Journal of Computational Physics 228 (2009) 1480–1503

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Efficient multi-dimensional solution of PDEs using Chebyshev spectral methods

Keith Julien, Mike Watson *

University of Colorado at Boulder, Dept. of Applied Mathematics, 526 UCB, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 28 January 2008
Received in revised form 26 October 2008
Accepted 28 October 2008
Available online 17 November 2008

MSC:
65F05
65F50
65N35
65N22

Keywords:
Numerical linear algebra
Spectral methods
Partial differential equations
Discretized equations
Chebyshev polynomials
Quasi-inverse

ABSTRACT

A robust methodology is presented for efficiently solving partial differential equations using Chebyshev spectral techniques. It is well known that differential equations in one dimension can be solved efficiently with Chebyshev discretizations, $O(N)$ operations for N unknowns, however this efficiency is lost in higher dimensions due to the coupling between modes. This paper presents the "quasi-inverse" technique (QIT), which combines optimizations of one-dimensional spectral differentiation matrices with Kronecker matrix products to build efficient multi-dimensional operators. This strategy results in $O(N^{2D-1})$ operations for N^D unknowns, independent of the form of the differential operators. QIT is compared to the matrix diagonalization technique (MDT) of Haidvogel and Zang [D.B. Haidvogel, T. Zang, The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, J. Comput. Phys. 30 (1979) 167–180] and Shen [J. Shen, Efficient spectral-Galerkin method. II. Direct solvers of second- and fourth-order equations using Chebyshev polynomials, SIAM J. Sci. Comp. 16 (1) (1995) 74–87]. While the cost for MDT and QIT are the same in two dimensions, there are significant differences. MDT utilizes an eigenvalue/eigenvector decomposition and can only be used for relatively simple differential equations. QIT is based upon intrinsic properties of the Chebyshev polynomials and is adaptable to linear PDEs with constant coefficients in simple domains. We present results for a standard suite of test problems, and discuss of the adaptability of QIT to more complicated problems.

Published by Elsevier Inc.

1. Introduction

1.1. Overview

The study of efficient solution strategies for numerical differential equations with Chebyshev discretizations has been an area of interest for several decades, with entire books dedicated to the subject Boyd [1], Mason [2], Trefethen [3] and Fornberg [4]. Chebyshev polynomials allow for the implementation of general non-periodic boundary conditions unlike Fourier discretizations, but can still make use of Fast Fourier Transforms for conversion between physical and spectral space. These methods also maintain spectral accuracy, which allow for lower resolutions than finite difference methodologies for equivalent accuracies. The utilization of rectangular computational domains continue to arise in many active areas of research, for instance in the geophysical and astrophysical sciences and engineering practices such as computational fluid dynamics, planetary dynamics, mechanics of materials and many others. Even for problems that do not use rectangular

* Corresponding author. Tel. +1 303 870 6907.
E-mail addresses: julien@colorado.edu (K. Julien), watson@colorado.edu (M. Watson).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2008.10.043

domains, decompositions using spectral element schemes are also commonly invoked, which may utilize Chebyshev discretizations on each element [5].

With periodic boundary conditions, spectral Fourier methods are often the solution strategy of choice, because the discretized differential operators are diagonal matrices which can be solved optimally fast in arbitrary dimensions, $O(N)$ operations for N unknown spectral coefficients. When periodicity is lost in a given direction, Chebyshev discretizations can be used, maintaining spectral accuracy and the utility of fast transforms, but the associated differentiation matrices in spectral space are upper triangular, as opposed to diagonal. It is well known that because of the structure of the Chebyshev polynomials, the one-dimensional upper triangular differentiation matrices can be reduced to banded diagonal matrices, with low bandwidth, by taking advantage of the three-term recursion relation. This system can again be inverted in $O(N)$ operations. In higher dimensions, the optimal strategy would be to find a sparse, purely diagonal system which separates in each spatial dimension, comparable to Fourier methods. The three term recursion relationship that leads to efficient solves in 1D prevents optimally fast solutions in higher dimensions, due to non-separability, and other strategies must be employed. Trefethen [3] uses collocation differentiation matrices for a variety of problems. Since collocation methods approximate differential operators in physical space, they are adaptable to many kinds of problems, including non-constant coefficient operators. The primary draw back of collocation techniques is that the differentiation matrices are dense in all dimensions.

For purely constant coefficient linear operators, significant computational saving can be realized via representations in spectral space. Haidvogel and Zang [6] developed a matrix diagonalization technique for the two-dimensional Poisson problem, $\Delta u(x,y) = f(x,y)$, where an eigenvector decomposition of the discretized system is used to reduce the differential equation to N 1D Poisson problems which can each be solved in $O(N)$ operations. This technique has been expanded to higher dimensions via Galerkin basis functions again utilizing Chebyshev polynomials by Shen [7] and more recently to Jacobi polynomials by Doha and Bhrawy [8]. These techniques all require an eigenvalue–eigenvector decomposition of the discretized linear operator, which Haidvogel worried would lead to poor conditioning for large N [6]. In an alternative approach, Dang-Vu and Delcarte [9] utilized the Lanczos Tau method [10] to enforce boundary conditions and exploited the three-term recursive structure of the Chebyshev polynomials to simplify the differential operators. This strategy avoids eigenpair calculations, but the performance is ultimately inhibited by the interaction of Tau lines in higher dimensions, see Section 4. Finally, Heinrichs [11] utilizing a Galerkin basis set analyzed the 1D and 2D Poisson problems to obtain very efficient differentiation matrices. The focus of Heinrichs work was on improving the conditioning number of the spectral operators, not optimizing efficiency. Heinrichs exploited the inherent structure of both the Galerkin differentiation matrices and the relationship between the Chebyshev and Galerkin spectral coefficients to maximize the sparsity and bandedness of his operators. However, Heinrichs work does not provide a systematic methodology for extending his result beyond simple Poisson operators. In this paper we build upon this idea so that it may be generalized to different differential operators and to higher dimensions.

1.2. A new strategy

In this paper, we present a solution strategy which fully exploits the inherent properties of the Chebyshev polynomials and the optimal structure of the 1D differentiation matrices. Further optimization is achieved by eliminating the requirement of Tau lines via the use of Galerkin basis functions for enforcement of boundary conditions. Novel to this methodology is the use of a "quasi-inverse matrix" which acts independently in each discretized spatial dimension, allowing for the optimal representation of differentiation operators in terms of banded diagonal matrices. For example, a "tri-diagonalization" in each coordinate for second-order operators. This is analogous to multi-dimensional Fourier methods which are purely diagonal in each coordinate. Facilitating extensions to multiple dimensions is the extensive use of Kronecker products for separating dimensional interaction utilized by Heinrichs [11] and popularized by Trefethen [3]. The power of this method is that once the one-dimensional differential operator is well characterized, extensions to multiple dimensions are almost trivial. The solution strategy we present is as efficient as the eigenpair matrix diagonalization method of Shen [7] in two dimensions and only slightly less efficient in higher dimensions. Additionally, this new technique remains well conditioned and leads to very sparse matrix systems which may be stored efficiently. Because our "quasi-inverse" methodology is not dependent on eigenpair decompositions, we can solve more general problems, and we present efficient solutions to the 2D and 3D general biharmonic problem, $\Delta^2 u - \alpha\Delta u + \beta u = f$, for which the matrix diagonalization method fails. Although this method realizes its full potential with the use of Galerkin basis functions, it is equally adaptable to standard Tau line solution strategies, which will improve the efficiency of solves and reduce storage requirements. In problems that require the enforcement of complicated boundary conditions where Galerkin basis functions are not available, this may be the strategy of choice.

1.3. Organization

For clarity in understanding the new methodology, the authors have selected, where necessary, a pedagogical tone. Therefore we have included some details which may already be familiar to the experienced user of spectral methods, but are contained here for completeness. The remainder of this paper is organized as follows. In Section 2, we introduce the idea of the "quasi-inverse" in the context of the 1D Poisson problem. We discuss the implementation of Tau lines, and ultimately show why they inhibit performance in higher dimensions. Two key features of matrix systems that can be efficiently inverted are sparsity and minimal bandwidth. Many of the plots that follow show the non-zero elements of the relevant operators to ex-

plain the efficiency of the solution strategy. Section 3 is a brief review of the properties of Kronecker products, which are fundamental to extension of the quasi-inverse technique to higher dimensions. Section 4 discusses how to implement the Tau method in higher dimensions while exploiting quasi-inverses and Kronecker products. In Section 5 we utilize the quasi-inverse concept in conjunction with Galerkin basis functions to develop an efficient solution technique in arbitrary dimensions. Section 6 is used to present numerical results, where we explicitly show the speed and spectral accuracy in 1, 2 and 3 dimensions. Section 7 provides commentary about implementing the methodologies of this paper, notes about mixing spectral and non-spectral differentiation matrices in multiple dimensions, and some additional insight into the power of the Kronecker representation. In the final section, we summarize our results and indicate additional applications for this technique.

1.4. Notation

We summarize the notation within this paper as follows:

- Spatial variables
  - $\mathbf{x} = (x,y) \in R^2$
  - $\mathbf{x} = (x,y,z) \in R^3$
  - $\mathbf{x} = (x_1,x_2,\ldots,x_n) \in R^n$
- Functional representation of variables
  - $u(x)$, function of spatial variable x
  - $f(x,y)$, function of two spatial variables x and y
- Functional representation of differential operators
  - ∂_{x_i}, partial derivative w.r.t. x_i
  - $\Delta_{ND} = \sum_{i=1}^{N} \partial^2_{x_i x_i}$, Laplacian operator in N dimensions
  - $*$, Standard matrix multiplication
  - $\otimes$, Kronecker matrix product
- Discrete spectral representation of variables
  - $\underline{u}$ vector of spectral coefficients associated with spectral modes
- Discrete spectral representation of operators for discrete variable x, with M points in discretization, size = $(M \times M)$
  - $D^p_{x_i}$, "pth" derivative w.r.t. x_i
  - I_{x_i}, Identity matrix w.r.t. x_i
  - $I^{(\pm Z)}_{x_i}$, "quasi-identity" matrix w.r.t. x_i. This is an identity matrix w.r.t. x_i with Z rows of zeroes at the top/bottom for Z +/−, respectively
  - $D^{-p}_{x_i}$, "pth" quasi-inverse for operator $D^p_{x_i}$
  - $E^{(\pm Z)}_{x}$ "Shifted Identity" with ones on the $\pm Z$ sub/super-diagonal in for the x_i variable
  - $S^{\{v\}}_{x_i}$ Stencil Matrix for the unknown v in the x_i spatial direction. The stencil matrix is used to transform between Chebyshev spectral coefficients and Galerkin spectral coefficients

2. Introduction to the "quasi-inverse

2.1. 1D Poisson equation

We begin the analysis with the 1D Poisson equation

$$\Delta_{1D} u(x) = f(x) \tag{1}$$

on the interval $[-1,1]$. Although the solution to 1D Poisson problem is well established, it will serve as the basis for both higher dimensional differential equations and higher-order operators. The problem is discretized in Chebyshev space, accordingly

$$u(x) \approx \sum_{m=0}^{M} u_m T_m(x) \tag{2}$$

$$f(x) \approx \sum_{m=0}^{M} f_m T_m(x) \tag{3}$$

$$\partial_{xx} u(x) \approx \sum_{m=0}^{M} u^{(2)}_m T_m(x) \tag{4}$$

where

$$u^{(2)}_m = \frac{1}{c_m} \sum_{\substack{p=m+2 \\ p+m\ even}}^{M} p(p^2 - m^2) u_p \tag{5}$$

and $c_0 = 2$, $c_m = 1$ for $m > 0$ [12]. The discrete 1D Poisson system of equations is

$$D_x^2 \underline{u} = \underline{f} \tag{6}$$

where the discrete 1D Laplacian operator D_x^2 is an upper triangular matrix with zeros on the main and lower diagonal, see Fig. 1.

