Time-stepping techniques for the incompressible Navier-Stokes equations

Strategies and methods for Computational Fluid Dynamics

Introduction

A great deal of computational research has been undertaken and published in the field of Computational Fluid Dynamics (CFD) since the advent of the digital computer. Before 1970, the Finite Difference Method (FDM) was almost universally used as a computer based numerical method in modeling fluid dynamics process [xyz0000]. Since then there has been a revolution in the general area of mathematical modeling. Highly sophicticated and detailed analysis of many engineering problems has become possible. However, it can be argued that the last three decades have in many ways belonged to the Finite Element Method (FEM) as the method of choice among the currently available numerical methods for solving mathematical equations [Hue1975, Hin1979]. Fluid dynamics being one of the oldest branches of physics, has consequently been one of the main arenas of activity for researchers and practitioners of FEM. Despite the continued use of FDM and related techniques for routine fluid dynamics problems, FEM is increasingly the preferred numerical method for analysis of the most complex types of flow problems with unrivalled accuracy [Hua1999].

The vast majority of CFD related research has concentrated on compressible or incompressible Newtonian fluids flow [Zie2000c, Tay1981]. Such fluids have a constant viscosity which is independent of the velocity gradient, temperature or the other quantities. This mean that the stress in Newtonian fluids is proportional to the rate of shear. There exist however, a fairly large category of fluids for which the viscosity is not independent of the rate shear and these fluids are referred to as non-Newtonian. Exact solutions for non-Newtonian flows are practically impossible. The necessitates the use of numerical methods for obtaining approximate solutions to most non-Newtonian flow problems.

Efficient and reliable numerical solution of the incompressible Navier-Stokes equations for industrial flow is extremely challenging. Very rapid changes in the velocity field may take place in thin boundary layers close to solid walls. Complex geometries can also lead to rapid local changes in the velocity. Locally refined grids, preferably in combination with error estimation and automatic grid adaption, are hence a key ingredient in robust methods. Most implicit solution methods for the Navier-Stokes equations end up with saddle-point problems, which complicates the construction of efficient iterative methods for solving the linear systems arising from the discretization process. Implicit solution methods also make a demand for solving large systems of nonlinear algebraic equations.

Many incompressible viscous flow computations involve large-scale flow applications with several million grid points and thereby a need for the next generation of super-computers before becoming engineering or scientific practice. We have also mentioned that Navier-Stokes solvers are often embedded in much more complex flow models, which couple turbulence, heat transfer, and multi-specie fluids. Before attacking such complicated problems it is paramount that the numerical state-of-the-art of Navier-Stokes solvers is satisfactory. Turek [Tur1996] summarizes the results of benchmarks that were used to assess the quality of solution methods and software for unsteady flow around a cylinder in 2D and 3D. The discrepancy in results for the lifting force shows that more research is needed to develop suficiently robust and reliable methods.

Numerical methods for incompressible viscous flow is a major part of the rapidly growing field computational fluid dynamics (CFD). CFD is now emerging as an operative tool in many parts of industry and science. However, CFD is not a mature field either from a natural scientist's or an application engineer's point of view; robust methods are still very much under development, many different numerical tracks are still competing, and reliable computations of complex multi-fluid flows are still (almost) beyond reach with today's methods and computers. We believe that at least a couple of decades of intensive research are needed to merge the seemingly different solution strategies and make them as robust as numerical models in, e.g., elasticity and heat conduction. Sound application of CFD today therefore requires advanced knowledge and skills both in numerical methods and fluid dynamics. To gain reliability in simulation results, it should be a part of common practice to compare the results from different discretizations, not only varying the grid spacings but also changing the discretization type and solution strategy. This requires a good overview and knowledge of different numerical techniques. Unfortunately, many CFD practitioners have a background from only one "numerical school" practicing a particular type of discretization technique and solution approach. One goal of the present paper is to provide a generic overview of the competing and most dominating methods in the part of CFD dealing with laminar incompressible viscous flow.

There are a number of FE techniques which can be used to model steady and transient flow.

Approximate solution strategies for time-dependent problems

X.1. Introduction

Writing a complete review of numerical methods for the Navier-Stokes equations is probably an impossible task. The literature on numerical solutions of the Navier-Stokes equations is overwhelming, and only a small fraction of the strategies is cited in this work. These strategies include modern stabilization techniques (pressure stabilization), penalty methods, artificial compressibility, and operator splitting techniques (explicit schemes, implicit velocity step). The latter family of strategies is popular and widespread and are known under many names in the literature, e.g., projection methods and pressure (or velocity) correction methods [Hua1999]. We end the overview of operating splitting methods with a framework where such methods can be viewed as special preconditioners in an iterative scheme for a fully implicit formulation of the Navier-Stokes equations.

