Improving Conservation for First-Order System Least-Squares Finite-Element Methods

J.H. Adler and P.S. Vassilevski

Abstract The first-order system least-squares (FOSLS) finite element method for solving partial differential equations has many advantages, including the construction of symmetric positive definite algebraic linear systems that can be solved efficiently with multilevel iterative solvers. However, one drawback of the method is the potential lack of conservation of certain properties. One such property is conservation of mass. This paper describes a strategy for achieving mass conservation for a FOSLS system by changing the minimization process to that of a constrained minimization problem. If the space of corresponding Lagrange multipliers contains the piecewise constants, then local mass conservation is achieved similarly to the standard mixed finite-element method. To make the strategy more robust and not add too much computational overhead to solving the resulting saddle-point system, an overlapping Schwarz process is used.

Keywords Conservation • First-order system least-squares • Finite elements • Domain decomposition • Two-level

Mathematics Subject Classification (2010): 65F10, 65N20, 65N30

The work of the author "P.S. Vassilevski" was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

J.H. Adler
Department of Mathematics, Tufts University, Medford, MA 02155, USA
e-mail: james.adler@tufts.edu

P.S. Vassilevski (✉)
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory,
P.O. Box 808, L-560, Livermore, CA 94551, USA
e-mail: panayot@llnl.gov

O.P. Iliev et al. (eds.), *Numerical Solution of Partial Differential Equations: Theory, Algorithms, and Their Applications*, Springer Proceedings in Mathematics & Statistics 45,
DOI 10.1007/978-1-4614-7172-1_1, © Springer Science+Business Media New York 2013

1 Introduction

The first-order system least-squares (FOSLS) approach is a finite-element discretization, which solves a system of linear partial differential equations (PDEs) by minimizing the L^2 norm of the residual of the PDE [14–16, 30, 31]. Least-squares finite-element methods, in general, have several nice properties and have been used on a wide variety of problems, e.g., [4, 6, 7, 9, 13, 29, 34]. One advantage is that they yield symmetric positive definite (SPD) algebraic systems, which are amenable to multilevel techniques. This is true for any PDE system, including systems like Stokes where a mixed finite-element method would yield a saddle-point problem and an indefinite linear system [10]. Another advantage is that they yield sharp and reliable a posteriori estimates [3]. This is useful for implementing adaptive local refinement techniques, which allow the approximations to be resolved more accurately in regions of higher error [11, 19]. A disadvantage of the least-squares methods noted in the literature is a loss of conservation for certain properties in a given system. For instance, the Stokes' or Navier–Stokes' system contains an equation for the conservation of momentum and one for the conservation of mass [20, 21]. Since the least-squares principle minimizes both equations equally, both quantities are only conserved up to the error tolerance given for the simulation. Attempts to improve the conservation of mass would result in a loss of accuracy in the conservation of momentum. Despite this, in several applications, conservation of a certain quantity is considered essential to capturing the true physics of the system. For instance, in electromagnetic problems, such as magnetohydrodynamics (the treatment of plasmas as charged fluids), loss of accuracy in the solenoidal constraint of the magnetic field, $\nabla \cdot \mathbf{B} = 0$, can lead to instabilities in the system [2, 8].

In this paper, we consider methods for improving the conservation of a divergence constraint, such as mass conservation, in a system, using the FOSLS finite-element method. There are many ways to improve the accuracy of mass conservation in such systems, including adaptive refinement to increase the spatial resolution of the discretization [6, 7], higher temporal accuracies or higher-order elements for time-dependent problems [32], using divergence-free finite-element spaces [1, 4, 17, 18], reformulating the first-order system into a more conservative one [23], as well as using a compatible least-squares method [5], which use ideas from mixed Galerkin methods to improve the mass conservation. In addition, an alternative approach called FOSLL* [27, 28] has been developed, in which an adjoint system is considered, and the error is minimized in the L^2 norm directly. This has been shown to improve conservation in satisfying the divergence constraint in incompressible fluid flow and electromagnetic problems. In this paper, we discuss an approach that simply corrects the solution approximated by the FOSLS discretization so that it conserves the given quantity. The goal is to keep the discretization as is, preserving all of the special properties of the least-squares minimization while still obtaining the appropriate conservation. As a result, the a posteriori error estimates and the simple finite-element spaces can still be used. More specifically, the aim of this paper is to show that it is possible to conserve a certain quantity in the least-squares finite-element setting by using a local subdomain correction post-processing scheme at relatively little extra cost.

The paper is outlined as follows. In Sect. 2, we consider the FOSLS discretization applied to a Poisson problem and show how the scheme can result in a type of "mass loss." Section 3 investigates a way of transforming the minimization principle into a constrained minimization problem and investigates what types of constraints are possible. Next, in Sect. 4, a local subdomain and coarse-grid correction solver is used to make the method more robust. This uses an overlapping Schwarz (Vanka-like) smoother with a coarse-grid correction to solve the constrained problem [35–37]. Finally, concluding remarks and a discussion of future work is given in Sect. 5.

2 First-Order System Least-Squares

To illustrate the FOSLS finite-element method, consider a PDE system that is first put into a differential first-order system of equations, denoted by $Lu = f$. Here, L is a mapping from an appropriate Hilbert space, $\mathscr{V}$, to an L^2 product space. In many contexts, $\mathscr{V}$ is chosen to be an H^1 product space with appropriate boundary conditions.

This minimization is written as

$$u_* = \arg\min_{u\in\mathscr{V}} G(u;f) := \arg\min_{u\in\mathscr{V}} ||Lu - f||_0^2, \tag{1}$$

where u_* is the solution in an appropriate H^1 space. The minimization results in the weak form of the problem:

Find $u_* \in \mathscr{V}$ such that

$$\langle Lu_*, Lv\rangle = \langle f, Lv\rangle \quad \forall v \in \mathscr{V}, \tag{2}$$

where $\langle\cdot,\cdot\rangle$ is the usual L^2 inner product on the product space, $(L^2)^k$, for k equations in the linear system. If the following properties of the bilinear form $\langle Lu, Lv\rangle$ are assumed,

$\exists$ constants, c_1 and c_2, such that

$$\begin{array}{llcll} \text{continuity} & \langle Lu, Lv\rangle & \leq & c_2||u||_{\mathscr{V}}||v||_{\mathscr{V}} & \forall \quad u,v \in \mathscr{V}, \qquad (3)\\ \text{coercivity} & \langle Lu, Lu\rangle & \geq & c_1||u||_{\mathscr{V}}^2 & \forall \quad u \in \mathscr{V}, \qquad (4)\end{array}$$

then, by the Riesz representation theorem, this bilinear form is an inner product on $\mathscr{V}$ [26]. In addition, these properties imply the existence of a unique solution, $u_* \in \mathscr{V}$, for the weak problem (2). Here, c_1 and c_2 depend only on the operator, L, and the domain of the problem. They are independent of u and v.

Next, u_* is approximated by restricting (1) to a finite-dimensional space, $\mathscr{V}^h \subseteq \mathscr{V}$, which leads to (2) restricted to $\mathscr{V}^h$. Since $\mathscr{V}^h$ is a subspace of $\mathscr{V}$, the discrete problem is also well posed. Choosing an appropriate basis, $\mathscr{V}^h = span\{\Phi_j\}$, and restricting (2) to this basis yields an algebraic system of equations involving the matrix, A, with elements

$$(A)_{ij} = \langle L\Phi_j, L\Phi_i \rangle. \tag{5}$$

It has been shown that, in the context of a SPD H^1-equivalent bilinear form restricted to a finite-element subspace, a multilevel technique exists that yields optimal convergence to the linear system [15].

