{u-s} = S_u\Delta x\Delta y + (p_w - p_e)\Delta y \tag{5}$$

where J_{u-e}, J_{u-w}, J_{u-n}, and J_{u-s} are the integral fluxes at the interfaces e, w, n, and s, respectively, and reduce to

$$J_{u-e} = (J_{u-x}\Delta y)_e$$
$$J_{u-w} = (J_{u-x}\Delta y)_w$$
$$J_{u-n} = (J_{u-y}\Delta x)_n$$
$$J_{u-s} = (J_{u-y}\Delta x)_s$$

Multiplying the discrete continuity equation [Eq. (2)] by u_P and subtracting it from the discrete momentum equation [Eq. (5)] yields

$$\frac{\mathrm{d}(\rho u)}{\mathrm{d}t}\Delta x\Delta y + (J_{u-e} - F_e u_P) - (J_{u-w} - F_w u_P) + (J_{u-n} - F_n u_P) - (J_{u-s} - F_s u_P) = S_u\Delta x\Delta y + (p_w - p_e)\Delta y \tag{6}$$

The previous equation is cast as an ordinary differential equation as follows:

$$\frac{\mathrm{d}(\rho u)}{\mathrm{d}t}\Delta x\Delta y = -[\bar{a}_{u-E}(u_P - u_E) - \bar{a}_{u-W}(u_W - u_P) + \bar{a}_{u-N}(u_P - u_N) - \bar{a}_{u-S}(u_S - u_P)] + S_u\Delta x\Delta y + (p_w - p_e)\Delta y$$

where the following substitutions have been made in accordance with [5]:

$$J_{u-e} - F_e u_P = \bar{a}_{u-E}(u_P - u_E)$$
$$J_{u-w} - F_w u_P = \bar{a}_{u-W}(u_W - u_P)$$
$$J_{u-n} - F_n u_P = \bar{a}_{u-N}(u_P - u_N)$$
$$J_{u-s} - F_s u_P = \bar{a}_{u-S}(u_S - u_P)$$

where

$$\bar{a}_{u-E} = D_e A(|P_e|) + [\![-F_e, 0]\!]$$
$$a_{u-W} = D_w A(|P_w|) + [\![F_w, 0]\!]$$
$$a_{u-N} = D_n A(|P_n|) + [\![-F_n, 0]\!]$$
$$a_{u-S} = D_s A(|P_s|) + [\![F_s, 0]\!]$$

The power-law profile is used for $A(|P|)$ from [5]. Upon rearrangement, the following equation is obtained:

$$\frac{\mathrm{d}(\rho u)}{\mathrm{d}t}\Delta x\Delta y = \bar{a}_{u-E}u_E + \bar{a}_{u-W}u_W + \bar{a}_{u-N}u_N + \bar{a}_{u-S}u_S - (\bar{a}_{u-E} + \bar{a}_{u-W} + \bar{a}_{u-N} + \bar{a}_{u-S})u_P + S_u\Delta x\Delta y + (p_w - p_e)\Delta y$$

Upon substituting

$$\bar{a}_{u-P} = (\bar{a}_{u-E} + \bar{a}_{u-W} + \bar{a}_{u-N} + \bar{a}_{u-S})$$
$$\bar{b}_u = S_u\Delta x\Delta y + (p_w - p_e)\Delta y$$

the momentum equation becomes

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{(\bar{a}_{u-E}u_E + \bar{a}_{u-W}u_W + \bar{a}_{u-N}u_N + \bar{a}_{u-S}u_S - \bar{a}_{u-P}u_P + \bar{b}_u)}{\rho\Delta x\Delta y}$$

If the numerator on the right-hand side is written as

$$R_u^i = (\bar{a}_{u-E}u_E + \bar{a}_{u-W}u_W + \bar{a}_{u-N}u_N + \bar{a}_{u-S}u_S - \bar{a}_{u-P}u_P + \bar{b}_u)$$

the previous equation reduces to

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{R_u^i}{\rho\Delta x\Delta y} \tag{7}$$

Similarly, the following holds true for the y momentum equation:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{(\bar{a}_{v-E}v_E + \bar{a}_{v-W}v_W + \bar{a}_{v-N}v_N + \bar{a}_{v-S}v_S - \bar{a}_{v-P}v_P + \bar{b}_v)}{\rho\Delta x\Delta y}$$
$$\bar{a}_{v-P} = (\bar{a}_{v-E} + \bar{a}_{v-W} + \bar{a}_{v-N} + \bar{a}_{v-S})$$
$$\bar{b}_v = S_v(\Delta x\Delta y) + (p_s - p_n)\Delta x$$
$$R_v^i = (\bar{a}_{v-E}v_E + \bar{a}_{v-W}v_W + \bar{a}_{v-N}v_N + \bar{a}_{v-S}v_S - \bar{a}_{v-P}v_P + \bar{b}_v)$$
$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{R_v^i}{\rho\Delta x\Delta y} \tag{8}$$

The spatial discretization of the x and y momentum equations yields a coupled set of ordinary differential equations suitable for time integration using the explicit Runge–Kutta scheme [12]. The generic ODE given by

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = \frac{R_i(\phi_i)}{V_i} \tag{9}$$

can be integrated in time using the four-stage explicit Runge–Kutta scheme to yield

$$\phi^{(0)} = \phi^n$$
$$\phi^{(1)} = \phi^{(0)} + \frac{1}{4}\frac{\Delta t}{V_i}R_i(\phi^{(0)})$$
$$\phi^{(2)} = \phi^{(0)} + \frac{1}{3}\frac{\Delta t}{V_i}R_i(\phi^{(1)})$$
$$\phi^{(3)} = \phi^{(0)} + \frac{1}{2}\frac{\Delta t}{V_i}R_i(\phi^{(2)})$$
$$\phi^{(4)} = \phi^{(0)} + \frac{\Delta t}{V_i}R_i(\phi^{(3)})$$
$$\phi^{n+1} = \phi^{(4)} \tag{10}$$

Equations (7) and (8) are similar to Eq. (9) and can be explicitly integrated in time, provided pressure is known, using the Runge–Kutta stage equations [Eq. (10)] described previously. In the next section, the development of the exact equation for pressure is undertaken.

