Massively Parallel Solver for the High-Order Galerkin Least Squares

Masayuki Yano* and David L. Darmofal†

Aerospace Computational Design Laboratory, Massachusetts Institute of Technology

A high-order Galerkin Least-Squares (GLS) finite element discretization is combined with massively parallel implicit solvers. The Balancing Domain Decomposition by Constraints (BDDC) algorithm is applied to the linear system arising from the two-dimensional, high-order discretization of the advection-diffusion equation and the Euler equation. The Robin-Robin interface condition is extended to the Euler equation using the entropy-symmetrized variables. The BDDC method maintains scalability for the high-order discretization of the diffusion-dominated flows, and achieves low iteration count in the advection-dominated regime. The BDDC method based on the inexact local solvers with incomplete factorization with $p = 1$ coarse correction maintains the performance of the exact counterpart for the wide range of the Peclet numbers considered.

I. Introduction

Although the Computational Fluid Dynamics (CFD) has matured significantly over the past decades, there remains a number of challenging problems that are beyond the current CFD capabilities. In,[1] Mavriplis lists some Grand Challenges in the aerospace community, including: a complete flight-envelope characterization, full engine simulations, and probabilistic computational optimization. Mavriplis points out that the biggest impediment to solving these problems is not the hardware capability, which has been increasing exponentially, but rather the lack of a robust, high-fidelity solver that can take advantage of massively parallel architectures that will deliver the computational power needed. In fact, today's most powerful computers house more than 100,000 processors, and the trend of massive parallelization is expected to continue.[2]

The difficulty in high-fidelity, efficient CFD simulations arise from the large range of temporal and spatial scales present in the flow structures; the scale of turbulence structures and the aircraft body can easily vary by more than six orders of magnitude. Thus, the discretization must be capable of efficiently capturing the widest range of scales and, given the geometric complexity, handle unstructured meshes. To meet the requirements, this work employs the Galerkin Least-Squares method, which enables arbitrarily high-order accurate discretization on unstructured meshes.

Furthermore, the stiff problem, which results from the wide range of scales present, necessitates the use of an implicit method for a robust simulation at a reasonable cost. The solver must also be highly scalable to take advantage of the massively parallel computers. To address these problems, the Balancing Domain Decomposition by Constraints, which was initially developed for large-scale structural dynamics problems, is adopted for a system of conservation laws and employed to solve the linear systems arising from the high-order discretization of advection-dominated flows.

Stabilized finite element methods have been developed extensively for hyperbolic and parabolic conservation laws, including the Euler equations and the Navier-Stokes equations. These methods provide consistent, locally conservative,[3,4] and arbitrarily high-order accurate discretization on unstructured meshes. The Galerkin Least-Squares (GLS) method[5] used in this work is a generalization of the Streamline-Upwind Petrov-Galerkin (SUPG) method.[6,7] The GLS method improves the stability of the classical Galerkin method using the least-squares operator and is conceptually simpler than SUPG in the presence of diffusion. The method has been applied to systems of hyperbolic equation[8,9] using the symmetrization theory for conservation laws.[10,11]

*Doctoral candidate, 77 Massachusetts Ave. 37-401, Cambridge, MA, 02139, myano@mit.edu

†Associate Professor, Senior Member AIAA, 77 Massachusetts Ave. 37-442, Cambridge, MA, 02139, darmofal@mit.edu

Massively parallel solvers that are most relevant to the current work are non-overlapping domain decomposition methods, known as iterative substructuring methods. These methods were developed to solve symmetric, positive-definite linear systems arising from finite element discretization of elliptic systems in parallel environments.[12] The Balancing Domain Decomposition by Constraints (BDDC) algorithm[13] used in this work evolved from the Balancing Domain Decomposition (BDD) method,[14] which belongs in the family of the Neuamnn-Neumann methods.[15] The BDDC method constructs the coarse, global component from a set of selected primal constraints, and achieves the condition number independent of the number of subdomains when applied to the Poisson equation, having the same spectrum as the FETI-DP methods.[16–18] The use of inexact solvers for the BDDC method has been considered recently in.[19,20] In these work, the subdomain problems or the partially assembled system is solved using an incomplete factorization or multigrid. The BDDC method for spectral elements using the Gauss-Lobatto-Legendre quadrature nodes has also appeared recently in.[21]

Although the iterative substructuring methods were originally designed for symmetric, positive-definite systems, the methods have been applied to the advection-diffusion equation to a lesser extent. In,[22] the typical Neumann-Neumann interface condition of the elliptic problem is replaced with the Robin-Robin interface condition to maintain the positivity of the local bilinear form. The interface condition has also been applied in the FETI[23] and BDDC[24] frameworks to solve the advection-diffusion equation.

This paper is organized as follows. Section II describes the high-order GLS discretization used in this work. Section III introduces the BDDC preconditioner, with the generalization of the Robin-Robin interface condition to a system of equations. Section IV present the BDDC algorithm based on the inexact local solvers, which are designed to maintain the scalability of the exact counterpart at a reduced cost. Section V presents the result of applying the BDDC algorithm to the advection-diffusion equation with a wide range of Peclet numbers and the Euler equation. Finally, Section VI discusses conclusions and ongoing work.

II. The Galerkin Least-Squares Method

II.A. Variational Form of Conservation Laws

Let $\Omega \in \mathbb{R}^d$ be an open, bounded domain, where d is the number of spatial dimensions. In general, a system of time-dependent conservation laws is expressed as

$$u_{k,t} + (F^{\text{inv}}_{ik})_{,x_i} - (F^{\text{vis}}_{ik})_{,x_i} = f_k, \quad \text{in } \Omega \tag{1}$$

where $k \in \{1, \dots, m\}$ is the component index of the governing equations, $i \in \{1, \dots, d\}$ is the spatial index, $(\cdot)_{,t}$ denote the temporal derivative, and $(\cdot)_{,x_i}$ denote the spatial derivatives with respect to x_i. The inviscid flux $F^{\text{inv}} = F^{\text{inv}}(u, x, t)$, viscous flux $F^{\text{vis}} = F^{\text{vis}}(u, \nabla u, x, t)$, and the source term, $f(x, t)$, characterize the governing equations to be solved. The quasi-linear form of the governing equation is given by

$$\mathcal{L}u \equiv u_{k,t} + A_{ikl} u_{l,x_i} - (K_{ijkl} u_{l,x_j})_{,x_i} = f_k, \tag{2}$$

where the inviscid flux Jacobian and viscous flux tensor are defined to satisfy

$$A_{ikl} = \frac{\partial F^{\text{inv}}_{ik}}{\partial u_l} \quad \text{and} \quad K_{ijkl} u_{l,x_j} = F^{\text{vis}}_{ik}. \tag{3}$$

The finite element discretization of the problem is performed on a space of functions

$$V_h = \{u \in [H^1(\Omega)]^m : u|_K \in [\mathcal{P}_p(K)]^m, \forall K \in \mathcal{T}_h\} \tag{4}$$

where $\mathcal{T}_h$ is the triangulation of domain Ω into non-overlapping elements, K, such that $\bar{\Omega} = \cup_{K \in \mathcal{T}_h} \bar{K}$, and $\mathcal{P}_p(K)$ is the space of p-th order polynomial on K. The superscript m implies the spaces are vector-valued. The finite element variational problem consists of finding $u \in V_h$ such that

$$(u_{k,t}, v_k)_\Omega + \mathcal{R}_{gal}(u, v) = 0 \quad \forall v \in V_h,$$

where

$$R_{gal}(u, v) = -(F^{\text{inv}}_{ik}, v_{k,x_i})_\Omega + (F^{\text{vis}}_{ik}, v_{k,x_i})_\Omega - (f_k, v_k)_\Omega + (\hat{F}_k(u, \text{B.C.data}, n), v_k)_{\partial\Omega},$$

where $(\cdot,\cdot)_\Omega : L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}$ and $(\cdot,\cdot)_{\partial\Omega} : L^2(\partial\Omega) \times L^2(\partial\Omega) \to \mathbb{R}$ denote the L^2 inner product over the domain and the boundary of the domain, respectively. The numerical flux function, $\hat{F}$, uses the interior state and the boundary condition to define the appropriate flux at the boundary. It is well known that the standard Galerkin method becomes unstable for a large grid Peclet number, P_e, and exhibits spurious oscillations in the vicinity of unresolved internal and boundary layers. The Galerkin Least-Squares (GLS) method remedies this problem by directly controlling the strong form of the residual. The GLS problem consists of finding $u \in V_h$ such that

$$(u_{k,t}, v_k)_\Omega + \mathcal{R}_{gal}(u,v) + \mathcal{R}_{ls}(u,v) = 0 \quad \forall v \in V_h, \tag{5}$$

where the least-squares residual is given by

$$\mathcal{R}_{ls}(u,v) = ((\mathcal{L}v)_l, \tau_{lk}(\mathcal{L}u - f)_k)_{\Omega,\mathcal{T}_h},$$

where $\mathcal{L}$ is the linear differential operator defined in Eq. (2), and τ is the stabilization parameter. $(\cdot,\cdot)_{\Omega,\mathcal{T}_h}$ denotes the summation of the element-wise L^2 inner product, $\sum_K (\cdot,\cdot)_K$. In the limit of $F^{\text{vis}} \to 0$, the stabilization term, $R_{ls}(\cdot,\cdot)$, adds viscosity only in the streamwise direction, and the scheme is equivalent to the Streamline Upwind Petrov-Galerkin method.[5]

II.A.1. Advection-Diffusion Equation

To assess the performance of the Galerkin Least-Squares method and the performance of the linear solvers, two set of equations are considered: the advection-diffusion equation and the Euler equation. The advection-diffusion equation is characterized by inviscid and viscous fluxes given by

$$F_i^{\text{inv}} = \beta_i u \quad \text{and} \quad F_i^{\text{vis}} = \kappa u_{,x_i},$$

where β is the advection field and κ is the diffusivity coefficient. For simplicity, the Dirichlet boundary condition is imposed everywhere on $\partial\Omega$ for all cases considered in this study. The viscous boundary condition is imposed using the second method of Bassi and Rebay (BR2)[25] , a popular method for handling the viscous terms in the discountinuous Galerkin method, instead of by changing the trial space, which is typically used in the continuous Galerkin framework. The BR2 discretization allows the use of the same function space for the trial and test functions and also admits more flexible treatment of the boundary conditions.