Our focus is to present the basic ideas of the most fundamental solution techniques for the Navier-Stokes equations in a form that is accessible to a wide audience. We consider approximate solution strategies where the Navier-Stokes equations are transformed to more common and tractable systems of partial differential equations. We will present the augmented Lagrangian method, basic operator-splitting algorithm for Newtonian and non-Newtonian fluids.

The most of the numerical strategies are based on discretizing governing equations first in time, to get a set of simpler partial differential equations, and then discretizing the time-discrete equations in space. One fundamental difficulty with the this approach is that we derive a second-order Poisson equation for the pressure itself or a pressure increment. Such a Poisson equation implies a demand for more boundary conditions for p than what is required in the original system.

X.2. A naive derivation of schemes for transient heat diffusion. The backward Euler method and the Crank–Nicolson.

In this subsection, we consider the transient heat diffusion equation

$$\frac{\partial T}{\partial t} = \nabla^2 T \qquad \text{(x.y)}$$

The first method uses literal timestepping and the second is based on timestepping using differences.

The backward Euler method uses the algorithm

$$\frac{T_{n+1} - T_n}{\delta t} = \nabla^2 T_{n+1} \qquad \text{(x.y)}$$

which is equivalent to

$$T_{n+1} - \delta t \nabla^2 T_{n+1} = T_n \ . \qquad \text{(x.y)}$$

In the heart of the algorithm, the equation (x.y) is assembled and solved at each timestep.

Let us now consider another algorithm for solving the heat equation. Firstly we introduce the Crank–Nicolson approximation

$$\frac{T_{n+1} - T_n}{\delta t} = \frac{1}{2} \nabla^2 \left(T_{n+1} + T_n \right) \qquad \text{(x.y)}$$

Next define $\delta T = T_{n+1} - T_n$ and verify that δT satisfies

$$\delta T - \frac{\delta t}{2} \nabla^2 \delta T = \delta t \nabla^2 T_n \, . \qquad \text{(x.y)}$$

You probably realise that the algorithms in this subsection are computationally inefficient in that they assemble and solve a sparse matrix system at each timestep. It would be much more efficient to assemble a large sparse matrix system once and only once, factorise it, store the factors, and use them in subsequent timesteps.

X.3. Basic iterative scheme for two-phase flow of non-Newtonian fluids

In this chapter a method for analyzing transient two-phase non-Newtonian flow is presented. We begin with the Navier-Stokes equation for non-Newtonian fluids represented by conservation of momentum

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) + \nabla p - \nabla \cdot \mathbf{S} = \mathbf{f} \ . \qquad \text{(x.y)}$$

and conservation of mass

$$\nabla \cdot \mathbf{v} = 0 \, . \qquad \text{(x.y)}$$

Let us consider two-phase flow of non-Newtonian fluids. The fluids are identified by the different value of the colour function *C*, which is convected by the flow field

$$\frac{\partial C}{\partial t} + (\mathbf{v}\cdot\nabla)C = 0\,. \qquad \text{(x.y)}$$

Fluid properties such as the density and the viscosity are assumed to be distributed in the same manner as *C*, i.e.

$$\rho = \rho_1 + \frac{\rho_2 - \rho_1}{C_2 - C_1}(C - C_1) \qquad \text{(x.y)}$$

and

$$\eta = \eta_1 + \frac{\eta_2 - \eta_1}{C_2 - C_1}(C - C_1)\,. \qquad \text{(x.y)}$$

Constitutive equation (extra-stress tensor) for non-Newtonian fluids we can write in the following form $\mathbf{S} = \eta(\gamma)\mathbf{G}$, where $\mathbf{G}$ is the rate of strain tensor (rate of deformation tensor), $\mathbf{G} = \nabla\mathbf{v} + (\nabla\mathbf{v})^T$, and $\eta(\dot\gamma)$ is the viscosity, and $\dot\gamma$ is a scalar measure of rate of strain tensor defined by $\dot\gamma = \sqrt{\frac{1}{2}tr(\mathbf{GG})}$.

To solve transient two-phase flow of non-Newtonian fluids the above equations are linearized. Let *r* denote the number of linearization iterations and the superscript *n* denotes the previous time step and *n+1* denotes the current time step. The linearization is carried out as follows:

$$\nabla\cdot(\mathbf{v})_{n+1}^{r} = 0 \qquad \text{(x.y)}$$

$$\begin{aligned}(\rho)_{n+1}^{r-1}&\left(\frac{(\mathbf{v})_{n+1}^{r} - \mathbf{v}_n}{\delta t} + (\mathbf{v})_{n+1}^{r-1}\cdot\nabla(\mathbf{v})_{n+1}^{r}\right)\\ &+ \nabla(p)_{n+1}^{r} - (\eta)_{n+1}^{r-1}\nabla^2(\mathbf{v})_{n+1}^{r} \qquad \text{(x.y)}\\ &- \nabla(\eta)_{n+1}^{r-1}\left(\nabla(\mathbf{v})_{n+1}^{r-1} + \left(\nabla(\mathbf{v})_{n+1}^{r-1}\right)^T\right) = (\mathbf{f})_{n+1}^{r-1}\end{aligned}$$

$$\frac{(C)_{n+1}^{r} - C_n}{\delta t} + \left((\mathbf{v})_{n+1}^{r}\cdot\nabla\right)(C)_{n+1}^{r} = 0 \qquad \text{(x.y)}$$