The corresponding residual vector is defined as $\underline{R} = (R_0, R_1, \ldots, R_M) \equiv D_x^2 \underline{u} - \underline{f}$. To increase the efficiency, the original system $D_x^2 \underline{u} = \underline{f}$ is replaced with a less costly system $A\underline{u} = B\underline{f}$, where A and B have banded structure. By taking linear combinations of rows of D_x^2, we can eliminate the upper triangular operator and replace it with a diagonal operator with ones on the main diagonal and zeros in the top two rows, which we define as $I_x^{(2)}$. This action of taking linear combinations of rows, resulting in the quasi-identity matrix $I_x^{(2)}$, can be expressed as a tri-diagonal matrix B, whose entries are derived from the recursion relation for Chebyshev polynomials [12]

$$c_{n-1} u_{m-1}^{(q)} - u_{m+1}^{(q)} = 2m \cdot u_m^{(q-1)} \tag{7}$$

where q is the order of the derivative. For the second-order operator, this result is well known [9], and the matrix B has entries

$$\left.\begin{aligned} b_{i,i-2} &= \frac{c_{i-2}}{4i(i-1)}, \text{2nd sub-diagonal} \\ b_{i,i} &= -\frac{e_{i+2}}{2(i^2-1)}, \text{main diagonal} \\ b_{i,i+2} &= \frac{e_{i+4}}{4i(i+1)}, \text{2nd super-diagonal} \end{aligned}\right\} \quad 2 \leqslant i \leqslant M \tag{8}$$

where c_i is defined as before and $e_i = 1$ for $i \leqslant M$, $e_i = 0$ for $i > M$. The matrix B acts as a "quasi-inverse" for the second-order 1D Laplacian, such that $B * D_x^2 = I_x^{(2)}$. From this relationship, we explicitly define the quasi-inverse matrix:

Definition 1. DEFINITION of the Quasi-Inverse Matrix: $D_{x_i}^{-P}$ is the quasi-inverse matrix of order P associated with the spectral differentiation matrix $D_{x_i}^P$ in the x_i spatial direction such that $D_{x_i}^{-P} * D_{x_i}^P \equiv I_{x_i}^{(P)}$ and $D_{x_i}^P * D_{x_i}^{-P} \equiv I_{x_i}^{(-P)}$. $I_{x_i}^{(P)}$ is the identity matrix with P rows of zeros at the top of the matrix and $I_{x_i}^{(-P)}$ is the identity matrix with zeros in the bottom P rows.

There are several important properties of the quasi-inverse operator which we note here:

(1) The quasi-inverse $D_{x_i}^{-P}$ is not the true inverse of the differential operator $D_{x_i}^P$ because this differentiation matrix is singular and therefore does not have a well defined inverse.
(2) The order of the operator P is the same size of the null space of the differential operator, and indicates the degrees of freedom that need to be satisfied by boundary conditions.
(3) A necessary condition of the definition of the quasi-inverse is that the matrix $D_{x_i}^{-P}$ has zeros in the first P rows and the last P columns.
(4) The non-zero entries of D_x^{-P} are defined analytically by the three term recursion relation derived from the basis polynomials, for Chebyshev polynomials (7).
(5) The action of the quasi-inverse is independent of boundary conditions.

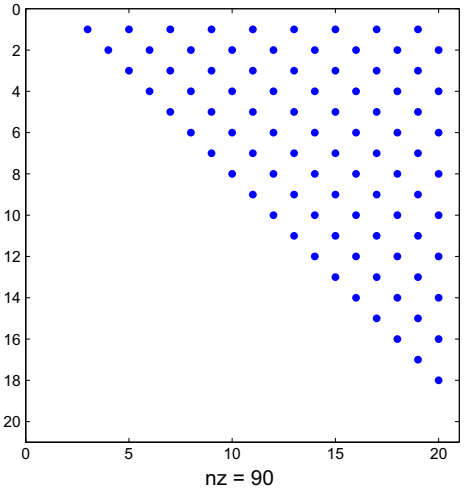


Fig. 1. Non-zeros elements of the 1D Poisson operator D_x^2 for $M = 20$. The nz = 90 is standard output from the MATLAB "spy" routine used to generate this plot, indicating that there are 90 non-zero elements out of 400 in this matrix.

(6) Structure from the basis polynomials translates naturally to the quasi-inverse representation such that there is a well defined structure between different order operators,
(a) $D_x^2 = D_x * D_x$
(b) $D_x^{-P} = I_x^{(P)} * D_x^{-P} * I_x^{(-P)} = I_x^{(P)} * (D_x^{-1})^P * I_x^{(-P)}$
(c) $D_x^{-P} D_x^Q \equiv I_x^{(P)} D_x^{-P+Q}$
We emphasize that property 6(c) defines the correct relation for matrix products equivalent to the analytic result from the three term recursion relation.

After multiplying both sides of the equation $D_x^2 \underline{u} = \underline{f}$ by the quasi-inverse $B = D_x^{-2}$, we get a simplified system of equations

$$\begin{aligned} A\underline{u} &= B\underline{f} \\ &\equiv I_x^{(2)} \underline{u} = D_x^{-2} \underline{f} \end{aligned} \tag{9}$$

This system has two rows of zeros at the top, where Tau lines (Section 2.1.1) can be substituted for enforcement of boundary conditions, see Fig. 2.

Since we know the form of the system analytically, we need never explicitly perform the pre-multiplication step of the quasi-inverse, instead directly solving the system $A\underline{u} = B\underline{f}$ in place of $D_x^2 \underline{u} = \underline{f}$. We can solve this system of equations using Gaussian elimination in $O(M)$ operations. Thus, by taking advantage of the structure of the operator, the complexity of the solve is reduced from $O(M^2)$ to $O(M)$. For the 1D problem, this is not a new result, but we will find this methodology useful for extension to more complicated examples.

It is the identification of the quasi-inverse operator B that we will use to solve more complicated systems in higher dimensions. The strategy will be as follows:

(1) Discretize the differential equation to obtain $\mathcal{L}(D)\underline{u} = \underline{f}$.
(2) Identify the correct quasi-inverse B for the highest order operator.
(3) Multiply the system on both sides by B to obtain a pre-multiplied system $A\underline{u} = B\underline{f}$.
(4) Determine the appropriate boundary conditions
(5) Solve the simplified equation set for $\underline{u}$.

2.1.1. Enforcement of boundary conditions via the Lanczos Tau method

Again consider the Poison problem in one dimension, Eq. (1). Note this system is singular because boundary conditions have not yet been enforced. The two rows of zeros at the bottom the matrix D_x^2, Fig. 1, provide a natural location for implementation of the Lanczos Tau method [10], where the two highest residual modes R_{M-1} and R_M are discarded in order to enforce the boundary conditions. For Dirichlet boundary conditions, $u(1) = \alpha$ and $u(-1) = \beta$, the appropriate restrictions are

$$\sum_{m=0}^{M} u_m = \alpha, \quad \sum_{m=0}^{M} (-1)^m u_m = \beta \tag{10}$$

where α and β replace f_{M-1} and f_M on the right-hand side of system (6), respectively. We note that the Tau method allows for more complicated boundary conditions than simply Dirichlet. For enforcement of higher-order derivatives, we find the closed form expression

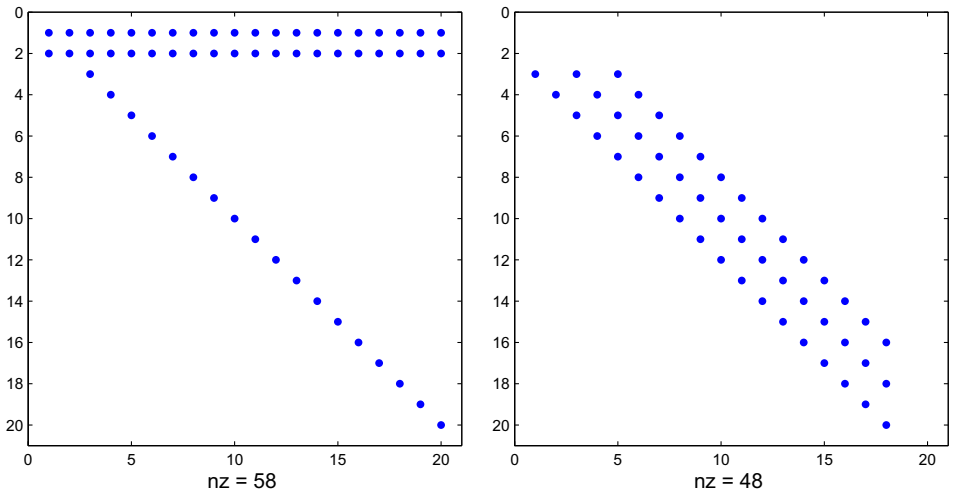


Fig. 2. The non-zero elements of the discrete 1D Poisson system of equations in (9) with Tau enforcement of boundary conditions: A (left) and B (right).

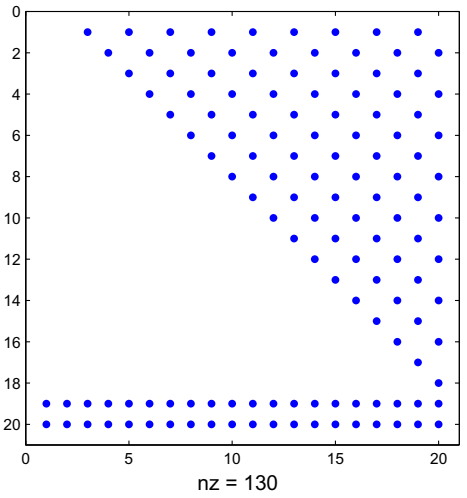


Fig. 3. Non-zeros elements of the 1D Poisson operator D_x^2 with Tau lines at the bottom.

$$u(\pm 1)^{(q+1)} = \frac{2^q (q!)}{(2q+1)!} \sum_{m=0}^{M} \prod_{p=0}^{q} (m^2 - p^2)(\pm 1)^{(m+q-1)} u_m \tag{11}$$

where q is the order of the derivative to be enforced at each endpoint. If we look at the non-zero entries of the discrete Poisson system with Dirichlet Tau lines, we find the structure in Fig. 3.

The bottom two lines are the Tau lines and the upper triangular portion represents the differentiation matrix D_x^2. This matrix is relatively full, and with rearrangement of the Tau lines would cost $O(M^2/2)$ operations to solve for $\underline{u}$.

2.2. 1D Helmholtz equation

It is important to understand how to implement the methodology of quasi-inverses for more general classes of 1D operators. A simple generalization of the Poisson equation is made by adding a scalar multiple λ of the unknown function $u(x)$, resulting in the Helmholtz equation

$$\begin{aligned} &\Delta_{1D} u(x) - \lambda u(x) = f(x) \\ &u(\pm 1) = 0 \\ &x \in [-1, 1] \end{aligned} \tag{12}$$

with λ real. Discretizing this equation, we find the system of equations

$$(D_x^2 - \lambda I_x) * \underline{u} = \underline{f} \tag{13}$$

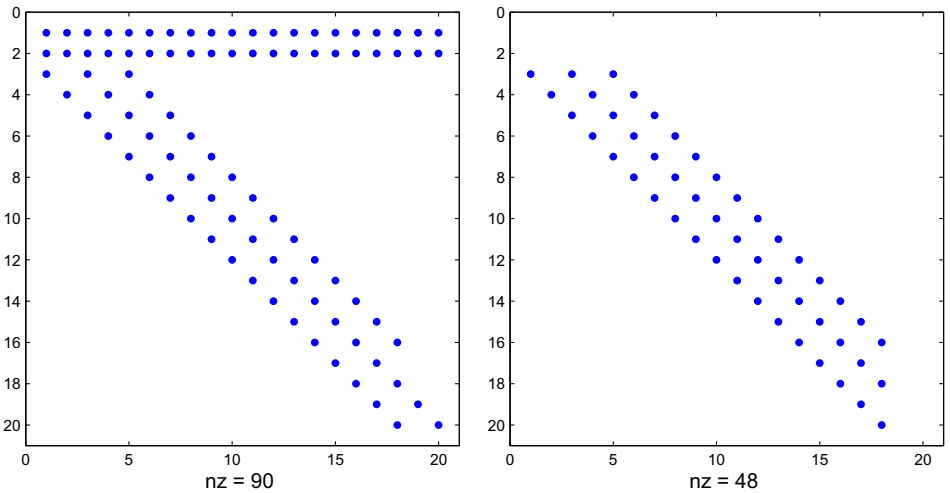


Fig. 4. 1D pre-multiplied Helmholtz operator (left) and quasi-inverse (right).

It is not beneficial at this point to define a new quasi-inverse based on the left-hand side operator $D_x^2 - \lambda I_x$ such that $B * (D_x^2 - \lambda I_x) \equiv I_x^{(2)}$, for B will be an upper triangular matrix which will result in the same computational cost as the original system Eq. (13). Instead, we utilize the quasi-inverse we have already identified for the highest order operator from the Poisson problem. From the Poisson problem we have: $B \equiv D_x^{-2}$. Pre-multiplying with the quasi-inverse, we find the simplified discretized system for the Helmholtz problem

$$(I_x^{(2)} - \lambda D_x^{-2}) * \underline{u} = D_x^{-2} \underline{f} \tag{14}$$

This is our pre-multiplied system of equations. Looking at the non-zero elements of the system after including Tau lines, Fig. 4, we see that the only change from the Poisson problem is that now we must back solve a tri-diagonal system of equations as opposed to a diagonal system. This can still be performed in $O(M)$ operations using Gaussian elimination.