II.A.2. Euler Equations

The Euler equation is formulated using the entropy-symmetrized variables. The two sets of entropy variables commonly used for the Euler equations are Harten's variables[10] and Hughes' variables.[11] When applied to the Navier-Stokes equations, Hughes' variables have the advantage that they also symmetrize the viscosity tensor. The entropy variables used in this study are the scaled version of Hughes' variables used in.[26] The entropy-symmetrized state and the flux functions are given by

$$u^{\text{entropy}} = \begin{pmatrix} \frac{-s+\gamma+1}{\gamma-1} - \frac{\rho E}{p} \\ \frac{\rho v_1}{p} \\ \frac{\rho v_2}{p} \\ -\frac{\rho}{p} \end{pmatrix} \quad \text{and} \quad F_i^{\text{inv}} = \begin{pmatrix} \rho v_i \\ \rho v_1 v_i + p\delta_{i1} \\ \rho v_2 v_i + p\delta_{i2} \\ \rho H v_i \end{pmatrix},$$

where ρ is the fluid density, v is the velocity vector, p is the pressure, and E is the specific stagnation internal energy. The entropy, s, is given by $s = \log(p/\rho^\gamma)$, the specific stagnation enthalpy, H, is given by $H = E + p/\rho$, and the pressure is given by $p = (\gamma-1)\rho(E - \|v\|^2/2)$, where γ is the ratio of specific heats.

II.B. Stabilization Parameter τ

The stabilization parameter controls the amount of stabilization added to the Galerkin Least-Squares discretization. From the variational multiscale perspective, it can be thought of as adding the dissipation that would have been provided by the unresolved, subgrid scale.[27,28] It is well known[8,29,30] that, in order to attain optimal convergence rate with the element size h, the stabilization parameter must scale as

$$\begin{aligned}\tau &= \mathcal{O}(h/|\beta|), \quad Pe \gg 1\\ \tau &= \mathcal{O}(h^2/\kappa), \quad Pe \ll 1.\end{aligned}$$

The first condition is necessary to obtain stability and optimal convergence in the streamline derivative, $\|\beta_i u_{,x_i}\|_{L^2(\Omega)}$, in advection-dominated cases, and the second condition is necessary to maintain the optimality in diffusion-dominated case. As the conservation laws of interest consist of inviscid and viscous parts, τ may be conveniently expressed as the sum of these two parts. In particular, the scaling relation can be satisfied by choosing

$$\tau^{-1} = \tau_{\text{inv}}^{-1} + \tau_{\text{vis}}^{-1},$$

where $\tau_{\text{inv}} = \mathcal{O}(h/|\beta|)$ and $\tau_{\text{vis}} = \mathcal{O}(h^2/\kappa)$. In addition to satisfying the scaling relations, the τ matrix must be symmetric and positive-definite for a system of equations.[9] Generalization of the inviscid stabilization parameter to a multidimensional system of equations follows from.[26] Let the directional vectors associated with the mapping from physical space, described by coordinate x, to the reference space, described by barycentric coordinate ξ, be

$$n^i = (\xi_{i,x_1}\xi_{i,x_2})^T, \quad i = 1, 2, 3,$$

and the unit normal vector be $\hat{n}^i = n^i/|n^i|$. For example, in two-dimensions, the normal vectors are

$$\begin{aligned}n^1 &= (\xi_{1,x_1}, \xi_{1,x_2})^T\\ n^2 &= (\xi_{2,x_1}, \xi_{2,x_2})^T\\ n^3 &= -(\xi_{1,x_1} + \xi_{2,x_1}, \xi_{1,x_2} + \xi_{2,x_2})^T.\end{aligned}$$

The directional flux Jacobian is given by

$$A(n^i) = n^i_j A_j,$$

where A_j is the flux Jacobian in the x_j coordinate direction. The inviscid stabilization parameter is defined as

$$\tau_{\text{inv}}^{-1} = |n^i||A(\hat{n}^i)|A_0,$$

where $|A(\hat{n}^i)|$ is the matrix absolute value of the flux Jacobian evaluated in the $\hat{n}^i$ direction. Note, the directional flux Jacobian, $A(\hat{n})$, has a real set of eigenvalues for any $\hat{n}$, as the inviscid problem constitutes a hyperbolic system.

The viscous stabilization parameter is chosen to be

$$\tau_{\text{vis}}^{-1} = \frac{p^2}{h_s^2} K_{ii},$$

where h_s is the shortest edge of an element, p is the interpolation order, and K_{ii} is the sum of the block diagonal entries of the viscosity tensor. The choice of the shortest edge, h_s, as the viscous scaling length follows from the analysis in.[31–33] The p-dependent reduction of the stabilization parameter is motivated by recalling that the role of τ is to recover the dissipation that would have been provided by the subgrid, unresolved features. As pointed out in,[28] p-refinement generates approximate subgrid scale Green's function, and some of the dissipation effects that the subgrid scale has on the resolved scale are captured. Thus, the proposed approach rescales the viscous scaling length based on the interpolation order.

II.C. Discrete Systems

Once a suitable basis for V_h is chosen, a solution $u \in V_h$ can be expressed as $u = U_j \phi_j$, where $\{\phi_j\}_{j=1}^{\text{dof}}$ is the set of basis functions. Then, Eq. 5 can be expressed as a system of ODE's

$$M\frac{dU}{dt} + R(U) = 0,$$

where $R(U)_i = \mathcal{R}_{gal}(u,\phi_i) + \mathcal{R}_{ls}(u,\phi_i)$ are the discrete nonlinear residuals, and $M_h = (\phi_j,\phi_i)_\Omega$ is the mass matrix. The steady state solution to the conservation laws is given at $R(U) = 0$. In order to solve for the steady state, a damped Newton method based on the backward Euler differencing of the ODE is used, i.e.

$$U^{n+1} = U^n + \left(\frac{1}{\Delta t} M + \left. \frac{dR}{dU} \right|_{U=U^n} \right)^{-1} R(U^n).$$

Even though this work is concerned with steady problems, the pseudo-time stepping improves the robustness of the solver for nonlinear equations. After the initial transient, $\Delta t \to \infty$, and the method approaches the full Newton method. The time marching scheme requires the solution of a linear equation at each time step, i.e.

$$\left(\frac{1}{\Delta t} M + \left. \frac{dR}{dU} \right|_{U=U^n} \right) \Delta U^n = -R(U^n).$$

For a small Δt, the linear equation is well-conditioned as the mass matrix dominates. On the other hand, as $\Delta t \to \infty$, the linear system becomes harder to solve as the condition number of the system increases. Note, the linear system solved at each Newton step is equivalent to the system arising from a discretization of advection-diffusion-reaction equation, i.e.

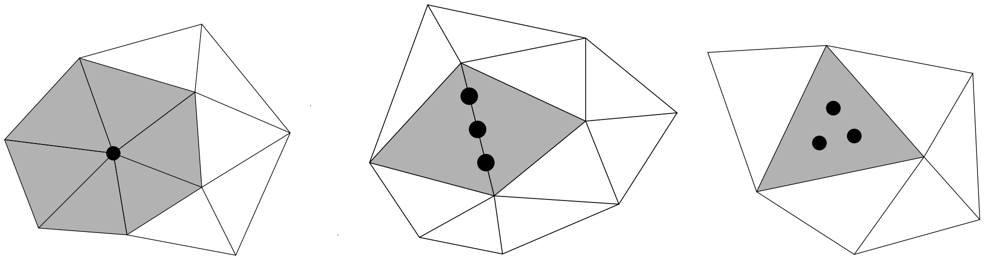
$$\mathcal{R}^{\text{lin}}(u,v) = -\left(A_s u, v_{,x_s}\right)_\Omega + \left(K_{sr} u_{i,x_r}, v_{i,x_s}\right)_\Omega - \left(C_k u, v\right)_\Omega,$$

where $A_s \equiv A + \frac{\partial K_{sr}}{\partial u} u_{,x_r}$ and $C = \frac{\partial f}{\partial u}$. For this reason, the term Jacobian matrix and the stiffness matrix are used interchangeably in the following sections. Similarly, the linear solver algorithm is described using the bilinear form, $a(\cdot,\cdot)$, instead of the linearized form $\mathcal{R}^{\text{lin}}(\cdot,\cdot)$, to be consistent with other literatures on domain decomposition methods.

II.D. High-Order C^0 Basis Functions

Throughout this study, triangular elements with a Lagrange basis are used to construct the approximation space. A set of equally-spaced nodes is used to define the basis functions on a reference element.

High-order geometry representation is achieved through polynomial mapping based on the Lagrange points, i.e. $x = x_j \phi_j(\xi)$ where ϕ_j is the Lagrange basis function corresponding to node j, x_j is the physical coordinate of the j-th Lagrange node, and ξ is the coordinate on the reference triangle. Note, as the mapping $T : \xi \to x$ is nonlinear, the basis functions in the physical space are not polynomials for curved elements.



