Trends in Mathematics
Information Center for Mathematical Sciences
Volume 2, Number 1, September 1999, Pages 1–

A STUDY OF ITERATIVE METHODS WITH ALTERNATING DIRECTION PREPROCESSING ON THE SIMULATION OF BACKWARD-FACING STEP FLOWS

KANG C. JEA, DANIEL LEE, AND MULDER YU

Abstract. We study the effectiveness of applying various iterative methods in the numerical simulation of backward-facing step flows. Alternating Direction Implicit method was investigated in Chao and Chou [1] for solving the associated linear system. However, it is felt that the ADI method needs large amount of computer time. In this paper we proposed the idea of using alternating direction approach for certain iterations and then switched to preconditioned iterative methods. It is called as alternating direction preprocessing here.
Comparisons were made among various accelerated methods and preconditioners. Observations on some characteristics of these methods and preconditioners were obtained. These are found to be more efficient, with better potential for solving similar application problems. Moreover, a parallel processing of the ADI method is also implemented using PVM on IBM SP2 environment.

1. Introduction

The dynamics of the step-region vortex structure in backward-facing step flows was investigated through direct numerical simulation using a high-order accuracy numerical procedure with strict treatment of the pressure solver. Alternating Direction Implicit method was suggested in Patankar[4,5] and further investigated in Chao and Chou[1], in which numerical results were found to be consistent with the discovery of Yang et al.[7]. However, it is felt that the ADI approach, in its simplest form, although can deal numerically with large pde system arising from applications, needs large amount of computer time. This is due to the relatively slow convergence.

In this paper we investigated several different approaches to solve the above mentioned problem. These are found to be more efficient, with better potential for solving similar application problems.

Alternating direction approach was adopted to solve the system component one equation each time, keeping other system variables as known values, in an iterative procedure, for each time step. Each component equation is further simplified to a (sub)system of linear algebraic equations by applying a Jacobi type iterative formulation. The resulting approximated linear system of each component equation was solved in Chao's work by an ADI method, which first solved for grid variables on x mesh lines, one by one, and then solved for grid variables on y mesh lines, again, one by one.

Key words and phrases. Alternating Direction Preprocessing, Iterative Methods, Preconditioners.

In comparison with experimental data, satisfactory prediction of the time-averaged flow characteristics is reached by this numerical procedure.

We will demonstrate in this paper that the inner loop of ADI procedure, described in next section, is very stable, showing very steady monotone convergence. We will also investigate the effects in applying several iterative methods in solving the resulting linear systems with or without preconditioners.

A parallel version of the ADI method was implemented using PVM on IBM SP2 environment, based on task parallel principle. The observed speedup indicates potential usage in similar applications.

2. Problem Formulation and Solution Procedures

Details concerning the governing equations, the discretization, the pressure treatment, the grid systems, and the initial and boundary conditions are omitted here. Interested readers can consult the work of Chao and Chou. What we concerned here is the numerical procedures in problem solving.

All methods are based on ADI procedure with 129 x 81 grid points, with number of sweeping in u,v,p,f directions equal to 4, 4, 6 and 4, respectively.

We note that the governing equation consists of a system of four tightly coupled partial differential equations. Alternating direction approach was adopted to solve the system component one equation each time, keeping other system variables as known values, in an iterative procedure, for each time step. Each component equation is further simplified to a (sub)system of linear algebraic equations by applying a Jacobi type iterative formulation.

This resulting approximated linear system of each component equation was solved in Chao's work by an ADI method, which solved for grid variables on x mesh lines, one by one, and then solved for grid variables on y mesh lines, again, one by one. Several cycles of this sequence of operations were carried out.

3. The Alternating Direction Approach

The application problem consists of a system of four partial differential equations. These are associated with the x-momentum, the y-momentum, the pressure correction, and the mixing. We will simply refer to them as the u-equation, the v-equation, the p-equation, and the f-equation, respectively. These equations are nonlinear and coupled.

In a standard alternating direction implicit approach, each component equation is solved in a sequential manner, with only one physics variable as target and the other physics variables treated as known. A five points stencil finite difference method is used to yield to, in the current case, a five diagonal matrix.

