Two-Dimensional Inductor-Capacitor Lattice Synthesis

Harish S. Bhat and Braxton Osting

Abstract—We consider a general class of two-dimensional passive propagation media, represented as a planar graph where nodes are capacitors connected to a common ground and edges are inductors. Capacitances and inductances are fixed in time but vary in space. Kirchhoff’s laws give the time dynamics of voltage and current in the system. By harmonically forcing input nodes and collecting the resulting steady-state signal at output nodes, we obtain a linear, analog device that transforms the inputs to outputs. We pose the lattice synthesis problem: given a linear transformation, find the inductances and capacitances for an inductor-capacitor circuit that can perform this transformation. Formulating this as an optimization problem, we numerically demonstrate its solvability using gradient-based methods. By solving the lattice synthesis problem for various desired transformations, we design several devices that can be used for signal processing and filtering.

Index Terms—lattice synthesis, analog circuit design, device sizing, Kirchhoff’s laws, inductor-capacitor lattice

I. INTRODUCTION

We investigate a general class of two-dimensional passive propagation media that can be used for signal processing and filtering. These media consist of two-dimensional (2-D) inductor-capacitor (LC) lattices, an example of which is shown in Fig. 1, with spatially varying inductance and capacitance. The lattice is a natural generalization of the one-dimensional transmission line. The 2-D LC lattice was first explored by Léon Brillouin [1], who showed its equivalence to 2-D mass-spring lattices used to model crystals.

In this paper, the input \( f e^{2\pi j t} \) is applied to node \( j \) on the left boundary of the lattice and the steady-state output \( g e^{2\pi j t} \) is tapped from node \( j \) on the right boundary. The choice of inductance \( L \) and capacitance \( C \) vectors defines a transfer function from the inputs to the outputs. If there are \( m \) rows in the lattice, then for a fixed basis in \( \mathbb{C}^m \), the transfer function can be represented by an \( m \times m \) complex matrix, denoted \( T = T(L, C) \). Note that \( T \) is a linear transformation from \( f \) to \( g \), but \( T \) depends nonlinearly on \( L \) and \( C \).

The central result of this paper is the derivation and demonstration of an algorithm that accepts as input a desired transfer matrix \( T_d \) and produces as output a 2-D LC lattice whose transfer matrix is very close to \( T_d \). We formulate this as the following optimization problem:

\[
(L^*, C^*) = \arg \min_{(L, C)} \| T(L, C) - T_d \|_F^2,
\]

where \( \| \cdot \|_F \) is the Frobenius norm. We cannot expect this optimization problem to be solvable for all possible matrices \( T_d \); however, we demonstrate that a large class of transfer matrices can be attained, with the norm difference between the true and desired transfer matrices on the order of \( 10^{-5} \). Our approach to solving the design problem can be generalized to lattice topologies other than the one chosen here.

The general outline of our paper is as follows. In Section II, the synthesis problem is formulated as an optimization problem. The objective function makes use of the transfer matrix steady-state solution of Kirchhoff’s laws on the lattice. The gradient and Hessian of the objective function are calculated analytically in Section III. In Section IV, we define design variables that reduce the dimensionality of the problem. In Section V, we present and discuss numerical solutions of the optimization problem formulated in this paper. We solve the design problem for four different transfer functions: (A) a diagonal transfer matrix, (B) a rank-one projection, (C) a low-pass filter, and (D) a power combiner/funnel. For the low-pass filter, we present results on the robustness of the optimal solution. Finally, we present two results on the ill-posedness of the synthesis problem.

A. Motivation and Context

The motivation for this work stems from a number of analog devices that operate in the 30-400 GHz range. Earlier work [2] demonstrated that an inhomogeneous 2-D LC lattice could be used as a power combiner, which was used to implement...
a power amplifier that generates 125mW at 85 GHz [3], more than three times the maximum reported power output for an amplifier in the same frequency range on a silicon substrate. Electrical prisms [4], filters that spatially separate the frequency content of an input signal, have been designed using 2-D LC lattices, implemented on chip, tested using 30-50 GHz inputs, and shown to have quality factors from 8 to 12. Simulations show that these filters should scale up to 200-400 GHz. Other work shows that 2-D LC lattices can be used to design a 4-bit quantizer that can process $2 \times 10^{10}$ samples/sec consuming 194 mW [5], as well as a device that performs discrete Fourier transforms in space [6].

Because the 2-D LC lattice consists only of passive components, it has the desirable properties of high cut-off frequency, low latency, and high throughput, especially as compared with active-device solutions on the same substrate [2].

This paper represents a first step towards automatic synthesis of 2-D LC lattices that can be used in high-frequency analog devices. We develop a framework to study and design these lattices, potentially including all applications listed above. Of course, framing the synthesis problem in the language of optimization does not guarantee its solvability. In this paper, we give computational evidence that, for a large class of desired transfer matrices, the synthesis problem is solvable using gradient-based algorithms.

We now place our problem in the context of problems that have appeared in the literature.

### Analog Circuit Design / Device Sizing

The idea of using optimization to synthesize analog circuits has been explored by many authors [7]–[12] for a variety of figures of merit. One popular approach wraps an optimization method, either gradient- or stochastic-based, around existing circuit simulation software, such as HSPICE or Spectre. There are several tools that employ this strategy, such as SPICE OPUS [13] and DELIGHT.SPICE [14], which can be used for sizing up to $\approx 100$ components. There have also been numerous efforts to use genetic algorithms and neural networks for analog device synthesis—see, e.g., [15] and [12, Chap 3.3.3].

Another approach is to mathematically model a circuit and then apply optimization to the model. Examples include [16], where convex programming is applied to a posynomial model of an op-amp; [17, 18], where Newton and quasi-Newton methods are applied to Kirchhoff’s law models of small analog circuits; and [19], where transistor-level simulations are used to fit quadratic models that are then optimized using geometric programming. For larger problems, hierarchical methods, which build large devices from smaller ones, may be applied [8, 10, 11]; a key step is the use of device-level simulations to extract macromodels that can be used for synthesis.

Our focus in this paper on the design of 2-D LC lattices has important ramifications for the structure and size of the resulting optimization problem and leads to several differences from the works cited above.