This example illustrates that a new quasi-inverse does not need to be defined for each problem. Instead, the correct quasi-inverse $D_{x_i}^{-2}$ is defined by the highest order operator $P = 2$ in the differential equation. If we considered the generalized biharmonic problem $\partial_{xxxx}u(x) - \alpha\partial_{xx}u(x) + \beta u(x) = f(x)$ as in Section 5.5, the appropriate quasi-inverse is the one associated with the discrete fourth-order operator D_x^4.

3. Kronecker products

The foundation of our extension to higher dimensions is dependent on the utilization of the Kronecker product for matrices, denoted by "$\otimes$". A good review of properties of the Kronecker product can be found in [13].

If A is a $(M \times N)$ matrix and B is a $(P \times Q)$ matrix, then $A \otimes B$ is a $(MP \times NQ)$ matrix with entries

$$A \otimes B \equiv \begin{bmatrix} a_{11}B & \cdots & a_{1N}B \\ \vdots & \ddots & \vdots \\ a_{M1}B & \cdots & a_{MN}B \end{bmatrix} \tag{15}$$

The primary property that we exploit in our implementation is the distributive property: if $A * C$ exists and $B * D$ exists then

$$(A \otimes B) * (C \otimes D) = (A * C \otimes B * D) \tag{16}$$

We utilize the Kronecker notation because it provides for a clear separation of operators in multiple dimensions. For instance, consider the discretization of the 2nd derivative operator in 1D, 2D and 3D shown below:

- 1D $u_{xx}(x) \rightarrow D_x^2 * \underline{u}$
- 2D $u_{xx}(x,y) \rightarrow (D_x^2 \otimes I_y) * \underline{u}$
- 2D $u_{yy}(x,y) \rightarrow (I_x \otimes D_y^2) * \underline{u}$
- 2D $u_{xy}(x,y) \rightarrow (D_x^1 \otimes D_y^1) * \underline{u}$
- 3D $u_{xx}(x,y,z) \rightarrow (D_x^2 \otimes I_y \otimes I_z) * \underline{u}$
- 3D $u_{yz}(x,y,z) \rightarrow (I_x \otimes D_y^1 \otimes D_z^1) * \underline{u}$, etc.

The action of the x differentiation matrix is clearly separated from the y discretization in this notation and similarly for z. This representation is key for the extension of the quasi-inverse concept to higher dimensions. For dimensions higher than one, the unknown matrix $\underline{u}$ is stretched into a single vector. For a detailed discussion of the practical implementation of Kronecker products with respect to collocation differentiation matrices, see Trefethen [3]. As quick example of the utility of this notation, consider the 2D Laplacian with periodic boundaries in y and Dirichlet boundaries in x,

$$\begin{aligned} &\Delta u(x,y) = u_{xx}(x,y) + u_{yy}(x,y) = f(x,y) \\ &u(x,\pm 1) = 0 \\ &u(1,y) = u(-1,y) \\ &x,y \in [-1,1]^2 \end{aligned} \tag{17}$$

The periodicity in x naturally suggests an equi-spaced discretization which leads to Fourier modes and we are constrained to pick Chebyshev nodes in y. The discretized functions take the form

$$u(x,y) \approx \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=0}^{N} u_{m,n} e^{imx} T_n(y) \tag{18}$$

Next we look at the discrete form of the x derivative operator for the Fourier modes, which is a diagonal matrix D_x^2 with entries $d_{kk} = (k-1)^2$ where $k = 1 \ldots M+1$. For the Chebyshev discretization, the operator is the upper triangular differentiation matrix from the previous section (see Eq. (5)), which we denote D_y^2. In discretized form the 2D Poisson equation is

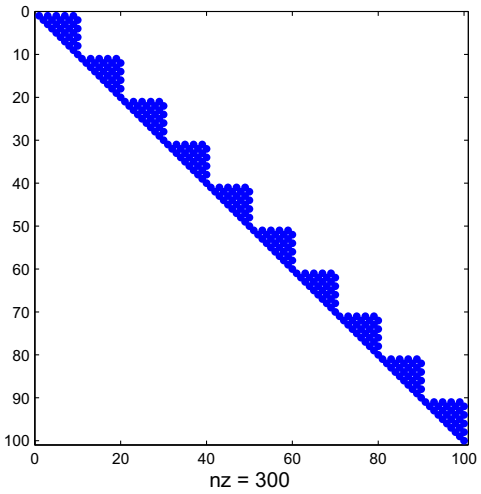


Fig. 5. 2D Poisson differentiation operator with Chebyshev nodes in y and Fourier nodes in x, for $N = 10$. $M = 10$ for 10,000 matrix elements.

$$(D_x^2 \otimes I_y + I_x \otimes D_y^2) * \underline{u} \equiv A * \underline{u} = \underline{f}. \tag{19}$$

A lot insight can be gained by looking at the form the 2D operator A, whose non-zeros entries are shown in Fig. 5.

The block diagonal structure seen here indicates that there is no communication between x modes for this operator which means that each x mode may be treated independently and further simplified as described above. This suggests that this 2D problem can be solved as M individual 1D Helmholtz problems in y, which is a familiar result Section 2.2. By visualizing the structure of the operator, we can gain immediate insight into the behavior of the system. Analytically, the structure of the Kronecker product provides a convenient way for keeping track of interaction between dimensions.

If the x direction had not been periodic, the only change to our discrete equation $(D_x^2 \otimes I_y + I_x \otimes D_y^2) * \underline{u} = \underline{f}$ would be to interchange the diagonal differentiation matrix D_x^2 associated with Fourier modes with the upper triangular Chebyshev differentiation matrix. Ignoring the related Tau lines for boundary conditions, this operator has the form seen in Fig. 6.

We can clearly see the coupling between modes in this figure in each spatial direction, resulting from the non-separability of the discretized spatial dimensions, since there is no longer a block diagonal structure indicating confinement of an operator to a single region in spectral space.

4. Tau line extensions to higher dimensions

We now investigate the discretization and boundary condition enforcement in two dimensions using Tau lines. The quasi-inverse method is used to simplify the differential portion of the discrete operator where the action of the quasi-inverse re-

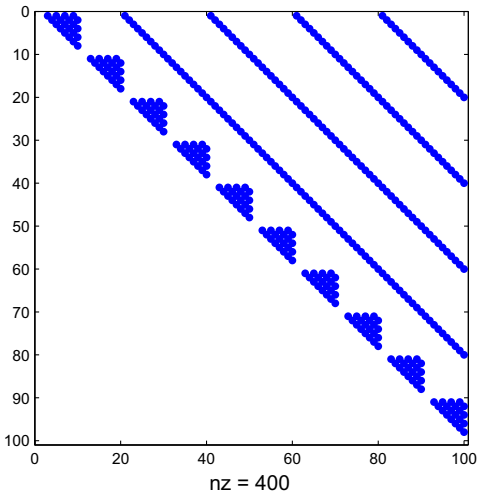


Fig. 6. 2D Poisson differentiation operator with Chebyshev nodes in x and y.

mains independent of the boundary conditions. The enforcement of boundary conditions ultimately communicate across spatial dimension and lead to increased computational costs.

4.1. 2D Poisson equation

We now analyze the Poisson equation in two dimensions with Dirichlet boundary conditions

$$\begin{aligned} &\Delta_{2D}u(x,y) = u_{xx}(x,y) + u_{yy}(x,y) = f(x,y) \\ &u(\pm 1, y) = u(x, \pm 1) = 0 \\ &x, y \in [-1,1]^2 \end{aligned} \tag{20}$$

The functions are discretized as truncated series of Chebyshev polynomials in each spatial dimension

$$u(x,y) \approx \sum_{m,n=0}^{M,N} u_{mn} T_m(x) T_n(y) \tag{21}$$

We discretize the operators by employing the Kronecker product notation, thus

$$(D_x^2 \otimes I_y + I_x \otimes D_y^2) * \underline{u} = \underline{f} \tag{22}$$

where D_x^2 and D_y^2 are the upper triangular differentiation matrices from the 1D problem for both the x and y discretizations. I_x and I_y are identity matrices associated with the x and y discretizations, respectively. For differential equations with operators in more than one dimension, we must identify a quasi-inverse for each dimension. However, we can easily take advantage of the one-dimensional solution strategy by utilizing the distributive property of the Kronecker product (16). In each dimension, the appropriate quasi-inverse is identical to that derived for the corresponding one-dimensional case (see Section 2.1). For the 2D Laplacian, the quasi-inverse is $B = D_x^{-2} \otimes D_y^{-2}$, where D_x^{-2} is the quasi-inverse for D_x^2 and D_y^{-2} is the quasi-inverse for D_y^2. Note, that the number of modes/grid-points associated with each spatial discretization are independent. Applying the quasi-inverse to both sides of (22), we find

$$([I_x^{(2)} \otimes (D_y^{-2} * I_y) + (D_x^{-2} * I_x) \otimes I_y^{(2)}] * \underline{u} = (D_x^{-2} \otimes D_y^{-2}) * \underline{f} \tag{23}$$

Enforcing boundary conditions as Tau lines to each operator matrix D_x^{-2} and D_y^{-2} individually, we find the resulting system, Fig. 7. This system has many of the same characteristics as the 1D problem.

First we note that there are two distinct sets of Tau lines within the operator; due to the ordering of discretization in the Kronecker notation, the smaller Tau lines embedded within the block diagonal structure correspond to the y boundary conditions, and the Tau lines extending across the top of the full matrix correspond to x boundary conditions. Using standard Gaussian elimination, the cost of solving a banded diagonal system is equal to the bandwidth P times the number of unknowns. While there exists some nice banded structure for the "inner" part of the matrix, the x boundary conditions spread the bandwidth across the whole system. This is the primary computational cost that must be paid for the ease of implementation of the Tau lines. As we go to higher dimensions, the Tau lines become more and more expensive as they spread across more unknowns. For the 3D Poisson problem with $L \times M \times N$ unknowns, the cost for this solve tends towards $O \sim (LMN)^2$. It

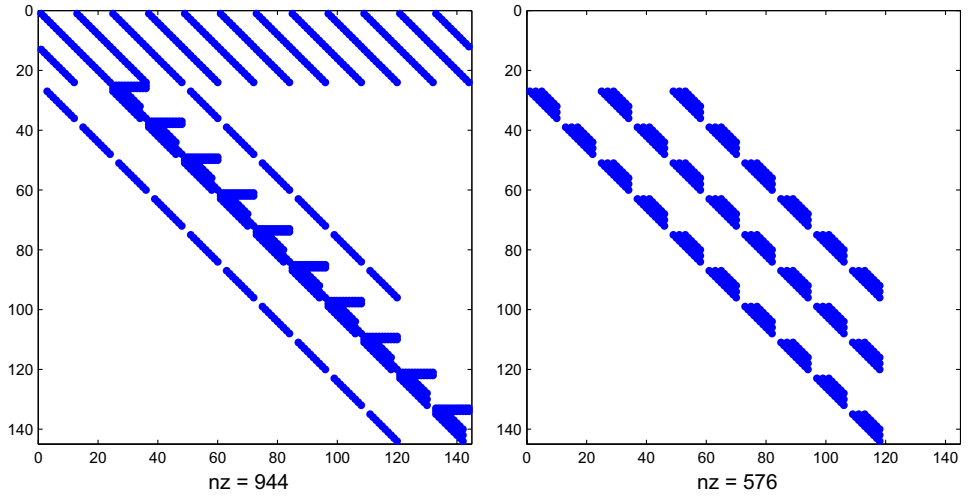


Fig. 7. (Left) 2D pre-multiplied Poisson operator A for $M = 12$, $N = 12$ for 20, 736 matrix elements. The lines extending across the top of the matrix are the boundary conditions in x and the smaller lines in the block diagonal are the boundary conditions in y. (Right) quasi-inverse B for the 2D Poisson operator.

is clear that it is the interaction with the Tau lines across several dimensions that is increasing the cost of the solve, so we are motivated to find an algorithm that does not depend on Tau lines for boundary condition enforcement, namely, the Galerkin basis approach.

