Weierstraß-Institut

für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 0946 – 8633

Efficient linear solvers for incompressible flow simulations using Scott-Vogelius finite elements

Ben Cousins[1], Sabine Le Borne[2], Alexander Linke[3], Leo G. Rebholz[1], Zhen Wang[4]

submitted: August 1, 2013

[1] Clemson University
Department of Mathematical Sciences
Clemson, SC 29634
USA
email: bcousin@clemson.edu
rebholz@clemson.edu

[2] Tennessee Technological University
Department of Mathematics
Cookeville, TN 38505
USA
email: sleborne@tntech.edu

[3] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
email: Alexander.Linke@wias-berlin.de

[4] Oak Ridge National Laboratory
National Center for Computational Sciences
Oak Ridge TN 37831
USA
email: wangz@ornl.gov

No. 1821

Berlin 2013

2010 *Mathematics Subject Classification.* 76D07, 65F05, 65F08.

Key words and phrases. Scott-Vogelius elements, linear solvers, static condensation, augmented Lagrangian preconditioning, H-LU.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Abstract

Recent research has shown that in some practically relevant situations like multi-physics flows [11] divergence-free mixed finite elements may have a significantly smaller discretization error than standard nondivergence-free mixed finite elements. In order to judge the overall performance of divergence-free mixed finite elements, we investigate linear solvers for the saddle point linear systems arising in $((P_k)^d, P_{k-1}^{disc}))$ Scott-Vogelius finite element implementations of the incompressible Navier-Stokes equations. We investigate both direct and iterative solver methods. Due to discontinuous pressure elements in the case of Scott-Vogelius elements, considerably more solver strategies seem to deliver promising results than in the case of standard mixed finite elements like Taylor-Hood elements. For direct methods, we extend recent preliminary work using sparse banded solvers on the penalty method formulation to finer meshes, and discuss extensions. For iterative methods, we test augmented Lagrangian and $\mathcal{H} - LU\ R$ preconditioners with GMRES, on both full and statically condensed systems. Several numerical experiments are provided that show these classes of solvers are well suited for use with Scott-Vogelius elements, and could deliver an interesting overall performance in several applications.

1 Introduction

We study efficient solving techniques for linear systems arising from finite element (FE) computations of the steady Navier-Stokes (NS) and related equations, when using Scott-Vogelius (SV) elements. On a domain Ω , the steady NS system is given by

$$u \cdot \nabla u + \nabla p - Re^{-1}\Delta u = f \; in \, \Omega \tag{1.1}$$

$$\nabla \cdot u = 0 \; in \, \Omega \tag{1.2}$$

where (u, p) are the unknowns representing velocity and pressure, Re is the Reynolds number, and f represents external forcing. This system must be coupled to appropriate boundary and initial conditions. For simplicity, we assume Dirichlet velocity boundary conditions, and extensions to other common boundary conditions can be done in the usual way [34]. Under this assumption, denoting the $L^2(\Omega)$ inner product by $(\cdot, \cdot)$ and defining the finite dimensional spaces by $(X_h, Q_h) \subset (H_0^1(\Omega)^d, L_0^2(\Omega))$, discretizing (1.1)-(1.2) with the FE method leads to the nonlinear problem: Given $f \in H^{-1}(\Omega)$, find $(u_h, p_h) \in (X_h, Q_h)$ satisyfing $\forall (v_h, q_h) \in (X_h, Q_h)$,

$$
\begin{aligned}
-(p_h, \nabla \cdot v_h) + Re^{-1}(\nabla u_h, \nabla v_h) + (u_h \cdot \nabla u_h, v_h) &= (f, v_h), && (1.3)\\
(\nabla \cdot u_h, q_h) &= 0. && (1.4)
\end{aligned}
$$

Since this system is nonlinear, solving it requires iteration of linear system solves via methods such as Newton or Picard; we use only Newton in this work. Solving these linear systems is the primary interest of this work, and since incompressible, viscous flow simulations arise in many engineering applications, the ability to efficiently solve such linear systems is of great interest across the scientific community.

The choice of the discrete spaces X_h and Q_h are well-known to be of paramount importance. Typically, these spaces are chosen to satisfy an inf-sup stability condition [12, 7], as well as optimal approximation properties [12, 21]. Recent work has found that an additional important criterion can be that the spaces enforce mass conservation strongly, i.e. $\nabla \cdot u_h = 0$ [8, 9, 25, 24, 26, 23, 22]. Satisfying this criterion leads to more physically relevant solutions, decouples the pressure error from the velocity error, and removes possible instabilities that can arise from poor discrete mass conservation [23]. The specific element choice made in these works to achieve pointwise mass conservation of discrete solutions is the Scott-Vogelius element pair. In the mid-1980's, L.R. Scott and M. Vogelius pioneered the use of the $(X_h, Q_h) := ((P_k)^d(\tau_h), P_{k-1}^{disc}(\tau_h))$ mixed FE pair on a triangular/tetrahedral mesh τ_h, for Stokes-type problems [32, 31, 35, 36]. Since $\nabla \cdot X_h \subset Q_h$ (which is quite uncommon in inf-sup stable pairs), the strong enforcement of mass conservation is easily seen by choosing $q_h = \nabla \cdot u_h$ in (1.4).

Even though SV elements were developed in the 1980's, their use did not immediately catch on as a popular element choice since, at the time, inf-sup stability was not known to hold for lower order elements. In 1994, [28] Qin showed that if using a mesh created as a barycenter refinement of a regular triangular mesh, then the SV pair is inf-sup stable in 2D for $k \geq 2$. For 3D, in 2005, [38] S. Zhang showed that inf-sup stability also holds if $k \geq 3$ on a barycenter refinement of a regular tetrahedral mesh. These results have led to a renewed interest in using SV elements. Indeed, recent work using SV elements with this macro-element structure (see Figure 1) have reported excellent results for approximating solutions to NS and related equations, both in 2D and 3D [23, 22, 26, 8, 24, 9].

With the resurgence of interest in SV elements for solving NS and related problems comes the need for efficient solvers for the arising linear systems. It is the purpose of this paper to explore several methods of solving linear systems arising from (1.3)-(1.4), when SV elements are used with $k = d$ and on barycenter refined meshes. Recent results of S. Zhang [39, 40] showed SV elements are inf-sup stable when $k = d-1$ if a so-called Powell-Sabin[39] mesh is used. However, we consider such meshes to be more restrictive than a simple barycenter refinement, and so we believe the case of $k = d$ on a barycenter refined mesh is the most practically relevant.

This paper is arranged as follows. Section 2 tests a direct solver on the penalty method formulation of the NS system. Recent work in [25] has shown that in settings where SV elements are

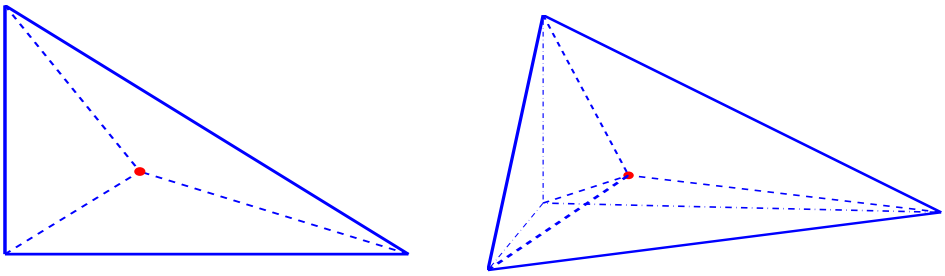

Figure 1: 2D (left) and 3D (right) macro-element sufficient for LBB stability of $((P_k)^d, P_{k-1})$ SV elements when $k \geq d$. Dashed lines represent barycenter refinements.

inf-sup stable, as the penalty parameter $\to \infty$, the penalty method formulation solution converges to the SV solution (assuming it is unique; if not, there is a result for subsequences to converge to a SV solution). Work in [27] showed that the linear systems arising from this technique can be solved efficiently with a sparse solver, and give excellent accuracy, and herein we extend this testing to finer meshes and discuss extensions. Section 3 details the implementation and motivation of several solving methods used herein, including static condensation, AL preconditioning, and $\mathcal{H}$-LU preconditioning. In section 4, we provide several numerical experiments that compare the methods, and show when each are effective. Conclusions and future directions are discussed in Section 5.