First, the 2-D LC lattice is, by definition, a multiple-input, multiple-output (MIMO) device. A vector of inputs applied to the left boundary is transformed spatially into a vector of outputs at the right boundary. While [17, 18] do use Kirchhoff’s law models and gradient-based optimization in much the same way we do, these works synthesize single-input, single-output (SISO) devices. Such devices operate in the time domain, and a typical application is pulse shaping.

One way that the difference between SISO and MIMO design manifests itself is that the optimization problem framed in this paper involves more degrees of freedom than considered in the above works. For a 2-D LC lattice of size $m \times n$, there are $N = (3m - 1)n$ unknown lattice components; note that in Section V-D, we design a $31 \times 31$ lattice where $N = 2,852$.

Second, the structure of our optimization problem allows us to fruitfully derive and apply analytical expressions for (i) the solution of the forward problem and (ii) the derivatives of the objective function. Using these analytical expressions in conjunction with quasi-Newton methods is what makes the design problem tractable, especially at large lattice sizes. Other analytical approaches for the forward problem have been explored [20]–[22] and may, in future works, be applied to the optimization problem as well.

Finally, we seek to synthesize a 2-D LC lattice from scratch, rather than improve upon an existing design, in contrast to some of the above papers and also, e.g., [23]–[26].

### Inverse Problems

We first mention transmission line synthesis: given a finite 1-D LC lattice, an input $f(t)$, and an output $g(t)$, solve for $(L, C)$ such that when we apply $f(t)$ to one side of the 1-D LC lattice, we obtain $g(t)$ at the other side. This problem was solved 30 years ago using inverse scattering [27]–[29]—here, $f(t)$ and $g(t)$ are prescribed for all $t$, including both transient and steady-state responses. In contrast, for 2-D LC lattice synthesis, we assume time-harmonic inputs and consider only the steady-state output.

Two-dimensional electromagnetic inverse problems have been considered by numerous authors, e.g., [30]–[33]. These problems are posed on infinite, continuous domains. Far-field scattering data is used to reconstruct unknown parameters $\varepsilon(x, y)$ and/or $\mu(x, y)$, assumed to be inhomogeneous within a compact region. Related work [34]–[36] seeks to design electromagnetic devices that either have prescribed radiative behavior in the far field, or that have optimal values of various far-field figures of merit, e.g., directivity, gain, and signal-to-noise ratio. In 2-D LC lattice synthesis, the domain is discrete and finite, and the output signal is collected immediately adjacent to the scattered obstacle, a completely different regime.

Inverse problems on lattices of resistors have been extensively studied by, e.g., [37], [38]. Like 2-D LC lattice synthesis, these problems are discrete inverse problems on finite domains. The goal is to reconstruct the conductivity in the interior of the lattice using measurements made using DC sources on the boundary. The resistor lattice is fundamentally different from the LC lattice: the forward problem for a resistor lattice is a discretization of the heat equation, and its steady-state solution is a smooth distribution. For 2-D LC lattices, on the other hand, the forward problem is a discretization of Maxwell’s equations for spatially varying $\varepsilon$ and $\mu$ [39], and the steady-state solution is a superposition of standing waves.
II. FORMULATION OF THE SYNTHESIS / DESIGN PROBLEM

The notation and formulation developed in this section is similar to that in [39], where we discuss the continuum limit of Kirchhoff’s laws on a lattice.

We consider a 2-D rectangular LC lattice, as shown in Fig. 1, which we represent as an oriented, planar graph, c.f. [40, Chap. 13]. Nodes represent capacitors and edges represent inductors. The orientation of the edge represents the direction of positive current flow through the associated inductor.

In a lattice of size $m \times n$, there are $mn$ nodes and $(2m-1)n$ edges, $mn$ horizontal ones and $(m-1)n$ vertical ones. Let $\mathcal{N} = \{1, 2, \ldots, mn\}$ denote the set of all nodes, and $\mathcal{E} = \{1, 2, \ldots, (2m-1)n\}$ the set of all edges. Let $\mathbf{C}$ be a vector of size $mn$ such that $C_j$ is the capacitance at node $j$. Let $\mathbf{L}$ be a vector of size $(2m-1)n$ such that $L_j$ is the inductance at edge $j$. We decompose $\mathbf{L} = [L_k, L_t]$ into the horizontal and vertical inductors, respectively. We denote by $V_j(t)$ the voltage across capacitor $j$ and by $I_k(t)$ the current across inductor $k$ at time $t$. By $\mathbf{V}(t)$ and $\mathbf{I}(t)$ we denote the vectors of all voltages and currents, respectively.

Of the horizontal edges, there are $m$ boundary edges that form a subset $\Gamma \subset \mathcal{E}$, each of which is incident upon only one node. In Fig. 1, $\Gamma$ is the left-most column of horizontal edges. All other edges in the graph are incident upon two nodes. In general, an edge is an ordered pair $(i_1, i_2)$, where $i_1 \in \mathcal{N}$. The direction of the edge is given by the order of these numbers, so that $i_1$ is the tail and $i_2$ is the head. For a boundary edge $j$ that is incident only upon node $i$, we write $j = (i, 0)$.

Let $\mathcal{B}$ denote the $|\mathcal{N}| \times |\mathcal{E}| = mn \times (2m - 1)n$ incidence matrix for the oriented graph that represents our circuit. Then

$$\mathcal{B}_{ij} = \begin{cases} 1 & \text{if } j = (i', i) \text{ for some } i' \in \mathcal{N} \cup \{0\} \\ -1 & \text{if } j = (i, i') \text{ for some } i' \in \mathcal{N} \\ 0 & \text{otherwise}. \end{cases}$$

The matrix $\mathcal{B}$ will be used shortly to write Kirchhoff’s laws in a compact form.

In addition to the structure described already, the 2-D rectangular LC lattice also has resistors and forcing along the boundary. We represent the set of nodes connected to resistors by $\mathcal{R} \subset \mathcal{N}$, and let $G_i$ be the conductance of node $i \in \mathcal{R}$. We then extend $G_i$ by defining $G_i \equiv 0$ for all $i \in \mathcal{N} \setminus \mathcal{R}$, so that $\mathbf{G} = (G_1, \ldots, G_{mn})$ is a vector in $\mathbb{R}^{mn}$.