5. Galerkin methods

5.1. Galerkin basis

The idea behind a Galerkin basis set is simple: utilize basis functions that satisfy the boundary conditions, and by extension any approximate solutions constructed using these functions automatically satisfy the boundary conditions. We first consider Dirichlet boundary conditions, but this time represent our solution $u(x)$ in terms of a Galerkin basis set. Given the set of Galerkin functions, $\phi_0(x),\cdots,\phi_{M-2}(x)$, we define the Galerkin decomposition

$$u(x) \approx \sum_{m=0}^{M-2} v_m \phi_m(x) \tag{24}$$

$$\phi(\pm 1) = 0 \tag{25}$$

Henceforth, u_m will denote Chebyshev spectral coefficients and v_m will denote Galerkin spectral coefficients. There are any number different basis function $\phi(x)$ that can be chosen, but we focus on linear combinations of Chebyshev polynomials so that we can continue to take advantage of fast transforms between physical and spectral representations. Therefore, we can discretize $u(x)$ in two separate ways.

$$u(x) = \begin{cases} \sum_{m=0}^{M} u_m T_m(x) \\ \sum_{m=0}^{M-2} v_m \phi_m(x) \end{cases}$$

Since the coefficients v_m are linear combinations of the Chebyshev coefficients, it natural to look for relationships between the two spectral representations. Three possible Galerkin basis sets satisfying Dirichlet boundary conditions are

- 1 $\phi_m(x) = \begin{cases} T_{m+2}(x) - T_0(x) & m \text{ even} \\ T_{m+2}(x) - T_1(x) & m \text{ odd} \end{cases}$
- 2 $\phi_m(x) = T_{m+2}(x) - T_m(x)$
- 3 $\phi_m(x) = T_{m+2}(x) - 2^* T_m(x) + T_{m-2}(x)$

Shen [7] points out that first set of basis functions lead to undesirable "ill-conditioning" properties and should be avoided. The latter basis is used by Trefethen [3] in solving the Helmholtz problem. We focus on the second set for satisfying Dirichlet boundary conditions. We also note that Zebib [14] makes use of a novel Galerkin representation derived from a truncated Chebyshev series of the highest order derivative, which could also be employed here. What we wish to find is a linear map, $S_x^{\{v\}}$, between the two sets spectral coefficients $\underline{u}$ and $\underline{v}$, so that we may efficiently transform between the two representations. To express the relationship between $\underline{u}$ and $\underline{v}$ mathematically, we project onto each $T_m(x)$ mode by applying inner products to each side defined by

$$\langle T_p(x), T_m(x)\rangle = c_p \delta_{p,m} \tag{26}$$

where $\delta_{p,m}$ denotes the Kronecker delta function. We begin by looking at the inner product of $\langle \phi_k(x), T_m(x)\rangle$, where $\phi_p(x) = T_{p+2}(x) - T_p(x)$. The inner product is a linear operator so

$$\begin{aligned} \langle \phi_p(x), T_m(x)\rangle &= \langle T_{p+2}(x), T_m(x)\rangle - \langle T_p(x), T_m(x)\rangle \\ &= c_{p+2}\delta_{p+2,m} - c_m \delta_{p,m} \end{aligned} \tag{27}$$

In matrix form we can express this inner product relation in terms of shifted identity matrices; $E^{(k)} \equiv [e_{p,m} \equiv \delta_{p,m-k}]$. For $k > 0$ this defines a square matrix $M \times M$ with ones on the kth super-diagonal, whereas for $k < 0$, this defines a square matrix $M \times M$ with ones on the kth sub-diagonal. It follows that the identity matrix $I_x = [\delta_{p,m}]$ is equivalent to $E_x^{(0)}$. By extension, a matrix with entries defined by $[e_{p,m} \equiv \delta_{p,m+2}]$ is represented by $E_x^{(-2)}$, corresponding to ones along the second sub-diagonal. We thus define a "stencil" matrix $S_x^{\{v\}} = E_x^{(-2)} - E_x^{(0)}$ for Dirichlet boundary conditions in x for the function v. This yields a map between the Chebyshev and Galerkin spectral representation such that $\underline{u} = S_x^{\{v\}}\underline{v}$. We refer to this matrix as a stencil because this matrix describes the linear combinations of Chebyshev modes used for a particular Galerkin basis set. Since this stencil matrix is size $M \times M$, we are required to pad $\underline{v}$ with two fictitious modes v_{M-1} and v_M, which we specify to be identically zero. In situations where there is only one unknown function, we drop the superscript on the stencil matrix for clarity, $S_x^{\{v\}} \equiv S_x$.

Clearly different boundary conditions will yield different stencil matrices, and as the three examples in Section 5.1 indicate, the choice of Galerkin basis functions is not unique to the specific boundary conditions. In Appendix A, we show how to quickly derive Galerkin basis functions utilizing the Tau line expressions defined by Eq. (11). We note that conversions between Galerkin and Chebyshev representations are straightforward, and can be accomplished in $O(M)$ operations. Further-

more, a transformation from between Galerkin and Chebyshev space can be separated using the formalism of Kronecker products, and therefore can be done optimally fast in higher dimensions.

5.2. 1D Poisson

To contrast the Galerkin and Tau approaches, we return to the 1D Poisson problem, $\Delta_{1D}u(x) = f(x)$, with Dirichlet boundary conditions. We first select an appropriate set of basis functions which satisfy the Dirichlet boundary conditions, then approximate $u(x)$ as a discrete sum, namely

$$\phi_m(x) = T_{m+2}(x) - T_m(x)$$
$$u(x) \approx \sum_{m=0}^{M-2} v_m \phi_m(x)$$

We express $u(x)$ in terms of spectral weights of Galerkin basis functions $\underline{v}$ and $f(x)$ in terms of spectral weights of Chebyshev functions $\underline{f}$. As described above, there is a corresponding stencil matrix $S_x^{\{v\}}$ for this Galerkin basis set, which expresses the relationship between the Galerkin basis set and the Chebyshev polynomials. The stencil matrix provides a convenient means for discretizing the differential equation with the boundary conditions embedded within the differentiation matrix. With this notation, the matrix form of the discretized equation becomes

$$D_x^2 * \underline{u} \equiv D_x^2 * (S_x * \underline{v}) = I_x * \underline{f} \tag{28}$$

where D_x^2 is the second-order Chebyshev differentiation matrix. By making use of the stencil matrix, we do not need to derive a new differentiation matrix for each Galerkin basis set, but instead re-use the well known spectral Chebyshev differentiation matrix. Mason and Handscomb [2] derive this same system for the 1D Poisson problem with Dirichlet boundary conditions, but do not utilize the structure of the Chebyshev polynomials to reduce the cost of the solve. We again exploit the idea of our quasi-inverse, and multiply both sides by D_x^{-2}.

$$I_x^{(2)} * S_x * \underline{v} = D_x^{-2} * I_x * \underline{f} \tag{29}$$

Notice that the quasi-inverse for the Galerkin methodology and Tau line methodology are the same because our differential operator is the Chebyshev spectral differentiation matrix in both cases. The non-zero elements of this system take the form Fig. 8.

Notice that since v_{M-1} and v_M are identically zero, we can solve the $(M-2)\times(M-2)$ sub-system where we ignore the top two rows and the last two columns of the operators A and B. We refer to this as the "restricted system" for the Galerkin basis set. As a general rule, the restricted system has size (M-order operator) $\times$ (M-order of Galerkin basis). For problems with only one unknown, these dimensions will be the same, yielding a square system. Examining Fig. 8, this system is nearly identical to pre-multiplied Tau system originally considered (c.f. Fig. 2), but there is an additional 2nd sub-diagonal, with unitary entries. This exactly corresponds to the additional shifted term in the Galerkin basis function $\phi_m(x) = T_{m+2}(x) - T_m(x)$. After we solve this system of equations, $\underline{v}$ represents the Galerkin spectral coefficients, and must be converted back to Chebyshev spectral coefficients $\underline{u}$, which requires $O(M)$ operations. The total cost for the solve is then $O(M)$, and is therefore not very different from the 1D Tau problem. The main savings will be realized in higher dimensions.

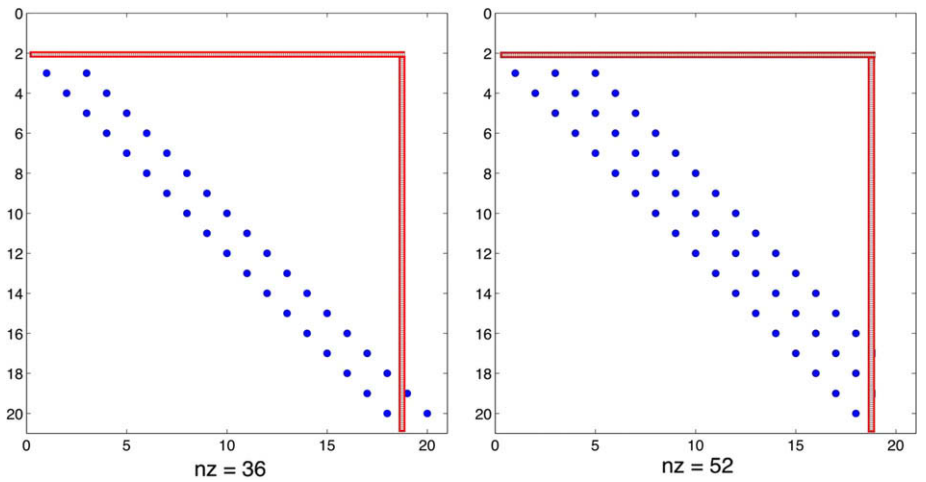


Fig. 8. 1D Pre-multiplied Galerkin Poisson operator $A \equiv I_x^{(2)} * S_x$ (left) and quasi-inverse $B \equiv D_x^{-2} * I_x$ (right) occurring in (9). The dashed lines indicate the restricted system used to solve for the Galerkin spectral coefficients, which already include boundary conditions.

The methodology for the pre-multiplied Galerkin methods is as follows:

(1) For each Galerkin basis in each coordinate, identify the appropriate stencil matrix, S_{x_i}.
(2) Discretize the differential equation using the standard Kronecker formalism to obtain $\mathcal{L}(D) * S_{x_i} * \underline{v} = \underline{f}$.
(3) Identify the correct quasi-inverse B for the highest order operator in each spatial direction.
(4) Multiply the system on both sides by B to obtain a pre-multiplied system $A\underline{v} = B\underline{f}$.
(5) Solve the equation set for $\underline{v}$.
(6) Convert from Galerkin Basis $\underline{v}$ to Chebyshev Basis $\underline{u}$.

5.3. 2D Poisson

For the 2D Poisson problem with Dirichlet boundary conditions, we discretize $u(x,y)$ with the product of two Galerkin basis functions

$$u(x,y) \approx \sum_{m,n=0}^{M-2,N-2} v_{mn}\phi_m(x)\phi_n(y) \tag{30}$$

$$\begin{aligned} \phi_m(x) &= T_{m+2}(x) - T_m(x) \\ \phi_n(y) &= T_{n+2}(y) - T_n(y) \end{aligned} \tag{31}$$

The stencil matrices for each Galerkin basis are the same as they are in 1 dimension $S_x = E_x^{(-2)} - E_x^{(0)}$ and $S_y = E_y^{(-2)} - E_y^{(0)}$, where again $E^{(-2)}$ corresponds shifted identity matrices in each discretization. The discretized equations take the now familiar form.

$$(D_x^2 * S_x \otimes S_y + S_x \otimes D_y^2 * S_y) * \underline{v} = (I_x \otimes I_y) * \underline{f}. \tag{32}$$

Upon multiplication with the quasi-inverse $B = D_x^{-2} \otimes D_y^{-2}$ and restricting each operator we get

$$\left[(I_x^{(2)} * S_x) \otimes (D_y^{-2} * S_y) + (D_x^{-2} * S_x) \otimes (I_y^{(2)} * S_y)\right] * \underline{v} = (D_x^{-2} \otimes D_y^{-2}) * \underline{f} \tag{33}$$

In two dimensions we finally see the strict banded structure that was lacking in the Tau formulation in Fig. 9.

There are total of NM unknowns and the bandwidth grows as $O(M+N)$, so the total computational cost of the solve is $O(MN^2 + M^2N)$. We compare this to the methodology of Doha and Bhrawy [8], Shen [7] and Haidvogel and Zang [6] with the same cost. These previous methods rely upon a matrix diagonalization technique, where in the eigenpairs of the discrete operator are first calculated, then the problem is recast into a lower dimensional problem so the fast 1D solve can be employed multiple times. The methodology we present here does not rely on any eigenpair calculation, and instead exploits the natural structure of the Chebyshev polynomials. While the two-dimensional Chebyshev Laplacian cannot be truly diagonalized, which would lead to an optimally fast solve $O(MN)$ similar to Fourier spectral methods, it can be made bi-diagonal in each dimension leading to banded matrix solve shown in Fig. 9.