2.1 Sample Problem and Loss of Conservation

To illustrate possible losses in conservation, consider the convection–diffusion equation for unknown p in two dimensions,

$$-\nabla \cdot D\nabla p + \mathbf{r} \cdot \nabla p + cp = f, \tag{6}$$

with D an SPD matrix that could depend on the domain, $\mathbf{r}$ a vector, and c a nonnegative constant, respectively. In order to make the system first order, a new variable, $\mathbf{u} = D\nabla p$, is introduced. The resulting FOSLS system becomes

$$-\nabla \cdot \mathbf{u} + D^{-1}\mathbf{r} \cdot \mathbf{u} + cp = f, \tag{7}$$

$$\nabla \times D^{-1}\mathbf{u} = 0, \tag{8}$$

$$D^{-1/2}\mathbf{u} - D^{1/2}\nabla p = 0. \tag{9}$$

Here, a scaling on D is performed to allow the resulting discrete system to be better conditioned and, thus, more amenable to multigrid methods. Also, the extra curl equation is introduced so that the weak system is continuous and coercive and, therefore, H^1 equivalent [14, 15]. For simplicity, let $D = I$, $\mathbf{r} = \mathbf{0}$, and $c = 0$. Then, the following functional is minimized:

$$\mathscr{G} = ||\nabla \cdot \mathbf{u} + f||_0^2 + ||\nabla \times \mathbf{u}||_0^2 + ||\mathbf{u} - \nabla p||_0^2.$$

The resulting discrete system is

$$A\mathscr{U} = b,$$

where $\mathscr{U} = (\mathbf{u}, p)^T$. Here, A is the matrix as defined in (5), where L now refers to system (7)–(9). Similarly, the right-hand side vector, b, is defined as $b_i = \langle \mathbf{f}, L\Phi_i \rangle$, where $\mathbf{f} = (f, 0, 0)^T$. When minimizing this functional, equal weight is given to each term in the system. Therefore, if better accuracy is needed on a certain term, such as the divergence constraint, accuracy is lost in the other portions. In many applications, however, exact conservation of certain terms is important for developing an accurate model of a physical system. For instance, one may want to conserve the “mass” of the system. This is defined as

$$\int_{\Omega} -\nabla \cdot \mathbf{u} \, d\Omega = \int_{\Omega} f \, d\Omega. \tag{10}$$

In other words, the amount of flow in or out of the system is equal to the flow contributed by the source (this has more physical meaning in a system like Stokes, where we assume $\nabla \cdot \mathbf{u} = 0$ [20]). In fact, in many applications *local* mass conservation is desired instead, where the mass is conserved in all regions of the domain, including a single element. Mixed finite-element methods can satisfy this exactly and are commonly used in these situations. However, for the least-squares methods, since the part of the functional concerned with this property is only minimized to a certain degree (i.e., truncation error of the scheme at best), this cannot be satisfied exactly. Another issue concerns the fact that in many applications of the FOSLS finite-element method, the same order of polynomials is chosen as the basis for every unknown in the discrete space. For instance, linear functions are chosen to approximate both $\mathbf{u}$ and p. As a result, in trying to satisfy the term $\mathbf{u} - \nabla p = 0$, one is trying to match linears with the gradient of linears or constants. This is not approximated very well and accuracy is lost. As a result the conservation property is also lost. Choosing higher-order elements does remedy this to some extent, especially in two dimensions. However, using higher-order elements increases the complexity of the discrete system and the grid hierarchy in a multigrid scheme, making the systems harder to solve. In addition, the effect of higher-order elements is lessened when going to three dimensions [22, 24, 32].

To improve on this, here, the idea of adding the mass conservation as a constraint to the system is considered. Thus, instead of just minimizing the FOSLS functional, the functional is minimized subject to a constraint. This constraint enforces the desired mass conservation, while still allowing the FOSLS functional to be minimized as usual, thus retaining its nice properties. We mention that the modified method can achieve full local mass conservation, if the space of corresponding Lagrange multipliers contains the piecewise constants, similarly to the standard mixed finite-element method. Next, several approaches for implementing this constraint are described.

3 Constrained FOSLS

To enforce the constraint mentioned above, a Lagrange multiplier, λ, is introduced, and the FOSLS system is augmented as follows:

$$\begin{pmatrix} A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ g \end{pmatrix}. \tag{11}$$

Here, A and $\mathscr{U}$ are as before for the FOSLS discretization, λ is the Lagrange multiplier, and C is a finite-element assembly of the constraint; in this example, $-\nabla \cdot \mathbf{u} = f$. Two possible ways to construct C are considered. For the rest of the paper, we consider a triangulation of a mesh in two dimensions, $\mathscr{T}_h$, with grid spacing h. In addition, consider the polynomial spaces of order k defined on this triangulation as $\mathscr{P}_k$. The following notation is used for matrices and spaces:

Definition 1. Let $\boldsymbol{\Phi}_j \in \left[\mathscr{P}_{k_1}\right]^2$ be a vector and let $q_i \in \mathscr{P}_{k_2}$ be a scalar. Let f be some right-hand side function as defined in (6). Then, we define the following matrices:

$$(\tilde{B})_{ij} = \langle -\nabla \cdot \boldsymbol{\Phi}_j, q_i \rangle,$$
$$\Lambda \Rightarrow (\Lambda)_{ij} = \langle -\nabla \cdot \boldsymbol{\Phi}_j, -\nabla \cdot \boldsymbol{\Phi}_i \rangle,$$

and vectors:

$$(\tilde{g})_i = \langle f, q_i \rangle, \qquad (g)_i = \langle f, -\nabla \cdot \boldsymbol{\Phi}_i \rangle.$$

3.1 "Galerkin Constraint"

Letting $C = \tilde{B}$, a standard Galerkin-type construction of the divergence constraint is obtained. It should be noted that the order of the polynomials for the constraints, k_2, can be different from the order for the FOSLS unknowns, k_1, and, in fact, should be chosen to have less degrees of freedom so as not to over-constrain the system. The pairs chosen in this paper are quadratics–linears ($\mathscr{P}_2 - \mathscr{P}_1$), quadratics–constants ($\mathscr{P}_2 - \mathscr{P}_0$), and linears–constants ($\mathscr{P}_1 - \mathscr{P}_0$). In this context, $\mathscr{U} \in \left[\mathscr{P}_{k_1}\right]^3$ and $\lambda \in \mathscr{P}_{k_2}$. The resulting system is

$$\begin{pmatrix} A & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ \tilde{g} \end{pmatrix}. \tag{12}$$

3.2 "Least-Squares Constraint"

To keep faith with the FOSLS methodology, a constraint is proposed that is of the same form as that is used in the FOSLS discretization, namely, letting $C = \Lambda$. This allows the same finite-element spaces for the FOSLS unknowns to be used for the Lagrange multiplier. The system is then

$$\begin{pmatrix} A & \Lambda \\ \Lambda & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ g \end{pmatrix}, \tag{13}$$

where $\mathscr{U} \in \left[\mathscr{P}_{k_1}\right]^3$ and $\lambda \in \left[\mathscr{P}_{k_1}\right]^2$.

