COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
Commun. Numer. Meth. Engng 2005; **21**:201–207
Published online 24 December 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm.743

Projective prediction of pressure increments

Rainald Löhner*,†

School of Computational Sciences, MS 4C7, George Mason University, Fairfax, VA 22030-4444, U.S.A.

SUMMARY

A simple projective predictor of pressure increments has been developed. The procedure requires the storage of previous pressure increments and right-hand sides, i.e. a modest amount of storage. Based on this information, the known right-hand sides are projected onto the right-hand side at the new timestep. The projection coefficients are then used to predict the pressure increment at the new timestep. Numerical tests indicate that the number of iterations required is reduced considerably. Furthermore, the main gains are achieved with a very modest number of basis vectors. Typically, no more than 2 previous results have to be stored. The procedure is easy to implement and should be applicable to a large number of codes. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: pressure-Poisson equation; incompressible flow solvers; finite elements; CFD

1. INTRODUCTION

Many incompressible solvers are based on the so-called projection techniques, whereby the advancement of the flowfield in time is split into the following three substeps [1–6]:

- Advance the velocity with an advective–diffusive predictor ($\mathbf{v}^n \rightarrow \mathbf{v}^*$);
- Solve a Poisson equation for the pressure increment ($p^n \rightarrow p^{n+1}$);
- Update the velocity field with the pressure increment to obtain a diverge-free solution at the new timestep ($\mathbf{v}^* \rightarrow \mathbf{v}^{n+1}$).

The solution of the Poisson equation, which is of the form

$$\nabla^2 (p^{n+1} - p^n) = \frac{\rho \nabla \cdot \mathbf{v}^*}{\Delta t} \tag{1}$$

is typically carried out with a preconditioned conjugate gradient [6] solver, and consumes a considerable percentage of the overall computational effort. Any gain (e.g. in the form of a

*Correspondence to: Rainald Löhner, School of Computational Sciences, MS 4C7, George Mason University, Fairfax, VA 22030-4444, U.S.A.
†E-mail: rlohner@gmu.edu

Received 8 June 2004
Accepted 9 November 2004

reduction in the number of iterations) will have an immediate impact on the overall execution speed of the solver.

Equation (1) results in a discrete system of the form

$$\mathbf{K} \cdot \Delta\mathbf{p} = \mathbf{r} \tag{2}$$

One way to reduce the number of iterations required is to start with a value of $\Delta\mathbf{p}$ that is close to the solution. For time-accurate problems with constant timesteps an extrapolation from previous increments seems a reasonable proposition. However, experience [3] indicates that this does not yield a reliable way of reducing the number of iterations. Most solvers tend to initialize the iterative solver for Equation (2) with $\Delta\mathbf{p} = 0$. The rationale given for this choice is that at steady state $\Delta\mathbf{p} = 0$, i.e. as the solution is converging, this represents a good choice. On the other hand, it can be argued that the pressure increment between timesteps is similar [3]. If we consider the case of a vortex street behind a cylinder or a car, this is certainly the case, as many timesteps are required per shedding cycle. For this reason, we seek an estimate of the starting value $\Delta\mathbf{p}$ based on the values obtained at previous timesteps.

2. PROJECTIVE PREDICTION

In what follows, the *basic assumption* is that $\mathbf{K}$ does not change in time. For many incompressible flow solvers this is indeed the case. Solvers that use some form of stabilization or consistent numerical fluxes (e.g. in the form of a fourth-order damping) do not fall under this category. For these cases it may be argued that $\mathbf{K}$ changes very little.

If we denote by $\Delta\mathbf{p}^i, \mathbf{r}^i, i = 1, l$ the values of the pressure increments and right-hand sides at previous timesteps $n - i$, we know that

$$\mathbf{K} \cdot \Delta\mathbf{p}^i = \mathbf{r}^i \tag{3}$$

Given the new right-hand side $\mathbf{r}$, we can perform a least-squares approximation to it in the basis $\mathbf{r}^i, i = 1, l$:

$$(\mathbf{r} - \alpha_i \mathbf{r}^i)^2 \rightarrow \min \tag{4}$$

which results in

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{s}, \quad A^{ij} = \mathbf{r}^i \cdot \mathbf{r}^j, \quad s^i = \mathbf{r}^i \cdot \mathbf{r} \tag{5}$$