Let $\mathcal{N} = |\mathcal{N}| + |\mathcal{E}| = 3(mn - 1)n$. Then we define the $|\mathcal{N}| \times N = m \times (3m - 1)n$ projection matrix $P_1$ by $P_{1ij} = 1$ if $\Gamma_j = i$ and $(P_1)_{ij} = 0$ otherwise. Note that because $\Gamma_i \in \mathcal{E}$, the final $mn$ columns of $P_1$ are all zero. The forcing applied at edges $\Gamma$ is given by $\mathbf{W}(t) = P_1 \mathbf{f} e^{2\pi i \omega t}$, where $\mathbf{f} \in \mathbb{C}^{mn}$.

Kirchhoff’s Laws on this inductor-capacitor lattice can now be written in the following matrix-vector form:

$$\begin{align*}
\text{diag}(\mathbf{L}) \frac{d\mathbf{I}}{dt} &= -\mathcal{B}^t \mathbf{V} + \mathbf{W} \quad (1a) \\
\text{diag}(\mathbf{C}) \frac{d\mathbf{V}}{dt} &= \mathcal{B} \mathbf{I} - \text{diag}(\mathbf{G}) \mathbf{V} \quad (1b)
\end{align*}$$

Define $\mathbf{z}(t) = (\mathbf{I}(t), \mathbf{V}(t))$ so for each $t$, $\mathbf{z}(t) \in \mathbb{C}^N$. Define $M(\mathbf{G}) = \begin{bmatrix} 0 & -\mathcal{B}^t \\ \mathcal{B} & -\text{diag}(\mathbf{G}) \end{bmatrix}$.

Then the system (1) can be written in the form

$$\text{diag}(\mathbf{L}) \frac{d\mathbf{z}}{dt} = M(\mathbf{G}) \mathbf{z}(t) + P_1 \mathbf{f} e^{2\pi i \omega t}. \quad (2)$$

Let $\Upsilon \subset \mathcal{R}$ denote the vector of right boundary nodes. Let $P_{\Upsilon}$ be the $|\Upsilon| \times N$ projection matrix defined by $(P_{\Upsilon})_{ij} = 1$ if $\Upsilon_j = j$ and $(P_{\Upsilon})_{ij} = 0$ otherwise. Note that because $\Upsilon_i \in \mathcal{R}$, columns 1 to $|\mathcal{E}| = (2m - 1)n$ of $P_{\Upsilon}$ are all zero.

**Forward Problem.** Let $\mathbf{z}(t) = \mathbf{u} e^{2\pi i \omega t}$. Then the forward problem is to find $\mathbf{g} = P_{\Upsilon} \mathbf{u}$ given $\mathbf{f}$, $\mathbf{L}$, $\mathbf{C}$, and $\mathbf{G}$. Using the Fourier transform, one can show that the solution of the forward problem is

$$\mathbf{f} \rightarrow \mathbf{g} = P_{\Upsilon} \left(2\pi i \omega \text{diag}(\mathbf{L}, \mathbf{C}) - M(\mathbf{G})\right)^{-1} P_1 \mathbf{f}. \quad (3)$$

Given $\text{diag}(\mathbf{L}, \mathbf{C}, \mathbf{G})$, we define the transfer matrix to be:

$$T(\mathbf{L}, \mathbf{C}, \mathbf{G}) := P_{\Upsilon} \left(2\pi i \omega \text{diag}(\mathbf{L}, \mathbf{C}) - M(\mathbf{G})\right)^{-1} P_1. \quad (4)$$

We have formulated the circuit as an oriented graph in order to write the equations compactly and take advantage of the graph-theoretic interpretation of the incidence matrix $\mathcal{B}$, which appears naturally in Kirchhoff’s laws. Though we have formulated the problem for an $m \times n$ rectangular lattice, the beauty of the graph-theoretic framework outlined above is that it easily accommodates other lattice topologies.

Note that since (3) is invariant under the transformation $\alpha \mapsto r \alpha$ and $(\mathbf{L}, \mathbf{C}) \mapsto r^{-1}(\mathbf{L}, \mathbf{C})$, any lattice with values $(\mathbf{L}, \mathbf{C})$ which performs a transfer function at frequency $\alpha$ can be rescaled by a factor of $\alpha'/\alpha$ to create a lattice that performs the same function at frequency $\alpha'$.

**Design / Synthesis Problem.** We define the admissible set

$$\mathcal{A} := \{(\mathbf{L}, \mathbf{C}, \mathbf{G}) : \mathbf{L} \leq L_i < T_i \quad \text{for all } i \in \mathcal{E}, \quad \mathbf{C} \leq C_i < C_i \quad \text{for all } j \in \mathcal{N}, \quad \mathbf{G} \leq G_j < G_j \quad \text{for all } j \in \mathcal{S} \subset \mathcal{N}\}$$

where $\mathcal{L}, T_i, C_i, G_i$ and $C_i$ are constants. Let

$$\{(f^i, g^i) \mid 1 \leq i \leq p\}$$

be a collection of desired input-output pairs. The design problem is: find $(\mathbf{L}, \mathbf{C}, \mathbf{G}) \in \mathcal{A}$ such that for each $i$, the steady-state output $Tf^i$ generated by input $f^i$ is equal to $g^i$. We formulate this as the constrained optimization problem:

$$\min_{(\mathbf{L}, \mathbf{C}, \mathbf{G}) \in \mathcal{A}} J(\mathbf{u}^i) := \frac{1}{2} \sum_{i=1}^{p} \left\| P_{\Upsilon} \mathbf{u}^i - \mathbf{g}^i \right\|^2 \quad (6a)$$

s.t. \(2\pi i \omega \text{diag}(\mathbf{L}, \mathbf{C}) - M(\mathbf{G})\mathbf{u}^i = P_1 \mathbf{f}^i, \quad 1 \leq i \leq p. \quad (6b)$$

It is convenient to set $p = m$ and choose the input basis vectors to be $(f^i)_j = \delta_{ij}$. The **desired transfer matrix** is then $T_d = \left[g^1, g^2, \ldots, g^m\right]$. We can then write the solution of (6b) using (4) and rewrite the optimization problem (6) in the following compact form:

$$\min_{(\mathbf{L}, \mathbf{C}, \mathbf{G}) \in \mathcal{A}} \tilde{J}(\mathbf{L}, \mathbf{C}, \mathbf{G}) := \frac{1}{2} \left\| T(\mathbf{L}, \mathbf{C}, \mathbf{G}) - T_d \right\|_F^2. \quad (7)$$
As written, the objective function \( J(u') \) in (6a) does not depend explicitly on \((L, C, G)\), only implicitly through the constraint (6b). We use the notation \( J(L, C, G) = J(u'(L, C, G)) \) to refer to the composition that explicitly depends on \((L, C, G)\).