As is shown below, the system that needs to be solved in the least-squares constraint approach may not be well conditioned. However, one can construct the constraint matrix C in such a way that it can be decomposed into a form which is much easier to solve. For instance, decompose $\Lambda = B^T B$ (see definition of B in Sect. 3.3.1), and thus, the system is rewritten as

$$\begin{pmatrix} A & B^T B \\ B^T B & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ g \end{pmatrix}. \tag{14}$$

However, the construction of B is not trivial in many cases (again, see Sect. 3.3.1) and it is easier to work with $\tilde{B}$ instead. If the system in the "Galerkin" approach is taken and modified, the following is obtained:

$$\begin{pmatrix} A & \tilde{B}^T \tilde{B} \\ \tilde{B}^T \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\mathscr{U}} \\ \tilde{\lambda} \end{pmatrix} = \begin{pmatrix} b \\ \tilde{B}^T \tilde{g} \end{pmatrix}. \tag{15}$$

As it turns out, due to the following lemma, it is reasonable to solve system (15) instead of system (14).

Lemma 1. *Consider systems* (12) *and* (15). *Let A, $\tilde{B}$, $\mathscr{U}$, λ, $\tilde{\lambda}$, g, and $\tilde{g}$ be all defined as above in Definition 1, then,*

$$\lambda = \tilde{B}\tilde{\lambda} \text{ and } \tilde{\mathscr{U}} = \mathscr{U}.$$

Proof. First combine the two systems:

$$A\mathscr{U} + \tilde{B}^T \lambda = b, \tag{16}$$

$$\tilde{B}\mathscr{U} = \tilde{g}, \tag{17}$$

$$A\tilde{\mathscr{U}} + \tilde{B}^T \tilde{B}\tilde{\lambda} = b, \tag{18}$$

$$\tilde{B}^T \tilde{B}\tilde{\mathscr{U}} = \tilde{B}^T \tilde{g}. \tag{19}$$

Next, multiply (17) on the left by $\tilde{B}^T$ and subtract the bottom two equations from the top two. Let $e_{\mathscr{U}} = \mathscr{U} - \tilde{\mathscr{U}}$ and $e_\lambda = \lambda - \tilde{B}\tilde{\lambda}$ to obtain

$$\begin{aligned} Ae_{\mathscr{U}} + \tilde{B}^T e_\lambda &= 0, \\ \tilde{B}^T \tilde{B} e_{\mathscr{U}} &= 0. \end{aligned}$$

Since $\tilde{B}^T$ is equivalent to a gradient operator, it can be shown that it is a one-to-one operator (since divergence and, thus, $\tilde{B}$ is onto). Therefore, $\tilde{B}e_{\mathscr{U}} = 0$ and the system becomes

$$\begin{pmatrix} A & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} e_{\mathscr{U}} \\ e_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

which is the global "Galerkin" system, which is known to be invertible. As a result, $e_{\mathscr{U}} = e_\lambda = 0$ and, more importantly, $\mathscr{U} = \tilde{\mathscr{U}}$, meaning solving either system results in the same solution.

Therefore, (12) and (15) are both viable options for the constraint system. Next, each of these and some variations are tested to see which yield the best mass conservation with little extra computational work.

3.3 Solvers

To solve the constrained system, the conjugate gradient (CG) method on the Schur complement is used [33]. Solving the system in this way yields the following set of equations:

$$\begin{aligned}\mathscr{U} &= A^{-1}b - A^{-1}C^T\lambda,\\ CA^{-1}C^T\lambda &= CA^{-1}b - g.\end{aligned}$$

The second equation is solved for λ via CG and a backsolve is used to get the original $\mathscr{U}$. For the results presented here, a direct solver is used to compute A^{-1}, but in the future a multigrid solver, or whatever is used to solve the FOSLS system itself, will be substituted instead.

For the first approach (12) and second (13), the system is solved exactly as described above. In the second approach, we consider $\Lambda = B^TB$, where $(B)_{ij}$ represents the construction of $\langle -\nabla\cdot\Phi_j, r_i\rangle$, but where $r_i = \nabla\cdot\Phi_i$ is in the divergence of the space used for A, i.e., $\nabla\cdot[\mathscr{P}_{k_1}]^2$ as opposed to the full $\mathscr{P}_{k_2}$. As a result, the Schur complement equation becomes

$$B^TBA^{-1}B^TB\lambda = B^TBA^{-1}b - g. \tag{20}$$

This is badly conditioned as the system B^TB is equivalent to a $-\nabla\nabla\cdot$ (grad-div) equation. However, to remedy this, the equation is multiplied on the left by BA^{-1}, resulting in

$$(BA^{-1}B^T)(BA^{-1}B^T)B\lambda = (BA^{-1}B^T)BA^{-1}b - (BA^{-1})g.$$

Notice that BB^T is equivalent to a $-\nabla\cdot\nabla$, or Laplace system, and, thus, $BA^{-1}B^T$ is well conditioned. In addition, one only needs to solve for $B\lambda$. This system simplifies further by eliminating one of the $BA^{-1}B^T$ blocks to obtain

$$(BA^{-1}B^T)B\lambda = BA^{-1}b - (BA^{-1}B^T)^{-1}BA^{-1}g. \tag{21}$$

However, in (21), two solves of $BA^{-1}B^T$ are required, increasing the number of iterations required to solve the system.

In addition, a problem with this approach is the construction of B. A simpler way is to construct $\tilde{B}$ and use this instead to get system (15). This results in

$$\tilde{B}^T\tilde{B}A^{-1}\tilde{B}^T\tilde{B}\lambda = \tilde{B}^T\tilde{B}A^{-1}b - \tilde{B}^T\tilde{g}. \tag{22}$$

Multiplying on the left by $\tilde{B}A^{-1}$ yields

$$(\tilde{B}A^{-1}\tilde{B}^T)(\tilde{B}A^{-1}\tilde{B}^T)\tilde{B}\lambda = (\tilde{B}A^{-1}\tilde{B}^T)\tilde{B}A^{-1}b - (\tilde{B}A^{-1}\tilde{B}^T)\tilde{g}$$

$$(\tilde{B}A^{-1}\tilde{B}^T)\tilde{B}\lambda = \tilde{B}A^{-1}b - \tilde{g}. \tag{23}$$

This, however, is the same system obtained from (12) and, as shown in Lemma 1, results in the same solution for $\mathscr{U}$.