2 Direct methods for solving linear systems arising from SV discretizations

It was found in [27] that in the setting where SV elements are inf-sup stable, the classical penalty method of Temam [33] used with direct solvers can be an effective approach for approximating NS solutions. This approach entails eliminating the saddle point structure in the NS problem at the continuous level by using an artificial pressure regularization in the continuity equation, then using the altered equation to eliminate the pressure in the momentum equation. In the steady case, after discretizing the steady NSE, we get

$$\epsilon^{-1}(\nabla \cdot u_h, \nabla \cdot v_h) + Re^{-1}(\nabla u_h, \nabla v_h) + (u_h \cdot \nabla u_h, v_h) \quad = \quad (f, v_h), \tag{2.1}$$

instead of (1.3)-(1.4). Note if ϵ is small, the divergence constraint will be enforced through the least squares penalty (grad-div stabilization) term $\epsilon^{-1}(\nabla \cdot u_h, \nabla \cdot v_h)$. Pressure is recovered by $p_h = -\epsilon^{-1}(\nabla \cdot u_h)$.

The system (2.1) can have numerical issues when ϵ is small for two reasons. First, there is typically no guarantee that the divergence free subspace of X_h has optimal approximation properties. Thus, solutions to (2.1) can be suboptimal, as is shown in [9]. However, in settings where SV elements are inf-sup stable (e.g. barycenter refined triangular meshes and $k \geq d$), we do know that the divergence free subspace of X_h has optimal approximation properties [38, 28], and so taking ϵ small in these settings will not cause suboptimal accuracy. Second, the condition number of the linear systems resulting from (2.1) will scale with ϵ^{-1}. Hence taking ϵ too small will lead to accuracy problems in the linear solves. It was shown in [27] that this penalty method approach (2.1), when

used on a barycenter refined mesh, $k = d$, and $\epsilon = 10^{-4}$ can give excellent results, and the resulting linear systems were able to be solved quite efficiently using Matlab's 'backslash'. However, those results were for rather small systems, and we extend this testing here to much larger systems.

We give two numerical tests for this method, one in 2D and one in 3D. The nonlinear system (2.1) is resolved using Newton's method, and for all tests, we take the initial condition $u_0 = 0 +$ boundary conditions, as the initial guess. The stopping criterion for Newton is a relative update size less than 10^{-8} in the natural solution norm

$$\|(u,p)\| := \sqrt{\|\nabla u\|^2 + \|p\|^2}.$$

The 2D computations were run with Matlab 2011a, on a 2x 2.66 GHz Quad-Core Intel Xeon processor, using 32GB 1066 MHz DDR3 memory. The 3D computations were run with Matlab 2011b, on a 4x 2.40 GHz Ten-Core Intel Xeon processor, using 1024GB 1067Mhz unregistered DDR3 memory.

2.1 2D test problem for direct solver: 2D driven cavity

Our first test is the benchmark problem of 2D driven cavity flow. The domain is the unit square, no-slip boundaries are enforced on the sides and bottom, and on the top (the lid), $u_{\text{lid}} = \langle 1, 0 \rangle^T$ is imposed as a Dirichlet boundary condition. We test with $Re := \nu^{-1} = 100$, and the solution is known to be steady. We use (P_2, P_1^{disc}) SV elements on barycenter refined meshes generated from barycenter refined uniform triangulations, with the artificial compressibility parameter is chosen to be $\epsilon = 10^{-4}$.

Timings, dof, and divergence error for these tests are given in Table 1. In all cases, the divergence error is of the order of 10^{-5} or better, which is as expected since $\epsilon = 10^{-4}$. Solve times are observed to be quite fast, and scale very well with dof until the finest meshes are used, and the scaling of the finest two meshes is nearly linear. A plot of the computed velocity solution from the finest mesh is shown in Figure 2, and agrees very well with the solutions to this problem found in the literature [20]. Solutions on the other meshes gave indistinguishable plots.

We also ran this test problem for Re=1,000, and found similar results. That is, computed solutions matched the known solution very well, and timings of the linear solves were nearly identical to the Re=100 case on respective meshes, which is as expected for a direct solver.

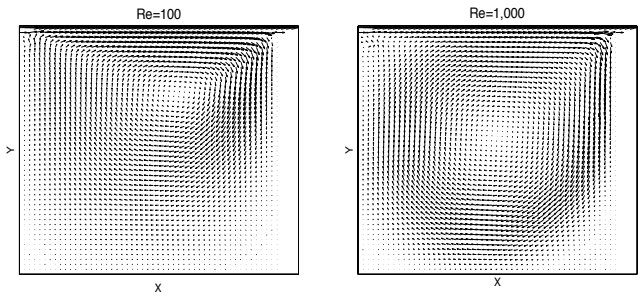

Figure 2: Velocity solution for the 2D driven cavity with Re=100 and Re=1,000.

H	$\dim(X_h)$	Time (sec.)	$\|\nabla \cdot u_h\|$
1/16	7,258	0.07	7.29E-6
1/32	29,370	0.39	8.21E-6
1/64	117,106	1.65	8.94E-6
1/96	221,954	3.35	9.34E-6
1/128	394,242	6.59	9.60E-6
1/256	1,575,914	27.77	9.83E-6
1/512	4,718,590	140.71	1.07E-5

Table 1: Solve times, mass conservation and dof for (2.1) solved for the 2D driven cavity at Re=100, using Matlab 'backslash' as the direct solver.

2.2 3D test problem for direct solver: 3D driven cavity

We now investigate solving the 3D driven cavity problem for NS at Re=100, which is known to have a steady solution. Here $\Omega = (0,1)^3$, the moving lid is represented by the boundary condition $u(x,y,1) = \langle 1,0,0\rangle^T$, while the sides and bottom of the cavity are prescribed no-slip boundary conditions. We use $k = 3$, and compute on barycenter refined uniform meshes. The artificial compressibility parameter was chosen to be $\epsilon = 10^{-4}$, and the nonlinearity is resolved using Newton's method. Plots of the mid-sliceplances for $Re = 100$, $H = 1/6$ are shown in Figure 3, which agrees with plots given in the literature [37]. Plots for smaller H are indistinguishable from these, and so are omitted.

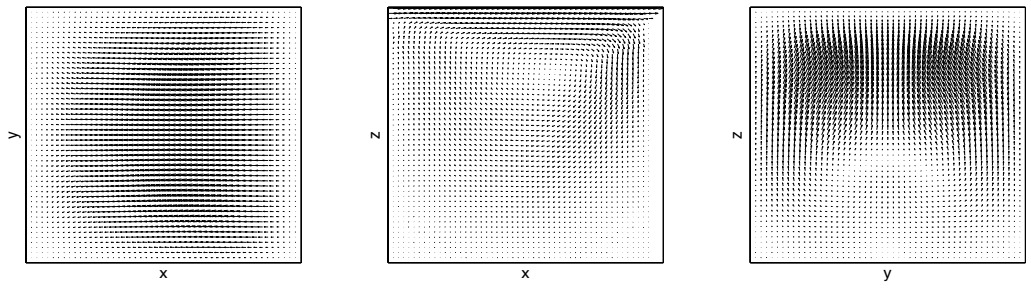

Figure 3: Midsliceplanes of the velocity fields for the solution of the 3D driven cavity (Re=100).

Solve times, divergence errors, and dof are shown in Table 2, and we observe the method is efficient even at higher dof; for the 3.7 million dof system, it takes about an hour to do the solve. The accuracy in mass conservation of the solutions is of the order of 10^{-5}; exact mass conservation cannot be expected since the penalty method system does not conserve mass exactly if $\epsilon > 0$.

2.3 Discussion

The results above indicate that the penalty method used with Matlab's 'backslash' direct solver can be quite effective on mid-size problems. In 2D, it worked well even on our finest mesh, which provided nearly 5 million velocity degrees of freedom (dof). In 3D, the efficiency of the method appears to scale well in the dof up to about 3 million velocity dof, but turns worse for even higher

H	$\dim(X_h)$	time (sec.)	$\|\nabla \cdot u_h\|$
1/2	3,189	0.06	4.05E-6
1/4	23,871	0.97	5.68E-6
1/6	78,897	4.93	6.45E-6
1/8	185,115	13.31	6.94E-6
1/10	359,373	42.48	8.15E-6
1/12	618,519	130.58	7.56E-6
1/14	979,401	244.47	7.68E-6
1/16	1,458,867	535.22	7.95E-6
1/18	2,073,765	1,181.00	8.10E-6
1/20	2,840,943	2,193.31	8.23E-6
1/22	3,777,249	3,711.66	8.34E-6
1/24	4,899,531	6,761.46	8.44E-6
1/26	6,224,637	11,258.09	8.53E-6
1/28	7,769,415	17,869.61	8.62E-6

Table 2: Solve times, divergence errors and dof for the 3D Re=100 driven cavity tests using (2.1) and Matlab's backslash for the linear solver.

numbers. Moreover, since the solver is direct, the solver's efficiency does not depend on Re, and we observed no deterioration of the solver when Re was increased.