We consider two different choices of boundary conditions:

(BC1) The resistive boundary \( \Theta \) consists of all nodes on the top, right, and bottom boundaries of the lattice. For each \( i \in \Theta \), we prescribe the locally impedance-matched conductance
\[
G_i = \sqrt{C_i/L_j},
\]
where \( j \in E \) is the edge incident on node \( i \) that is normal to the boundary. This impedance boundary condition can be viewed as a first-order discretization of the Silver-\M"uller outgoing boundary condition for Maxwell's equations, as described in [39].

(BC2) The resistive boundary \( \Theta \) consists only of \( \Upsilon \), i.e., the nodes on the right boundary of the lattice. For each \( i \in \Theta \), we set \( G_i \) according to (8), as before. Unlike the previous case, \( G_i = 0 \) along top/bottom boundaries.

Slightly abusing notation, we take \( \tilde{J}(L, C) \) to be the composition of \( J(L, C, G) \) with (8). We thus arrive at the following \( N \)-dimensional optimization problem:
\[
\min_{(L, C) \in \mathcal{A}} \tilde{J}(L, C) \tag{9}
\]
where \( \mathcal{A} \) is also modified to reflect (8) by letting \( G_j = 0 \) and \( C_j = \infty \) for all \( j \in \Theta \). Thus the only constraints in (9) are box constraints on the design variables \( L \) and \( C \).

Numerical tests show (9) is not convex, which implies that the solution to (9) is not guaranteed to be unique.

III. COMPUTATION OF THE GRADIENT AND HESSIAN

In this section, we compute the gradient and Hessian of \( \tilde{J}(\epsilon) \) in preparation for quasi-Newton and Newton numerical solutions of the optimization problem (6).

A. Computation of the Gradient via the Adjoint Method

Here we set \( s = (L, C) \) and \( A = 2\pi\alpha \text{ diag}(s) - M \). We introduce the dual variables \( v^i \in \mathbb{C}^p \) and the Lagrangian
\[
\mathcal{L}(u^i, v^i, s) = J(u^i) + \sum_{i=1}^{p} \Re(v^i, Au^i - P_{i}^f). \tag{10}
\]
The state equations (6b) are obtained by setting the derivative of (10) with respect to \( v^i \) equal to zero. The adjoint equations are obtained by setting the derivative of (10) with respect to the state variables \( u^i \) equal to zero:
\[
\frac{\partial \mathcal{L}}{\partial u^i} = \frac{\partial J}{\partial u^i} + \frac{1}{2} v^i A = 0. \tag{11}
\]

The decision equations are obtained by setting the derivative of (10) with respect to the design variables \( s \) equal to zero and recalling \( \frac{\partial J}{\partial s_k} = 0 \) for all \( k \).
\[
\frac{\partial \mathcal{L}}{\partial s_k} = \sum_{i=1}^{p} \Re\left(v^i, \frac{\partial A}{\partial s_k} u^i\right) = 0 \tag{13}
\]

To compute \( \partial A/\partial s_k \), we must compute
\[
\frac{\partial \text{ diag}(s)}{\partial s_k} = \delta_{ij} \delta_{ik} \quad \text{and} \quad \frac{\partial M}{\partial s_k} = \left[0 \quad 0 \begin{array}{c} \frac{\partial G}{\partial s_k} \end{array}\right].
\]

It is easy to show that \( \partial G_{ij}/\partial s_k = 0 \) unless the node \( k \) is a top, right, or bottom boundary node. There are three cases for the non-zero entries: non-corner top/bottom, non-corner right, and corners, each of which can be computed using (8).

The KKT equations consist of (6b), (11), and (13). A full space method involves the simultaneous solution of these three nonlinear equations. Alternatively, the reduced space method consists of taking \( \tilde{J}(s) = J(u'(s)) \). Then we have
\[
\frac{\partial \tilde{J}}{\partial s_k} = \Re\left(\sum_{i=1}^{p} \left(v^i, \frac{\partial A}{\partial s_k} u^i\right)\right), \tag{14}
\]

where \( v^i \) and \( u^i \) are solutions of (6b) and (11).

B. Direct Computation of the Gradient

Here we compute
\[
\frac{\partial \tilde{J}}{\partial s_k} = \frac{\partial J}{\partial s_k} + \sum_{i=1}^{p} \left( \frac{\partial J}{\partial u^i} \frac{\partial u^i}{\partial s_k} + \frac{\partial J}{\partial u^{i*}} \frac{\partial u^{i*}}{\partial s_k} \right)
\]
\[
= 2\Re\sum_{i=1}^{p} \left( \frac{\partial J}{\partial u^i} \right) \tag{15}
\]
where we have used
\[
\frac{\partial A}{\partial s_k} u^i + A \frac{\partial u^i}{\partial s_k} = 0, \tag{16}
\]
obtained from differentiating (6b). We now see that (14) and (15) are the same by (11). The advantage to computing \( v^i \) first and then computing the gradient via (14) is that only \( p \) adjoint solves are required (one for each input-output pair). Computing (15) literally (i.e., computing the expression in parentheses first and then computing the vector-matrix product) would require \( N \cdot p \) state solves [41].

C. Computation of the Hessian

Differentiating (14) enables us to write the Hessian
\[
\frac{\partial^2 \tilde{J}}{\partial s_j \partial s_k} = \Re\sum_{i=1}^{p} v^{i*} \left[ \frac{\partial A}{\partial s_j} \right] u^i + v^i \left[ \frac{\partial A}{\partial s_k} \right] u^{i*}. \tag{12}
\]
Differentiating the adjoint eq. (11) with respect to $s_j$, gives

$$\frac{\partial v^i}{\partial s_j} = -\left(2\frac{\partial}{\partial s_j} \frac{\partial \mathcal{J}}{\partial u^i} + v^i \left[ \frac{\partial A}{\partial s_j} \right] \right) A^{-1}$$

Combining the previous two equations with (16), and defining $h_{ij} = P_L A^{-1} \frac{\partial A}{\partial s_j} u^i$, we have the Hessian

$$\frac{\partial^2 \mathcal{J}}{\partial s_j \partial s_k} = \sum_{i=1}^{P} h^*_j h_{ki} + v^i \left[ \frac{\partial^2 A}{\partial s_j \partial s_k} - \frac{\partial A}{\partial s_j} \frac{\partial A}{\partial s_k} A^{-1} \frac{\partial A}{\partial s_k} A^{-1} \frac{\partial A}{\partial s_j} \right] u^i.$$  \hspace{1cm} (17)

IV. DESIGN VARIABLES

To reduce the size of the optimization problem (9), we introduce design variables, a reduced representation for $L$ and $C$. There are many natural choices for the design variables $r$. The following choices are labeled for future reference.

