

Frequency Response and Gap Tuning for Nonlinear Electrical Oscillator Networks

Harish S. Bhat*, Garnet J. Vaz

Applied Mathematics Unit, University of California Merced, Merced, California, United States of America

Abstract

We study nonlinear electrical oscillator networks, the smallest example of which consists of a voltage-dependent capacitor, an inductor, and a resistor driven by a pure tone source. By allowing the network topology to be that of any connected graph, such circuits generalize spatially discrete nonlinear transmission lines/lattices that have proven useful in high-frequency analog devices. For such networks, we develop two algorithms to compute the steady-state response when a subset of nodes are driven at the same fixed frequency. The algorithms we devise are orders of magnitude more accurate and efficient than stepping towards the steady-state using a standard numerical integrator. We seek to enhance a given network's nonlinear behavior by altering the eigenvalues of the graph Laplacian, i.e., the resonances of the linearized system. We develop a Newton-type method that solves for the network inductances such that the graph Laplacian achieves a desired set of eigenvalues; this method enables one to move the eigenvalues while keeping the network topology fixed. Running numerical experiments using three different random graph models, we show that shrinking the gap between the graph Laplacian's first two eigenvalues dramatically improves a network's ability to (i) transfer energy to higher harmonics, and (ii) generate large-amplitude signals. Our results shed light on the relationship between a network's structure, encoded by the graph Laplacian, and its function, defined in this case by the presence of strongly nonlinear effects in the frequency response.

Citation: Bhat HS, Vaz GJ (2013) Frequency Response and Gap Tuning for Nonlinear Electrical Oscillator Networks. PLoS ONE 8(11): e78009. doi:10.1371/journal.pone.0078009

Editor: Jesus Gomez-Gardenes, Universidad de Zaragoza, Spain

Received: July 30, 2013; **Accepted:** September 10, 2013; **Published:** November 4, 2013

Copyright: © 2013 Bhat, Vaz. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: hbhat@ucmerced.edu

Introduction

Networks of nonlinear electrical oscillators have found recent application in several microwave frequency analog devices [1–6]. The fundamental unit in these networks is a nonlinear oscillator wired as in Figure 1; this oscillator consists of one inductor, one voltage-dependent capacitor, one source, and one sink (a resistor). While many nonlinear oscillatory circuits have been studied for their chaotic behavior, the particular oscillator in Figure 1 does not exhibit sensitive dependence on initial conditions in the regime of operation that we consider. Instead, assuming the source is of the form $A \cos(\omega t + \phi)$, the oscillator reaches a steady-state consisting of a sum of harmonics with fundamental frequency ω [7].

When networks of these oscillators have been studied, the network topology has either been a one-dimensional linear chain, in which case the circuit is called a nonlinear transmission line [8–13]—see Figure 2, or a two-dimensional rectangular lattice [14–18]—see Figure 3. Even if each individual block in the chain/lattice is weakly nonlinear, the overall circuit can exhibit strongly nonlinear behavior. It is this property that is exploited for microwave device applications, enabling low-frequency, low-power inputs to be transformed into high-frequency, high-power outputs.

The first objective of this work is to develop numerical algorithms to compute the frequency response of a nonlinear electrical network with topology given by an arbitrary connected graph. Here we are motivated by the successful application of

computational techniques in the design of the high-frequency analog devices referenced above. As we show, to compute steady-state solutions with comparable accuracy, both the perturbative and iterative algorithms developed in this paper require orders of magnitude less computational time than standard numerical integration. While the perturbative algorithm generalizes derivations given in prior work [7,18], the iterative algorithm has not been previously applied to nonlinear electrical networks. Both new algorithms show exponential convergence in the number of iterations, and for a test problem on a network with $N=400$ nodes, less than 20 iterations are required to achieve machine precision errors.