It is important to note that the accuracy of the method is limited by the choice of the penalty parameter, and thus with a fixed ϵ, locking phenomena of the accuracy will occur on very fine meshes. We have found ϵ can be reduced below 10^{-4} without affecting the solver efficiency, but since conditioning of the matrices scales with ϵ^{-1}, taking ϵ too small can cause significant inaccuracy in the linear solves. One fix for finer meshes is to use the iterated penalty method discussed by S. Zhang in [40], which is given for the steady NS system by

$$\alpha(\nabla \cdot u_h^m, \nabla \cdot v_h) + Re^{-1}(\nabla u_h^m, \nabla v_h) + (u_h^m \cdot \nabla u_h^m, v_h) = (f, v_h) + \left(\nabla \cdot \sum_{j=0}^{m-1} u_h^j, \nabla \cdot v_h \right). \quad (2.2)$$

In a setting where SV is inf-sup stable, this iteration will converge to the optimally accurate SV solution. Note if only one iteration is performed, we recover the penalty method system (2.1). We also observe that for each subsequent iteration, this nonlinear system is the same as (2.1), except for the last term on the right hand side. Hence on finer meshes, accuracy locking can be avoided at the expense of doubling or tripling the number of linear solves performed. If one can save the factorization of the first solve (which may or may not be possible), this can be an efficient and accurate method. An important open question concerning (2.2) is whether preconditioned iterative methods can be developed to efficiently solve the resulting linear systems (i.e. after linearization by Newton or Picard), when α is large enough so that convergence can be reached quickly (e.g. $\alpha \geq 100$). The only work in this direction that we are aware of is a geometric multigrid algorithm by J. Schöberl[30] with the special property that the prolongation operator maps discretely divergence-free coarse grid functions to discretely divergence-free fine grid functions. For that, the author exploits additional knowledge about the basis of the space of discretely divergence-free functions for the P_2-P_0 element.

Finally, we note that the method described above does not appear to work better on statically condensed systems (see section 3.1). Roughly speaking, it is folklore (i.e. hoped) that good direct solvers will find and exploit the structure in the full system matrix, that would have allowed for static condensation in the first place. We performed tests both with Matlab's backslash and the direct solver PARDISO[29], both of which did not give improvement over solving the full system that would justify the extra work associated with performing the static reduction.

3 Iterative methods for solving linear systems arising from SV discretizations

We turn our attention now to iterative solvers for the linear systems arising from (1.3)-(1.4). As seen in the previous section, direct solvers fail to be efficient on sufficiently fine meshes and/or large problems. Iterative solvers offer an alternative that scales better with larger dof and increased number of processors. We investigate herein two methods, which seem well suited for use with SV: augmented Lagrangian based preconditioned GMRES, and $\mathcal{H}$-LU preconditioned GMRES. A key to the success of these methods is combining them with static condensation, which, when used with SV and its required macro-element structure when $k = d$, dramatically reduces the size of the global system matrices. This section will first discuss static condensation applied to SV systems on barycenter refined meshes, then introduce the preconditioners.

3.1 Static condensation of SV systems on barycenter-refined meshes

The classical idea of static condensation (SC) is to remove from the global system the dof with only local support on a macro-element, by locally solving for these internal dof in terms of the dof with support outside the macro-element. This creates a smaller global system solve, and after the global solve is performed, internal degrees of freedom are easily recovered. The use of SC in solving Stokes/NS type problems was made popular by Arnold, Brezzi and Fortin [1], who used it to remove the cubic bubble function of the (P_1^{bub}, P_1) element from the global linear system. With the improvements in algorithms for linear solvers in recent years, lately this process can be overlooked and even avoided, as a modest reduction size may not be worthwhile if the sparsity structure of the reduced system is adversely affected. For SV elements on barycenter refined meshes, however, a very significant size reduction is possible because the macro-element structure creates locally supported velocity dof, and since pressures are approximated by discontinuous polynomials, all pressure dof have only local support, although one pressure dof must remain on each macro-element to maintain the global coupling of pressure to the divergence constraint. We will show now that the size reduction offered by SC for SV is quite dramatic. Consider the $((P_2)^2, P_1^{disc})$ SV pair on a barycenter-refined triangulation. A diagram of the macro-element and the location of its dof is shown in Figure 4. Here we count 10 velocity nodes and 9 pressure nodes, and thus 20 velocity dof and 9 pressure dof, and observe that 8 internal velocity dof, and all 9 of the pressure dof, have support only in the macro-element. Denoting the (exterior) edge velocity dof by u_e, the locally supported velocity dof by u_i, a single pressure p_e, and the remaining 8 pressures by p_i, we have the following linear system resulting from a Newton or Picard iteration of (1.3)-(1.4) on each macro-element:

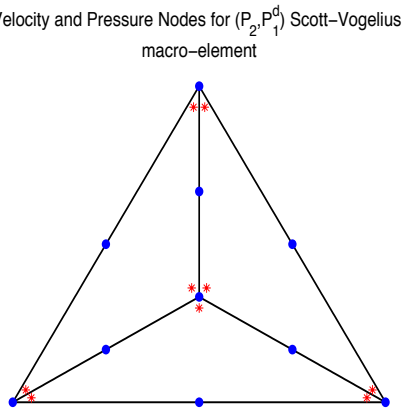

Figure 4: The 2D macro-element structure and degree of freedom locations for the $k = 2$ SV elements. Blue dots are for velocity nodes (so 2 velocity dof at each), and red asterisks are for pressure dof.

$$\begin{pmatrix} A_{11} & A_{12} & B_1^T & b_1^T \\ A_{21} & A_{22} & B_2^T & b_2^T \\ B_1 & B_2 & 0 & 0 \\ b_1 & b_2 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_e \\ u_i \\ p_i \\ p_e \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ g_1 \\ g_2 \end{pmatrix}, \tag{3.1}$$

where A_{11} is 12×12, A_{22} is 8×8, B_1 is 8×12, B_2 is 8×8, b_1 is 1×12, b_2 is 1×8, A_{12} is 12×8, and A_{21} is 8×12. Swapping the last row and column with the ninth row and column, respectively, gives

$$\begin{pmatrix} A_{11} & b_1^T & A_{12} & B_1^T \\ b_1 & 0 & b_2 & 0 \\ A_{21} & b_2^T & A_{22} & B_2^T \\ B_1 & 0 & B_2 & 0 \end{pmatrix} \begin{pmatrix} u_e \\ p_e \\ u_i \\ p_i \end{pmatrix} = \begin{pmatrix} f_1 \\ g_2 \\ f_2 \\ g_1 \end{pmatrix}. \tag{3.2}$$

Rewriting the 4×4 system's pieces as 2×2 blocks as

$$\widetilde{M}_{11} := \begin{pmatrix} A_{11} & b_1^T \\ b_1 & 0 \end{pmatrix}, \quad \widetilde{M}_{12} := \begin{pmatrix} A_{12} & B_1^T \\ b_2 & 0 \end{pmatrix}, \quad \widetilde{M}_{21} := \begin{pmatrix} A_{21} & b_2^T \\ B_1 & 0 \end{pmatrix},$$

$$\widetilde{M}_{22} := \begin{pmatrix} A_{22} & B_2^T \\ B_2 & 0 \end{pmatrix}, \quad x_e := \begin{pmatrix} u_e \\ p_e \end{pmatrix}, \quad x_i := \begin{pmatrix} u_i \\ p_i \end{pmatrix}, \quad \tilde{f}_1 := \begin{pmatrix} f_1 \\ g_2 \end{pmatrix}, \quad \tilde{f}_2 := \begin{pmatrix} f_2 \\ g_1 \end{pmatrix},$$

the local system can be represented by

$$\begin{pmatrix} \widetilde{M}_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{pmatrix} \begin{pmatrix} x_e \\ x_i \end{pmatrix} = \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \end{pmatrix}.$$

The key for reducing this system is to observe that $\widetilde{M}_{22}$ is nonsingular. We refer to [10] for the solvability of such systems, but a simple way to think of it is that if the entire domain were just this macro-element and Dirichlet boundary conditions were being used, then removing a pressure

dof, which is done in our system by the row and column swaps, creates a global system matrix identical to $\widetilde{M}_{22}$ which is known to be nonsingular. Therefore, we can write the second row of the system as

$$x_i = \widetilde{M}_{22}^{-1}(\widetilde{f}_2 - \widetilde{M}_{21}x_e),$$

which yields, after substitution into the first row,

$$\left(\widetilde{M}_{11} - \widetilde{M}_{12}\widetilde{M}_{22}^{-1}\widetilde{M}_{21}\right)x_e = \tilde{f}_1 - \widetilde{M}_{12}\widetilde{M}_{22}^{-1}\tilde{f}_2. \tag{3.3}$$

Assembly of a global system now only needs to contain u_e and p_e from each element. For this example, this means only 13 of the 29 local degrees of freedom contribute to the global system matrix. Note that since the remaining velocity dof are only on the exterior edges of each macro-element, the faces of the barycenter macro-element are faces of its interior elements, and only one pressure dof is retained on each macro-element.