(a) Node mode (b) Edge mode (c) Element mode

Figure 1. Diagram of basis modes and their support. The degree of freedom for each mode for $p = 4$ discretization is shown in dots.

The basis functions for the high-order continuous Galerkin method must be C^0 continuous across the element boundaries. In order to enforce the continuity, it is convenient to categorize the basis functions into one of the three types of modes in two dimensions, as shown in Figure 1. The first is the node mode, which has support on all elements sharing the node. For a given node, this mode always has a single degree of freedom regardless of the polynomial order. This is the only active mode for $p = 1$ discretization. Except on the boundary, the node mode has the largest support out of all basis types. The second type of basis is the edge mode, which has support on two elements sharing an edge. The degrees of freedom associated with a given edge are equal to $p - 1$. The third type of basis is the element mode, which has support on a single element. The degrees of freedom associated with a given element are equal to $\frac{1}{2}(p-2)(p-1)$. As the element mode has elementally compact support, this mode can be eliminated locally prior to solving the global linear system via static condensation.

III. Balancing Domain Decomposition by Constraints

III.A. The BDDC Preconditioner

Let $\{\Omega_i\}_{i=1}^N$ be a decomposition of domain Ω into N non-overlapping subdomains such that $\Omega_i \cap \Omega_j = \emptyset, i \neq j$ and $\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$. The decomposition is assumed to align with the finite element triangulation, $\mathcal{T}_h$. The interface of a subdomain Ω_i is denoted by Γ_i and defined as $\Gamma_i \equiv \partial\Omega_i \setminus \partial\Omega$. The collection of the subdomain interfaces is denoted by Γ, and is defined as $\Gamma \equiv \cup_{i=1}^N \Gamma_i$. The space of finite element functions on Ω_i is denoted by $V_h^{(i)}$. The subset of these functions that vanish on Γ_i is denoted by $V_I^{(i)}$ and the space of traces on Γ_i of the functions in $V_h^{(i)}$ is denoted by $V_\Gamma^{(i)}$. In order to define the local Schur complement system, the local degrees of freedom are decomposed into the interior part $u_I^{(i)} \in V_I^{(i)}$ and the interface part $u_\Gamma^{(i)} \in V_\Gamma^{(i)}$. The local stiffness matrix, the solution vector, and the load vector are denoted by

$$A^{(i)} = \begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix}, \quad u^{(i)} = \begin{pmatrix} u_I^{(i)} \\ u_\Gamma^{(i)} \end{pmatrix}, \quad \text{and} \quad f^{(i)} = \begin{pmatrix} f_I^{(i)} \\ f_\Gamma^{(i)} \end{pmatrix},$$

where the subscript I and Γ denote the interior and interface degrees of freedom, respectively.

In order to relate the local problem on $V_h^{(i)}$ to the global problem on V_h, let $R^{(i)} : V_h \to V_h^{(i)}$ be the restriction operator that extracts the degrees of freedom associated with $V_h^{(i)}$ from V_h. The global stiffness matrix and the load vector are obtained by assembling the interface degrees of freedom, i.e.

$$A = \sum_{i=1}^N (R^{(i)})^T A^{(i)} R^{(i)} \quad \text{and} \quad f = \sum_{i=1}^N (R^{(i)})^T f^{(i)}.$$

The global system $Au = f$ is solved using the Generalized Minimum Residual (GMRES) algorithm,[34] with the BDDC preconditioner.

In order to define the BDDC algorithm, the space of local interface degrees of freedom, $V_\Gamma^{(i)}$, is decomposed into two parts, i.e. $V_\Gamma^{(i)} = V_\Delta^{(i)} \oplus V_\Pi^{(i)}$. The local primal variables, $V_\Pi^{(i)}$, are a selected few degrees of freedom on the interface, which constitute the global coarse space for the BDDC preconditioner. The local dual variables, $V_\Delta^{(i)}$, consists of functions in $V_\Gamma^{(i)}$ that vanish on the primal degrees of freedom. For convenience, let the product of the interior and the dual space be denoted by $V_r^{(i)} \equiv V_I^{(i)} \oplus V_\Delta^{(i)}$. Then, the local stiffness matrix, the solution vector, and the load vector can be partitioned as

$$A^{(i)} = \begin{pmatrix} A_{rr}^{(i)} & A_{r\Pi}^{(i)} \\ A_{\Pi r}^{(i)} & A_{\Pi\Pi}^{(i)} \end{pmatrix}, \quad u^{(i)} = \begin{pmatrix} u_r^{(i)} \\ u_\Pi^{(i)} \end{pmatrix}, \quad \text{and} \quad f^{(i)} = \begin{pmatrix} f_r^{(i)} \\ f_\Pi^{(i)} \end{pmatrix},$$

where

$$A_{rr}^{(i)} = \begin{pmatrix} A_{II}^{(i)} & A_{I\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{pmatrix}, \quad A_{r\Pi}^{(i)} = \begin{pmatrix} A_{I\Pi}^{(i)} \\ A_{I\Delta}^{(i)} \end{pmatrix}, \quad \text{and} \quad A_{\Pi r}^{(i)} = \begin{pmatrix} A_{\Pi I}^{(i)} & A_{\Pi\Delta}^{(i)} \end{pmatrix}.$$

The space $V_\Pi^{(i)}$ is chosen such that the problem defined on $V_r^{(i)}$ has a unique solution, i.e. $A_{rr}^{(i)}$ is non-singular. For example, for the Poisson equation, it is sufficient to choose the degrees of freedom on the corners of the subdomain as primal variables, which prevents the subdomains from floating. Two coarse primal variable spaces are considered in this study: the first is the space spanned by coarse basis functions defined on the subdomain corners, and the second is the space spanned by coarse basis functions on the subdomain corners and edges. When the edge average constrains are employed, the basis functions of $V_h^{(i)}$ are modified such that all primal and dual degrees of freedom are explicit, using the method described in.[17,35] For either set of the primal variables, the degrees of freedom of $V_\Pi^{(i)}$ is in order of the number of subdomains.

The problem on $A_{rr}^{(i)}$ is refered to as the constrained Neumann problem from hereon, as, for the Poisson equation, the inversion of $A_{rr}^{(i)}$ corresponds to solving the problem with the Neumann boundary condition almost everywhere on Γ, except the boundary values are constrained to vanish at the primal constraints.

The key space used in the BDDC algorithm is the partially assembled space, $\tilde{V}$, defined by

$$\tilde{V} = \left(\oplus_{i=1}^N V_r^{(i)}\right) \oplus V_\Pi, \tag{6}$$

where V_Π is the primal variable space, which is continuous across the interface. Let $\bar{R}^{(i)} : \tilde{V} \to V_h^{(i)}$ be the restriction operator that extracts the degrees of freedom associated with $V_h^{(i)}$ from $\tilde{V}$. The stiffness matrix on $\tilde{V}$ is given by

$$\tilde{A} = \sum_{i=1}^{N} (\bar{R}^{(i)})^T A^{(i)} \bar{R}^{(i)} = \begin{pmatrix} A_{rr}^{(1)} & & & \tilde{A}_{r\Pi}^{(1)} \\ & \ddots & & \vdots \\ & & A_{rr}^{(N)} & \tilde{A}_{r\Pi}^{(N)} \\ \tilde{A}_{\Pi r}^{(1)} & \cdots & \tilde{A}_{\Pi r}^{(N)} & \tilde{A}_{\Pi\Pi} \end{pmatrix} \equiv \begin{pmatrix} A_{rr} & \tilde{A}_{r\Pi} \\ \tilde{A}_{\Pi r} & \tilde{A}_{\Pi\Pi} \end{pmatrix},$$

where

$$A_{rr}^{(i)} \equiv \begin{pmatrix} A_{II}^{(i)} & A_{I\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{pmatrix}, \qquad \tilde{A}_{r\Pi}^{(i)} \equiv \begin{pmatrix} A_{I\Pi}^{(i)} R_\Pi^{(i)} \\ A_{\Delta\Pi}^{(i)} R_\Pi^{(i)} \end{pmatrix},$$

$$\tilde{A}_{\Pi r}^{(i)} \equiv \begin{pmatrix} (R_\Pi^{(i)})^T A_{\Pi I}^{(i)} & (R_\Pi^{(i)})^T A_{\Pi\Delta}^{(i)} \end{pmatrix}, \qquad \tilde{A}_{\Pi\Pi} = \sum_{i=1}^{N} (R_\Pi^{(i)})^T A_{\Pi\Pi}^{(i)} R_\Pi^{(i)}.$$

In other words, $\tilde{A}$ is created by assembling the local spaces with respect to only the primal variables.

In order to inject a function from the partially assembled space, $\tilde{V}_\Gamma$, into the fully assembled space, V_Γ, a scaling operator that takes weighted average of the interface variables must be defined.[13] For this study, the scaling operator $\delta_i^\dagger$ is simply set to $1/\mathcal{N}_x$, where $\mathcal{N}_x$ is the number of subdomains sharing the same degree of freedom on the interface. For an elliptic problem with discontinuous coefficients, it is known that the scaling parameter should be weighted by the coefficient of neighboring subdomains.[36] The scaling factors, $\delta_i^\dagger$, are collected to form a diagonal matrix, $D^{(i)}$, which operates on $V_\Gamma^{(i)}$. The weighted interface restriction operator, $\tilde{R}_{D,\Gamma}^{(i)}$, is obtained by multiplying the rows of $R^{(i)}$ corresponding to the dual variables by $\delta_i^\dagger$ and applying identity mapping to the primal variables.