The second objective of this work is to relate structural properties of the network to the dynamics of the nonlinear oscillator system. The derivation of the perturbative algorithm indicates that nonlinearity in the electrical network manifests itself through energy transfer from the fundamental forcing frequency to higher harmonics. This helps us understand why properties such as amplitude boosting [7,18] and frequency upconversion [1], observed in nonlinear electrical networks with regular lattice topologies, can be expected when the topology is that of a random, disordered network. Additionally, we observe that an inductance-weighted graph Laplacian matrix features prominently in both algorithms for computing the steady-state solution. This graph Laplacian matrix encodes the structure of the network, and its eigenvalues are the squares of the resonant frequencies for the undamped, linear version of the circuit. Driving the damped,

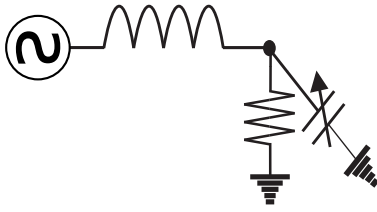


Figure 1. Schematic of a single nonlinear oscillator. This oscillator is the basic building block of the networks considered in this paper. The circuit contains one inductor, one voltage-dependent capacitor, one source, and one resistor. doi:10.1371/journal.pone.0078009.g001

linearized circuit at one of these resonances results in large amplitude outputs. It is reasonable to hypothesize that the locations of these resonances play a large role in the dynamics of the nonlinear network.

This motivates the following question: how do the eigenvalues of the graph Laplacian influence the nonlinear network's properties of frequency upconversion and amplitude boosting? While it is possible to alter the spectrum of the graph Laplacian by changing the node-edge relationships in the graph, we can also change its spectrum by keeping the topology fixed and manipulating the network's inductances. We formulate and solve the inverse problem of finding the inductances such that the graph Laplacian achieves a prescribed spectrum. The solution proceeds via a Newton-type algorithm that takes the desired spectrum as input and iteratively alters the inductances until a convergence criterion is met.

For three types of random graphs, we find that the Newton-type method effectively finds circuit inductances that close the gap between the first two eigenvalues of the graph Laplacian. We conduct a series of numerical experiments to examine the effect of closing this eigenvalue gap on a given circuit's ability (i) to transfer energy from the fundamental driving frequency to higher harmonics, and (ii) to generate high-amplitude output signals. The results indicate that the two metrics (i-ii) can be improved dramatically by closing the gap between the graph Laplacian's first two eigenvalues. Table 1 shows results we obtained for graphs with $N = 175$ nodes. Though this is a small portion of the results we describe later, this table already illustrates the effect of gap tuning on network performance. Note that each *pre* and *post* circuit have the same graph topology, differing only in their edge inductances.

Note that we have made available open-source Python implementations of all algorithms described in this work. The Python code, together with R code used for plotting, has been posted on a public repository. This enables the reader to reproduce all results in this paper. Instructions on how to download this code is given below.

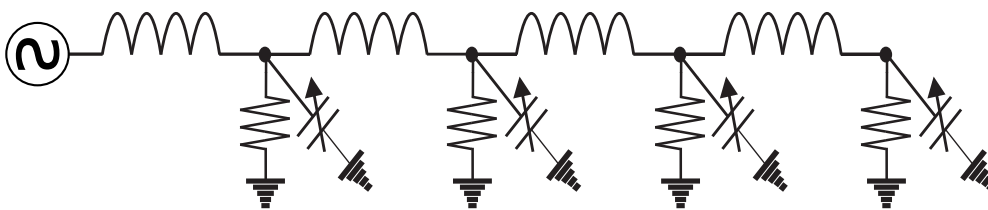


Figure 2. An example of a nonlinear transmission line. A nonlinear transmission line is a nonlinear electrical network on a one-dimensional linear graph. doi:10.1371/journal.pone.0078009.g002

Connections to Other Systems

We can make several connections between the problem studied in this paper and other problems of interest:

- **Random elastic networks.** Using a mechanical analogy between inductors/capacitors and masses/springs, the nonlinear electronic network can be transformed into a mathematically equivalent network of masses and anharmonic springs [19, Appendix I]. Such random elastic networks have been of recent interest as models of amorphous solids [20–22]. For such networks, quartic spring potential energies have been considered [23]. Nonlinear random elastic networks have also been used to model molecular machines; in this context, tuning the gap between the first two eigenvalues of the linearized system enables the construction of networks with properties similar to those of real proteins [24]. Despite this activity, algorithms for computing and manipulating the frequency response of nonlinear elastic networks have not been developed. Our work addresses this issue directly.
- **Nonlinear electromagnetic media.** The circuit we analyze, for particular values of the circuit parameters, arises naturally as a finite volume discretization of Maxwell's equations for TE/TM modes in a nonlinear medium [25,26]. The arbitrary connected graph topology of the circuit corresponds to a finite volume discretization on an arbitrary unstructured mesh. The algorithms developed here can be used to compute and optimize the frequency response of nonlinear electromagnetic media.
- **Coupled phase oscillator networks.** There has been intense interest in nonlinear phase oscillator networks, primarily due to the ability of such networks to model biophysical systems featuring synchronization. Though synchronization is not of primary interest in our system, we may still draw parallels. The effect of network topology on the properties of coupled phase oscillators has been studied extensively [27–30]. Manipulating eigenvalues of the Laplacian matrix enables one to enhance a network's synchronization properties [31]. More recently, several authors have developed algorithms for optimizing the synchronization of phase oscillator networks [32–37]. The questions considered in this subset of the coupled phase oscillator literature are related to the issues addressed in the present work.