3.3.1 Construction of B

Despite being able to use the simpler construction, $\tilde{B}$, it is possible to construct B for the type of constraint considered here, $\nabla\cdot\mathbf{u} = f$. In fact, the matrix B is constructed locally using the simpler construction of $\tilde{B}$. Consider an element (triangle) T and let $[\mathscr{P}_{k_1}]^2(T)$ be the vector polynomials of degree k_1. Next, consider the "least-squares" constraint, where the space of Lagrange multipliers, λ, is $\nabla\cdot[\mathscr{P}_{k_1}]^2(T)$, which is a subspace of $[\mathscr{P}_{k_1-1}](T)$. Let $\{\varphi_s\}_{s=1}^l$ be the basis (restricted to T) of $[\mathscr{P}_{k_1-1}](T)$. For $k_1 = 2$, $l = 3$ (since $[\mathscr{P}_{k_1-1}](T) = [\mathscr{P}_1](T)$—the space of linears). Also, let $\{\boldsymbol{\Phi}_i\}_{i=1}^n$ be the basis of $[\mathscr{P}_{k_1}]^2(T)$. Since $\nabla\cdot\boldsymbol{\Phi}_i \in \nabla\cdot[\mathscr{P}_{k_1}]^2(T) \subset [\mathscr{P}_{k_1-1}](T)$,

$$\nabla\cdot\boldsymbol{\Phi}_i = \sum_{s=1}^{l} c_{i,s}\varphi_s = [\varphi_1,\ \ldots,\ \varphi_l]\mathbf{c}_i, \tag{24}$$

for some coefficients $\mathbf{c}_i = (c_{i,s}) \in \mathbb{R}^l$. Therefore,

$$\left(\tilde{B}_T\right)_{s,i} = \langle\nabla\cdot\boldsymbol{\Phi}_i,\ \varphi_s\rangle = [\langle\varphi_s,\varphi_1\rangle,\ \ldots,\ \langle\varphi_s,\varphi_l\rangle]\mathbf{c}_i = \mathbf{e}_s^T M\mathbf{c}_i.$$

Here, $\mathbf{e}_s \in \mathbb{R}^l$ is the sth unit coordinate vector and $M = M_T$ is the element mass matrix coming from the space $[\mathscr{P}_{k_1-1}](T)$. In conclusion, the element matrix $\tilde{B} = \tilde{B}_T = (\langle\nabla\cdot\boldsymbol{\Phi}_i,\ \varphi_s\rangle)_{1\le i\le n,\ 1\le s\le l}$ admits the following form:

$$\tilde{B}_T = M_T[\mathbf{c}_1,\ \mathbf{c}_2,\ \ldots,\ \mathbf{c}_n].$$

For the entries $\langle\nabla\cdot\boldsymbol{\Phi}_j,\ \nabla\cdot\boldsymbol{\Phi}_i\rangle = (B_T^T B_T)_{ij}$, using the representation (24) yields

$$(B_T^T B_T)_{ij} = \langle\nabla\cdot\boldsymbol{\Phi}_j,\ \nabla\cdot\boldsymbol{\Phi}_i\rangle = \mathbf{c}_j^T\left(\langle\varphi_r,\ \varphi_s\rangle\right)_{r,s=1}^{l}\mathbf{c}_i = \mathbf{c}_j^T M_T\mathbf{c}_i = \left(\tilde{B}_T^T M_T^{-1}\tilde{B}_T\right)_{ij}.$$

Therefore,

$$B_T = M_T^{-\frac{1}{2}}\tilde{B}_T.$$

Thus, B is constructed relatively easily. Namely, over each element the local matrix, $\tilde{B}_T$, is built, which is the Galerkin finite-element construction of the divergence operator using $\mathscr{P}_{k_1} - \mathscr{P}_{k_1-1}$ elements. Then, $B_T = M_T^{-1/2}\tilde{B}_T$, where $M_T^{-1/2}$ is the mass matrix associated with the given element and $\mathscr{P}_{k_1-1}$.

3.4 Numerical Results

In the following numerical tests, four approaches are considered:

- Method 1: Solve the "Galerkin" constraint system (12), resulting in (23). Note that this is the same as solving system (15) and simplifying the Schur complement system.
- Method 2: Solve the "least-squares" constraint system (13), resulting in (20).
- Method 3: Solve the "least-squares" constraint system using the simpler construction, (15), resulting in (22).
- Method 4: Solve the "least-squares" constraint system (14), with the simplified Schur complement system (21).

Again, $D = I$, $\mathbf{r} = \mathbf{0}$, and $c = 0$. The right-hand side is chosen as $f = 2\pi^2 \sin(\pi x)\sin(\pi y)$ so that the true solution is $p = \sin(\pi x)\sin(\pi y)$. The problem is solved on a unit square with homogeneous Dirichlet boundary conditions for p. The system is solved using the four approaches described above for a combination of the finite-element spaces, $\mathscr{P}_2$, $\mathscr{P}_1$, and $\mathscr{P}_0$. The L^2 norms of the errors of the numerical solutions, p and $\mathbf{u} = \nabla p$, are shown in the following tables. Here, $u_{err} = ||\mathbf{u} - \mathbf{u}^*||_0 / ||\mathbf{u}^*||_0$ and $p_{err} = ||p - p^*||_0 / ||p^*||_0$ for the constrained system, where $\mathbf{u}^*$ and p^* are the true solutions. The FOSLS functional, $\mathscr{F} = ||L\mathscr{U} - f||_0$, is given for both the unconstrained system, $\mathscr{F}$, and the constrained system, $\mathscr{F}_c$. In addition, the mass conservation (or mass loss) is shown as

$$m_L = \left| \int_{\Omega} (\nabla \cdot \mathbf{u} + f)\, d\Omega \right|,$$

for the unconstrained FOSLS system as well as with the constraint, m_L^c. In addition, we consider local conservation of mass by integrating over each element measuring the largest mass loss over all elements in the domain,

$$\hat{m}_L = \max_T \left| \int_T (\nabla \cdot \mathbf{u} + f)\, dT \right|,$$

as this is a more practical measurement for satisfying physical conservation laws. Finally, the number of iterations needed in the CG algorithm to reduce the algebraic residual by 10^{-8} is shown (Tables 1–4).

3.5 Discussion

A couple of things to note are the fact that the first test yields some of the most optimal results. Method 2 attempts to solve the ill-conditioned $\nabla\nabla\cdot$-like system and, as is shown, requires too many iterations to be used reliably. Methods 3 and 4 improve on this; however, as they require extra matrix inversions in the solution process, they require more work than in the first case.

Table 1 (Method 1) Solve $\tilde{B}A^{-1}\tilde{B}^T\lambda = \tilde{B}A^{-1}b - \tilde{g}$

k_1	k_2	h	$\hat{m}_L$	$\hat{m}_L^c$	m_L	m_L^c	$\mathscr{F}$	$\mathscr{F}_c$	u_{err}	p_{err}	Iterations
1	0	1/16	6.9e−4	5.5e−12	2.8e−2	4.9e−11	0.90	1.12	0.181	0.015	113
1	0	1/32	6.6e−5	1.6e−12	1.0e−2	2.9e−11	0.48	0.86	0.181	0.004	232
2	0	1/16	1.7e−5	1.2e−14	3.7e−5	5.7e−14	3.7e−2	3.7e−2	1.16e−3	1.40e−4	4
2	0	1/32	1.1e−6	9.5e−16	2.4e−6	3.0e−14	9.5e−3	9.5e−3	1.82e−4	1.73e−5	2
2	1	1/16	1.7e−5	1.7e−5	3.7e−5	1.9e−13	3.7e−2	3.7e−2	1.12e−3	1.38e−4	13
2	1	1/32	1.1e−6	1.0e−6	2.4e−6	3.4e−14	9.5e−3	9.5e−3	1.81e−4	1.72e−5	7

This approach is equivalent to using the “Galerkin” approach (12) and the “least-squares” approach plus simplification of the Schur complement system on $\tilde{B}$ (15)