Rearranging terms we obtain:

$$\nabla \cdot (\mathbf{v})_{n+1}^{r} = 0 \tag{x.y}$$

$$(\rho)_{n+1}^{r-1}\left(\frac{(\mathbf{v})_{n+1}^{r}}{\delta t} + (\mathbf{v})_{n+1}^{r-1}\cdot\nabla(\mathbf{v})_{n+1}^{r}\right) + \nabla(p)_{n+1}^{r} - (\eta)_{n+1}^{r-1}\nabla^{2}(\mathbf{v})_{n+1}^{r} =$$
$$= (\mathbf{f})_{n+1}^{r-1} + (\rho)_{n+1}^{r-1}\frac{\mathbf{v}_n}{\delta t} + \nabla(\eta)_{n+1}^{r-1}\left(\nabla(\mathbf{v})_{n+1}^{r-1} + \left(\nabla(\mathbf{v})_{n+1}^{r-1}\right)^{T}\right) \tag{x.y}$$

$$\frac{(C)_{n+1}^{r}}{\delta t} + \left((\mathbf{v})_{n+1}^{r}\cdot\nabla\right)(C)_{n+1}^{r} = \frac{C_n}{\delta t} \tag{x.y}$$

In the above:

$$(\mathbf{v})_{n+1}^{0} = \mathbf{v}_n,\ (p)_{n+1}^{0} = p_n,\ (C)_{n+1}^{0} = C_n,\ (\eta)_{n+1}^{0} = \eta_n,\ (\rho)_{n+1}^{0} = \rho_n. \tag{x.y}$$

In three dimensional case we can write equation (x.y) for i-th component of velocity vector in the following form

$$\frac{\partial(v_1)_{n+1}^{r}}{\partial x_1} + \frac{\partial(v_2)_{n+1}^{r}}{\partial x_2} + \frac{\partial(v_2)_{n+1}^{r}}{\partial x_2} = 0 \tag{x.y}$$

$$(\rho)_{n+1}^{r-1}\left(\frac{(v_i)_{n+1}^{r}}{\delta t} + (\mathbf{v})_{n+1}^{r-1}\cdot\nabla(v_i)_{n+1}^{r}\right) + \frac{\partial(p)_{n+1}^{r}}{\partial x_i} - (\eta)_{n+1}^{r-1}\nabla^{2}(v_i)_{n+1}^{r} =$$
$$= (f_i)_{n+1}^{r-1} + (\rho)_{n+1}^{r-1}\frac{(v_i)_n}{\delta t} + \nabla(\eta)_{n+1}^{r-1}(G_{ij})_{n+1}^{r-1} \tag{x.y}$$

The set of nonlinear simultaneous equations is solved by a siutable iterative process in which a simple convergence sequence and method of variable updating is employed. The procedure can be summarised in few steps.
First we solve set of equations for unknown value $(v_i)_{n+1}^{r}$ (similarly for all components of velocity vector). Next we evaluate

$$\frac{(v_i)_{n+1}^{r} - (v_i)_{n+1}^{r-1}}{(v_i)_{n+1}^{r}} \tag{x.y}$$

at all node points. If these are within a specified tolerance, *TOL*, at all points then assume that the calculation is complete for time equal t_{n+1}. If the differences do not come within tolerance then update $(v_i)_{n+1}^r$

$$(v_i)_{n+1}^r = (1-\omega)(v_i)_{n+1}^{r-1} + \omega(v_i)_{n+1}^r \qquad \text{(x.y)}$$

where ω is the weighting factor. For the simple arithmetic mean $\omega = 0.5$.
The process is repeated (*r* is increased) until *TOL* is satisfied at all points within domain and on all boundary points subject to gradient boundary conditions.
We can apply mentioned algorithm for all t_{n+1}.

X.4. Pressure correction scheme for non-isothermal flow

The pressure correction scheme decouples the velocity and pressure terms of the momentum equations and implies the consideration of a Poisson equation for the pressure at each time step [Hua1999].