(D1) If $L$ and $C$ are symmetric in the sense that

$$L^h_{i,j} = L^h_{m+1-i,j}, \quad L^v_{i,j} = L^v_{m+1-i,j},$$

then the transfer matrix satisfies $T_{i,j} = T_{m+1-i,m+1-j}$. Thus, if the desired transfer matrix has this property, $r$ can be chosen to enforce this symmetry on $L$ and $C$. This reduces the dimension of the design variable space by a factor of approximately two.

(D2) The vectors $L_h$ and $L_v$ can be chosen as a discretization of a single continuous function $\mu(x)$ as in [39]. This imposes a compatibility condition on $L_h$ and $L_v$, reducing the dimension of the design space by approximately three. Specifically, we let $\mu$ be a $m+1 \times n+1$ matrix and set

$$L^h_{ij} = \frac{1}{2} (\mu_{ij} + \mu_{i+1,j}), \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \hspace{1cm} (18a)$$

$$L^v_{ij} = \frac{1}{2} (\mu_{ij} + \mu_{i,j+1}), \quad 2 \leq i \leq m, \quad 1 \leq j \leq n. \hspace{1cm} (18b)$$

The design variables then consist of $C$ and $\mu$.

(D3) Restricting to lattices with $L = 1$ reduces the dimension of the design space by a factor of three. This is analogous to considering media with constant permeability [39].

(D4) Combining the ideas in (D1) and (D3), we take $L = 1$ and force $C$ to have symmetry. This reduces the design variable space by a factor of six.

(D5) The vectors $L$ and $C$ can also be represented in terms of a truncated basis, such as the Fourier, wavelet, or block bases, but we do not pursue this here.

For a (BC1) lattice, energy leaks out of the top/bottom boundaries, so the total energy collected at the output is less than the input energy. Since we are primarily interested in the shape of the output $g(y)$, we include an extra design variable $\delta$ in the objective function (9), replacing $T_d$ by $\delta T_d$. For all design variable choices, we let $r_1 = \delta$.

Let $s = s(r)$ denote the dependence of $s$ on a set of design variables $r$. Then the gradient and Hessian can be computed

$$g \equiv \nabla_r \mathcal{J}(s(r)) = s_v \nabla_s \mathcal{J}$$

$$H \equiv \nabla_r \nabla_r \mathcal{J}(s(r)) = s_v \nabla_s \nabla_s \mathcal{J} s_v^T,$$

where $s_v$ denotes the Jacobian and $\nabla_s \mathcal{J}$ and $\nabla_s \nabla_s \mathcal{J}$ were computed in (14) and (17) respectively.

Once the design variables are chosen, the optimization problem (9) can be written

$$\min_{r \in \mathcal{A}_r} \mathcal{J}(r) := \frac{1}{2} \|T(r) - r T_d\|_F^2,$$

where $\mathcal{A}_r$ is an admissible set for the design variables $r$.

$$\mathcal{A}_r := \{ r : \frac{r}{p} \leq r_j \leq \frac{p}{r_j} \forall j \}.$$

V. COMPUTATIONAL RESULTS

In Sections V-A through V-D, we apply gradient-based optimization tools [42] to solve the lattice synthesis problem (20) for four desired transfer matrices. In Section V-A, we also compare the performance of several different optimization methods. In Sec. V-C we compare the two choices of boundary conditions given in Sec. II. In all other sections, we use (BC1). In Section V-E, we discuss the sensitivity of the transfer matrix of an inductor-capacitor lattice to small perturbations in $L$ or $C$. Finally, in Sections V-F and V-G, we study numerically the well-posedness of the synthesis problem.

A. Diagonal Transfer Matrix

In this section, we define the desired transfer matrix to be the diagonal matrix $T_d = \text{diag}(t)$. For a lattice with $m$ rows, let $j_c = (m+1)/2$ and $t_j = \exp(-2(j-j_c)^2)$. $j = 1, \ldots, m$. We set $\alpha = .08$ and choose (D1) design variables with lower and upper bounds 0.05 and 5.

We now solve the synthesis problem (20) for an $m \times m$ lattice for $m = 8 (N = 184)$ and $m = 16 (N = 752)$ using several different numerical methods. For each $m$ and numerical method used, in Fig. 3, we plot both iteration number and wall time vs. the objective function value. In what follows, we describe the methods compared in Fig. 2. All computations were done using Matlab 7.11 on a 2.4 GHz Intel Core 2 Duo desktop computer with 2GB of RAM. In each case, the convergence criteria was set using the Matlab options: `MaxIter = 2000`, `TolX = 10^{-14}`, and `TolFun = 10^{-13}`. In all examples here and below, the optimization method is initialized with constant design variables, $r$. We compare Matlab’s `fmincon` implementation of the following nonlinear constrained optimization algorithms:

**(SQ)** `sqp`: The sequential quadratic programming (SQP) approach is to approximate (20) by a quadratic minimization problem at each iteration. This quadratic form involves the Hessian of the objective function, which is approximated using the BFGS method [42, Ch. 18].

**Optimization tools:**

- **fmincon**: The active set method solves a sequence of unconstrained optimization problems. The optimization variables do not necessarily satisfy the bounds at each iteration.
(IP) interior-point: This line-search based quasi-Newton method uses the BFGS method to update the approximate Hessian at each iteration. The constraints are enforced using a logarithmic barrier function.

(TR) trust-region-reflective: We use this subspace trust-region method with large-scale = off.

From Fig. 2, we conclude that all tested methods are able to find solutions with low objective values. The other methods perform approximately the same in both iteration count and wall time. The interior point method (IP) performs best; however, the solution obtained tends to be less smooth than that obtained via the other methods. In what follows, we primarily use the (AS) method. In addition to the four methods described above, we also tried Newton’s method, but found that the cost of computing the Hessian (17) was prohibitively large for lattice sizes of interest.