Table 2 (Method 2) Solve $\Lambda A^{-1}\Lambda\lambda = \Lambda A^{-1}b - g$

k_1	k_2	h	$\hat{m}_L$	$\hat{m}_L^c$	m_L	m_L^c	$\mathscr{F}$	$\mathscr{F}_c$	u_{err}	p_{err}	Iterations
1	1	1/16	6.9e−4	2.9e−12	2.8e−2	5.1e−12	0.90	1.12	0.181	0.015	1,730
1	1	1/32	6.6e−5	1.6e−11	1.0e−2	8.1e−11	0.48	0.86	0.181	0.004	20,375
2	2	1/16	1.7e−5	1.6e−13	3.7e−5	1.5e−11	3.7e−2	9.8e−2	0.012	1.38e−4	1,100
2	2	1/32	1.1e−6	9.8e−15	2.4e−6	1.7e−12	9.5e−3	4.8e−2	4.94e−3	1.72e−5	4,319

This approach is equivalent to using the “least-squares” approach, but without splitting the constraint matrix and solving the full Schur complement system (13)

Table 3 (Method 3) Solve $\tilde{B}^T\tilde{B}A^{-1}\tilde{B}^T\tilde{B}\lambda = \tilde{B}^T\tilde{B}A^{-1}b - \tilde{B}^T\tilde{g}$

k_1	k_2	h	$\hat{m}_L$	$\hat{m}_L^c$	m_L	m_L^c	$\mathscr{F}$	$\mathscr{F}_c$	u_{err}	p_{err}	Iterations
1	1	1/16	6.9e−4	2.2e−12	2.8e−2	2.8e−12	0.90	1.12	0.181	0.015	1,600
1	1	1/32	6.6e−5	1.4e−11	1.0e−2	2.9e−11	0.48	0.86	0.181	0.004	15,268
2	1	1/16	1.7e−5	9.2e−14	3.7e−5	9.3e−13	3.7e−2	3.7e−2	1.16e−3	1.40e−4	15
2	1	1/32	1.1e−6	1.2e−14	2.4e−6	7.5e−14	9.5e−3	9.5e−3	1.82e−4	1.73e−5	4
2	2	1/16	1.7e−5	1.7e−5	3.7e−5	5.9e−14	3.7e−2	3.7e−2	1.15e−3	1.38e−4	12
2	2	1/32	1.1e−6	1.0e−6	2.4e−6	7.3e−14	9.5e−3	9.5e−3	1.81e−4	1.72e−5	6

This approach is equivalent to using the “least-squares” approach with the simpler construction of the constraint, but without splitting the constraint matrix and solving the full Schur complement system (15)

Table 4 (Method 4) Solve $\Lambda A^{-1}\Lambda\lambda = \Lambda A^{-1}b - g$

k_1	k_2	h	$\hat{m}_L$	$\hat{m}_L^c$	m_L	m_L^c	$\mathscr{F}$	$\mathscr{F}_c$	u_{err}	p_{err}	Iterations
1	1	1/16	6.9e−4	3.9e−8	2.8e−2	1.3e−10	0.90	1.12	0.181	0.015	84+134
1	1	1/32	6.6e−5	4.0e−8	1.0e−2	7.8e−10	0.48	0.86	0.181	0.004	146+307
2	2	1/16	1.7e−5	9.6e−10	3.7e−5	5.1e−10	3.7e−2	9.8e−2	0.012	1.38e−4	72+101
2	2	1/32	1.1e−6	5.7e−10	2.4e−6	1.7e−9	9.5e−3	4.8e−2	7.73e−3	1.72e−5	124+198

This approach is equivalent to using the “least-squares” approach and using the simplification of the full Schur complement system using B (21). Note that since two solves of $BA^{-1}B^T$ are required, the iterations for both solves are displayed in the last column of the table

In addition, only when a stable pair of elements with the constraint is used (i.e., $\mathscr{P}_2 - \mathscr{P}_0$ or $\mathscr{P}_2 - \mathscr{P}_1$) are the optimal results obtained. This results from the fact that only for the stable combinations is there enough room to minimize the FOSLS functional. All cases yield improved conservation as this is enforced directly. However, for the unstable pairings as the constraint is enforced, only a few possible solutions are allowed and, as a result, when the FOSLS functional is minimized, there is no longer enough room to minimize certain terms in the functional any more (such as $\mathbf{u} - \nabla p = 0$). Thus, the best $\mathbf{u}$ is not found. The solution has better mass conservation, but the approximation is not necessarily capable of minimizing the FOSLS functional. This can be seen by looking at the reduction in the error of $\mathbf{u}$. In all cases, the solution, p, is approximated well and the error is reduced with h as expected. However, for the unstable pairs, the gradient, $\mathbf{u}$, is not approximated well. Thus, the functional is no longer estimating the H^1 error accurately and the a posteriori error estimator is lost. Therefore, the conclusion is that the constraint always needs to be chosen from a space which gives a stable finite-element pair with whatever unknowns from the FOSLS system that you wish to conserve. This requires considering an inf–sup condition for the FOSLS unknown and Lagrange multiplier pairs, but in many applications these pairs of spaces are well known [12, 20, 21]. In addition, it should be noted that we also obtain *local* conservation across the elements when the constraint space uses discontinuous elements (i.e., $\mathscr{P}_0$, $\nabla \cdot [\mathscr{P}_1]^2$, or $\nabla \cdot [\mathscr{P}_2]^2$). This is similar to mixed finite-element methods where $\int_T (\nabla \cdot \mathbf{u} + f)\; dT$ will be zero (or small if the system is solved only approximately) for each element T.

Alternatively, we may use for the constraints test functions from a coarse subspace of a space that generally may not provide a stable fine–grid pair. For instance, if the constraint matrix, $\tilde{B}$, is constructed using the "Galerkin-like" approach using the same polynomial space as the FOSLS system, the finite-element pairs are not stable. However, if this operator is restricted to a coarser space, H, and the Lagrange multiplier, λ_H, is chosen in that coarser space, stability is regained (assuming the coarse space is "coarse enough"). In the following results, this is tested using linears and quadratics. An interpolation operator is constructed via standard finite-element interpolation, Q_H, which takes DOF from a grid of size H and interpolates it to the fine–grid, h. Thus, the constrained system becomes

$$\begin{pmatrix} A & \tilde{B}^T Q_H \\ Q_H^T \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U} \\ \lambda_H \end{pmatrix} = \begin{pmatrix} b \\ Q_H^T \tilde{g} \end{pmatrix}. \tag{25}$$

As is seen in Table 5, using $\mathscr{P}_1 - \mathscr{P}_1$ and $\mathscr{P}_2 - \mathscr{P}_2$ pairs yields conservation and still allows the FOSLS functional to be minimized as expected. Thus, the solution, p, and its gradient, $\mathbf{u}$, are approximated well with only a handful of extra iterations needed. Again, if the coarse Lagrange multiplier space were discontinuous, local conservation would also be obtained over the coarse elements.

