Predictor-Corrector Preconditioned Newton-Krylov Method for Cavity Flow

Jianwei Ju and Giovanni Lapenta

Los Alamos National Laboratory, Los Alamos, NM 87545, USA
{jju, lapenta}@lanl.gov

Abstract. The Newton-Krylov method is used to solve the incompressible Navier-Stokes equations. In the present study, two numerical schemes are considered for the method: employing the predictor-corrector method as preconditioner, and solving the equations without the preconditioner. The standard driven cavity flow is selected as the test problem to demonstrate the efficiency and the reliability of the present preconditioned method. It is found that the Newton-Krylov method becomes more efficient if combined with the preconditioner.

1 Introduction

A classic problem of computational science and engineering is the search for an efficient numerical scheme for solving the incompressible Navier-Stokes equations. Explicit and semi-implicit methods can provide simple solution techniques but are seriously limited by the time step limitations for stability (explicit methods) and accuracy (implicit methods).

Recently, significant progress has been made in the development of a new fully implicit approach for solving nonlinear problems: the inexact Newton method [1, 2, 3]. The method is developed from the Newton iterative method, by applying a linear iterative method to the Jacobian equation for the Newton step and terminating that iteration when the convergence criterion holds [4]. The rapid increases in both speed and memory capacity of computing resources makes it practical to consider inexact Newton methods for incompressible flow problem.

For the solution of the linear Jacobian equation, Krylov methods are often the choice, leading to the Newton-Krylov (NK) approach. However, for most cases, Krylov solvers can be extremely inefficient solvers. The need for good preconditioners techniques becomes a constraining factor in the development of NK solvers.

In the field of incompressible and geophysical flows, recent work based on the multi-grid preconditioners [7, 9] have shown near ideal scaling, resulting in extremely competitive alternatives to classic explicit and semi-implicit methods.

In the present study, we present a new approach to preconditioning: the predictor-corrector (PC) preconditioner. The approach has two novelties. First, it preconditions directly the non-linear equations rather than the linear Jacobian equation for the Newton step. The idea is not new [4], but it is implemented here

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3514, pp. 82–89, 2005.

in a new way that leads to great simplification of the implementation. We note that this simplification is designed also to minimize the effort in refitting existing semi-implicit codes into full fledged implicit codes, representing perhaps a greater advance in software engineering than in computational science. Second, we test new ways of preconditioning the equations by using a combination of predictor-corrector semi-implicit preconditioning.

The fundamental idea is to use a predictor to advance a semi-implicit discretization of the governing equations and use a corrector Newton step to correct for the initial state of the predictor step. In substance, the Newton method iterates for a modification of the actual initial state to find a modified initial state that makes the semi-implicit predictor step give the solution of the fully implicit method.

Two advantages are obvious. First, the initial step is likely to be a better first guess for the modified initial step of the predictor than it is for the final state of the corrector step. Second, by modifying the non-linear function, the PC preconditioner gives the same type of speed-up of the Krylov convergence without requiring to formulate an actual preconditioning of the Krylov solver.

We use the standard driven cavity flow problem as the test problem to demonstrate the efficiency and the reliability of the present preconditioned method.

2 Governing Equations and Numerical Method

In the present study, we selected the standard driven 2-D incompressible cavity flow as the test example. The geometry and the velocity boundary conditions are shown in Fig. 1. The following non-dimensional variables are introduced:

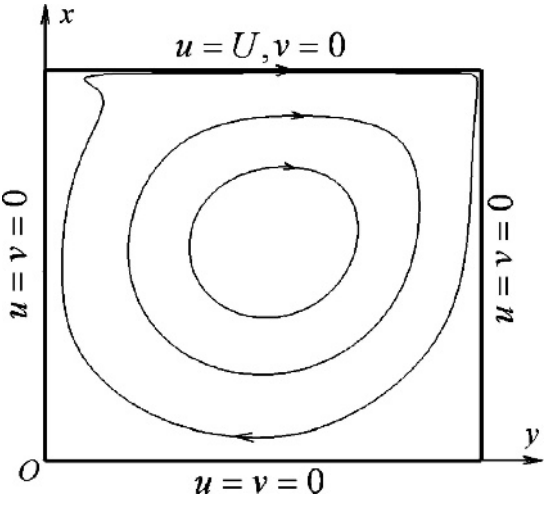

Fig. 1. Standard driven cavity flow and the velocity boundary conditions

$$(u, v) = \frac{(\hat{u}, \hat{v})}{U}\,,\ (x, y) = \frac{(\hat{x}, \hat{y})}{L} \tag{1}$$

where the hatted variables represent the dimensional variables. The scales are the cavity width L and the upper boundary velocity U. Time is normalized according to eq.(1). The governing equation, in term of the vorticity ω, and the stream function ψ, can be expressed as:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \tag{2}$$

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) \tag{3}$$

where $u = \partial\psi/\partial y$, $v = -\partial\psi/\partial x$, $\omega = \partial v/\partial x - \partial u/\partial y$, and $\mathrm{Re} = UL/\nu$ is the Reynolds number based on the viscosity ν. Dirichlet boundary conditions are applied for the stream function and the boundary conditions for vorticity is determined by the physical boundary conditions on the velocity [11]. Thus, at the left wall $\omega = \partial v/\partial x = -\partial^2\psi/\partial y^2$. We can obtain expressions for ω at other walls in an analogous manner.

Eqs. (2) and (3) are discretized using the centered difference scheme in space. Two types of discretization are considered in time, the semi-implicit Euler scheme (where the velocity in eq.(3) is taken at the initial time level of the time step) and the fully implicit Euler scheme.

The resulting set of coupled non-linear difference equations are solved using the Newton-Krylov method. As Krylov solver we use GMRES since the Jacobian system is non-symmetric. The resulting method is completely matrix-free as only matrix-vector products, rather than details of the matrix itself are needed. This circumstance greatly simplifies the application of the method to complex problems. We design the preconditioner to maintain this property.

3 Preconditioners

In the present study, a preconditioner is constructed by using the predictor-corrector method. The key idea lies on modifying the target point of the Newton iteration.

Most discretization schemes can be expressed as a set of difference equations for a set of unknowns $\mathbf{x}$ representing the unknown fields on a spatial grid. Once the time is discretized, the state vector $\mathbf{x}$ is computed at a sequence of discrete time levels. We label the initial state of a time step as $\mathbf{x}^0$ and the final time as $\mathbf{x}^1$.

When the time discretization scheme is fully implicit, the most general two-level scheme can be formulated as:

$$\mathbf{x}^1 = \mathbf{x}^0 + f(\mathbf{x}^0, \mathbf{x}^1) \tag{4}$$

where the vector function f depends both on the initial and the final states. The implicit nature of the scheme resides in the fact that the function f is a function

of the new time level, requiring the solution of a set of non-linear (if the function f is non-linear) coupled equations. As noted above this can be accomplished with the NK method [4]. The method is based on solving the Jacobian equation obtained linearizing the difference eq. (4) around the current available estimate $\mathbf{x}_k^1$ of the solution in the Newton iteration:

$$\delta\mathbf{x} + \mathbf{x}_k^1 = \mathbf{x}^0 + f(\mathbf{x}^0, \mathbf{x}_k^1) + J\delta\mathbf{x} \tag{5}$$

where $J = \partial f/\partial\mathbf{x}$ is the Jacobian matrix and $\delta\mathbf{x}$ is the correction leading to the new estimation by the Newton iteration: $\mathbf{x}_{k+1}^1 = \mathbf{x}_k^1 + \delta\mathbf{x}$.

The solution of eq. (5) is conducted with a Krylov solver, we use here GMRES since the Jacobian matrix is non symmetric. While the pure inexact NK method works in giving a solution, the number of Krylov iterations required for each Newton step to solve eq. (5) can be staggering. In particular, as the grid is refined and the size of the unknown vector $\mathbf{x}^1$ is increased the number of Krylov iterations tends to increase. This is the reason why a preconditioner is needed.

