Noname manuscript No.
(will be inserted by the editor)

A Numerical Investigation on the Interplay Amongst Geometry, Meshes, and Linear Algebra in the Finite Element Solution of Elliptic PDEs

Jibum Kim ·
Shankar Prasad Sastry ·
Suzanne M. Shontz[*]

Received: date / Accepted: date

Abstract In this paper, we study the effect the choice of mesh quality metric, preconditioner, and sparse linear solver have on the numerical solution of elliptic partial differential equations (PDEs). We smooth meshes on several geometric domains using various quality metrics and solve the associated elliptic PDEs using the finite element method. The resulting linear systems are solved using various combinations of preconditioners and sparse linear solvers. We use the inverse mean ratio and radius ratio metrics in addition to conditioning-based scale-invariant and interpolation-based size-and-shape metrics. We employ the Jacobi, SSOR, incomplete LU, and algebraic multigrid preconditioners and the conjugate gradient, minimum residual, generalized minimum residual, and bi-conjugate gradient stabilized solvers. We focus on determining the most efficient quality metric, preconditioner, and linear solver combination for the numerical solution of various elliptic PDEs with isotropic coefficients. We also investigate the effect of vertex perturbation and the effect of increasing the problem size on the number of iterations required to converge and on the solver time. In this paper, we consider Poisson's equation, general second-order elliptic PDEs, and linear elasticity problems.

1 Introduction

Discretization methods, such as the finite element (FE) method, are commonly used in the numerical solution of partial differential equations (PDEs). The accuracy of the computed PDE solution depends on the degree of the approximation scheme, the number of elements in the mesh [1], and the quality of the mesh [2,3]. More specifically, it is known that as the element dihedral angles become too large, the discretization error in the finite element solution increases [1]. In addition, the stability and convergence of the finite element method is affected by poor quality elements [4]. In particular, it is known that as the angles become too small, the condition number of the finite element matrix increases [4].

Analytical studies have been performed at the intersection of meshing and linear solvers. For example, mathematical connections between mesh geometry, interpolation errors, and stiffness matrix conditioning for triangular and tetrahedral finite element meshes have been studied [5]. Quality metrics which determine the relevant fitness of elements for the purposes of interpolation or for creating a global stiffness matrix with a low condition number have been developed [5].

Further mathematical research has been performed at the intersection of finite element meshes and linear solvers. A mesh and solver co-adaptation strategy for anisotropic problems has been developed [6]. In addition, relationships between the spectral condition number of the stiffness matrix and mesh geometry for second-order elliptic problems for general finite element spaces defined on simplicial meshes have been determined [7].

Several computational studies have been performed which examined the connections between finite element meshes and linear solvers in various contexts. For example, the effect of unstructured meshes on the preconditioned conjugate gradient solver performance for the solution of the Laplace and Poisson equations has been examined [8,9]. In [10], the relative performance of multigrid methods for unstructured meshes was studied on fluid flow and radiation diffusion problems. Trade-offs associated with the cost of mesh improvement in terms of solution efficiency has been examined for problems in computational fluid dynamics [11,12]. Composite linear solvers which provide better average performance and reliability than single linear solvers for large-scale, nonlinear PDEs have been designed [13].

In this paper, we examine the connections between geometry, mesh smoothing, and linear solver convergence for elliptic PDEs via an engineering approach. In particular, we seek answers to the following questions pertaining to the solution of an elliptic PDE on a given geometric domain. What

Jibum Kim
The Pennsylvania State University
University Park, PA 16802
E-mail: jzk164@cse.psu.edu

Shankar Prasad Sastry
The Pennsylvania State University
University Park, PA 16802
E-mail: sps210@cse.psu.edu

Suzanne M. Shontz
[*]Corresponding author
The Pennsylvania State University
University Park, PA 16802
E-mail: shontz@cse.psu.edu

is the most efficient combination of mesh quality metric, preconditioner, and solver for solving each PDE problem? Which combinations are most and least sensitive to vertex perturbation? What is the effect of increasing the problem size on the number of iterations required to converge and on the solver time? Our goal is to determine the best combination of quality metric, preconditioner, and linear solver which results in a small condition number of the preconditioned matrix and fast solver convergence for a given PDE, geometric domain, and initial mesh.

To answer the above questions, we use Mesquite [14], a mesh quality improvement toolkit, and PETSc [15], a linear solver toolkit, to perform a numerical study investigating the performance of several mesh quality metrics, preconditioners, and sparse linear solvers on the solution of various elliptic PDEs (i.e., Poisson's equation, general second-order elliptic PDEs, and linear elasticity) of interest. Mesh quality metrics used in this study are as follows: inverse mean ratio (IMR) [16], radius ratio (RR) [5], a conditioning-based scale-invariant metric (SI) [5], and an interpolation-based size-and-shape metric (SS) [5]. Furthermore, we investigate the performance of the following preconditioners: Jacobi [17], symmetric successive over relaxation (SSOR) [17], incomplete LU (ILU) [17] and algebraic multigrid [18]; and linear solvers: conjugate gradient (CG) [19], minimum residual solver (MINRES) [20], generalized minimal residual solver (GMRES) [21], and bi-conjugate gradient stabilized solver (Bi-CGSTAB) [22]. The quality metric/preconditioner/linear solver combinations are compared on the basis of efficiency, robustness, and complexity in solving several elliptic PDEs on realistic unstructured tetrahedral finite element meshes. We use Mesquite and PETSc in their native state with the default parameters. Only these two toolkits are employed so that differences in solver implementations, data structures, and other such factors would not influence the results.

The rest of this paper is organized as follows. In Section 2, we describe the three elliptic PDEs we are solving. We describe the quality metrics we use for optimizing the mesh in Section 3. In Section 4, the mesh optimization procedure is described. In Sections 5 and 6, we discuss the iterative linear solvers and preconditioners used in our numerical experiments. The results from our numerical experiments are explained in Section 7. We present our conclusions and plans for future work in Section 8.

2 Finite Element Solution of Elliptic PDEs

In this paper, we consider the solution of three elliptic PDE problems: Poisson's equation, general second-order elliptic PDE problems, and linear elasticity. On a given geometric domain, we consider only homogeneous Dirichlet boundary conditions, since we observed in [23] that modifying the boundary conditions does not alter the efficiency ranking of the solver time for combinations of mesh quality metrics, preconditioners, and linear solvers. To solve the elliptic PDE problems, triangular meshes are used to discretize the domain, Ω. The standard Galerkin finite element (FE) method [24] is used to solve the given PDE problem resulting in the linear system

$$A\xi = b. \tag{1}$$

The approximate solution, ξ, of the analytical solution, u, can be computed by solving (1). In general, (1) is a sparse linear system, and iterative methods such as CG and GMRES are often used to solve the system.

Poisson's equation. Poisson's equation is used to model many mechanical and electromagnetic problems. Poisson's equation is given by

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f \text{ on } \Omega, \tag{2}$$

where f is a given function. For Poisson's equation, the matrix A is given by A=K+N, where K is the stiffness matrix and N is a matrix containing boundary information.

General second-order elliptic PDE problem. The general second-order elliptic PDE problem on Ω is defined as

$$-\alpha\frac{\partial^2 u}{\partial x^2} - \beta\frac{\partial^2 u}{\partial y^2} + au = f \text{ on } \Omega, \tag{3}$$

where α and β are PDE coefficients and a and f are given functions. If a = 0, (3) reduces to Poisson's equation. We consider the case when a is nonzero. The coefficients α and β form a coefficient matrix, C, given by

$$C = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}.$$

For example, C is the identity matrix, I, for (2). The major difference between (3) and Poisson's equation in (2) is the existence of a mass matrix, M, in the decomposition for matrix A. For this problem, the matrix A is given by A=K+M+N, where K is the stiffness matrix, M is the mass matrix, and N is a matrix containing boundary information. One application of the general second-order elliptic PDE problem is the scattering problem.

Linear elasticity. Linear elasticity is a common problem in structural mechanics and is used to compute the displacement vector, u, given a body force, f, and (or) a traction load. In this paper, we do not consider a traction load. The linear elasticity problem with the Lamé parameters λ and μ is defined as

$$\begin{cases} -\bigtriangledown \cdot \tau = f \\ \tau = \lambda \left(\alpha \frac{\partial u}{\partial x} + \beta \frac{\partial u}{\partial y} \right) I + 2\mu\varepsilon(u), \end{cases} \tag{4}$$

where α and β are PDE coefficients, and τ and $\varepsilon(u)$ are the stress and strain tensors, respectively. The strain tensor is given by

$$\varepsilon_{ij}(u) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), i, j = 1, 2.$$

For homogeneous materials, the Lamé parameters are given by λ=$E\nu/((1+\nu)(1-2\nu))$ and μ=$E/(2(1+\nu))$, where E and ν are the elastic modulus and Poisson's ratio, respectively [25].

Elliptic PDE coefficients and optimal mesh element shape. The efficiency of elliptic PDE solvers is highly connected with the condition number, κ, of the matrix A. A smaller $\kappa(A)$ results in a faster convergence time for solving (1). The optimal element shape that reduces $\kappa(A)$ depends on the coefficients of the elliptic PDEs [5,7,26]. It has been shown that for $C = \gamma I$ over Ω, where γ is a constant, the optimal shape of a triangular element is an equilateral triangle [5,7,26]. If C is a constant, but is not γI on Ω, the optimal shape is not an equilateral triangle, and anisotropic triangular elements can be used to obtain a smaller $\kappa(A)$ [5, 7]. If C is non-constant over Ω, the optimal shape of a triangular element varies over Ω [26]. In this paper, we consider elliptic PDEs with continuous, isotropic coefficients which are constant over Ω (i.e., $C = \gamma I$). Therefore, for the PDEs considered in this paper, the ideal triangular element shape is an equilateral triangle.

3 Mesh Quality Metrics

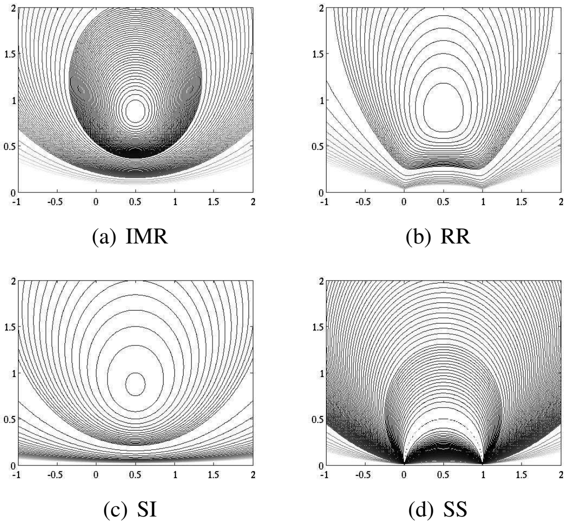
Table 1 provides the notation used to define the following mesh quality metrics: inverse mean ratio (IMR) [16], radius ratio (RR) [5], a conditioning-based scale-invariant metric (SI) [5], and an interpolation-based size-and-shape metric (SS) [5]. Table 2 defines IMR, RR, SI, and SS. These four quality metrics were chosen based on their geometric features, which result in varying contour plots as shown in Fig 1. The plots show the contour lines of the quality of a triangle as a function of a free vertex when the two other vertices held fixed at $(0,0)$ and $(0,1)$. Darker contour lines indicate better quality triangles.