Several approaches as listed in the followings are quite natural at this point.

Approach 1 : Simple ADI scheme as is in the work of Chao and Chou[1].

Approach 2 : Simple Iterative method with (poor) initial iterate. We demonstrated the applicability of iterative methods to the above mentioned problem.

Approach 3 : Iterative method with alternating direction preprocessing for obtaining better initial iterate.

Approach 4 : Simple task parallel implementation of these methods.

Approach 5 : A pipelined parallel processing to the ADI method in the numerical modelling procedure. Domain decomposition and Task parallel is built in the design.

The approaches 1 to 4 were implemented and will be discussed in next session, while approach 5 will be briefly commented in the end of this paper and reported elsewhere in details.

We describe below the methods and abbreviations used in the sequel. The NSPCG package developed by Oppe et al.[3] was adopted.

ADI : m line sweeps in x direction, followed by m line sweeps in y directions, with m kept a constant. For equation of u, v, p, and f, these constants are 4, 4, 6, and 4, respectively.

AD100 : 100 alternating direction operations.

AD200 : 200 alternating direction operations.

J : Jacobi iteration.

AD100+J : AD100 operation followed by Jacobi iteration.

AD200+J : AD200 operation followed by Jacobi iteration.

J+BCGS : Jacobi iteration accelerated by Biconjugated Gradient Squared iteration.

J+GMR : Jacobi iteration accelerated by GMRES.

J+Cheb : Jacobi iteration accelerated by Chebyshev iteration.

IC : Incomplete Cholesky factorization.

IC+ORES : IC operation accelerated by ORTHORES .

IC+BCGS : IC operation accelerated by BCGS .

IC+Cheb : IC operation accelerated by Chebyshev iteration.

IC+GMR : IC operation accelerated by GMRES.

4. The Numerical Experiments and Results

Iterative methods, with or without preconditioners, with or without accelerations, were applied to solve the linear system of equations in space variables iterations. The results were shown in Table 1 - 4.

We remark that nonsymmetric diagonal format was used as matrix storage scheme in calling NSPCG. Furthermore, only numerical results for one time step were reported here. Although extensive experiments were made for long time.

5. Discussions and Conclusions

Here we give observations, Table 1, 2, 3, and analysis, Table 4, 5, 6, based on detailed output of our experiments. Similar results were obtained using the SLAP[6] package. The corresponding tables are not presented here for limitation of space.

1. ADI method is stable and shows slow monotone convergence.

2. Iterative methods with AD100 or AD200 as initial iterates, are more efficient in computing time than the straight ADI method or pure iterative methods. As an example, the time spent for the cases, the straight ADI, AD100+J+ORES, are at the ratio $1 : 0.5181 (\approx 1.93)$, and the ratio for straight ADI, AD200+J +ORES is $1 : 0.6023 (\approx 1.66)$. Certainly this shows that Jacobi-ORTHORES with AD100 or AD200 preprocessing is about twice faster than straight ADI. Detailed analysis is shown on Tables 4, 5 and 6. Moreover, with Jacobi or IC as preconditioner, ORTHORES is the most efficient iterative method in our study.

3. Without counting the time spent for AD100 or AD200 preprocessing, we can fin out the AD100+J+ORES case shows a $1 : 0.1913 (\approx 5.23)$ speedup as shown in Table 4, and the AD200+J+ORES case even shows a $1 : 0.0235 (\approx 42.55)$ speedup, see Table 5. As initial guess, one will think AD200 is certainly more efficient than

AD100. In fact, the numerical results in Table 6 showed that it is not always true. Some iterative methods can do much faster if taking AD100 rather than AD200 as initial guess. We note that with the arbitrarily chosen initial guess $(1, 1, \cdots, 1)$, some iterative methods oscillate more or less.

4. Since the ADI method is programmed directly, while the other iterative methods were called via the NSPCG package which required a lot overhead. We expect these iterative methods will perform even much better when embeded in the main code.