Methods

Problem Formulation

Let $H(N, e)$ be a connected, simple graph with N nodes and e edges. Each edge corresponds to an inductor that physically connects two nodes. Each node corresponds to a capacitor and resistor, wired in parallel, that physically connect the node to a common ground. Let $f \leq N$ be the number of nodes that are

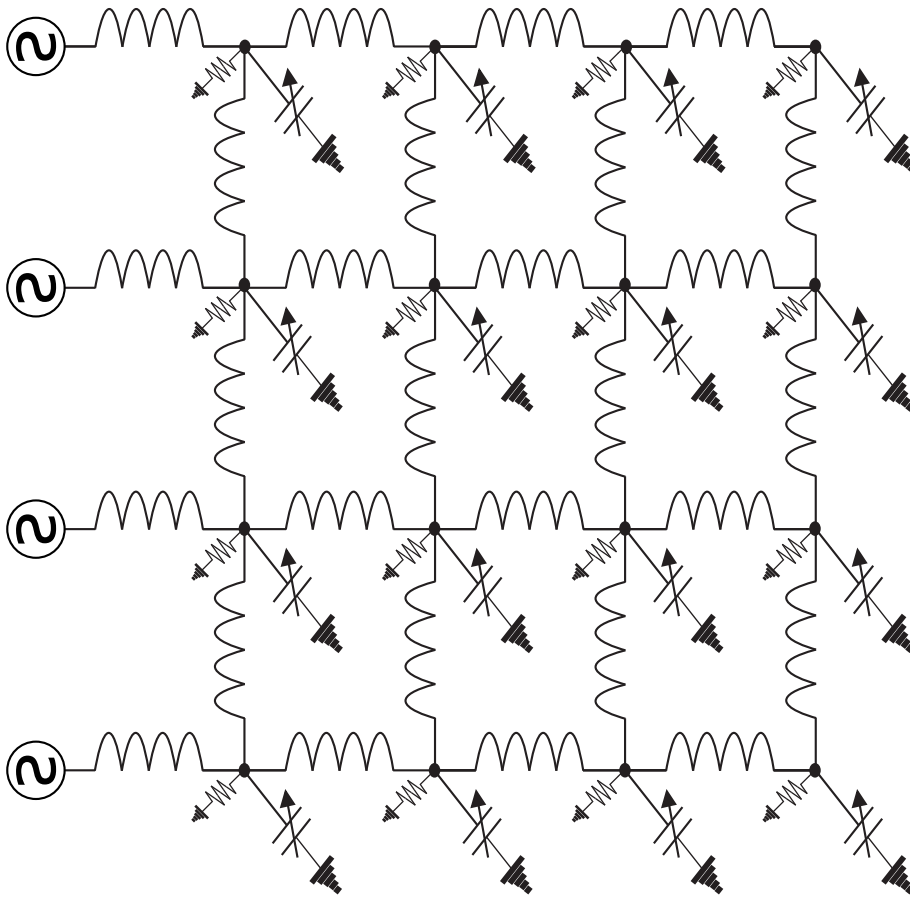


Figure 3. An example of a nonlinear lattice. A nonlinear lattice is a nonlinear electrical network on a two-dimensional rectangular grid graph. doi:10.1371/journal.pone.0078009.g003

driven by prescribed sources. Since the voltage at the prescribed source is known, we do not model it using a node. The connection between the source and the node that it drives is modeled by a half-edge, also known as a dangling edge since one end is connected to a driven node and the other end does not connect to any node. We let $H(N,e,f)$ denote the graph together with the f half-edges.

The capacitance and conductance (inverse resistance) at node j are C_j and G_j , respectively. We let $V_j(t)$ denote the voltage from node j to ground at time t . The inductance of edge k is L_k , while the current through edge k at time t is $I_k(t)$. The exact dimensions for each component of H , along with the currents and voltages, are tabulated in Table 2.