III. Exact Equation for Pressure

The continuity equation provides an implicit equation for pressure. The main goal is that the discretized equation for pressure should have no approximations. Once again, the pressure equation in SIMPLER [5] is a good candidate because it is well established and is an exact equation.

A. Time Integration

The method of time integration for several schemes including Crank–Nicholson can be conveniently stated as

$$\int_{t_o}^{t}\phi\,\mathrm{d}t = [\alpha\phi + (1-\alpha)\phi_o]\Delta t \tag{11}$$

where ϕ is the quantity to be integrated, and the value of α may be different depending on the scheme of choice. All time integrations in the following few sections are performed using the previous stencil.

The value of $\alpha = 0.5$ in the previous equation corresponds to the Crank–Nicholson time integration scheme. This scheme makes use of trapezoidal differencing to achieve second-order accuracy [21].

The unsteady form of the continuity equation in two-dimensional Cartesian coordinates [Eq. (1)] is discretized with the help of Eq. (11) to yield

$$(\rho - \rho^o)\frac{\Delta x \Delta y}{\Delta t} + \alpha(F_e - F_w + F_n - F_s) + (1-\alpha)(F_e^o - F_w^o + F_n^o - F_s^o) = 0 \tag{12}$$

where F_e, F_w, F_n, and F_s as before are the mass flow rates through the faces of the control volume, and F_e^o, F_w^o, F_n^o, and F_s^o represent the mass flow rates calculated at the previous time step. For incompressible flows, the previous equation can be written as

$$\begin{aligned} F_e - F_w + F_n - F_s &= 0 \\ F_e^o - F_w^o + F_n^o - F_s^o &= 0 \end{aligned} \tag{13}$$

B. Discretization of Momentum Equation

Upon spatial and time integration of the x momentum equation [Eq. (3)], we get

$$\begin{aligned} &[(\rho u) - (\rho u)^o]\frac{\Delta x \Delta y}{\Delta t} + \alpha(J_{u-e} - J_{u-w} + J_{u-n} - J_{u-s}) \\ &\quad + (1-\alpha)(J_{u-e}^o - J_{u-w}^o + J_{u-n}^o - J_{u-s}^o) \\ &\quad = [\alpha S_u + (1-\alpha)S_u^o]\Delta x \Delta y + (p_w - p_e)\Delta y \end{aligned}$$

Following [5], the x momentum equation after algebraic manipulations becomes

$$\begin{aligned} &u_P \frac{\rho \Delta x \Delta y}{\Delta t} + \alpha(\bar{a}_{u-E} + \bar{a}_{u-W} + \bar{a}_{u-N} + \bar{a}_{u-S})u_P \\ &\quad = \alpha(\bar{a}_{u-E}u_E + \bar{a}_{u-W}u_W + \bar{a}_{u-N}u_N + \bar{a}_{u-S}u_S) + \frac{\rho \Delta x \Delta y}{\Delta t}u_P^o \\ &\quad - (1-\alpha)(\bar{a}_{u-E}^o + \bar{a}_{u-W}^o + \bar{a}_{u-N}^o + \bar{a}_{u-S}^o)u_P^o \\ &\quad + (1-\alpha)(\bar{a}_{u-E}^o u_E^o + \bar{a}_{u-W}^o u_W^o + \bar{a}_{u-N}^o u_N^o + \bar{a}_{u-S}^o u_S^o) \\ &\quad + [\alpha S_u + (1-\alpha)S_u^o]\Delta x \Delta y + (p_w - p_e)\Delta y \end{aligned}$$

where the following substitutions have been made:

$$\begin{aligned} J_{u-e} - F_e u_P &= \bar{a}_{u-E}(u_P - u_E) \\ J_{u-w} - F_w u_P &= \bar{a}_{u-W}(u_W - u_P) \\ J_{u-n} - F_n u_P &= \bar{a}_{u-N}(u_P - u_N) \\ J_{u-s} - F_s u_P &= \bar{a}_{u-S}(u_S - u_P) \\ J_{u-e}^o - F_e^o u_P^o &= \bar{a}_{u-E}^o(u_P^o - u_E^o) \\ J_{u-w}^o - F_w^o u_P^o &= \bar{a}_{u-W}^o(u_W^o - u_P^o) \\ J_{u-n}^o - F_n^o u_P^o &= \bar{a}_{u-N}^o(u_P^o - u_N^o) \\ J_{u-s}^o - F_s^o u_P^o &= \bar{a}_{u-S}^o(u_S^o - u_P^o) \end{aligned}$$

Or in a compact form, the temporally discretized x momentum equation can be written as

$$\begin{aligned} a_{u-p}u_P = a_{u-E}u_E + a_{u-W}u_W + a_{u-N}u_N + a_{u-S}u_S + b_u \\ + (p_w - p_e)\Delta y \end{aligned} \tag{14}$$

where the coefficient a_{u-p} becomes

$$a_{u-p} = \frac{\rho \Delta x \Delta y}{\Delta t} + \alpha(\bar{a}_{u-E} + \bar{a}_{u-W} + \bar{a}_{u-N} + \bar{a}_{u-S})$$

and

$$\begin{aligned} a_{u-E} &= \alpha(\bar{a}_{u-E}) \\ a_{u-W} &= \alpha(\bar{a}_{u-W}) \\ a_{u-N} &= \alpha(\bar{a}_{u-N}) \\ a_{u-S} &= \alpha(\bar{a}_{u-S}) \end{aligned}$$