5. Finally, we note that results of each iteration was written to files, and the cpu time spent in this regards was included in calculating total cpu time. The calculated time-spent will be much less if the results were not printed at each iteration. Since the output needs a constant time, this implies that the advantage of iterative methods over simple ADI method will be even more.

It is therefore concluded that precondtioned iterative methods when preprocessed with AD100 or AD200 can perform much better than the straight ADI method. The reduction in computing time is about 0.5 to 0.6, see Table 4 and Table 5.

6. The Implementation of Parallel Processing

A parallel design of the basic ADI method on IBM SP2 was done. The implementation was based on task parallel principle.

We recall the application problem consists of a system of four partial differential equations, which are non-linear and coupled. In a standard ADI approach, each component equation is solved in sequential manner in such a way that only one physics variable is treated as target and the other physics variables are considered as known. Each component equations is further simplified to a (sub)system of linear algebraic equations by a Jacobi type iterative formulation. In fact, after analyzing the approximated linear systems of u, v, p and f, we found the linear systems of u, v and f are independent and therefore can be solved in parallel. However, the values of u and v will later be modified according to the value of p. In other words, the computation of u, v and p can not be implemented in parallel. Therefore, we adopt the strategy of tasks parallel to distribute the computation of u, v and f to three processing elements. Then, we use one processing element to calculate p. We implement this parallel approach on the environment of IBM SP2 cluster. Numerical results for one time step are reported in Table 7. As expected, the sequential version takes almost two times of the cpu time of the parallel version.

7. Acknowledgement

We are grateful for the support of grant NCHC-85-08-008 and the opportunity to run on IBM/SP2 series parallel computers at the National Center for High-Performance Computing, Hsinchu, Taiwan. Very sincere gratitude is expressed here.

References

[1] Yei-Chin Chao and Wen-Fuh Chou, *Time-Dependent Numerical Simulation of Turbulent Backward-Facing Step Flows with and without Mass Bleed, Proceedings of the 19th national conference on theoretical and applied mechanics held in Taiwan, ROC, pp. 223-231,1995.*

[2] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM3 user's guide and reference manual, Tech. Rept. ORNL-TM12187, Oak Ridge National Laboratory (1993).*

[3] Thomas C. Oppe, Wayne D. Joubert and David Kincaid *NSPCG User's Guide, Technical Report CNA-216, 1988, The University of Texas at Austin.*

[4] S. V. Patankar and D. B. Spalding, *A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensitional Parabolic Flows, Int.J.Heat Mass Transfer, Vol.15, 1972.*

[5] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow, Hemisphere, New York,1980.*

[6] M. Seager, *"A SLAP for the Masses", Lawrence Livermore Nat. Laboratory Technical Report, UCRL-100267, December 1988.*

[7] J. T. Yang, B. B. Tsai, and G. L. Tsai, *"Separated-Reattaching Flow over a Backstep with Uniform Normal Mass Bleed", Journal of Fluid Engineering, Vol. 116 , pp. 29- 35, 1994.*

Method	N-iter	error-u	error-v	error-p	error-f	time
AD100	100	.13E+01	.76E-02	.75E+00	.76E-04	.156E+03
ADI	461	.10E-01	.11E-02	.71E-02	.44E-06	.718E+03
AD100+J+BCGS	193	.42E-01	.36E-02	.72E+00	.91E-04	.203E+03
AD100+J+GMR	187	.47E-01	.19E-02	.72E+00	.91E-04	.202E+03
AD100+J	236	.48E-01	.21E-02	.72E+00	.91E-04	.226E+03
AD100+J+Cheb	431	.98E-02	.13E-01	.16E-01	.52E-04	.137E+04
AD100+J+ORES	187	.49E-01	.19E-02	.72E+00	.91E-04	.201E+03
AD100+IC+BCGS	645	.80E-02	.15E-01	.10E-01	.36E-04	.120E+04
AD100+IC+GMR	555	.14E-01	.12E-01	.63E-01	.94E-05	.936E+03
AD100+IC	302	.30E-01	.14E-02	.11E+00	.30E-04	.533E+03
AD100+IC+Cheb	305	.98E-02	.26E-02	.14E-02	.28E-04	.975E+03
AD100+IC+ORES	1353	.21E-01	.27E-01	.16E-01	.86E-05	.203E+04

