Use of inexact solution of unsteady Navier-Stokes equations to solve steady state problems

Günter Bärwolff

Technische Universität Berlin, Institut für Mathematik

Abstract. In order to do flow control it is necessary to solve the Navier-Stokes equations using efficient and robust numerical methods. Based on a Finite-Volume spatial discretisation of the Navier-Stokes equation, simple time discretisation methods will be characterised and discussed for the solution of both unsteady problems and of asymptotic steady state problems. An easy-to-implement semi-implicit time integration method of the Navier-Stokes equation will be compared to a Newton method for the solution of the stationary Navier-Stokes equation.

Keywords: Navier-Stokes equation, incompressible flow, FV discretisation, staggered grids, Newton method, semi-implicit time integration
PACS: 45.70.Vn, 47.10.ad, 47.11.-j, 47.11.Df, 47.85.L-

INTRODUCTION

Modeling and numerical simulations of incompressible flow problems are essential in the research fields of fluid dynamics and chemical engineering. For example it is important to know and to manipulate flow fields for an optimisation of vehicle geometries or the design of chemical reactors. In all these cases efficient and adequate numerical solution mehtods are required.

THE MATHEMATICAL PROBLEM

From the mass and impulse balance for an incompressible fluid we get the following partial differential equation system

$$\partial_t y + (y\cdot\nabla)y = -\nabla p + \nu\Delta y + f\ , \tag{1}$$

$$-\nabla\cdot y = 0\ , \tag{2}$$

which is valid in the integration region $\Omega\subset\mathbb{R}^n$ $(n=2,3)$. With appropriate boundary conditions, for example Dirichlet data $y=g$ on $\partial\Omega$, i.e. the boundary of Ω, and an initial condition $y|_{t=0}=y_0$ on $\Omega\cup\partial\Omega$, we have together with (1), (2) an initial boundary value problem for the description of an incompressible flow problem. y and p denote the velocity and the pressure field, ν indicates the kinematic viscosity and f is volume force. By adding a convective heat conduction equation it is possible to describe a thermally coupled flow problem, where the coupling could be realised by f via a Boussinesq approximation.

Spatial discretisation of the problem

In a threedimensional case we use four staggered grids for the spatial discretisation of equations (1) and (2), three staggered grids for the components y_i of the velocity y, and one grid for the pressure p. If ω is a finite volume or in general a finite cell with the boundary $\partial\omega$, we use the theorem of Gauß

$$\int\limits_{\omega}\nabla\cdot v\,dV = \int\limits_{\partial\omega} v\cdot n\,dO\ , \tag{3}$$

in which v denotes a vector field. To use this theorem we write eqution (1) in the so-called divergence or conservative form. For the components of (1) applying the equation (2) we get with $y = (y_1, y_2, y_3)$, $f = (f_1, f_2, f_3)$

$$\partial_t y_i + \nabla \cdot \begin{pmatrix} y_1 y_i \\ y_2 y_i \\ y_3 y_i \end{pmatrix} = -\nabla \cdot \begin{pmatrix} p\delta_{1i} \\ p\delta_{2i} \\ p\delta_{3i} \end{pmatrix} + \nabla \cdot (\nu \nabla y_i) + f_i,\ i = 1,2,3\ , \tag{4}$$

with the Kronecker-delta δ_{ij}. The application of (3) to the equations (4) and (2) leads to a canonical finite volume discretisation of the Navier-Stokes equation, which one can find in [1], [2] or [3]. I will describe the discretised Navier-Stokes equation including the discretised boundary conditions by

$$\partial_t u + C_h(u,u) = -G_h q + V_h u + f_h\ , \quad -D_h u = 0 \tag{5}$$

in which C_h and V_h are the FV-discretisations of the convective and viscous terms of the impulse balance, and G_h, D_h imply the discretisation results for the gradient and divergence operator. $u = (u_1, u_2, u_3)$ is the velocity considered in the centers of the velocity grid cells and q is the pressure located in the center of the pressure grid cells.

The time discretisation

The equation system (5) is a differential algebraic system (DAE), which is not easy to solve. There are special routines, for example DASSL as a MATLAB-tool for DAEs, but these tools are not able to solve non-linear systems of type (5) efficiently.