The term b_u is given by

$$\begin{aligned} b_u = (1-\alpha)&[\bar{a}_{u-E}^o u_E^o + \bar{a}_{u-W}^o u_W^o + \bar{a}_{u-N}^o u_N^o + \bar{a}_{u-S}^o u_S^o \\ &- (\bar{a}_{u-E}^o + \bar{a}_{u-W}^o + \bar{a}_{u-N}^o + \bar{a}_{u-S}^o)u_P^o] \\ &+ \left[\frac{\rho}{\Delta t}u_P^o + \alpha S_u + (1-\alpha)S_u^o\right]\Delta x \Delta y \end{aligned}$$

In a similar fashion, the y momentum equation can be written as

$$\begin{aligned} a_{v-p}v_p = a_{v-E}v_E + a_{v-W}v_W + a_{v-N}v_N + a_{v-S}v_S + b_v \\ + (p_s - p_n)\Delta x \end{aligned} \tag{15}$$

where

$$\begin{aligned} a_{v-p} &= \frac{\rho(\Delta x \Delta y)}{\Delta t} + \alpha(\bar{a}_{v-E} + \bar{a}_{v-W} + \bar{a}_{v-N} + \bar{a}_{v-S}) \\ b_v &= (1-\alpha)[\bar{a}_{v-E}^o v_E^o + \bar{a}_{v-W}^o v_W^o + \bar{a}_{v-N}^o v_N^o + \bar{a}_{v-S}^o v_S^o \\ &\quad - (\bar{a}_{v-E}^o + \bar{a}_{v-W}^o + \bar{a}_{v-N}^o + \bar{a}_{v-S}^o)v_p^o] \\ &\quad + \left[\frac{\rho}{\Delta t}v_p^o + \alpha S_v + (1-\alpha)S_v^o\right]\Delta x \Delta y \end{aligned}$$

and the other coefficients can be defined in a manner similar to the coefficients in the x momentum equation.

C. Pressure Equation

Now an exact equation for pressure is derived as follows. From the x momentum and y momentum [Eqs. (14) and (15)], rewritten for control volume interfaces (e, w, n, s), we obtain following Ref. [5]:

$$\begin{aligned} u_e &= \hat{u}_e + d_e(p_P - p_E) \\ u_w &= \hat{u}_w + d_w(p_W - p_P) \\ v_n &= \hat{v}_n + d_n(p_P - p_N) \\ v_s &= \hat{v}_s + d_s(p_S - p_P) \end{aligned} \tag{16}$$

where

$$\begin{aligned} \hat{u} &= \frac{\sum a_{u-nb}u_{nb} + b_u}{a_{u-p}} \\ \hat{v} &= \frac{\sum a_{v-nb}v_{nb} + b_v}{a_{v-p}} \end{aligned} \tag{17}$$

Substituting these expressions in the discretized incompressible continuity equation:

$$[(\rho u)_e - (\rho u)_w]\Delta y + [(\rho v)_n - (\rho v)_s]\Delta x = 0$$

yields upon rearrangement an exact equation for pressure:

$$a_p p_p = a_E p_E + a_W p_W + a_N p_N + a_S p_S + b_p \tag{18}$$

where

$$a_E = \rho d_e \Delta y$$
$$a_W = \rho d_w \Delta y$$
$$a_N = \rho d_n \Delta x$$
$$a_S = \rho d_s \Delta x$$
$$a_p = a_E + a_W + a_N + a_S$$

and

$$b_p = [(\rho\hat{u})_w - (\rho\hat{u})_e]\Delta y + [(\rho\hat{v})_s - (\rho\hat{v})_n]\Delta x$$

IV. Numerical Algorithm

As mentioned earlier, no approximations are introduced in the derivation of the pressure equation. Thus, if the pseudovelocities [$\hat{u}$ and $\hat{v}$ from Eq. (17)] could be calculated from the velocity field, the pressure equation would yield the pressure field corresponding to the velocity field.

A. RK-SIMPLER: Explicit Algorithm Based on Runge–Kutta Scheme

This algorithm is based on the fact that, if the pressure field is known, the momentum equations can be solved to obtain the velocities correctly. Therefore, we solve the pressure equation first and then update the velocities using the Runge–Kutta time integration scheme. The sequence of operations can be stated as follows.

1) Start with a guessed velocity field.

2) Calculate the coefficients for the pressure Eq. (18), and solve it to obtain the pressure field.

3) Update the u velocity [Eq. (7)] and v velocity [Eq. (8)] using the four-stage explicit Runge–Kutta steps given by Eq. (10).

4) Start a new time level, and go to step 2 with the new velocity field.

The flowchart of this algorithm is shown in Fig. 2. It should be noted that the RK-SIMPLER algorithm does not require subiterations for time integration and advances to the next time step by updating the velocity field by the Runge–Kutta stage equations. The pressure field and the nonlinear coefficients for the momentum equations remain constant over all the Runge–Kutta stages even though the velocities change. The coefficients are updated at the beginning of every time step. Such a procedure (no iterations) implies that the difference in the two time levels has to be small and is a characteristic of explicit algorithms. However, as will be shown, the algorithm converges well for low-Reynolds-number internal flow (polar cavity) as well as the high-Reynolds-number flow over a two-dimensional half-cup. Purohit [14] compares the total time to resolve a flow completely to convergence and shows it to be economical.