TABLE 1. Best accuracy achieved with AD100 preprocessing

Method	N-iter	error-u	error-v	error-p	error-f	time
AD100	100	.13E+01	.76E-02	.75E+00	.76E-04	.156E+03
ADI	347	.50E-01	.15E-02	.26E-01	.21E-05	.386E+03
AD100+J+BCGS	187	.50E-01	.36E-02	.72E+00	.91E-04	.200E+03
AD100+J+GMR	185	.50E-01	.19E-02	.72E+00	.91E-04	.201E+03
AD100+J	232	.50E-01	.21E-02	.72E+00	.91E-04	.225E+03
AD100+J+Cheb	258	.50E-01	.48E-01	.25E-01	.44E-04	.785E+03
AD100+J+ORES	185	.50E-01	.19E-02	.72E+00	.91E-04	.200E+03
AD100+IC+BCGS	494	.50E-01	.14E-01	.21E-01	.74E-04	.955E+03
AD100+IC+GMR	273	.50E-01	.20E-01	.55E-01	.32E-04	.462E+03
AD100+IC	277	.50E-01	.14E-02	.11E+00	.30E-04	.500E+03
AD100+IC+Cheb	226	.50E-01	.29E-02	.13E-02	.28E-04	.829E+03
AD100+IC+ORES	238	.50E-01	.90E-02	.27E-01	.29E-04	.396E+03

TABLE 2. Experimental results of various methods with AD100 preprocessing

Method	N-iter	error-u	error-v	error-p	error-f	time
AD200	200	.38E+00	.30E-02	.19E+00	.14E-04	.310E+03
ADI	338	.57E-01	.16E-02	.28E-01	.24E-05	.523E+03
AD200+J+BCGS	244	.57E-01	.30E-02	.18E+00	.20E-04	.344E+03
AD200+J+GMR	242	.57E-01	.30E-02	.18E+00	.20E-04	.333E+03
AD200+J+ORES	242	.57E-01	.30E-02	.18E+00	.20E-04	.315E+03
AD200+J	282	.57E-01	.30E-02	.18E+00	.20E-04	.355E+03
AD200+J+Cheb	307	.57E-01	.14E-01	.18E-01	.20E-04	.636E+03
AD200+IC+BCGS	360	.58E-01	.12E-01	.18E-01	.10E-03	.633E+03
AD200+IC+GMR	279	.55E-01	.18E-01	.73E-01	.33E-04	.443E+03
AD200+IC+ORES	289	.56E-01	.16E-01	.19E-01	.28E-04	.464E+03
AD200+IC+Cheb	276	.57E-01	.30E-02	.11E-02	.29E-04	.712E+03
AD200+IC	271	.57E-01	.13E-02	.11E+00	.31E-04	.427E+03

TABLE 3. Experimental results of various methods with AD200 preprocessing

Method	N-iter	error-u	time	ratio1	time-AD100	ratio2
AD100	100	.13E+01	.156E+03	0.0000	.000E+00	0.0000
ADI	347	.50E-01	.386E+03	1.0000	.230E+03	1.0000
AD100+J+BCGS	187	.50E-01	.200E+03	0.5181	.044E+03	0.1913
AD100+J+GMR	185	.50E-01	.201E+03	0.5207	.045E+03	0.1957
AD100+J+ORES	185	.50E-01	.200E+03	0.5181	.044E+03	0.1913
AD100+J	232	.50E-01	.225E+03	0.5829	.069E+03	0.3000
AD100+J+Cheb	258	.50E-01	.785E+03	2.0337	.629E+03	2.7348
AD100+IC+BCGS	494	.50E-01	.955E+03	2.4741	.799E+03	3.4739
AD100+IC+GMR	273	.50E-01	.462E+03	1.1969	.306E+03	1.3304
AD100+IC+ORES	238	.50E-01	.396E+03	1.0259	.240E+03	1.0435
AD100+IC	277	.50E-01	.500E+03	1.2953	.344E+03	1.4957
AD100+IC+Cheb	226	.50E-01	.829E+03	2.1477	.673E+03	2.9261

