Journal of Computational Physics 354 (2018) 344–369

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Approximate tensor-product preconditioners for very high order discontinuous Galerkin methods

Will Pazner [a,*], Per-Olof Persson [b]

[a] Division of Applied Mathematics, Brown University, Providence, RI, 02912, United States
[b] Department of Mathematics, University of California, Berkeley, Berkeley, CA, 94720-3840, United States

ARTICLE INFO

Article history:
Received 15 April 2017
Received in revised form 6 October 2017
Accepted 19 October 2017
Available online xxxx

Keywords:
Preconditioners
Discontinuous Galerkin method
Matrix-free

ABSTRACT

In this paper, we develop a new tensor-product based preconditioner for discontinuous Galerkin methods with polynomial degrees higher than those typically employed. This preconditioner uses an automatic, purely algebraic method to approximate the exact block Jacobi preconditioner by Kronecker products of several small, one-dimensional matrices. Traditional matrix-based preconditioners require $\mathcal{O}(p^{2d})$ storage and $\mathcal{O}(p^{3d})$ computational work, where p is the degree of basis polynomials used, and d is the spatial dimension. Our SVD-based tensor-product preconditioner requires $\mathcal{O}(p^{d+1})$ storage, $\mathcal{O}(p^{d+1})$ work in two spatial dimensions, and $\mathcal{O}(p^{d+2})$ work in three spatial dimensions. Combined with a matrix-free Newton–Krylov solver, these preconditioners allow for the solution of DG systems in linear time in p per degree of freedom in 2D, and reduce the computational complexity from $\mathcal{O}(p^9)$ to $\mathcal{O}(p^5)$ in 3D. Numerical results are shown in 2D and 3D for the advection, Euler, and Navier–Stokes equations, using polynomials of degree up to $p = 30$. For many test cases, the preconditioner results in similar iteration counts when compared with the exact block Jacobi preconditioner, and performance is significantly improved for high polynomial degrees p.

1. Introduction

The discontinuous Galerkin (DG) method, introduced in [29] by Reed and Hill for the neutron transport equation, is a finite element method using discontinuous basis functions. In the 1990s, the DG method was extended to nonlinear systems of conservation laws by Cockburn and Shu [8]. The method has many attractive features, including arbitrarily high formal order of accuracy, and the ability to use general, unstructured meshes with complex geometry. In particular, the promise of a high-order method for fluid flow problems has spurred recent interest in the DG method [26]. Higher-order methods promise highly-accurate solutions for less computational cost than traditional low-order methods. Additionally, high-order methods are more computationally intensive per degree of freedom than corresponding low-order methods, resulting in a higher computation-to-communication ratio, and thus rendering these method more amenable to parallelization [3].

High-order accuracy is achieved with the DG method by using a high-degree local polynomial basis on each element in the mesh. There are several challenges that can prevent the use of very high-degree polynomials as basis functions. The number of degrees of freedom per element scales as $\mathcal{O}(p^d)$, where p is the degree of polynomial approximation, and d is the spatial dimension, resulting in very computationally expensive methods. Using tensor-product evaluations and

* Corresponding author.
E-mail address: will_pazner@brown.edu (W. Pazner).

https://doi.org/10.1016/j.jcp.2017.10.030
0021-9991/

sum factorizations [25], it is possible to reduce the computational cost of these methods, however, the spectrum of the semi-discrete operator grows at a rate bounded above by $(p+1)(p+2)/h$, and well approximated by $(p+1)^{1.78}/h$ where p is the degree of polynomial approximation, and h is the element size [16,38]. As a result, when using explicit time integration schemes, the time step must satisfy a restrictive stability condition given by (approximately) $\Delta t \leq Ch/(p+1)^{1.78}$ [20]. On the other hand, the DG method couples all the degrees of freedom within each element, so that implicit time integration methods result in block-structured systems of equations, with blocks of size $p^d \times p^d$. Strategies for solving these large linear systems include Newton–Krylov iterative solvers coupled with an appropriate preconditioner [28]. Examples of preconditioners considered include block Jacobi and Gauss–Seidel [24], incomplete LU factorizations (LU) [27], and domain decomposition techniques [11]. Multigrid and multi-level solvers have also been considered [15,18,4].