There are several possible time discretisations of which I will consider here the implicit discretisation

$$\frac{u^{n+1} - u}{\tau} + C_h(u,u) = -G_h q^{n+1} + V_h u^{n+1} + f_h\ , \quad -D_h u^{n+1} = 0 \tag{6}$$

and the semi-implicit discretisation (s. also [1], [2])

$$\frac{u^{n+1} - u}{\tau} + C_h(u,u) = -G_h q^{n+1} + V_h u + f_h\ , \quad -D_h u^{n+1} = 0 \tag{7}$$

with time step parameter τ. The index h stands for a spatial discretisation parameter. The upper index means the time level, i.e. $u^{n+1} = u((n+1)\tau)$. Quantities without an upper index are considered at the time level n. In both discretisations it is necessary to hold a CFL-condition $\|u\|\tau/h \leq 1$ because of the explicitness of the convective term $C_h(u,u)$.

The systems (6) or (7) are saddle point problems of the form

$$\begin{pmatrix} A_h & G_h \\ G_h' & \mathbf{0} \end{pmatrix} \begin{pmatrix} u^{n+1} \\ q^{n+1} \end{pmatrix} = \begin{pmatrix} R_h(u) \\ \mathbf{0} \end{pmatrix} \tag{8}$$

and the operator A_h is a symmetric positive one. In the case of (6) we have

$$A_h = \frac{1}{\tau} Id - V_h \quad \text{and for (7) it is} \quad A_h = \frac{1}{\tau} Id\ .$$

$G_h' = -D_h$ is the transpose or adjoint of G_h and thus the coefficient matrix of (8) is a symmetric one. For the solution of the saddle point problems (8) we apply the operator $G_h' A_h^{-1}$ to the first row of system (8) with the result of

$$G_h' A_h^{-1} G_h q^{n+1} = G_h' A_h^{-1} R_h(u) \tag{9}$$

and the coefficient matrix $S = G_h' A_h^{-1} G_h$ is called Schur complement matrix. Especially in the case of (7), the matrix S is of a simple structure, i.e. we have

$$S = -\tau D_h G_h = -\tau \Delta_h \quad \text{and} \quad r(u) = G_h' A_h^{-1} R_h = -\tau D_h R_h(u)$$

and the equation (9) is a discrete Poisson equation for the pressure

$$S q^{n+1} = r(u)\ . \tag{10}$$

Numerical solution of the equation (8) and (10) respectively

The direct solution of (10) is possible with sparse matrix packages but it is important to note that the Schur complement matrix $S \in \mathbb{R}^{M\times M}$ is singular with the rank $M-1$ because of the Navier-Stokes property of non-uniqueness of the pressure. Only the pressure gradient is unique.

Here, we will discuss an iterative solution method for (10). In [2] and [1] the following method was proposed. (10) means in detail

$$-\tau D_h G_h q^{n+1} = -D_h(u - \tau[C_h(u,u) - \nu V_h u - f_h]) \tag{11}$$

and the idea of [2] and [1] was that of prediction correction method. The first step consists of a velocity prediction

$$\tilde{u} = u - \tau[C_h(u,u) - \nu V_h u - f_h]\ .$$

With the predicted velocity $\tilde{u}$ an iteration for the computation of u^{n+1}, q^{n+1} is initiated. It is obvious that the iteration process

$$\begin{aligned} \operatorname{diag}(S)\delta q^{(s+1)} &= -\omega_r D_h u^{(s)} && (12)\\ q^{(s+1)} &= q^{(s)} + \delta q^{(s+1)}\ ,\ q^{(0)} = 0\ , && (13)\\ u^{(s+1)} &= u^{(s)} - \tau G_h \delta q^{(s+1)}\ ,\ u^{(0)} = \tilde{u}\ , && (14) \end{aligned}$$

provides in the case of convergence with $u^{n+1} = \lim_{s\to\infty} u^{(s)}$ and $q^{n+1} = \lim_{s\to\infty} q^{(s)}$ the solution of the problem (8).