Having solved for the approximation coefficients α_i, we can estimate the start value $\Delta\mathbf{p}$ from

$$\Delta\mathbf{p} = \alpha_i \Delta\mathbf{p}^i \tag{6}$$

We remark that, in principle, the use of the right-hand sides $\mathbf{r}^i, i = 1, l$ as a basis may be numerically dangerous. After all, if any of these vectors are parallel, the matrix $\mathbf{A}$ is singular. One could perform a Gram–Schmidt orthogonalization instead. This option was invoked by Fischer [3] who looked at a number of possible schemes to accelerate the convergence of iterative solvers using successive right-hand sides within the context of incompressible flow solvers based on spectral elements. However, we have not found this to be a problem for any of the cases tried to date. The advantage of using simply the original right-hand sides is that the update of the basis is straightforward. We keep an index for the last entry in the basis, and simply insert the new entries at the end of the timestep in the position of the oldest basis vector. The storage requirements for this projective predictor scheme are rather modest: `2*npoin*nvecl`. We typically use 1–4 basis vectors, i.e. the storage is at most `8*npoin`.

3. EXAMPLES

1. *Cylinder*: This classic case considers a cylinder of unit diameter in a uniform flow. The Reynolds-number based on the diameter is approximately $Re = 200$. The geometry is shown in Figure 1(a), and the velocity field in the plane $z = 0$ in Figure 1(b). The mesh had 13 kpts and 54 kpts elements. The case was run for 100 timesteps, with a

