Journal of Computational Physics 299 (2015) 649–666

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# A minimum Sobolev norm technique for the numerical discretization of PDEs

S. Chandrasekaran [a,*,1], H.N. Mhaskar [b,c,2]

[a] Department of Electrical & Computer Engineering, University of California, Santa Barbara, USA
[b] Department of Mathematics, California Institute of Technology, Pasadena, CA 91125, USA
[c] Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA 91711, USA

ARTICLE INFO

*Article history:*
Received 4 July 2013
Received in revised form 9 March 2015
Accepted 6 July 2015
Available online 20 July 2015

*Keywords:*
Minimum Sobolev norm
Golomb–Weinberger
First-order PDEs

ABSTRACT

Partial differential equations (PDEs) are discretized into an *under*-determined system of equations and a minimum Sobolev norm solution is shown to be efficient to compute and converge under very generic conditions. Numerical results of a single code, that can handle PDEs in first-order form on complicated polygonal geometries, are shown for a variety of PDEs: variable coefficient div–curl, scalar elliptic PDEs, elasticity equation, stationary linearized Navier–Stokes, scalar fourth-order elliptic PDEs, telegrapher's equations, singular PDEs, etc.

## 1. Introduction

In this paper we describe a novel numerical technique for the discretization of partial differential equations (PDEs). The technique that we present in this paper shares a lot of ideas in common with non-conforming finite element methods (FEMs) [5].

The FEM tries to set up discrete sets of *square* equations; that is, the number of equations exactly matches the number of unknowns. These are delicate to set up for each PDE type [22] and require sophisticated proofs of convergence [5,15]. However, when successful, they can be very efficient. For example, Maxwell's equations which *appear* to be non-square in first-order form, can prove difficult to discretize correctly with FEM. Typically they are converted to second-order equations before being discretized [18]. The least-squares FEM [3,15], on the other hand, tries to set up a least-squares problem which can then be solved in a variety of ways. The normal equation approach can cause condition number problems and more refined least-squares approaches exist and these can handle first-order systems of PDEs *provided* the boundary conditions are of the Agmon–Douglis–Nirenberg type [3].

The method proposed in this paper is a high-order method, that relies primarily on the linear independence of the PDE and its boundary conditions along with the assumption of the existence and uniqueness of a sufficiently smooth solution. The key ideas of the method are to discretize to an *under*-determined set of equations and then to pick the solution that minimizes a diagonal Sobolev norm. Unlike least-squares method which require determination of the right norm on the range space, the method proposed here requires the right norm on the domain space, and this seems to be an easier choice.

[*] Corresponding author.
*E-mail addresses:* shiv@ece.ucsb.edu (S. Chandrasekaran), hmhaska@gmail.com (H.N. Mhaskar).

[1] This material is based upon work supported by the National Science Foundation under Grant Nos. CCF-0830604 and CCF-1450321.

[2] The research of this author was supported, in part, by Grant DMS-0908037 from the National Science Foundation and Grant W911NF-09-1-0465 from the U.S. Army Research Office.

http://dx.doi.org/10.1016/j.jcp.2015.07.025
0021-9991/© 2015 Elsevier Inc. All rights reserved.

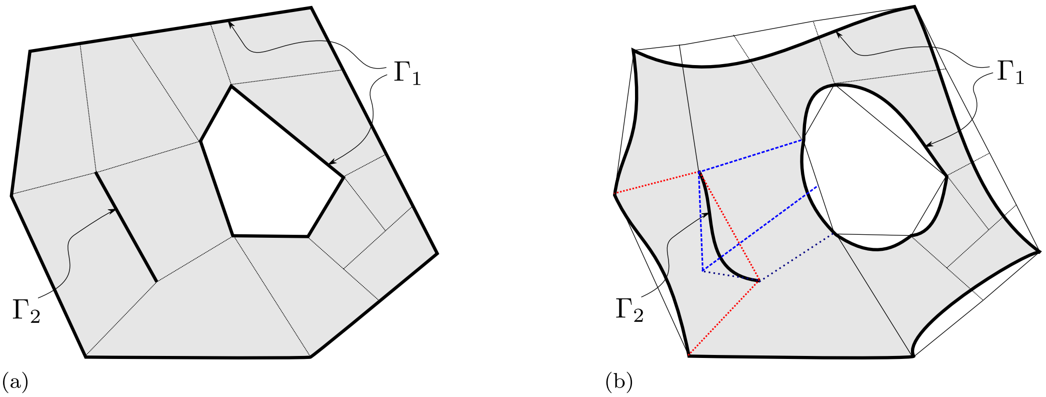

**Fig. 1.** Examples of patching. (a) Example polygonal domain showing exterior boundary $\Gamma_1$, internal boundary $\Gamma_2$, and a set of convex quadrilaterals that cover the domain. The boundaries are shown with thicker lines. The interior of the domain has been shaded a light gray. (b) Example domain showing curved exterior boundary $\Gamma_1$, curved internal boundary $\Gamma_2$, and a set of convex quadrilaterals that cover the domain. The boundaries are shown with thicker lines. The interior of the domain has been shaded a light gray. Note that the curved internal boundary $\Gamma_2$ is covered by three overlapping convex quadrilaterals; one of them has red edges, one has light blue edges, and the third has dark blue edges. The vertices are chosen to coincide with the location of the singularities of the solution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

This method has been observed to maintain high-order convergence even in the presence of curved boundaries. The method prefers that the PDE is presented in first-order system form [21]. In particular, Maxwell type equations and fourth-order equations seem to present no special problems. Some singular PDEs and over-posed PDEs have also been handled. These ideas are further clarified in the rest of the paper.

In Section 9 on numerical experiments we show a single (<400 lines) Octave code being used to solve a variety of PDEs (in two independent variables) on a variety of geometries. This is to be contrasted with existing techniques. All the code, including the geometry files and test files, is freely available from our web-site http://scg.ece.ucsb.edu/octapde.html.

## 2. Problem set-up

We restrict ourselves to PDEs in two independent variables, $(x_1, x_2) = x \in \mathbb{R}^2$. We assume that the PDE is defined on a connected open region of $\mathbb{R}^2$ that we will denote by $\Omega$. The closure of $\Omega$ will be denoted by $\bar{\Omega}$, and its boundary by $\Gamma$. Let $q$ and $p$ be positive integers. The solution of the PDE will be denoted by $u : \Omega \mapsto \mathbb{R}^p$, and we will impose some continuity onto the boundary $\Gamma$. In the open region $\Omega$, we will assume the PDE is of the form

$$\mathbf{A}_1(x)\,\partial_1 u(x) + \mathbf{A}_2(x)\,\partial_2 u(x) + \mathbf{B}(x)\,u(x) = f(x) \tag{1}$$

where $f : \Omega \mapsto \mathbb{R}^q$ and $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B} : \Omega \mapsto \mathbb{R}^{q\times p}$. Note that we do *not* assume that $q$ is equal to $p$.

We split the boundary into dis-joint parts: $\Gamma = \Gamma_1 \cup \Gamma_2$, where $\Gamma_1$ consists of all points of $\Gamma$ all of whose neighborhoods have a non-trivial intersection with the exterior of $\bar{\Omega}$ (defined as $\mathbb{R}^2 \setminus \bar{\Omega}$). We will call $\Gamma_1$ the exterior boundary of $\Omega$, and $\Gamma_2$ as the interior boundary of $\Omega$. (We will also refer to $\Gamma_2$ as the slits in the domain.) Note that $\Gamma_2$ is *not* part of the boundary of $\bar{\Omega}$. See Figs. 1(a) and 1(b). We will also assume that $\Gamma$ is made up of a finite number of infinitely differentiable finite-length curves. We will refer to the end-points of each of these smooth curves as *corner* points of $\Gamma$.

On the exterior boundary $\Gamma_1$ we will assume that the boundary conditions are local and linear of the form

$$\mathbf{C}(x)\,u(x) = g(x), \qquad x \in \Gamma_1. \tag{2}$$

The number of rows in $g(x)$ and $\mathbf{C}(x)$ is allowed to vary from point to point on $\Gamma_1$, and can even be zero on some parts of the boundary.

On the interior boundary $\Gamma_2$ we will assume that the solution $u$ satisfies a linear homogeneous jump condition. To set this up formally, we assume the existence of a vector field $\nu : \Gamma_2 \mapsto \mathbb{R}^2$. The intent is that this vector field should be non-tangential to the curve that represents $\Gamma_2$. We then impose the jump condition on $\Gamma_2$ as follows:

$$\lim_{\epsilon\to 0+} \left(\mathbf{C}_+(x)\,u(x+\epsilon\,\nu(x)) - \mathbf{C}_-(x)\,u(x-\epsilon\,\nu(x))\right) = 0, \qquad x \in \Gamma_2, \tag{3}$$

where $\mathbf{C}_+$ and $\mathbf{C}_-$ are two matrix-valued functions on $\Gamma_2$. Usually there are $p$ rows in $\mathbf{C}_+(x)$ and $\mathbf{C}_-(x)$. It is easy to allow inhomogeneous jump conditions on the internal boundary if so desired.

Our problem formulation is sufficiently general to cover most classes of systems of linear elliptic PDEs in two dimensions. The most obvious exclusion is integro-differential equations including non-local boundary conditions, but there is no obstacle in principle to extending the method to handle these types of equations too.

## 3. Overview of the method

In this section we attempt to give a brief overview of our approach. We will assume that the domain $\bar{\Omega}$ is covered by a finite number of *patches*. Patches are closed sets on which we have a countable basis (or frame) for suitable spaces of functions. On each patch the unknown solution has an expansion, and the coefficients of each expansion will be considered as unknowns.

To obtain linear equations that constrain these unknown coefficients, we pick a finite number of linear functionals that can operate on the range space of the differential operators restricted to each patch. These functionals can be chosen in many ways, including Galerkin style schemes. In this paper we will stick with Dirac delta functions—in other words, collocation schemes.

These linear functionals are applied to the PDEs that operate on the unknown solution. If $\Gamma$ has a non-trivial intersection with the patch then the linear functionals are also applied to the boundary conditions.

Since our PDEs are all local, the unknowns in each patch will only occur in equations associated with that patch. To couple these equations together we enforce continuity conditions on the intersection of two patches. Again, these can be done via some linear functionals, or, as in this paper, via a simple collocation scheme.

At this point we should have a finite number of linear algebraic equations coupling all the expansion coefficients. Unlike traditional FEM approach we do not restrict the number of expansion coefficients to match the number of discrete equations. Rather we choose to pick a minimum Sobolev norm solution for the finite number of equations.

There are two ways to achieve this. The first is the Golomb–Weinberger principle [12]. They observed that the normal equations form a finite system of linear algebraic equations that can be solved to get a representation of the solution. However, this requires computing the kernel of the matrix and this can be non-trivial. Furthermore, the resulting linear system can be highly ill-conditioned, especially if high-order convergence is desired.