Table 5 (Alternative approach) solve (25), where $Q_H^T \tilde{B}$ is the "Galerkin" constraint on a coarser mesh

k_1	k_2	h	H	m_L	m_L^c	$\mathscr{F}$	$\mathscr{F}_c$	u_{err}	p_{err}	Iterations
1	1	1/8	1/4	5.3e−2	1.9e−12	1.65	1.70	0.222	0.056	21
1	1	1/16	1/8	2.8e−2	9.0e−12	0.90	0.91	0.132	0.013	27
1	1	1/16	1/4	2.8e−2	3.0e−11	0.90	0.90	0.134	0.015	17
1	1	1/32	1/16	1.0e−2	6.6e−13	0.48	0.48	0.053	0.003	25
1	1	1/32	1/8	1.0e−2	7.1e−12	0.48	0.48	0.053	0.004	17
1	1	1/32	1/4	1.0e−2	4.0e−12	0.48	0.48	0.053	0.004	13
2	2	1/8	1/4	5.5e−4	1.5e−11	0.14	0.15	0.008	0.001	31
2	2	1/16	1/8	3.7e−5	7.0e−13	3.7e−2	3.9e−2	1.10e−3	1.88e−4	28
2	2	1/16	1/4	3.7e−5	8.6e−13	3.7e−2	3.8e−2	1.11e−3	1.39e−4	22
2	2	1/32	1/16	2.4e−6	4.6e−13	9.5e−3	9.9e−3	1.79e−4	3.69e−5	22
2	2	1/32	1/8	2.4e−6	7.5e−14	9.5e−3	9.6e−3	1.79e−4	2.32e−5	17
2	2	1/32	1/4	2.4e−6	9.4e−14	9.5e−3	9.5e−3	1.80e−4	1.83e−5	16

4 Locally Constrained FOSLS Correction

4.1 *Overlapping Schwarz Corrections*

Now that it has been shown that augmenting the FOSLS system with a constraint gives better mass conservation with only a few extra iterations, a more robust local way of solving the problem is described here. An overlapping Schwarz process, as described in [37] (Sect. 9.5), is considered to break the constrained problem into smaller local problems. First consider that the FOSLS discrete system has been solved. In other words, no constraints are yet imposed. Then, the following post-processing step is performed. Let $\{\Omega_i\}_{i=1}^{N_{sd}}$ be an overlapping partition of Ω into N_{sd} mesh subdomains (i.e., each Ω_i is a union of fine–grid elements). Then, correct the current solution $\mathscr{U}$ with

$$\mathscr{U}_i \in V_h^0(\Omega_i) = \left\{\mathbf{v} \in V_h : \text{supp}\,(\mathbf{v}_i) \subset \overline{\Omega_i}\right\},$$

by solving the locally constrained minimization problem for $\mathscr{U}_i \in V_h^0(\Omega_i)$ and $\lambda_i \in \mathscr{R}_i = \nabla \cdot V_h^0(\Omega_i)$ posed in Ω_i:

$$\begin{aligned} a(\mathscr{U} + \mathscr{U}_i,\, \mathbf{v}_i) + \langle \lambda_i,\, \nabla \cdot \mathbf{v}_i \rangle &= \langle F,\, \mathbf{v}_i \rangle, \text{ for all } \mathbf{v}_i \in V_h^0(\Omega_i), \\ \langle \nabla \cdot (\mathscr{U} + \mathscr{U}_i),\, \varphi \rangle &= \langle f,\, \varphi \rangle \text{ for } \varphi \in \mathscr{R}_i. \end{aligned}$$

Here, for the local space $\mathscr{R}_i \equiv \nabla \cdot V_h^0(\Omega_i)$, the local systems can be constructed as in Sect. 3.3.1. Likewise, a computational basis, based on QR or SVD, can be obtained as well. This is feasible if the domains Ω_i are relatively small. Next, set $\mathscr{U} := \mathscr{U} + \mathscr{U}_i$ and move onto the next subdomain Ω_{i+1}.

After several loops over the Schwarz subdomains, a global coarse-space correction is performed. For this, a coarse space, $\mathscr{R}_H \subset \nabla \cdot V_h$, is needed with an explicit locally supported basis such that the pair $(V_h,\ \mathscr{R}_H)$ is LBB-stable (Ladyzenskaya–Babuska–Brezzi condition) [10, 12]. Alternatively, based on a coarse space, $V_H \subset V_h$, and coarser subdomains, $\{\Omega_i^H\}$ (i.e., union of coarse elements in $\mathscr{T}_H$), for the current approximation $\mathscr{U} \in V_h$, local coarse-space corrections, $\mathscr{U}_i^H \in V_H^0(\Omega_i^H) = \{\mathbf{v}_H \in V_H :\ \text{supp}\ (\mathbf{v}_H) \subset \overline{\Omega}_i^H\}$, are obtained by solving the local saddle-point problems for $\mathscr{U}_i^H \in V_H^0(\Omega_i)$ and $\lambda_i^H \in \mathscr{R}_i^H = \nabla \cdot V_H^0(\Omega_i^H)$ posed in Ω_i^H:

$$\begin{aligned} a(\mathscr{U} + \mathscr{U}_i^H,\ \mathbf{v}_i^H) + \langle \lambda_i^H,\ \nabla \cdot \mathbf{v}_i^H \rangle &= \langle F,\ \mathbf{v}_i^H \rangle, \text{ for all } \mathbf{v}_i^H \in V_H^0(\Omega_i^H), \\ \langle \nabla \cdot (\mathscr{U} + \mathscr{U}_i^H),\ \varphi \rangle &= \langle f,\ \varphi \rangle \text{ for } \varphi \in \mathscr{R}_i^H. \end{aligned}$$

Here, the coarse spaces can be constructed in a variational way by using standard interpolation and restriction operators for polynomial finite-element spaces. Finally, let $\mathscr{U} := \mathscr{U} + \mathscr{U}_i^H$ and move onto the next coarse subdomain Ω_{i+1}^H. The process can be applied recursively in a V-cycle iteration exploiting the above constrained overlapping Schwarz (Vanka-like) smoothing corrections [36]. For this paper, however, we consider only a two-level method with one global coarse space.

4.2 Numerical Results

To test the scheme described above in Sect. 4.1, the "Galerkin"-like constrained system (12) is considered on subdomains and a coarse grid. This system gave the most optimal results (fewer iterations and better mass conservation) and, therefore, appears to be the natural choice for performing the subdomain corrections. As described above, the standard FOSLS system is solved yielding, $\mathscr{U}_0$, which is used as the initial guess for the overlapping Schwarz method. Next, the finite-element triangulation of Ω is divided into overlapping subdomains, $\mathscr{T}_i$ of Ω_i. The restriction of the FOSLS system, A, and the constraint equation, $\tilde{B}$, is formed by a simple projection onto the subdomains giving, $A_i = P_i^T A P_i$ and $B_i = Q_i^T \tilde{B} P_i$. Here, P_i and Q_i are the natural injection operators of DOFs on $\mathscr{T}_i$ to the original mesh, $\mathscr{T}$, for elements of $\mathscr{P}_{k_1}$ and $\mathscr{P}_{k_2}$, respectively. Then, on each subdomain the Schur complement system of the error equations is solved as described above in Sect. 4.1. Once all corrections on subdomains are updated, the system is projected onto a coarse grid, $\mathscr{T}_H$, where an update is again solved for. We use the standard finite-element interpolation operators to move between a coarse grid of size H to a fine grid of size h. We define these as P_H for $\mathscr{P}_{k_1}$ and Q_H for $\mathscr{P}_{k_2}$. Note that P_H is a block matrix of interpolation operators for each unknown in the FOSLS system. The transposes are used as restriction operators from fine grid to coarse grid.