Let us return to the design problem for the diagonal transfer matrix $T_d$. The optimal solution $(L^*, C^*)$ for $m = 16$ obtained using (SQ) is plotted in Fig. 3 and has objective value $J = 7.3 \times 10^{-5}$. The method terminated when the maximum number of iterations, MaxIter = 2000, was reached.

For this transfer function and all transfer functions considered in the subsequent sections, the design variable $r_1 = \delta$ attains the lower bound constraint of $L_2 = .6$. This indicates it is easier to synthesize energy-dissipative lattices.

**B. Waveguide Filter / Rank-One Projection**

In this section, we define

$$T_d = \begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{pmatrix},$$

the discrete analogue of a waveguide transfer function $f \rightarrow (\psi, f)\psi$, where $\psi$ is a desired state.

With $\alpha = .32$, we use (D1) design variables with lower and upper bounds given by 0.05 and 50. For a $24 \times 24$ lattice, we use the active set method (AS) to obtain the optimal solution $(L^*, C^*)$ plotted in Fig. 4 with objective value $J = 6 \times 10^{-6}$. The method terminated after 547 iterations because the predicted change in the objective function was less than TolFun = $10^{-13}$.

The optimal solution, plotted in Fig. 4, has horizontal inductors $L^h$ and capacitors $C$ which take large values in a strip from the center inputs to the center outputs. Outside of this strip, the $C$ matrix has periodic structure arranged to impede an incoming wave. The fact that we can recognize structure in the solution to an optimization problem in $\mathbb{R}^{1704}$ is remarkable, and suggests rigidity in the synthesis problem.

**C. A Low-Pass Filter / Smoothing Convolution**

In [39], we used separation of variables to obtain the exact solution for the continuous analogue of the forward problem (3) for a homogeneous lattice. We concluded that a homogenous lattice strongly damps oscillatory input, which suggests that this type of lattice is well-suited for performing low-pass filtering functions. We investigate this intuition here by constructing a circuit that behaves as a low-pass filter.

For an $8 \times 6$ lattice, we define the transfer matrix:

$$T_d = \frac{1}{44} \begin{pmatrix}
2 & 1 & 0 & 0 & 0 & 0 \\
4 & 2 & 1 & 0 & 0 & 0 \\
8 & 4 & 2 & 1 & 0 & 0 \\
4 & 8 & 4 & 2 & 1 & 0 \\
2 & 4 & 8 & 4 & 2 & 1 \\
0 & 1 & 2 & 4 & 8 & 4 \\
0 & 0 & 1 & 2 & 4 & 8 \\
\end{pmatrix}. \tag{21}$$

The matrix $T_d$ can be obtained by removing the first two and last two columns from an $8 \times 8$ Toeplitz matrix. We also remove the first and last two columns of the transfer matrix $T$ in (20). With $\alpha = 0.16$ and (D1) design variables with lower and upper bounds given by .05 and 50, we use the active set method (AS) for each of the boundary conditions given in Sec. II. For (BC1), the final objective function value is $J = 6.24 \times 10^{-7}$ and for (BC2), the final objective value is $J = 2.98 \times 10^{-5}$. In both cases, the method terminated because the predicted change in the objective function was less than TolFun = $10^{-13}$. In Fig. 5, we plot the optimal solution $(L^*, C^*)$ for both choices of boundary conditions.

**D. Power Combiner / Funnel**

Motivated by the power combiner introduced in [2], [3], we consider the transfer matrix that maps all inputs to the center output. The desired transfer matrix $T_d$ of size $m \times m$ (where $m = 2j + 1$ is odd) consists of a matrix where row $j + 1$ has a 1 in each column, and all other rows are identically zero.

We set $\alpha = 0.08$ and choose (D2) design variables. The upper and lower bounds were .05 and 20. In Fig. 6, we plot the optimal solution $(L^*, C^*)$ for the synthesis problem attained using the active set method (AS). The solution is plotted for $m \times m$ lattices where $m = 11, 21, 31$ with respective objective function values $2 \times 10^{-5}, 3 \times 10^{-5}$, and $3 \times 10^{-5}$.
Fig. 3. The \((L, C)\) matrices for the 16 \times 16 diagonal transfer lattice found in Section V-A with objective value \(J = 7.3 \times 10^{-5}\).

Fig. 4. The \((L, C)\) matrices for the 24 \times 24 waveguide in Section V-B with objective value \(J = 6 \times 10^{-6}\).

Fig. 5. The \((L, C)\) matrices for the low-pass filter in Section V-C for the 8 \times 6 lattice for boundary conditions as described by (BC1) in the top panel and (BC2) in the lower panel with resp. objective values \(J = 6.24 \times 10^{-7}\) and \(J = 2.98 \times 10^{-5}\).

In each case, the method terminated because the maximum number of iterations, \(\text{MaxIter} = 3000\), was reached.

E. Robustness / Sensitivity of Optimal Devices

In this section, we consider the sensitivity of optimal devices to small changes in \((L, C, G)\). We begin with a proposition that is proved in Appendix A.
Proposition 1. Let \( T_j = P_j A_j^{-1} P_j^* \), \( j = 1, 2 \) be the transfer matrices for two different circuits with capacitances, inductances, and conductances given by \((C_j, L_j, G_j)\), where \( A_j := 2\pi\alpha\, \text{diag}(L_j, C_j) - M(G_j) \).

Assume \( \rho := \| A_1^{-1}(A_2 - A_1) \|_2 < 1 \), and define \( \gamma = 1/\sigma_1(A_1) \) where \( \sigma_1(A_1) > 0 \) is the smallest singular value of \( A_1 \). Then

\[
\| T_1 - T_2 \|_F \leq \frac{mN\gamma^2}{1 - \rho} \left[ 2\pi\alpha \left( \| L_2 - L_1 \|_2 + \| C_2 - C_1 \|_2 \right) + \| G_2 - G_1 \|_2 \right].
\]

The upshot of this proposition is that if a circuit is perturbed by modifying \((L, C, G)\), then the change in the transfer matrix for the circuit is bounded by the size of the perturbation. However, the bounding constant could be large and increases with increasing circuit size.