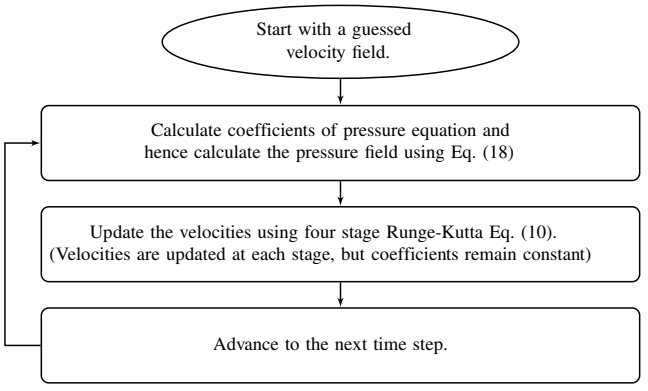

Fig. 2 RK-SIMPLER flowchart.

To appreciate the degree of difference between the conventional SIMPLER algorithm and the RK-SIMPLER algorithm, both the algorithms are presented for one time step.

SIMPLER: Start of a new time level
Do subiterations to convergence
Step 1: Solve the pressure Poisson equation.
Step 2: Solve the vector momentum equation.
Step 3: Solve the pressure correction equation.
Step 4: Update the velocity field.
End do

Thus, from the algorithmic steps outlined previously, for two-dimensional (2-D) simulations, the original SIMPLER algorithm solves four nonlinear equations iteratively: one for the pressure, two for the momentum equations, and one for the pressure correction equation every subiteration, in addition to updating the velocity field at the end of every subiteration. In contrast, the only nonlinear equation that is iteratively solved in the RK-SIMPLER algorithm is the pressure–velocity coupled exact equation based on mass conservation and an explicit velocity update once per time step without subiterations as outlined next.

RK-SIMPLER: Start of a new time level
Step 1: Solve the pressure Poisson equation.
Step 2: Update the velocity field.

B. Implementation Details of RK-SIMPLER

All the velocities (pseudo and regular) are calculated at the interface between the cells. The pressure is evaluated at the cell centers. Only gauge pressure is evaluated, and this eliminates the need to specify boundary pressures. Pressure is computed once at the beginning of a time step using the Poisson equation for pressure and is held constant through the rest of the time step. With the pressure field assumed known, the velocity field is advanced to the next time using the Runge–Kutta stage equations. The next time step requires only the updated velocity field and not the pressure field.

In summary, in addition to having fewer steps within a time step, the RK-SIMPLER algorithm does not require subiterations for time integration. Additionally, the explicit nature of the Runge–Kutta algorithm has time-step restrictions that make it more economical for unsteady problems compared to steady problems.

V. Results

The new RK-SIMPLER algorithm presented has had a significant amount of testing for various situations.

1) Purohit [14] gives more details of the comparisons between the conventional SIMPLER algorithm and the RK-SIMPLER algorithm for unsteady problems including time to convergence. Purohit reports that, in comparison to SIMPLER using Crank–Nicholson algorithm, roughly 10X speed-up is achieved (in RK-SIMPLER) for convergence of 2-D problems despite the fact that the time step allowable in the explicit RK-SIMPLER is an order of magnitude smaller.

2) Fischels and Rajagopalan [22] present the RK-SIMPLER algorithm with implicit Runge–Kutta stages as opposed to explicit Runge–Kutta stages, thereby increasing the time step that can be taken.

3) Garrick and Rajagopalan [23] prove that the velocity update by Runge–Kutta stages presented in this paper can be replaced by other schemes for the time advancement of velocity field, further attesting to the fact that the pressure–velocity coupled Poisson equation is the key to the solution of the incompressible flow.

4) Lestari [17] shows that the RK-SIMPLER algorithm works equally well for 2-D unstructured collocated grid systems.

5) Thistle [24] demonstrates that the RK-SIMPLER algorithm works for three-dimensional unstructured tetrahedral grids and offers significant run-time benefits.

The algorithm as presented with the power law assumption is first-order in space. Pressure and the coefficients are maintained constant in the Runge–Kutta stages used to advance the velocity field in time. The algorithm is stable and establishes that the Poisson equation for pressure is the key equation for incompressible flows.

Two specific cases (polar cavity [25] and a two-dimensional half-cup) suitable for cylindrical polar grid system are presented here. All the computations use laminar conservation equations in a non-uniform cylindrical coordinate system. Similar cases using the Cartesian grid systems are discussed in [14]. The Poison equation for pressure is solved for the gauge pressure within the domain for all the cases and thus does not require a boundary condition for pressure at any of the boundaries. The velocity boundary conditions are specific to the problem and are discussed within the problem description.

A. Lid-Driven Polar Cavity Flow

A polar, lid-driven cavity was simulated to validate the RK-SIMPLER presented here for various grid sizes. The cavity, shown in Fig. 3, is bound by a moving lid (with a prescribed velocity) and three viscous walls modeled by no-slip boundary condition. The lid boundary is the arc that subtends an angle of 1 rad at a radial distance of 1 ft from a vertex. The two viscous side walls are located along the rays that define the 1 rad angle. They begin at a radius of 1 ft from the vertex and continue to 2 ft. The bottom viscous wall is the arc that subtends a 1 rad angle at a distance of $r_o = 2$ ft from the vertex. The lid boundary imparts motion to the fluid inside of the cavity via applying a tangential velocity. The tangential velocity U_{lid} is set at 1 ft/s, corresponding to a Reynolds number of 350 (based on U_{lid} and r_{lid}). In Fig. 4, the tangential velocity grid-independence study is presented, and in Fig. 5, the radial velocity grid-independence study is presented. The RK-SIMPLER algorithm responds well to the grid-refinement study, and the values tend toward the experimental values from [25] as the grid is refined.