The second approach is to truncate the expansion coefficients at a large number and just solve the resulting finite under-determined system of linear algebraic equations. The condition number can still be large for high-order convergence, but there exist numerical linear algebra techniques that can exploit some special properties that exist in these systems to produce accurate solutions. This is the approach taken in the paper.

Since we will compute a minimum Sobolev norm solution, we take care to ensure that the Sobolev norm of our solution remains bounded as we increase the number of equations. Now, using standard compactness arguments, we can prove that our numerical solution will converge to the true solution.

There are many technical details that must be handled with care if the above scheme is to work, and these are detailed in the remaining sections, but, this is essentially the method we are going to present in this paper.

## 4. Representation on $\bar{\Omega}$

### 4.1. Basis

We choose first-kind Chebyshev polynomials [16], $T_n(x)$, as the basis to use on each patch. While such a specific choice is not necessary from the mathematical point of view, it eases the exposition and makes the numerical issues, and their resolution, more obvious. In $\mathbb{R}^2$ we will use the tensor product of Chebyshev polynomials $T_{m,n}(x) = T_m(x_1)\, T_n(x_2)$. These form an orthonormal basis on $[-1,1]^2$ for a suitable inner product, but this domain is too restrictive for our use. We would like to be able to get a reasonably well-conditioned basis on convex quadrilaterals that are not close to being degenerate (triangular). We will refer only to these convex quadrilaterals as *patches* for the rest of the paper.

*Homographies* [14] are bilinear maps, $\phi_A : \mathbb{R}^2 \mapsto \mathbb{R}^2$, that are parametrized by a $3 \times 3$ matrix $A$, and preserve straight lines. They are defined by the following pair of equations:

$$A \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \qquad \phi_A(x) = \left( \frac{r_1}{r_3}, \frac{r_2}{r_3} \right).$$

These maps satisfy many nice properties. The first one is that given two patches there is always a homography $\phi_A$, that maps one to the other. Second, these maps will be invertible with $\phi_A^{-1} = \phi_{A^{-1}}$. Also, the Jacobian of these maps has a nice easy formula, which makes it convenient for numerical use.

Let $\phi_A$ be a homography that maps a given patch onto $[-1,1]^2$. Then we will take our basis on the patch to be the set of pulled-back 2D Chebyshev polynomials, $T_{m,n} \circ \phi_A$.

### 4.2. Patches

We will assume that the solution is reasonably smooth on $\Omega$. We allow for the fact that it might be dis-continuous across $\Gamma_2$ and at certain points of $\Gamma_1$. (More singular solutions will be studied elsewhere.)

Assume that there is a finite set of patches such that $\bar{\Omega}$ is a subset of the union of those patches. (See Figs. 1(a) and 1(b) for some examples.) We further assume that each patch is split into an *active* part and an *inactive* part such that intersection of any pair of the active parts has measure zero and $\bar{\Omega}$ is the union of the closure of the active parts. Furthermore, it is

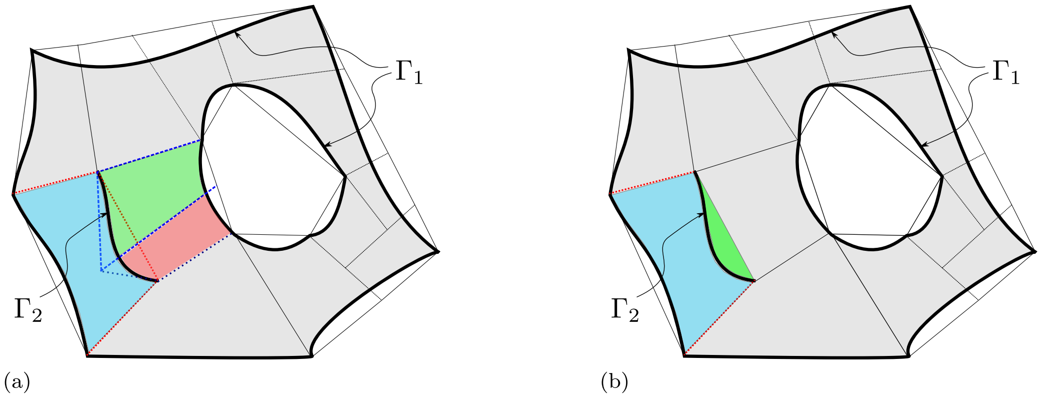

**Fig. 2.** Examples of active parts. (a) Example from Fig. 1(b) showing the active parts of the three overlapping patches. We have colored each active part in a different color. (b) Example from Fig. 2(a) showing an alternative way to cover $\Gamma_2$. In this case there are two copies of the patch with red edges. One copy has the blue region as the active part and the other has the green region as the active part. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

assumed that $\Gamma$ is contained in the boundaries of the active parts and each component of $\Gamma_2$ is in the boundary of exactly two adjacent active parts. Finally the end points of the components of $\Gamma$ are vertices of patches. (See Figs. 2(a) and 2(b) for further examples.)

Let $\phi_r$ denote the homography that maps patch $r$ to $[-1,1]^2$. On each patch $r$, the function $u$ has a representation of the form

$$u(x) = \sum_{(m,n)=(0,0)}^{(\infty,\infty)} \hat{u}_{m,n}(r)\,(T_{m,n}\circ\phi_r)(x), \qquad x \in r. \tag{4}$$

Convergence issues will be formalized later.

Now, by our assumptions, if $x \in \Omega \cup \Gamma_1$, and if $x$ does not fall on the boundary of an active part, then $x$ belongs exactly to one active part of some patch $r$. Then the coefficients $\hat{u}_{m,n}(r)$ uniquely determine $u(x)$. If $x \in \Omega \cup \Gamma_1$ falls on the boundary belonging to two active parts, then continuity of $u$ at $x$ will permit us to use either representation from the two corresponding patches to get the same value for $u(x)$. If $x \in \Gamma_2$ (and not an end-point) then $u(x)$ will in general be double-valued. From our assumptions, each $x \in \Gamma_2$ belongs at most to two active parts of two patches. Furthermore, each active part will be on "one side" of $\Gamma_2$. Hence we can determine the one-sided limits on $\Gamma_2$ using the coefficients of the corresponding patches.

### 4.3. Assumptions

**Assumption 1.** $s$ is a fixed real number strictly greater than 3.

Let $H_s(r)$ denote the set of all functions $u$ such that

$$\|\,u|_r\|_s^2 \equiv \|\hat{u}(r)\|_s^2 \equiv \sum_{m,n} \|\hat{u}_{m,n}(r)\|^2\,(1+m^2+n^2)^s < \infty.$$

Since $|T_m| \le 1$, $|T_m'| \le m^2$, $\|\phi'(r)\| < \infty$ on $[-1,1]$ and $s > 3$ it can be easily established that for all $u \in H_s(r)$, the gradient is well-defined and continuous on the whole closed patch.

**Assumption 2.** The left hand side of equation (1) is the action of a bounded operator from $H_s(r)$ to $H_{s-2}(r)$ for every patch $r$, and $f \in H_{s-2}(r)$.

Let

$$H_t([-1,1]) = \left\{ w = \sum_n w_n T_n \,\middle|\, \sum_n w_n^2 (1+n^2)^t < \infty \right\}.$$

Let $\sigma : [-1,1] \to \mathbb{R}^2$ denote a $C^\infty$ smooth simple curve in patch $r$. It is well known that if $u \in H_s(r)$ then $u \circ \sigma \in H_{s-\frac{1}{2}}([-1,1])$ [21, Theorem 7.39].

**Assumption 3.** Let $\sigma$ denote any smooth segment of $\Gamma$ that lies in some patch $r$. Then, $g \circ \sigma \in H_{s-\frac{1}{2}}([-1,1])$, and the left hand side of equation (2) can be viewed as the action of a bounded operator from $H_s(r)$ to $H_{s-\frac{1}{2}}([-1,1])$.

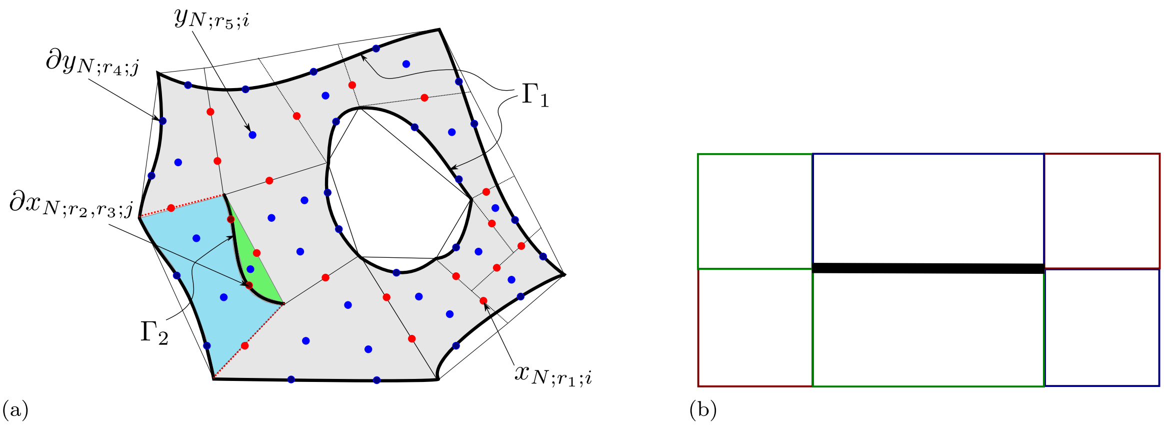

**Fig. 3.** (a) Example from Fig. 2(b) showing the placement of grid points. Points labeled $y$ lie in a single patch, while points labeled $x$ lie at the interface of two patches. Points prefixed with $\partial$ lie on the boundary $\Gamma$. (b) Rectangle $[-2,2]\times[-1,1]$ with slit $[-1,1]\times[0]$. The region is covered by 6 patches. $\Gamma_2$ is the horizontal slit in the middle denoted by the thick black line. Note that the slit is a line segment of infinitesimal width in spite of appearances.

**Assumption 4.** The matrix $[\,\mathbf{A}_1(x)\quad \mathbf{A}_2(x)\quad \mathbf{B}(x)\,]$ has full row-rank for every $x\in\Omega$; the matrix $\mathbf{C}(x)$ has full row-rank for every $x\in\Gamma_1$; and the matrix $[\,\mathbf{C}_+(x)\quad \mathbf{C}_-(x)\,]$ has full row-rank for every $x\in\Gamma_2$.

Let $H_s(P)=\cap_r H_s(r)$ denote the set of functions $u$ such that

$$\|u\|_s^2 \equiv \sum_r \|\hat{u}(r)\|_s^2 < \infty, \tag{5}$$

and note that this is a finite sum since we only have a finite number of patches.