In order to write down Kirchhoff's laws, we must choose an orientation of the edges. The orientation of an edge records the direction of positive current flow through the edge. If we solve the problem with opposite orientations, the only difference we will notice is that the currents will pick up a factor of -1 . Consequently, the orientation we choose does not affect the solution in any material way. In what follows, we will choose a random orientation of the edges.

In Figure 4 we show an example graph corresponding to $H(6,9,2)$. The edges are oriented randomly. The inputs are connected at nodes 1 and 6 through two inductors. These input nodes correspond to half-edges in H . On the right we view node 3 in detail. Each of the two edges connected to this node correspond

Table 1. Portion of results for graphs with $N = 175$ nodes.

	% of energy in higher harmonics		Maximum magnitude voltage	
	Pre	Post	Pre	Post
Barabási-Albert (BA)	0.410	4.063	0.01548	0.31684
Watts-Strogatz (WS)	1.006	8.701	0.03399	0.51157
Erdős-Rényi (ER)	0.033	7.534	0.002956	0.78902

Simulation results for three different types of random graphs with $N = 175$ nodes, averaged over 100 runs. "Pre" and "Post" stand for before and after circuit inductances are changed to reduce the gap between the graph Laplacian's first two eigenvalues. Note that *pre* and *post* circuits have the same graph topology and differ only in their inductances.

doi:10.1371/journal.pone.0078009.t001

Table 2. Summary of the notation used in the paper.

Notation	Significance	Size
C	Capacitance at node	$N \times 1$
L	Inductance of edge	$(e+f) \times 1$
G	Conductance at node	$N \times 1$
V	Voltage at node	$N \times 1$
I	Current through edge	$(e+f) \times 1$
W	Input forcing	$N \times 1$
B	Signed incidence matrix	$N \times (e+f)$

doi:10.1371/journal.pone.0078009.t002

to an inductor. A capacitor with capacitance C_3 and a resistor with conductance G_3 connect node 3 to ground.

To arrange Kirchhoff's laws compactly, we use the $N \times (e+f)$ incidence matrix of $H(N, e, f)$, denoted by B . Let j be an edge connecting the nodes i' and i . If j is oriented such that positive current starts at node i' and flows to node i , we write $j = (i', i)$. If j is a half-edge attached to node i , we write $j = (\emptyset, i)$, leaving the first slot empty and orienting the half-edge so it always points toward the forced node. The entries of the incidence matrix B are

$$B_{i,j} = \begin{cases} 1 & \text{if } j = (i', i) \text{ for some node } i' \text{ or } i' = \emptyset \\ -1 & \text{if } j = (i, i') \text{ for some node } i' \\ 0 & \text{otherwise.} \end{cases}$$

This paper will only consider single frequency time-harmonic forcing of the form $ae^{i\omega t} + \bar{a}e^{-i\omega t}$ where $a \in \mathbb{C}^N$. Let P be an $N \times (e+f)$ matrix with entries $P_{i,j} = 1$ if node i is connected to an input edge j and 0 otherwise. Using the projection matrix P we define the forcing

$$W(t) = P^T(ae^{i\omega t} + \bar{a}e^{-i\omega t}). \tag{1}$$

Using the notation summarized in Table 2, Kirchhoff's laws for the nonlinear circuit on the graph $H(N, e, f)$ can now be written compactly as

$$L \frac{dI}{dt} = -B^T V + W \tag{2}$$

$$C \frac{dV}{dt} = BI - GV. \tag{3}$$

Here $L(dI/dt)$, $C(dV/dt)$, and GV are examples of component-wise multiplication of vectors. For $a, b \in \mathbb{C}^m$, we define $c = ab \in \mathbb{C}^m$ by $c_j = a_j b_j$ for $1 \leq j \leq m$. Note that in this case, we can also write $c = \text{diag}(a)b$. Here $\text{diag}(a)$ is the $m \times m$ matrix that contains the vector a along its diagonal ($[\text{diag}(a)]_{ii} = a_i$) and is zero elsewhere.

The formulation (2–3) generalizes previous formulations [25,38] in which the capacitors were constant and the systems considered were linear.