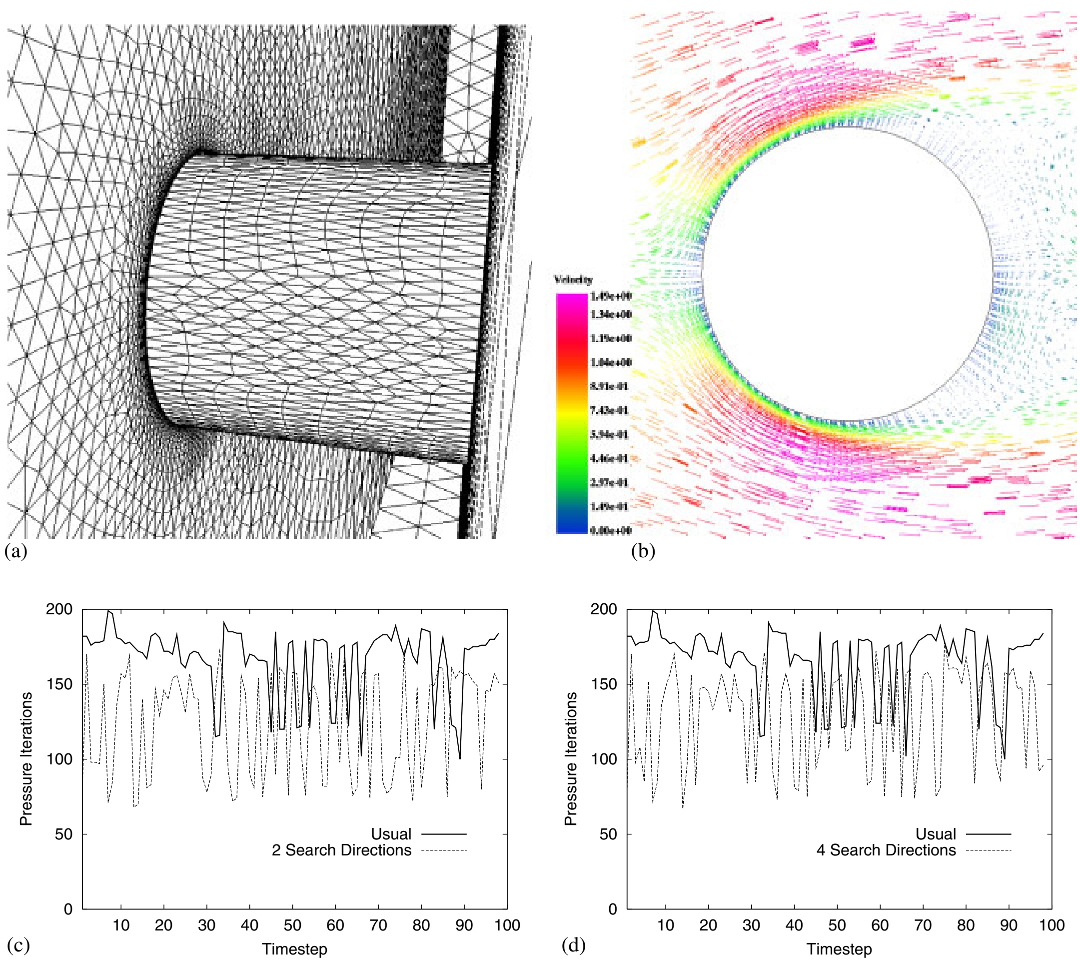

Figure 1. (a,b) Cylinder: mesh and flowfield; and (c,d) Cylinder: iterations for the pressure-Poisson system.