Now consider the set of functions that are continuous on $\Omega$ and have appropriate non-tangential limits on $\Gamma$ (in particular one-sided limits along the vector field $\nu$ on $\Gamma_2$). Let us denote by $C_s(\bar{\Omega})$ the sub-set of such functions which lie in $H_s(P)$.

**Assumption 5.** The solution $u$ of the PDE (equations (1), (2), and (3)) lies in $C_s(\bar{\Omega})$, and it is the only such solution.

## 5. Collocation

### 5.1. Grid points

To collocate we choose $N$ points in $\bar{\Omega}$. The grid points are of two types (see Fig. 3(a)).

Type 1 grid points, denoted by $y_{N;r;i}$, lie in the active part of exactly one patch $r$. Even if they lie on the boundary of the active part they do *not* lie in any other patch. If $y_{N;r;i}\in\Gamma$ we will denote this point as $\partial y_{N;r;i}$. Note that $\partial y_{N;r;i}\notin\Gamma_2$ by our assumptions.

Type 2 grid points lie on the boundaries of *exactly two* active parts (this can correspond to more than two patches). We denote them by $x_{N;r_1,r_2;i}$, if they belong to the boundaries of the active parts of patches $r_1$ and $r_2$.

We require that the points for a particular $N$ be distinct and that the points become dense in $\bar{\Omega}$ as $N$ tends to infinity in the following manner:

- Suppose $y\in\Gamma_1$ and $y$ belongs to the active part of patch $r$. Then we require that for each $N$ there exists a type 1 point $\partial y_{N;r;i_N}\in\Gamma_1$ such that $\lim_{N\uparrow\infty}\partial y_{N;r;i_N}=y$.
- Suppose $y\in\Omega$ and $y$ belongs to the active part of patch $r$. Then we require that for each $N$ there exists a type 1 point $y_{N;r;i_N}\in\Omega$ such that the limit of the sequence is $y$.
- Suppose that $x\in\Gamma_2$. Then we require that for each $N$ there exists a type 2 point $\partial x_{N;r_1,r_2;i_N}\in\Gamma_2$ such that the limit of the sequence is $x$.

### 5.2. Uncoupled equations

The active part of patch $r$ contains points $y_{N;r;i}\in\Omega$ and $\partial y_{N;r;i}\in\Gamma_1$, and these points are not shared with any other active part. On this patch the unknown solution $u$ has an expansion of the form (4). Substituting this expansion in equation (1) at the point $y_{N;r;i}$, and assuming that the sum commutes with the derivatives, we obtain

$$\sum_{m,n}\left(\mathbf{A}_1(y_{N;r;i})\,\partial_1+\mathbf{A}_2(y_{N;r;i})\,\partial_2+\mathbf{B}(y_{N;r;i})\right)(T_{m,n}\circ\phi_r)(y_{N;r;i})\,\hat{u}_{m,n}(r)=f(y_{N;r;i}). \tag{6}$$

Substituting the expansion in equation (2) at the point $\partial y_{N;r;i}$ we obtain

$$\sum_{m,n} \mathbf{C}(\partial y_{N;r;i})\,(T_{m,n}\circ\phi_r)(y_{N;r;i})\,\hat{u}_{m,n}(r) = g(\partial y_{N;r;i}). \tag{7}$$

Notice that these equations only involve the unknown expansion coefficients of patch $r$. So we can gain some numerical efficiency by solving (or compressing) these equations as much as we can locally. To that end we introduce some notation.

Let $y_{N;r}$ denote the ordered set of points $y_{N;r;i}$. Similarly let $\partial y_{N;r}$ denote the ordered set of points $\partial y_{N;r;i}$. We assume that the tuples $(m,n)$ have been put in some ordering. This ordering will be assumed to be fixed throughout this paper. The notation $\partial_1 \mathbf{T}_r(y_{N;r})$ will denote the matrix whose $(i,(m,n))$-th entry is given by $\partial_1(T_{m,n}\circ\phi_r)(y_{N;r;i})$. Similarly we define the matrix $\partial_2\mathbf{T}_r(y_{N;r})$ and $\mathbf{T}_r(y_{N;r})$. Let $\hat{u}(r)$ be the column matrix whose block $(m,n)$-th entry is $\hat{u}_{m,n}(r)$. Let $f(y_{N;r})$ be the column matrix whose block $i$-th entry is $f(y_{N;r;i})$. Similarly define $g(\partial y_{N;r})$. Let $\mathbf{A}_1(y_{N;r})$ denote the block diagonal matrix whose $i$-th diagonal block is $\mathbf{A}_1(y_{N;r;i})$. Similarly define $\mathbf{A}_2(y_{N;r})$, $\mathbf{B}(y_{N;r})$ and $\mathbf{C}(\partial y_{N;r})$. Let $\otimes$ denote the Kronecker product of two matrices [13].

Now observe that equation (6) for all points $y_{N;r;i}$, and equation (7) for all points $\partial y_{N;r;i}$, can be written as the single matrix equation:

$$\begin{bmatrix} \mathbf{A}_1(y_{N;r})\,(\partial_1\mathbf{T}_r(y_{N;r})\otimes I) + \mathbf{A}_2(y_{N;r})\,(\partial_1\mathbf{T}_r(y_{N;r})\otimes I) + \mathbf{B}(y_{N;r})\,(\mathbf{T}_r(y_{N;r})\otimes I) \\ \mathbf{C}(\partial y_{N;r})\,(\mathbf{T}_r(\partial y_{N;r})\otimes I) \end{bmatrix} \hat{u}(r) = \begin{bmatrix} f(y_{N;r}) \\ g(\partial y_{N;r}) \end{bmatrix},$$

where $I$ denotes the $p\times p$ identity matrix. For ease of future use we introduce some obvious notation and re-write the above equation as

$$\mathbf{F}_r\hat{u}(r) = \mathbf{b}_r. \tag{8}$$

**Lemma 6.** *The matrix $\mathbf{F}_r$ has full row-rank.*

**Proof.** Follows from Assumption 4 and elementary properties of Chebyshev–Vandermonde matrices. □

### 5.3. Coupled equations

Collocating the continuity conditions will couple the solutions in each patch appropriately. The continuity conditions are of two types. The first ones are the *natural* continuity conditions on the points $x_{N;r_1,r_2;i}$, and the second are the *jump* conditions on the points $\partial x_{N;r_1,r_2;i}$.

The natural continuity conditions at the set of points $x_{N;r_1,r_2}$ are of the form:

$$(\mathbf{T}_{r_1}(x_{N;r_1,r_2})\otimes I)\,\hat{u}(r_1) - (\mathbf{T}_{r_2}(x_{N;r_1,r_2})\otimes I)\,\hat{u}(r_2) = 0. \tag{9}$$

The jump conditions at the set of points $\partial x_{N;r_1,r_2}$ are of the form:

$$\mathbf{C}_+(\partial x_{N;r_1,r_2})(\mathbf{T}_{r_1}(\partial x_{N;r_1,r_2})\otimes I)\,\hat{u}(r_1) - \mathbf{C}_-(\partial x_{N;r_1,r_2})(\mathbf{T}_{r_2}(\partial x_{N;r_1,r_2})\otimes I)\,\hat{u}(r_2) = 0, \tag{10}$$

where we have assumed that $\nu(\partial x_{N;r_1,r_2})$ consistently points into the active area of patch $r_1$.

**Lemma 7.** *The combined equations (8), (9) and (10) are linearly independent.*

**Proof.** Follows from Assumption 4, Lemma 6 and elementary properties of Chebyshev–Vandermonde matrices. □

For future use we introduce some notation. Let $u(x_{N;r_1,r_2}) = (\mathbf{T}_{r_1}(x_{N;r_1,r_2})\otimes I)\,\hat{u}(r_1)$, and $\tilde{u}(\partial x_{N;r_1,r_2}) = \mathbf{C}_+(\partial x_{N;r_1,r_2})\times(\mathbf{T}_{r_1}(\partial x_{N;r_1,r_2})\otimes I)\,\hat{u}(r_1)$. We can re-write all of the continuity equations together as

$$\begin{bmatrix} \mathbf{T}_{r_1}(x_{N;r_1,r_2})\otimes I \\ \mathbf{C}_+(\partial x_{N;r_1,r_2})\,(\mathbf{T}_{r_1}(\partial x_{N;r_1,r_2})\otimes I) \end{bmatrix}\hat{u}(r_1) - \begin{bmatrix} u(x_{N;r_1,r_2}) \\ \tilde{u}(\partial x_{N;r_1,r_2}) \end{bmatrix} = 0,$$

and, for future ease of use, re-write it as

$$\begin{bmatrix} \mathbf{G}_{r_1,r_2} \\ \partial\mathbf{G}_{r_1,r_2} \end{bmatrix}\hat{u}(r_1) - \begin{bmatrix} u(x_{N;r_1,r_2}) \\ \tilde{u}(\partial x_{N;r_1,r_2}) \end{bmatrix} = 0. \tag{11}$$

## 6. Convergence of Golomb–Weinberger solution

**Lemma 8.** *Let $w_N$ denote a bounded equi-continuous sequence on a compact set $X$, and let $x_N \in X$ denote a sequence of points with $x$ as the limit point and $w_N(x_N) = 0$. Then $\lim_{N\uparrow\infty} w_N(x) = 0$.*

**Proof.** From the Ascoli–Arzella theorem it follows that we have a converging sub-sequence $w_{N_k}$ with continuous limit $w$. Then

$$|w(x)| \leq |w(x) - w_{N_k}(x)| + |w_{N_k}(x) - w_{N_k}(x_{N_k})| + |w_{N_k}(x_{N_k})| \leq \epsilon_1 + \epsilon_2 + 0,$$

where by choosing $N_k$ sufficiently large we can make both $\epsilon_1$ and $\epsilon_2$ as small as we wish. Therefore it follows that $w(x) = 0$. Since $w$ came from an arbitrary converging sub-sequence of $w_N$ and $w(x)$ is exactly 0, it follows that every limit point of $w_N$ takes the value 0 at $x$. It follows that the sequence $w_N$ has only a single limit point at $x$, namely 0; that is, $\lim_{N\uparrow\infty} w_N(x) = 0$. □

**Lemma 9.** *Let $w_N$ be a sequence of functions in $H_s(r)$ with $\|w_N\|_s < M < \infty$ for some constant $M$. Let $\mathcal{L}$ be a bounded affine-linear operator from $H_s(r)$ to $H_{s-2}(r)$. Let $x_N$ be a sequence of points in the patch $r$ with limit $x$ in the patch $r$ such that $\mathcal{L}(w_N)(x_N) = 0$. Then $\lim_{N\uparrow\infty} \mathcal{L}(w_N)(x) = 0$.*

**Proof.** From the assumptions it follows that $\|\mathcal{L}(w_N)\|_{s-2} < K < \infty$ for some constant $K$. Let $w_{N_k}$ be a convergent sub-sequence of $w_N$ such that $\mathcal{L}(w_{N_k})$ is also convergent. The existence of the sub-sequence $w_{N_k}$ is guaranteed by the fact that $w_N$ and $\mathcal{L}(w_N)$ lie in compact sets, which is easy to prove. Let $w$ denote the limit of the sub-sequence $w_{N_k}$. It follows from Lemma 8 that $\mathcal{L}(w)(x) = 0$. Therefore it follows that this result holds for any convergent sub-sequence of $w_N$. Therefore the limit point of $\mathcal{L}(w_N)(x)$ is well-defined as 0. □

Note, we are *not* claiming that $w_N$ has a unique limit; only $\mathcal{L}(w_N)$ has a unique limit, because the interpolation condition is on $\mathcal{L}(w_N)$.