An example of the reduction of the system sizes resulting from static condensation in 2D is shown in Table 3, when using $(X_h, Q_h) := ((P_2)^2, P_1^{disc})$ SV elements on a barycenter refined uniform triangulation of the unit square. The mesh is created by dividing into squares with side length H, then cutting the squares into 2 triangles, and finally a barycenter refinement is applied. (X_h^{sc}, Q_h^{sc}) represent the statically condensed velocity-pressure spaces. We observe from the table that the total degrees of freedom are cut dramatically.

Level	H	$dim(X_h)$	$dim(Q_h)$	Total (full)	$dim(X_h^{sc})$	$dim(Q_h^{sc})$	Total (reduced)
1	1/16	7,258	5,346	12,604	2,506	594	3,100
2	1/32	29,370	21,834	51,204	9,962	2,426	12,388
3	1/64	117,106	87,444	204,550	39,378	9,716	49,094
4	1/96	221,954	165,888	387,842	74,498	18,432	92,930
5	1/128	394,242	294,912	689,154	132,098	32,768	164,866

Table 3: System size reduction of $((P_2)^2, P_1^{disc})$ SV elements on a barycenter-refined uniform mesh of the unit square, by static condensation, for the 2D driven cavity test problem.

For the 3D case, analogous arguments and local matrix manipulation can be used to reduce the global system size. The usual choice of degree of the polynomial for velocities for SV in 3D is $k = 3$ (recall $k \geq d$ is required for stability), and so pressures are approximated with discontinuous quadratics. A barycenter refinement of a tetrahedron produces 4 tetrahedra, and so there will be 40 pressure dof per macro-element, 39 of which can be condensed (by the same argument as in the 2D case). For velocity, of the 125 total degrees of freedom provided by P_3 approximating polynomials on each macro-element, 45 have only local support. Thus the amount of condensation for this element is quite dramatic; however, for this case we must locally invert an 84×84 matrix on each macro-element. We contend this can be done efficiently and accurately, and can be distributed across processors if the assembly is done in parallel. A single inversion of a random 84×84 matrix takes on average 0.001 seconds in Matlab, on a Mac Workstation with 2 x 2.66 GHz Quad-Core Intel Xeon processor and 32 GB 1066 MHz DDR3 memory. This explicit inversion must be done as many times as there are macro-elements, and then matrix multiplication with the inverse must be done to recover the condensed dof. In our 3D computations, this added about 20% to our assembly time, but this increase is negligible compared to the savings in solve time. Note also that if the

local inverse matrices are saved, then recovery of the internal dof can be done with a single matrix multiplication. Otherwise, since this recovery is purely a local operation, it can be split across processors.

For a 3D example of the amount of reduction offered, consider τ_H to be a uniform tetrahedralization of the unit cube, where the unit cube is split into cubes with side length H, and then each cube is split into 6 tetrahedra. The finer mesh τ_h is then created as a barycenter refinement of τ_H, where each tetrahedron is split into 4 tetrahedra. The velocity-pressure space (X_h, Q_h) is defined by $((P_3)^3, P_2^{disc})$ elements, and (X_h^{sc}, Q_h^{sc}) is the statically condensed velocity-pressure spaces. Table 4 shows the dof associated with each of these spaces, for several mesh levels, and we observe the dramatic reduction of dof in the condensed system.

Level	H	$dim(X_h)$	$dim(Q_h)$	Total	$dim(X_h^{sc})$	$dim(Q_h^{sc})$	Total
1	1/2	3,189	1,920	5,109	1,029	48	1,077
2	1/4	23,871	15,630	39,231	6,591	384	6,975
3	1/6	78,897	51,840	130,787	20,577	1,296	21,873
4	1/8	185,115	122,880	307,995	46,875	3,072	49,947
5	1/10	359,373	240,000	599,373	89,373	6,000	95,373
6	1/12	618,519	414,720	1,033,239	151,959	10,368	162,327

Table 4: System size reduction of $((P_3)^3, P_2^{disc})$ SV elements on a barycenter-refined mesh, made by using static reduction.

3.2 A modified-AL preconditioner for GMRES

We now briefly present the modified AL-preconditioner of Benzi and Olshanskii [4], which is designed for use with GMRES, and discuss why it fits well for solving the condensed global system.

The system (1.3)-(1.4) leads to the block linear system

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{3.4}$$

Following [5], the equivalent AL formulation is

$$\begin{pmatrix} A + \gamma B^T W^{-1} B & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} \hat{f} \\ g \end{pmatrix}, \tag{3.5}$$

where $\hat{f} = f + \gamma B^T W^{-1} g$, W is an arbitrary symmetric positive definite (SPD) matrix and $\gamma > 0$. Typically W is taken to be the diagonal of the pressure mass matrix M_p. Denoting $A_\gamma := A + \gamma B^T W^{-1} B$, an ideal preconditioner for (3.5), when used with GMRES, (see [5, 4]) is given by

$$P = \begin{pmatrix} A_\gamma & B^T \\ 0 & -\gamma^{-1} W \end{pmatrix}. \tag{3.6}$$

For large systems, it is not practical to use such a preconditioner, as it requires exact solves using A_γ. Thus, some approximation needs to be made, and for this we employ the modified

AL idea of Benzi and Olshanskii [4]. Assuming for simplicity $d = 2$ ($d = 3$ follows analogously), $A = (A_{11}, A_{12}; A_{21}, A_{22})$, and writing $B = (B_1, B_2)$, we have

$$A_\gamma = \begin{pmatrix} A_{11} + \gamma B_1^T W^{-1} B_1 & A_{12} + \gamma B_1^T W^{-1} B_2 \\ A_{21} + \gamma B_2^T W^{-1} B_1 & A_{22} + \gamma B_2^T W^{-1} B_2 \end{pmatrix} := \begin{pmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \\ \widetilde{A}_{21} & \widetilde{A}_{22} \end{pmatrix}. \tag{3.7}$$

Defining

$$\widetilde{A}_\gamma := \begin{pmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \\ 0 & \widetilde{A}_{22} \end{pmatrix},$$

the *modified-AL preconditioner* is defined by

$$\widetilde{P} = \begin{pmatrix} \widetilde{A}_\gamma & B^T \\ 0 & \widetilde{S} \end{pmatrix}. \tag{3.8}$$

where $\widetilde{S}$ is typically chosen to be some approximation to M_p. However, in our statically condensed systems, the pressure space is effectively the same as for P_0 elements elements (in 2D or 3D), leaving M_p diagonal, and thus $\widetilde{S} = -\gamma^{-1} M_p$ can be used.

One reason this preconditioner appears well suited for the reduced SV systems is because the pressure space of the reduced system is equivalent to using P_0 elements on the pre-refined mesh τ_H, and thus the pressure mass matrix M_p will be diagonal, and B and B^T matrices will be much sparser than if a higher order pressure basis was used (e.g. as in Taylor-Hood elements, where pressure basis functions have support on adjacent elements). Consequently, the diagonal blocks of $\widetilde{A}_\gamma$ are sparser compared to typical element choices.