Here we propose to use the predictor-corrector method as a preconditioner. The approach requires to design alongside the fully implicit scheme in eq. (4), a second semi-implicit method. We note that this is typically no hardship as semi-implicit methods were developed and widely used before the implicit methods became tractable. Using the same notation, we can write the most general two-level semi-implicit algorithm as:

$$\mathbf{x}^1 = \mathbf{x}^0 + A\mathbf{x}^1 + f_{SI}(\mathbf{x}^0) \tag{6}$$

where A is a linear operator (matrix) and the function f_{SI} depends only on the initial state $\mathbf{x}^0$. The semi-implicit nature of the scheme resides on the fact that the difference eq. (6) depends non-linearly on the (known) initial state $\mathbf{x}^0$ but only linearly on the new (unknown) state $\mathbf{x}^1$.

In the classic implementation of preconditioners [8], the equation for the semi-implicit scheme (6) is rewritten in terms of the modification $\delta\mathbf{x}$ in a given Newton iteration, essentially leading to the matrix A of the semi-implicit scheme to be used as a preconditioner for the Jacobian matrix J of eq. (5). The approach has been extremely successful in terms of providing a robust and effective solution scheme. For example in the case of incompressible flows, the number of Krylov iteration has been shown [9, 7] to be reduced drastically and to become nearly independent of the grid size.

However, a substantial modification of existing codes follows from the need to modify the GMRES solver to use the matrix A as a preconditioner, especially when the method is formulated in a matrix-free form where the matrix J and the matrix A are not explicitly computed and stored.

We propose a different approach. We consider the following predictor-corrector algorithm:

$$\begin{cases} \text{(P)}\ \mathbf{x}^1 = \mathbf{x}^\star + A\mathbf{x}^1 + f_{SI}(\mathbf{x}^0) \\ \text{(C)}\ \mathbf{r} = \mathbf{x}^1 - \mathbf{x}^0 - f(\mathbf{x}^0, \mathbf{x}^1) \end{cases} \tag{7}$$

The predictor step uses the semi-implicit scheme to predict the new state $\mathbf{x}^1$ starting from a modification of the initial state $\mathbf{x}^*$. The corrector step computes

the residual $\mathbf{r}$ for the fully implicit scheme when $\mathbf{x}^1$ from the predictor step is used.

We propose to use scheme (7) by using $\mathbf{x}^0$ as the initial guess of $\mathbf{x}^*$ and using the NK method to find the solution for $\mathbf{x}^*$ that makes the residual $\mathbf{r}$ of the corrector equation vanish. Once $\mathbf{r} = 0$ (within a set tolerance), the fully implicit scheme is solved, but it is solved not iterating directly for $\mathbf{x}^1$ but iterating for the $\mathbf{x}^*$ that makes the predictor step predict the correct solution $\mathbf{x}^1$ of the corrector step.

Two points are worth noting.

First, we have modified the task of the NK iteration changing our unknown variable from $\mathbf{x}^1$ to $\mathbf{x}^*$. This corresponds to change the non-linear residual function that the Newton method needs to solve. To first order in the Taylor series expansion leading to the Jacobian equation this practice is identical to applying the traditional preconditioners directly to the Jacobian equation [4]. However, to higher order this might be a better approach as it reduces the distance between the initial guess ($\mathbf{x}^0$) and the solution for $\mathbf{x}^*$. If the semi-implicit method works properly, $\mathbf{x}^0$ is closer to the converged $\mathbf{x}^*$ than to the final state $\mathbf{x}^1$.

Second, programming the PC preconditioner is easier. The NK solver can be used as a black box, without any need to formally go into it and modify the Jacobian eq. (5) by adding a preconditioner. The semi-implicit method can be used directly on the actual states and not on their variation $\delta\mathbf{x}$ between two subsequent Newton iterates. This latter operation is complex as boundary conditions and source terms in equations need to be treated differently.