The BDDC preconditioner is defined by the two set of extension operators and the partially assembled matrix, i.e.

$$M_{\text{BDDC}}^{-1} = \tilde{\mathcal{H}}_1 \tilde{A}^{-1} \tilde{\mathcal{H}}_2, \tag{7}$$

where the extension operators are given by

$$\tilde{\mathcal{H}}_1 = \begin{pmatrix} I & A_{II}^{-1} \tilde{A}_{I\Gamma} J_{D,\Gamma} \\ & \tilde{R}_{D,\Gamma}^T \end{pmatrix} \quad \text{and} \quad \tilde{\mathcal{H}}_2 = \begin{pmatrix} I & \\ J_{D,\Gamma} \tilde{A}_{\Gamma I} A_{II}^{-1} & \tilde{R}_{D,\Gamma} \end{pmatrix}, \tag{8}$$

where $A_{II} \equiv \text{diag}_{i=1}^N \left(A_{II}^{(i)} \right)$ and $J_{D,\Gamma}$ is the jump operator on the interface variables defined by $J_{D,\Gamma} = I_\Gamma - \tilde{R}_{D,\Gamma} R_\Gamma^T$. Recall, because the primal variables V_Π are continuous across the subdomain interfaces, the jump operator maps the primal variables to zero. Thus, the operator $A_{II}^{-1} \tilde{A}_{I\Gamma} J_{D,\Gamma}$ extends the jump in the dual variables to the interior using the discrete harmonic extension, which is the minimum energy extension for the symmetric positive definite systems.[12]

III.B. Implementation of the BDDC Algorithm

A single step of the BDDC preconditioning requires the inversion of the partially assembled matrix, $\tilde{A}$, and the application of extension operators, $\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{H}}_2$. Most of the operations required for the application of these operators parallelizes, making the algorithm suitable for massively parallel architectures.

With a Cholesky-like factorization, the inverse of the partially assembled matrix can be expressed as

$$\tilde{A}^{-1} = \begin{pmatrix} A_{rr}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -A_{rr}^{-1} \tilde{A}_{r\Pi} \\ I \end{pmatrix} S_\Pi^{-1} \begin{pmatrix} -\tilde{A}_{\Pi r} A_{rr}^{-1} & I \end{pmatrix},$$

where $A_{rr} \equiv \text{diag}_{i=1}^N \left(A_{rr}^{(i)} \right)$ and $S_\Pi = A_{\Pi\Pi} - A_{\Pi r} A_{rr}^{-1} A_{r\Pi}$, which is called the primal Schur complement. The application of A_{rr} corresponds to solving the constrained Neumann problem on each subdomain, which parallelizes completely as the continuity across in the subdomain interface is not enforced in $\oplus_{i=1}^{N} V_r^{(i)}$. The inversion of the primal Schur complement is a globally coupled problem; however, the primal Schur complement problem is much smaller than the local problems, as the degrees of freedom associated with V_Π is approximately equal to the number of subdomains when only the corner constrains are employed and is approximately three times the number of subdomains when both the corner and edge average constrains are employed. Moreover, S_Π is sparse as only the primal degrees of freedom that share the same subdomain connect to each other. Thus, in this work, S_Π is factorized exactly.

Similarly, application of the extension operators, $\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{H}}_2$, can be carried out almost completely in parallel; the communication is required only for calculating the action of the jump operator, $J_{D,\Gamma}$. The majority of the cost comes from the application of A_{II}, which corresponds to solving the local Dirichlet problem on each subdomain.

At first glance, the application of $\tilde{A}^{-1}$ seems to require solving the constrained Neumann problem, A_{rr}, twice. However, when the factorization of the local stiffness matrix, $A^{(i)}$, is available, $\tilde{A}^{-1}$ can be applied using a forward substitution, followed by the inversion of S_Π, and finally a backword substitution. Thus, the cost of applying $\tilde{A}^{-1}$ is approximately equal to the cost of solving a single constrained Neumann problem. Similarly, when the factorization of the local problem is available, the extension operators $\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{H}}_2$ can be applied by single backward solve and forward solve, respectively. Thus, the total cost of a single step of the BDDC algorithm is approximately equal to the cost of solving the local problem twice. The implementation details of the BDDC method is discussed in.[37]

III.C. Robin-Robin Interface Condition

As the BDDC preconditioner was originally designed for symmetric positive-definite systems, the method must be modified for nonsymmetric systems. A local bilinear form that conserves the energy stability property of the global form is first derived for the advection equation. Then, the approach is generalized to symmetrized system of conservation laws.

III.C.1. Advection Equation

The variational form of the time-dependent advection equation is: find $u \in V \equiv H^1(\Omega \times I)$, with $I = (0,T)$, such that

$$a(u,v) = \ell(v), \quad \forall v \in V$$

where

$$\begin{aligned} a(u,v) &= (v, u_{,t})_{\Omega\times I} - (v_{,x_j}, \beta_j u)_{\Omega\times I} + (v, \beta_j u n_j)_{\partial\Omega^+\times I} \\ \ell(v) &= (v, f)_{\Omega\times I} - (v, \beta_j g n_j)_{\partial\Omega^-\times I}, \end{aligned}$$

with $(\cdot,\cdot)_{\Omega\times I}$ denoting the L^2 inner product over the space-time domain. The inflow and outflow boundaries are denoted by $\partial\Omega^- = \{x \in \partial\Omega | \beta_j(x) n_j(x) < 0\}$ and $\partial\Omega^+ = \{x \in \partial\Omega | \beta_j(x) n_j(x) > 0\}$, respectively. Integrating by parts the temporal and spatial derivative terms, the energy stability of the advection equation follow from

$$\begin{aligned} &\tfrac{1}{2}\|u|_{t=T}\|^2_{L^2(\Omega)} + \tfrac{1}{2}(u, u\beta_{j,x_j})_{\Omega\times I} + \tfrac{1}{2}(u, u|\beta_j n_j|)_{\partial\Omega\times I} \\ &\qquad = \tfrac{1}{2}\|u|_{t=0}\|^2_{L^2(\Omega)} + (u, f)_{\Omega\times I} + (u, g|\beta_j n_j|)_{\partial\Omega^-\times I}. \end{aligned}$$

In other words, assuming $\beta_{j,x_j} \geq 0$, the solution at $t = T$ is bounded by the initial state, the source function, and the inflow condition.

Consider designing a local bilinear form, $\check{a}(\cdot,\cdot) : V(\check{\Omega}) \times V(\check{\Omega}) \to \mathbb{R}$, with the criterion that the local form maintains the energy conservation property of the global bilinear form on a subet $\check{\Omega} \subset \Omega$. Although the analysis generalizes to the case where $\check{\Omega}$ is any subset of Ω, the setting that is most relevant to the domain decomposition method is the case where the subset is the subdomain ($\check{\Omega} = \Omega_i$) and the local bilinear form acts on the subdomain, i.e. $\check{a}(\cdot,\cdot) = a_i(\cdot,\cdot)$ with $a_i(\cdot,\cdot) : V^{(i)} \times V^{(i)} \to \mathbb{R}$. First, consider the local bilinear form and the linear form defined by restricting the global bilinear form and the linear form, respectively, to the subdomain, i.e.

$$\begin{aligned} \hat{a}_i(u,v) &= (v, u_{,t})_{\Omega_i\times I} - (v_{,x_j}, \beta_j u)_{\Omega_i\times I} + (v, \beta_j u n_j)_{(\partial\Omega_i\cap\partial\Omega^+)\times I} \\ \hat{\ell}(v) &= (v, f)_{\Omega_i\times I} - (v, \beta_j g n_j)_{(\partial\Omega_i\cap\partial\Omega^-)\times I}. \end{aligned}$$

The energy statement arising from the bilinear form $\hat{a}_i(\cdot,\cdot)$ is

$$\begin{aligned}
&\tfrac{1}{2}\|u|_{t=T}\|^2_{L^2(\Omega_i)} + \tfrac{1}{2}(u, u\beta_{j,x_j})_{\Omega_i\times I} + \tfrac{1}{2}(u, u|\beta_j n_j|)_{(\partial\Omega_i\cap\partial\Omega)\times I} \\
&\quad - \tfrac{1}{2}(u, u|\beta_j n_j|)_{\Gamma_i^+\times I} + \tfrac{1}{2}(u, u|\beta_j n_j|)_{\Gamma_i^-\times I} \\
&\qquad = \tfrac{1}{2}\|u|_{t=0}\|^2_{L^2(\Omega_i)} + (u, f)_{\Omega_i\times I} + (u, g|\beta_j n_j|)_{(\partial\Omega_i\cap\partial\Omega^-)\times I}.
\end{aligned} \tag{9}$$

Note, the local problem does not maintain the energy conservation property of the global problem due to the presence of the term $-\frac{1}{2}(u, u|\beta_j n_j|)_{\Gamma_i^+\times I} + \frac{1}{2}(u, u|\beta_j n_j|)_{\Gamma_i^-\times I}$. In particular, the local problem can be unstable due to the negative term on Γ_i^+. In order to preserve the energy stability property of the global problem, the terms on the interface Γ_i must be eliminated by adding $\frac{1}{2}(u, u\beta_j n_j)_{\Gamma_i^+\times I}$ to the interfaces. The resulting bilinear form is

$$a_i(u,v) = \hat{a}_i(u,v) + \tfrac{1}{2}(u, u\beta_j n_j)_{\Gamma_i^+\times I}. \tag{10}$$

As the interface term added to one side of the interface is subtracted from the neighboring subdomain, this does not modify the global bilinear form, i.e. $\sum_i^N a_i(u,v) = a(u,v)$. The same modification to the interface condition is proposed in[22] to make the local bilinear form positive definite. The resulting interface condition is called the Robin-Robin interface condition as the resulting local problem involves the Robin boundary condition. The method has been also successfully applied to solve the advection-diffusion equation using Finite Element Tearing and Interconnect[23] and recently BDDC.[24]