3.3 $\mathcal{H}$-LU preconditioned GMRES

$\mathcal{H}$-matrices were introduced in [17] and since then have entered into a wide range of applications. The basic $\mathcal{H}$-matrix construction and corresponding arithmetic have reached a relatively mature state and are documented in the comprehensive lecture notes [6] and books [3, 18].

$\mathcal{H}$-matrices are based on a hierarchical subdivision of the matrix into subblocks and low-rank approximations of matrix data within (most of) these subblocks. Originally, $\mathcal{H}$-matrices were introduced in the context of fully populated matrices arising from solution operators of elliptic differential equations and in boundary element methods. In the finite element context, the stiffness matrix itself does not require an efficient approximation by an $\mathcal{H}$-matrix since it is sparse. Its LU-factors, however, suffer from inacceptable fill-in if computed exactly and can be efficiently computed or approximated by $\mathcal{H}$-LU factors [16, 2]. The $\mathcal{H}$-matrix construction and arithmetic as originally developed for fully populated matrices have a straightforward generalization to sparse matrices. However, there are two modifications for $\mathcal{H}$-matrices which have been designed for sparse matrices in particular and have been used here.

In this section, we will specify the particular variant of $\mathcal{H}$-matrix construction which is proposed for the $\mathcal{H}$-LU factorization of the reduced matrix to be used as a preconditioner in the GMRES method, and point to the literature for further details on $\mathcal{H}$-matrices.

The first modification concerns the construction of the block structure of the $\mathcal{H}$-matrix. In the classical $\mathcal{H}$-matrix, the block structure is generated through a repeated bisection of the respective index sets, i.e., row and column index sets are divided into two subsets, respectively, which leads

to four matrix subblocks. In the case of sparse matrices, the bisection has been replaced by a nested dissection approach in which row and column index sets are divided into three subsets each; two subsets S_1, S_2 of indices that are pairwise disconnected in the sense that $a_{ij} = 0 = a_{ji}$ if $i \in S_1$, $j \in S_2$ where $A = (a_{ij})$ denotes the stiffness matrix, and a third subset S_3 containing the remaining indices of the interior boundary. Such a subdivision results in a 3×3 matrix structure with zero blocks in the 1×2 and 2×1 positions. These zero blocks remain zero in a subsequent LU factorization which results in considerably faster ($\mathcal{H}$-)LU factorizations compared to bisection-based $\mathcal{H}$-matrices [19, 15]. The blocks are subdivided recursively until a minimum blocksize $nmin$ is reached, i.e., a block is further subdivided if the minimum of its number of rows and columns is greater than $nmin$. In the subsequent numerical results in section 4, we set $nmin = 128$.

The second modification concerns the development of a "blackbox" clustering algorithm. The classical construction of $\mathcal{H}$-matrices requires geometric information associated with the underlying indices in order to determine a suitable block structure. For sparse matrices, the information contained in the associated matrix graph can replace the need for geometric information [14].

Whereas the classical $\mathcal{H}$-matrix uses a fixed rank for the low rank approximations within matrix subblocks, it is possible to replace it by *adaptive ranks* in order to enforce a desired accuracy within the individual blocks. In particular, given a matrix block C and a desired $\mathcal{H}$-accuracy $0 < \delta_{\mathcal{H}} < 1$, we set the rank k_C of the approximation to C as

$$k_C := \min\{k' \mid \sigma_{k'} \leq \delta_{\mathcal{H}} \sigma_1\} \tag{3.9}$$

where σ_i denotes the i'th largest singular value of C. A decrease in $\delta_{\mathcal{H}}$ leads to an increase in the accuracy of the $\mathcal{H}$-LU factorization, i.e., an improved preconditioner, at the expense of increased setup times and storage requirements.

These modifications from the classical $\mathcal{H}$-matrix setting have led to highly efficient $\mathcal{H}$-LU preconditioners for a wide range of sparse matrices [13].

4 Numerical experiments

We now test the preconditioned iterative methods and static condensation described above on some benchmark NS test problems. We solve the nonlinear system (1.3)-(1.4) using Newton's method, and test the methods on the resulting linear systems. For all tests, we take $u_0 = 0$ + boundary conditions, as the initial guess. The stopping criteria for Newton is a relative update size less than 10^{-8} in the natural solution norm

$$\|(u, p)\| := \sqrt{\|\nabla u\|^2 + \|p\|^2}.$$

All computations, except those using $\mathcal{H}$-LU preconditioning (due to licensing issues), were run with Matlab 2011a, on a 2x 2.66 GHz Quad-Core Intel Xeon processor, using 32GB 1066 MHz DDR3 memory. The $\mathcal{H}$-LU preconditioned GMRES were run on a Dell Latitude E6500 notebook (2.80GHz, 8GB). GMRES was set to restart after every 50 iterations for all tests. For the modified-AL preconditioned GMRES solver, we chose ILU to approximately solve the subproblems. Although this is effective, there are many possible variations (and thus improvements) that can be made, such as algebraic multigrid method as solvers for diagonal blocks, inner-outer iterations and so forth. We plan to consider other strategies in future work, but for now we choose ILU for its simplicity and to set a starting point from which improvements can be made. The drop tolerance for ILU

was optimized for each problem, as was the AL parameter γ. The optimal parameters turned out to be essentially mesh independent, but were determined essentially by brute force. We note the mesh independence of optimal γ was also reported in [5].

4.1 3D problem with known analytical solution

Our first experiment is a comparison of SV and Taylor-Hood (TH) approximations to a known analytical solution, for the purpose of comparing accuracy, including accuracy in mass conservation. The purpose of this simple test problem is to compare velocity and divergence errors for these very similar elements. We leave solver comparisons for experiments that follow.

We compute on a uniform barycenter refined tetrahedral mesh, and use $((P_3)^3, P_2^{disc})$ SV elements and $((P_3)^3, P_2)$ TH elements. We note these elements have the same velocity space, and differ in their pressure spaces in that SV has discontinuous pressures while TH uses a globally continuous pressure approximation. Both of these element choices are known to be inf-sup stable and admit optimal approximation properties in this setting. A major advantage of using SV is that it gives pointwise mass conservation, while mass conservation in TH is not exact, although it is bounded by the H^1 velocity error (which is optimal for TH elements). Hence we expect

$$\begin{aligned} \|\nabla \cdot u_h^{SV}\| &= O(\epsilon_{mach}) \\ \|\nabla \cdot u_h^{TH}\| &= O(h^3), \end{aligned}$$

however in practical computations there is a minimum h one can use, and so here we will quantify the mass conservation errors for both element choices.

This test problem uses $\Omega = (0,1)^3$, the steady analytical solution

$$u = (\cos(2\pi z), \sin(2\pi z), \sin(2\pi(x+y)))^T, \quad p = \sin(2\pi(x+y+z))$$

with $\nu = 1$, and f calculated from this and the steady NS equations. We use this f, and Dirichlet boundary conditions taken to be the interpolant of the solution at the boundary nodes, and initial condition $u_0 = 0$ but equipped with the boundary condition, and solved (1.3)-(1.4).

The results of the computations are given in Table 5 as the H^1 velocity errors and L^2 divergence errors, and we observe that TH has slightly better H^1 error on each mesh, but the divergence errors for TH are large while the divergence errors for SV are very small even on the coarsest mesh.

4.2 3D driven cavity

We now investigate the proposed methods on the 3D driven cavity problem for Re=100 (the same test problem as is used in Section 2). We compute using (1.3)-(1.4) with both $((P_3)^3, P_2^{disc})$ SV elements and $((P_3)^3, P_2)$ TH elements, using barycenter refinements of uniform tetrahedral meshes. We compare the iterative solvers with SV on the condensed and non-condensed systems, and TH on the non-condensed system (condensing TH systems is not worth the effort, as there is not much reduction, and the resulting condensed systems are much more difficult to solve). Also, we note that the $\mathcal{H}$-LU preconditioned GMRES converged only on the condensed SV systems.