**Lemma 10.** *Let $w_N$ be a sequence of functions in $H_s(r)$ with $\|w_N\|_s < M < \infty$ for some constant $M$. Let $\sigma : [-1, 1] \to \mathbb{R}^2$ be a smooth simple curve in patch $r$. Let $\mathcal{L}$ be a bounded affine-linear operator from $H_s(r)$ to $H_{s-\frac{1}{2}}([-1, 1])$. Let $x_N$ be a sequence of points in $[-1, 1]$ with limit $x$, such that $\mathcal{L}(w_N)(\sigma(x_N)) = 0$. Then $\lim_{N\uparrow\infty} \mathcal{L}(w_N)(\sigma(x)) = 0$.*

**Proof.** Similar to that of Lemma 9. □

**Lemma 11.** *Let $\gamma : [-1, 1] \to \mathbb{R}^2$ be a smooth simple curve that lies simultaneously in patches $r_1$ and $r_2$ only. Let $w_N$ be a sequence of functions in $H_s(P)$ with $\|w_N\|_s < M < \infty$. Let $w_N(r_i)$ denote the restriction of $w_N$ to patch $r_i$. Let $\mathcal{L}_i$ denote bounded affine-linear operators from $H_s(r_i)$ to $H_{s-\frac{1}{2}}([-1, 1])$. Let $x_N$ be a sequence of points in $[-1, 1]$ with limit $x$, such that $\mathcal{L}_1(w_N(r_1))(\gamma(x_N)) = \mathcal{L}_2(w_N(r_2))(\gamma(x_N))$. Then $\lim_{N\uparrow\infty}(\mathcal{L}_1(w_N(r_1))(\gamma(x)) - \mathcal{L}_2(w_N(r_2))(\gamma(x))) = 0$.*

**Proof.** Follows from Lemma 10. □

Let $v_N$ represent the minimum Sobolev norm (MSN) (5) solution of equations (8), (9) and (10) with $N$ grid points. We will also refer to this as the Golomb–Weinberger solution [12], especially when the number of unknowns is infinite.

**Theorem 12.** *For $x \in \bar{\Omega}$, $\lim_{N\uparrow\infty} v_N(x) = u(x)$.*

**Proof.** Follows from Assumptions 1, 2, 3 and 5, and Lemmas 7, 9, 10, and 11. □

## 7. Truncation

The Golomb–Weinberger MSN solution, $v_N$, can in principle be computed by forming the normal equations. However the condition number of the resulting system is large even for moderate values of $s$. We therefore truncate the infinite Chebyshev series to a finite expansion and find the MSN solution by direct numerical linear algebra techniques.

Let $A_N\hat{u}_N = b_N$ denote the finite set of linear algebraic equations (8), (9) and (10), satisfied by the infinite Chebyshev expansions. Let $\hat{v}_N$ denote the Chebyshev coefficients of the Golomb–Weinberger MSN solution $v_N$. Let $\hat{w}_{N,M}$ denote the vector $\hat{w}_N$ in which only the leading terms of order less than $M$ are retained in each Chebyshev expansion. That is, the coefficient $\hat{w}_{m,n}(r)$ is retained if $m^2 + n^2 < M^2$. Similarly let $A_{N,M}$ denote the matrix obtained by dropping appropriate columns of $A_N$. Let $\hat{v}_{N,M}$ denote the minimum Sobolev norm solution of the linear algebraic equations $A_{N,M}\hat{w}_{N,M} = b_N$. Let $v_{N,M}$ denote the function whose Chebyshev coefficients are $\hat{v}_{N,M}$.

**Theorem 13.** *For each $N$ there exists an integer $M < \infty$, and a finite constant $c$ independent of $N$, such that $\|v_{N,M}\|_s \leq c\|v_N\|_s$.*

**Proof.** Let $D(s)$ denote an infinite diagonal matrix with positive diagonal entries, such that $\|D(s)\,\hat{u}\| = \|u\|_s$. $D(s)\hat{v}_N$ is the just the minimum Euclidean 2-norm solution of the equation $A_N D^{-1}(s)\hat{w}_N = b_N$. By Lemma 7 $A_N$ has linearly independent rows. So we can write the solution explicitly as

$$\hat{v}_N = D^{-1}(s)(A_N D^{-1}(s))^{\dagger} b_N = D^{-2}(s)A_N^T(A_N D^{-2}(s)A_N^T)^{-1} b_N.$$

The rows of $A_N D^{-1}(s)$ have finite Euclidean 2-norm. Therefore the matrix $A_N D^{-2}(s)A_N^T$ is a *finite* square invertible matrix.

Now assume that the unknowns in $\hat{u}_N$ (and the corresponding columns in $A_N$) are re-ordered so that we can define $E_{N,M}$ and $e_{N,M}$ by the partitionings

$$A_N = [\, A_{N,M} \quad E_{N,M} \,], \qquad \hat{u}_N = \begin{bmatrix} \hat{u}_{N,M} \\ e_{N,M} \end{bmatrix}.$$

Let the corresponding re-ordered $D(s)$ be partitioned commensurately as follows

$$D(s) = \begin{bmatrix} D_M(s) & 0 \\ 0 & G_M(s) \end{bmatrix},$$

which defines $G_M(s)$ (an infinite diagonal matrix with positive diagonal entries). Then the truncated MSN solution $\hat{v}_{N,M}$ can be written explicitly as

$$\hat{v}_{N,M} = D_M^{-1}(s)(A_{N,M}D_M^{-1}(s))^{\dagger} b_N = D_M^{-2}(s)A_{N,M}^T(A_{N,M}D_M^{-2}(s)A_{N,M}^T)^{-1} b_N,$$

if there exists a finite $M$ such that $A_{N,M}D_M^{-2}(s)A_{N,M}^T$ is invertible.

The coefficients of the PDE are all bounded point-wise on $\Omega$. Coupling this with the facts that $\|\phi_r'\|$ is bounded, $|T_m(x)| \le 1$ and $|T_m'(x)| \le m^2$ for $x \in [-1, 1]$, it follows that the entries of the matrix $A_N$ in a column associated with the index $(m, n)$ do not grow faster than $O(m^2 + n^2)$. Therefore it follows that the Frobenius norm of $A_N D^{-1}(s)$ is finite as $s > 3$.

The operator 2-norm of $G_M^{-1}(s)$ decreases like $O(1/M^s)$ as the largest entry in the diagonal matrix $G^{-1}(s)$ is $O(1/M^s)$. Therefore the Frobenius norm of $E_{N,M}G_M^{-1}(s)$ can be made as small as desired by making $M$ sufficiently large.

$A_N D^{-1}(s)$ has linearly independent rows so the smallest singular value of $A_N D^{-1}(s)$ is strictly greater than zero. Choosing $M$ so large that the Frobenius norm of $E_{N,M}G_M^{-1}(s)$ is smaller than the smallest singular value of $A_N D^{-1}(s)$, it follows from standard perturbation theory (see Theorem 5.7.1 in [13])[3] that the smallest singular value of $A_{N,M}D_M^{-1}(s)$ would be greater than zero, and hence the matrix would have linearly independent rows.

Note that $[\, \hat{v}_{N,M} \quad 0 \,]$ can be viewed as the minimum Sobolev norm solution of the infinite system with $E_{N,M}$ set to 0. Therefore, from standard perturbation theory of minimum norm solutions (see Theorem 5.7.1 in [13]), it follows that

$$\left\| D(s)\left(\begin{bmatrix} \hat{v}_{N,M} \\ 0 \end{bmatrix} - \hat{v}_N\right)\right\| \le \tau(M),$$

where $\tau(M)$ is a decreasing function of $M$ that can be made as small as desired. Hence we can choose $M$ large enough (but finite) such that $\|v_{N,M}\|_s \le c\|v_N\|_s$, for some constant $c$. □

We conjecture that $M = O(N)$ for elliptic PDEs, provided the minimum grid point separation, when viewed on the torus, behaves like $O(N^{-1/2})$. However, the proof of this estimate is long, and the details will appear in a separate paper.

## 8. Direct numerical solver

The condition number of square Chebyshev–Vandermonde matrices, even at equi-spaced grid points, can grow exponentially [4,11]. The condition number of the first derivative of a Chebyshev–Vandermonde matrix is about two orders larger [4]. However, by allowing higher degree polynomials we get fat Chebyshev–Vandermonde matrices whose condition numbers can be much milder. Intuitively, this is because the higher degree Chebyshev polynomials behave pseudo-randomly on the grid points, and their oscillations enable the condition number to improve. We are not aware of any rigorous analysis of these claims, but they are easy to check numerically. When such a fat matrix is column scaled by $D_M^{-1}(s)$ the condition number can be potentially multiplied by the condition number of $D_M(s)$, which is $O(M^s)$. It has been observed that the order of convergence of our scheme is proportional to $s$. Therefore, to use high values for $s$, we detail a numerical method that seems to be capable of dealing with these extreme condition numbers while at the same time being quite efficient.

In this section, when we truncate a matrix to $M$ columns per patch we will continue to use the notation of the untruncated version when there is no room for confusion.

We first compute a high accuracy rank-revealing decomposition [6] of the fat matrix $\mathbf{F}_r D_M^{-1}(s)$ appearing in equation (8). Since such routines are not readily available we quickly outline a poor man's version. First compute the left singular vector matrix $U$ of the SVD of $\mathbf{F}_r D_M^{-1}(s)$. This SVD is correct to a small backward error. Next sort the columns of $(U^T\mathbf{F}_r)D_M^{-1}(s)$ in decreasing order of 2-norm after forming the product in the indicated order. At this point, to a small backward error in $\mathbf{F}_r$ we should have a matrix that looks badly row-scaled. Now compute the $LQ$ factorization of the matrix with row-pivoting.

[3] The proof of Theorem 5.7.1 in [13] can be extended to handle the case when the matrix has an infinite number of columns, with each row having finite 2-norm.

The culmination of all of this is the 2-sided orthogonal decomposition $\mathbf{F}_r\, D_M^{-1}(s) \approx W\,L\,V_r^T$, where $W$ and $V_r$ are orthogonal matrices and $L = [\,L_1 \quad 0\,]$, with $L_1$ a lower triangular matrix.