TABLE 4. Efficiency of various methods with AD100 preprocessing

Method	N-iter	error-u	time	ratio1	time-AD200	ratio2
AD200	200	.38E+00	.310E+03	0.0000	.000E+00	0.0000
ADI	38	.57E-01	.523E+03	1.0000	.213E+03	1.0000
AD200+J+BCGS	244	.57E-01	.344E+03	0.6577	.034E+03	0.1596
AD200+J+GMR	242	.57E-01	.333E+03	0.6367	.023E+03	0.1080
AD200+J+ORES	242	.57E-01	.315E+03	0.6023	.005E+03	0.0235
AD200+J	282	.57E-01	.355E+03	0.6788	.045E+03	0.2113
AD200+J+Cheb	307	.57E-01	.636E+03	1.2161	.326E+03	1.5305
AD200+IC+BCGS	360	.58E-01	.633E+03	1.2103	.323E+03	1.5164
AD200+IC+GMR	279	.55E-01	.443E+03	0.8470	.133E+03	0.6244
AD200+IC+ORES	289	.56E-01	.464E+03	0.8872	.154E+03	0.7230
AD200+IC	271	.57E-01	.427E+03	0.8164	.117E+03	0.5493
AD200+IC+Cheb	276	.57E-01	.712E+03	1.3614	.402E+03	1.8873

TABLE 5. Efficiency of various methods with AD200 preprocessing

Method	N-iter	error-u	error-v	error-p	error-f	time	time-AD
AD100	100	.13E+01	.76E-02	.75E+00	.76E-04	.156E+03	.000E+00
AD200	200	.38E+00	.30E-02	.19E+00	.14E-04	.312E+03	.000E+00
ADI	334	.60E-01	.16E-02	.30E-01	.25E-05	.522E+03	
AD100+ J+BCGS	180	.60E-01	.36E-02	.72E+00	.90E-04	.197E+03	.041E+03
AD100+ J+GMR	179	.60E-01	.19E-02	.72E+00	.91E-04	.198E+03	.042E+03
AD100+ J	214	.60E-01	.21E-02	.72E+00	.91E-04	.217E+03	.061E+03
AD100+ J+Cheb	246	.60E-01	.12E-01	.26E-01	.47E-04	.738E+03	.582E+03
AD100+ IC+Cheb	220	.60E-01	.29E-02	.98E-03	.28E-04	.820E+03	.664E+03
AD200+ J+BCGS	243	.60E-01	.30E-02	.18E+00	.20E-04	.343E+03	.031E+03
AD200+ J+GMR	241	.60E-01	.30E-02	.18E+00	.20E-04	.332E+03	.020E+03
AD200+ J	277	.60E-01	.30E-02	.18E+00	.19E-04	.352E+03	.040E+03
AD200+ J+Cheb	304	.60E-01	.27E-01	.29E-01	.47E-04	.625E+03	.313E+03
AD200+ IC+Cheb	285	.60E-01	.30E-02	.48E-03	.29E-04	.753E+03	.441E+03

TABLE 6. Comparison of two choices of preprocessing

Grid	(N)	No. of iterations (u, v, p, f)	Sequential	Parallel	Speedup (S/P)
128×80	10240	$(8, 8, 12, 12)$	.4578E3	.3238E3	1.41
		$(16, 16, 24, 24)$	.6756E3	.3773E3	1.79
256×160	20960	$(8, 8, 12, 12)$	.7642E4	.4934E4	1.55
		$(16, 16, 24, 24)$	.1007E5	.6219E4	1.62

Table 7. Speedup of parallel ADI method

Department of Mathematics Fu Jen University
E-mail address: kcjeasun.math.fju.edu.tw

National Center for High-Performance Computing P.O. Box 19-136, Hsinchu, Taiwan, R.O.C.
E-mail address: c00dle00nchc.gov.tw

National Center for High-Performance Computing P.O. Box 19-136, Hsinchu, Taiwan, R.O.C.
E-mail address: c00fox00nchc.gov.tw