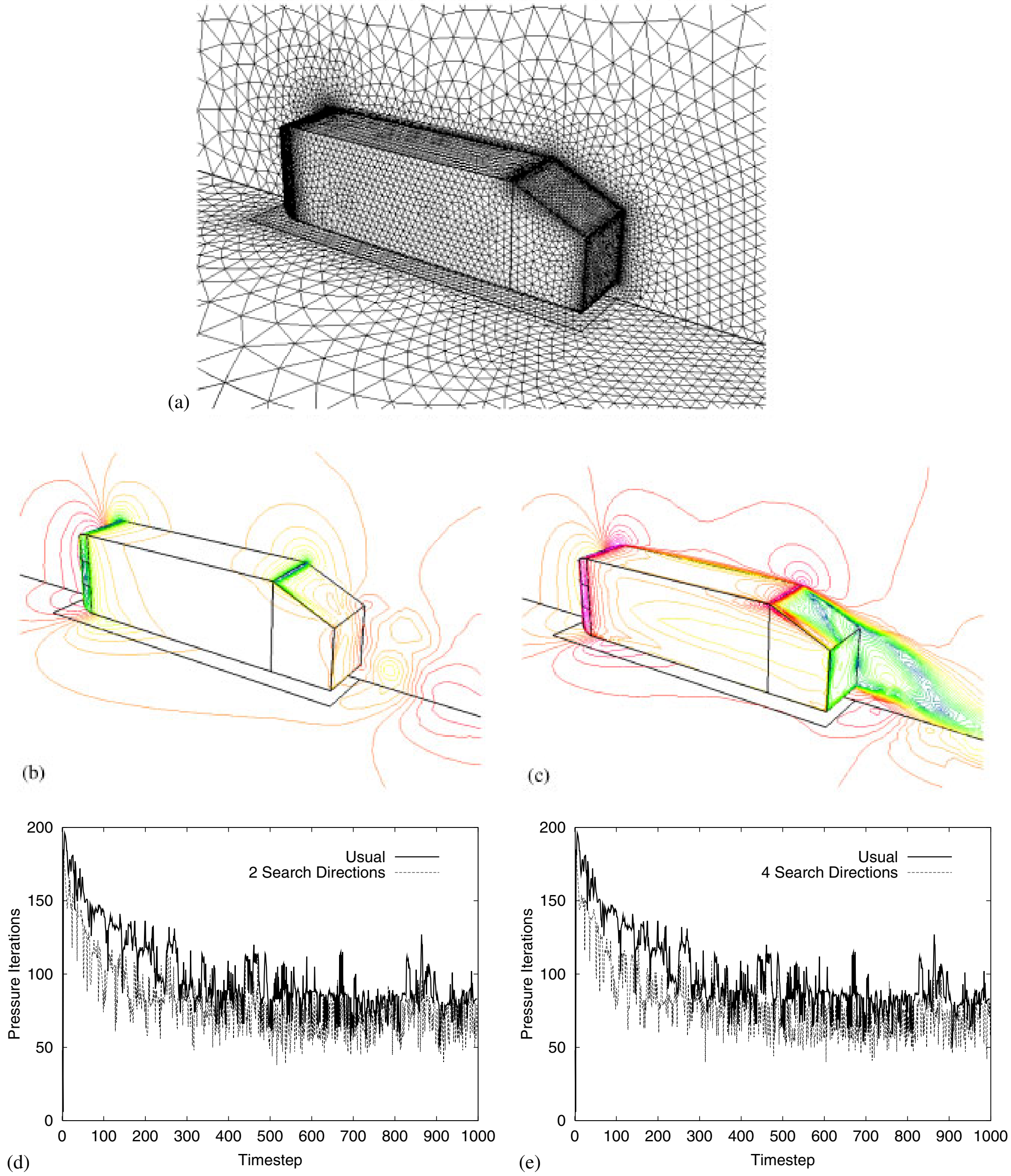

Figure 2. (a) Ahmed body: surface mesh; and (b,c) Ahmed body: surface pressure and speed. (d,e) Ahmed body: iterations for the pressure-Poisson system.

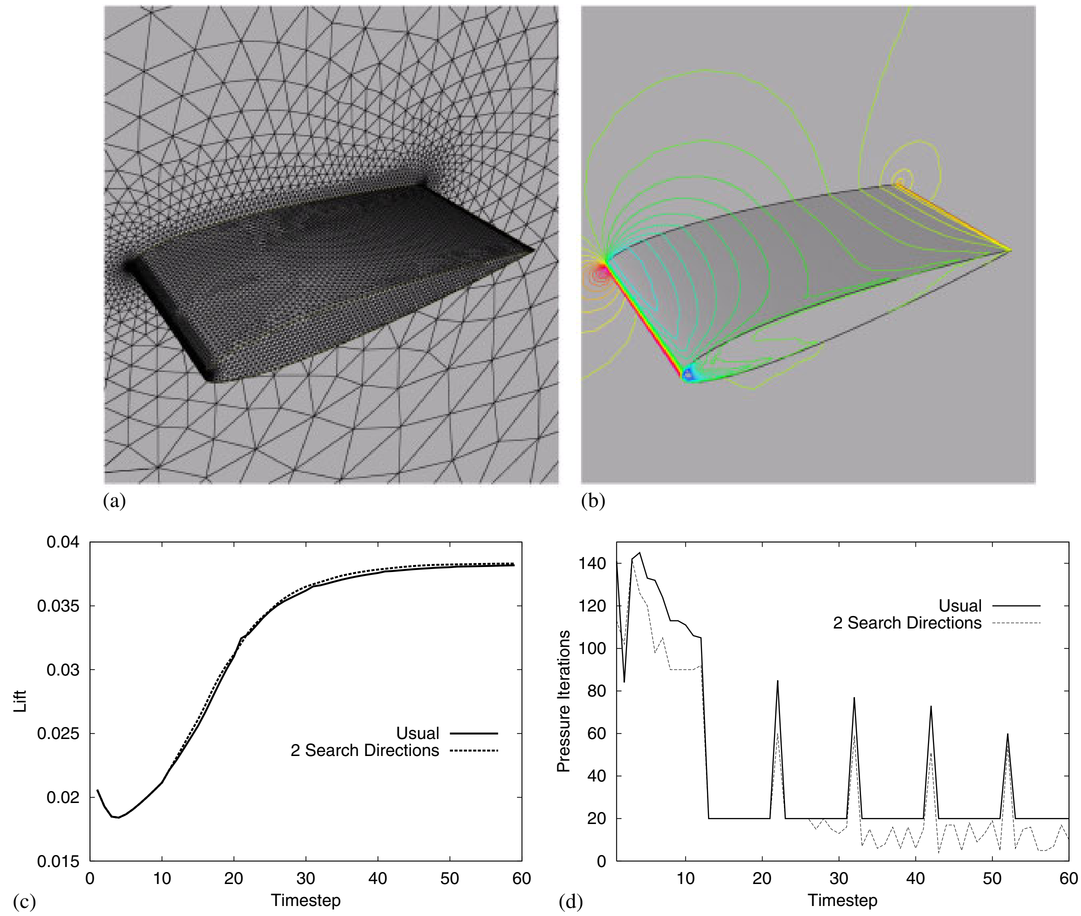

Figure 3. (a,b): NACA 0012: surface mesh and pressure.
(c,d): NACA 0012: lift convergence and pressure iterations.