Beginning with the dimensionless incompressible Navier-Stokes equations

$$\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) + \nabla p - Pr\,\nabla\cdot\mathbf{S} = \mathbf{f}\,. \qquad \text{(x.y)}$$

and conservation of mass

$$\nabla\cdot\mathbf{v} = 0\,. \qquad \text{(x.y)}$$

where $\mathbf{S} = \mathbf{S}(\mathbf{v}) = \eta(\mathbf{v})\mathbf{G}(\mathbf{v})$ is the extra stress tensor.

With the midpoint rule or Crank Nicolson scheme we can write

$$\frac{1}{\delta t}(\mathbf{v}_{n+1} - \mathbf{v}_n) = Pr\,\nabla\cdot\mathbf{S}_n - \mathbf{v}_n\cdot\nabla\mathbf{v}_n - \nabla p_{n+1} + \mathbf{f}_n \qquad \text{(x.y)}$$

$$\nabla\cdot\mathbf{v}_{n+1} = 0\,. \qquad \text{(x.y)}$$

According to the projection concept we can always find an intermediate velocity field $\mathbf{v}^*$ which may satisfy the both following equations

$$\frac{1}{\delta t}(\mathbf{v}^* - \mathbf{v}_n) = Pr\,\nabla\cdot\mathbf{S}_n - \mathbf{v}_n\cdot\nabla\mathbf{v}_n + \mathbf{f}_n \qquad \text{(x.y)}$$

and

$$\frac{1}{\delta t}\left(\mathbf{v}_{n+1}-\mathbf{v}^{*}\right)=-\nabla p_{n+1}\,. \qquad \text{(x.y)}$$

Applying ∇ operator to the both sides of above equations and considering the equation of conservation of mass we obtain

$$\frac{1}{\delta t}\nabla\cdot\mathbf{v}^{*}=-\nabla^{2}p_{n+1}\,. \qquad \text{(x.y)}$$

It is now apart that a three step scheme can be summarized from the above operations, that is

Step 1:

$$\frac{1}{\delta t}\left(\mathbf{v}^{*}-\mathbf{v}_{n}\right)=Pr\nabla\cdot\mathbf{S}_{n}-\mathbf{v}_{n}\cdot\nabla\mathbf{v}_{n}+\mathbf{f}_{n} \qquad \text{(x.y)}$$

Step 2:

$$\nabla^{2}p_{n+1}=-\frac{1}{\delta t}\nabla\cdot\mathbf{v}^{*} \qquad \text{(x.y)}$$

Step 3:

$$\frac{1}{\delta t}\left(\mathbf{v}_{n+1}-\mathbf{v}^{*}\right)=-\nabla p_{n+1}\,. \qquad \text{(x.y)}$$

Such a scheme is specifically designed do deal with the incompressibility constraint and introduces a Poisson equation for the pressure at each time step.

Using a semi-implicit pressure correction scheme we can obtain the following steps

Step 1a:

$$\frac{2}{\delta t}\left(\mathbf{v}_{n+1/2}-\mathbf{v}_{n}\right)=\frac{Pr}{2}\nabla\cdot\mathbf{S}_{n+1/2}+\frac{Pr}{2}\nabla\cdot\mathbf{S}_{n}-\mathbf{v}_{n}\cdot\nabla\mathbf{v}_{n}-\nabla p_{n}+\mathbf{f}_{n} \qquad \text{(x.y)}$$

Step 1b:

$$\frac{1}{\delta t}\left(\mathbf{v}^* - \mathbf{v}_n\right) = \frac{Pr}{2}\nabla\cdot\mathbf{S}^* + \frac{Pr}{2}\nabla\cdot\mathbf{S}_n - \mathbf{v}_{n+1/2}\cdot\nabla\mathbf{v}_{n1/2} - \nabla p_n + \mathbf{f}_n \qquad \text{(x.y)}$$

Step 2:

$$\theta\nabla^2 q_{n+1} = -\frac{1}{\delta t}\nabla\cdot\mathbf{v}^* \qquad \text{(x.y)}$$

Step 3:

$$\frac{1}{\delta t}\left(\mathbf{v}_{n+1} - \mathbf{v}^*\right) = -\nabla q_{n+1} \qquad \text{(x.y)}$$

where $q_{n+1} = p_{n+1} - p_n$ and $\theta = 1/2$, $\mathbf{v}_{n+1/2}$ is a half step velocity field.

