CS 458 Project Results
Jacob Schroder
Fall, 2005

1 Introduction

As an extension of my summer work with Dr. Tuminaro and Dr. Howle, I tested three new strategies for calculating an approximate inverse of the Schur complement, $\widetilde{S}^{-1}$. $\widetilde{S}^{-1}$ is in turn used to update rows or columns that correspond to boundaries in the F_p matrix used in the pressure-convection diffusion preconditioner in IFISS [1]. It is not well-understood what boundary conditions should be used for the pressure space preconditioner as pressure boundary conditions are not always defined by the physical problem. The overall aim of testing different F_p update strategies with different $\widetilde{S}^{-1}$'s was too find a cheap strategy that was insensitive to the boundary conditions used in the pressure-convection diffusion preconditioner.

2 Testing Framework

2.1 Problem Description

Below is a brief description of the testing framework and previous results. The new update strategies are tested in the same manner as during my Summer 2005 work at Sandia. The backward-facing step problem in IFISS was used with grid sizes 4 and 5. [2] The following Reynolds numbers are considered: $\mathfrak{Re}$ = 10, 100, 200. A Q2-Q1 discretization for the velocity - pressure spaces is used. The system solved by GMRES is the final linear system generated by Picard iterations with a nonlinear tolerance of 10^{-5} and a maximum iteration count of 9.

	Grid 4	Grid 5	Grid 6
Pressure Space	24×8	48×16	96×32
Velocity Space	48×16	96×32	192×64

Table 1: Test Problem Grid Sizes

Eight Dirichlet boundary condition combinations for the pressure-convection diffusion preconditioner were tested; all non-Dirichlet boundaries were Neumann. Results will be shown; however, for only the outflow and the inflow cases, as these were the "worst" and "best" cases, respectively.

[1] Incompressible Flow and Iterative Solver Software by David Silvester, Howard Elman, Alison Ramage

[2] Grid size 6 was too computationally demanding for Chris' Huygens.

1. Inflow
2. Outflow
3. Top
4. Bottom
5. Inflow and outflow
6. Inflow and bottom
7. Inflow and top
8. Inflow, top and bottom

2.2 Summary of Summer 2005 Results

The matrix system generated by IFISS for my test problem, the backward-facing step, is

$$\begin{bmatrix} F & B^T \\ B & -\frac{1}{\nu}C \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ g \end{bmatrix}. \tag{1}$$

$C = 0$ for our test problem because of the *Q2-Q1* discretization used. In the pressure convection-diffusion preconditioning method, the Schur complement matrix is approximated as

$$S \approx M_S = A_p F_p^{-1} Q, \tag{2}$$

where Q is the pressure mass matrix associated with the pressure discretization (or a spectrally equivalent approximation to it), and A_p and F_p are discrete Laplace and convection-operators defined on the pressure space. The actual Schur complement with $C = 0$ is,

$$S = BF^{-1}B^T. \tag{3}$$

By rearranging the terms in Equation 2, we arrive at an update for F_p at its boundary conditions, bcs,

$$F_p(bcs, :) = Q(bcs, :)(S^{-1}A_p)\,, \tag{4}$$

where Q is the mass matrix. IFISS already produces naive versions of the F_p and A_p operators, but the use of these naive versions did not exhibit acceptable convergence for a wide range of boundary condition combinations. Convergence results using the F_p and A_p provided by IFISS are shown in Tables 2 and 3. Convergence data are GMRES iterations.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	22	31	43
5	25	33	41
6	30	42	47

Table 2: Inflow only Dirichlet, GMRES iterations

My summer work produced two promising strategies for approximating $\widetilde{S}^{-1}$. A short description and results for these two methods are described below.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	36	59	79
5	44	500+	500+
6	54	500+	500+

Table 3: Outflow only Dirichlet, GMRES iterations

- Ideal boundary condition updates of F_p yielded mesh independence and good convergence as shown in Tables 4 and 5. These updates were essentially exact calculations of Equation 4. We then attempted approximations of this expensive update.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	12	23	37
5	14	21	34

Table 4: Inflow only Dirichlet with F_p update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	12	22	39
5	14	20	32

Table 5: Outflow only Dirichlet with F_p update, GMRES iterations

- We found that updates within the F_p operator of the form of Equation 4 worked better than analogous updates within the A_p operator.
- A combination of probing and ILU was used to update the F_p operator and while being computationally feasible, this method yielded good, but not excellent convergence. Table 6 and Table 7 contain GMRES iterations using F_p updates of the form:

$$\begin{aligned} &S = probing(B\ F^{-1}\ B'); \\ &[L, U] = luinc(S); \qquad \qquad \%compute\ ILU \\ &Fp(bcs, :) = Q(bcs, :) * (U\backslash(L\backslash A_p)); \qquad (5) \end{aligned}$$

Grid Param	$\mathfrak{Re} = 10$	100	200
4	20	38	77
5	23	32	70
6	27	30	44

Table 6: Inflow only Dirichlet with F_p update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	25	50	82
5	31	41	72
6	34	50	65