Notation in	Definition
a, b, and c	Position vectors for vertices in a triangular element
$C = [b-a;\ c-a]$	Jacobian of a triangular element
$W = \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix}$	Incidence matrix for an equilateral triangle
Area	Area of a triangular element
r_{circ}	Radius of a triangular element's circumscribing circle
r_{in}	Radius of a triangular element's inscribing circle
s_1, s_2, and s_3	Edge lengths of a triangular element

Table 1 Notation used in the definition of the 2D mesh quality metrics in Table 2.

Quality Metric	Formula
Inverse mean ratio (IMR)	$\frac{\lVert CW^{-1} \rVert_F^2}{2\lvert \det(CW^{-1}) \rvert}$ [16]
Radius ratio (RR)	$\sqrt{r_{\text{circ}}/r_{\text{in}}}$ [5]
Conditioning-based scale-invariant (SI)	Area/$(s_1^2 + s_2^2 + s_3^2)$ [5]
Interpolation-based size-and-shape (SS)	Area/$(s_1 s_2 s_3)$ [5]

Table 2 The mesh quality metric definitions.



(a) IMR (b) RR

(c) SI (d) SS

Fig. 1 Contour plots of the quality metric of a triangle as a function of a free vertex when the two other vertices held fixed at $(0,0)$ and $(0,1)$.

IMR is one of the most well-known mesh quality metrics for mesh quality improvement and is evaluated using the position vectors in the element. RR is a quality metric which is computed using the radius of the triangular element's circumscribing and inscribing circles. SI is a quality metric based on stiffness matrix conditioning, whereas SS is a quality metric based on interpolation error bounds. Note that a conditioning-based size-and-shape quality metric is not defined for 2D cases (triangles) [5].

4 Mesh Optimization

We denote the elements of a mesh and the number of mesh elements by E and $|E|$, respectively. The overall quality of the mesh, Q, is a function of the individual element qualities, q_i, where q_i is the quality of the i^{th} element in the mesh. The mesh quality depends on the choice of q_i (Section 3) and the objective function used to combine the individual element qualities.

The value of the IMR quality metric lies between 1 and ∞, whereas the RR quality metric's value lies between 0.5 and ∞. For IMR and RR, a lower value indicates a higher quality element. For the IMR and RR quality metrics, we define the overall mesh quality, Q, as the sum of squares of the individual element qualities:

$$Q = \sum_{i=1}^{|E|} q_i^2. \tag{5}$$

The SI quality metric's value lies between 0 and $1/(4\sqrt{3})$, whereas the SS quality metric's value lies between 0 and ∞. For SI and SS, a higher value indicates a higher quality element. For the SI and SS quality metrics, we define the overall mesh quality, Q, as the sum of the squares of the reciprocal of the individual element qualities:

$$Q = \sum_{i=1}^{|E|} \frac{1}{q_i^2}. \tag{6}$$

We use the reciprocal of the quality metric instead of the additive inverse to optimize the mesh because the optimization process for the latter results in meshes with elements of both very good and very poor quality, which is not desirable. Using the reciprocal (as was done in [27]) results in smoothed meshes with a more even distribution of element qualities.

We use the Fletcher-Reeves nonlinear conjugate gradient method [28] in Mesquite [14] to minimize Q (as defined by either (5) or (6)). The local mesh smoothing technique is used for the mesh optimization procedure. Mesquite employs a line search version of the nonlinear conjugate gradient method. The implementation ensures that the triangular elements do not tangle as a result of vertex movement.

Our preliminary experiments indicate that the total time, which is the sum of the smoothing time and the solver time, is minimal when the reduction in the objective function is 95% of the possible reduction (when the mesh is completely smoothed). The smoothing time significantly dominates the solver time when the mesh is completely smoothed. Thus, we perform inaccurate mesh smoothing, i.e., we smoothe the mesh only to the extent required to yield the optimal total time for numerically solving the PDE on the mesh (as in [29]).

5 Iterative Linear Solvers

Four iterative Krylov subspace methods are employed to solve the preconditioned linear system. The conjugate gradient (CG) solver [19] is a well-known iterative method for solving systems with symmetric positive definite matrices. It produces a sequence of orthogonal vectors on successive iterations. Let P be the preconditioner (to be described in Section 6). The convergence rate of the preconditioned CG method depends upon the condition number, κ, of $P^{-1}A$. In the 2-norm, κ can be approximated by

$$\begin{aligned}\kappa_2(P^{-1}A) &= \left\|P^{-1}A\right\|_2 \left\|P^{-1}A\right\|_2^{-1} \\ &\approx \lambda_{max}(P^{-1}A)/\lambda_{min}(P^{-1}A),\end{aligned}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of $P^{-1}A$, respectively. The fastest convergence occurs when eigenvalues are clustered around the same non-null value, and hence, κ is near 1. Some theoretical convergence results for the CG solver are given in [12,30]. In particular, if the eigenvalues of A are uniformly distributed, the number of iterations required to converge, n, to reduce the error in the energy norm by a factor of ε satisfies the following inequality [12]:

$$n \leq \left\lfloor \sqrt{\kappa}\, \log \frac{2}{\varepsilon} \right\rfloor + 1.$$

The minimal residual (MINRES) algorithm [20] solves linear systems with symmetric indefinite matrices. It generates a sequence of orthogonal vectors and attempts to minimize the residual in each iteration. Similar to CG, a small condition number for $P^{-1}A$ and clustering of eigenvalues around the same non-null value results in fast convergence.

For nonsymmetric matrices, the generalized minimal residual (GMRES) method [21] is one of the most widely used iterative solvers. Similar to CG and MINRES, GMRES computes orthogonal vectors on each iteration; however, the entire sequence needs to be stored. Therefore, the version of GMRES which restarts GMRES every m steps, i.e., GMRES-(m), is used in practice. It is known that a large value of m is effective in decreasing the number of iterations required to converge; however, the optimal value of m depends upon the problem [31].

The biconjugate gradient stabilized (Bi-CGSTAB) method [22] is a biorthogonalization technique, which generates two sets of biorthogonal vectors instead of producing long orthogonal vectors. Bi-CGSTAB is known to have comparable or even faster convergence than other biorthogonalization methods such as the conjugate gradient squared method. However, Bi-CGSTAB sometimes shows an irregular convergence rate similar to other biorthogonalization methods [17]. Bi-CGSTAB and GMRES are the most widely-used iterative methods for solving systems based on nonsymmetric matrices.

6 Preconditioners

The objective of introducing a preconditioner, P, into the solution of a linear system is to make the system easier to solve, whereby reducing the convergence time of the iterative solver. The reader is referred to [32] (and the references therein) for further information on iterative solvers and preconditioners. In this paper, four preconditioners are employed. The first is the Jacobi preconditioner, which is simply the diagonal of A.

The second is the symmetric successive over relaxation (SSOR) preconditioner. SSOR is similar to Jacobi but decomposes A into L (the strictly lower triangular part), D (the diagonal), and U (the strictly upper triangular part), i.e., A=L+D+U. The SSOR preconditioner is given by

$$P = (D - \omega L)D^{-1}(D - \omega U),$$

where ω represents the relaxation coefficient. The default ω value in PETSc is 1.

The incomplete LU (ILU) preconditioner with level zero fill-in (ILU(0)) is the third preconditioner. ILU is commonly used in the solution of elliptic PDE problems. The basic idea of the ILU preconditioner is to determine lower ($\tilde{L}$) and upper triangular ($\tilde{U}$) matrices such that the matrix $\tilde{L}\tilde{U}$-A satisfies certain constraints [32]. The ILU preconditioner works well for many problems but fails when it encounters negative or zero pivots.

The fourth is the algebraic multigrid (AMG) preconditioner. Different from geometric multigrid methods, the AMG preconditioner does not need any mesh information to generate the preconditioner and hence is known as blackbox technique. AMG only requires the matrix A to generate the preconditioner. Therefore, the AMG preconditioner can be used to solve linear systems which arise from unstructured meshes. The main idea of the AMG preconditioner is to eliminate the smooth error using restriction and interpolation which is not removed by relaxation on the fine grid [18]. In this paper, we use the default options for HYPRE BoomerAMG in PETSc. The default option in PETSc employs 25 levels of V-cycles. For further infomation on HYPRE BoomerAMG, the reader is referred to [15].

7 Numerical Experiments

7.1 Experimental Setup

We consider the following questions which we investigate on the three elliptic PDE problems shown in Table 3. What is the most efficient combination of mesh quality metric, preconditioner, and solver for solving each PDE problem? Which combinations are most and least sensitive to vertex perturbation? What is the effect of increasing the problem size on the number of iterations required to converge? Table 3 summarizes the experiments and corresponding PDE problems to be solved. For all three PDE problems in Table 3, we consider only a simple homogeneous Dirichlet boundary condition with u=0 on the boundary because we already observed that modifying boundary condition does not affect the efficiency ranking [23]. For the elasticity problem in Table 3, E and ν are set to 1 and 0.3, respectively. The body force, f, is set to $[1\ 0]^T$. The machine employed for this study is equipped with an Intel Nehalem processor (2.66 GHz) and 24GB of RAM [33].

In this paper, we consider elliptic PDEs with isotropic coefficients, which is constant (i.e., $C = \alpha I$) over 2D geometric domains. The optimal element shapes for solving elliptic PDEs with anisotropic coefficients are different from the optimal shapes for elliptic PDEs with isotropic coefficients [5,7,26]. Therefore, our experimental results cannot be generalized to the solution of elliptic PDEs with anisotropic coefficients. The quality metrics used to determine the mesh element quality are different for 2D and 3D elements. In addition, the sparsity patterns of the A matrices in the linear systems obtained from unstructured meshes are different for 2D and 3D meshes. Therefore, our experimental results cannot be generalized to 3D meshes.

Geometric Domains. The two 2D geometric domains, wrench and hinge, considered in our experiments are shown in Figure 2. Triangle [34] is used to generate initial meshes on the two domains. Half the interior vertices in each mesh are perturbed to create test meshes that are further from optimal. Properties of the test meshes and the corresponding finite element matrices are shown in Table 4.