Now let $V_r^T D_M(s)\,\hat{u}_{N,M}(r) = \begin{bmatrix} \breve{u}_1(r) \\ \breve{u}_2(r) \end{bmatrix}$, where the number of rows in $\breve{u}_1(r)$ is identical to that of $L$. Therefore equation (8) can be re-written as $L_1\breve{u}_1(r) = U^T\mathbf{b}_r$, and that $\breve{u}_1(r) = L_1^{-1}\mathbf{b}_r$. Note that this stage of the computation can be done in parallel for each patch if necessary. However, on single core machines we recommend doing them sequentially to conserve memory.

At this stage $\hat{u}_{N,M}(r)$ only appears in equation (11) with $r = r_1$, and $r_2$ representing those patches which have non-trivial intersection with $r$. Let $R$ denote the set of all such patches $r_2$. We gather all these equations together and write them as

$$\begin{bmatrix} \vdots \\ \mathbf{G}_{r,r_2} \\ \partial\mathbf{G}_{r,r_2} \\ \vdots \end{bmatrix}_{r_2\in R} D_M^{-1}(s)\,V_r V_r^T D_M(s)\,\hat{u}_{N,M}(r) - \begin{bmatrix} \vdots \\ u_I(N;r,r_2) \\ \vdots \end{bmatrix}_{r_2\in R} = 0, \tag{12}$$

where $u_I(N;r,r_2)$ denotes the union of $u(x_{N;r,r_2})$ and $\tilde{u}(\partial x_{N;r,r_2})$ for some ordering of the grid points. Let

$$\begin{bmatrix} \vdots \\ \mathbf{G}_{r,r_2} \\ \partial\mathbf{G}_{r,r_2} \\ \vdots \end{bmatrix}_{r_2\in R} D_M^{-1}(s)\,V_r = [\,\mathbf{G}_{r;1} \quad \mathbf{G}_2\,],$$

where the number of columns in $\mathbf{G}_{r;1}$ is the same as the number of rows in $\breve{u}_1(r)$. Then equation (12) can be written as

$$[\,\mathbf{G}_{r;1} \quad \mathbf{G}_2\,]\begin{bmatrix} \breve{u}_1(r) \\ \breve{u}_2(r) \end{bmatrix} = \begin{bmatrix} \vdots \\ u_I(N;r,r_2) \\ \vdots \end{bmatrix}_{r_2\in R}.$$

We assume that $M$ is chosen large enough that $\mathbf{G}_2$ is a fat matrix. Now repeat our previously described high-accuracy 2-sided orthogonal factorization for badly column scaled fat matrices: $\mathbf{G}_2 = U_r\,[\,L_r \quad 0\,]\,V_2^T$. Let $\begin{bmatrix} \breve{u}_3(r) \\ \breve{u}_4(r) \end{bmatrix} = V_2^T\breve{u}_2(r)$, where the number of rows in $\breve{u}_3(r)$ is the same as that of $L_r$. Since $\breve{u}_4(r)$ disappears from the current equation and appears in no other equation, and we want the minimum norm solution, we can set $\breve{u}_4(r) = 0$. We are then left with the following equation for $\breve{u}_3(r)$

$$\breve{u}_3(r) = L_r^{-1}U_r^T \begin{bmatrix} \vdots \\ u_I(N;r,r_2) \\ \vdots \end{bmatrix}_{r_2\in R} - L_r^{-1}U_r^T\mathbf{G}_{r;1}\breve{u}_1(r).$$

Note that $\breve{u}_1(r)$ is already known, whereas $u_I$ has yet to be chosen. So, the minimum Sobolev norm solution is reduced to the following sparse linear least-squares problem

$$\min_{u_I(N;\cdot,\cdot)} \sum_r \left\| L_r^{-1}U_r^T \begin{bmatrix} \vdots \\ u_I(N;r,r_2) \\ \vdots \end{bmatrix}_{r_2\in R} - L_r^{-1}U_r^T\mathbf{G}_{r;1}\breve{u}_1(r) \right\|_2^2.$$

We go ahead and solve this sparse linear least-squares problem by standard techniques [2]. We note that the size of the problem at this stage is comparable to those produced by more standard finite-element and finite-difference techniques after Gaussian elimination in nested dissection order when all unknowns interior to the patches have been eliminated. This observation gives an encouraging indication about the size of the least-squares problem. Moreover, the method is high-order, which suggests that comparable accuracy to low-order methods can be attained with coarser grids.

In spite of the high accuracy numerical techniques (rank-revealing two-sided orthogonal decompositions) that we have used in the local compressions, we have *not* provided a proof of the numerical stability and accuracy of the overall method. In Section 9 we provide extensive numerical tests that do seem to show that the method is numerically reliable. However, more extensive theoretical and experimental investigations are needed. Note that many (if not most) current high-order methods are worse off, as there are no known practical and efficient schemes to overcome their loss of accuracy due to ill-conditioning.

If there are $K$ patches and on each patch we use a $k \times k$ grid of points, then forming the sparse least squares problem takes $O(Kk^3)$ flops and seems to be the dominant cost in the `Octave` code as seen in the next section. However, formation of the least-squares problem is embarrassingly parallel and can utilize up to $K$ processors efficiently. The sparse least-squares solve can be carried out in $O(K^{1.5}p^3)$ flops with the nested dissection ordering. Therefore, we anticipate that in a more efficient `C` or `Fortran` implementation the sparse least-squares solver will become the asymptotic bottle-neck.

## 9. Numerical experiments

In this section we present a small subset of a broad range of numerical experiments that were carried out.

**Choice of** $M$. Assume that $x_i$ are $N$ distinct points on $[-1,1]^2$. Let

$$h_N = \min_{j\neq i} \| \cos^{-1}(x_i) - \cos^{-1}(x_j)\|_2,$$

where $\cos^{-1}$ is applied component-wise to the grid points $x_i$. Note that $h_N$ is proportional to the shortest distance between the grid points after they have been mapped to the torus. We then choose $M(N) = \lceil c_M\pi/h_N\rceil$ for some suitable constant $c_M$. The reason for this choice is analyzed in more detail in [8] for the special case of interpolation.

In practice, for elliptic PDEs, we have found that values of $c_M$ between 2 and 3 work very well. Note that if we chose the $N$ zeros of the $N$-th degree Chebyshev polynomial on $[-1,1]$ as the grid points, then $M(N) = N$ with $c_M = 1$. So by picking $c_M = 2$, for example, we are choosing the highest degree of the polynomial to be *twice* the classical choice. In the following experiments we chose $c_M = 2.5$ almost always. The one exception was for the hyperbolic problem in Section 9.1.7, where we chose $c_M = 6.5$.

**Choice of** $s$. Higher values of $s$ lead to large convergence rates. However, it also leads to large condition numbers. In our numerical experiments we have observed that generally the large condition numbers are not a problem as long as underflow and overflow are avoided. Typically underflow is first detected in the high-accuracy two-sided orthogonal decompositions: some diagonal entry of the lower-triangular factor will underflow to zero. This is the signal that the $s$ value is too high, or the grid is too fine in some patch. Note that as long as we use floating-point arithmetic with a fixed mantissa size, there is a limit to the number of digits of accuracy our code can produce. The only way to do better is to somehow simulate higher precision in the code. This usually leads to two orders of magnitude penalty in the runtime. So we do not attempt such tricks in our implementation. In our experiments in *double* precision IEEE arithmetic we have chosen $s$ to be 10.

**Miscellaneous notation**. To specify the experiments it is useful to make the following definitions. Let $\theta(x) = x_1/(1+x_2)$, $\lambda(x) = (1+x_2)/(1+x_1)$, $\mu(x) = (1+x_1)/(1+x_2)$, and

$$\mathcal{A}(\cdot) = \begin{bmatrix} \lambda\cos^2(\theta) + \mu\sin^2(\theta) & 0.5\,(\mu-\lambda)\sin(2\theta) \\ 0.5\,(\mu-\lambda)\sin(2\theta) & \mu\cos^2(\theta) + \lambda\sin^2(\theta) \end{bmatrix}.$$

Note that $\mathcal{A}$ is positive-definite in the positive quadrant. Let $\omega_a(x) = 1/(1+a(x_1 - x_2^2)^2)$ and $\rho_b(x) = (1+x_1^2+x_2^2)^b$, and note that $\omega_a$ is a rational function whenever $a \neq 0$, and $\rho_b$ is not a rational function or polynomial whenever $b \notin \mathbb{Z}$.

The experiments were run on a laptop with 7.7 GB RAM and a 4-core Intel i7 processor running at 2.7 GHz. The results are summarized in Table 1(a). The code was written in `Octave` [19] and is freely available from our web-site, http://scg.ece.ucsb.edu; look for the software package `Octapde`.

**Tables**. The experimental results are reported in Tables 1 to 6. We report numbers that try to show how the accuracy and running time of the computed solution increases with grid size. The first column in all tables is labeled "$h$", and shows the grid size $h$, which is defined as the largest minimum distance (mesh norm) for points in the grid for that run. In all our experiments we picked the number of grid points to be the *same* in all the patches. However, since the patches have different sizes the mesh norm varies from patch to patch. What we are reporting in the first column is the largest mesh norm.

Since it is natural for the problem to return the value of the solution on a grid that is restricted to the edges between patches, we measure the maximum error only on those grid points and then divide it by the maximum value of all solution components on the domain of the PDE. This is the number reported in the second column of all tables and labeled as "Rel. err.".

We also report the average time (in seconds) for compression per patch. We report it per patch since the compression across all patches is embarrassingly parallel and, in a real computation, this is the relevant number. Since all the code is in `Octave` and this part of the code utilizes a large number of `for loops`, these run-times are, not unexpectedly, fairly slow. It should be possible to improve these times by an order of magnitude, or more, by switching to an implementation in `C` or `Fortran`. This is the number reported in the third column of all tables and labeled as "Secs./patch".

We also report the time taken by the sparse solver (in seconds). This essentially becomes a call to the correct routine in `SuiteSparse` [10], and we believe the time taken is a better reflection of the cost of the algorithm implemented in `C` or `Fortran`. This is reported in the column labeled "Sp. solve (secs.)" in all tables.

### 9.1. Exterior of car geometry

We group our experiments according to the domains we used. In this section we consider the geometry shown in Fig. 4. The bottom left corner is at $(0,0)$ and the top right corner is at $(36,14)$.