For incompressible flow under non-isothermal conditions, the energy conservation equation must also be included in the formulation

$$\frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T = \nabla^2 T + PrEc\,\eta\Phi\,. \qquad \text{(5.8)}$$

In the non-isothermal case a semi-implicit pressure correction procedure can be modified by adding following formulas in Step 1a and 1b

Step 1a:

$$\frac{2}{\delta t}\left(T_{n+1/2} - T_n\right) = \nabla^2 T_n - \mathbf{v}_n\cdot\nabla T_n + PrEc\,\eta\Phi_n \qquad \text{(x.y)}$$

Step 1b:

$$\frac{1}{\delta t}\left(T_{n+1} - T_n\right) = \nabla^2 T_{n+1/2} - \mathbf{v}_{n+1/2}\cdot\nabla T_{n+1/2} + PrEc\,\eta\Phi_{n+1/2}\,. \qquad \text{(x.y)}$$

X.5. 6. The augmented Lagrangian method

The flow equations have the difficulty that the pressure *p*, required in the momentum equations, does not occur explicitly in the continuity equation. Rather, the continuity equation acts like a constraint to the momentum equations, and this constraint determines the pressure. The augmented Lagrangian method is a refinement of the penalty method [Hua1999]. In the penalty method, a fictitious representation for the pressure in the continuity equation is introduced

$$p = -Pen(\nabla \cdot \mathbf{v}). \tag{6.1}$$

If *Pen* is large, then $\nabla \cdot \mathbf{v}$ is forced to be small, therefore approximately satisfying the continuity equation.

We introduce a timestepping algorithm to solve the above equations.

Step 1. In each timestep, we first solve thermal diffusion equation by assuming a given velocity field and using an implicit method to handle the time derivative. We have

$$\frac{\theta_n - \theta_{n-1}}{\delta t} + \mathbf{v}_{n-1} \cdot \nabla \theta_n - \mathbf{v}_{n-1} \cdot \mathbf{k} = \nabla^2 \theta_n + PrEc\eta_{n\text{-}1}\Phi_{n-1} \tag{6.2}$$

which can be expressed as a PDE for θ_n, where $\theta_n = \theta(t_n)$, $\mathbf{v}_n = \mathbf{v}(t_n)$ and n denotes n-th time step.

Step 2. The momentum equation is then solved using an implicit method to handle the time derivative $\frac{\partial \mathbf{v}}{\partial t}$ and the augmented Lagrangian method to handle the incompressibility constraint. The second step of actual algorithm that is implemented is obtained by setting

$$p_n^{(i)} = p_n^{(i-1)} + \Delta p = p_n^{(i-1)} - Pen\left(\nabla \cdot \mathbf{v}_n^{(i)}\right) \tag{6.3}$$

in the momentum equation.
This results in the iterative algorithm:

- obtain the $\mathbf{v}_n^{(i)}$ by solving the equation:

$$\begin{aligned} &\frac{\mathbf{v}_n^{(i)} - \mathbf{v}_{n-1}}{\delta t} + \mathbf{v}_n^{(i-1)} \cdot \nabla \mathbf{v}_n^{(i)} = Pen\left(\nabla\left(\nabla \cdot \mathbf{v}_n^{(i)}\right)\right) + \\ &-\nabla p_n^{(i-1)} + Pr\,\nabla \cdot \mathbf{S}_n^{(i-1)} + Ra\,Pr\,\theta_n \mathbf{k} = \mathbf{0} \end{aligned} \tag{6.4}$$

with $\mathbf{v}_n^{(0)} = \mathbf{v}_{n-1}$ and $p_n^{(0)} = p_{n-1}$;

- calculate the Δp using the equation:

$$\Delta p + Pen\left(\nabla \cdot \mathbf{v}_n^{(i)}\right) = 0\,; \tag{6.5}$$

- the pressure is updated using:

$$p_n^{(i)} = p_n^{(i-1)} + \Delta p\,. \tag{6.6}$$

During each timestep, several iterations (for example $i\max$) of the solution of the discretised momentum equation are required in order to obtain a converged velocity field. The number of iteration is pointed by superscript of velocity vector and pressure term.

<u>Step 3.</u> Update:

$$t_n = t_{n-1} + \delta t\,,\ \mathbf{v}_n = \mathbf{v}_n^{(0)} = \mathbf{v}_{n-1}^{(i\max)} \text{ and } p_n^{(0)} = p_{n-1} = p_{n-1}^{(i\max)}\,. \tag{6.7}$$

The algorithm, if it converges, does not introduce any further error and provides an answer for the pressure. It can be implemented using finite element method [Hue1975, Tay1981].

X.6. Basic operator-splitting algorithm for Newtonian fluid

The most popular numerical solution strategy today for the Navier-Stokes equations are based on operator-splitting [Li1991; Li1993; Luo1996]. This means that the system is split into a series of simpler, familiar equations, such as advection equations, diffusion equations, advection-diffusion equations, Poisson equations, and explicit or implicit updates. Efficient numerical methods are much easier to construct for these standard equations that for the original system directly. In particular, the evolution of the velocity consists of two main steps. First we neglect the incompressibility condition and compute a predicted velocity. Thereafter, the velocity is corrected by performing a projection onto the divergence free vector field.