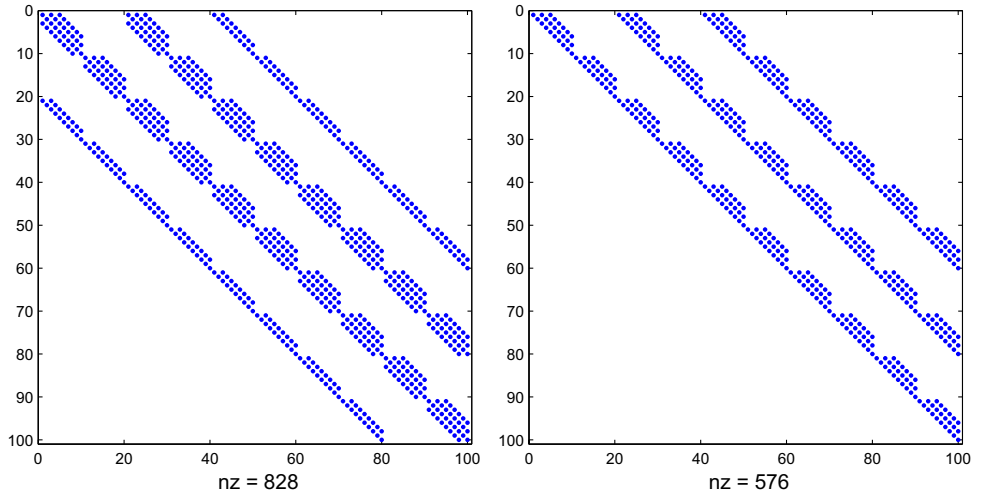


Fig. 9. Non-zero (nz) elements of 2D pre-multiplied restricted Galerkin Poisson operator A (left) and quasi-inverse B (right) from Eq. (33) for $M = 12$ and $N = 12$.

We note that this system was derived by Heinrichs [11] using Galerkin functions $\phi_m(x) = \frac{1}{4}T_{m-2}(x) - \frac{1}{2}T_m(x) + \frac{1}{4}T_{m+2}(x)$. Heinrichs goes on to show that the condition number for this system is $O(N^2)$, a considerable improvement over the full differentiation matrix which has condition number $O(N^4)$.

5.4. 3D Poisson

The formulation of the 3D Poisson problem with Dirichlet is so similar to 2D problem we simply write down the solution, as the methodology for Laplacian type problems should be clear. We discretize with $(M+1)$, $(N+1)$ and $(P+1)$ points in x, y and z, respectively, utilizing the same Galerkin basis functions as in the 1D and 2D cases. After pre-multiplication by the 3D quasi-inverse and restricting in each spatial direction, we get the system

$$\begin{aligned}&[(I_x^{(2)} * S_x) \otimes (D_y^{-2} * S_y) \otimes (D_z^{-2} * S_z) + (D_x^{-2} * S_x) \otimes (I_y^{(2)} * S_y) \otimes (D_z^{-2} * S_z) + (D_x^{-2} * S_x) \otimes (D_y^{-2} * S_y) \otimes (I_z^{(2)} * S_z)] * \underline{v} \\ &\quad = (D_x^{-2} \otimes D_y^{-2} \otimes D_z^{-2}) * \underline{f}\end{aligned}$$

which again has a well defined banded structure. Because of the multiple interaction within the Kronecker product with the tri-diagonal bands, the cost of the solve for (MNP) unknowns is $O(MN^2P^2 + M^2NP^2 + M^2N^2P)$. This suggests that for d dimensions with N^d unknowns the computational cost of the solve scales as $O(N^{2d-1})$.

5.5. 1D general biharmonic

We can easily extend our methodology of quasi-inverses to more complicated problems. Consider the 1D general biharmonic problem with Dirichlet and Neumann boundary conditions:

$$\begin{aligned}&\Delta_{1D}^2 u(x) - \alpha\Delta_{1D} u(x) + \beta u(x) = f(x) \\ &u(\pm 1) = u'(\pm 1) = 0 \\ &x \in [-1, 1]\end{aligned} \tag{34}$$

The first step in the process is to identify the appropriate Galerkin basis function. Shen [7] suggests the basis function

$$\psi_m(x) = T_m(x) - \frac{2(m+2)}{(m+3)}T_{m+2}(x) + \frac{(m+1)}{(m+3)}T_{m+4}(x), \quad m = 0, \ldots, M-4. \tag{35}$$

We must next identify the stencil matrix for this Galerkin basis. We make the following definitions:

- W_2 is the weight matrix with diagonal entries $-\frac{2(m+2)}{(m+3)}$.
- W_4 is the weight matrix with diagonal entries $\frac{(m+1)}{(m+3)}$.

All of these definitions arise naturally from the inner product $\langle \psi_k(x), T_m(x) \rangle$. The stencil matrix for this Galerkin basis set is $S_x = E_x^{(0)} + E_x^{(-2)}W_2 + E_x^{(-4)}W_4$. We can now write down the matrix form of the discretized equations.

$$(D_x^4 - \alpha D_x^2 + \beta I_x) * S_x * \underline{v} = I_x * \underline{f} \tag{36}$$

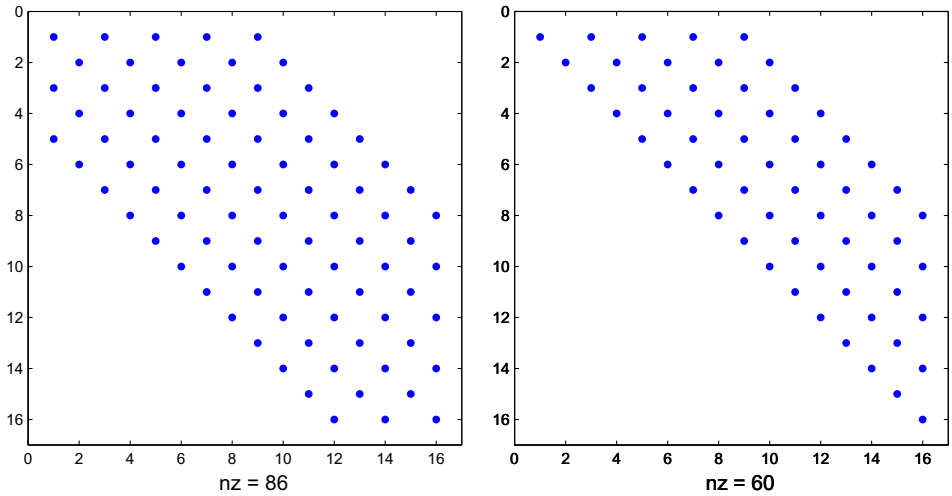


Fig. 10. 1D pre-multiplied Galerkin biharmonic operator A (left) and quasi-inverse B (right) for the 1D biharmonic problem.

where D_x^4 is the fourth-order differen
D_x^4 is the highest order operator, we
where $I_x^{(4)}$ is the quasi-identity matri
four rows of D_x^4 are identically zero, w
note is the $D_x^{-4} * D_x^2 = I_x^{(4)} * D_x^{-2}$,
$D_x^{-4} = I_x^{(4)} * (D^{-1})^4 * I_x^{(-4)}$, or may foun
appropriate quasi-inverse, we multip

$$I_x^{(4)} * (I_x - \alpha D_x^{-2} + \beta D_x^{-4} * I_x) * S_x$$

Note the use of property 6(c) from Se
trivial to identify appropriate sub-sy

Notice that the bands are wider be
do not grow in 1D, the system can st
structed directly since the formulatio
formed explicitly upon computation
remains quasi-optimal. Shen [7] clain
ted with caution. While we agree wi
using the described method that con

5.6. 2D general biharmonic

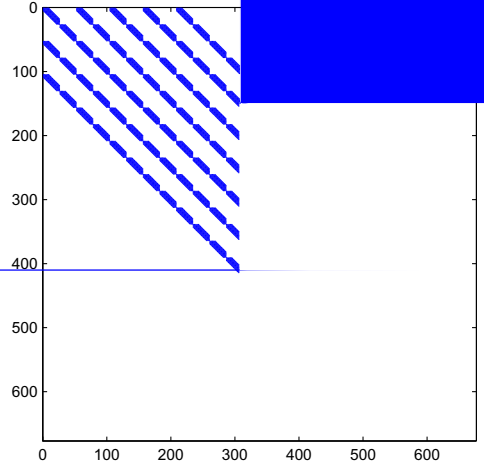
The true beauty and power of this
of differential equations. Consider th
tion formalism of Doha and Bhrawy

$$\begin{aligned}
&\Delta_{2D}^2 u(x,y) - \alpha \Delta_{2D} u(x,y) + \beta u(x \\
&u(\pm 1, y) = u'(\pm 1, y) = 0 \\
&u(x, \pm 1) = u'(x, \pm 1) = 0 \\
&x, y \in [-1, 1]
\end{aligned}$$

Note that Trefethen [3] solves the si
methods. However, the solve is quite
The quasi-inverse methodology allov
tified the correct basis functions an
functions

$$u(x,y) \approx \sum_{m,n=0}^{M-4,N-4} v_{mn} \psi_m(x) \psi_n(y)$$

where $\psi_m(x)$ is defined as in the 1D ca
the differential operator for the simp



angular form. Returning to the generalized biharmonic problem, upon discretization and multiplication by the quasi-inverse $B = D_x^{-4} \otimes D_y^{-4}$, we find the system

$$\begin{aligned}&\left[\left(I_x^{(4)} * \left(I_x - \alpha D_x^{-2} + \frac{\beta}{2} D_x^{-4}\right) * S_x\right) \otimes (D_y^{-4} * S_y) + (D_x^{-4} * S_x) \otimes \left(I_y^{(4)} * \left(I_y - \alpha D_y^{-2} + \frac{\beta}{2} D_y^{-4}\right) * S_y\right)\right] * \underline{v} \\ &\quad = (D_x^{-4} \otimes D_y^{-4}) * \underline{f}\end{aligned} \tag{40}$$

Because of the structure, it is again easy identify the trivial equations and extract the restricted system, Fig. 11.

There are three things to note about this system. First, the complexity of the solve is roughly the same as that of the 2D Poisson/Helmholtz problem. This is because the number of unknowns is the same, and although the bandwidth is slightly wider, it still only grows like $M + N$. Note that the growth of the bandwidth is a consequence of the spatial coupling, causing the bandwidth to grow proportionally with the matrix. Thus computationally the cost of the biharmonic problem and the Poisson problem are of the same order. Second, the banded structure and the Kronecker representation of the operators means that very little information about the operators need to be stored in memory. These storage considerations also extend to the banded solves, which can be made very efficient with standard techniques. Finally, the code modifications are minor in extending from 1D to 2D to 3D. Once the operators are well characterized in 1D, the extensions to higher dimensions is made through simple additions of Kronecker products to the existing code.

5.7. 2D coupled operators

We present one final example which arises from the study of rotationally constrained fluid flow in a differentially heated cavity [16]. The problem presents itself as a coupled partial differential equation in two variables and two dimensions. This is an interesting case because it is not a standard problem, such as the Helmholtz or biharmonic, and we present it here because many different differential equations may arise in practical applications. The equations that we wish to solve are

$$\begin{aligned}&u_{xx}(x,y) + g_y(x,y) = f_1(x,y) \\ &-u_y(x,y) + g_{xxxx}(x,y) = f_2(x,y) \\ &u(\pm 1, y) = 0 \\ &g(\pm 1, y) = g'(\pm 1, y) = g(x, \pm 1) = 0 \\ &x, y \in [-1, 1]\end{aligned} \tag{41}$$

In the x variable, there is a 1D Laplacian operator on $u(x,y)$ with Dirichlet boundary conditions and a 1D biharmonic operator on $g(x,y)$ with Dirichlet/Neumann boundary conditions. In the y variable, we wish to enforce two Dirichlet boundary conditions on $g(x,y)$. We proceed as before by first selecting the appropriate Galerkin basis sets.

$$u(x,y) \approx \sum_{m=0}^{M-2} \sum_{n=0}^{N} v_{mn} \phi_m(x) T_n(y) \tag{42}$$

$$g(x,y) \approx \sum_{m=0}^{M-4} \sum_{n=0}^{N-2} h_{mn} \psi_m(x) \phi_n(y) \tag{43}$$

where

$$\begin{aligned}\psi_m(x) &= T_m(x) - \frac{2(m+2)}{(m+3)} T_{m+2}(x) + \frac{m+1}{m+3} T_{m+4}(x) \\ \phi_m(x) &= T_{m+2}(x) - T_m(x)\end{aligned} \tag{44}$$

Note that because there are no boundary conditions to enforce in the y direction for $u(x,y)$, we simply allow this direction to be expanded in Chebyshev polynomials. The Galerkin spectral coefficients for the unknown function $g(x,y)$ are $\underline{h}$ and the Galerkin spectral coefficients for $u(x,y)$ are $\underline{v}$. Again, we define the appropriate stencil matrix for each variable based upon the defined Galerkin set

$$\begin{aligned}S_x^{\{v\}} &= E_x^{(-2)} - E_x^{(0)} \\ S_y^{\{v\}} &= E_y^{(0)} \\ S_x^{\{h\}} &= E_x^{(0)} + E_x^{(-2)} W_2 + E_x^{(-4)} W_4 \\ S_y^{\{h\}} &= E_y^{(-2)} - E_y^{(0)}\end{aligned} \tag{45}$$