5-stage explicit advective–diffusive predictor [7] and a Courant-number of $C = 1.2$. Due to the presence of highly stretched elements, a linelet preconditioner [6] was used for the pressure-Poisson equation. The required number of iterations per timestep may be seen in Figures 1(c) and (d). The average number of pressure iterations required was 165 for the original scheme and 128, 122 and 128 for the projective prediction with 1,2 and 4 search directions, i.e. a reduction of 25%.

2. *Ahmed Car Body*: This example considers high Reynolds-number flow past the so-called Ahmed body. It is a standard test case for external car aerodynamics. The parameters were set to: $\rho = 1.0, \mathbf{v} = (1,0,0), \mu = 2.33 \times 10^{-7}, L = 1$, which implies a Reynolds-number of $Re = 4.29 \times 10^{6}$. The Smagorinsky turbulence model was used. The resulting flow is quasi-steady and shows the development of a vortex train behind the body. Figures 2(a)–(c) show the surface mesh employed, as well as the pressure and velocity

field obtained. Note the boundary layer mesh. The complete mesh had 77 kpts and 420 kels. The case was run for approximately 1000 timesteps, with a 4-stage explicit advective–diffusive predictor and a Courant-number of $C = 0.8$. As before, a linelet preconditioner was used for the pressure-Poisson equation. The required number of iterations per timestep may be seen in Figures 2(d) and (e). As before, a reduction of approximately 25% is achieved.

3. *NACA0012*: This classic example considers a NACA0012 wing at $\alpha = 5°$ angle of attack. The aim of this test is to guage the performance of the projective pressure increment predictor for a steady, inviscid (Euler) case. Figures 3(a) and (b) show the surface mesh employed, as well as the surface pressures obtained. The mesh consisted of 68 kpts and 368 kels.
   For the advective–diffusive predictor, an implicit LU-SGS/GS scheme was used [8]. Local timesteps were employed with a Courant number based on the advective terms of $C = 5$. Given that the mesh is isotropic, a diagonal preconditioner for the pressure-Poisson equation was used. As the flow is started impulsively, a divergence cleanup (first three iterations) precedes the advancement of the flowfield. After the residuals have converged an order of magnitude, the number of pressure iterations is reduced artificially to 20, and the relative tolerance for residual convergence is increased to $\text{tol}_p = 10^{-2}$. Every 10th timestep, the residual convergence is lowered to the usual value of $\text{tol}_p = 10^{-4}$ in order to obtain a divergence-free flowfield. We have found this procedure to work well for inviscid, steady flows, reducing CPU requirements considerably as compared to keeping the relative tolerance constant throughout the run. The case was run for 60 steps, which was sufficient to lower the relative change in lift forces to below $\text{tol}_f = 10^{-3}$. The convergence of the lift may be seen in Figure 3(c). The required number of iterations per timestep is shown in Figure 3(d). Note that even for this steady, inviscid case the projective pressure predictor yields a considerable reduction in the number of pressure iterations.

4. CONCLUSIONS

A simple projective predictor of pressure increments has been developed. The examples shown here, as well as others run by the author, indicate that the number of iterations required is reduced considerably by using such a projective predictor of pressure increments. Furthermore, the main gains are achieved with a very modest number of basis vectors. Typically, no more than two previous results have to be stored. The procedure is easy to implement and should be applicable to a large number of codes.

REFERENCES

1. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
2. Gunzburger MD, Nicolaides R (eds). *Incompressible Computational Fluid Dynamics: Trends and Advances*. Cambridge University Press: Cambridge, 1993.
3. Fischer PF. Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides. *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:193–204.
4. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112–140.

5. Löhner R. *Applied CFD Techniques*. Wiley: New York, 2001.
6. Soto O, Löhner R, Camelli F. A Linelet preconditioner for incompressible flows. *International Journal of Numerical Methods for Heat and Fluid Flow* 2003; **13**(1):133–147.
7. Löhner R. Multistage explicit advective prediction for projection-type incompressible flow solvers. *Journal of Computational Physics* 2004; **195**:143–152.
8. Löhner R, Chi Yang, Cebral JR, Soto O, Camelli F, Waltz J. Improving the speed and accuracy of projection-type incompressible flow solvers. *AIAA-03-3991-CP*, 2003.