By differentiating (3) and inserting it into (2), we obtain a second-order system for the voltages:

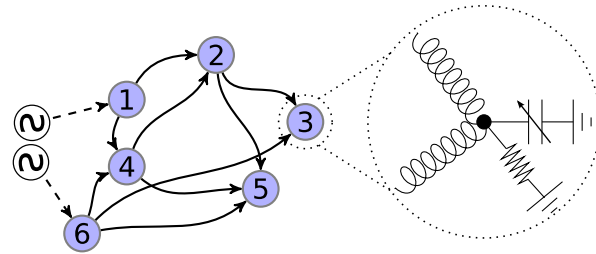


Figure 4. An example of a nonlinear electrical network. In the graph on the left, the numbered circles are nodes, the solid arrows are edges, and the dashed arrows are half-edges. Orientation of the arrows indicates the direction of positive current flow. Each node corresponds to a voltage-dependent capacitor to ground, wired in parallel with a resistor to ground, as depicted in the zoomed-in schematic for node 3. Each edge corresponds to an inductor. Each half-edge connects one prescribed voltage source to one given node. In this paper, all methods that are developed are valid for connected graphs with at least one half-edge. Note that the circuits in Figures 1–3 can all be represented using this graph formalism.

doi:10.1371/journal.pone.0078009.g004

$$\frac{d}{dt} \left[C \frac{dV}{dt} \right] + G \frac{dV}{dt} + \Delta V = V_{in}. \tag{4}$$

Here

$$V_{in}(t) = B[\text{diag}(L)]^{-1} W(t) \tag{5}$$

$$\Delta = B[\text{diag}(L)]^{-1} B^T. \tag{6}$$

Note that Δ is the weighted Laplacian for the network with edge weights given by reciprocal inductance.

We assume that the capacitance at node i depends on the voltage at node i :

$$C_i(V_i) = C^0(1 - \epsilon V_i), \tag{7}$$

where $C^0 \in \mathbb{R}$ is a constant. Note that this choice of capacitance function means that (4) features a quadratic nonlinearity.

We can then formulate the *frequency response problem* for the nonlinear electrical network: given the amplitude vector a and frequency ω for the forcing function (1), determine the steady-state solution $V(t)$ of (4).

Perturbative Algorithm

We first solve the frequency response problem using a perturbative expansion in powers of ϵ . We use dots to denote differentiation with respect to time. Substituting the capacitance function (7) in (4) and rearranging, we obtain

$$C^0 \ddot{V} + G \dot{V} + \Delta V = V_{in} + \frac{\epsilon C^0}{2} \frac{d^2}{dt^2} [V^2]. \tag{8}$$

We expand

$$V(t) = V_0(t) + \varepsilon V_1(t) + \varepsilon^2 V_2(t) + \dots \tag{9}$$

Inserting (9) into (8), we obtain equations for each order of ε . At zeroth order, we obtain

$$C^0 \dot{V}_0 + G \dot{V}_0 + \Delta V_0 = V_{in}. \tag{10}$$

For $k \geq 1$, the k -th order equation is

$$C^0 \dot{V}_k + G \dot{V}_k + \Delta V_k = C^0 \frac{d}{dt} \left[\sum_{m=0}^{k-1} V_{k-1-m} \dot{V}_m \right]. \tag{11}$$

We now solve (10–11). Let us introduce the Fourier transform in time,

$$\hat{\psi}(\alpha) = \int_{-\infty}^{\infty} e^{-i\alpha t} \psi(t) dt, \tag{12}$$

with inverse Fourier transform

$$\psi(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\alpha t} \hat{\psi}(\alpha) d\alpha. \tag{13}$$

Note that with these definitions,

$$\hat{\dot{\psi}}(\alpha) = i\alpha \hat{\psi}(\alpha).$$

This implies that the Fourier transforms of both sides of (10–11) can be summarized by writing

$$\mathcal{L}(\alpha) \hat{V}_k(\alpha) = \begin{cases} \hat{V}_{in} & k=0 \\ -(\alpha^2/2) C^0 \hat{\phi}_k(\alpha) & k \geq 1, \end{cases} \tag{14}$$

where $\mathcal{L}(\alpha)$ is the linear operator

$$\mathcal{L}(\alpha) = -\alpha^2 \text{diag}(C^0) + i\alpha \text{diag}(G) + \Delta, \tag{15}$$

and

$$\hat{\phi}_k(t) = \sum_{m=0}^{k-1} V_{k-1-m} V_m. \tag{16}$$

By (5) and (1), we see that

$$\hat{V}_{in} = 2\pi B[\text{diag}(L)]^{-1} P^T (a\delta(\alpha - \omega) + \bar{a}\delta(\alpha + \omega)), \tag{17}$$

where δ is the Dirac delta. Then the $k=0$ branch of (14) yields

$$\hat{V}_0(\alpha) = 2\pi [a_{0,1}(\alpha)\delta(\alpha - \omega) + a_{0,-1}(\alpha)\delta(\alpha + \omega)] \tag{18a}$$

$$a_{0,1}(\alpha) = [\mathcal{L}(\alpha)]^{-1} B[\text{diag}(L)]^{-1} P^T a \tag{18b}$$

$$a_{0,-1}(\alpha) = [\mathcal{L}(\alpha)]^{-1} B[\text{diag}(L)]^{-1} P^T \bar{a}. \tag{18c}$$