For modified AL preconditioned GMRES, optimal parameters for condensed SV were $\gamma = 0.005$ and an ILU drop tolerance of 10^{-3}, while for the non-condensed SV system we used $\gamma = 0.05$ and an ILU drop tolerance of 10^{-2}. Using a smaller tolerance for the non-condensed systems became very inefficient in setup times. For AL preconditioned GMRES we used 10^{-6} relative residual stopping

Mesh level	Element	$dim(X_h)$	$dim(Q_h)$	$\|u-u_h\|_1$	$\|\nabla \cdot u_h\|$
1	TH	3,189	365	6.48E-1	1.68E-1
1	SV	3,189	1,920	7.12E-1	6.24E-10
2	TH	23,871	2,649	1.93E-2	7.11E-2
2	SV	23,871	15,360	7.82E-2	3.59E-10
3	TH	78,897	8,677	5.78E-2	2.07E-2
3	SV	78,897	51,840	2.28E-2	1.63E-10
4	TH	185,115	20,273	8.67E-3	2.46E-3
4	SV	185,115	122,280	9.56E-3	1.78E-10
5	TH	359,373	39,261	8.02E-3	2.33E-3
5	SV	359,373	240,000	8.85E-3	3.06E-10
6	TH	618,519	67,465	2.55E-3	7.33E-4
6	SV	618,519	414,720	2.81E-3	9.14E-12

Table 5: Shown above are velocity and divergence errors for SV and TH for test problem 1.

criteria, and for $\mathcal{H}$-LU preconditioned GMRES, we used a relative residual stopping criteria of 10^{-12}, and use a relative accuracy of δ_H =1E-3 (see (3.9)). The reason for the small stopping criteria is that with $\mathcal{H}$-LU, it typically takes just a few extra iterations to reduce the residual past 10^{-6} to 10^{-12}, and so we take advantage of that here. We also test TH with modified AL preconditioned GMRES, and found optimal $\gamma = 0.03$ and ILU drop tolerance of 10^{-2}. The timings, iteration counts, and divergence errors of the methods are shown in Table 6 and Figure 5. The mesh levels used in the table are the same as the previous experiment.

It is immediately clear from Table 6 and Figure 5 that TH with the modified AL preconditioner is not competitive on this problem, for two reasons. First, the setup times are much worse than the other methods. This is not unexpected because two of the other methods are on condensed systems, and the sparsity of the matrices associated with the pressure space are much denser than in the SV case. A second serious problem with TH (independent of the preconditioner) is that the mass conservation cannot shrink as the mesh is refined. TH mass conservation is controlled by the H^1 velocity error, which does not tend to 0 for this problem due to the (nonphysical) discontinuous velocity boundary condition. Plots of the Mesh level 5 TH solution's divergence contours are shown in Figure 6 at the midsliceplanes, and we observe the large divergence error are near intersection of the lid and walls, where the discontinuity in the boundary occurs.

For the methods used with SV, modified AL preconditioned GMRES appears to be the best. It has the best setup and iteration times, and the scaling of iterations with mesh refinement appears to be growing slowly. The advantage of using static condensation is clear, as in no way is the non-condensed system competitive.

The results of $\mathcal{H}$-LU preconditioner are interesting. We note again that due to licensing issues, the $\mathcal{H}$-LU timings were not run on the same machine or with the same software as the rest of

Mesh level	Total dof	Element, full/red	Method	setup(s.)	iter.(s.)	iterations	$\|\nabla \cdot u_h\|$
1	1,077	SV, red	Mod-AL GMRES	0.01	0.20	69.3	1.45E-13
1	5,109	SV, full	Mod-AL GMRES	0.15	0.34	39.8	8.24E-13
1	3,554	TH, full	Mod-AL GMRES	0.26	0.18	26.3	2.93E-1
1	1,077	SV, red	HLU GMRES	0.07	0.01	12.0	4.33E-12
2	6,975	SV, red	Mod-AL GMRES	0.29	1.29	113.3	1.83E-10
2	39,231	SV, full	Mod-AL GMRES	1.27	2.94	47.0	1.87E-10
2	26,520	TH, full	Mod-AL GMRES	11.46	2.26	47.8	3.38E-1
2	6,975	SV, red	HLU GMRES	7.28	0.23	22.0	1.81E-11
3	21,873	SV, red	Mod-AL GMRES	3.25	5.93	131.5	1.76E-11
3	130,787	SV, full	Mod-AL GMRES	7.81	13.67	70.8	7.79E-11
3	87,574	TH, full	Mod-AL GMRES	93.14	11.89	82.75	3.35E-1
3	21,873	SV, red	HLU GMRES	89.60	1.82	31.0	2.76E-11
4	49,947	SV, red	Mod-AL GMRES	15.01	13.27	144.3	1.65E-11
4	307,995	SV, full	Mod-AL GMRES	30.39	47.05	114.8	7.52E-11
4	205,388	TH, full	Mod-AL GMRES	336.26	42.64	133.00	3.35E-1
4	49,947	SV, red	HLU GMRES	283.85	7.15	44.0	3.47E-11
5	95,373	SV, red	Mod-AL GMRES	50.76	29.45	167.0	1.51E-11
5	599,373	SV, full	Mod-AL GMRES	91.73	109.39	131.0	2.10E-10
5	398,634	TH, full	Mod-AL GMRES	1,425.13	92.02	142.50	3.37E-1
5	95,373	SV, red	HLU GMRES	738.87	70.67	194.0	4.33E-11
6	162,327	SV, red	Mod-AL GMRES	164.42	51.52	163.5	1.27E-11
6	1,033,239	SV, full	Mod-AL GMRES	197.80	606.54	220.8	1.13E-10
6	685,984	TH, full	Mod-AL GMRES	2,856.58	428.02	242.50	3.37E-1
6	162,377	SV, red	HLU GMRES	1,891.21	71.73	90.0	5.56E-11

Table 6: Solve times and mass conservation of the methods for the 3D Re=100 driven cavity problem.

the tests. Still, we can safely conclude that the setup time is not good compared to the modified AL preconditioner. However, the iteration time compares well with modified AL preconditioned GMRES, and the number of iterations it needs to converge is generally smaller. Hence for time dependent problems on large distributed systems, this method may be attractive for problems where the preconditioner can be reused many times.

4.3 Test problem: 2D driven cavity

Our final test is the benchmark problem of 2D driven cavity flow, which is the same test problem as is done for the direct solver in Section 2, but now we test the iterative solvers with it. We consider here two Reynolds numbers, Re=100 and Re=1,000, using (P_2, P_1^{disc}) SV elements on barycenter refinements of uniform triangulations. We test the same methods as the previous experiment, but due the poor performance of TH elements in previous experiments, we test only using SV here.

For AL-preconditioned GMRES, we used 10^{-6} relative residual stopping criteria, and ILU for the inner solves with drop tolerance of 10^{-4} in all cases (smaller drop tolerances of 10^{-5} and 10^{-6} made little difference in number of iterations needed). Optimal γ for each case was determined by

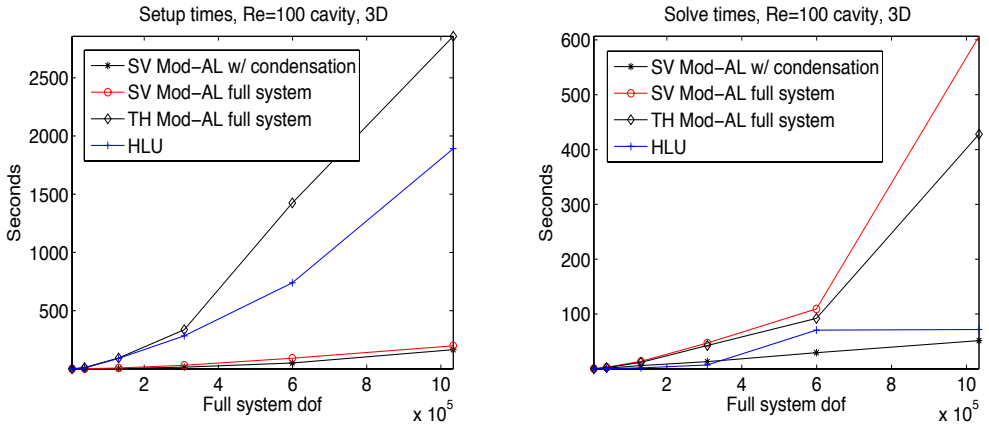

Figure 5: Shown above are plots of setup and solve times vs dof, for the Re=100 2D test.

brute force to an accuracy of 0.001, and the optimal parameters were found to be mesh independent. These γ are given with the results in Tables 7 and 8. The $\mathcal{H}$-LU preconditioned GMRES solver used the relative accuracy δ_H =1E-4 (see (3.9)), and a stopping criteria of 10^{-12} for the relative residual was used. As in the 3D case, the reason for such a small stopping criteria is that lowering it does not cost much to do so; roughly speaking, most of the work is done in the setup. As in the 3D case, $\mathcal{H}$-LU was effective only on the reduced systems. Solutions plots using the different solving methods were indistinguishable (and so are omitted), and on the finer meshes, agree very well with the solutions to this problem found in Section 2 with the direct solver and in the literature [20].