III.C.2. Interface Condition for Symmetrized System of Equations

The Robin-Robin interface condition can be generalized to a system of symmetrized nonlinear equations using the linearized form presented in Section II.C. For a time-dependent hyperbolic system of conservation laws, the semilinear form is expressed as

$$\mathcal{R}(u,v) = (u^{\text{cons}}_{l,t}, v_k)_{\Omega\times I} - (F^{\text{inv}}_{jk}, v_{k,x_j})_{\Omega\times I} + (\hat{F}_k(u, \text{B.C.data}, n), v_k)_{\partial\Omega\times I}.$$

and the linearized equation solved at a given step of the Newton iteration is

$$\mathcal{R}^{\text{lin}}(u,v) = (A_{0kl}u_{l,t}, v_k)_{\Omega\times I} - (A_{jkl}u_l, v_{k,x_j})_{\Omega\times I} + (A_{bc,kl}u_l, v_k)_{\partial\Omega\times I}$$

where $A_0 = du^{\text{cons}}/du^{\text{entropy}}$ is the matrix mapping the entropy symmetrized variables to the conservative variables, A_j is the flux Jacobian in x_j direction, and $A_{bc} \equiv \frac{\partial\hat{F}}{\partial u}$. The energy statement of the global equation is of the form

$$\begin{aligned}
&\tfrac{1}{2}\|u|_{t=T}\|^2_{A_0,\Omega} - \tfrac{1}{2}(A_{0kl,t}u_l, u_k)_{\Omega\times I} + \tfrac{1}{2}(A_{jkl,x_j}u_l, u_k)_{\Omega\times I} - \tfrac{1}{2}(A(n)_{kl}u_l, u_k)_{\partial\Omega\times I} \\
&\quad + (A_{bc,kl}u_l, v_k)_{\partial\Omega\times I} = \tfrac{1}{2}\|u|_{t=0}\|^2_{A_0,\Omega}
\end{aligned}$$

where $\|u\|_{A_0,\Omega} \equiv (A_0 u, u)_\Omega$ is the energy norm weighted by the symmetric positive definite matrix A_0. Note, for a linear system of equations with the boundary flux $\hat{F}_k = A^+_{kl}u_l + A^-_{kl}g_l$, the boundary flux Jacobian is given by $A_{bc} = A^+$, and the integrals on the boundary can be simplified as $-\frac{1}{2}(A(n)_{kl}u_l, u_k)_{\partial\Omega\times I} + (A_{bc,kl}u_l, v_k)_{\partial\Omega\times I} = \frac{1}{2}(|A|u, u)_{\partial\Omega\times I}$.

In order to develop an appropriate local bilinear form, first consider the form obtained by simply restricting the global linearized form to the local space, i.e.

$$\hat{\mathcal{R}}^{\text{lin}}_i(u,v) = (A_{0kl}u_{l,t}, v_k)_{\Omega_i\times I} - (A_{jkl}u_l, v_{k,x_j})_{\Omega_i\times I} + (A_{bc,kl}u_l, v_k)_{(\partial\Omega_i\cap\partial\Omega)\times I}$$

Integrating by parts the temporal and spatial derivative terms and taking advantage of the symmetry of A_0 and A_j, the energy statement for the local bilinear form is

$$\begin{aligned}
&\tfrac{1}{2}\|u|_{t=T}\|^2_{A_0,\Omega_i} - \tfrac{1}{2}(A_{0kl,t}u_l, u_k)_{\Omega_i\times I} + \tfrac{1}{2}(A_{jkl,x_j}u_l, u_k)_{\Omega_i\times I} \\
&\quad - \tfrac{1}{2}(A(n)_{kl}u_l, u_k)_{(\partial\Omega_i\cap\partial\Omega)\times I} + (A_{bc,kl}u_l, v_k)_{(\partial\Omega_i\cap\partial\Omega)\times I} - \tfrac{1}{2}(A(n)_{kl}u_l, u_k)_{\Gamma_i\times I} \\
&\quad = \tfrac{1}{2}\|u|_{t=0}\|^2_{A_0,\Omega_i}
\end{aligned}$$

Again, due to the presence of the term $-\frac{1}{2}(A_{kl}(n)u_l, u_k)_{\Gamma_i \times I}$, the local energy statement is not consistent with the global energy statement. Using the same approach as the advection equation case, the consistent local energy statement is derived by modifying the local linearized form to

$$\mathcal{R}_i^{\text{lin}}(u,v) = \hat{\mathcal{R}}_i^{\text{lin}}(u,v) + \tfrac{1}{2}(A_{kl}(n)u_l, u_k)_{\Gamma_i \times I}.$$

A local semilinear form that has the desired linearized form is

$$\mathcal{R}_i(u,v) = -(F_{jk}^{\text{inv}}, v_{k,x_j})_{\Omega_i} + \tfrac{1}{2}(F_{jk}^{\text{inv}} n_i, v_k)_{\Gamma_i} + (\hat{F}_k(u, \text{B.C.data}, n), v_k)_{\partial\Omega \cap \partial\Omega_i}.$$

Thus, for a system of nonlinear equations with a symmetric flux Jacobian, the Robin-Robin interface condition corresponds to adding half of the nonlinear flux on the interfaces.

IV. Inexact BDDC

For a large scale problem, the cost of computing and storing the exact factorization can be prohibitively high, and the cost only escalate for three-dimensional problem. In this section, the inexact BDDC preconditioner is constructed by replacing the solutions to the Dirichlet and the partially subassembled problem with actions of inexact local solvers. The inexact BDDC algorithms have been previously applied to linear systems arising from the linear finite element discretization of elliptic systems.[19, 20] In,[19] the exact local solvers are replaced by the action of h-multigrid V-cycles and applied to the Poisson equation, for which the multigrid converges uniformly. In,[20] the incomplete factorization is employed to approximate the solutions to local problems.

The local solver used in this work is a two-level multiplicative preconditioner consists of the dual-threshold incomplete factorization (ILUT)[38] and a p-multigrid like coarse correction, following the approach used in[39] for the high-order discontinuous Galerkin discretization. The use of the two-level preconditioner is motivated by the presence of the convective mode and the elliptic mode in the equations of interest. The incomplete factorization, with a proper reordering, have been shown to work well for advection-dominated problems.[39, 40] This work uses the minimum discarded fill (MDF) algorithm introduced in,[41] and generalized to block matrices in.[39] The method orders the unknown that produces the least discarded fill-in first and repeats the process in a greedy manner. The incomplete factorization is performed block-wise to take advantage of the block structure of the stiffness matrix arising from the high-order discretization or system of equations. The Frobenius norm is used to measure the contribution of a given block to the factorization. The coarse correction is performed by projecting the high-order solution to the $p = 1$ space, and solving the coarse problem exactly. The prolongation operator $P : V_{h,1} \to V_{h,p}, p > 1$ is defined by a simple interpolation.

The two-level preconditioner induced by the $p = 1$ variational coarse correction followed by the ILUT smoothing is

$$M_{MG,1}^{-1} = PA_0^{-1}P^T + M_s^{-1}(I - APA_0^{-1}P^T),$$

where $A_0 \equiv P^T A P$ is the coarse correction matrix and M_s^{-1} represents the action of the ILUT preconditioner. Alternatively, the order of the multiplicative projectors can be commutated, resulting in the preconditioner given by

$$M_{MG,2}^{-1} = PA_0^{-1}P^T + (I - PA_0^{-1}P^T A)M_s^{-1}.$$

As shown in Figure 1, the stiffness matrix for the linear finite element discretization is significantly smaller than that for the high-order discretization, making the direct solution to the coarse problem practical at least in two dimensions. While the coarse problem arising from the h-multigrid based on the variational correction is known to be unstable due to the mesh-dependent bilinear form of the GLS discretization,[42, 43] a similar problem has not been observed for the p-based coarse correction despite the presence of the p-dependent stabilization parameter, τ.

IV.A. Inexact Discrete Harmonic Extensions

The inexact discrete harmonic extension operators are given by

$$\tilde{\mathcal{H}}_1 \approx \begin{pmatrix} I & M_{II,1}^{-1}\tilde{A}_{I\Gamma}J_{D,\Gamma} \\ & \tilde{R}_{D,\Gamma}^T \end{pmatrix} \quad \text{and} \quad \tilde{\mathcal{H}}_2 \approx \begin{pmatrix} I & \\ J_{D,\Gamma}\tilde{A}_{\Gamma I}M_{II,2}^{-1} & \tilde{R}_{D,\Gamma} \end{pmatrix}, \tag{11}$$

p	1	2	3	4	5
Elemental basis	3	6	10	15	21
DOF/elem.	0.5	2.0 (4)	4.5 (9)	8.0 (16)	12.5 (25)
Matrix nnz/elem.	3.5	23.0 (6.6)	76.5 (21.9)	188.0 (53.7)	387.5 (110.7)

Table 1. The degrees of freedom associated with high-order continuous Galerkin discretization on an average mesh. Numbers in the parenthesis are the ratios with respect to the $p = 1$ discretization.

where $M_{II,1}^{-1}$ and $M_{II,2}^{-1}$ are approximate solvers. When ILUT is used as the approximate local solver, $M_{II,1}^{-1} = M_{II,2}^{-1} = M_{II,\text{ILUT}}^{-1}$, where $M_{II,\text{ILUT}}^{-1}$ denote the operator obtained by applying the incomplete factorization to A_{II}. Similar to the exact factorization case discussed in Section III.B, the approximate extension operators $\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{H}}_2$ can be applied by a single backward solve and forward solve, respectively, of the approximation factorization.