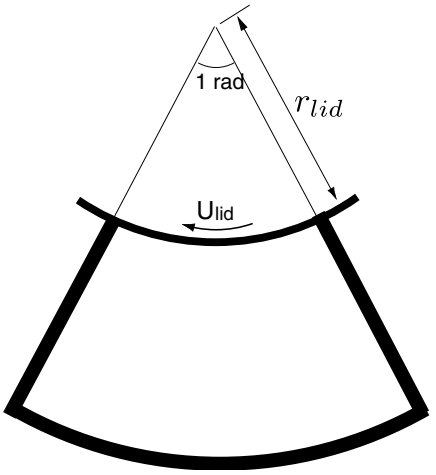

Fig. 3 Schematic of lid-driven polar cavity.

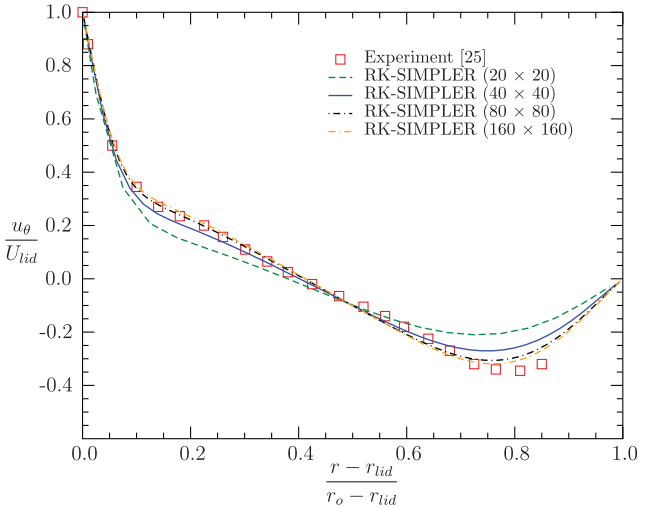

Fig. 4 u_θ velocity profile at $\theta = \frac{1}{2}$ radial line for $Re = 350$.

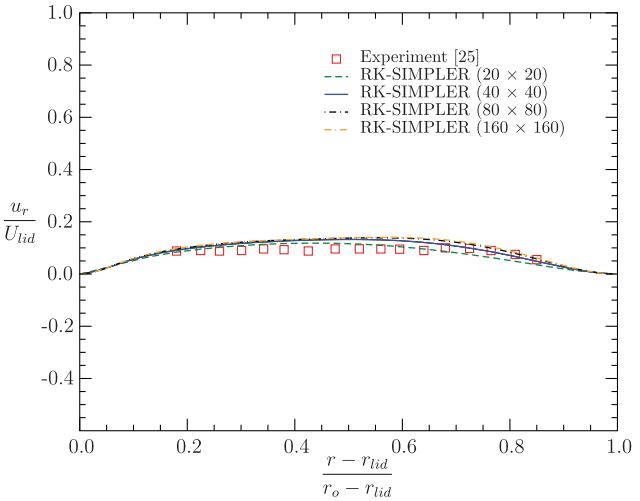

Fig. 5 u_r velocity profile at $\theta = \frac{1}{2}$ radial line for $Re = 350$.

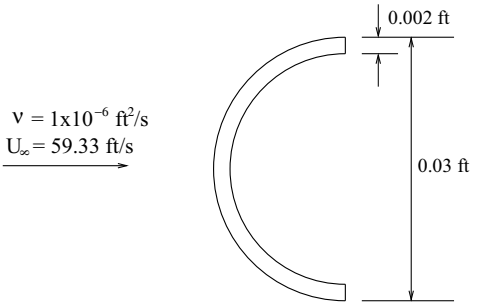

Fig. 6 Schematic of the two-dimensional cup.

B. Two-Dimensional Cup

Unsteady incompressible flow simulations at high Reynolds numbers involve ODEs that are stiff and are known to have convergence problems. Here, the analysis of the flow over a two-dimensional circular cup at a Reynolds number of 1.78E6 based on the outer diameter of the cup is presented. The velocity at the inlet, top and bottom boundaries are assumed to be freestream. The outflow boundary has a velocity profile deduced from the interior node subject to mass conservation. The cup has a finite thickness, and the dimensions are presented in Fig. 6. Because the cup has sharp edges, it has good unsteady characteristics and the points of separation are fixed.

Depicted in Fig. 7 is the time history of the drag coefficient (based on the diameter of the cup)

$$C_d = \frac{\text{Drag}}{\frac{1}{2}\rho V_\infty^2 (\text{dia})}$$

using the new RK-SIMPLER and the Crank–Nicholson based SIMPLER algorithm as presented in [14–17] on a coarse grid of (74×98) points. The domain covers a circular region of 21.5 diameters in the radial direction and 2π rad in the theta direction. The principal aim of this exercise is to verify that the Crank–Nicholson-based SIMPLER algorithm and Runge–Kutta-based RK-SIMPLER converge to the same unsteady solution given a geometry and grid. Hence, correlation with experimental values is not attempted. The codes produce identical results with the same transient behavior of an initial steep fall in the drag coefficient, followed by a steady

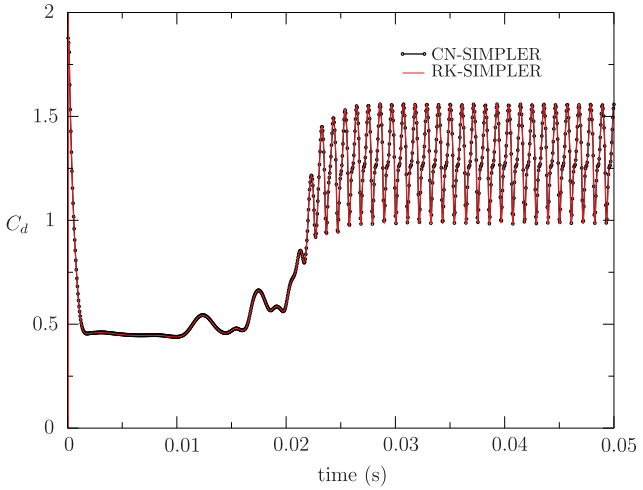

Fig. 7 Time history of the drag coefficient of the cup.