The algorithm is described below, letting M_s be the maximum number of subdomain smoothing steps and N_{sd} being the number of overlapping subdomains:

```
Solve FOSLS System: $A\mathscr{U}_0 = b$.
Compute Residuals: $r_A = b - A\mathscr{U}_0$ and $r_B = \tilde{g} - \tilde{B}\mathscr{U}_0$.
Set $\mathscr{U} = \mathscr{U}_0$ and $\lambda = 0$.
Perform Subdomain Smoothing Steps:
for $s = 1$ to $M_s$ do
    for $i = 1$ to $N_{sd}$ do
        Restrict Matrices and Residuals to Subdomains.
        Solve: $\begin{pmatrix} A_i & B_i^T \\ B_i & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U}_i \\ \lambda_i \end{pmatrix} = \begin{pmatrix} P_i^T r_A \\ Q_i^T r_B \end{pmatrix}$.
        Update: $\mathscr{U} = \mathscr{U} + P_i \mathscr{U}_i$ and $\lambda = \lambda + Q_i \lambda_i$.
        Recompute Residuals: $r_A = b - A\mathscr{U} - \tilde{B}^T \lambda$ and $r_B = \tilde{g} - \tilde{B}\mathscr{U}_0$.
    end
end
Perform Coarse-Grid Correction:
Solve: $\begin{pmatrix} P_H^T A P_H & P_H^T \tilde{B}^T Q_H \\ Q_H^T \tilde{B} P_H & 0 \end{pmatrix} \begin{pmatrix} \mathscr{U}_H \\ \lambda_H \end{pmatrix} = \begin{pmatrix} P_H^T r_A \\ Q_H^T r_B \end{pmatrix}$.
Update: $\mathscr{U} = \mathscr{U} + P_H \mathscr{U}_H$.
```

The results for $\mathscr{P}_2 - \mathscr{P}_0$ and $\mathscr{P}_1 - \mathscr{P}_0$ pairs of elements for the FOSLS solution and the constraint variable are given in Table 6 using various grid spacings. The first set of results is given for the original FOSLS system with no constraint correction. The FOSLS functional is reduced by h^{k_1} as expected and it gives a good approximation of the reduction in error for both $\mathbf{u}$ and p. However, the mass loss is rather large. Using quadratics improves the results but not exactly. The remaining blocks of data give the results using various numbers of smoothing steps and with or without coarse-grid corrections. In all cases, using $\mathscr{P}_2 - \mathscr{P}_0$ elements gives much better results. As seen in Tables 1 and 2, mass conservation is obtained, and the FOSLS functional is still minimized, retaining its error approximation properties. Moreover, using unstable pairs of elements can even result in the divergence of the FOSLS functional. In the context of this problem, the solution is still obtained accurately, but the gradient of the solution is not captured well. The solution process is no longer minimizing the residual in the H^1 norm.

In addition, the results show that the use of a coarse grid improves the performance of the method. The second block in Table 6 shows results for performing one smoothing step of the subdomain solver with no coarse-grid correction. This does improve the conservation results, but not significantly. Performing 100 smoothing steps of the subdomain solver with no coarse-grid correction improves the mass conservation, but of course these iterations are expensive. Finally, the fourth set shows results for using one step of the subdomain solver with one

Table 6 Mass loss, least-squares functional, and relative errors of solutions for $\mathscr{P}_1 - \mathscr{P}_0$ elements (left) and $\mathscr{P}_2 - \mathscr{P}_0$ elements (right)

	$\mathscr{P}_1 - \mathscr{P}_0$				$\mathscr{P}_2 - \mathscr{P}_0$			
	m_L	$\mathscr{F}$	u_{err}	p_{err}	m_L	$\mathscr{F}$	u_{err}	p_{err}
$1/h$	FOSLS							
8	5.3e−2	1.65	0.223	0.070	5.5e−4	0.14	8.3e−3	1.2e−3
16	2.8e−2	0.90	0.134	0.019	3.7e−5	0.04	1.1e−3	1.4e−4
32	1.0e−2	0.48	0.053	0.005	2.4e−6	0.01	1.8e−4	1.7e−5
$1/h$	$N_{sd} = 9$, $M_s = 1$, No coarse-grid correction							
8	2.6e−3	1.77	0.180	0.061	2.3e−6	0.14	8.4e−3	1.2e−3
16	1.3e−3	1.09	0.185	0.016	1.0e−7	0.04	1.2e−3	1.4e−4
32	2.9e−5	1.47	0.201	0.004	9.2e−9	0.01	1.8e−4	1.7e−5
$1/h$	$N_{sd} = 9$, $M_s = 100$, No coarse-grid correction							
8	4.2e−12	1.83	0.184	0.059	1.0e−11	0.14	8.4e−3	1.2e−3
16	4.7e−8	1.12	0.181	0.015	9.1e−11	0.04	1.2e−3	1.4e−4
32	2.2e−1	7.81	0.413	0.005	4.5e−11	0.01	1.8e−4	1.7e−5
$1/h$	$N_{sd} = 9$, $M_s = 1$, $H = 2h$							
8	8.8e−11	1.83	0.181	0.060	2.3e−13	0.14	8.3e−3	1.2e−3
16	1.3e−12	1.92	0.188	0.015	1.2e−13	0.04	1.2e−3	1.4e−4
32	1.1e−10	10.09	0.209	0.004	2.5e−14	0.01	1.8e−4	1.7e−5
$1/h$	$N_{sd} = 9$, $M_s = 1$, $H = 4h$							
8	5.3e−3	1.84	0.191	0.060	1.2e−6	0.14	8.3e−3	1.2e−3
16	1.2e−3	1.20	0.186	0.016	1.8e−8	0.04	1.2e−3	1.4e−4
32	2.8e−4	2.63	0.200	0.004	3.0e−9	0.01	1.8e−4	1.7e−5
$1/h$	$N_{sd} = 9$, $M_s = 10$, $H = 4h$							
8	3.9e−4	1.83	0.195	0.060	1.1e−7	0.14	8.3e−3	1.2e−3
16	4.0e−3	1.29	0.192	0.015	7.8e−9	0.04	1.2e−3	1.4e−4
32	4.3e−2	9.93	0.363	0.007	6.7e−10	0.01	1.8e−4	1.7e−5

solve on a coarse grid. The mass conservation is retained and not much work is needed. Combining with the results from Table 1, this process requires around four iterations of MINRES for each local subdomain and for the coarse grid. Each of these subdomains has less DOFs, and therefore, the work required to solve the constrained system is a fraction of the cost of solving the original FOSLS system.

5 Conclusions

In summary, the results of this paper have shown that properties such as mass conservation can be obtained using the least-squares finite-element method and a post-process subdomain correction method. There are many other methods, as mentioned in the Introduction (Sect. 1), that also improve conservation properties for least-squares problems. These may involve reformulating the system or choosing better finite-element spaces for the original FOSLS system. For instance, nonconforming elements can be used that satisfy the mass conservation across interfaces much better than the standard polynomial spaces used here [1, 17, 18, 25]. The goal of our approach in this paper is to show that the system can be solved as is, with no alterations to the original FOSLS method. Thus, it should be considered a robust finite-element method for such systems which obtains physically accurate solutions efficiently. Care needs to be given in choosing the right spaces for the constraint system, so that a stable method is obtained and the FOSLS functional retains its important a posteriori error estimator properties. This includes considering discontinuous spaces, in order to ensure *local* conservation across smaller regions of the domain. However, since this post-processing is done on local subdomains and/or on coarse grids, only a fractional amount of computational cost is added to the solution process. Future work involves implementing the above algorithms in a multilevel way and including the coarse-space constraints in the local subdomain process. Also, other applications such as Stokes flow and magnetohydrodynamics are worth considering.