The Navier–Stokes momentum equation is:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla \cdot \nabla \mathbf{v}\,. \tag{7.1}$$

For the unsteady Navier–Stokes equation, we solve (7.1) as an initial value problem, that is, at time step *n*, we know the velocity value $\mathbf{v}_n$, and we obtain the velocity at time step n+1 by solving for the velocity increment $\delta\mathbf{v}$ so that $\mathbf{v}_{n+1} = \mathbf{v}_n + \delta\mathbf{v}$. From the Navier–Stokes equation, we see that $\delta\mathbf{v}$ satisfyies:

$$\begin{aligned}&\frac{\partial \delta\mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\delta\mathbf{v} - \nabla\cdot\nabla\delta\mathbf{v} = \\ &-\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right] - \nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\nabla\mathbf{v}\end{aligned} \tag{7.2}$$

where due to the nonlinear convection term, the $\mathbf{v}$ on the RHS is the latest updated velocity value from $\mathbf{v}_n$, that is $\mathbf{v} = \mathbf{v}_n + \delta\mathbf{v}$. The pressure p will be solved through the continuity equation in a way that is explained later.
The left-hand side of (7.2) can be split into two principal operators as follows:

$$\begin{aligned}&\frac{\partial \delta\mathbf{v}}{\partial t} + L_c(\delta\mathbf{v}) - L_d(\delta\mathbf{v}) = \\ &-\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right] - \nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\nabla\mathbf{v}\end{aligned} \tag{7.3}$$

where $L_c(\delta\mathbf{v})$ and $L_d(\delta\mathbf{v})$ are the convective and diffusive operators, respectively. This equation for velocity increment can be split as follows:

$$\frac{\partial \delta\mathbf{v}^*}{\partial t} + L_c\left(\delta\mathbf{v}^*\right) = -\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right] - \nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\nabla\mathbf{v} \tag{7.4}$$

$$\mathbf{v}^* = \mathbf{v} + \delta\mathbf{v}^* \tag{7.5}$$

$$\frac{\partial \delta\mathbf{v}}{\partial t} - L_d(\delta\mathbf{v}) = -\left[\frac{\partial \mathbf{v}^*}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}^*\right] - \nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\nabla\mathbf{v}^* \tag{7.6}$$

We can set up numerical schemes to integrate equations (7.4) and (7.6) respectively. We define the numerical scheme for equation (7.4) as N_c and the scheme for equation (7.6) as N_d. Equation (7.3) is advanced from time step n to n+1 by the two-stage convection-diffusion split:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + N_d N_c \delta\mathbf{v} \tag{7.7}$$

Alternatively we can set up a symmetric sequence of the numerical schemes to have

$$\mathbf{v}_{n+2} = \mathbf{v}_n + N_c N_d N_d N_c \delta\mathbf{v} \tag{7.8}$$

In our Navier–Stokes equation, the convective operator L_c is nonlinear. Therefore, a local iteration loop is needed for equation (7.8). The numerical scheme N_c will be a fully implicit backward Crank–Nicolson method. The three components of velocity vector $\mathbf{v}$ can be solved separately by N_c. In this way, less computer memory is required for inverting the linear matrices.

For the *i*-th component of vector $\mathbf{v}$, v_i, N_c can be shown to be

$$\begin{aligned} \frac{\delta v_i^{(k)}}{\delta t} + \mathbf{v}\cdot\nabla\delta v_i^{(k)} = &-\left[\frac{\sum_{j=0}^{k-1}\delta v_i^{(j)}}{\delta t} + \mathbf{v}\cdot\nabla v_{i,n+1}^{(k)}\right] \\ &-\frac{\partial p}{\partial x_i} + Ra\,Pr\,\theta\,k_i + Pr\,\nabla\cdot\nabla v_{i,n+1}^{(k)} \end{aligned} \tag{7.9}$$

where *k* denotes the *k*-th iterative step for the nonlinear convective operator, n denotes n-th time step , and

$$v_{i,n+1}^{(k+1)} = v_{i,n+1}^{(k)} + \delta v_i^{(k)} \tag{7.10}$$

Similarly, the numerical scheme N_d can be set up for the *i*-th component in approach:

$$\begin{aligned} \frac{\delta v_i^{(k)}}{\delta t} - \nabla\cdot\nabla\delta v_i^{(k)} = &-\left[\frac{\sum_{j=0}^{k-1}\delta v_i^{(j)}}{\delta t} + \mathbf{v}\cdot\nabla v_{i,n+1}^{(k)}\right] \\ &-\frac{\partial p}{\partial x_i} + Ra\,Pr\,\theta\,k_i + Pr\,\nabla\cdot\nabla v_{i,n+1}^{(k)} \end{aligned} \tag{7.11}$$

where $v_{i,n+1}$ is always updated at each step through (7.10).