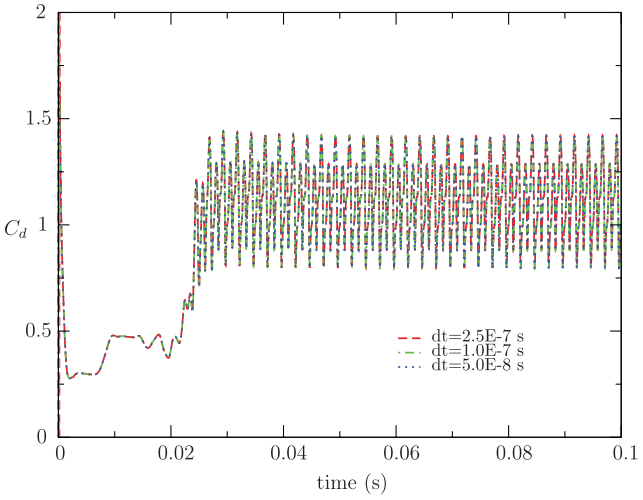

Fig. 8 Time history of the drag coefficient of the cup.

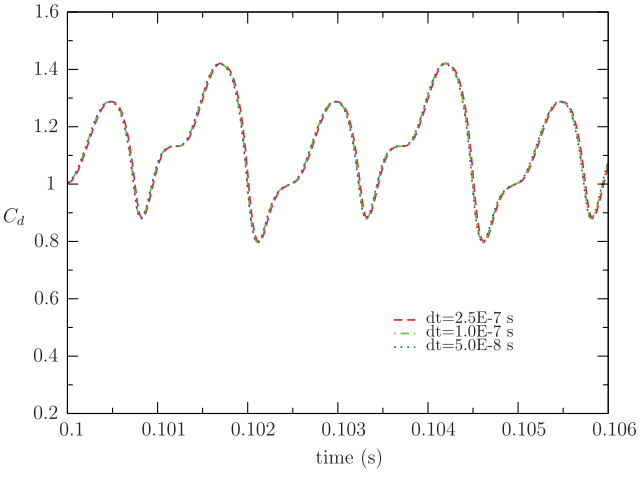

Fig. 9 Close-up view of the drag coefficient of the cup.

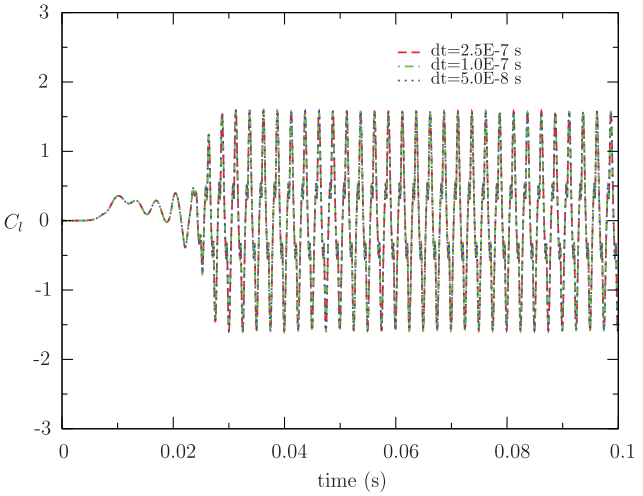

Fig. 10 Time history of the lift coefficient of the cup.

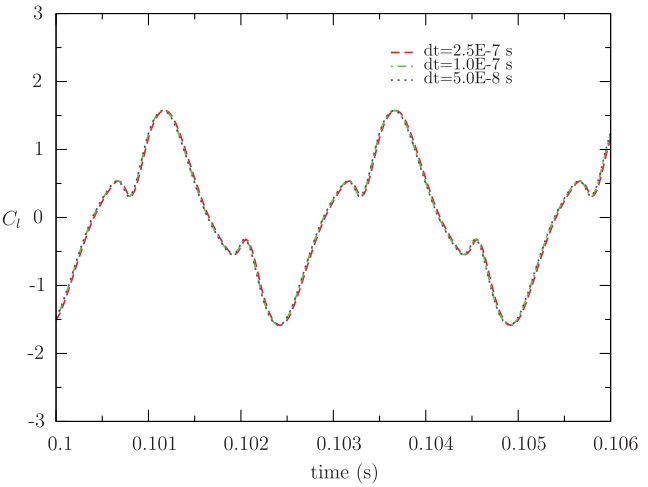

Fig. 11 Close-up view of the lift coefficient of the cup.

level region, followed by the unsteady shedding related periodic variations.

Time convergence of the drag coefficient is shown in Fig. 8 for various time steps using the new RK-SIMPLER algorithm. The transient behavior, the period, and the amplitude all agree well for this intermediate grid of size (134×168) for the different time steps, establishing that the RK-SIMPLER algorithm is temporally accurate. A close-up view of this time history is shown in Fig. 9. The two-dimensional cup is a symmetric object and, for a steady flow, produces zero mean vertical force. However, because of the unsteady shedding of the vortices from the top and bottom corners that alternate, there is a net force in the vertical direction, which oscillates about the zero mean. In Fig. 10, the unsteady convergence of the oscillations of lift coefficient

$$C_l = \frac{\text{Life}}{\frac{1}{2}\rho V_\infty^2 (\text{dia})}$$

is captured for various time steps. The degree to which the curves match for various time steps in capturing the unsteady lift coefficient for the two-dimensional cup is illustrated in Fig. 11. There is time convergence in every aspect of the simulation including the amplitude and frequency of oscillations. The velocity magnitude of the flowfield at the instant of shedding from the top and bottom portion of the cup corresponding to the two peak-valley cycles in Fig. 11 are presented in Figs. 12a, 12c, 12e, and 12g. Corresponding images for instantaneous unsteady pressures are presented in Figs. 12b, 12d, 12f, and 12h.