References

1. Baker, G.A., Jureidini, W.N., Karakashian, O.A.: Piecewise solenoidal vector-fields and the Stokes problem. SIAM J. Numer. Anal. **27**(6), 1466–1485 (1990)
2. Barth, T.: On the role of involutions in the discontinuous Galerkin discretization of Maxwell and magnetohydrodynamic systems. In: Arnold, D.N., Bochev, P.B., Lehoucq, R.B., Nicolaides, R.A., Shashkov, M. (eds.) Compatible Spatial Discretizations, pp. 69–88. Springer, New York (2006)
3. Berndt, M., Manteuffel, T.A., McCormick, S.F.: Local error estimates and adaptive refinement for first-order system least squares (FOSLS). Electron. Trans. Numer. Anal. **6**, 35–43 (1997)
4. Bochev, P., Gunzburger, M.D.: Analysis of least-squares finite-element methods for the Stokes equations. Math. Comput. **63**(208), 479–506 (1994)
5. Bochev, P.B., Gunzburger, M.D.: A locally conservative least-squares method for Darcy flows. Commun. Numer. Meth. Eng. **24**(2), 97–110 (2008)
6. Bochev, P., Cai, Z., Manteuffel, T.A., McCormick, S.F.: Analysis of velocity-flux first-order system least-squares principles for the Navier-Stokes equations: part I. SIAM J. Numer. Anal. **35**(3), 990–1009 (1998)
7. Bochev, P., Manteuffel, T.A., McCormick, S.F.: Analysis of velocity-flux least-squares principles for the Navier-Stokes equations: part II. SIAM J. Numer. Anal. **36**(4), 1125–1144 (1999)
8. Brackbill, J., Barnes, D.C.: The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations. J. Comput. Phys. **35**(3), 426–430 (1980)
9. Bramble, J., Kolev, T., Pasciak, J.: A least-squares approximation method for the time-harmonic Maxwell equations. J. Numer. Math. **13**(4), 237 (2005)
10. Brenner, S.C., Scott, L.R.: Mathematical Theory of Finite Element Methods, 2nd edn. Springer, New York (2002)
11. Brezina, M., Garcia, J., Manteuffel, T., McCormick, S., Ruge, J., Tang, L.: Parallel adaptive mesh refinement for first-order system least squares. Numer. Lin. Algebra Appl. **19**, 343–366 (2012)
12. Brezzi, F., Fortin, M.: Mixed and Hybrid Finite Elements Methods. Springer Series in Computational Mathematics. Springer, Berlin (1991)

13. Cai, Z., Starke, G.: Least-squares methods for linear elasticity. SIAM J. Numer. Anal. **42**(2), 826–842 (electronic) (2004). doi:10.1137/S0036142902418357. http://dx.doi.org.ezproxy.library.tufts.edu/10.1137/S0036142902418357
14. Cai, Z., Lazarov, R., Manteuffel, T.A., McCormick, S.F.: First-order system least squares for second-order partial differential equations: part I. SIAM J. Numer. Anal. **31**, 1785–1799 (1994)
15. Cai, Z., Manteuffel, T.A., McCormick, S.F.: First-order system least squares for second-order partial differential equations 2. SIAM J. Numer. Anal. **34**(2), 425–454 (1997)
16. Carey, G.F., Pehlivanov, A.I., Vassilevski, P.S.: Least-squares mixed finite element methods for non-selfadjoint elliptic problems II: performance of block-ILU factorization methods. SIAM J. Sci. Comput. **16**, 1126–1136 (1995)
17. Cockburn, B., Kanschat, G., Schotzau, D.: A locally conservative LDG method for the incompressible Navier-Stokes equations. Math. Comput. **74**(251), 1067–1095 (2005)
18. Cockburn, B., Kanschat, G., Schötzau, D.: A note on discontinuous Galerkin divergence-free solutions of the Navier-Stokes equations. J. Sci. Comput. **31**, 61–73 (2007)
19. De Sterck, H., Manteuffel, T., McCormick, S., Nolting, J., Ruge, J., Tang, L.: Efficiency-based h- and hp-refinement strategies for finite element methods. Numer. Lin. Algebra Appl. **15**, 89–114 (2008)
20. Girault, V., Raviart, P.A.: Finite Element Approximation of the Navier-Stokes Equations, revised edn. Springer, Berlin (1979)
21. Girault, V., Raviart, P.A.: Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms (Springer Series in Computational Mathematics). Springer, Berlin (1986)
22. Heys, J.J., Lee, E., Manteuffel, T.A., McCormick, S.F.: On mass-conserving least-squares methods. SIAM J. Sci. Comput. **28**(5), 1675–1693 (2006)
23. Heys, J.J., Lee, E., Manteuffel, T.A., McCormick, S.F.: An alternative least-squares formulation of the Navier-Stokes equations with improved mass conservation. J. Comput. Phys. **226**(1), 994–1006 (2007)
24. Heys, J.J., Lee, E., Manteuffel, T.A., McCormick, S.F., Ruge, J.W.: Enhanced mass conservation in least-squares methods for Navier-Stokes equations. SIAM J. Sci. Comput. **31**(3), 2303–2321 (2009)
25. Karakashian, O., Jureidini, W.: A nonconforming finite element method for the stationary Navier-Stokes equations. SIAM J. Numer. Anal. **35**(1), 93–120 (1998)
26. Lax, P.D.: Functional analysis. Wiley-Interscience, New York (2002)
27. Lee, E., Manteuffel, T.A.: FOSLL* method for the eddy current problem with three-dimensional edge singularities. SIAM J. Numer. Anal. **45**(2), 787–809 (2007)
28. Manteuffel, T.A., McCormick, S.F., Ruge, J., Schmidt, J.G.: First-order system LL* (FOSLL*) for general scalar elliptic problems in the plane. SIAM J. Numer. Anal. **43**(5), 2098–2120 (2006)
29. Münzenmaier, S., Starke, G.: First-order system least squares for coupled Stokes-Darcy flow. SIAM J. Numer. Anal. **49**(1), 387–404 (2011). doi:10.1137/100805108. http://dx.doi.org.ezproxy.library.tufts.edu/10.1137/100805108
30. Pehlivanov, A.I., Carey, G.F.: Error-estimates for least-squares mixed finite-elements. ESAIM Math. Model. Numer. Anal. **28**(5), 499–516 (1994)
31. Pehlivanov, A.I., Carey, G.F., Vassilevski, P.S.: Least-squares mixed finite element methods for non-selfadjoint elliptic problems I: error analysis. Numer. Math. **72**, 501–522 (1996)
32. Pontaza, J.P., Reddy, J.N.: Space-time coupled spectral/hp least-squares finite element formulation for the incompressible Navier-Stokes equations. J. Comput. Phys. **197**(2), 418–459 (2004)
33. Saad, Y.: Iterative Methods for Sparse Linear Systems. Society for Industrial & Applied Mathematics, Philadelphia (2003)
34. Starke, G.: A first-order system least squares finite element method for the shallow water equations. SIAM J. Numer. Anal. **42**(6), 2387–2407 (electronic) (2005). doi:10.1137/S0036142903438124. http://dx.doi.org.ezproxy.library.tufts.edu/10.1137/S0036142903438124

35. Toselli, A., Widlund, O.B.: Domain Decomposition Methods—Algorithms and Theory. Springer, New York (2005)
36. Vanka, S.P.: Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. J. Comput. Phys. **65**(1), 138–158 (1986)
37. Vassilevski, P.S.: Multilevel Block Factorization Preconditioners. Matrix-based Analysis and Algorithms for Solving Finite Element Equations. Springer, New York (2008)