We conduct a numerical experiment to further investigate this dependence for the low-pass filtering device introduced in Section V-C. Let \((L^*, C^*)\) denote the \(8 \times 6\) device with (BC1) boundary conditions plotted in Fig. 5(top panel) that minimizes \( J \) for the desired transfer matrix in (21) with objective value \( J(L^*, C^*) = 6.24 \times 10^{-7} \). We now evaluate \( J \) for a distribution of perturbations to \((L^*, C^*)\). Specifically, we consider multiplicative noise and evaluate \( J(L, u, C, v) \), where \( uv \) denotes entry-wise multiplication of the vectors \( u \) and \( v \), and \((u, v)\) have entries which are normally distributed with mean 1 and standard deviation 0.02. We interpret a structure \((L, u, C, v)\) to be a low-pass filtering device manufactured with 2% tolerance. In Fig. 7, we plot a histogram of the objective function value evaluated on a sample size of 100,000 drawn from this distribution. The 10th, 50th and 90th percentiles are \(1.8 \times 10^{-3}\), \(6.9 \times 10^{-3}\), and \(3.9 \times 10^{-2}\).

We might also consider the sensitivity of optimal devices to small changes in \( \alpha \). However, since (6) is invariant under the transformation in (5), perturbing \( \alpha \) is equivalent to choosing a multiplicative perturbation \((u, v)\) from a skewed distribution.

F. Known Lattice Recovery / Inverse Crime Study

In the preceding sections, our goal was to obtain useful circuits. Here and in the next section, we conduct numerical experiments to quantify the ill-posedness of the problem.
In this first numerical experiment, we commit a so-called “inverse crime.” We take \( p = m \) and generate a transfer matrix \( T_d \) by solving the forward problem for known values of \((L^0, C^0)\). We then set aside \((L^0, C^0)\) and solve the design problem for the transfer matrix \( T_d \), giving us a computed solution \((L^*, C^*)\).

This is referred to as an “inverse crime” since the model of the forward problem used to generate the transfer matrix is precisely the same model assumed in the solution of the design problem. For these transfer matrices, we happen to know that there is a point—namely \((L^0, C^0)\)—where the objective function is zero. We can then measure how well our algorithm does by comparing \( J(L^*, C^*) \) with zero. We can also characterize the solution space by checking how \( J(L^*, C^*) \) depends on the known solution \((L^0, C^0)\).

Let us describe how we generate a random matrix \( C^0 \). We fix integer parameters \( \nu > 0 \) and \( \sigma \) as well as real parameters \( \rho_{\text{min}} \) and \( \rho_{\text{max}} \). We choose two random vectors of Fourier sine coefficients \( k^x \) and \( k^y \), both of size \( \nu \times 1 \). The \( j \)-th entry \( k_{j}^{xy} \) is sampled from a \( U(0,1) \) distribution and then multiplied by \( j^{-\sigma} \). We sample \( P(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} k_{i}^{x} k_{j}^{y} \sin(ix)\sin(jy) \) to create an \( m \times n \) matrix \( C^0 \), which is then scaled and translated so its max/min values are, respectively, \( \rho_{\text{max}} \) and \( \rho_{\text{min}} \).

For \( L^h \) and \( L^v \), we follow (18) after generating an \( (m + 1) \times (n + 1) \) matrix \( \mu \) by sampling \( P(x,y) \). We scale and translate the matrices \( L^h \) and \( L^v \) so their max/min values are, respectively, \( \rho_{\text{max}} \) and \( \rho_{\text{min}} \). In all cases, sampling of \( P(x,y) \) is performed on a regular grid in the square \([0, 2\pi]^2\).

Using the above approach for generating random pairs \((L^0, C^0)\), we solved the design problem 750 times on an \( 8 \times 8 \) lattice. We used the active set method (AS) with TolX = \( 10^{-14} \), TolFun = \( 10^{-13} \), and (B2) design variables with lower and upper bounds of \( 0.05 \) and \( 50 \). For all 750 runs, the code terminated because the magnitude of the direction was less than \( 2 \times \text{TolFun} \).

We stepped \( \nu \) from 0 to 5 and \( \sigma \) from 1 to 5. We stepped \( \rho \) through 25 equispaced values in the interior of \((0,2)\), and set \( \rho_{\text{max}} = 1 + \rho/2 \), \( \rho_{\text{min}} = 1 - \rho/2 \).

In the left panel of Fig. 8, we plot the objective function value \( J(L^*, C^*) \) versus amplitude \( \rho \) for all 750 runs. The plot shows that the code performed very well across all runs, with the maximum value of \( J(L^*, C^*) \) less than \( 10^{-7} \). The plot also reflects a correlation coefficient of 0.82, which indicates a large amplitude of spatial oscillations in \((L^0, C^0)\), the poorer the quality of the local optimum reached.

In the right panel of Fig. 8, we plot the objective function value \( J(L^*, C^*) \) versus reconstruction error \( \| (L^*, C^*) - (L^0, C^0) \|_{\infty} \) for all 750 runs. The plot reflects that, as we move further away from the global minimum \((L^*, C^*)\), we are still able to achieve transfer matrices that are very close to what is desired. However, the correlation coefficient of 0.80 indicates a small degradation in the quality of the local optimum as a function of distance from a global optimum.

### G. Lattice Refinement and Coarsening

For a lattice with homogeneous \((L,C)\) the Nyquist principle states that \( \alpha \sqrt{LC} < \sqrt{2/\pi} \). In [21], we found that Kirchhoff’s laws (3) behave like their continuum limit if \( \alpha \sqrt{LC} < 1/(2\pi) \), which is roughly one-third of the Nyquist frequency. In [39] we showed that the continuum limit is precisely the system of equations for the \((H_1, H_2, E)\) polarized mode for Maxwell’s equations in a planar medium. Thus we expect that for \( \alpha \) sufficiently small, even if \((L,C)\) is inhomogeneous, one may increase the size of the lattice and rescale \((L,C)\) so that both problems are a discretization of the same continuum problem.

In this section, we use this principle to provide quantitative estimates on the ill-posedness of the synthesis problem. Throughout, we set \( L = 1 \) and \( \alpha = 1 \).

On a \( 40 \times 40 \) lattice, we set \( C_{ij} = 1 + \text{sech}^2(\gamma (i - 20.5)^2 + (j - 20.5)^2) \) for \( \gamma = 25/39^2 \). Using this \( C_{40} \), we solve (3) for the transfer matrix \( T_{40} \).