The iteration formulas (12)-(14) are explicit and very easy to implement. Similar to the iterative solution theory of linear equation systems, a single step iteration with relaxation (straightforward SOR-type method) is expected to be more efficient then a Jacobi type method. $\omega_r \in]0,2[$ is a relaxation parameter which works only in the single step iteration. In a Jacobi type method ω_r must be equal to 1. The following pseudo code is an iteration loop for a single step iteration described (2d, equidistant FV grid, spatial grid parameter h, k, rectangular region).

```
...
divp = tau/(h^2+k^2) % diagonal of S, constant for all cells
qpmax = eps+1;
while (dpmax > eps) do
 dpmax = 0;
 do i = 1,n; j = 1,m
  dq = -((u(i,j)-u(i-1))/h + (v(i,j)-v(i,j-1))/k)/divp;q(i,j) = q(i,j) + dq;
  if (i < n) u(i,j) = u(i,j) + tau*dq/h;if (1 < i) u(i-1,j) = u(i-1,j) - tau*dq/h;
  if (j < m) v(i,j) = v(i,j) + tau*dq/k;if (1 < j) v(i,j-1) = v(i,j-1) - tau*dq/k;
  dpmax = max(dpmax,abs(dp));
 enddo
endwhile
...
```

The iteration method (12)- (14) is very flexible with respect to non-trivial region geometries because of its explicitness it is used as a basic tool in a lot of CFD codes. This method has the property that an inexact solution, be it by a limitation of the number of iteration steps per time level, or by a coarse error bound ε, does not lead to an error accumulation during the further time levels. This is getting investigated in a diploma thesis [4] at our university.

Numerical Examples

As an example for the application of the above discussed methods we consider a twodimensional stationary driven cavity problem ($f = 0$, $\nu = 1/Re$, H cavity hight, B width of the cavity, no slip boundary conditions on solid walls, constant tangential velocity at the top boundary of the cavity).

We compare the above discussed prediction correction method (12)-(14) with Newtons method for the solution of

$$C_h(u,u) + G_h q - V_h u = 0\ , \quad -D_h u = 0\ .$$

In Tables 1, 2 and 3 we compare the methods.

TABLE 1. Comparison of the inexact iterative method to Newtons method; iter. method/Newton < 1 means iterative method is more efficient then Newtons method

$H:B$	$m \times n$	Re	**iter. method/Newton**	$\varepsilon = 0.1$	0.01	0.001	0.0001
$1:1$	50×50	700		0.47	0.60	0.91	1.30
$1:1$	50×50	1000		0.49	0.74	1.32	1.82
$1:1$	100×100	1000		0.63	1.37	2.07	2.84
$1:2$	50×100	1000		0.72	1.50	2.29	2.87
$1:2$	100×200	1000		0.69	1.32	1.99	2.59

TABLE 2. Results of the iterative method

$H:B$	$m \times n$	Re	**iterations**	**time (sec.)**	$\varepsilon = 0.1$	0.01	0.001	0.0001
$1:1$	50×50	700	4096		0.95	1.38	2.10	2.97
$1:1$	50×50	1000	7675		1.15	1.92	3.44	5.42
$1:1$	100×100	1000	19774		17.32	37.73	60.54	83.21
$1:2$	50×100	1000	17663		4.65	11.07	16.83	23.41
$1:2$	100×200	1000	32987		99.00	196.01	294.93	396.20

CONCLUSION

Especially for problems which do not require very strong exactness, i.e. $\varepsilon = 0.1, 0.01$, the iterative methods are more time efficient then Newtons method. Thus, it is worth to use the iteration method (12)-(14) as a real alternative to Newtons method.

ACKNOWLEDGMENTS

Some of the numerical experiments were done by J. Rückelt who I would like to thank very much.

REFERENCES

1. T.D. Butler: *Recent Advances in CFD* in Comp. Methods in Appl. Sciences and Engineering, Lecture Notes in Comp. Science No. 11 (Part 2) 1974.
2. A.J. Chorin: *Numerical Solution of Navier-Stokes Equation*, Math. of Computation, Vol. 22, 1968.
3. F.H. Harlow and A.A. Amsden: *A Numerical Fluid Dynamics Calculation Method for All Flow Speeds*, Journal Comp. Physics 8, 1971.
4. J. Rückelt: *Optimisation and control of incompressible fluid flows*, Diploma thesis, Technische Universität Berlin, 2007.

TABLE 3. Results of Newtons method

$H:B$	$m \times n$	Re	**iterations**	**time (sec.)**	$\varepsilon = 0.1$	0.01	0.001	0.0001
$1:1$	50×50	700	7		1.99	2.30	2.30	2.30
$1:1$	50×50	1000	9		2.33	2.61	2.61	2.98
$1:1$	100×100	1000	14		27.64	27.64	29.30	29.30
$1:2$	50×100	1000	10		6.50	7.36	7.36	8.16
$1:2$	100×200	1000	29		144.24	148.54	148.54	153.12