Results for this test problem are given in Table 7 and Figure 7 for Re=100, and in Table 8 and Figure 8 for Re=1,000. All methods' solutions have excellent mass conservation. As in the 3D case, static condensation significantly reduces solve times with the modified AL preconditioned GMRES

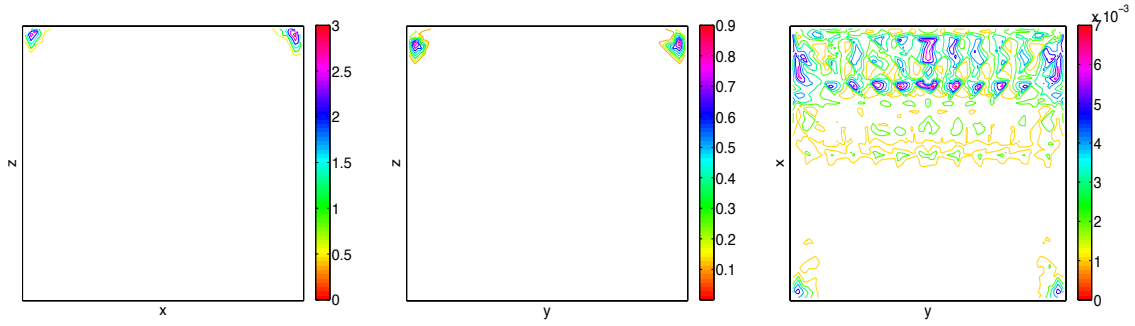

Figure 6: Contour plots at the mid-sliceplanes of the absolute divergence errors in the Re=100 3D driven cavity simulations with TH elements.

method, and the $\mathcal{H}$-LU preconditioned GMRES only converges with the condensed system. What is most interesting in these results are the timings for $\mathcal{H}$-LU: setup time is poor, but in most cases the solve time is competitive or better than the other methods. Hence $\mathcal{H}$-LU preconditioned GMRES may be a good solver candidate for time dependent NSE type problems where the preconditioner setup is reused many times.

H	full/red	Total dof	Method	γ	setup(s.)	iter.(s.)	avg iterations	$\|\nabla \cdot u_h\|$
1/16	red	3,100	Mod-AL GMRES	0.005	0.02	0.18	36.4	2.94E-14
1/16	full	12,604	Mod-AL GMRES	0.05	0.11	0.21	21	6.02E-14
1/16	red	3,100	HLU GMRES	-	0.26	0.02	6	3.99E-11
1/32	red	12,388	Mod-AL GMRES	0.005	0.16	0.45	37.8	2.54E-14
1/32	full	51,204	Mod-AL GMRES	0.05	0.84	0.81	21.8	4.47E-14
1/32	red	12,388	HLU GMRES	-	1.75	0.15	9	8.61E-11
1/64	red	49,094	Mod-AL GMRES	0.005	0.99	1.73	39.4	2.34E-14
1/64	full	204,550	Mod-AL GMRES	0.05	5.11	4.26	28.8	3.03E-14
1/64	red	49,094	HLU GMRES	-	9.03	1.07	14	4.27E-14
1/128	red	164,866	Mod-AL GMRES	0.005	5.80	7.89	45.2	4.24E-14
1/128	full	689,154	Mod-AL GMRES	0.05	27.93	91.56	68.2	7.09E-14
1/128	red	164,866	HLU GMRES	-	38.27	12.78	49	3.37E-10

Table 7: Solve times and divergence error of several methods for the 2D Re=100 driven cavity problem using SV elements.

H	full/red	Total dof	Method	γ	setup(s.)	iter.(s.)	avg iterations	$\|\nabla \cdot u_h\|$
1/16	red	3,100	Mod-AL GMRES	0.003	0.02	0.51	96.3	4.50E-14
1/16	full	12,604	Mod-AL GMRES	0.01	0.11	0.80	65.8	3.65E-14
1/16	red	3,100	HLU GMRES	-	0.27	0.36	105	4.67E-11
1/32	red	12,388	Mod-AL GMRES	0.003	0.17	1.39	107.7	6.67E-13
1/32	full	51,204	Mod-AL GMRES	0.01	0.74	3.11	63.0	1.06E-14
1/32	red	12,388	HLU GMRES	-	1.88	1.55	96	1.09E-10
1/64	red	49,094	Mod-AL GMRES	0.003	1.14	5.51	116.5	4.14E-14
1/64	full	204,550	Mod-AL GMRES	0.01	4.75	12.43	78.2	3.62E-12
1/64	red	49,094	HLU GMRES	-	9.63	0.47	6	1.97E-10
1/128	red	164,866	Mod-AL GMRES	0.003	6.22	23.04	134.1	4.44E-14
1/128	full	689,154	Mod-AL GMRES	0.01	24.67	164.07	114.0	5.44E-14
1/128	red	165,866	HLU GMRES	-	39.83	1.92	7	3.84E-10

Table 8: Solve times and divergence error of several methods for the 2D Re=1,000 driven cavity problem using SV elements.

The setup times, iteration times, and average number of iterations needed for convergence are given in Table 7 for Re=100 and in Table 8 for Re=1,000, for all the methods. On these 2D test problems, the direct solvers appear the most attractive. Applying backslash to both the condensed system resulting from (1.3)-(1.4), and to the non-condensed penalty method system, was very

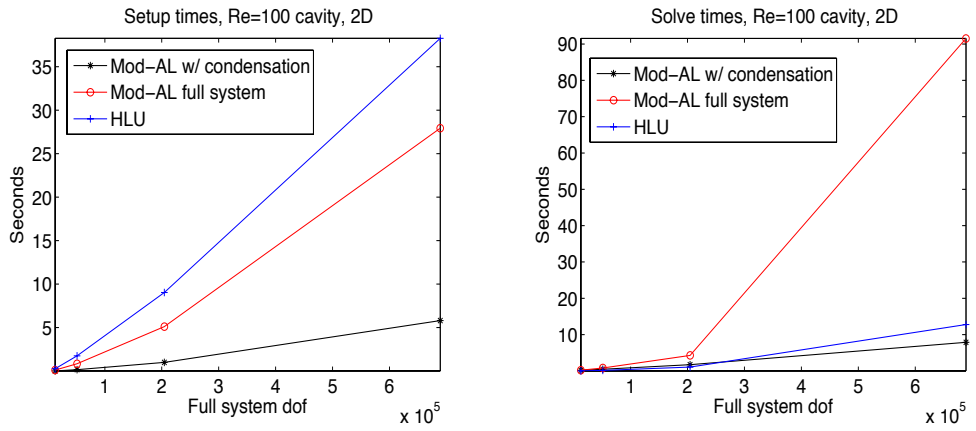

Figure 7: Shown above are plots of setup and solve times vs dof, for the Re=100 2D test.

efficient. These methods are robust with respect to Reynolds number, require no setup time, and are simple to use as there are no parameters to optimize. But we also observed that direct solvers do not seem to benefit from static condensation. This was found using backslash from MATLAB, and was confirmed by numerical tests with the direct solver PARDISO.

For the iterative methods, modified AL preconditioned GMRES shows it is an effective method, but as expected its effectiveness deteriorates as Re increases. $\mathcal{H}$-LU preconditioned GMRES displays interesting behavior in that it works much better on the higher Reynolds number problem. For Re=100, this method does not perform well, but for Re=1,000, this method has the fastest iteration time of all methods. The drawback of this method is clearly the setup time, but on problems where a setup can be reused, this method could be very competitive, particularly since it can solve to such a high degree of of accuracy and thus produce solutions with excellent mass conservation.

5 Conclusions

We have found efficient solvers for use with Scott-Vogelius finite element discretizations of the NSE. Both modified AL preconditioned GMRES applied to the statically condensed system, and the penalty method applied to the full system, are effective solvers for steady NS problems. $\mathcal{H}$-LU preconditioned GMRES was less efficient on these steady problems, but showed promise in that it was competitive in iterations needed and iteration time if the setup can be reused (e.g. in a time dependent problem).

6 Acknowledgements

The authors would like to thank Prof. Vince Ervin for helpful discussions that improved this work.