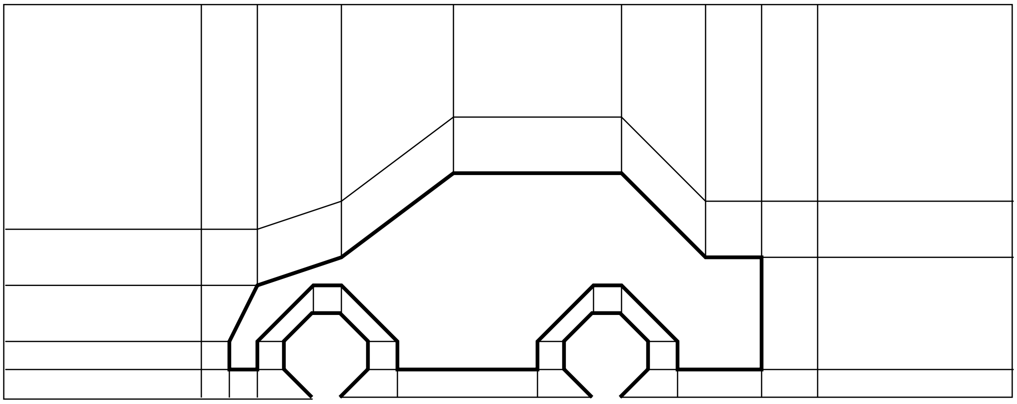

**Fig. 4.** Exterior of car-like geometry. The region has one hole, and the outer boundary is *not* a rectangle as it includes the "wheels" of the car. The region is covered by 45 patches. There are no slits in this geometry, so $\Gamma_2$ is empty.

**Table 1**
(a) Numerical results for variable coefficient div–curl on exterior of car geometry.
(b) Numerical results for variable coefficient elastic PDE on exterior of car geometry.

| $h$ | Rel. err. | Secs./patch | Sp. solve (secs.) |
|---|---|---|---|
| | | (a) | |
| 0.62 | 1E−2 | 5 | 0.5 |
| 0.35 | 1E−3 | 78 | 2.3 |
| 0.24 | 3E−4 | 599 | 6.3 |
| | | (b) | |
| 0.73 | 9E−4 | 16 | 3.4 |
| 0.62 | 3E−4 | 39 | 5.8 |
| 0.53 | 1E−4 | 84 | 8.6 |
| 0.47 | 6E−5 | 169 | 12.8 |
| 0.42 | 3E−5 | 319 | 17.4 |
| 0.38 | 1E−5 | 551 | 23.0 |
| 0.35 | 5E−6 | 960 | 30.1 |

### 9.1.1. *Variable coefficient div–curl*

On this geometry $\Omega$ we consider a natural generalization of the standard div–curl problem [1]

$$\nabla^T(Ev) = f_1, \qquad \nabla^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} v = f_2. \tag{13}$$

We choose the coefficients of a more general PDE as follows:

$$\mathbf{A}_1(\cdot) = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{A}_2(\cdot) = \begin{bmatrix} \mathcal{A}_{21} & \mathcal{A}_{22} \\ -1 & 0 \end{bmatrix}, \qquad \mathbf{B}(\cdot) = \begin{bmatrix} \mu\cos(\theta) - \lambda\sin(\theta) & \lambda\cos(\theta) + \mu\sin(\theta) \\ 0 & 0 \end{bmatrix}.$$

Let $\nu : \Gamma \to \mathbb{R}^2$ yield the normal on the boundary, and let $\tau : \Gamma \to \mathbb{R}^2$ yield the tangent. On the part of the boundary that is contiguous with the boundary of the rectangle we impose the tangential boundary condition $\mathbf{C} = \tau^T$. On the part of the boundary that comes from the car we impose the normal (more precisely, non-tangential) boundary conditions $\mathbf{C} = \nu^T \mathcal{A}$. We chose the solution to be

$$u(x) = \begin{bmatrix} \omega_1(x)\sin(x_1 - x_2) \\ \rho_{\frac{1}{4}}(x)\cos(x_1 - x_2) \end{bmatrix}.$$

Therefore $f$ and $g$ of the PDE are also determined. We skip their formulas which are best computed using a symbolic software package like `Maxima` [17]. We note that the true solution $u$ has a highly non-trivial singularity structure (especially the first component), and that this would be a problem that would tax any numerical solver.

### 9.1.2. *Variable coefficient scalar elliptic PDE*

We use the following operator notations $\nabla = \begin{bmatrix} \partial_1 \\ \partial_2 \end{bmatrix}$ and $\nabla^T = [\,\partial_1 \quad \partial_2\,]$. Consider the scalar elliptic PDE $\nabla^T \mathcal{A}\nabla v + b^T \mathcal{A}\nabla v + cv = f_1$. This can be written either as a $4 \times 3$ or a $3 \times 3$ first-order system. In our own experiments we have found no accuracy advantage in using the $4 \times 3$ formulation, so we only report results on the $3 \times 3$ formulation, as the latter is smaller and faster. To that end let $u = \begin{bmatrix} v \\ \mathcal{A}v \end{bmatrix}$. Then the coefficients of the first-order $3 \times 3$ system will be

**Table 2**
Numerical results for variable coefficient scalar elliptic PDE on exterior of car geometry. Experiments run on a desktop with 7.6 GB RAM and Intel Pentium dual core E2200 CPU running at 2.20 GHz.

| $h$ | Rel. err. | Secs./patch | Sp. solve (secs.) |
|---|---|---|---|
| 0.62 | 4E−2 | 23 | 2.4 |
| 0.35 | 1E−3 | 413 | 13.4 |
| 0.24 | 8E−5 | 3192 | 39.5 |

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 \\ \mathcal{A}_{11} & 0 & 0 \\ \mathcal{A}_{21} & 0 & 0 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 \\ \mathcal{A}_{12} & 0 & 0 \\ \mathcal{A}_{22} & 0 & 0 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} c & b_1 & b_2 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \qquad f = \begin{bmatrix} f_1 \\ 0 \\ 0 \end{bmatrix}.$$

For boundary conditions we can choose from Dirichlet: $\mathbf{C} = [\,1 \;\; 0 \;\; 0\,]$, Neumann: $\mathbf{C} = [\,0 \;\; \nu_1 \;\; \nu_2\,]$, tangential: $\mathbf{C} = [\,0 \;\; \tau_1 \;\; \tau_2\,]$, and mixed: $\mathbf{C} = [\,* \;\; * \;\; *\,]$. In this experiment we stick with Dirichlet boundary conditions. We chose $b_1(\cdot) = \mu\cos(\theta) - \lambda\sin(\theta)$, $b_2(\cdot) = \lambda\cos(\theta) + \mu\sin(\theta)$, and $c = -\sqrt{\lambda^2 + \mu^2}$. We picked the solution to be $\omega_{1/10}$. The numerical results are shown in Table 2. Note that these experiments were run on a slower machine with half the number of cores. We also point out that the reported error includes the error in the first derivatives of the solution.

### 9.1.3. *Variable coefficient elasticity equations*

We next consider the linear elasticity equations of a two dimensional body in equilibrium [23]. Let $w : \mathbb{R}^2 \to \mathbb{R}^2$ denote the displacement of the body. As usual we define the elastic strain tensor

$$\xi = \begin{bmatrix} \partial_1 & 0 \\ 0 & \partial_2 \\ \partial_2 & \partial_1 \end{bmatrix} w.$$

The elastic stress tensor $\sigma$, is then usually introduced through Hooke's law

$$\sigma = \mathcal{D}\xi = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1-v & v & 0 \\ v & 1-v & 0 \\ 0 & 0 & \frac{1}{2}(1-2v) \end{bmatrix} \xi,$$

where $E > 0$ is Young's modulus, and $-1 < v < \frac{1}{2}$ is Poisson's ratio. If $F$ is the external body force then the static equilibrium condition can be written as

$$\begin{bmatrix} \partial_1 & 0 & \partial_2 \\ 0 & \partial_2 & \partial_1 \end{bmatrix} \sigma = -F.$$

We pick our unknowns to be $u = \begin{bmatrix} w \\ \sigma \end{bmatrix}$. Note that $u$ takes values in $\mathbb{R}^5$. The full linear elasticity equations can be written as a $5 \times 5$ system with coefficients:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \mathcal{D}_{11} & 0 & 0 & 0 & 0 \\ \mathcal{D}_{12} & 0 & 0 & 0 & 0 \\ 0 & \mathcal{D}_{33} & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \mathcal{D}_{12} & 0 & 0 & 0 \\ 0 & \mathcal{D}_{22} & 0 & 0 & 0 \\ \mathcal{D}_{33} & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \qquad f = \begin{bmatrix} -F_1 \\ -F_2 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Usually one either specifies the displacement on parts of the boundary $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$, or the normal forces $\mathbf{C} = \begin{bmatrix} 0 & 0 & \nu_1 & 0 & \nu_2 \\ 0 & 0 & 0 & \nu_2 & \nu_1 \end{bmatrix}$. In our experiments we chose $v = (\mu - 2\lambda)/(2(\mu + \lambda))$, $E = \lambda$, $w = \begin{bmatrix} \omega_{1/10} \\ \rho_{3/4} \end{bmatrix}$, and displacement boundary conditions everywhere on the boundary. In Table 1b we report the results of the numerical experiments. We point out that the error we report includes the error in the elastic stress $\sigma$, and not just the displacement $w$.

### *9.1.4. Linearized stationary Navier–Stokes equation for incompressible flow*

We next consider the stationary incompressible flow Navier–Stokes equation (see [9]) linearized around a base flow $b : \mathbb{R}^2 \to \mathbb{R}^2$. Let $w : \mathbb{R}^2 \to \mathbb{R}^2$ denote the (deviation of the) velocity field and $p$ the pressure. Let $\nu$ denote the kinematic viscosity coefficient. We consider the following set of equations

$$-\nabla p + \nu \nabla^T \nabla w + (b^T \nabla) w + (w^T \nabla) b = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

$$\nabla^T w = f_3.$$

We take our unknowns to be

$$u = \begin{bmatrix} w \\ p \\ \nabla w_1 \\ \nabla w_2 \end{bmatrix}.$$

The coefficients of the first-order form of the PDE are

$$\mathbf{A}_1 = \begin{bmatrix} b_1 & 0 & -1 & \nu & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 & 0 & \nu & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} b_2 & 0 & 0 & 0 & \nu & 0 & 0 \\ 0 & b_2 & -1 & 0 & 0 & 0 & \nu \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \partial_1 b_1 & \partial_2 b_1 & 0 & 0 & 0 & 0 & 0 \\ \partial_1 b_2 & \partial_2 b_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \qquad f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Usually one specifies the flow on the boundary $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, or the pressure $\mathbf{C} = [\, 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \,]$. For the numerical experiment we chose $b = \begin{bmatrix} \lambda \\ \mu \end{bmatrix}$, $\nu = \frac{1}{10}$, $w = \begin{bmatrix} \omega_{1/10} \\ \rho_{3/4} \end{bmatrix}$, and $p(x) = \sin(x_1 - x_2)$. We chose pressure boundary conditions on the outermost left and right vertical edges of the rectangle. On the remaining parts of the boundary we chose flow boundary conditions. The results of the numerical experiments are presented in Table 3(a).