When ILUT with the $p = 1$ coarse correction (ILUT-$p1$) is used, the preconditioners are given by

$$M_{II,1}^{-1} = PA_{II,0}^{-1}P^T + (I - PA_{II,0}^{-1}P^T A_{II})M_{II,\text{ILUT}}^{-1}$$
$$M_{II,2}^{-1} = PA_{II,0}^{-1}P^T + M_{II,\text{ILUT}}^{-1}(I - A_{II}PA_{II,0}^{-1}P^T)$$

The order of the coarse correction and the smoothing are different for the two cases to exploit the structure of the ILUT factorization generated. Namely, when the projectors are applied in this order, the extension operator $\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{H}}_2$ require a single backward solve and forward solve, respectively, to apply the ILUT smoother, $M_{II,\text{ILUT}}^{-1}$.

Note, due to the block structure of A_{II}, both the ILUT smoothing and the coarse correction can be performed in parallel. In other words, applying the ILUT-$p1$ preconditioner to each local problem results in the ILUT-$p1$ preconditioning of the global problem.

IV.B. Inexact Partially Subassembled Solve

Two types of the inexact partially assembled solvers are considered in this study. The first is based on performing the ILUT factorization on the partially assembled matrix, i.e.

$$\tilde{A}^{-1} \approx \tilde{M}_{\text{ILUT}}^{-1}. \tag{12}$$

The second is based on the ILUT factorization with the $p = 1$ coarse correction, i.e.

$$\tilde{A}^{-1} \approx \tilde{P}\tilde{A}_0^{-1}\tilde{P}^T + \tilde{M}_{\text{ILUT}}^{-1}(I + \tilde{A}\tilde{P}\tilde{A}_0^{-1}\tilde{P}^T), \tag{13}$$

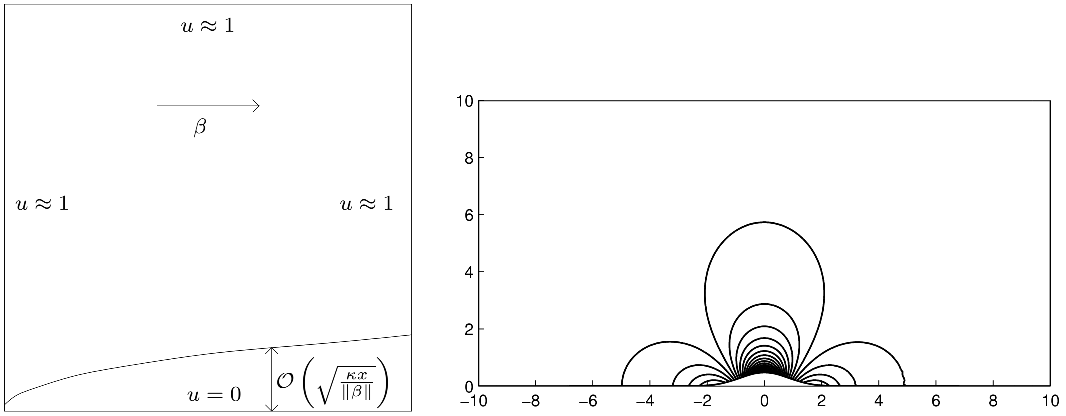
where $\tilde{A}_0 \equiv \tilde{P}^T\tilde{A}\tilde{P}$ is the $p = 1$ correction of the partially assembled system and $\tilde{M}_{\text{ILUT}}^{-1}$ is the ILUT factorization of the partially assembled system. The prolongation operator, $\tilde{P}$, interpolates the $p = 1$ solution on the partially assembled space to the higher-order partially assembled space.

V. Results

V.A. BDDC with Exact Local Solver

V.A.1. Advection-Diffusion Equation: Isotropic Mesh

The advection-diffusion equation is solved on a unit square domain to evaluate the performance of the parallel preconditioner applied to the high-order GLS discretization. The cross-stream boundary layer problem solved has the solution that is essentially 1 everywhere, except it decreases to 0 in the vicinity of the boundary $y = 0$, as shown in Figure 2(a). The boundary layer thickness is inversely proportional to the square root of the Peclet number. The solution is initialized to random values, which results in a random right hand side vector for the linear system, and the system is solved using the BDDC preconditioned GMRES until the 2-norm of the residual reduces by a factor of 10^{13}. Three different values of the viscosity are considered: the diffusion-dominated case ($\kappa = 10^2$), the balanced advection-diffusion case ($\kappa = 10^{-2}$), and the advection-dominated case ($\kappa = 10^{-6}$). The Robin-Robin interface condition is used for all cases.



(a) Cross-stream boundary layer

(b) Euler Gaussian bump

Figure 2. Advection-diffusion problem on a unit square domain and inviscid subsonic flow over a Gaussian bump.

The result of solving the advection-diffusion equation using the BDDC preconditioner using the exact local solver is shown in Table 2. In the upper half of the table, the size of the subdomain (H/h) is fixed to 8×8 (128 triangular elements), and the number of subdomains is varied from four to 256. In the lower half of the table, the number of subdomains is fixed to 16 and the size of the subdomain is varied from $H/h = 4$ to $H/h = 16$. For each case, the interpolation order is varied from $p = 1$ to 4.

(a) Corner constraints only

		$\kappa = 10^2$				$\kappa = 10^{-2}$				$\kappa = 10^{-6}$			
#. Sub.	H/h	$p = 1$	2	3	4	1	2	3	4	1	2	3	4
4	8	5	7	7	7	5	8	8	8	3	3	3	3
16		11	15	17	18	8	10	10	11	7	6	6	6
64		16	21	24	25	11	13	14	16	10	9	8	8
256		17	22	24	27	19	22	23	25	14	13	12	13
16	4	10	13	15	16	8	10	11	11	10	10	9	9
	8	11	15	17	18	8	10	10	11	7	6	6	6
	16	13	17	18	18	8	10	11	12	5	5	5	5

(b) Corner and edge average constraints

		$\kappa = 10^2$				$\kappa = 10^{-2}$				$\kappa = 10^{-6}$			
#. Sub.	H/h	$p = 1$	2	3	4	1	2	3	4	1	2	3	4
4	8	4	6	6	7	6	8	8	8	4	4	4	4
16		7	10	11	12	8	10	10	11	8	7	7	7
64		8	11	12	14	10	12	13	14	11	10	10	9
256		8	10	12	13	10	12	14	16	15	14	14	13
16	4	6	8	10	11	8	10	11	11	10	10	9	9
	8	7	10	11	12	8	10	10	11	8	7	7	7
	16	9	11	12	13	8	10	11	12	6	6	6	6

Table 2. The GMRES iteration count for the BDDC method with the Robin-Robin interface condition for the boundary layer problem on uniform meshes.

In the diffusion-dominated case ($\kappa = 10^2$), the iteration count is independent of the number of subdomains for $N \geq 64$ for the corner constraints case and $N \geq 16$ for the corner and edge average constraints case. When the size of subdomains is varied while keeping number of subdomains fixed, the iteration count increases slowly, as expected from the $(1 + \log(H/h))^2$ dependence of the condition number on the size of the subdomain. In general, using the higher-order interpolation increases the iteration count, but the scalability is maintained for all values of p.

With the Robin-Robin interface condition, the BDDC precondition maintains a low iteration count in the advection-dominated cases. In general, the iteration count increases with the maximum number of the subdomains that a characteristic passes through, N_{char}, as the hyperbolic equation does not possess the smoothing property of the elliptic equation. In particular, the residual does not decay significantly for the first $N_{\text{char}}/2$ iterations, but decays rapidly after $N_{\text{char}}/2$ iterations; this is consistent with the theoretical analysis of the Robin-Robin interface condition conducted in.[22] Note, the iteration count is independent of the size of the subdomain, the interpolation order, and the choice of constraints. Thus, in the advection limit, a partitioning strategy based on capturing the strong characteristics, such as those used in[40] and,[44] is expected to improve the performance of the preconditioner. However, these strategies are sequential in nature and tend to increase the communication volume, especially in three-dimensions.

V.A.2. Advection-Diffusion Equation: Anisotropic Mesh

In realistic problems, highly-anisotropic meshes are employed to resolve the thin boundary layer of high Peclet number flows. The meshes used in this section have the exponential y-spacing such that the elements on the boundary have the aspect ratio approximately equal to $1/\sqrt{\kappa}$. Thus, for $\kappa = 10^{-6}$, the elements on the boundary have the aspect ratio of 1000, and the area of an element on the boundary is less than 1/1000 of the largest element in the domain.

(a) Corner constraints only

		$\kappa = 10^{-2}$				$\kappa = 10^{-4}$				$\kappa = 10^{-6}$			
#. Sub.	H/h	$p = 1$	2	3	4	1	2	3	4	1	2	3	4
4	8	7	9	9	8	7	9	9	10	8	9	10	11
16		9	10	12	13	10	11	13	15	9	11	11	12
64		18	20	21	22	17	18	18	19	14	15	17	19
256		37	40	41	43	45	38	37	36	34	36	41	46
16	4	9	11	10	11	11	11	12	14	12	12	12	12
	8	9	10	12	13	10	11	13	15	9	11	11	12
	16	9	12	13	15	10	12	14	15	9	10	12	13

(b) Corner and edge average constraints

		$\kappa = 10^{-2}$				$\kappa = 10^{-4}$				$\kappa = 10^{-6}$			
#. Sub.	H/h	$p = 1$	2	3	4	1	2	3	4	1	2	3	4
4	8	7	9	9	8	7	9	10	11	7	8	9	10
16		8	10	11	12	9	11	12	13	9	10	11	12
64		12	14	16	17	14	15	17	18	12	13	15	17
256		19	23	25	27	31	26	26	28	26	28	30	32
16	4	8	11	10	10	10	11	12	13	11	12	12	13
	8	8	10	11	12	9	11	12	13	9	10	11	12
	16	9	11	13	14	10	12	13	14	8	10	12	12

Table 3. The GMRES iteration count for the BDDC method with the Robin-Robin interface condition for the boundary layer problem on anisotropic meshes.