Table 7: Outflow only Dirichlet with F_p update, GMRES iterations

- Another promising method of approximating the F_p update is by using ILU twice, as shown below. However, this method is very expensive, which is reflected in the density of the resulting factors of $luinc(S)$. Table 8 and Table 9 contain GMRES iterations using F_p updates of the form:

$$
\begin{aligned}
&[L, U] = luinc(F, `0'); && \%compute\ ILU \\
&S = B * (U \backslash (L \backslash B^T)); && \\
&[L, U] = luinc(S, `0'); && \%compute\ ILU \\
&Fp(bcs, :) = Q(bcs, :) * (U \backslash (L \backslash A_p)); &&
\end{aligned} \tag{6}
$$

Grid Param	$\mathfrak{Re} = 10$	100	200
4	14	26	40
5	16	24	33
6	20	24	31

Table 8: Inflow only Dirichlet with F_p update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	15	26	44
5	17	27	38
6	21	32	39

Table 9: Outflow only Dirichlet with F_p update, GMRES iterations

In an effort to gauge computational cost the number of non-zeroes for the operators resulting from the above two methods was recorded. As Table 10 shows, using ILU twice is much more costly in this respect.

Grid Param	Probing - Method 5	*luinc()* - Method 6
4	1,681	43,681
5	6,529	591,361

Table 10: Number of Non-zeros in S

Method 6 resulted in an $\widetilde{S}^{-1}$ which had the same number of non-zeros as the actual Schur complement. The sparsity of S is a good measure of an update strategy's cost because it reflects both the cost of calculating the approximate S^{-1} and the resulting matrix-matrix multiplies with S^{-1} to update F_p.

2.3 New Strategies

The three new strategies for calculating an approximate Schur complement follow.

1. $\widetilde{S}^{-1} \approx SPAI(B * SPAI(F) * B')$, where SPAI is used twice to calculate an approximate Schur complement.
2. $\widetilde{S}^{-1} \approx SPAI(B * ILU(F, 0) * B')$, where ILU-0 is used to approximate an inverse to the $1-1$ block which is then used by SPAI to calculate an approximate Schur complement.
3. $\widetilde{S}^{-1} \approx SPAI(Probe(B, F, B'))$, where a probed approximation to $\widetilde{S}$ is formed and then approximately inverted by SPAI.[3]

3 Results

Below are GMRES iterations, number of non-zeroes in the resulting approximate inverses, and MATLAB pseudo-code for the three new strategies. Convergence results for the different strategies presented are for inflow only Dirichlet boundary conditions and outflow only Dirichlet boundary conditions as these two are the "best" and "worst" cases. *bcs* refers to boundary indices.

3.1 SPAI-SPAI

Below is MATLAB pseudocode for this method, using F_p column updates.

$$\begin{aligned}
&\widetilde{F}^{-1} = SPAI(F);\\
&\widetilde{S} = B\ \widetilde{F}^{-1}\ B';\\
&\widetilde{S}^{-1} = SPAI(S);\\
&F_p(:, bcs) = Q\ \ (\widetilde{S}^{-1}\ A_p(:, bcs));
\end{aligned} \tag{7}$$

[3]Chris Siefert's MATLAB structured probing code was used.

The two below tables hold GMRES convergence data for the column updates.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	20	34	51
5	24	39	56

Table 11: Outflow only Dirichlet with F_p column update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	23	33	49
5	28	43	57

Table 12: Inflow only Dirichlet with F_p column update, GMRES iterations

The below two tables hold convergence data for F_p row updates which are identical to column updates only

$$Fp(bcs,:) = Q(bcs,:)\ \widetilde{S}^{-1}\ A_p;$$

500+ refers to GMRES not converging after 500 iterations.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	31	54	68
5	31	500+	500+

Table 13: Outflow only Dirichlet with F_p row update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	23	34	51
5	26	34	49

Table 14: Inflow only Dirichlet with F_p row update, GMRES iterations

The number of non-zeroes for $\widetilde{S}$ were constant for all Reynolds numbers and are

Grid Param	
4	7102
5	30084

Table 15: Number of Non-zeros in S^{-1}

The number of non-zeroes for $\widetilde{S}^{-1}$ are

Grid Param	$\mathfrak{Re} = 10$	100	200
4	2774	3892	4745
5	9951	12175	13829

Table 16: Number of Non-zeros in S^{-1}

3.2 ILU-SPAI

Below is MATLAB pseudocode for this method, using F_p column updates.

$$\begin{aligned} &[L_f, U_f] = ILU(F, 0); \\ &\widetilde{S} = B(U_f \backslash (L_f \backslash B')); \\ &\widetilde{S}^{-1} = SPAI(\widetilde{S}); \\ &Fp(:, bcs) = Q\ (\widetilde{S}^{-1}\ A_p(:, bcs)); \end{aligned} \tag{8}$$

The two below tables hold GMRES convergence data for the column updates.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	16	29	45
5	17	28	42

Table 17: Outflow only Dirichlet with F_p column update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	16	27	45
5	19	25	37

Table 18: Inflow only Dirichlet with F_p column update, GMRES iterations

The below two tables hold convergence data for F_p row updates which are identical to column updates only

$$Fp(bcs,:) = Q(bcs,:)\ \ \widetilde{S}^{-1}A_p;$$

500+ refers to GMRES not converging after 500 iterations.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	24	49	61
5	29	500+	82

Table 19: Outflow only Dirichlet with F_p row update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	18	29	49
5	19	26	39

Table 20: Inflow only Dirichlet with F_p row update, GMRES iterations

The number of non-zeroes for $\widetilde{S}^{-1}$ are

Grid Param	$\mathfrak{Re} = 10$	100	200
4	3029	3644	4244
5	11779	11924	13339

Table 21: Number of Non-zeros in S^{-1}

The number of non-zeros for $\widetilde{S}$ resulting from using ILU were same as above in the section summarizing previous work.