### *9.1.5. Variable coefficient scalar fourth-order elliptic PDE*

Let

$$\Box = \begin{bmatrix} \partial_1^2 \\ \partial_1 \partial_2 \\ \partial_2^2 \end{bmatrix}, \qquad \Box^T = [\, \partial_1^2 \quad \partial_1 \partial_2 \quad \partial_2^2 \,].$$

Let $\mathcal{B} : \mathbb{R}^2 \to \mathbb{R}^{3\times 3}$ take values in the set of symmetric positive-definite matrices. Let $\mathcal{C} : \mathbb{R}^2 \to \mathbb{R}^{3\times 2}$. Let

$$\mathcal{C} \circ \nabla = [\, \textstyle\sum_j \mathcal{C}_{1j} \partial_j \quad \sum_j \mathcal{C}_{2j} \partial_j \quad \sum_j \mathcal{C}_{3j} \partial_j \,].$$

Let $\mathfrak{d} : \mathbb{R}^2 \to \mathbb{R}^3$, $\mathfrak{e} : \mathbb{R}^2 \to \mathbb{R}^2$, and $\mathfrak{c}$, $w$, $f_1 : \mathbb{R}^2 \to \mathbb{R}$. We shall consider the fourth-order PDE

$$\Box^T \mathcal{B} \Box w + (\mathcal{C} \circ \nabla) \mathcal{B} \Box w + \mathfrak{d}^T \mathcal{B} \Box w + \mathfrak{e}^T \nabla w + \mathfrak{c} w = f_1,$$

for the unknown $w$. Note that this includes the bi-harmonic equation as a special case with most of the coefficients being zero, and $\mathcal{B} = \operatorname{diag}(1, 2, 1)$. For boundary conditions we will assume that both Dirichlet and Neumann data about $w$ are provided everywhere on the boundary. We choose our unknowns to be

**Table 3**
(a) Numerical results for the linearized stationary Navier–Stokes equation for incompressible flow on exterior of car geometry. (b) Numerical results for the variable coefficient fourth-order scalar elliptic PDE on exterior of car geometry.

| $h$ | Rel. err. | Secs./patch | Sp. solve (secs.) |
|---|---|---|---|
| | | (a) | |
| 1.14 | 1E−3 | 3 | 2.4 |
| 0.89 | 4E−4 | 11 | 5.0 |
| 0.73 | 1E−4 | 35 | 9.1 |
| 0.62 | 7E−5 | 91 | 14.9 |
| 0.53 | 3E−5 | 205 | 23.0 |
| 0.47 | 7E−6 | 416 | 33.3 |
| | | (b) | |
| 1.14 | 5E−3 | 6 | 5.0 |
| 0.89 | 2E−3 | 24 | 10.6 |
| 0.73 | 9E−4 | 73 | 19.1 |
| 0.62 | 4E−4 | 185 | 32.3 |
| 0.53 | 1E−4 | 422 | 48.4 |
| 0.47 | 8E−5 | 850 | 70.0 |

$$u = \begin{bmatrix} w \\ \nabla w \\ \mathcal{B}\square w \\ \operatorname{diag}(\partial_1, \partial_1, \partial_2)\mathcal{B}\square w \end{bmatrix} \in \mathbb{R}^9.$$

The coefficients of the PDE in first-order form are:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \mathcal{C}_{31} & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{B}_{11} & \mathcal{B}_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{B}_{21} & \mathcal{B}_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{B}_{31} & \mathcal{B}_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & \mathcal{C}_{12} & \mathcal{C}_{22} & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{B}_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{B}_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{B}_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \mathfrak{c} & \mathfrak{e}_1 & \mathfrak{e}_2 & \mathfrak{d}_1 & \mathfrak{d}_2 & \mathfrak{d}_3 & \mathcal{C}_{11} & \mathcal{C}_{21} & \mathcal{C}_{32} \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \nu_1 & \nu_2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

For our experiment we chose

$$\mathcal{B} = \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1+\mu^2 & \lambda \\ 0 & \lambda & 1+\lambda^2 \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} 0 & \lambda \\ \mu & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathfrak{d} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}\mu,$$

$$\mathfrak{e} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}\lambda, \quad \mathfrak{c} = \rho_{3/4}, \quad w = \omega_{1/100}.$$

**Table 4**
(a) Numerical results for Poisson's equation in polar coordinates on exterior of car geometry. (b) Numerical results for variable coefficient telegrapher's equation on exterior of car geometry. We chose $c_M = 6.5$ for these experiments.

| $h$ | Rel. err. | Secs./patch | Sp. solve (secs.) |
|---|---|---|---|
| | | (a) | |
| 0.62 | 9E−2 | 10 | 1.5 |
| 0.53 | 4E−2 | 20 | 2.1 |
| 0.35 | 6E−3 | 212 | 7.2 |
| 0.32 | 3E−3 | 351 | 10.3 |
| 0.30 | 2E−3 | 577 | 13.4 |
| | | (b) | |
| 1.14 | 9E−5 | 4 | 0.1 |
| 0.89 | 7E−5 | 13 | 0.1 |
| 0.73 | 3E−5 | 36 | 0.2 |
| 0.62 | 2E−5 | 92 | 0.3 |
| 0.53 | 5E−5 | 201 | 0.5 |

The numerical results are presented in Table 3(b). We note that the reported error includes the error in (some linear combination of) the third derivatives of the solution.

### 9.1.6. Poisson's equation in polar coordinates

Next we consider Poisson's equation in polar coordinates: $x_1^2\partial_1^2 w + x_1\partial_1 w + \partial_2 w = f_1$. We choose our unknowns to be $u = \begin{bmatrix} w \\ \nabla w \end{bmatrix}$. The coefficients of the first-order form are chosen to be:

$$\mathbf{A}_1(x) = \begin{bmatrix} x_1 & x_1^2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ 0 \\ 0 \end{bmatrix}.$$

For boundary conditions we can choose between Dirichlet, Neumann and mixed boundary conditions (see Section 9.1.2). In our experiment we chose $w(x) = x_1^{5/2}\omega_1(x)$, and Dirichlet boundary conditions everywhere. The results are presented in Table 4(a).

### 9.1.7. Variable coefficient telegrapher's equation with two-point boundary conditions

This is a hyperbolic problem. We are going to assume that the horizontal axis in the car geometry denotes time (starts at 0 and ends at 36) and the vertical axis denotes space. In particular we are assuming that the geometry of the cable varies in time. Let $V$ denote the voltage on the cable and $I$ denote the current in the cable. The coefficients of the telegrapher's equation in first-order form are

$$\mathbf{A}_1 = \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & R \\ G & 0 \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix},$$

where $C$, $L$, $R$ and $G$ are the capacitance, inductance, resistance and conductance per unit length along the cable. We choose the unknowns to be $u = \begin{bmatrix} V \\ I \end{bmatrix}$, where $V$ and $I$ denote the voltage and current respectively along the cable. It is conventional to provide Dirichlet, or Neumann or mixed boundary conditions at the ends of the cable for either $V$ or $I$, which in our case varies with time. We just choose Dirichlet for $V$ for our experiments. It is also conventional to provide initial conditions in time: $V(0,x)$ and $I(0,x)$. However we provide two-point boundary conditions instead: $V(0,x)$ and $I(36,x)$. For our experiments we chose $C = \lambda$, $L = \mu$, $R = (1/2)\lambda + \mu$, $G = \lambda + (1/2)\mu$, $V = \omega_{1/10}$, $I = \rho_{3/4}$. Therefore the cable properties not only vary in space, but they also vary with time. We conjecture that for hyperbolic problems it is generally not possible to choose $c_M$ to be a constant and expect convergence always. In Table 4(b) we can see this as the solution error stalls even though we chose $c_M$ as large as 6.5. On the other hand, our method has given quite high accuracy for a hyperbolic problem with space and time varying speed and geometry, and two-point boundary conditions.

### 9.2. Div–curl on simple slit geometry

We consider a rectangle with a horizontal straight line slit in the middle (see Fig. 3(b)). The region is covered by 6 patches. We consider the standard constant coefficient div–curl problem (13) with first-order coefficients

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.$$

**Table 5**
(a) Numerical results for div–curl problem on a rectangle with slit and single normal boundary condition on slit. (b) Numerical results for div–curl problem on a rectangle with slit and double tangential boundary conditions on the slit.

| $h$ | Rel. err. | Secs./patch | Sp. solve (secs.) |
|---|---|---|---|
| | | (a) | |
| 0.29 | 6E−4 | 0.5 | 0.001 |
| 0.22 | 3E−4 | 1 | 0.001 |
| 0.18 | 2E−4 | 2 | 0.002 |
| 0.15 | 9E−5 | 4 | 0.002 |
| 0.09 | 1E−5 | 70 | 0.012 |
| 0.06 | 3E−6 | 533 | 0.035 |
| 0.05 | 1E−6 | 4661 | 0.182 |
| | | (b) | |
| 0.29 | 1E−3 | 0.5 | 0.001 |
| 0.22 | 5E−4 | 1 | 0.001 |
| 0.18 | 2E−4 | 2 | 0.002 |
| 0.15 | 1E−4 | 4 | 0.003 |
| 0.09 | 1E−5 | 67 | 0.012 |
| 0.06 | 4E−6 | 523 | 0.037 |
| 0.05 | 1E−6 | 4417 | 0.280 |

For the solution $u : \mathbb{R}^2 \to \mathbb{R}^2$ we pick it as follows. Consider the function $(z^2 - 1)^{3/2}$ of a complex variable $z$, where the branch cut is chosen to be the line segment $[-1, 1]$ on the real axis, which coincides with the location of the slit in our geometry. Let us make the following notation for the real and imaginary parts $(z^2 - 1)^{3/2} = u_R + \iota u_I$, where $\iota^2 = -1$. One can check that $u_R$ is continuous across the slit, while $u_I$ has a jump discontinuity across the slit. We pick our solution to be $u = \begin{bmatrix} u_I \\ u_R \end{bmatrix}$.

### 9.2.1. *Single normal boundary condition on slit*

For boundary conditions we pick tangential conditions on the outer boundary. On the slit we pick a *single* set of normal boundary conditions. Note that this works since the normal on the slit is $[\,0 \quad 1\,]^T$, and the second component of $u$ is indeed continuous across the boundary. The numerical results are presented in Table 5(a).

### 9.2.2. *Double tangential boundary conditions on slit*

For boundary conditions we pick normal conditions on the outer boundary. On the slit we pick a *double* set of tangential boundary conditions; one set is as we approach the slit from the top and the other as we approach the slit from below. Note that this works since the tangential on the slit is $[\,1 \quad 0\,]^T$, and the first component of $u$ has two values on the boundary, depending on the direction in which you approach it, either from above or below. The numerical results are presented in Table 5(b).