Table 3 shows the scalability result using the series of anisotropic meshes. Comparing to Table 2, the iteration count increases in general. In particular, the performance degrades considerably on the fine mesh used for the 256-subdomain partition. The difference in the iteration count for the corner only and the corner and edge average constraints suggests that the problem exhibits elliptic behavior even for $\kappa = 10^{-6}$ with the highly anisotropic mesh.

V.A.3. Euler Equation

The performance of the BDDC method applied to the Euler equation is assessed in this section. The equation is solved on a $(-10, 10) \times (0, 10)$ rectangular domain with a Gaussian bump, $y = 1/(\sigma\sqrt{2\pi}) \exp(-x/(2\sigma^2))$, with $\sigma = 5/6$, as shown in Figure 2(b). The stagnation pressure, stagnation temperature, and the flow angle are specified at the inflow, and the static pressure is specified at the outflow such that the freestream Mach number is $M_\infty = 0.2$. The boundary state for the subsonic inflow and the outflow are reconstructed from the compatibility relations of the Riemann invariants. The flow tangency condition is set on the upper and the lower walls.

The problem is solved using the subdomain-wise block Jacobi, BDDC with corner constraints, and BDDC with corner and edge average constraints. The use of the naturally arising interface condition and the Robin-Robin interface condition are considered. The size of the subdomain is fixed to 160 elements, and increasingly larger problem is solved as more subdomains are added. The linear system arising from the Jacobian for the converged solution is used, and the CFL number is set to infinity so that there is no mass matrix contribution to the linear system.

(a) Naturally arising interface condition

	Subdomain			BDDC			BDDC		
	Block Jacobi			(Corner only)			(Corner + edge)		
#. Sub.	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
2	50	87	113	34	57	75	30	55	74
8	190	359	565	156	284	428	223	276	455
32	914	-	-	580	-	-	476	-	-

(b) The Robin-Robin interface condition

	Subdomain			BDDC			BDDC		
	Block Jacobi			(Corner only)			(Corner + edge)		
#. Sub.	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
2	37	50	63	16	20	21	15	18	20
8	92	155	199	51	77	106	38	66	89
32	209	334	415	90	130	180	52	82	116
128	478	-	-	158	220	309	65	106	159

Table 4. The GMRES iteration count for the Euler bump problem. (160 elem. per subdomain, $\Delta t = \infty$)

Table 4 shows the result of the comparison. For all types of the preconditioners considered, the Robin-Robin interface condition improves the performance of the preconditioner significantly. Unlike the advection-dominated case of the advection-diffusion equation, the iteration count is dependent on the type of constraints for the BDDC method, and using both the corner and edge average constraints reduces the iteration count, especially when a large number of subdomains is employed. The difference suggests the underlying ellipticity of the acoustic modes in steady, subsonic flow.

Elem. per sub.	$p=1$	$p=2$	$p=3$
40	30	49	71
160	38	66	89
640	46	74	100
2560	52	78	102

Table 5. Variation in the GMRES iteration count with the size of subdomains using the BDDC method with the Robin-Robin interface condition. (8 subdomains, corner and edge constraints)

Table 5 shows the variation in the iteration count with the size of the subdomains using eight subdomains for the BDDC method with the Robin-Robin interface condition and the corner and edge average constraints. There is no significant increase in the iteration count as the subdomain size grows, particularly when 640 or more elements are employed per subdomain for all interpolation orders considered.

The GMRES convergence histories for a typical subsonic Euler equation case is shown in Figure 3. The linear residual decays exponentially with the number of iterations. Since the convergence criterion of the linear problem is typically relaxed in the early stage of Newton iteration, the number of GMRES iteration would decrease proportionally in the early stage of the nonlinear solve.

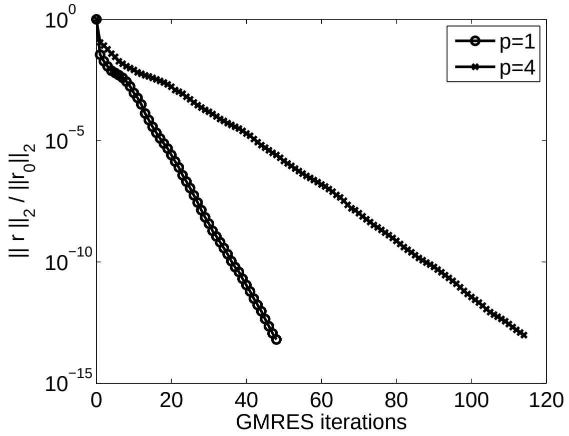


Figure 3. Typical GMRES convergence history for the Euler problem. (32 subdomains, 160 elem. per subdomain, corner and edge constraints)

V.B. BDDC with Inexact Local Solvers

This section presents the result of solving the advection-diffusion equation using the inexact BDDC algorithm. Before using the inexact solvers with the BDDC algorithm, the performance of the inexact solvers on a single subdomain is studied. Table 6 shows the result of solving the linear system arising from the $p = 4$ discretization of the advection-diffusion equations for three different values of viscosity: $\kappa = 10^2, 10^{-2}, 10^{-6}$. In the advection-dominated cases ($\kappa = 10^{-6}$), the ILUT with or without the $p = 1$ coarse correction achieves low iteration count, as the MDF algorithm orders the unknowns along the characteristics and the factorization becomes nearly exact. This is similar to the result obtained for the DG discretization in,[39] despite the worse connectivity structure of the stiffness matrix arising from the high-order GLS discretization. In the diffusion-dominated case, the ILUT with $p = 1$ coarse correction performs significantly better than the method without the coarse correction. In the absence of the coarse correction, the iteration count increases linearly with $1/h$, whereas the iteration count is independent of h with the $p = 1$ coarse correction. Although the bilinear form is dependent of p due to the presence of the p-dependent stabilization term, the variational coarse correction works well for the cases considered.

	ILUT$(10^{-8}, 2)$			ILUT$(10^{-8}, 2)$-$p1$		
$h \setminus p$	$\kappa = 10^2$	10^{-2}	10^{-6}	10^2	10^{-2}	10^{-6}
1/8	22	9	6	10	7	5
1/16	40	11	6	10	7	6
1/32	74	17	7	9	8	6
1/64	145	32	8	8	7	7
1/128	290	66	8	7	7	7

Table 6. The GMRES iteration count for $p = 4$ discretization of the advection-diffusion equation using the ILUT and ILUT with $p = 1$ coarse correction on a single domain.

Table 7 shows the result of applying the inexact BDDC algorithm discussed in Section IV to the $p = 4$ discretization of the advection-diffusion equation. The result of solving the equation using the exact local solver is reproduced for convenience. Similar to the single domain case, the incomplete factorization with or without the $p = 1$ coarse correction achieves low iteration count for the advection-dominated cases. However, in the diffusion-dominated case, the performance of the BDDC algorithm degrade significantly when only the ILUT is employed as the local solver. With the $p = 1$ correction, the inexact BDDC algorithm achieves the similar iteration count as the exact counterpart. The reason the BDDC preconditioner with ILUT-$p1$ local solver achieves lower iteration count than the exact local solver is because the convergence criterions are slightly different for these two cases, as the preconditioned residual is used as the convergence criterion.

		ILUT($10^{-8}, 2$)			ILUT($10^{-8}, 2$)-$p1$			Exact		
# Sub.	H/h	$\kappa = 10^2$	10^{-2}	10^{-6}	10^2	10^{-2}	10^{-6}	10^2	10^{-2}	10^{-6}
4	8	38	12	12	13	9	11	7	8	3
16		73	17	15	19	12	14	18	11	6
64		135	31	19	23	17	18	25	16	8
256		266	61	26	24	24	25	27	25	13
16	4	37	12	17	18	11	16	16	11	9
	8	73	17	15	19	12	14	18	11	6
	16	146	33	14	21	14	14	18	12	5

Table 7. The GMRES iteration count for the $p = 4$ discretization of the advection-diffusion equation using the inexact BDDC method with the Robin-Robin interface condition.

VI. Conclusion

This work presented the high-order accurate Galerkin Least-Squares method combined with massively parallel implicit solvers. The BDDC method was applied to the high-order discretization of the advection-diffusion equation and the Euler equation. In the diffusion-dominated case, the BDDC method achieves the iteration count independent of the number of subdomains, as predicted by the theory. In the advection-dominated case, the method maintains low iteration count when the Robin-Robin interface condition is employed such that the local problem inherit the energy stability of the global problem. The Robin-Robin interface condition was generalized to the Euler equation using the entropy symmetrization theory, and the BDDC method is shown to achieve good performance for the subsonic inviscid flow considered.