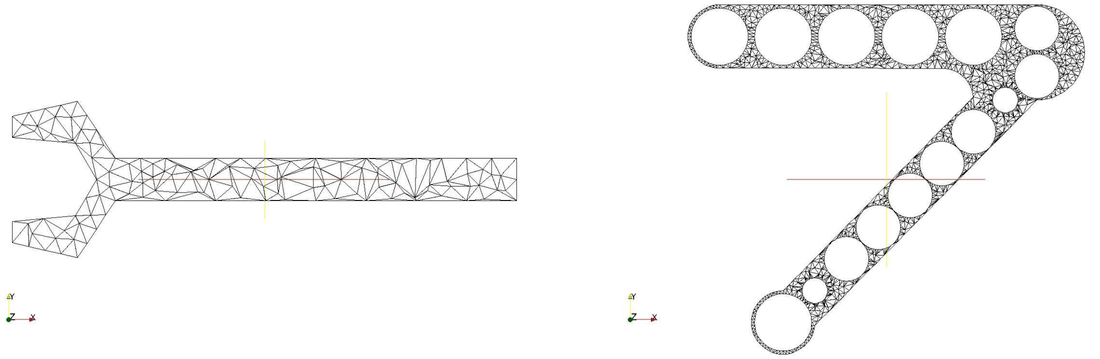
Finite Element Solution. The FE method described in Section 2 is used to discretize the domain, Ω, and to generate a linear system of the form $A\xi$=b. PETSc [15] is used to generate the preconditioners, P, and to solve the linear system, $P^{-1}A\xi$=$P^{-1}b$. We employ the solvers and preconditioners described in Sections 5 and 6, respectively, to solve the linear system. Table 5 enumerates the 16 preconditioner-solver combinations used in our experiments. The default parameters for each preconditioner and solver were employed.

The default stopping criteria in PETSc were employed. For example, the absolute tolerance, abstol, and the relative tolerance, rtol, are set to 1e-50 and 1e-05, respectively. The maximum number of iterations for solving the preconditioned linear system is set to 10,000. When the preconditioned linear system is solved, ξ_0 is set to the default value of 0. The preconditioned linear system converges on the i^{th} iteration if the following inequality is satisfied:

$$\|r_i\| < \max(\text{rtol}\,\|r_0\|, \text{abstol}), \qquad (7)$$

Exp. No.	Exp. Name.	Examples of PDE Problems
7.3.1	Determination of restart values of the GMRES solver	(A) $-\Delta u = 1$ on $\Omega, u = 0$ on $\partial\Omega$
7.3.2	Poisson's equation	(A) $-\Delta u = 1$ on $\Omega, u = 0$ on $\partial\Omega$
7.3.3	General second-order elliptic PDE problem	(B) $-\Delta u + 100\,u = 1$ on $\Omega, u = 0$ on $\partial\Omega$
7.3.4	Elasticity problem	(C) $-\nabla \cdot \tau = f$ on $\Omega, u = 0$ on $\partial\Omega$, where $\tau = \lambda(\nabla \cdot u)I + 2\mu\varepsilon(u)$

Table 3 Listing of numerical experiments and examples of PDE problems. The letters (a) through (c) are representative examples of the three types of PDE problems under consideration.



(a) Wrench mesh

(b) Hinge mesh

Fig. 2 Coarse initial meshes on the wrench and hinge geometric domains indicative of the actual meshes to be smoothed.

mesh	# vertices	# elements	mesh	# vertices	# elements
Wrench (10K)	10,142	19,796	Hinge (10K)	9,989	18,359
Wrench (50K)	50,161	99,197	Hinge (50K)	49,986	96,483
Wrench (100K)	100,267	199,055	Hinge (100K)	99,949	195,131
Wrench (200K)	199,981	397,764	Hinge (200K)	200,894	394,743
Wrench (500K)	499,340	995,496	Hinge (500K)	498,978	987,162

Table 4 Properties of meshes on geometric domains. Initial angle distributions for the meshes are given in Table 7.

where r_i is the residual at the i^{th} iteration and r_0 is the initial residual.

Accuracy of the Solution. The exact solutions of the PDE problems in Table 3 on the geometric domains in Figure 2 are unknown. Therefore, we conduct a mesh-independence study using meshes with 10K, 50K, 100K, 200K, and 500K vertices and verify that the angle distribution and resulting PDE solution is independent of the mesh size. Our experimental results show that the finite element method converges to the same PDE solution if the mesh consists of more than 10K vertices. We also observe that the finite element solution is not affected by choice of quality metric, preconditioner, and linear solver for these mesh sizes. We will investigate the accuracy of the finite element solution for boundary value PDEs as future work.

Timing. In our experiments, the total time is defined as the sum of the smoothing time and the solver time. The smoothing time is the time to achieve an accurately smoothed mesh as described in Section 4. The time required to smoothe the test meshes using various quality metrics is shown in Table 6. The solver time is the time the solver takes to satisfy (7) and includes the time to generate P and to solve $P^{-1}A\xi = P^{-1}b$.

For our experiments, we use Mesquite version 1.1.7 and PETSc version 3.0.0. Mesquite and PETSc are widely used for solving linear systems [12, 13, 35] and mesh quality improvement [29, 36, 23], respectively. Both PETSc and Mesquite are numerical libraries containing data structures and routines. Algorithms in each software package are coded using similar data structures and routines. We report both the timing (in seconds) and the number of iterations required to converge. Note the latter is not affected by the software version or the hardware used to solve the problem. However, the timing can be affected by these factors. We expect that the relative difference amongst different combinations of quality metric, preconditioner, and linear solvers will remain the same.

Preconditioner	**Solver**			
	CG	**GMRES**	**MINRES**	**BI-CGSTAB**
Jacobi	1	2	3	4
SSOR	5	6	7	8
ILU(0)	9	10	11	12
AMG	13	14	15	16

Table 5 The sixteen combinations of preconditioners and solvers. For example, 10 refers to using the ILU(0) preconditioner with the GMRES solver.

Mesh	Mesh quality metrics			
	IMR	RR	SI	SS
Wrench (500K)	55	515	501	605
Hinge (500K)	55	526	516	633

Table 6 Mesh smoothing time (sec) for various mesh quality metrics

7.2 Description of Experiments

Experiment 1: Best combination of mesh quality metric, preconditioner and solver. In this experiment, we seek to determine the most efficient combination of mesh quality metric, preconditioner, and solver for each type of PDE problem in Table 3. We will examine the most efficient combinations for both the solver time and the total time.

Experiment 2: Effect of perturbation. We discuss the effect of varying the percentage of vertex perturbation on the efficiency of the preconditioner-solver combinations. In [12], it is reported that the CG solver is more robust to perturbation than is the GMRES solver for the Jacobi preconditioner. We determine the robustness for various combinations of preconditioners and solvers to such perturbations. We randomly perturb a certain number of vertices in each mesh while ensuring that no perturbation results in a very poorly-shaped element (e.g., an inverted element). We perturb the interior vertices such that they move less than half the distance at which element inversion would occur. In our experiments, we perturb 5%, 10%, 25%, and 50% of the elements and investigate the robustness of the preconditioner-solver combinations to the vertex perturbations.

We define the relative increase, i.e., the percentage increase (PI), in the solver time due to the vertex perturbation as follows:

$$PI = \frac{T_{max} - T_{min}}{T_{min}},$$

where T_{max} and T_{min} are the maximum and minimum convergence time among 0% (i.e., fully-smoothed unperturbed mesh), 5%, 10%, 25%, and 50% perturbed elements, respectively. Note that we observe monotonic convergence behavior with all four mesh quality metrics.

Experiment 3: Increasing the problem size. In this experiment, we examine the convergence rates of various preconditioner-solver combinations with an increasing number of vertices in a mesh for a given domain. We execute our numerical experiments for meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain. Let N be the number of vertices in the mesh. We compute the order of convergence, denoted $O(N^s)$, using the linear least-squares method for both the number of iterations to converge and the solver time. Thus, the total work required to solve the system is $O(N^{s+1})$ for each combination of preconditioner and solver.

7.3 Numerical Experiments

We now discuss the results for the three numerical experiments discussed in Section 7.1 for each PDE problem in Table 3.

7.3.1 Preliminary experiment for determination of restart value of the GMRES solver

The default restart value, m, in PETSc is 30, and we examine whether that value is optimal to solve each of the PDE problems. We experimentally determine the optimal m value, which yields the fastest solver time, for each application by varying it as follows: 10, 30, 50, 100, and 200.

Figure 3(a) shows the number of iterations required to converge for different m values as the problem size increases for the wrench domain. For this experiment, the meshes are smoothed with the IMR quality metric, and the SSOR preconditioner with the GMRES solver is employed to solve Poisson's equation (problem (A) in Table 3). The results obtained are representative of the results obtained with other geometric domains, quality metrics, and preconditioners. We observe that as we increase the values of m, the total number of iterations to converge reduces. This is consistent with the theoretical analysis presented in [31].

Figure 3(b) shows the solver time for different m values as we increase the problem size. We observe that the solver time is least when the restart value, m, is 30. As we reduce or increase the restart value, the time taken to converge to a solution increases. Large restart values (e.g., $m = 100$ and 200) result in slower solver times because the cost of orthogonalization increases with an increase in m. Small restart values (e.g., $m = 10$) also slow the solver since it needs a greater number of iterations to converge.

For the other types of PDE problems in Table 3, we observe similar results. The GMRES solver with $m = 30$ yields the least solver time, although $m = 200$ results in the fewest number of iterations to converge. Therefore, we choose the GMRES solver with $m = 30$ for all elliptic PDE problems in Table 3.

7.3.2 Numerical Results for Poisson's Equation

Exp. 1: Determination of the best combinations of mesh quality metric, preconditioner, and solver for Poisson's equation. Table 8 shows the solver time and the number of iterations required for convergence as a function of mesh quality metrics for different combinations of preconditioners and solvers. The most efficient combinations (e.g., the IMR mesh quality metric with the AMG preconditioner and the CG solver) are 97% faster than the least efficient combinations (e.g., the IMR quality metric with the Jacobi preconditioner and the GMRES solver).

We observe that the AMG preconditioner with any choice of quality metric and solver is faster to converge than other combinations of quality metric, preconditioner and solver. When we solve linear equations derived from a mesh, high frequency terms are not eliminated only by relaxation on the fine grid. Thus, the AMG preconditioner used in this paper employs 25 level of V-cycles and efficiently eliminates high frequency terms (which correspond to large eigenvalues in the matrix A) using coarse grid correction techniques. The most time-consuming step during the generation of the AMG preconditioner is an aggregation step which finds the vertex-neighborhood information from the matrix A. However, the aggregation step is fast for Poisson's equation as it contains a single degree of freedom. Note that the number of iterations required to converge for combinations with the AMG preconditioner represent the number of outer iterations only (not including V-cycles).

We also observe that the Bi-CGSTAB solver (combinations 4, 8, 12, and 16 in Table 5) is most sensitive to the choice of different mesh quality metric, whereas, the CG solver is least affected by the choice of the quality metric. These results are related with the irregular convergence behavior discussed in [23]. The number of iterations required to converge and the solver time taken by the BiCG-STAB solver with a given preconditioner varies by more than 20% in many cases. For other combinations, however, the variation is small and is restricted to less than 10% in most cases.