## 9.3. *Generalized div–curl with dis-continuous coefficients*

We consider the generalized variable coefficient div–curl problem (see Section 9.1.1) with dis-continuous coefficients. We choose the coefficients in first-order form to be

$$\mathbf{A}_1(\cdot) = \begin{bmatrix} \mathcal{F}_{11} & \mathcal{F}_{12} \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{A}_2(\cdot) = \begin{bmatrix} \mathcal{F}_{21} & \mathcal{F}_{22} \\ -1 & 0 \end{bmatrix}, \qquad \mathbf{B}(\cdot) = \begin{bmatrix} \mu\cos(\theta) - \lambda\sin(\theta) & \lambda\cos(\theta) + \mu\sin(\theta) \\ 0 & 0 \end{bmatrix},$$

where $\mathcal{F}$ is dis-continuous on a set of curves in $\Omega$. The standard continuity conditions on these curves for the solution $u$ are

$$\begin{bmatrix} \nu^T \mathcal{F}_+ \\ \tau^T \end{bmatrix} u_+ = \begin{bmatrix} \nu^T \mathcal{F}_- \\ \tau^T \end{bmatrix} u_-,$$

where plus denotes values when approaching non-tangentially from one side of the curve and minus denotes values approaching non-tangentially from the other side of the curve. We use the geometry in Fig. 5(a) and choose

$$\mathcal{F}\,|_{P_1 \cup P_3} = \mathcal{A}, \qquad \mathcal{F}\,|_{P_2} = \begin{bmatrix} \mu\cos^2(\theta) + \lambda\sin^2(\theta) & 0.5\,(\lambda - \mu)\sin(2\theta) \\ 0.5\,(\lambda - \mu)\sin(2\theta) & \lambda\cos^2(\theta) + \mu\sin^2(\theta) \end{bmatrix},$$

where $P_i$ denotes patch $i$ in Fig. 5(a). $\mathcal{F}$ is dis-continuous across the two edges that $P_2$ shares with $P_1$ and $P_3$. We choose our solution to be

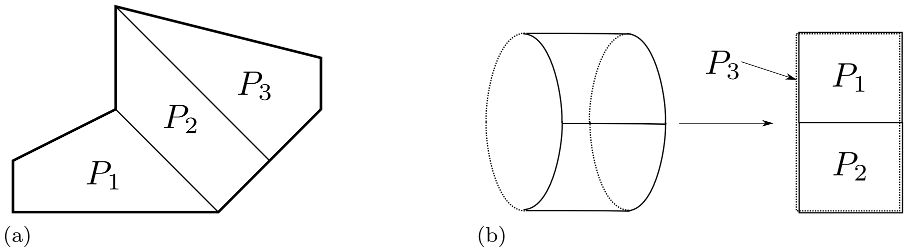

**Fig. 5.** (a) Polygonal region covered by 3 patches labeled $P_1$, $P_2$ and $P_3$. The region is contained in the rectangle $[0,3] \times [0,2]$. Note that $P_2$ is a trapezoid with its two parallel sides making an angle of 45° with the horizontal. This is exploited in constructing the test solution in Section 9.3. (b) Cylindrical surface covered by 3 patches. Note that in $\mathbb{R}^2$ patches $P_1$ and $P_2$ overlay patch $P_3$ (dotted outline). Patches $P_1$ and $P_2$ are squares, while patch $P_3$ is a tall rectangle that exactly matches $P_1 \cup P_2$. There are two left vertical boundary edges, and two right vertical boundary edges. The top and bottom horizontal edges are not part of the boundary. $P_3$ is the rectangle $[0,1] \times [0,2]$.

**Table 6**
(a) Numerical results for variable coefficient dis-continuous div–curl problem on a polygonal three patch domain. (b) Numerical results for Poisson's equation in polar coordinates on a cylindrical geometry with singular solution.

| $h$ | Rel. err. | Secs./patch | Sp. solve (secs.) |
|---|---|---|---|
| | | (a) | |
| 0.18 | 1E−3 | 2 | 0.001 |
| 0.15 | 5E−4 | 4 | 0.001 |
| 0.10 | 5E−6 | 45 | 0.001 |
| 0.09 | 2E−6 | 74 | 0.001 |
| 0.08 | 8E−7 | 115 | 0.001 |
| 0.06 | 2E−8 | 557 | 0.003 |
| | | (b) | |
| 0.18 | 1E−2 | 4 | 0.002 |
| 0.10 | 8E−6 | 125 | 0.009 |
| 0.06 | 2E−6 | 1241 | 0.027 |

$$u\,|_{P_1 \cup P_3} = \frac{1}{\lambda + \mu + (\mu - \lambda)\sin(2\theta)} \begin{bmatrix} 1 & 0.5(\lambda - \mu)\sin(2\theta) - \mu\cos^2\theta - \lambda\sin^2\theta \\ 1 & \lambda\cos^2\theta + \mu\sin^2\theta + 0.5(\mu - \lambda)\sin(2\theta) \end{bmatrix} \begin{bmatrix} \omega_{1/10} \\ \rho_{3/4} \end{bmatrix},$$

$$u\,|_{P_2} = \frac{1}{\lambda + \mu + (\lambda - \mu)\sin(2\theta)} \begin{bmatrix} 1 & 0.5(\mu - \lambda)\sin(2\theta) - \lambda\cos^2\theta - \mu\sin^2\theta \\ 1 & \mu\cos^2\theta + \lambda\sin^2\theta + 0.5(\lambda - \mu)\sin(2\theta) \end{bmatrix} \begin{bmatrix} \omega_{1/10} \\ \rho_{3/4} \end{bmatrix}.$$

The matrices that arise in the above formulas are essentially the inverses of the appropriate jump operators. This solution satisfies the jump conditions on the edges that $P_2$ shares with $P_1$ and $P_3$. The results from the numerical experiments are shown in Table 6(a).

### *9.4. Poisson's equation in polar coordinates on cylindrical geometry*

We re-consider Poisson's equation in polar coordinates (see Section 9.1.6), on a domain topologically equivalent to the surface of a finite cylinder (without caps). See Fig. 5(b). We chose $w(x) = x_1^{5/2}\cos(5\pi x_2/2)$ as the solution. The results of the numerical experiments are shown in Table 6(b).

## 10. Conclusion

We have presented a method for the numerical solution of partial differential equations. The main idea was to use more terms in the bases expansion of the solution than there were equations. The function that minimizes a diagonal Sobolev norm was chosen as the numerical solution. A convergence proof was presented for the Golomb–Weinberger solution. Based on this it was shown that truncating the bases expansions liberally would not loose convergence. Detailed discussion of the numerical solver that can overcome the high ill-conditioning associated with large Sobolev parameter values was presented. The numerical solver was also designed to minimize memory consumption. Several numerical experiments covering a gamut of PDEs and geometries showed the code in action.

In the future the code will be improved to handle inhomogeneous jump conditions and curved boundaries without loss of order of convergence. Applications to eigenvalue problems and non-linear PDEs is being investigated. A detailed theoretical study of the truncation parameter and order of convergence of the method is being carried out. Replacing collocation with Galerkin style functionals is also being investigated.

## Acknowledgements

We would like to thank Professors Yu Chen, Froilan Dopico, Ming Gu and John Strain for useful discussions. We would also like to thank Karthik Jayaraman Raghuram and Joseph Moffitt for help with the early experiments and code [7,20].

## References

[1] Giles Auchmuty, James C. Alexander, $L^2$ well-posedness of planar div–curl systems, Arch. Ration. Mech. Anal. 160 (2001) 91–134.
[2] Ake Bjorck, Numerical Methods for Least-Squares Problems, SIAM, 1996.
[3] Pavel B. Bochev, Max D. Gunzburger, Least-Squares Finite Element Methods, Springer, 2009.
[4] John P. Boyd, Chebyshev and Fourier Spectral Methods, Dover Publications, 1999.
[5] Susanne C. Brenner, L. Ridgway Scott, The Mathematical Theory of Finite Element Methods, Springer, 2008.
[6] Nieves Castro-Gonzalez, Johan Ceballos, Froilan M. Dopico, Juan M. Molera, Accurate solution of least squares problems via rank-revealing decompositions, SIAM J. Matrix Anal. Appl. 34 (2013) 1112–1128.
[7] S. Chandrasekaran, K.R. Jayaraman, M. Gu, H.N. Mhaskar, J. Moffitt, Higher order numerical discretization methods with Sobolev norm minimization, in: Proceedings of the International Conference on Computational Science, ICCS 2011, in: Proc. Comput. Sci., vol. 4, 2011.
[8] S. Chandrasekaran, H.N. Mhaskar, K.R. Jayaraman, Minimum Sobolev norm interpolation with trigonometric polynomials on the torus, J. Comput. Phys. 249 (2013) 96–112.
[9] A. Chorin, J.E. Marsden, A Mathematical Introduction to Fluid Mechanics, Springer-Verlag, 1992.
[10] T. Davis, Algorithm 915: SuiteSparseQR: multifrontal multithreaded rank-revealing sparse QR factorization, ACM Trans. Math. Softw. 38 (1) (2011).
[11] W. Gautschi, The condition of Vandermonde-like matrices involving orthogonal polynomials, Linear Algebra Appl. 52/53 (1983) 293–300.
[12] M. Golomb, H.F. Weinberger, Optimal approximation and error bounds, in: R.E. Langer (Ed.), Proceedings of a Symposium on Numerical Approximation, Madison, April 21–23, 1958, in: Publ. Army Math. Res. Center, vol. 1, University of Wisconsin Press, Madison, WI, 1958, pp. 117–190.
[13] G.H. Golub, C.F. van Loan, Matrix Computations, third edition, Johns Hopkins University Press, 1996.
[14] Richard Hartley, Andrew Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.
[15] Mats G. Larson, Fredrik Bengzon, The Finite Element Method: Theory, Implementation, and Applications, Springer, 2013.
[16] J.C. Mason, D.C. Handscomb, Chebyshev Polynomials, Chapman & Hall/CRC, 2003.
[17] Maxima, a computer algebra system, http://maxima.sourceforge.net/.
[18] Peter Monk, Analysis of a finite element method for Maxwell's equations, SIAM J. Numer. Anal. 29 (3) (1992) 714–729.
[19] GNU Octave, http://www.gnu.org/software/octave/.
[20] K.J. Raghuram, S. Chandrasekaran, J. Moffitt, M. Gu, H. Mhaskar, Higher order numerical discretizations for exterior and biharmonic type PDEs, J. Comput. Appl. Math. 236 (18) (December 2012) 4762–4774.
[21] M. Renardy, R.C. Rogers, An Introduction to Partial Differential Equations, second edition, Springer, 2003.
[22] Zhong-Ci Shi, Nonconforming finite element methods, J. Comput. Appl. Math. 149 (2002) 221–225.
[23] S.P. Timoshenko, J.M. Gere, Theory of Elastic Stability, McGraw–Hill, 1985.