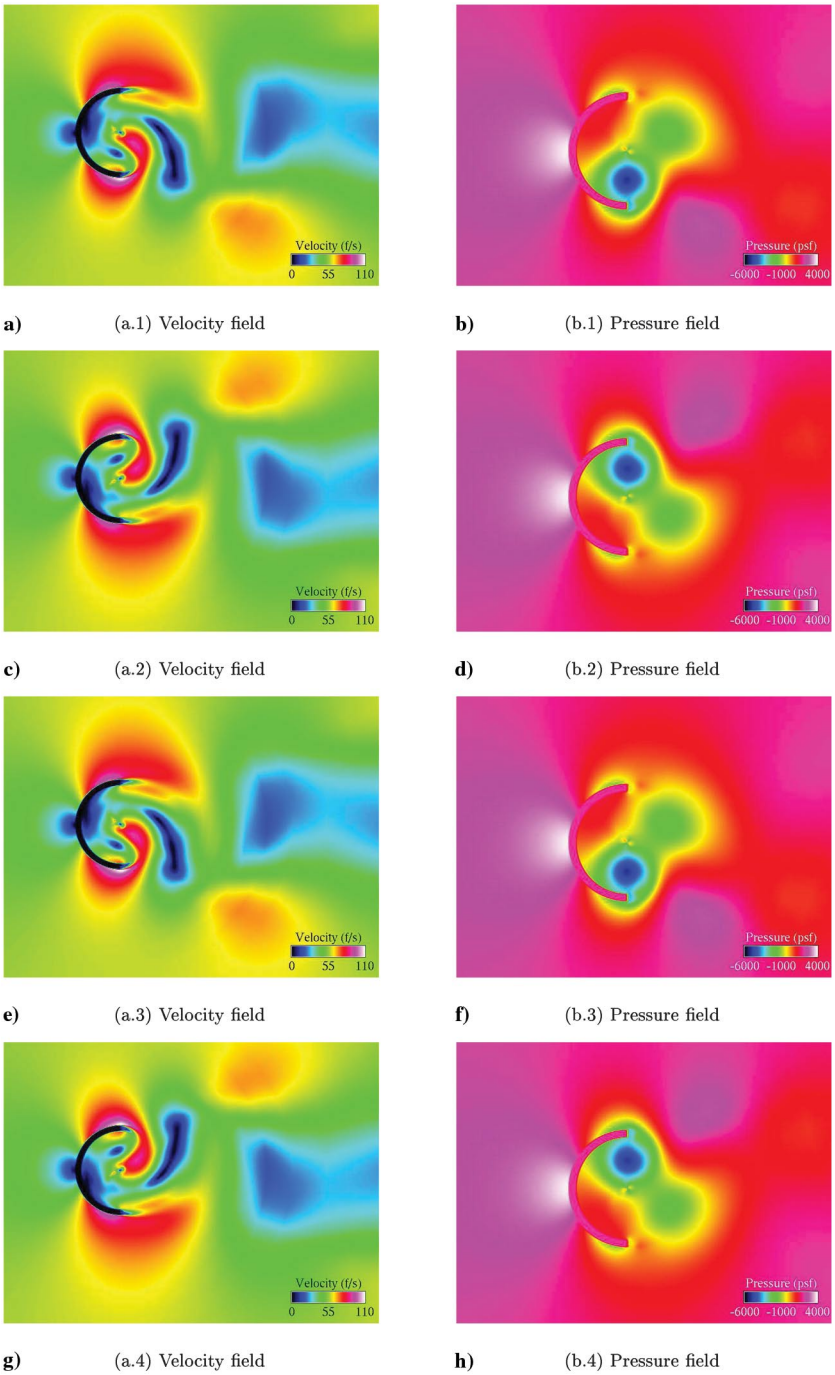

Fig. 12 Velocity and pressure fields.

VI. Conclusions

A new algorithm, RK-SIMPLER, has been developed for incompressible flows that establishes that the entire equation set for the incompressible flow reduces to the nonlinear pressure–velocity coupled Poisson equation for pressure derived from the mass conservation equation. The momentum equations are reduced to ordinary differential equations and are integrated in time using Runge–Kutta stages with pressure field held constant through the stages. The resulting explicit equations are used to update the velocity field once every time step without the need to solve nonlinear momentum equations. The RK-SIMPLER algorithm uses all exact equations and does not require a pressure correction equation or relaxation factors. Also, there is no need for subiterations within a time step. Hence, the explicit RK-SIMPLER algorithm proves to be inherently economical where explicit algorithms are viable. The RK-SIMPLER algorithm, as presented, allows existing codes based on SIMPLE and SIMPLER to be modified easily to adopt to the new algorithm. The RK-SIMPLER algorithm is shown to work well for both internal low-Reynolds-number steady flows and external high-Reynolds-number unsteady flows with good temporal accuracy.

Acknowledgments

The authors acknowledge the contributions made by Prabir Purohit in the initial development of this algorithm. The authors express their gratitude to graduate students Daniel Garrick, Avinaash Murali, and Matthew Fischels for their suggestions and help in preparation of this manuscript.