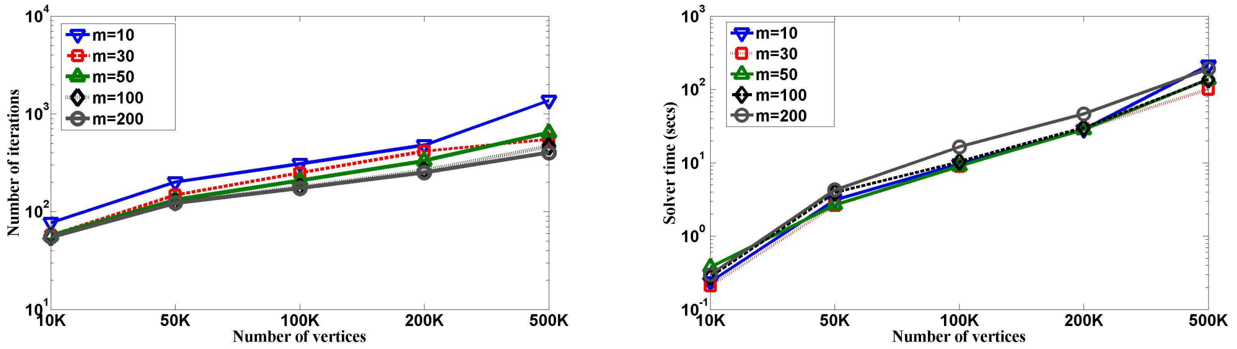
On the choice of mesh quality metrics, the SS metric is the least efficient mesh quality metrics in most cases. The method using the SS metric often fails to converge because meshes smoothed by the SS metric fail to yield a positive definite preconditioner. Note that both the CG and the MINRES solvers require the preconditioner to be a positive definite matrix. In most cases, the maximum eigenvalue of the matrix A is larger and the minimum eigenvalue of the matrix A is smaller than the eigenvalues of the corresponding matrix A for the other quality metrics. These eigenvalues are connected to the shape (angle) of the elements in the mesh.

Table 7 shows the angle (θ) distributions for elements in the initial mesh (i.e., with 50% of the interior vertices perturbed) and different quality metrics on the wrench (500K) and hinge (500k) domains. ***We observe that all four quality metrics try to generate nearly equilateral elements, but the SS metric yields meshes containing elements with more small and large angles than do the other quality metrics***. This is because the SS metric penalizes small angles less than do other quality metrics [5]. For elliptic PDEs with isotropic coefficients, it is well-known that equilateral triangles are desirable for efficiency, whereas small angles have a bad effect on the condition number of matrix A and on the efficiency [5]. This explains why the SS metric is less efficient for mesh smoothing than other quality metrics. We also observe that the elements with small angles or large angles (i.e., poorly-shaped) occur most often on the boundary of the mesh. Interestingly, among the four quality metrics in Table 7, ***the RR metric has the fewest poorly-shaped elements (even fewer than the IMR).***

In terms of the total time, the IMR metric outperforms the other quality metrics, because numerical computation of the IMR metric for mesh optimization is highly optimized in Mesquite. Numerical computation of the other mesh quality metrics, i.e., RR, SI, and SS, are not as optimized and hence are less efficient to compute. Similar to the solver time, the SS metric is the least efficient quality metric in terms of the total time. The total time for the most efficient combination (i.e., AMG with CG) for the IMR metric is 90% less than it is for SS. The least efficient combination (i.e., Jacobi with GMRES) for the IMR metric is 45% faster than that of the least efficient quality metric, i.e., SI.

Exp. 2: Effect of perturbation for Poisson's equation. Table 9 shows the linear solver time and the number of iterations required to converge as a function of the amount of vertex perturbation on the wrench and hinge meshes, respectively. Figures 4(a) and 4(b) show the PI for different combinations of preconditioners and solvers when the number of perturbed vertices is increased. For these experiments, the meshes are smoothed with the IMR quality metric. The results obtained are representative of the results obtained with other quality metrics.

More preconditioner-solver combinations are able to solve the PDEs on the wrench domain than on the hinge domain. The PDE on the hinge domain is harder to solve because there are many holes in the domain and the holes cre-



(a) Number of iterations

(b) Solver time

Fig. 3 Effect of the mesh size on the number of iterations (a) and solver time (b) to convergence as a function of GMRES restart value, *m*, for Poisson's equation (problem (A) in Table 3) on the wrench domain.

(a) Wrench (500K)

Quality	Angle (θ) distribution					
Metric	$\theta < 20°$	$20° < \theta < 40°$	$40° < \theta < 60°$	$60° < \theta < 80°$	$80° < \theta < 100°$	$\theta > 100°$
Initial mesh	0.768	15.64	38.30	28.83	12.03	4.43
IMR	0.008	1.60	52.27	42.40	3.47	0.25
RR	0.006	0.96	52.62	43.85	2.49	0.07
SI	0.005	0.70	53.68	42.39	3.08	0.09
SS	0.027	2.08	52.67	39.47	5.70	0.25

(b) Hinge (500K)

Quality	Angle (θ) distribution					
Metric	$\theta < 20°$	$20° < \theta < 40°$	$40° < \theta < 60°$	$60° < \theta < 80°$	$80° < \theta < 100°$	$\theta > 100°$
Initial mesh	0.950	15.48	38.79	28.39	11.85	4.54
IMR	0.021	2.05	52.23	41.55	3.76	0.39
RR	0.019	1.44	51.67	43.02	2.82	0.19
SI	0.016	1.17	52.85	41.52	3.38	0.23
SS	0.084	2.44	51.73	38.67	5.81	0.43

Table 7 Angle (θ) distribution for various mesh quality metrics. The reported values indicate a percentage of angles in the mesh. The initial mesh is the mesh with 50% of the interior vertices perturbed.

ate thin areas near the boundary. The vertices are highly constrained in these areas and perturbation produces poorly-shaped elements in many cases.

Table 9 shows that the CG and MINRES solvers fail to converge in many cases, whereas the GMRES and Bi-CGSTAB solvers do not. ***The CG and the MINRES solver often fail to converge because they fail to generate a positive definite preconditioner for two reasons.*** First, generation of the preconditioners fails when there are poorly-shaped elements in the mesh and when the minimum eigenvalue of the matrix A is too small relative to its maximum eigenvalue. Poorly-shaped elements represent the elements which have large (e.g., angle $> 100°$) or small angles (e.g., angle $< 20°$). This was also observed in the above experiment for the SS metric. Secondly, preconditioners for the CG and MINRES solvers fail when the element lengths are very small. In these cases, the minimum eigenvalue of the matrix A is very small and corresponds to a large condition number. The minimum eigenvalue of the matrix A depends on the edge lengths in the mesh. For these reasons, when the element lengths are very small or when the meshes include lots of poorly-shaped elements, we see that GMRES and Bi-CGSTAB are more robust than CG and MINRES are to vertex perturbation.

We also observe the following rank-ordering of preconditioners for robustness to vertex perturbation: ILU(0) $>$ SSOR $\approx$ AMG $>$ Jacobi. The ranking is in order of most robust to least robust. Note that the solver time for the AMG preconditioner is less than 10 seconds, whereas it is greater than 40 seconds for the other preconditioners. Hence, the PI values are sensitive to small changes of the solver time.

Exp. 3: Increasing the problem size for Poisson's equation. We observe that the maximum eigenvalues of $P^{-1}A$ stay constant, but the minimum eigenvalue of $P^{-1}A$ rapidly decreases as N increases. This is consistent with the results observed in [5]. In [5], it was discussed that the minimum eigenvalue in matrix A is a function of the edge lengths in the mesh. It was also discussed in Section 5 that clustering

(a) Wrench (500K)

Quality	Combinations of preconditioners and solvers															
Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IMR	132.7	547.3	85.8	90.2	56.3	104.3	56.4	58.2	49.9	83.6	46.7	43.4	3.6	3.9	4.0	3.9
	983	3704	851	680	427	550	390	296	379	487	349	228	5	5	6	3
RR	133.3	466.6	85.3	78.9	57.9	102.4	60.6	52.5	48.8	83.1	46.8	43.3	4.3	3.7	4.4	3.9
	976	3455	845	640	426	547	389	262	378	481	348	230	5	5	6	3
SI	136.3	471.9	87.2	80.1	58.2	100	60.5	60	50	85.2	48.2	52.6	4.1	3.6	3.9	3.6
	976	3301	844	622	426	547	389	278	378	482	348	268	5	5	5	3
SS	*	499.4	*	76.7	*	92.4	54.2	57.6	*	87.4	51.3	44.0	3.8	4.1	4.4	4.2
	*	3657	*	616	*	530	394	288	*	483	356	220	5	5	6	3

(b) Hinge (500K)

Quality	Combinations of preconditioners and solvers															
Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IMR	*	324.5	*	49.2	40.6	93.2	40.8	42.2	34.8	74.8	34.1	39.9	3.5	3.4	3.9	4.1
	*	2404	*	402	317	528	299	213	282	445	268	208	5	5	5	3
RR	*	322.2	*	68.3	41.3	92.8	41.0	37.7	34.7	73.1	34.8	35.0	3.4	3.3	3.7	4.0
	*	2360	*	521	316	527	299	192	281	435	267	179	5	5	5	3
SI	*	318.4	*	57.4	40.2	90.3	41.3	43.6	34.6	73.0	33.9	37.3	3.3	3.4	3.7	3.7
	*	2380	*	465	316	527	299	217	281	436	266	197	5	5	5	3
SS	*	380.4	*	63.5	*	87.8	*	43.9	35.6	71.7	36.9	38.1	4.4	4.0	3.6	3.8
	*	2531	*	501	*	517	*	226	287	453	273	195	5	5	5	3

Table 8 Linear solver time (secs) and number of iterations required to converge for Poisson's equation (problem (A) in Table 3) as a function of mesh quality metric for the 16 preconditioner-solver combinations (see Table 5) on the wrench and hinge domains. A '*' denotes failure. For each quality metric, the numbers in the top and bottom rows represent the linear solver time and number of iterations to convergence, respectively.

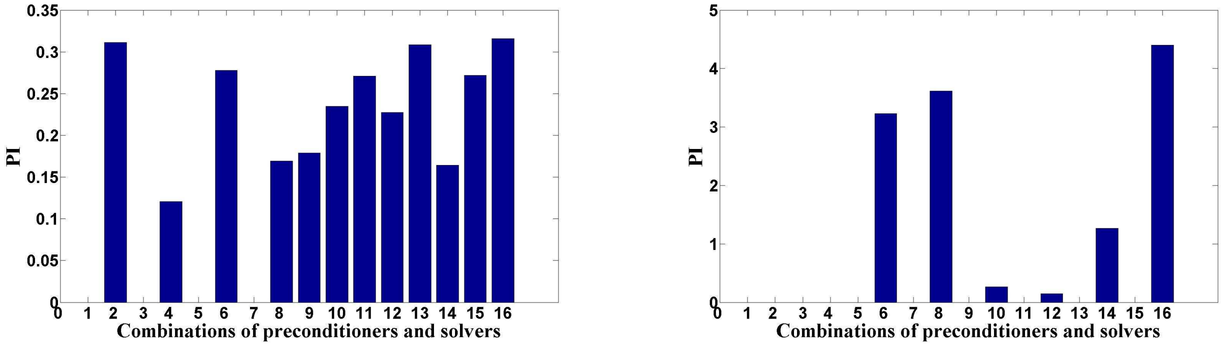
(a) Wrench (500K)