X.7. Basic operator-splitting algorithm for non-Newtonian fluid

The Navier–Stokes momentum equation for non-Newtonian fluid is:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} = -\nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\mathbf{S}\,. \tag{8.1}$$

If we consider extra stress tensor $\mathbf{S}$ as a function of the deformation rate tensor $\mathbf{G}$ and viscosity μ as a function of velocity $\mathbf{v}$ we can write

$$\mathbf{S}(\mathbf{v}) = \eta(\mathbf{v})\,\mathbf{G}(\mathbf{v}) \tag{8.2}$$

For the unsteady Navier–Stokes equation, we solve (8.1) as an initial value problem, that is, at time step *n*, we know the velocity value $\mathbf{v}_n$, and we obtain the velocity at time step *n*+1 by solving for the velocity increment $\delta\mathbf{v}$ so that $\mathbf{v}_{n+1} = \mathbf{v}_n + \delta\mathbf{v}$. From the Navier–Stokes equation, we see that $\delta\mathbf{v}$ satisfies:

$$\begin{aligned}&\frac{\partial \delta\mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\delta\mathbf{v} - Pr\,\nabla\cdot\left(\eta(\mathbf{v})\,\mathbf{G}(\delta\mathbf{v})\right) = -\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right]\\ &-\nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\left(\eta(\mathbf{v})\,\mathbf{G}(\mathbf{v})\right)\end{aligned} \tag{8.3}$$

where due to the nonlinear convection term, the $\mathbf{v}$ on the RHS is the latest updated velocity value from $\mathbf{v}_n$, that is $\mathbf{v} = \mathbf{v}_n + \delta\mathbf{v}$. The pressure *p* will be solved through the continuity equation in a way that is explained later.
The left-hand side of (8.3) can be split into two principal operators as follows:

$$\begin{aligned}&\frac{\partial \delta\mathbf{v}}{\partial t} + L_c(\delta\mathbf{v}) - L_{ds}(\delta\mathbf{v}) = -\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right]\\ &-\nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\left(\eta(\mathbf{v})\,\mathbf{G}(\mathbf{v})\right)\end{aligned} \tag{8.4}$$

where $L_c(\delta\mathbf{v})$ and $L_{ds}(\delta\mathbf{v})$ are the convective and diffusive operators, respectively. This equation for velocity increment can be split as follows:

$$\frac{\partial \delta\mathbf{v}^*}{\partial t} + L_c\left(\delta\mathbf{v}^*\right) = -\left[\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right] - \nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\left(\eta(\mathbf{v})\,\mathbf{G}(\mathbf{v})\right) \tag{8.5}$$

$$\mathbf{v}^* = \mathbf{v} + \delta\mathbf{v}^* \tag{8.6}$$

$$\frac{\partial \delta\mathbf{v}}{\partial t} - L_{ds}(\delta\mathbf{v}) = -\left[\frac{\partial \mathbf{v}^*}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}^*\right] - \nabla p + Ra\,Pr\,\theta\,\mathbf{k} + Pr\,\nabla\cdot\left(\eta(\mathbf{v})\,\mathbf{G}\left(\mathbf{v}^*\right)\right) \tag{8.7}$$

We can set up numerical schemes to integrate equations (8.5) and (8.7) respectively. We define the numerical scheme for equation (8.5) as N_c and the scheme for equation (8.7) as N_{ds}. Equation (8.4) is advanced from time step *n* to *n*+1 by the two-stage convection-diffusion split:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + N_{ds} N_c \delta\mathbf{v} \tag{8.8}$$

Alternatively we can set up a symmetric sequence of the numerical schemes to have

$$\mathbf{v}_{n+2} = \mathbf{v}_n + N_c N_{ds} N_{ds} N_c \delta\mathbf{v} \tag{8.9}$$

In our Navier–Stokes equation operators L_c and L_{ds} are nonlinear. Therefore, a local iteration loop is needed for equation (8.8-8.9). The numerical scheme N_c will be a fully implicit backward Crank–Nicolson method.