3.3 PROBE-SPAI

Below is MATLAB pseudocode for this method, using F_p column updates.

$$\begin{aligned} &\widetilde{S} = probe(B, F, B');\\ &\widetilde{S}^{-1} = SPAI(\widetilde{S});\\ &Fp(:, bcs) = Q\ (\widetilde{S}^{-1}A_p(:, bcs)); \end{aligned} \tag{9}$$

The two below tables hold GMRES convergence data for the column updates.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	22	34	58
5	30	39	55

Table 22: Outflow only Dirichlet with F_p column update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	25	33	54
5	35	39	53

Table 23: Inflow only Dirichlet with F_p column update, GMRES iterations

The below two tables hold convergence data for F_p row updates which are identical to column updates only

$$Fp(bcs,:) = Q(bcs,:)\ \ \widetilde{S}^{-1}A_p;$$

500+ refers to GMRES not converging after 500 iterations.

Grid Param	$\mathfrak{Re} = 10$	100	200
4	34	53	76
5	38	500+	500+

Table 24: Outflow only Dirichlet with F_p row update, GMRES iterations

Grid Param	$\mathfrak{Re} = 10$	100	200
4	23	33	61
5	25	31	48

Table 25: Inflow only Dirichlet with F_p row update, GMRES iterations

The number of non-zeroes for $\widetilde{S}^{-1}$ are

Grid Param	$\mathfrak{Re} = 10$	100	200
4	1364	2559	2830
5	4265	6535	9947

Table 26: Number of Non-zeros in S^{-1}

The number of non-zeros for the probed $\widetilde{S}$ were same as in the above section summarizing previous work.

4 Timings

I did do some timings as you had suggested, but I was thwarted by the SPAI code. The SPAI code did not let me suppress all output to the standard output. This is most likely the reason that the timings for SPAI were dismally slow. I ran the tests remotely off of Huygens (Chris' machine) and sending the standard out to my laptop slowed SPAI down a great deal. I also tried just running MATLAB in the background and writing everything to a file, but this still resulted in a very slow SPAI. The version of SPAI used is not based on M-files, so removing the *sprintf()* statements was not something I could readily do. I also don't have access to change files outside of my home directory on Huygens. It would not be terribly difficult, though, in the future to recompile SPAI with the print statements commented out. I think the SPAI code is written in C/C++.

5 Conclusions

Some conclusions on the results are

1. Column updates for Fp were noticeably superior to row updates. For outflow only Dirichlet boundary conditions, row updates even resulted in a lack of convergence in at least one case for each method. It does not appear as if the row updates are feasible strategies for updating F_p. Column updates also resulted in overall better GMRES convergence rates. I have thought about possible reasons for this, but I cannot as of now postulate something reasonable.

2. The lack of GMRES convergence for the outflow only Dirichlet boundary conditions used with F_p row updates can be fixed by drastically increasing the number of non-zeroes in the $\widetilde{S}^{-1}$ produced by SPAI for each of the three new strategies. Usually, an increase by a factor of 3 or more to the number of non-zeroes was required to obtain convergence.

3. All three methods exhibited grid independence for F_p column updates at $\mathfrak{Re} = 200$, but for lower Reynolds numbers this was not the case.

4. Of the three new strategies tried, ILU-SPAI exhibited slightly better convergence results than the other two.

5. Using SPAI requires a fair amount of tweaking to get a good approximate inverse. Tweaking the default parameters, such that a moderate increase in the number of non-zeroes resulted, can halve the convergence rate of GMRES. This is a big drawback when compared to probing, where you only have to choose one of a few coloring algorithms.

6. SPAI is attractive because it gives fine control over the number of non-zeroes in the resulting approximate inverse, which allows for the calculation of approximate inverses that are "cheap" to do calculations with.

This allowed me to create a $\widetilde{S}$ that is as sparse as the $\widetilde{S}$ from probing. This is in stark contrast to using ILU twice as shown in the summary of previous work, where the resulting approximate inverse had far more nonzeroes than any other method. This made the resulting matrix-matrix multiplies to update F_p very expensive.

Some worthwhile possibilities for future work are

1. Conduct timings and develop a better understanding of the computational cost of SPAI.

2. Test grid size 6 for the 3 new strategies.

3. Dr. Tuminaro has suggested considering doing SPAI or probing only near the boundaries, as opposed to the whole matrix.