The paper also presented the inexact BDDC method based on the inexact local solvers. Two inexact local solvers used in this study are the ILUT preconditioner and the two-level multiplicative preconditioner based on the ILUT and $p = 1$ coarse correction. On a single domain case, the ILUT preconditioner with the minimum discarded fill ordering achieves low iteration count in the advection-dominated case. In the diffusion-dominated case, the precondition achieves the iteration count independent of the grid size when $p = 1$ coarse correction is employed. When combined with the BDDC preconditioner, the ILUT preconditioner with the MDF ordering was sufficient to maintain the low iteration count of the exact counterpart in the advection-dominated case. However, similar to the single domain case, the $p = 1$ coarse correction was necessary to maintain the scalability of the exact counterpart in the diffusion-dominated cases. The inexact BDDC method based on the ILUT-$p1$ local solver appears to be an effective strategy to maintain the performance of the exact counterpart at significantly reduced cost for high-order discretization for the wide range of the flow regime considered.

Acknowledgment

This work was supported by funding from The Boeing Company with technical monitor Dr. Mori Mani.

References

[1] Mavriplis, D. J., Darmofal, D., Keyes, D., and Turner, M., "Petaflops Opportunities for the NASA Fundamental Aeronautics Program," AIAA Paper 2007-4084, 2007.

[2] Meuer, H., Strohmaier, E., Dongarra, J., and Simon, H., "Top500 Supercomputer Sites," 2009.

[3] Hughes, T. J., Engel, G., Mazzei, L., and Larson, M. G., "The continuous Galerkin method is locally conservative," *Journal of Computational Physics*, Vol. 163, 2000, pp. 467–488.

[4] Venkatakrishnan, V., Allmaras, S. R., Kamenetskii, D. S., and Johnson, F. T., "Higher Order Schemes for the Compressible Navier-Stokes Equations," AIAA Paper 2003-3987, 2003.

[5] Hughes, T. J. R., Franca, L. P., and Hulbert, G. M., "A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations," *Comput. Methods Appl. Mech. Eng.*, Vol. 73, 1989, pp. 173–189.

[6] Brooks, A. N. and Hughes, T. J. R., "Streamline upwind / Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer methods in applied mechanics and engineering*, Vol. 32, 1982, pp. 199–259.

[7] Kelly, D. W., Nakazawa, S., and Zienkiewicz, O. C., "A note on upwinding and anisotropic balancing dissipation in finite

element approximations to convection diffusion problems," *International journal for numerical methods in engineering*, Vol. 15, 1980, pp. 1705–1711.

[8] Hughes, T. J. R. and Tezduyar, T. E., "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations," *Comput. Methods Appl. Mesh. Engrg.*, Vol. 45, 1984, pp. 217–284.

[9] Hughes, T. J. R. and Mallet, M., "A new finite element formulation for computational fluid dynamics: III The generalized streamline operator for multidimensional advective-diffusive systems," *Comput. Methods Appl. Mech. Engrg.*, Vol. 58, 1986, pp. 305–328.

[10] Harten, A., "On the symmetric form of systems of conservation laws with entropy," *Journal of computational physics*, Vol. 49, 1983, pp. 151–164.

[11] Hughes, T. J. R., Franca, L., and Mallet, M., "A new finite element formulation for computational fluid dynamics: I Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics," *Comput. Methods Appl. Mech. Engrg.*, Vol. 54, 1986, pp. 223–234.

[12] Toselli, A. and Widlund, O., *Domain Decomposition Methods Algorithm and Theory*, Springer-Verlag, 2005.

[13] Dohrmann, C. R., "A Preconditioner for Substructuring Based on Constrained Energy Minimization," *SIAM Journal on Scientific Computing*, Vol. 25, No. 1, 2003, pp. 246–258.

[14] Mandel, J., "Balancing Domain Decomposition," *Communications in Numerical Methods in Engineering*, Vol. 9, 1993, pp. 233–241.

[15] Bourgat, J.-F., Glowinski, R., Tallec, P. L., and Vidrascu, M., "Variational formulation and algorithm for trace operator in domain decomposition calculations," *Domain decomposition methods. Second international symposium on domain decomposition methods*, edited by T. Chan, R. Glowinski, J. Periaux, and O. Widlund, SIAM, 1988, pp. 3–16.

[16] Mandel, J., Dohrmann, C. R., and Tezaur, R., "An Algebraic Theory for Primal and Dual Substructuring Methods by Constraints," *Applied Numerical Mathematics*, Vol. 54, 2005, pp. 167–193.

[17] Li, J. and Widlund, O. B., "FETI-DP, BDDC, and Block Cholesky Methods," *International Journal for Numerical Methods in Engineering*, Vol. 66, 2006, pp. 250–271.

[18] Brenner, S. C. and yeng Sung, L., "BDDC and FETI-DP without Matrices or Vectors," *Comput. Methods Appl. Mech. Engrg.*, Vol. 196, 2007, pp. 1429–1435.

[19] Li, J. and Widlund, O. B., "On the use of inexact subdomain solvers for BDDC algorithms," *Computational Methods in Applied Mechanics and Engineering*, Vol. 196, 2007, pp. 1415–1428.

[20] Dohrmann, C. R., "An approximate BDDC preconditioner," *Numerical Linear Algebra with applications*, Vol. 14, 2007, pp. 149–168.

[21] Klawonn, A., Pavarino, L. F., and Rheinbach, O., "Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains," *Comput. Methods Appl. Mech. Engrg.*, Vol. 198, 2008, pp. 511–523.

[22] Achdou, Y., Tallec, P. L., Nataf, F., and Vidrascu, M., "A Domain Decomposition Preconditioner for an Advection-Diffusion Problem," *Computer Methods in Applied Mechanics and Engineering*, Vol. 184, 2000, pp. 145–170.

[23] Toselli, A., "FETI Domain Decomposition Methods for Scalar Advection-Diffusion Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, 2001, pp. 5759–5776.

[24] Tu, X. and Li, J., "A Balancing Domain Decomposition Method by Constraints for Advection-Diffusion Problems," *Communications in Applied Mathematics and Computational Science*, Vol. 3, No. 1, 2008, pp. 25–60.

[25] Bassi, F. and Rebay, S., "GMRES Discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontinuous Galerkin Methods: Theory, Computation and Applications*, edited by K. Cockburn and Shu, Springer, Berlin, 2000, pp. 197–208.

[26] Barth, T. J., "Numerical Methods for Gasdynamic Systems on Unstructured Meshes," *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, edited by D. Kroner, M. Olhberger, and C. Rohde, Springer-Verlag, 1999, pp. 195 – 282.

[27] Hughes, T. J. R., "Multiscale Phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods," *Comput. Methods Appl. Mech. Engrg.*, Vol. 127, 1995, pp. 387–401.

[28] Hughes, T. J. R., Feijoo, G. R., Mazzei, L., and Quincy, J.-B., "The variational multiscale method - a paradigm for computational mechanics," *Comput. Methods Appl. Mech. Engrg.*, Vol. 166, 1998, pp. 3–24.

[29] Hughes, T. J. R., "A simple shcme for developing upwind finite elements," *International journal for numerical methods in engineering*, Vol. 12, 1978, pp. 1359–1365.

[30] Hughes, T. J. R., Mallet, M., and Mizukami, A., "A new finite element formulation for computational fluid dynamics: II Beyond SUPG," *Comput. Methods Appl. Mech. Engrg.*, Vol. 54, 1986, pp. 341–355.

[31] Harari, I. and Hughes, T. J. R., "What are C and h? : Inequalities for the analysis and design of finite element methods," *Comput. Methods Appl. Mech. Engrg.*, Vol. 97, 1992, pp. 157–192.

[32] Apel, T. and Lube, G., "Anisotropic mesh refinement in stabilized Galerkin methods," *Numer. Math.*, Vol. 74, 1996, pp. 261–282.

[33] Micheletti, S., Perotto, S., and Picasso, M., "Stabilized finite elements on anisotropic meshes: A priori error estimates for the advection-diffusion and the stokes problems," *SIAM J. Numer. Anal.*, Vol. 41, No. 3, 2003, pp. 1131–1162.

[34] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[35] Klawonn, A. and Widlund, O. B., "Dual-primal FETI methods for linear elasticity," *Communications on Pure and Applied Mathematics*, Vol. 59, 2006, pp. 1523–1572.

[36] Klawonn, A., Widlund, O. B., and Dryja, M., "Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients," *SIAM Journal of Numerical Analysis*, Vol. 43, No. 1, 2002, pp. 159–179.

[37] Yano, M., *Massively Parallel Solver for the High-Order Galerkin Least-Squares Method*, Masters thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, May 2009.

[38] Saad, Y., "ILUT: a Dual Threshold Incomplete LU Factorization," *Numerical Linear Algebra with Applications*, Vol. 1, No. 4, 1994, pp. 387–402.

[39] Persson, P.-O. and Peraire, J., "Newton-GMRES Preconditioning for Discontinuous Galerkin discretizations of the Navier-Stokes Equations," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 2709–2722.

[40] Diosady, L. T., *A Linear Multigrid Preconditioner for the Solution of the Navier-Stokes Equations using a Discontinuous Galerkin Discretization*, Masters thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, May 2007.

[41] D'Azevedo, E. F., Forsyth, P. A., and Tang, W.-P., "Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems," *SIAM J. matrix anal. appl.*, Vol. 13, No. 3, 1992, pp. 944–961.

[42] Okusanya, T. O., *Algebraic Multigrid for Stabilized Finite Element Discretizations of the Navier-Stokes Equations*, PhD dissertation, M.I.T., Department of Aeronautics and Astronautics, June 2002.

[43] Okusanya, T., Darmofal, D., and Peraire, J., "Algebraic multigrid for stabilized finite element discretizations of the Navier-Stokes equations," *Computational Methods in Applied Mechanics and Engineering*, Vol. 193, No. 1, 2004, pp. 3667–3686.

[44] Persson, P.-O., "Scalable Parallel Newton-Krylov Solvers for Discontinuous Galerkin discretizations," AIAA Paper 2009-606, 2009.