With these definitions the discretized equations take the form

$$\begin{aligned}&(D_x^2 * S_x^{\{v\}} \otimes S_y^{\{v\}}) \underline{v} + (S_x^{\{h\}} \otimes D_y^1 * S_y^{\{h\}}) * \underline{h} = (I_x \otimes I_y) * \underline{f_1} \\ &(-S_x^{\{v\}} \otimes D_y^1 * S_y^{\{v\}}) \underline{v} + (D_x^4 * S_x^{\{h\}} \otimes S_y^{\{h\}}) * \underline{h} = (I_x \otimes I_y) * \underline{f_2}\end{aligned} \tag{46}$$

Next, we must identify the quasi-inverse for each variable in each of the two equations. For Eq. (46a), the quasi-inverse is $D_x^{-2} \otimes D_y^{-1}$ and for Eq. (46b) $D_x^{-4} \otimes D_y^{-1}$, where D_x^{-2} and D_x^{-4} are defined as before and D_y^{-1} can be derived from the recursion relation in Eq. (7). Upon pre-multiplication and restriction we get the coupled system

$$\begin{aligned} &(i)\ (I_x^{(2)} * S_x^{\{v\}} \otimes D_y^{-1} * S_y^{\{v\}}) * \underline{v} + (D_x^{-2} * S_x^{\{h\}} \otimes I_y^{(1)} * S_y^{\{h\}}) * \underline{h} = (D_x^{-2} \otimes D_y^{-1}) * \underline{f_1} \\ &(ii)\ (-D_x^{-4} * S_x^{\{v\}} \otimes I_y^{(1)} * S_y^{\{v\}}) * \underline{v} + (I_x^{(4)} * S_x^{\{h\}} \otimes D_y^{-1} * S_y^{\{h\}}) * \underline{h} = (D_x^{-4} \otimes D_y^{-1}) * \underline{f_2} \end{aligned} \tag{47}$$

Note that for coupled systems care must be taken in extracting the restricted matrix system, since we are multiplying $\underline{v}$ by the quasi-inverse for $\underline{h}$ and vice-versa. This banded system has several properties that make it attractive. First, the time to solve system is greatly reduced from a comparable Tau line implementation because we have eliminated the communication from the Tau lines across all of the nodes. Second, the sparse structure of the matrix reduces storage costs, which become prohibitively expensive even in two dimensions. Finally, we point out that the matrix diagonalization technique of Haidvogel et al., Shen and Doha et al. cannot be applied to this complex system of equations, so previous solution strategies have relied on expensive Tau method implementations with full differentiation matrices. The adaptability of our quasi-inverse technique in each variable allows for a much broader class of problems to be readily discretized and efficiently solved.

6. Numerical results

For our numerical tests, we study the same test problems as Haidvogel and Zang [6], Dang-Vu and Delcarte [9], Shen [7] and Doha and Bhrawy [8]. For comparison between multiple dimensions, all spatial discretizations will be of size N. Thus, 1D systems will have N unknowns, 2D systems N^2 unknowns, and 3D systems N^3 unknowns. All test codes are implemented in MATLAB and are performed on desktop Dell workstations with Intel Pentium 3.2 GHz dual core processors, although only one core is used during test runs. We note that we utilize MATLABs' built-in "sparse" representation for matrices and solves are performed via the "backslash" operator, which manipulates the sparse structure of the matrix to optimize the computation time. This is done extremely efficiently for banded matrices, often utilizing non-intuitive data organizations for peak

Table 1
Timing results for Tau and Galerkin solutions of the Poisson problem in 1D, 2D and 3D.

N – Dimension	Method	Unknowns	$Error_\infty$	CPU (s)	Condition
16 – 1D	Tau	16	1.6E−6	1.55E−4	6.39E+3
32 – 1D	Tau	32	2.72E−15	1.77E−4	1.04E+5
64 – 1D	Tau	64	1.88E−15	2.18E−4	1.66E+6
128 – 1D	Tau	128	2.22E−15	2.98E−4	2.66E+7
256 – 1D	Tau	256	2.13E−15	4.53E−4	4.26E+8
16 – 1D	Galerkin	16	4.03E−6	7.54E−5	9.35E+0
32 – 1D	Galerkin	32	2.77E−15	8.44E−5	1.96E+1
64 – 1D	Galerkin	64	1.77E−15	9.4E−5	4.01E+1
128 – 1D	Galerkin	128	2.22E−15	1.22E−4	8.08E+1
256 – 1D	Galerkin	256	2.05E−15	1.71E−4	1.62E+2
16 – 2D	sTau	16^2	1.45E−6	3.65E−3	1.48E+4
32 – 2D	Tau	32^2	1.55E−15	2.11E−2	1.56E+5
64 – 2D	Tau	64^2	1.66E−15	1.05E−1	3.14E+6
128 – 2D	Tau	128^2	1.86E−15	0.69	5.04E+7
256 – 2D	Tau	256^2	2.27E−15	3.07E+2	[a]
16 – 2D	Galerkin	16^2	6.37E−6	3.46E−3	2.37E+5
32 – 2D	Galerkin	32^2	1.41E−15	1.67E−2	1.87E+7
64 – 2D	Galerkin	64^2	2.00E−15	7.84E−2	1.31E+9
128 – 2D	Galerkin	128^2	2.22E−15	0.39	8.77E+10
256 – 2D	Galerkin	256^2	2.05E−15	2.25	5.75E+12
16 – 3D	Tau	16^3	1.34E−6	1.26	4.12E+7
20 – 3D	Tau	20^3	1.32E−9	3.89	2.14E+8
24 – 3D	Tau	24^3	4.80E−13	9.31	[a]
28 – 3D	Tau	28^3	4.85E−15	26.52	[a]
32 – 3D	Tau	32^3	4.62E−15	58.16	[a]
40 – 3D	Tau	40^3	[a]	[a]	[a]
16 – 3D	Galerkin	16^3	8.32E−6	0.10	6.72E+9
20 – 3D	Galerkin	20^3	7.74E−9	0.35	9.07E+10
24 – 3D	Galerkin	24^3	2.54E−12	0.91	7.43E+11
28 – 3D	Galerkin	28^3	6.67E−15	2.00	4.34E+12
32 – 3D	Galerkin	32^3	9.2E−15	5.16	1.98E+12
40 – 3D	Galerkin	40^3	2.83E−15	20.41	[a]

[a] Matrix too large for memory, no estimate of condition number.

performance. However, we establish that this is consistent with estimates of the computational cost associated with the complexity of the matrix. We present tables to show spectral accuracy for each test case and "time to solve vs. number unknowns" plots to demonstrate the complexity of the algorithm in each dimension. The timing is performed by averaging the time to solve each test problem over 20 separate runs for each number of unknowns.

6.1. Poisson solvers

The standard test problem for the Poisson problem in D dimensions with Dirichlet boundary conditions is

$$\begin{aligned} &\Delta u(\underline{x}) = f(\underline{x}) \in \Omega \\ &u = 0 \text{on} \Gamma x \in [-1,1]^D \end{aligned} \tag{48}$$

where the forcing term is

$$f(\underline{x}) = -D * (4 * \pi^2) \prod_{i=1}^{D} \sin(2 * \pi * x_i) \tag{49}$$

and the analytic solution is

$$u(\underline{x}) = \prod_{i=1}^{D} \sin(2 * \pi * x_i) \tag{50}$$

We solve this problem with both a Tau line implementation and a Galerkin basis, $\phi_m = T_{m+2} - T_m$, to compare the complexity of the methods. Once the number of points in the discretization exceeds 26 in each spatial dimension, the forcing function is resolved to machine precision in a 32-bit architecture and the solution is accurate to machine precision. Table 1 below shows this behavior for 1, 2 and 3 dimensions. Also take note that for very large discretizations, the solutions remain accurate to machine precision, indicating that the direct solve is unaffected by the conditioning of the discretized matrix and is therefore very stable. The condition numbers shown in this table were estimated numerically using built-in Matlab functions. Had we utilized an iterative method which relied on forward applications of the operators, the conditioning of the operator would immediately arise in the solution.

The 3D cases are getting close to the limits of computation that is possible on a desktop computer. It is still interesting to note that the 3D Galerkin solve for 64,000 unknowns can be performed with spectral precision in about 20 s. The efficiency gain of the Galerkin method over the Tau method is most pronounced in 3D, where the time to solve is improved by an order of magnitude.

Next we examine Fig. 12, which shows the time to solve vs. number discretization points in a single spatial direction N on a log–log scale.

In this graph, the slope of the line is indicative of the order of the method and the gap between Tau and Galerkin lines is the overhead associated the Tau lines. We see that 1D problems scale like $O(N)$, the 2D solves scale like $\sim O(N^3)$, and the 3D system is more expensive still, $O(N^5)$. Thus we establish for N^D the scaling law $O(N^{2D-1})$. What is important computationally is the order of solve compared to the total number of unknowns, which is N, N^2, and N^3 in 1, 2 and 3 dimensions respectively for a scaling of $O(N^{\frac{2D-1}{D}})$, Fig. 13.

Let us compare the Tau and Galerkin methods in each dimension. In the 1D case, both the Tau line method and the Galerkin basis scale linearly with number of unknowns, as would be expected. In 2D, the Tau solution and the Galerkin

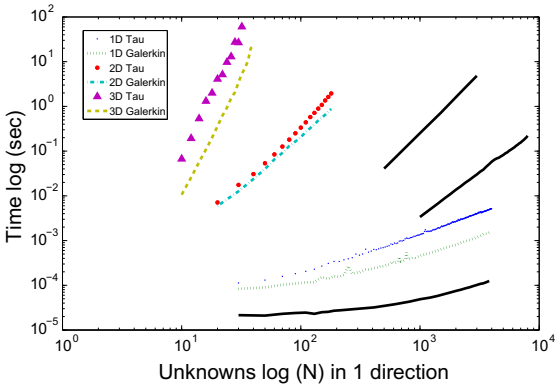


Fig. 12. Timing of Tau and Galerkin methods in 1, 2 and 3 dimensions verses the number of discretization points N in a single dimension. The solid black lines are shown for comparison, the lowest being a purely diagonal system $O(N)$ to be compared with the 1D solutions, the middle an upper triangular $O(N^2)$ and the highest a dense random matrix $O(N^3)$ to be compared with the 2D solutions. This figures illustrates the dimensional scaling law $O \sim N^{2D-1}$.

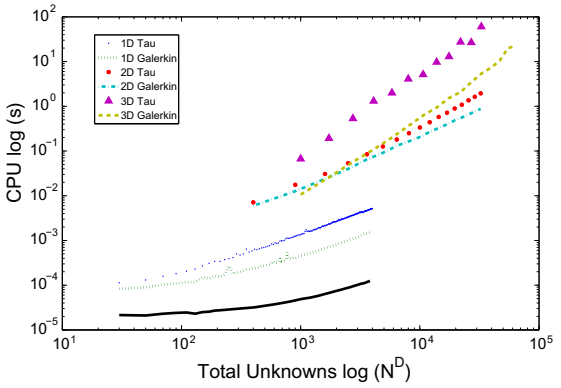


Fig. 13. Timing of Tau and Galerkin methods vs. the total number of unknowns. The bold black line at the bottom of the plot is a purely diagonal solve matrix solve, which scale as $O(N)$. The slopes of the lines show the scaling law complexity $\sim O(N^{\frac{2D-1}{D}})$.

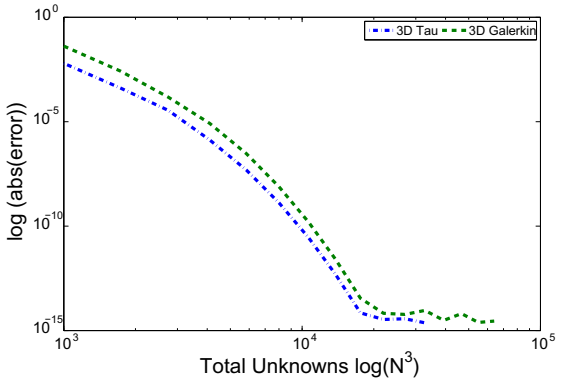


Fig. 14. Spectral convergence of the 3D Poisson operator for Tau and Galerkin methodologies. Log–log scale: time to solve (seconds) vs. number of unknowns.

solution have similar scalings, but the slope of Galerkin method is less for total higher unknowns. At this point the communication of the Tau lines becomes dominant, and the Galerkin method is clearly superior for large discretizations. In 3D dimensions cost associated with Tau lines has become very significant, making the Galerkin method an order of magnitude faster than the Tau method. Finally, we look at the convergence rate of the $\|\|_\infty$ error. As expected, we see spectral convergence which is shown for the 3D Poisson problem in Fig. 14.