The approach described above is ideally suited for refitting an existing semi-implicit code by simply taking an off the shelf NK solver and wrapping it around the semi-implicit method already implemented. The only change being that in the semi-implicit scheme the initial state $\mathbf{x}^0$ is replaced by the guess of $\mathbf{x}^*$ provided by the NK solver. We have indeed proceeded in this fashion by wrapping the standard NK solver provided in the classic textbook by Kelley [5] around our previously written semi-implicit solver for the incompressible flow equations.

4 Results and Discussion

In the example below we present results for a case with a mesh of 129×129 cells. The classic cavity flow solution is computed starting from a stagnant flow and allowing the boundary conditions to drive the cavity to a steady state. The time evolution of the vorticity at the center of the cavity is shown in Fig. 2.

The flow condition at steady state is shown in Fig. 3. The figure is generated using the same contour lines used in the reference benchmark solution presented by Chia et al. [10]. We compared visually our solution with the published reference benchmark obtaining complete agreement.

We have compared the efficiency of the NK solver with and without the PC preconditioner described above. The number of GMRES iterations per Newton iteration for the two simulations are shown in Fig. 4 as a function of time. At the beginning of the simulation when the flow is in a transient state and is

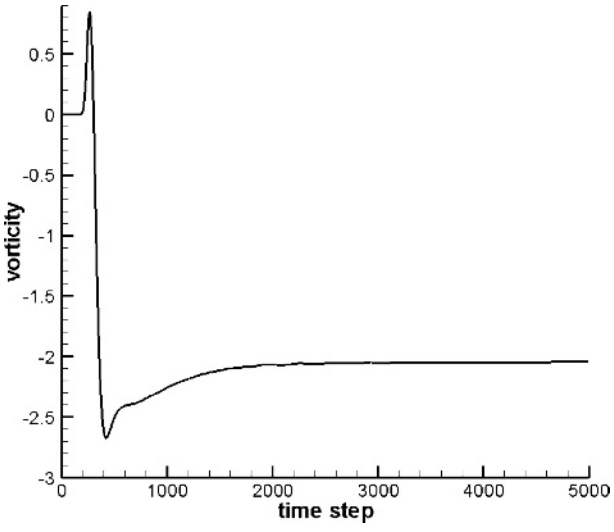

Fig. 2. Evolution of the vorticity at the cavity center as a function of time step

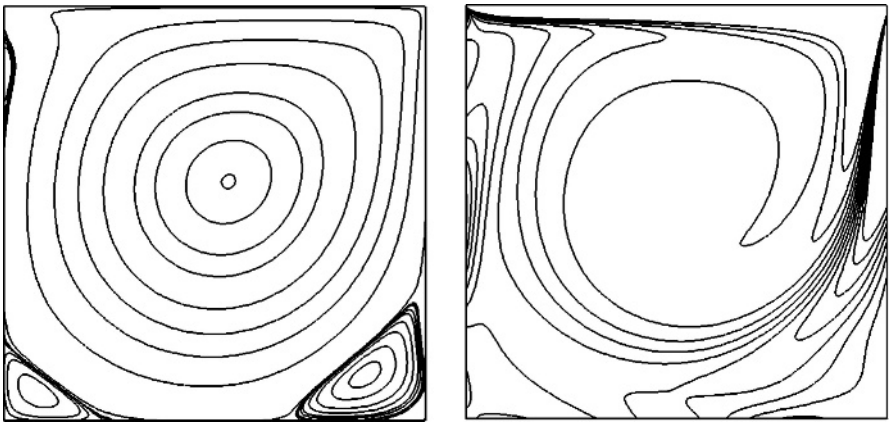

Fig. 3. (a) Contours of stream function, (b) Contours of vorticity. Flow structure at steady state for Re=1000

adjusting to the driving upper boundary condition, the evolution is dynamical. In that condition, the solver needs more Krylov iterations. The average number of iterations before 2000 time steps is 5.3.