We then average 2 \times 2 subblocks of both \( T_{40} \) and \( C_{40} \) to obtain a transfer function \( T_{20} \) and capacitances \( C_{20} \) on a \( 20 \times \)
20 lattice. The vector \( C_{20} \) is divided by 4 based on the finite volume derivation in [39]. The synthesis problem with desired transfer function \( T_{20} \) is then initialized using \( C_{20} \) and solved using (D4) decision variables. We denote this solution \( C_{20} \) and note that the objective function value is \( \mathcal{J} = 3.7 \times 10^{-3} \).

We now refine \( C_{20} \) to a 40 \times 40 lattice by repeating 2 \times 2 blocks of \( C_{20} \). We denote these capacitances by \( C_{40} \). The synthesis problem with transfer function \( T_{40} \) is initialized using \( C_{40} \) and solved to obtain \( C_{40} \). Both surfaces are plotted in Fig. 9. The final value of the objective function is \( 9.5 \times 10^{-8} \) and \( \| C_{40} - C_{40} \|_{\mathcal{F}} = 11.1 \). Thus, \( C_{40} \) and \( C_{40} \) are far apart in the Frobenius norm but achieve almost the same transfer function. Although the problem is ill-posed and the solution obtained is different than \( C_{40} \), we emphasize that we view \( C_{40} \) as an excellent solution to the synthesis problem since it achieves a phenomenally low objective function value.

In inverse problems, one applies regularization methods to enforce a priori known information such as smoothness. Similarly, in the design problem considered here, where the “data” (i.e., desired transfer matrix) is known perfectly, one could apply regularization methods to force \( L \) and \( C \) to have desired properties. We do not pursue this direction here.

VI. CONCLUSION / DISCUSSION

We have formulated the two-dimensional transmission lattice synthesis problem as an optimization problem, the solution of which yields inductor-capacitor lattices that can be fabricated for custom/novel applications in analog signal processing and filtering. For several chosen transfer functions, we have demonstrated that gradient-based optimization methods can be used to obtain excellent solutions to the synthesis problem.

In other contexts, the ideas presented in this paper are familiar: one can engineer the permittivity \( \varepsilon \) and permeability \( \mu \) of a medium to control the propagation of EM waves [43], [44], and in quantum mechanics, one may engineer a potential to have desired scattering properties [45]. As the frequency of analog circuits marches into the THz range, it is increasingly important that the circuit model be related, both qualitatively and quantitatively, to Maxwell’s equations. Ultimately, if one is interested in designing a microwave frequency device, one performs a direct numerical simulation of Maxwell’s equations to confirm that the circuit model accurately predicts the device’s behavior. Based on our findings in [39], these connections can be made more precise. Kirchhoff’s laws for the 2-D LC lattice (1) can be viewed as a finite volume discretization of Maxwell’s equations for a planar, inhomogeneous medium. For large circuits with smoothly varying \( (C, L) \), this discretization is accurate, and one can interpret the present work as a discrete-then-optimize approach to solving the \( (\varepsilon, \mu) \) synthesis problem for Maxwell’s equations. This is the subject of forthcoming work.

APPENDIX A

PROOF OF THE PROPOSITION IN SECTION V-E

The matrix identity \( A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1} \) gives

\[
T_1 - T_2 = P_T A^{-1}_2 (A_2 - A_1) A^{-1}_2 P_T^T.
\]

Taking the Frobenius norm of both sides and using the sub-multiplicative property of the Frobenius norm we obtain

\[
\|T_1 - T_2\|_{\mathcal{F}} \leq \|P_T A^{-1}_1\|_{\mathcal{F}} \|A_2 - A_1\|_{\mathcal{F}} \|A^{-1}_2 P_T^T\|_{\mathcal{F}}. \tag{22}
\]

We treat the 3 pieces on the right hand side of (22) in turn. First note \( \|P_T\|_{\mathcal{F}} = \|T\|_{\mathcal{F}}^{1/2} = \sqrt{m} \) and \( \|P_T^T\|_{\mathcal{F}} = \|T\|_{\mathcal{F}}^{1/2} = \sqrt{m} \).

1. We compute

\[
\|P_T A^{-1}_1\|_{\mathcal{F}} \leq \|P_T\|_{\mathcal{F}} \|A^{-1}_1\|_{\mathcal{F}} \leq \sqrt{m} \sqrt{N} \|A^{-1}_1\|_2 = \sqrt{mN}.\gamma
\]

Here we used the relation \( \|A\|_{\mathcal{F}} \leq \sqrt{\gamma} \|A\|_2 \) where \( r \) is the rank of \( A \) and \( \|A^{-1}\|_2 = \sigma_N(A^{-1}) = 1/\sigma_1(A) = \gamma \).

2. We compute

\[
\|A_2 - A_1\|_F \leq 2\pi\alpha \| \text{diag}(L_2, C_2) - \text{diag}(L_1, C_1) \|_F
+ \| M(G_2) - M(G_1) \|_F
= 2\pi\alpha \left( \| L_2 - L_1 \|_2 + \| C_2 - C_1 \|_2 \right)
+ \| G_2 - G_1 \|_2.
\]

3. As above, we compute

\[
\|A^{-1}_2 P_T^T\|_{\mathcal{F}} \leq \|A^{-1}_2\|_{\mathcal{F}} \|P_T^T\|_{\mathcal{F}} \leq \sqrt{mN} \|A^{-1}_2\|_2.
\]

Our goal now is to estimate \( \|A^{-1}_2\|_2 \) in terms of \( \gamma \). We compute

\[
\|A^{-1}_2\|_2 = \| A_1^T \left( \text{Id} + A^{-1}_1(A_2 - A_1) \right)^{-1} \|_2
\]

Note that \( (\text{Id} + A^{-1}_1(A_2 - A_1))^{-1} \) exists by the assumption \( \rho < 1 \). Summing the Neumann series for this expression gives

\[
\| \left( \text{Id} + A^{-1}_1(A_2 - A_1) \right)^{-1} \|_2 \leq \sum_{j=0}^{\infty} \|A^{-1}_1(A_2 - A_1)\|_2^j
= \sum_{j=0}^{\infty} \rho^j = \frac{1}{1 - \rho}.
\]

Putting these 3 pieces together yields the desired result.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation (NSF) under Grants DMS09-13048 and DMS06-02235, EMSW21-RG: Numerical Mathematics for Scientific Computing. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. The authors would like to thank the NSF Institute for Pure and Applied Mathematics (IPAM) for its hospitality.

REFERENCES