Percent Vertices	Combinations of preconditioners and solvers															
Pert.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0%	132.7	547.3	85.8	90.2	56.3	104.3	56.4	58.2	50.0	83.6	46.7	43.4	3.6	3.9	4.0	3.9
	983	3704	851	680	427	550	390	296	379	487	349	228	5	5	6	3
5%	*	420.4	*	101.1	*	113.2	*	53.4	51.6	93.7	55.6	53.3	4.6	4.2	4.6	5.2
	*	2981	*	738	*	647	*	255	403	492	371	264	5	5	6	3
10%	*	443.O	*	91.9	*	115.5	58.7	55.3	55.2	86.9	53.2	51.7	4.7	4.1	4.7	4.4
	*	3152	*	701	*	649	418	255	404	492	371	262	5	5	6	3
25%	*	435.1	*	93.3	*	116.6	*	62.4	54.0	92.6	52.4	45.2	4.4	4.5	5.2	4.7
	*	2930	*	724	*	670	*	307	411	528	378	237	5	5	6	3
50%	*	551.2	*	95.6	*	133.3	*	58.9	58.9	103.2	59.3	49.8	4.4	4.4	4.7	5.0
	*	3869	*	809	*	713	*	294	423	570	389	261	5	5	6	3

(b) Hinge (500K)

Percent Vertices	Combinations of preconditioners and solvers															
Pert.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0%	*	324.5	*	49.2	40.6	93.2	40.8	42.2	34.8	74.8	34.1	39.9	3.5	3.4	3.9	4.1
		2404	*	402	317	528	299	213	282	445	268	208	5	5	5	3
5%	*	*	*	661.8	*	259.4	*	184.6	*	84.2	*	42.6	*	7.3	*	20.0
	*	*	*	5432	*	1500	*	952	*	505	*	234	*	15	*	27
10%	*	*	*	657.5	*	268.4	*	160.4	*	90.6	*	39.4	*	7.8	14.0	20.2
	*	*	*	4988	*	1577	*	844	*	543	*	208	*	14	29	31
25%	*	*	*	54.0	*	394.0	*	195.1	*	90.0	*	44.4	*	7.6	11.6	21.7
	*	*	*	4122	*	2412	*	970	*	493	*	230	*	13	27	32
50%	*	*	*	*	*	297.5	*	154.0	*	94.4	*	45.2	*	7.5	*	22.1
	*	*	*	*	*	1676	*	740	*	535	*	237	*	15	*	34

Table 9 Linear solver time (secs) and number of iterations required to converge for Poisson's equation (problem (A) in Table 3) as a function of vertex perturbation for the 16 preconditioner-solver combinations (see Table 5) on the two geometric domains. A '*' denotes failure. For each percentage of vertices perturbed, the numbers in the top and bottom rows represent the linear solver time and number of iterations to convergence, respectively.



(a) Wrench (500K)

(b) Hinge (500K)

Fig. 4 Percentage increase (PI) as a function of the solver time for different combinations of preconditioners and solvers for Poisson's equation (problem (A) in Table 3). Preconditioner-solver combinations which fail to generate a preconditioner or do not converge correspond to the missing bars in these figures. Note that PI values for the hinge domain are significantly greater than those for the wrench domain.

of eigenvalues around the same non-null value results in a faster convergence rate. Thus, as the minimum eigenvalues of $P^{-1}A$ decrease, the eigenvalue spectrum becomes larger, and the number of iterations required to converge increases.

Figure 5(a) shows the value of s in $O(N^s)$ for the number of iterations required to converge for different combinations of preconditioners and solvers on the wrench domain. In this experiment, we use meshes smoothed by employing the IMR quality metric. The results are similar for other quality metrics. ***The AMG preconditioner (i.e., combinations* $13-16$ *in Table 5) with any solver yields values of* s *less than*** 0.1**.** The other combinations of preconditioners and solvers yield s values around 0.5. The Jacobi-GMRES combination has the largest value with s approximately 0.8. Figure 5(b) shows the same results on the hinge domain. Figures 6(a) and 6(b) show the order of convergence of the solver time for the wrench and the hinge domain, respectively.

The best combination of quality metric, preconditioner, and solver is not affected by increasing the problem size. The AMG preconditioner with any solver beats the other combinations. In terms of the quality metric, the RR metric is the most efficient, whereas the SS metric is the least efficient in most cases.

7.3.3 Numerical Results for General Second-order Elliptic PDEs

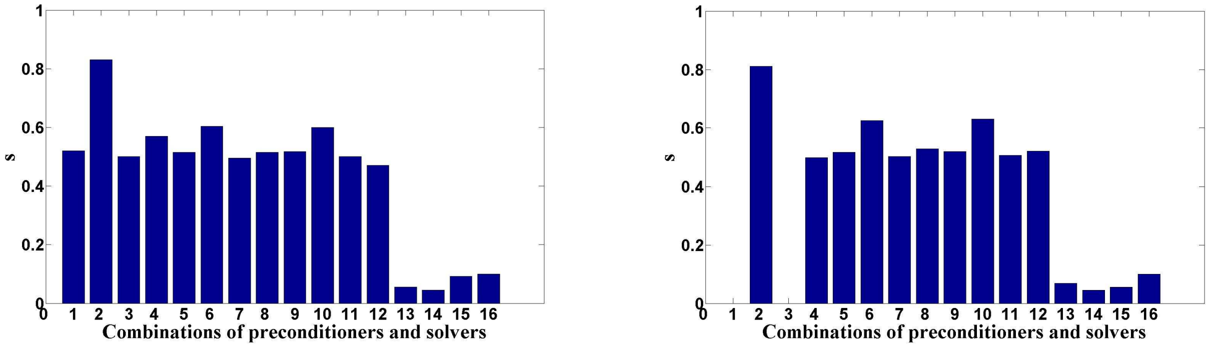
Exp. 1: Determination of the best combinations of mesh quality metric, preconditioner, and solver for general second-oder elliptic PDEs. Table 10 shows the solver time as a function of quality metric for various combinations of preconditioners and solvers for problem (B) in Table 3. We observe that the overall solver time for problem (B) is lower than it is for the other problems. The results show the most efficient combination for the solution of problem (B) is similar to the best combinations to solve problem (A).

The most efficient combination is the AMG preconditioner with any choice of quality metric and solver. Similar to Poisson's equation, general second-order elliptic PDEs have one degree of freedom during the aggregation step for generation of the AMG preconditioner. Therefore, it results in fast solver time. The solver time for the most efficient combination is 95% less than the least efficient combination (e.g., the SI quality metric with the Jacobi preconditioner and the GMRES solver). For the choice of mesh quality metric, RR is more efficient than other quality metrics in most cases, and the SS quality metric is the least efficient quality metric. This result is also related to the angle distribution of mesh elements as discussed before. Similar to Poisson's equation (Problem (A)), the Bi-CGSTAB solver is most sensitive to the choice of different mesh quality metric, whereas, the CG solver is least affected by the choice of quality metric.

We also observe that the use of any preconditioner in this paper significantly mitigates poorly-shaped elements for the SS metric when solving problem (B). This implies that, for this problem, some poorly-shaped mesh elements can be overcome, and the solver time does not increase when the best preconditioner is not chosen.

In terms of the total time, the overall trend is similar to that seen for Poisson's equation (problem (A)). In most cases, we observe the following rank ordering of quality metrics with respect to the total time: IMR > RR > SI > SS. The ranking is in order of fastest to slowest. The most efficient combination is the IMR quality metric and the AMG preconditioner with any choice of the solver. Note that for problem (B), the smoothing time dominates the solver time more when compared with problem (A) because the overall solver time for solving problem (B) takes less time compared with that for problem (A).

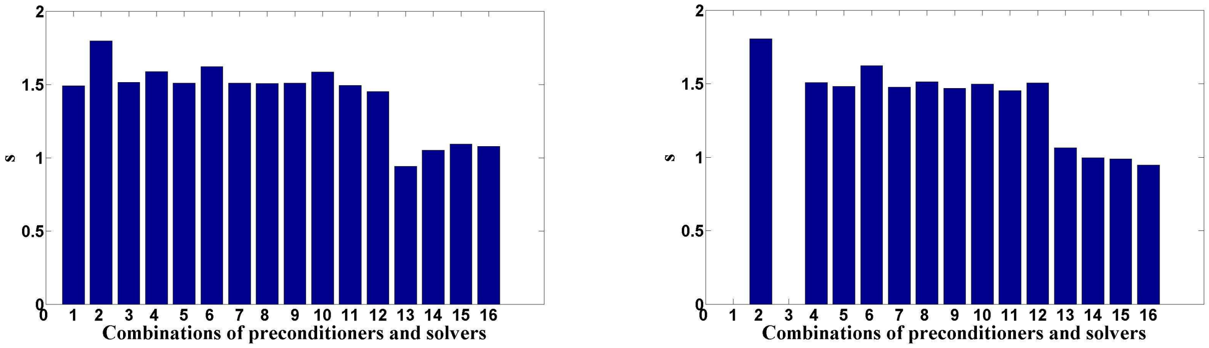
We also conducted experiments in which we modified the PDE coefficients in (3) such that the coefficient matrix



(a) Wrench

(b) Hinge

Fig. 5 The order of convergence for the solver time based on the number of iterations for various combinations of preconditioners and solvers for Poisson's equation (problem (A) in Table 3). Meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain are employed in computing s using the linear least-squares method.



(a) Wrench

(b) Hinge

Fig. 6 Similar to Figure 5, this figure displays the order of convergence based on the solver time for Poisson's equation (problem (A) in Table 3). Meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain are employed in computing s using the linear least-squares method.

$C = \gamma I$, where γ is a constant. We verified that the efficiency rankings are not affected by these modifications. Because they do not affect the optimal triangular shape, and hence $\kappa(A)$ is not affected. Also, the sparsity pattern of A is not changed by these modifications. We also modified the functions, f in (3) and observed consistent efficiency ranking results because modifying f only affects b of the linear system in (1). This is consistent with the theoretical analysis which is explained in Section 2.

In terms of PDE parameter a in (3), we consider a to be a constant and modify its values. We investigate the existence of a value on the efficiency rankings. If a is too large (i.e., a $>$ 100) or too small (i.e., a $<$ -100) compared with f (e.g., $f = 1$), the problem is dominated by the linear term and affects the efficiency rankings. Therefore, we assume -100 $<$ a $<$ 100 and study the presence of the mass matrix on the efficiency. Further discussion on the effect of the presence of a nonzero value of a on efficiency rankings is presented in our previous paper [23].

Exp. 2: Effect of perturbation for general second-oder elliptic PDEs. Figures 7(a) and 7(b) show the changes in the solver time to compute the solution for the perturbed wrench (500K) and hinge meshes (500K), respectively. For this experiment, the meshes are smoothed with the IMR quality metric. The results obtained are representative of the results obtained with other quality metrics. The Jacobi preconditioner is most sensitive to the vertex perturbation. Note that the solver time for the AMG preconditioner is less than 10 seconds, whereas it is greater than 40 seconds for the other preconditioners. Hence the values are more sensitive to small changes.