Table 2
Timing Results for the 1D, 2D and 3D generalized biharmonic problem with Galerkin basis sets.

N	Dimension	Unknowns	$Error_\infty$	CPU (s)	Condition
16	1D	16	5.54E−2	2.99E−5	2.81E+2
32	1D	32	1.00E−15	3.83E−5	1.72E+3
64	1D	64	1.18E−13	5.53E−5	8.88E+3
128	1D	128	1.17E−14	9.11E−5	4.27E+4
256	1D	256	7.31E−14	1.59E−4	2.36E+5
4096	1D	4096	1.51E−14	2.74E−3	7.47E+7
16	2D	16^2	1.67E−1	2.28E−3	3.52E+7
24	2D	24^2	7.24E−7	1.22E−2	5.51E+9
32	2D	32^2	9.68E−14	2.53E−2	1.90E+11
64	2D	64^2	4.39E−14	1.52E−1	8.85E+14
128	2D	128^2	3.68E−14	1.04	3.87E+18
16	3D	16^3	1.77E−2	1.23E−1	5.91E+13
24	3D	24^3	3.93E−8	1.48	6.76E+17
32	3D	32^3	3.74E−13	12.96	4.70E+20

The quasi-inverse methodology with Galerkin basis functions has a complexity of $O(N^5)$ for N^3 unknowns in the 3D Poisson problem. We will see, below, that the 3D biharmonic problem is solved in roughly the same number of operations. We compare this to the complexity of a full collocation scheme, which scales like $O(N^9)$ in three dimensions.

6.2. Biharmonic solvers

For the biharmonic problem,

$$\Delta_2^2 u(\mathbf{x}) - \alpha \Delta_2 u(x) + \beta u(x) = f(x) \tag{51}$$

with Dirichlet and Neumann boundary conditions in each direction we choose a test function

$$u(\mathbf{x}) = \prod_{i=1}^{D} sin(2\pi x_i)(T_2(x_i) - T_0(x_i)) \tag{52}$$

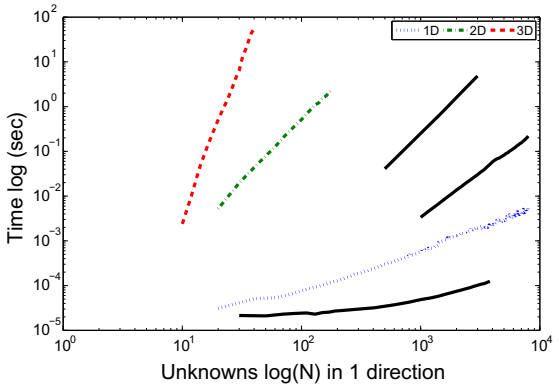


Fig. 15. Galerkin biharmonic timing log–log plot: number of unknowns in 1 spatial direction N vs. time to solve. The slope for the 1D curve (dotted) is nearly 1, indicating an $O(N)$ solve. The dashed-dot (2D) and dashed (3D) lines are steeper indicating more computational cost. The solid lines are timing line for: lowest = purely diagonal system $O(N)$, middle = upper triangular system $O(N^2)$; upper = dense matrix $O(N^3)$.

Table 3
Timing and error results for the coupled system of equations Eq. (53).

N	V_Error_∞	G_Error_∞	CPU (s)
16	1.02E−3	5.61E−2	1.70E−2
20	1.84E−6	1.86E−4	4.48E−2
24	1.26E−9	2.05E−7	9.43E−2
28	4.65E−13	9.70E−11	1.70E−1
32	2.15E−13	1.02E−13	2.98E−1
64	2.89E−13	9.61E−14	3.2

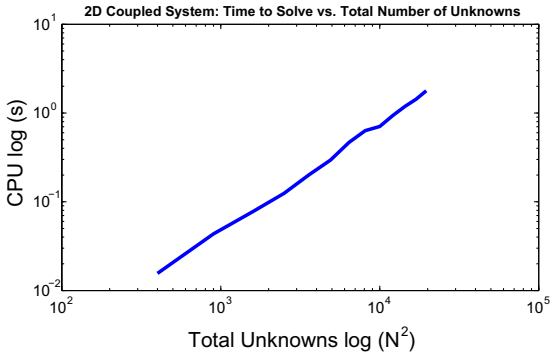


Fig. 16. Log–log 2D coupled system timing, total number of unknowns N^2 vs. CPU time. In the log–log plot, the slope of this line is 1.5, indicating a complexity of $O(N^{3/2})$ for N^2 unknowns. This is exactly the same cost associated with the 2D Helmholtz, Poisson and biharmonic problems, suggesting that complexity scales with spatial dimension independent of the form the differential operator.

with D ranging from 1 to 3. We study this problem with $\alpha = 1.1$ and $\beta = 1$. The corresponding forcing function $f(\mathbf{x})$ involves to many unknowns to warrant explicitly writing it here, and is left out for compactness. Below, Table 2, we see the both timing and infinity norm errors for 1, 2 and 3 dimensions with a range of discretization points.

Table 4
Tau lines for enforcement of boundary conditions for Chebyshev discretizations on the interval $x \in [-1, 1]$.

Boundary conditions	Linear combination of spectral coefficients
$u(1) = 0$	$\sum_{m=0}^{M} u_m = 0$
$u(-1) = 0$	$\sum_{m=0}^{M} (-1)^m u_m = 0$
$u'(1) = 0$	$\sum_{m=0}^{M} m^2 u_m = 0$
$u'(-1) = 0$	$\sum_{m=0}^{M} m^2 (-1)^{(m+1)} u_m = 0$
$u''(1) = 0$	$\frac{1}{3}\sum_{m=0}^{M} m^2 (m^2 - 1) u_m = 0$
$u''(-1) = 0$	$\frac{1}{3}\sum_{m=0}^{M} m^2 (m^2 - 1)(-1)^m u_m = 0$
$\vdots$	$\vdots$
$u^{(q+1)}(1) = 0$	$\frac{2^q (q!)}{(2q+1)!} \sum_{m=0}^{M} \prod_{p=0}^{q} (m^2 - p^2) u_m$
$u^{(q+1)}(-1) = 0$	$\frac{2^q (q!)}{(2q+1)!} \sum_{m=0}^{M} \prod_{p=0}^{q} (m^2 - p^2)(-1)^{(m+q-1)} u_m$

Table 5
Some Chebyshev Galerkin basis functions and there associated boundary conditions.

Boundary conditions	Discretization	Galerkin basis functions
$u(\pm 1) = 0$	$u(x) \approx \sum_{m=0}^{M-2} v_m \phi_m(x)$	$\phi_m(x) = \begin{cases} T_{m+2}(x) - T_0(x) & m \text{ even} \\ T_{m+2}(x) - T_1(x) & m \text{ odd} \end{cases}$ [a]
$u(\pm 1) = 0$	$u(x) \approx \sum_{m=0}^{M-2} v_m \phi_m(x)$	$\phi_m(x) = T_{m+2}(x) - T_m(x)$
$u(\pm 1) = 0$	$u(x) \approx \sum_{m=0}^{M-2} v_m \phi_m(x)$	$\phi_m(x) = T_{m+2}(x) - \frac{2}{c_m} T_m(x) + e_{m-2} T_{m-2}(x)$ [b]
$u'(\pm 1) = 0$	$u(x) \approx \sum_{m=0}^{M-2} v_m \phi_m(x)$	$\phi_m(x) = \frac{(m+2)6(x)372e5.548366.3360}{}$

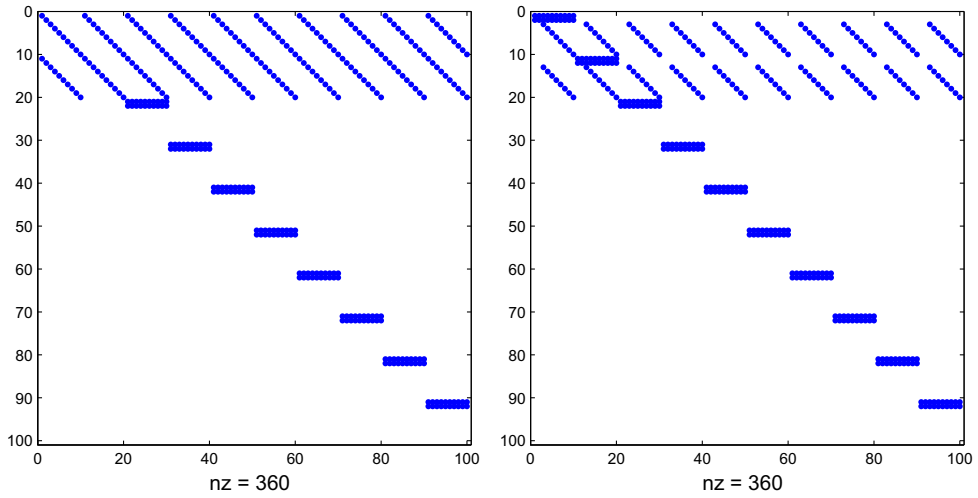


Fig. 18. Two correct implementations of two-dimensional Dirichlet Tau lines. (Left) All of the x Tau lines have been retained, and four y Tau lines have been thrown out. (Right) All of the y Tau lines have been retained and four x Tau lines have been thrown out.

In the 1D data, we include a discretization of 4096 points. This is done only to illustrate the well conditioned nature of the biharmonic operator when solved with this methodology. We again show timing comparisons between the various dimensions, Fig. 15.

If we compare the slope of the 3D biharmonic solve to the slope of the 3D Laplacian Galerkin solve, we see that the scaling exponents are the same, again indicating that the dimension of the problem is the critical component of determining complexity.

6.3. 2D coupled system

For the 2D coupled system

$$\begin{aligned} &u_{xx}(x,y) + g_y(x,y) = f_1(x,y) \\ &-u_y(x,y) + g_{xxxx}(x,y) = f_2(x,y) \\ &u(\pm 1, y) = 0 \\ &g(\pm 1, y) = g'(\pm 1, y) = g(x, \pm 1) = 0 \\ &x, y \in [-1, 1] \end{aligned} \tag{53}$$

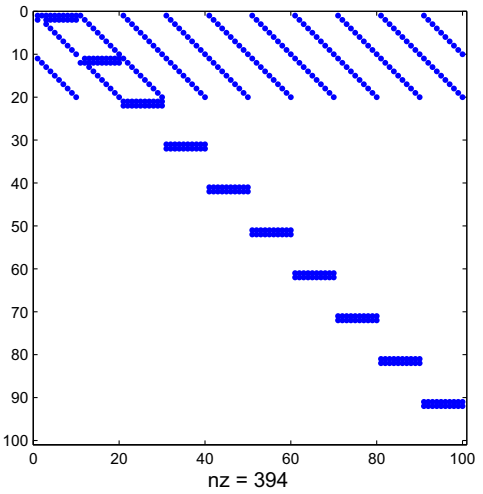


Fig. 19. Incorrect implementation of Tau lines. The Tau lines for the spectral modes $[u_{00}, u_{01}, u_{10}, u_{11}]$ have been added together (top of figure), see lines 1,2 and 11,12 in Fig. 18.

we chose to test the quasi-inverse method with functions

$$u(x,y) = sin(\pi x) * cos(\pi y)$$
$$g(x,y) = sin(2 * \pi * x) * \left(\frac{16}{5}(x^2-1)^2 * (6 * x - 1)\right) * sin(\pi * y) \tag{54}$$

with the corresponding forcing functions, $f_1(x,y)$ and $f_2(x,y)$. Table 3 below summarizes the timing and convergence results.

We estimate the performance of this solve by plot the timing data in log–log coordinates, Fig. 16. Measuring the slope of this line, we find that for N^2 unknowns the complexity is $O(N^4)$. While we do not achieve an optimal $O(N^2)$ complexity, this result of $O(N^4)$ is an improvement when compared to the original coupled system which is nearly dense matrix. We have also realized a substantial improvement through the Galerkin representation both in time to solve and memory allocation.