Many of the above preconditioners require the inversion of large the $p^d \times p^d$ blocks corresponding to each element. Using dense linear algebra, this requires $\mathcal{O}(p^{3d})$operations, which quickly becomes intractable. One approach to reduce the computational complexity of implicit methods is to combine Kronecker and sum-factorization techniques with a matrix-free approach. Matrix-free approaches for the DG method have been considered in *e.g.* [9] and [19]. Past work on efficiently preconditioning these systems includes the use of alternating-direction-implicit (ADI) and fast diagonalization method (FDM) preconditioners [12]. Kronecker-product approaches have been studied in the context of spectral methods [31], and applications to the Navier–Stokes equations were considered in [13]. In this work, we describe a new approximate Kronecker-product preconditioner that, when combined with a matrix-free tensor product evaluation approach, allows for efficient solution of the linear systems that arise from implicit time discretizations for high polynomial degree DG methods. This preconditioner requires tensor-product bases on quadrilateral or hexahedral elements. Then, the $p^d \times p^d$ blocks that arise in these systems can be well-approximated by certain Kronecker products of one dimensional $p \times p$ matrices. Using a shuffled singular value decomposition introduced by Van Loan in [34], it is possible to compute decompositions into tensor products of one-dimensional terms that are optimal in the Frobenius norm. Using these techniques, it is possible to construct an approximate tensor-product version of the standard block Jacobi preconditioner, that avoids inverting, or even storing, the large diagonal blocks of the Jacobian matrix.

In Section 2, we give a very brief description of the discontinuous Galerkin method for a general system of hyperbolic conservation laws. In Section 3, we outline the sum-factorization approach, and describe equivalent Kronecker-product representations. Then, in Section 4 we develop the approximate Kronecker-product preconditioners, and provide a new set of algorithms that can be used to efficiently compute and apply these preconditioners. Finally, in Section 5, we apply these preconditioners to several test problems, including the scalar advection equation, compressible Navier–Stokes equations, and the Euler equations of gas dynamics, in two and three spatial dimensions.

2. Equations and spatial discretization

We give a brief overview of the discontinuous Galerkin method for solving a hyperbolic conservation law of the form

$$\partial_t u + \nabla \cdot F(u) = 0. \tag{1}$$

In order to formulate the method, we first discretize the spatial domain Ω by means of a triangulation $\mathcal{T}_h = \{K_j : \bigcup_j K_j = \Omega\}$. Common choices for the elements K_j of the triangulation are simplex and block elements. Given a triangulation $\mathcal{T}_h$, we now introduce the finite element space V_h, given by

$$V_h = \left\{ v_h : v_h|_{K_j} \in V(K_j) \right\}, \tag{2}$$

where $V(K_j)$ is a function space local to the element K_j. Such functions admit discontinuities along the element interfaces ∂K_j. In the case of simplex elements, the local function space $V(K_j)$ is taken to be the space of multivariate polynomials of at most degree p, $\mathcal{P}^p(K_j)$. Of particular interest to this paper are the block elements, which in $\mathbb{R}^d$ are defined as the image of the d-fold Cartesian product of the interval $[0,1]$ under an isoparametric polynomial transformation map.

By looking for a solution $u_h \in V_h$, multiplying by a test function $v_h \in V_h$, and integrating by parts over each element, we derive the weak formulation of (1),

$$\int_{K_j} (\partial_t u_h) v_h \, dx - \int_{K_j} F(u_h) \cdot \nabla v_h \, dx + \int_{\partial K_j} \widehat{F}(u_h^-, u_h^+, n) v_h \, dA = 0, \qquad \text{for all } K_j \in \mathcal{T}_h, \tag{3}$$

where u_h^- and u_h^+ are the interior and exterior traces (respectively) of u_h on ∂K_j, and $\widehat{F}$ is an appropriately defined *numerical flux function*. The integrals in (3) are approximated using an appropriate quadrature rule, and the resulting system of ordinary differential equations is termed the semi-discrete system. In this work, we use quadrature rules that are given by tensor products of one-dimensional quadratures. Typically, using the method of lines, the time derivative (3) is discretized by means of one of many standard (implicit or explicit) methods for solving ordinary differential equations.

3. The sum-factorization approach

In order to numerically represent the solution, we expand the function u_h in terms of basis functions local to each element. In $\mathbb{R}^d$, the number of degrees of freedom n per element thus scales as $\mathcal{O}(p^d)$. In this work, we will make the