References

[1] Gosman, A. D., Pun, W. M., Runchal, A. K., Splading, D. B., and Wolfshtein, M., "Heat and Mass Transfer in Recirculating Flows," Academic Press, New York, 1969, pp. 43–52.

[2] Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow problems," *Journal of Computational Physics*, Vol. 53, No. 12, 1967, pp. 12–26. doi:10.1016/0021-9991(67)90037-X

[3] Chorin, A. J., and Marsden, J. E., *A Mathematical Introduction to Fluid Mechanics*, 3rd ed., Springer–Verlag, New York, 1993, pp. 51–55.

[4] Soh, W. Y., and Goodrich, J. W., "Unsteady Solution of Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 79, No. 12, 1988, pp. 113–134. doi:10.1016/0021-9991(88)90007-1

[5] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publ., New York, 1980, pp. 126–134.

[6] Van Doormal, J. P., and Raithby, G. D., "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows," *Numerical Heat Transfer*, Vol. 7, No. 2, 1984, pp. 147–163. doi:10.1080/01495728408961817

[7] Ozoe, H., and Tao, W.-Q., "A Modified Pressure-Correction Scheme for the SIMPLER Method, MSIMPLER," *Numerical Heat Transfer Part B: Fundamentals*, Vol. 39, No. 5, 2001, pp. 435–449. doi:10.1080/104077901750188831

[8] Tao, W., Qu, Z., and He, Y., "A Novel Segregated Algorithm for Incompressible Fluid Flow and Heat Transfer Problems—CLEAR (Coupled and Linked Equations Algorithm Revised) Part 1: Mathematical Formulation and Solution Procedure," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 45, No. 12, 2004, pp. 1–17. doi:10.1080/10407790490254855

[9] Cheng, Y. P., Lee, T. S., Low, H. T., and Tao, W. Q., "Improvement of SIMPLER Algorithm for Incompressible Flow on Collocated Grid System," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 51, No. 5, 2007, pp. 463–486. doi:10.1080/10407790600939767

[10] Sun, D. L., Qu, Z. G., and Tao, W. Q., "An Efficient Segregated Algorithm for Incompressible Fluid Flow and Heat Transfer Problems—IDEAL (Inner Doubly-Iterative Efficient Algorithm for Linked-Equations) Part 1: Mathematical Formulation and Solution Procedure," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 53, No. 12, 2008, pp. 1–17. doi:10.1080/10407790701632543

[11] Reggio, M., and Camarero, R., "Numerical Solution Procedure for Viscous Incompressible Flows," *Numerical Heat Transfer, Part A: Applications*, Vol. 10, No. 2, 1986, pp. 131–146. doi:10.1080/10407788608913512

[12] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," *14th Fluid and Plasma Dynamics Conference*, AIAA Paper 1981-1259, June 1981.

[13] Jorgenson, P. C. E., and Chima, R. V., "An Explicit Runge–Kutta Method for Unsteady Rotor/Stator Interaction," *26th Aerospace Sciences Meeting*, AIAA Paper 1988-0049, Jan. 1988; also NASA TM 100787.

[14] Purohit, P., "Development of an Explicit Time Accurate Scheme for Incompressible Flows," M.S. Thesis, Iowa State Univ., Ames, IA, 2001.

[15] Kelkar, K. M., and Patankar, S. V., "Numerical Prediction of Vortex Shedding Behind a Square Cylinder," *International Journal for Numerical Methods in Fluids*, Vol. 14, No. 3, 1992, pp. 327–341. doi:10.1002/(ISSN)1097-0363

[16] Sutikno, W., "Unsteady Simulation of Two- and Three-Dimensional Flows," M.S. Thesis, Iowa State Univ., Ames, IA, 1994.

[17] Lestari, A., "Development of Unsteady Algorithms for Pressure-Based Unstructured Solver for Two-Dimensional Incompressible Flows," M.S. Thesis, Iowa State Univ., Ames, IA, 2009.

[18] Ijaz, M., and Anand, N. K., "Simulation of Unsteady Incompressible Viscous Flow Using Higher-Order Implicit Runge–Kutta Methods—Staggered Grid," *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 52, No. 6, 2007, pp. 471–488. doi:10.1080/10407790701563367

[19] Sanderse, B., and Koren, B., "New Explicit Runge–Kutta Methods for the Incompressible Navier–Stokes Equations," *Proceedings of the 7th International Conference on Computational Fluid Dynamics (ICCFD7)*, ICCFD7-1403, Big Island, HI, July 2012.

[20] Cabuk, H., Sung, C.-H., and Modi, V., "Explicit Runge–Kutta Method for Three-Dimensional Internal Incompressible Flows," *AIAA Journal*, Vol. 30, No. 8, 1992, pp. 2024–2031. doi:10.2514/3.11175

[21] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Dynamics and Heat Transfer*, McGraw–Hill, New York, 1984, pp. 120–122.

[22] Fischels, M. V., and Rajagopalan, R. G., "Comparison of Pressure Based Runge–Kutta Schemes for Unsteady Incompressible Flows," *22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-3203, June 2015.

[23] Garrick, D. P., and Rajagopalan, R. G., "An Explicit Pressure-Based Algorithm for Incompressible Flows," *22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-3201, June 2015.

[24] Thistle, J., "Development of an Explicit Pressure-Based Unstructured Solver for Three-Dimensional Incompressible Flows with Graphics Hardware Acceleration," M.S. Thesis, Iowa State Univ., Ames, IA, 2012.

[25] Fuchs, L., and Tillmark, N., "Numerical and Experimental Study of Driven Flow in a Polar Cavity," *International Journal for Numerical Methods in Fluids*, Vol. 5, No. 4, 1985, pp. 311–329. doi:10.1002/(ISSN)1097-0363

W. K. Anderson
Associate Editor