Similar to Poisson's equation, the SSOR preconditioner with the CG or MINRES solver (combinations 5 and 7 in Table 5) shows the most robust performance with respect to perturbation for both the wrench and hinge domains. A few poor quality elements can increase the values of the maximum eigenvalues and the condition number of the linear system. We observe that both CG and MINRES are able to circumvent the numerical difficulties associated with large eigenvalues [5]. The GMRES solver (GMRES(30)) is most sensitive to the poor elements because it restarts in every 30 iterations without executing the addi-

(a) Wrench (500K)

Quality	Combinations of preconditioners and solvers															
Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IMR	25.7	60.4	27.2	23.1	16.0	24.1	17.1	20.1	13.2	21.8	15.3	17.4	3.4	3.3	3.4	3.4
	284	442	259	181	125	138	119	96	111	123	107	73	4	4	4	2
RR	26.8	59.4	27.1	22.8	16.3	25.4	17.1	17.9	14.0	22.0	15.2	13.8	3.4	3.8	3.4	3.4
	282	434	257	175	124	138	119	86	111	122	106	70	4	4	4	2
SI	26.8	61.1	26.2	25.9	15.9	24.7	17.1	14.4	14.1	21.3	15.2	14.8	3.5	3.5	3.5	3.3
	282	436	257	204	124	138	119	69	111	122	106	73	4	4	4	2
SS	25.7	60.6	28.2	27.1	16.0	24.2	16.5	15.4	14.0	21.8	15.2	15.0	3.5	3.6	3.7	3.4
	294	461	266	205	126	139	120	78	113	127	109	79	4	4	4	2

(b) Hinge (500K)

Quality	Combinations of preconditioners and solvers															
Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IMR	23.5	52.8	23.9	19.6	14.1	22.2	14.8	16.1	13.4	18.6	13.3	13.7	3.4	3.2	3.2	3.0
	261	383	237	159	113	127	108	81	100	110	97	68	4	4	4	2
RR	23.3	51.7	24.2	20.4	13.9	21.6	15.0	13.7	12.5	18.7	13.5	12.6	3.1	3.2	3.2	3.1
	258	376	234	163	112	127	108	68	100	109	96	64	4	4	4	2
SI	24.2	54.5	23.5	21.5	13.9	23.3	15.3	15.1	12.7	19.5	13.5	12.9	3.0	3.1	3.1	3.1
	259	378	235	174	112	127	108	74	100	109	96	67	4	4	4	2
SS	23.8	57.2	25.3	22.9	14.8	23.2	15.9	17.0	12.5	19.9	14.0	13.0	3.1	3.5	3.4	3.2
	267	396	242	180	114	129	109	82	102	112	98	66	4	4	4	2

Table 10 Linear solver time (secs) and number of iterations required to converge for general second-order elliptic PDEs (problem (B) in Table 3) as a function of mesh quality metric for the 16 preconditioner-solver combinations (see Table 5) on the two geometric domains. A '*' denotes failure. For each quality metric, the numbers in the top and bottom rows represent the linear solver time and number of iterations to convergence, respectively.

tional iterations needed to converge faster. The least robust combinations are the Jacobi preconditioner with any choice of the solver.

Exp. 3: Increasing the problem size for general second-oder elliptic PDEs. Figure 8(a) and 8(b) show the order of convergence for the number of iterations required to converge as we increase N. The overall trend is similar to that seen for problem (A). The Jacobi preconditioner with the GMRES solver has the largest s value which is 0.63. The s value of the CG solver with any preconditioner is approximately 0.5. ***We observe that the AMG preconditioner is not sensitive to an increase in N.*** Four solvers when combined with the AMG preconditioner have s values less than 0.1. Figures 9(a) and 9(a) show the same results for the solver times. The results are similar to the results for problem (A). The best combinations (the RR quality metric and the AMG preconditioner with any choice of the solver) is not affected by increasing the problem size.

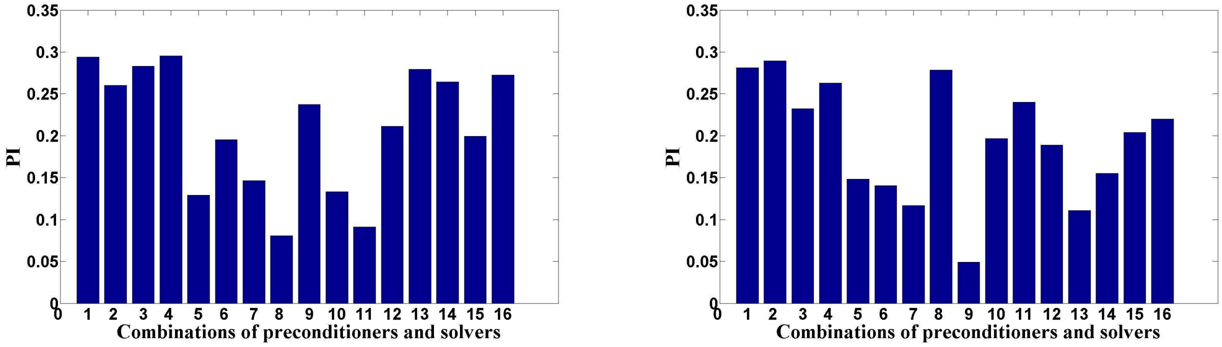
7.3.4 Numerical Results for the Linear Elasticity Problem

Exp. 1: Determination of the best combinations of mesh quality metric, preconditioner, and solver for linear elasticity. Table 11 shows the solver time and number of iterations required for convergence as a function of various combinations of preconditioners and solvers for problem (C) in Table 3. We observe that the efficiency rankings are different from those obtained from Poisson's equation (problem (A)) and general elliptic PDEs (problem (B)).

The ILU(0)-preconditioned solvers are more efficient than the AMG-preconditioned solvers. The reason for the difference is the difference in sparsity pattern of the A matrix for each application. For Poisson's equation, the matrix A has a sparsity pattern that corresponds to the mesh connectivity. Thus, the AMG preconditioner works very well when an aggregation step is performed. For the linear elasticity equations, A has twice the number of rows and columns as the number of vertices in the mesh. Thus, the AMG coarsening algorithm generates aggregates of physically-incompatible degrees of freedom [37]. This results in an increased solver time for the AMG preconditioner to solve the linear elasticity problem.

Also, the BiCG-STAB solver shows an approximately 20% variation in the number of iterations required for preconditioner-solver combinations as a function of quality metric. The variation of other solvers is in the range of 10%. We observe the power of preconditioners in some cases. Although the meshes smoothed by the SS metric have more poorly-shaped elements than those obtained by smoothing using other quality metrics, the solver time for the SS metric is 50% less than that for the IMR metric.

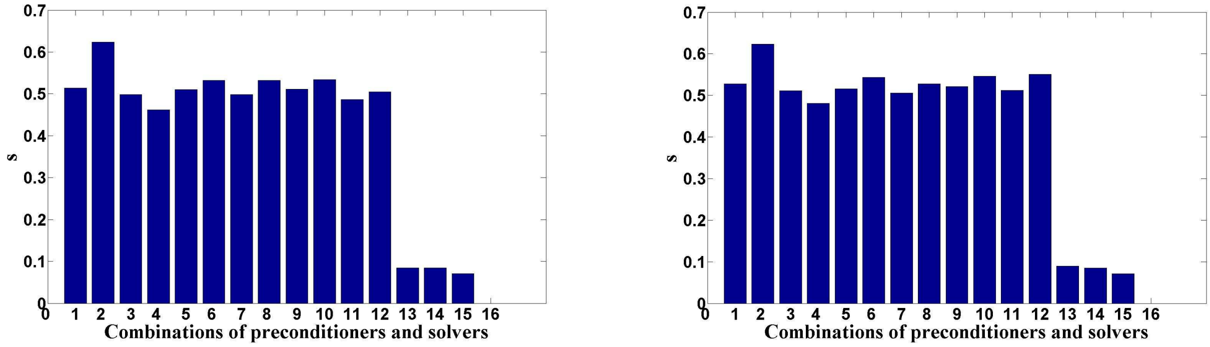
Among the 16 combinations of preconditioners and solvers, the most efficient combinations are the IMR or the SI quality metric with the ILU(0) preconditioner and the CG solver with respect to the solver time. On the wrench do-



(a) Wrench (500K)

(b) Hinge (500K)

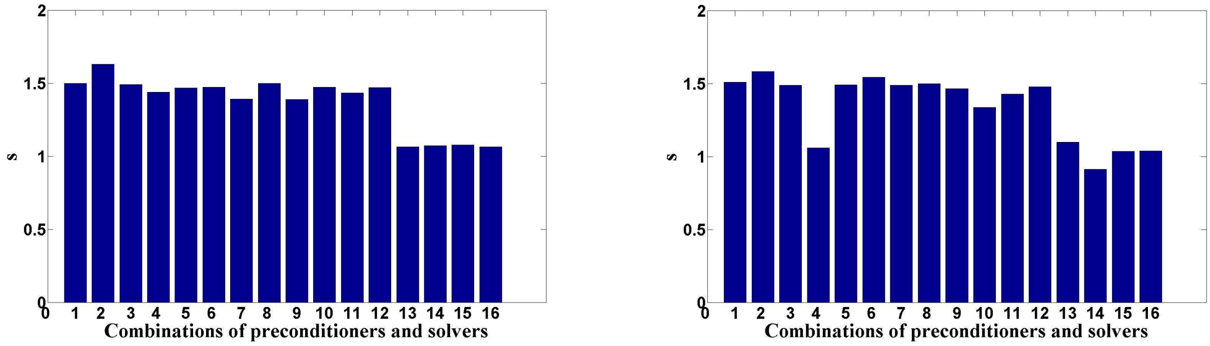
Fig. 7 PI as a function of the solver time for various combinations of preconditioners and solvers after vertex perturbation for general second-order elliptic PDEs (problem (B) in Table 3).



(a) Wrench

(b) Hinge

Fig. 8 The order of convergence for the solver time based on the number of iterations for the combinations of preconditioners and solvers for general second-order elliptic PDEs (problem (B) in Table 3). Meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain are employed in computing s using the linear least-squares method.



(a) Wrench

(b) Hinge

Fig. 9 Similar to Figure 8, this figure displays the order of convergence based on the solver time for general second-order elliptic PDEs (problem (B) in Table 3). Meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain are employed in computing s using the linear least-squares method.

main, the most efficient combination is 92% faster than the least efficient combination (i.e., the RR quality metric with the Jacobi preconditioner and the GMRES solver) with respect to the solver time.

In terms of the total time, the IMR quality metric with the ILU(0) preconditioner and the MINRES solver outperforms other combinations. The least efficient combination is the SS quality metric with the Jacobi preconditioner and the GMRES solver. The total time for the most efficient combination (i.e., ILU(0) with MINRES) with the IMR metric is 71% less than it is for this combination with SS. The least efficient combination (i.e., Jacobi with GMRES) for the IMR metric is 16% faster than that of the least efficient quality metric, i.e., SI. In this case, the total time is more significantly affected by the solver time.

We conducted experiments in which we modified the PDE coefficients in (4) such that the coefficient matrix $C = \gamma I$, where γ is a constant. Similar to the second-order general elliptic PDE problems, experimental results show that the sparsity pattern of A, the optimal triangular shape, and hence $\kappa(A)$ are not affected by these modifications as expected based on the discussion in Section 2.