7. Conclusions

We have presented a technique for solving differential equations discretized with Chebyshev polynomials that is efficient, adaptable to different operators, and easily generalizes to multiple dimensions. For situations where the boundary conditions are not well represented by a Galerkin basis set, the concept of a quasi-inverse may still be applied to the more basic Lanczos Tau method. For non-homogeneous boundary conditions, the Tau methodology is much easier to implement than finding an appropriate Galerkin basis set, which are easily constructed using (11), a general analytic expression for boundary conditions which we believe to not have been previously published. We have shown that this method is as efficient in two dimensions as the standard matrix diagonalization technique of Haidvogel and Shen, and does not exhibit signs of poor conditioning for fourth-order operators even for very large systems of equations, i.e. – 1000's of unknowns in 1D. The quasi-inverse method is based upon direct solves, not iterative methods, and therefore is not effected by ill-conditioning associated with forward matrix operations. Furthermore, the robustness of the method has been demonstrated through the efficient solution of the 2D and 3D fourth-order generalized biharmonic equation. The numerical results in this paper indicate that the complexity of a solve for non-coupled problems is dependent only on the spatial dimension in which the problem is embedded and is independent of the differential form of the operator. The quasi-inverse exploits the one-dimensional three term recursion relation in each variable, making it analogous to diagonal Fourier methods, instead of relying on eigenvector decompositions which may have conditioning problems for large values of N. As with other spectral methods, the quasi-inverse method is well suited for linear PDEs.

It is the hope of the authors that this methodology can be widely adapted for existing codes that require spectral accuracy in several dimensions. The quasi-inverse concept is easily extended to other fields of study. The simple formulation of the quasi-inverse methodology allows for problems to be rapidly coded once a few basic subroutines have been defined. The authors have provided a "Quasi-Inverse" toolkit which is available for download from the Mathworks community file sharing website, under the name "ChebyshevTools". In Appendix B we have provided details about implementing the quasi-inverse method and a comparison to the matrix diagonalization technique. In a future note, we will show how this methodology can be extended to cylindrical coordinates and used for stability analysis in the study of fluid dynamics.

Acknowledgments

We would like to thank Bengt Fornberg, Geoff Vasil, and Joe Werne for their insightful comments in reviewing this paper. We would also like to thank our reviewers for their suggestions on improving the readability of this submission. This work was supported by: NSF award DMS 0602284, NASA award NNG05GD37G, and a University of Colorado SEED Grant.

Appendix A. Deriving stencil matrices from Tau-line boundary conditions

It is well known that there is a close relationship between Tau methods and Galerkin methods, and authors have shown equivalence between the two methods for many cases, McFadden et al. [17]. In this appendix, we provide for reference a straight forward way to derive Galerkin basis sets from known Tau conditions. Consider the following expressions for enforcing different boundary conditions using the Lanczos Tau method. The second column indicates what linear combination of spectral coefficients need to be enforced so that the boundary condition in the first column is satisfied.

Utilizing these expressions we can easily determine a related Galerkin basis set. We begin by defining an ansatz for the form of our Galerkin set

$$\phi_m(x) = \alpha T_m(x) + \beta T_{m+2}(x) + \gamma T_{m+4}(x) + \cdots \tag{55}$$

The number of unknowns $\alpha, \beta, \gamma, \ldots$ is generally, but not necessarily, determined by the number of boundary conditions that need to be enforced. However, the choice of our ansatz is not unique and we could have just as easily of chosen

$$\phi_m(x) = \Gamma T_{m-2}(x) + \alpha T_m(x) + \beta T_{m+2}(x) + \cdots$$

Not all ansatz will necessarily be consistent for all modes or for a given set of boundary conditions. Using Table 4, we can extract a system of equations for the unknown coefficients in our ansatz which will used to satisfy the boundary conditions.

Consider the boundary conditions $u(1) = u(-1) = u''(1) = u''(-1) = 0$. Using the ansatz in Eq. (55) with 3 unknowns, this gives a set of four equations

$$\phi_m(1) = \alpha + \beta + \gamma = 0$$
$$\phi_m(-1) = -\alpha - \beta - \gamma = 0$$
$$\phi_m''(1) = \frac{1}{3}(m^2(m^2-1)\alpha + (m+2)^2((m+2)^2-1)\beta + (m+4)^2((m+4)^2-1)\gamma) = 0$$
$$\phi_m''(-1) = \frac{1}{3}(-m^2(m^2-1)\alpha - (m+2)^2((m+2)^2-1)\beta - (m+4)^2((m+4)^2-1)\gamma) = 0$$

Two of the lines in system are clearly redundant, indicating that we will have a degree of freedom in determining our constants. Solving for the unknowns we find the following constraints for β and γ, where α is left arbitrary.

$$\beta_m = -\frac{2(m+2)(15+2m(4+m))\alpha}{(m+3)(19+2m(6+m))}$$
$$\gamma_m = \frac{(m+1)(3+2m(2+m))\alpha}{(m+3)(19+2m(6+m))}$$

Using this technique and different ansatzs, we compile Table 5. We also note that the weight matrices used in assembling the stencil matrices from Section 5 can be obtained directly from this table with the associated basis polynomial.

Appendix B. Implementation details and the matrix diagonalization technique

In this section for readers interested in implementing the quasi-inverse methodology, we provide specific notes about multi-dimensional Tau lines, a brief discussion about "mixed" operator differential equations, and further illustrate the clarity provided by the Kronecker notation by outlining to the matrix diagonalization technique.

B.1. Tau lines in multiple dimensions

When lines are used in multiple dimensions, it is important that correct number of Tau lines are used so that specific boundary conditions are not enforced twice. Each Tau line represents the enforcement of a specific boundary condition, either a boundary point in physical space or a relationship between spectral mode. Let us consider a 1D problem with Dirichlet boundary conditions and $N = 10$ unknowns $\underline{u} = [u_0, u_1, \ldots, u_9]$. In physical space, we specify the values of the left and right endpoints with individual Tau lines. In spectral space, we specify the constant contribution u_0 and the linear contribution u_1. This leaves eight unknowns u_2 through u_9 to be determined by the differential portion of the operator. In two dimensions, the domain is a square, and the four bounding sides must be specified for Dirichlet boundary conditions. Again consider $N = 10$ points for the discretization in each direction, resulting in 100 unknowns:

$$\underline{u} = \begin{bmatrix} u_{00} & u_{01} & \cdots & u_{09} \\ u_{10} & u_{11} & \cdots & \vdots \\ \vdots & & \ddots & \\ u_{90} & \ldots & & u_{99} \end{bmatrix}$$

In physical space, there are four points (the corners of the domain) which share a boundary condition in both x and y, and these points can be enforced either with x boundary conditions or y conditions, but not both at the same time. In spectral space, the overlapping modes are the constant, linear, and bilinear modes = $[u_{00}, u_{01}, u_{10}, u_{11}]$, respectively. In Fig. 17, we see the form of the x Tau lines on the left and the y Tau lines on the right, where the overlapping conditions are highlighted red.

Hence, there are 36 Tau lines which specify boundary conditions and $8 \times 8 = 64$ unknowns which must be determined by differential operator. In Fig. 18 we show two correct implementations of Tau lines in 2D. Compare this to Fig. 19, where overlapping Tau conditions have been counted twice, so the "corner" boundary conditions will not be correct.

B.2. Additional applications of the kronecker product

In this paper we have made extensive use of the Kronecker product, which provides for clear separation of operators in multiple dimensions. This property can be very useful in analyzing different problems, and we briefly present a derivation of the matrix diagonalization technique using Kronecker notation, which can be compared to tensor product notation employed by Shen [7]. For a test problem, we begin with the 2D Helmholtz problem, $\Delta_{2D}u(x,y) + \alpha u(x,y) = f(x,y)$. Following Shen's approach, we pick the appropriate Galerkin basis set for the given boundary conditions, then discretize the equation using the 1D differentiation matrix $D_x^2 \equiv D_x^2 * S_x$ for the Galerkin basis

$$[(\widetilde{D_x^2} \otimes I_y) + (I_x \otimes \widetilde{D_y^2}) + \alpha(I_x \otimes I_y)] * \underline{u} = \underline{f}$$

The next step is to find the eigenvector/eigenvalue decomposition for the x-operator $\widetilde{D_x^2}$ such that $\widetilde{D_x^2}E = E\Lambda$ where E has columns of the eigenvectors of $\widetilde{D_x^2}$ and Λ is a diagonal matrix with eigenvalues λ_n corresponding to the E_n eigenvector. Posing a change of variables $\underline{u} = (E \otimes I_y)\underline{v}$, the discretized system becomes

$$\begin{aligned}\underline{f} &= [(\widetilde{D_x^2} \otimes I_y) + (I_x \otimes \widetilde{D_y^2}) + \alpha(I_x \otimes I_y)] * (E \otimes I_y) * \underline{v} \\ &= [(\widetilde{D_x^2} * E \otimes I_y) + (E \otimes \widetilde{D_y^2}) + \alpha(E \otimes I_y)] * \underline{v} \\ &= [(E\Lambda \otimes I_y) + (E \otimes \widetilde{D_y^2}) + \alpha(E \otimes I_y)] * \underline{v}\end{aligned}$$

If we pre-compute the inverse of the eigenvalue matrix E^{-1}, we can multiply both sides of the equation by $(E^{-1} \otimes I_y)$ and make the change of variables $\underline{g} = (E^{-1} \otimes I_y)\underline{f}$ to arrive at the system below.

$$[(\Lambda \otimes I_y) + (I_x \otimes \widetilde{D_y^2}) + \alpha(I_x \otimes I_y)] * \underline{v} = \underline{g}$$

$(\Lambda \otimes I_y)$ is a diagonal matrix, so this system of equations is block diagonal and can be decomposed into N – 1D solves of the form $[(\lambda_n + \alpha)I_y + \widetilde{D_y^2}] * v_n = g_n$. Using the Kronecker notation, it is generally clear when system has been reduced to a collection of lower dimensional problems. We note that the primary cost for the 2D Helmholtz problem is in the calculation E and E^{-1}, which both require $O(N^3)$ calculations. We could continue the process and diagonalize the y-differential operator in the same fashion, which would result in the system

$$[(\Lambda_x \otimes I_y) + (I_x \otimes \Lambda_y) + \alpha(I_x \otimes I_y)] * \underline{w} = \underline{h}$$

where $\underline{h} = (E^{-1} \otimes E^{-1})^* \underline{f}$ and $\underline{w} = (E \otimes E)^* \underline{v}$. This extra step results in a purely diagonal operator acting on $\underline{w}$, which could be solved in $O(N^2)$ operations, but the matrix multiplication $(E^{-1} \otimes E^{-1})^* \underline{f}$ is a dense calculation requiring $O(N^4)$ operations. For this problem, it is computationally cheaper to only diagonalize in only one direction.

References

[1] J. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover, 2001.
[2] J.C. Mason, D.C. Handscomb, Chebyshev Polynomials, Prentice Hall, SIAM, 2002.
[3] L. Trefethen, Spectral Methods in MATLAB, SIAM, 2000.
[4] Bengt Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge Monographs, 1995.
[5] Heiner Igel, Wave propagation in three-dimensional spherical sections by the Chebyshev spectral method, Geophysical Journal International 136 (1999) 559–566.
[6] D.B. Haidvogel, T. Zang, The accurate solution of Poisson's equation by expansion in Chebyshev polynomials, Journal of Computational Physics 30 (1979) 167–180.
[7] J. Shen, Efficient spectral-Galerkin method. II. Direct solvers of second- and fourth-order equations using Chebyshev polynomials, SIAM Journal on Scientific Computing 16 (1) (1995) 74–87.
[8] E.H. Doha, A.H. Bhrawy, Efficient spectral-Galerkin algorithms for direct solution for second-order differential equations using Jacobi polynomials, Numerical Algorithms 42 (2006) 137–164.
[9] H. Dang-Vu, C. Delcarte, An accurate solution of the Poisson equation by the Chebyshev collocation method, Journal of Computational Physics 104 (1993) 211–220.
[10] C. Lanczos, Evaluation of noisy data, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis 1 (1) (1964) 76–85. URL: <http://link.aip.org/link/?SNA/1/76/1>.
[11] W. Heinrichs, Improved condition number for spectral methods, Mathematics of Computation 53 (197) (1989) 103–119.
[12] D. Gottlieb, S. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, Cambridge Press, 1977.
[13] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1991.
[14] A. Zebib, A Chebyshev method for the solution of boundary value problems, Journal of Computational Physics 53 (1984) 443–455.
[15] M. Awan, T. Phillips, Well-conditioned spectral discretizations of the biharmonic operator, International Journal of Computer Mathematics 48 (1992) 179–189.
[16] Michael Watson, A study of rotationally constrained convection in a tall aspect ratio annulus, Ph.D. thesis, University of Colorado at Boulder, 2008.
[17] G.B. McFadden, B.T. Murray, R.F. Boisvert, Elimination of spurious eigenvalues in Chebyshev Tau spectral method, Journal of Computational Physics 91 (1990) 228–239.