At later stages of the simulation, the system reaches a steady state and flow remains unchanged from time step to time step. In this very favorable condition for the NK method, the average number of Krylov iterations after 2000 time steps is 1.3 for the preconditioned simulation and 2.5 for un-preconditioned simulation.

We should point out that the test provides confirmation that the preconditioning approach presented here is promising. But it cannot test how well it

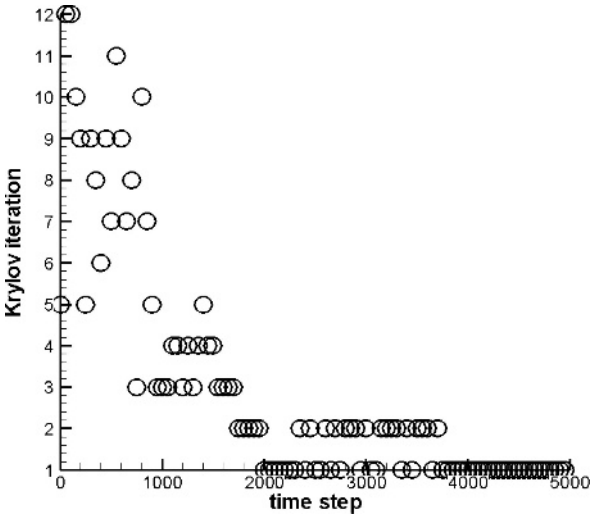

Fig. 4. Number of GMRES iterations per Newton iteration

will do in more realistic situations. As Fig. 4 shows, the Krylov solver is already working remarkably well and the space for improvement was rather limited. Nevertheless, the preconditioner reduced the number of iterations by a factor of 2.

5 Conclusions

We have presented a new numerical approach for the solution of incompressible flows based on the Newton-Krylov method. A new approach to preconditioning based on using a predictor semi-implicit time differencing step followed by a fully implicit corrector step has been presented. The results for the standard driven flow indicate that the preconditioner can reduce the number of Krylov iterations by a factor of 2. Future work will focus on more complex flows to investigate the generality of the approach presented here.

References

1. Dembo, R.S , Eisenstat, S.C , Steihaug, T.: Inexact Newton Method. SIAM J. Numer. Anal.,19 (1982), 400-408.
2. Eisenstat, S.C , Walker, H.F : Globally Convergent Inexact Newton Methods. SIAM J. Optim., 4 (1994), 393-422.
3. Eisenstat, S.C , Walker, H.F : Choosing The Forcing Terms In An Inexact Newton Method. SIAM J. Sci. Comput., 17 (1996), 16-32.
4. Kelley, C.T : Solving Nonlinear Equations with Newton's Method. Society for Industrial and Applied Mathematics (2003).
5. Kelley, C.T : Iterative Methods for Linear and Nonlinear Equations. Society for Industrial and Applied Mathematics (1995).

6. Kim, J.H , Kim, Y.H : A Predictor-Corrector Method For Structural Nonlinear Analysis. Comput. Methods Appl. Mech. Engrg. 191 (2001) 959-974.
7. Knoll, DA, Mousseau, VA: On Newton-Krylov multigrid methods for the incompressible Navier-Stokes equations., J. Comput. Phys. 163 (2000), 262-267.
8. Knoll, DA; Keyes, DE: Jacobian-free Newton-Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004), 357-397.
9. Pernice, M, Tocci, M.D : A Multigrid-Preconditioned Newton-Krylov Method for the Incompressible Navier-Stokes Equations. SIAM J. Sci. Comput., 23 (2001). 398-418.
10. Chia, U., Chia, K. N., Shin, T., High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. J. Comput. Phys. 48 (1982), 387-411.
11. Roache, P.J.: Fundamentals of Computational Fluid Dynamics. Hermosa Publishers (1998).