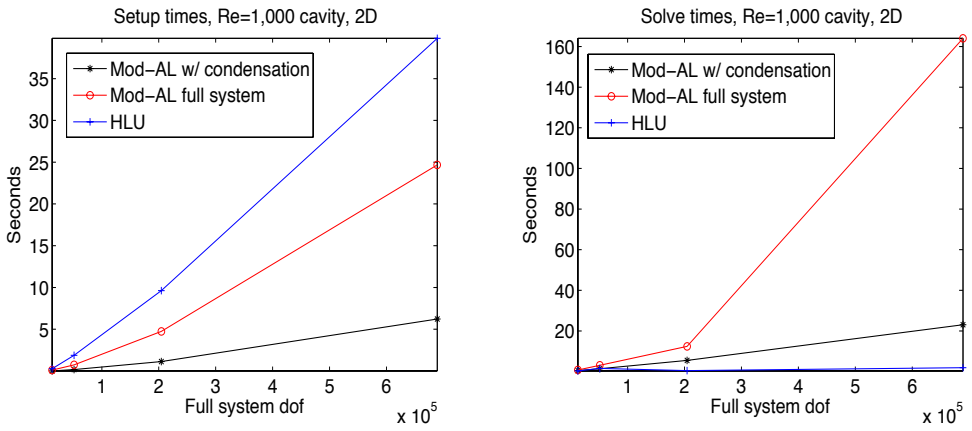

Figure 8: Shown above are plots of setup and solve times vs dof, for the Re=1,000 2D test.

References

[1] D. N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21:337–344, 1984.

[2] M. Bebendorf. Why finite element discretizations can be factored by triangular hierarchical matrices. *SIAM Journal on Numerical Analysis*, 45:1472–1494, 2007.

[3] M. Bebendorf. *Hierarchical Matrices. A Means to Efficiently Solve Elliptic Boundary Value Problems*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer, 2008.

[4] M. Benzi and M. Olshanksii. An augmented Lagrangian-based approach to the Oseen problem. *SIAM Journal of Scientific Computing*, 28(6):2095–2113, 2005.

[5] M. Benzi, M.A. Olshanskii, and Z. Wang. Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 66(4):486–508, 2011.

[6] S. Börm, L. Grasedyck, and W. Hackbusch. Hierarchical matrices, 2003. Lecture Notes No. 21, Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany, available online at www.mis.mpg.de/preprints/ln/, revised version June 2006.

[7] S. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, 1994.

[8] E. Burman and A. Linke. Stabilized finite element schemes for incompressible flow using Scott-Vogelius elements. *Applied Numerical Mathematics*, 58(11):1704–1719, 2008.

[9] M. Case, V. Ervin, A. Linke, and L. Rebholz. Improving mass conservation in FE approximations of the Navier Stokes equations using C^0 velocity fields: A connection between grad-div stabilization and Scott-Vogelius elements. *SIAM Journal on Numerical Analysis*, 49(4):1461–1481, 2011.

[10] H.C. Elman, D.J. Silvester, and A.J. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numerische Mathematik*, 90:665–688, 2002.

[11] K. J. Galvin, A. Linke, L. G. Rebholz, and N. E. Wilson. Stabilizing poor mass conservation in incompressible flow problems with large irrotational forcing and application to thermal convection. *Comput. Methods Appl. Mech. Engrg.*, accepted, 2012.

[12] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes equations : Theory and Algorithms*. Springer-Verlag, 1986.

[13] L. Grasedyck, W. Hackbusch, and R. Kriemann. Performance of $\mathcal{H}$-LU preconditioning for sparse matrices. *Comput. Meth. Appl. Math.*, 8:336–349, 2008.

[14] L. Grasedyck, R. Kriemann, and S. Le Borne. Parallel black box $\mathcal{H}$-LU preconditioning for elliptic boundary value problems. *Comput. Visual. Sci.*, 4-6:273–291, 2008.

[15] L. Grasedyck, R. Kriemann, and S. Le Borne. Domain decomposition based $\mathcal{H}$-LU preconditioning. *Numerische Mathematik*, 112:565–600, 2009.

[16] L. Grasedyck and S. Le Borne. $\mathcal{H}$-matrix preconditioners in convection-dominated problems. *SIAM J. Mat. Anal.*, 27:1172–1183, 2006.

[17] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[18] W. Hackbusch. *Hierarchische Matrizen*. Springer, 2009. (in German).

[19] I. Ibragimov, S. Rjasanow, and K. Straube. Hierarchical Cholesky decomposition of sparse matrices arising from curl-curl-equations. *J. Numer. Math.*, 15:31–58, 2007.

[20] S. Kaya and B. Riviere. A two-grid stabilization method for solving the steady-State Navier-Stokes equations. *Numer. Meth. Part. Diff. Equations*, 3(3):728–743, 2006.

[21] W. Layton. *An Introduction to the Numerical Analysis of Viscous Incompressible Flows*. SIAM, 2008.

[22] A. Linke. *Divergence-free mixed finite elements for the incompressible Navier-Stokes Equation*. PhD thesis, University of Erlangen, 2007.

[23] A. Linke. Collision in a cross-shaped domain — A steady 2d Navier-Stokes example demonstrating the importance of mass conservation in CFD. *Comp. Meth. Appl. Mech. Eng.*, 198(41–44):3278–3286, 2009.

[24] A. Linke, G. Matthies, and L. Tobiska. Non-nested multi-grid solvers for mixed divergence free Scott-Vogelius discretizations. *Computing*, 83(2-3):87–107, 2008.

[25] A. Linke, L. Rebholz, and N. Wilson. On the convergence rate of grad-div stabilized Taylor-Hood to Scott-Vogelius solutions for incompressible flow problems. *Journal of Mathematical Analysis and Applications*, 381:612–626, 2011.

[26] C. Manica, M. Neda, M.A. Olshanskii, L. Rebholz, and N. Wilson. On an efficient finite element method for navier-stokes-omega with strong mass conservation. *Computational Methods in Applied Mathematics*, 11(1):3–22, 2011.

[27] M.A. Olshanskii and L. Rebholz. Application of barycenter refined meshes in linear elasticity and incompressible fluid dynamics. *ETNA*, 38:258–274, 2011.

[28] J. Qin. *On the convergence of some low order mixed finite elements for incompressible fluids.* PhD thesis, Pennsylvania State University, 1994.

[29] O. Schenk and K. Gärtner. Solving unsymmetric sparse systems of linear equations with pardiso. *Journal of Future Generation Computer Systems*, 20(3):475–487, 2004.

[30] J. Schöberl. Multigrid methods for a parameter dependent problem in primal variables. *Numerische Mathematik*, 84:97–119, 1999.

[31] L.R. Scott and M. Vogelius. Conforming finite element methods for incompressible and nearly incompressible continua. In *Large-scale computations in fluid mechanics, Part 2*, volume 22-2 of *Lectures in Applied Mathematics*, pages 221–244. Amer. Math. Soc., 1985.

[32] L.R. Scott and M. Vogelius. Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials. *Mathematical Modelling and Numerical Analysis*, 19(1):111–143, 1985.

[33] R. Temam. Sur lapproximation des solutions des equations de Navier-Stokes. *C.R. Acad. Sci. Paris, Series A*, 262:219–221, 1966.

[34] R. Temam. *Navier-Stokes Equations : Theory and Numerical Analysis.* Elsevier North-Holland, 1979.

[35] M. Vogelius. An analysis of the p-version of the finite element method for nearly incompressible materials. Uniformly valid, optimal error estimates. *Numer. Math.*, 41:39–53, 1983.

[36] M. Vogelius. A right-inverse for the divergence operator in spaces of piecewise polynomials. Application to the p-version of the finite element method. *Numer. Math.*, 41:19–37, 1983.

[37] K.L. Wong and A.J. Baker. A 3d incompressible Navier-Stokes velocity-vorticity weak form finite element algorithm. *International Journal for Numerical Methods in Fluids*, 38:99–123, 2002.

[38] S. Zhang. A new family of stable mixed finite elements for the 3d Stokes equations. *Math. Comp.*, 74(250):543–554, 2005.

[39] S. Zhang. On the P1 Powell–Sabin divergence-free finite element for the Stokes equations. *Journal of Computational Mathematics*, 26(3):456–470, 2008.

[40] S. Zhang. Quadratic divergence-free finite elements on Powell-Sabin tetrahedral grids. *Calcolo*, 48(3), 2011.