Numerical scheme N_c can be shown to be

$$\begin{aligned} \frac{\delta\mathbf{v}^{(k)}}{\delta t} + \mathbf{v}_{n+1}^{(k)} \cdot \nabla \delta\mathbf{v}^{(k)} = -\left[\frac{\sum_{j=0}^{k-1} \delta\mathbf{v}^{(j)}}{\delta t} + \mathbf{v}_{n+1}^{(k)} \cdot \nabla \mathbf{v}_{n+1}^{(k)} \right] \\ - \nabla p + Ra\, Pr\, \theta\, \mathbf{k} \;\; + Pr \nabla \cdot \left(\eta\left(\mathbf{v}_{n+1}^{(k)}\right) \mathbf{G}\left(\mathbf{v}_{n+1}^{(k)}\right)\right) \end{aligned} \tag{8.10}$$

where *k* denotes the *k*-th iterative step for the nonlinear convective operator, and

$$\mathbf{v}_{n+1}^{(k+1)} = \mathbf{v}_{n+1}^{(k)} + \delta\mathbf{v}^{(k)} \tag{8.11}$$

Similarly, the numerical scheme N_{ds} can be set up in approach:

$$\begin{aligned} \frac{\delta\mathbf{v}^{(k)}}{\delta t} - Pr \nabla \cdot \left(\eta\left(\mathbf{v}_{n+1}^{(k)}\right) \mathbf{G}\left(\delta\mathbf{v}^{(k)}\right)\right) = -\left[\frac{\sum_{j=0}^{k-1} \delta\mathbf{v}^{(j)}}{\delta t} + \mathbf{v}_{n+1}^{(k)} \cdot \nabla \mathbf{v}_{n+1}^{(k)} \right] \\ - \nabla p + Ra\, Pr\, \theta\, \mathbf{k} + Pr \nabla \cdot \left(\eta\left(\mathbf{v}_{n+1}^{(k)}\right) \mathbf{G}\left(\mathbf{v}_{n+1}^{(k)}\right)\right) \end{aligned} \tag{8.12}$$

where $\mathbf{v}_{n+1}$ is always updated at each step through (8.11).

Alternatively the three components of velocity vector $\mathbf{v}$ can be solved separately by N_c. In this way, less computer memory is required for inverting the linear matrices. For the *i*-th component of vector $\mathbf{v}$ numerical schemes N_c can be shown to be

$$\frac{\delta v_i^{(k)}}{\delta t} + \mathbf{v}_{n+1}^{(k)} \cdot \nabla \delta v_i^{(k)} = -\left[\frac{\sum_{j=0}^{k-1} \delta v_i^{(j)}}{\delta t} + \mathbf{v}_{n+1}^{(k)} \cdot \nabla v_{i,n+1}^{(k)} \right] - \frac{\partial p}{\partial x_i} + Ra\,Pr\,\theta\,k_i + Pr\left(\nabla \cdot \left(\eta\left(\mathbf{v}_{n+1}^{(k)}\right)\mathbf{G}\left(\mathbf{v}_{n+1}^{(k)}\right)\right)\right) \cdot \mathbf{e}_i \qquad (8.13)$$

where

$$v_{i,n+1}^{(k+1)} = v_{i,n+1}^{(k)} + \delta v_i^{(k)} \qquad (8.14)$$

Similarly, the numerical scheme N_{ds} can be set up for the *i*-th component in approach:

$$\frac{\delta v_i^{(k)}}{\delta t} - Pr\,\nabla \cdot \left(\eta\left(\mathbf{v}_{n+1}^{(k)}\right)\mathbf{G}\left(\mathbf{v}^*\right)\right)\nabla \delta v_i^{(k)} = -\left[\frac{\sum_{j=0}^{k-1} \delta v_i^{(j)}}{\delta t} + \mathbf{v}_{n+1}^{(k)} \cdot \nabla v_{i,n+1}^{(k)} \right] - \frac{\partial p}{\partial x_i} + Ra\,Pr\,\theta\,k_i + Pr\left(\nabla \cdot \left(\eta\left(\mathbf{v}_{n+1}^{(k)}\right)\mathbf{G}\left(\mathbf{v}_{n+1}^{(k)}\right)\right)\right) \cdot \mathbf{e}_i \qquad (8.15)$$

where $v_{i,n+1}$ is always updated at each step through (8.14).

In this algorithm, we have assumed that pressure *p* is known, and mass conservation is satisfied. The pressure *p* needs to be obtained through an equation that is derived from the mass conservation law. For incompressible fluid flows, this mass conservation law produces the continuity equation

$$\nabla \cdot \mathbf{v} = 0 \tag{8.16}$$

An artificial compressibility method is used here to derive the equation for pressure. From the continuity equation, we can assume that the pressure *p* satisfies a pseudo-transient state

$$\frac{\partial p}{\partial \tau} = -\beta \nabla \cdot \mathbf{v} \tag{8.17}$$

where parameter τ is the pseudo-time. Through a second-order Taylor expansion of the term $\partial p / \partial \tau$ and substitution of the Navier–Stokes momentum equation, we can arrive at a Poisson equation for pressure:

$$\delta p - \beta \frac{(\Delta \tau)^2}{2} \nabla^2 \delta p = -\Delta \tau \, \beta \nabla \cdot \mathbf{v} \tag{8.18}$$

where $\Delta \tau$ is the pseudo-time step.