Exp. 2: Effect of perturbation for linear elasticity. Table 12 shows the effect of the vertex perturbation on the solution of the linear elasticity problem. For these experiments, the meshes are smoothed with the IMR quality metric. The results shown here are typical. We observe that the effect of perturbation is different from problem (A) and (B). As was explained before, this is because the sparsity pattern of the linear system in this application is different from that of the previous applications. Figures 10(a) and 10(b) show the PI for different preconditioner-solver combinations when the number of perturbed vertices is increased. The AMG preconditioner with the GMRES solver (combination 14 in Table 5) is least sensitive to vertex perturbation. ***Similar to problem (B), the SSOR preconditioner with the CG and MINRES solvers (combinations*** 5 ***and*** 7 ***in Table 5) are also not very sensitive to perturbation.***

We observe that the SSOR preconditioner with the GMRES solver or Bi-CGSTAB solver and the ILU(0) preconditioner with the GMRES solver (combinations 6, 8 and 10 in Table 5) are most sensitive to perturbation. These two solvers are not efficient to circumvent the numerical difficulties associated with poor eigenvalues. The SSOR preconditioner with the Bi-CGSTAB solver takes 52% more time to converge than does the most robust combination (combination 14 in Table 5). On both domains, the AMG preconditioner is less affected by vertex perturbation than the other preconditioners because the coarse grid correction during the V-cycles removes large eigenvalues effectively.

Exp. 3: Increasing the problem size for linear elasticity. Figures 11(a) and 11(b) show the s values for the number of iterations required to converge for the wrench and hinge domains, respectively. For this application, the experiments are performed on meshes smoothed using the IMR quality metric. For the simple wrench domain, the s values for the AMG preconditioner are less than for the other preconditioners. For the more complex hinge domain, however, the order of convergence of the number of iterations for all the combinations are all close to 0.5. Note that for the Jacobi preconditioner with the GMRES solver, s is approximately 0.8. Figures 12(a) and 12(b) show the order of convergence of the solver time for the wrench and the hinge domain, respectively.

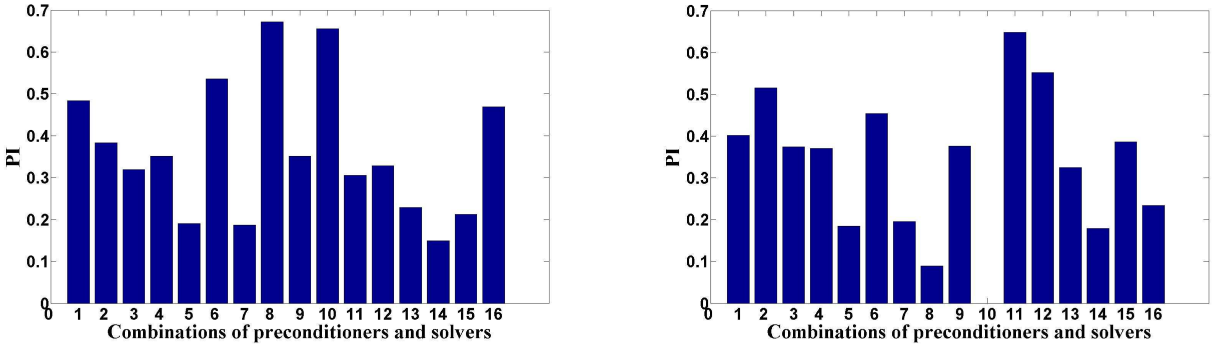
In most cases, the best combinations are the RR quality metric with the ILU(0) preconditioner and the MINRES (or CG) solver. The least efficient combinations are the SS quality metric with the Jacobi preconditioner and the GMRES solver.

8 Conclusions and Future Work

We studied the most efficient combinations of quality metric, preconditioner, and sparse linear solver for the numerical solution of various elliptic PDEs on 2D geometric domains. Our paper is the first to simultaneously study these three important factors, which affect the efficiency of the solution for various elliptic PDEs. According to our experimental results, by choosing the most efficient combination, solver time and total time can be reduced by 90% and 97%, respectively, when compared to those with the most inefficient combination.

For all elliptic PDEs considered here, we observe that the radius ratio (RR) metric is the most efficient metric for minimizing the solver time, as the mesh smoothed by the RR metric contains the fewest poorly-shaped elements. Poorly-shaped elements increase both the maximum eigenvalue and the condition number of the linear system. The interpolation-based size-and-shape (SS) metric is the least efficient mesh quality metric in terms of its effect on the solver time because meshes smoothed with the SS metric have more poorly-shaped elements than the meshes smoothed by other quality metrics. We also observe that the choice of the preconditioner is the most important factor that affects the solver time for all elliptic PDEs. The choice of a good preconditioner is more important than the choice of a good quality metric or good linear solver even if the initial mesh quality is extremely poor.

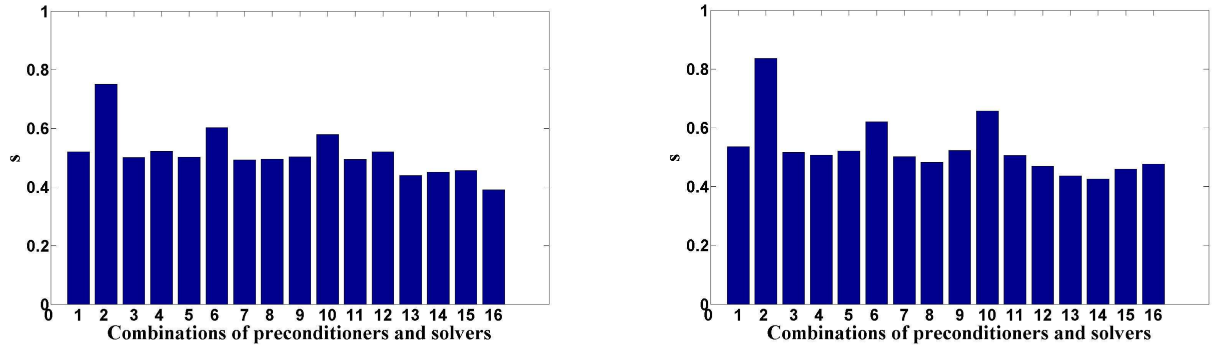
For solving Poisson's equation and general second-order elliptic PDEs, the most efficient combination for minimizing the solver time is the RR quality metric with the algebraic multigrid (AMG) preconditioner and any choice of linear solver. For these problems, which have a single degree



(a) Wrench (500K)

(b) Hinge (500K)

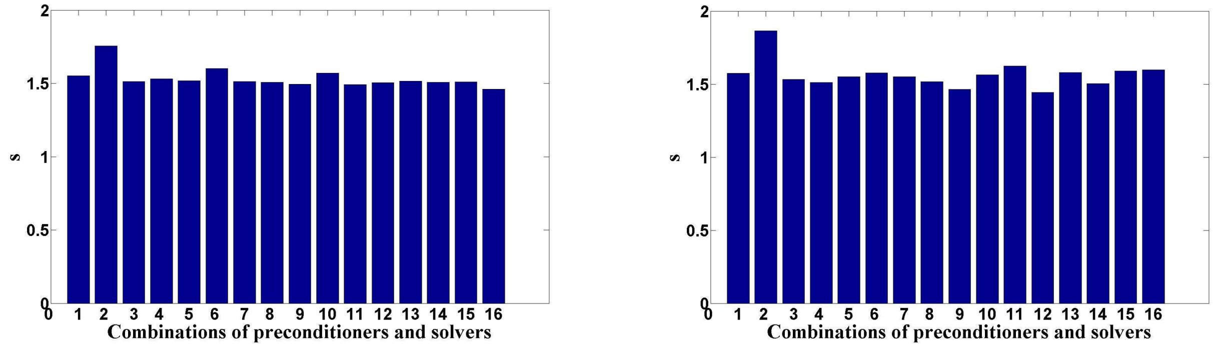
Fig. 10 PI as a function of the solver time after vertex perturbation for the combinations of preconditioners and solvers for the linear elasticity problem (problem (C) in Table 3). The missing bar (combination 10) for the hinge domain corresponds to a preconditioner-solver combination which does not converge.



(a) Wrench

(b) Hinge

Fig. 11 The order of convergence for the solver time based on the number of iterations for the different combinations of preconditioners and solvers for the linear elasticity problem (problem (C) in Table 3). Meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain are employed in computing s using the linear least-squares method.



(a) Wrench

(b) Hinge

Fig. 12 Similar to Figure 11, this figure displays the order of convergence based on the solver time for the linear elasticity problem (problem (C) in Table 3). Meshes with 10K, 50K, 100K, 200K, and 500K vertices on each domain are employed in computing s using the linear least-squares method.

(a) Wrench (500K)

Quality	Combinations of preconditioners and solvers															
Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IMR	475	2006	295	291	181	375	175	167	165	306	163	182	294	303	311	306
	1528	6330	1353	1046	595	974	554	367	545	792	510	402	127	136	149	75
RR	464	2101	294	361	180	376	172	189	163	301	158	173	304	299	312	326
	1508	7120	1336	1280	594	973	553	416	545	792	510	382	130	135	148	78
SI	454	1801	300	355	177	395	170	189	167	342	165	181	350	327	341	309
	1512	6410	1338	1199	594	1010	553	394	545	798	510	377	126	135	147	81
SS	495	1846	300	322	196	459	188	164	155	401	171	186	259	291	336	349
	1555	6192	1377	1083	601	1202	559	358	554	1066	519	404	128	138	150	82

(b) Hinge (500K)

Quality	Combinations of preconditioners and solvers															
Metric	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
IMR	271	1282	194	178	121	226	127	124	105	259	192	102	212	195	243	300
	1054	4274	915	620	414	615	370	244	372	651	339	219	88	80	103	67
RR	346	1181	198	196	202	323	112	126	188	246	105	108	169	214	201	222
	1043	4119	906	715	395	613	366	283	361	655	337	239	88	80	103	60
SI	261	1086	200	183	111	239	126	139	103	239	108	105	158	238	275	391
	1048	4074	910	677	395	612	366	277	361	653	337	227	87	80	103	70
SS	253	1045	195	175	119	256	105	111	111	245	103	118	236	176	198	244
	1047	3681	909	639	400	606	371	250	368	614	344	266	86	81	102	64

Table 11 Linear solver time (secs) and number of iterations required to converge for the linear elasticity problem (problem (C) in Table 3) as a function of mesh quality metric for the 16 preconditioner-solver combinations (see Table 5) on the two geometric domains. A '*' denotes failure. For each quality metric, the numbers in the top and bottom rows represent the linear solver time and number of iterations to convergence, respectively.

(a) Wrench (500K)

Percent Vertices	Combinations of preconditioners and solvers															
Pert.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0%	475	2006	295	291	181	375	175	167	165	306	163	182	294	303	311	306
	1528	6330	1353	1046	595	974	554	367	545	792	510	402	127	136	149	75
5%	581	2294	372	320	195	462	191	211	185	465	178	173	323	308	345	401
	1669	8066	1475	1133	630	1167	586	455	576	1144	539	358	125	139	150	81
10%	567	2272	365	394	183	481	202	195	176	380	167	173	284	348	339	273
	1693	7904	1496	1421	633	1259	588	414	582	1028	543	387	126	139	151	71
25%	642	2775	389	312	216	475	196	197	176	506	178	198	263	331	286	309
	1799	9751	1597	1103	639	1180	595	443	621	1385	582	431	125	139	151	80
50%	705	2428	358	348	216	576	207	280	222	448	213	229	279	311	347	318
	1900	8841	1680	1247	653	1301	606	538	666	1198	622	463	128	140	151	78

(b) Hinge (500K)

Percent Vertices	Combinations of preconditioners and solvers															
Pert.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0%	271	1282	194	178	121	226	127	124	105	259	192	102	212	195	243	300
	1054	4274	915	620	414	615	370	244	372	651	339	219	88	80	103	67
5%	288	975	267	213	121	291	141	127	117	252	147	124	207	173	244	245
	1120	3194	972	732	420	645	389	259	386	617	360	271	89	82	106	65
10%	301	846	213	211	119	267	132	134	113	277	126	117	169	169	201	247
	1136	2928	985	684	422	649	391	280	388	606	361	267	89	81	105	62
25%	313	1084	233	234	134	251	118	133	112	227	116	126	210	198	278	259
	1192	3806	1034	781	427	659	395	301	404	612	375	272	88	82	106	64
50%	380	1149	252	244	141	329	128	136	144	*	156	159	160	199	240	302
	1282	4131	1114	801	439	671	403	276	499	*	476	303	88	82	107	64

Table 12 Linear solver time (secs) and number of iterations required to converge for the linear elasticity problem (problem (C) in Table 3) as a function of vertex perturbation for the 16 preconditioner-solver combinations (see Table 5) on the two geometric domains. A '*' denotes failure. For each percentage of vertices perturbed, the numbers in the top and bottom rows represent the linear solver time and number of iterations to convergence, respectively.

of freedom, the AMG preconditioner efficiently eliminates large eigenvalues, and its aggregation step is fast. This results in a solver time which is 97% faster than the least efficient combination, i.e., the Jacobi preconditioner and the GMRES solver with any choice of quality metric.

For linear elasticity problems, the RR metric with the ILU(0) preconditioner and the MINRES (or CG) solver is the most efficient combination, which are up to 92% faster than the least efficient combination, which is the Jacobi preconditioner and the GMRES solver with any choice of the quality metric. Due to its higher degree of freedom and different sparsity pattern, where displacements for x and y are computed in a coupled manner, the AMG preconditioner is not as efficient as it is with the other two problems.

The most efficient combinations are also studied with respect to the total time, which includes both the smoothing and the solver time. For all elliptic PDEs, choosing an efficient quality metric is more important than the other factors in order to minimize the total time (as opposed to just the solver time). The inverse mean ratio (IMR) metric computation is highly optimized in Mesquite, and thus combinations with the IMR metric appear in the best combination. For Poisson's equation and general second-order elliptic PDEs, the IMR metric with the AMG preconditioner and the CG solver is 90% faster than the same combination with the SS metric. For linear elasticity problems, the IMR metric with the ILU(0) preconditioner and MINRES (or CG) solver outperforms other combinations.

We also investigated the robustness of the combinations with respect to vertex perturbation. These results are useful for choosing the most robust preconditioner-solver combination when the initial mesh quality is poor and mesh smoothing is not performed. In most cases, the SSOR preconditioner with the CG or MINRES solver is more robust than other combinations because they are able to circumvent numerical difficulties associated with large eigenvalues as discussed above. The AMG preconditioner is less affected by vertex perturbation than the other preconditioners because the coarse grid correction during the V-cycles removes large eigenvalues effectively.

Finally, we examined the effect of increasing the problem size on the number of iterations required to converge and on the solver time. For all three PDEs, the AMG preconditioner with any quality metric or solver exhibits up to 94% faster convergence than the other combinations of preconditioners and solvers as the problem size is increased. The order of convergence for Poisson's equation and general second-order elliptic PDEs, combinations with the AMG preconditioner are asymptotically much faster than the other combinations.

Our experimental results can be generalized to the solution of other elliptic PDE problems with constant isotropic coefficients on various 2D geometric domains and homogeneous boundary conditions. However, they cannot be generalized to elliptic PDE problems with anisotropic coefficients or to 3D geometric domains for the following two reasons. First, elliptic PDEs with anisotropic coefficients require different ideal element shapes. Second, the quality metrics for 3D geometric domains are different from those for 2D elements.

For future research, we will experimentally and theoretically analyze finite element solutions of elliptic PDEs with anisotropic and discontinuous PDE coefficients. We will also investigate the relationship between the choice of mesh quality metric and the efficient solution of parabolic and hyperbolic PDEs on anisotropic unstructured meshes and computational fluid dynamics problems will be also considered.

Acknowledgements

The authors would like to thank Anirban Chatterjee and Padma Raghavan for interesting the third author in this area of research and Nicholas Voshell for helpful discussions. This work was funded in part by NSF grant CNS 0720749 and an Institute for Cyberscience grant from The Pennsylvania State University. This work was supported in part through instrumentation funded by the National Science Foundation through grant OCI-0821527. They also wish to thank the two anonymous referees for their careful reading of the paper and for their helpful suggestions which strengthened it.

References

1. I. Babuska and M. Suri, *The p and h-p versions of the finite element method, basic principles, and properties*, SIAM Rev., 35: 579-632, 1994.
2. M. Berzins, *Solution-based mesh quality for triangular and tetrahedral meshes*, in Proc. of the 6^{th} International Meshing Roundtable, Sandia National Laboratories, pp. 427-436, 1997.
3. M. Berzins, *Mesh quality - Geometry, error estimates, or both?*, in Proc. of the 7^{th} International Meshing Roundtable, Sandia National Laboratories, pp. 229-237, 1998.
4. E. Fried, *Condition of finite element matrices generated from nonuniform meshes*, AIAA Journal, 10: 219-221, 1972.
5. J. Shewchuk, *What is a good linear element? Interpolation, conditioning, and quality measures*, in Proc. of the 11^{th} International Meshing Roundtable, Sandia National Laboratories, pp. 115-126, 2002.
6. Q. Du, Z. Huang, and D. Wang, *Mesh and solver co-adaptation in finite element methods for anisotropic problems*, Numer. Meth. Part. D. E., 21: 859-874, 2005.
7. Q. Du, D. Wang, and L. Zhu, *On mesh geometry and stiffness matrix conditioning for general finite element spaces*, SIAM J. Numer. Anal., 47(2): 1421-1444, 2009.
8. A. Ramage and A. Wathen, *On preconditioning for finite element equations on irregular grids*, SIAM J. Matrix Anal. Appl., 15: 909-921, 1994.
9. A. Chatterjee, S.M. Shontz, and P. Raghavan, *Relating Mesh Quality Metrics to Sparse Linear Solver Performance*, in Proc. of the SIAM Conference on Computational Science and Engineering, Costa Mesa, CA, 2007.

10. D. Mavripilis, *An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers*, J. Comput. Phys., 175: 302-325, 2002.
11. M. Batdorf, L. Freitag, and C. Ollivier-Gooch, *Computational study of the effect of unstructured mesh quality on solution efficiency*, in Proc. of the 13^{th} CFD Conference, AIAA, Reston, VA, 1997.
12. L. Freitag and C. Ollivier-Gooch, *A cost/benefit analysis of simplicial mesh improvement techniques as measured by solution efficiency*, Internat. J. Comput. Geom. Appl., 10: 361-382, 2000.
13. S. Bhowmick, P. Raghavan, L. C. McInnes and B. Norris, *Faster PDE-based Simulations using Robust Composite Linear Solvers*, Future Generation Computer Systems, 20(3), 2004.
14. M. Brewer, L. Freitag Diachin, P. Knupp, T. Leurent, and D. Melander, *The Mesquite Mesh Quality Improvement Toolkit*, in Proc. of the 12^{th} International Meshing Roundtable, Sandia National Laboratories, pp. 239-250, 2003.
15. S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B. F. Smith, and Hong Zhang, *PETSc Webpage, http://www.mcs.anl.gov/petsc*, 2009.
16. T. Munson, *Mesh Shape-Quality Optimization Using the Inverse Mean-Ratio Metric*, Math. Program., 110: 561-590, 2007.
17. R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H.Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2^{nd} Edn., SIAM, 1994.
18. A. Brandt, S. McCormick, and J. Ruge, *Algebraic Multigrid for Sparse Matrix Equations,* Cambridge Univ. Press, Cambridge, 1985.
19. M.R. Hestenes and E. Stiefel. *Methods of Conjugate Gradients for solving linear systems*. J. Res. Natl. Bur. Stand., 49:409-436, 1952.
20. C.C. Paige and M.A. Saunders. *Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal.*, 12:617-629, 1975.
21. Y. Saad and M. Schultz. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 7(3):856-869, 1986.
22. H.A. van der Vorst. *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*. SIAM J. Sci. Comput., 13:631-644, 1992.
23. J. Kim, S.P. Sastry, and S.M. Shontz, *Efficient Solution of Elliptic Partial Differential Equations via Effective Combination of Mesh Quality Metrics, Preconditioners, and Sparse Linear Solvers,* in Proc. of the 19^{th} International Meshing Roundtable, Sandia National Laboratories, pp. 103-120, 2010.
24. E.B. Becker, G.F. Carey, and J.T. Oden. *Finite Elements: An Introduction. Prentice-Hall*, Englewood Cliffs, New Jersey, 1981.
25. Y. Kwon and H. Bang, *The Finite Element Method using Matlab,* CRC Press,2^{nd} Edn., 2000.
26. S. Oh and J. Yim, *Optimal finite element mesh for elliptic equation of divergence form,* Appl. Math. Comput., 162:969-989, 2005.
27. L. Diachin, P. Knupp, T. Munson, and S. Shontz, *A comparison of two optimization methods for mesh quality improvement*, Eng. Comput., 22(2): 61-74, May 2006.
28. J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, 2^{nd} Edn., 2006.
29. M. Brewer, *Obtaining smooth mesh transitions using vertex optimization*, Internat. J. Numer. Methods Engrg., 75: 555-576, 2008.
30. C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, 1987.
31. A.H. Baker, E.R. Jessup, and Tz.V. Kolev, *A simple strategy for varying the restart parameter in GMRES(m)*, J. Comput. Appl. Math. 230: 751-761, 2009.
32. Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2^{nd} Edn., 2003.
33. Cyberstar Webpage: http://www.ics.psu.edu/research/cyberstar/index.html.
34. J.R. Shewchuk, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator,* Lecture Notes in Computer Science, vol. 1148, pp. 203-222, 1996.
35. B. Norris, L. McInnes, S. Bhowmick, and L. Li, *Adaptive Numerical Components for PDE-Based Simulations,* ICIAM Proceedings of Applied Mathematics and Mechanics, 2007.
36. Suzanne M. Shontz and Patrick Knupp, *The effect of vertex reordering on 2D local mesh optimization efficiency,* in Proc. of the 17^{th} International Meshing Roundtable, pp. 107-124, 2008.
37. P. Vaněk, J. Mandel, and M. Brezina, *Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems,* Computing, 56(3):179-196, 1996.