Using the inverse Fourier transform, we have

$$\begin{aligned} V_0(t) &= a_{0,1}(\omega)e^{i\omega t} + a_{0,-1}(-\omega)e^{-i\omega t} \\ &= a_{0,1}(\omega)e^{i\omega t} + \text{c.c.}, \end{aligned}$$

where ‘‘c.c.’’ stands for the complex conjugate of the previous terms. Here we have used the property that $\mathcal{L}(-\alpha) = \overline{\mathcal{L}(\alpha)}$.

Once we have computed $V_0(t)$, we can insert it into (16) to compute $\hat{\phi}_1(t)$. We will find that $\hat{\phi}_1(t)$ is a linear combination of $e^{-2i\omega t}$, $e^{0i\omega t}$, and $e^{2i\omega t}$. Using this fact in the $k=1$ branch of (14), we can solve for $\hat{V}_1(\alpha)$ and then apply the inverse Fourier transform to compute $V_1(t)$. We will find that $V_1(t)$ contains the same modes as $\hat{\phi}_1(t)$.

The above shows how we get the perturbative solution algorithm started. Now let us move to the more general case where we seek $V_k(t)$ for any $k \geq 1$. Assume that we have already computed $V_j(t)$ for $0 \leq j \leq k-1$, and that the solution takes the following form:

$$V_{2m}(t) = \sum_{\ell=0}^m a_{2m,2\ell+1} e^{(2\ell+1)i\omega t} + \text{c.c.} \tag{19a}$$

$$V_{2m+1}(t) = \sum_{\ell=0}^{m+1} a_{2m+1,2\ell} e^{2\ell i\omega t} + \text{c.c.} \tag{19b}$$

In words, V_{2m} contains odd modes $1, 3, \dots, 2m+1$, and V_{2m+1} contains even modes $0, 2, \dots, 2m+2$. Here we assume that $0 \leq 2m < 2m+1 \leq k-1$, and that the $a_{i,j} \in \mathbb{C}^N$ coefficients are known.

In order to solve for $V_k(t)$, we use the $k \geq 1$ branch of (14), which requires us to compute (16). We have two cases, when k is odd and when k is even. In both cases, it is a simple (if tedious) algebraic exercise to show that $\hat{\phi}_k(t)$ yields:

- when k is odd, a sum of even Fourier modes $-(k+1), \dots, -2, 0, 2, \dots, (k+1)$, and
- when k is even, a sum of odd Fourier modes $-(k+1), \dots, -3, -1, 1, 3, \dots, (k+1)$.

In both cases, it is clear that using (14) to solve for $\hat{V}_k(\alpha)$ results in a sum of Dirac delta’s. Applying the inverse Fourier transform yields $V_k(t)$, which will be a sum of Fourier modes. One can check that $V_k(t)$ will have precisely the form (19a) or (19b) depending on whether k is even or odd, respectively.

The algorithm is then clear. Starting with (19), we apply component-wise multiplication to particular pairs of the $a_{i,j}$ vectors in order to compute the coefficients of the Fourier modes of $\hat{\phi}_k(t)$ defined in (16). Next, we combine the step of solving for $\hat{V}_k(\alpha)$ using the $k \geq 1$ branch of (14) together with the step of computing

the inverse Fourier transform. After component-wise multiplication of the Fourier coefficients of ϕ_k by $-(\alpha^2/2)C^0$, we multiply each coefficient on the left by $[\mathcal{L}(\alpha)]^{-1}$ with α set to match the frequency of the corresponding Fourier mode. Dividing these coefficients by 2π yields the Fourier coefficients of $V_k(t)$, as desired.

While we have presented the algorithm in an intuitive way, the statements made above can be made rigorous, and a convergence theory for the perturbative expansion (9) can be established. This is the subject of ongoing work.

There are a few brief remarks to make about the algorithm presented above:

- As described above, we consider only those networks that contain resistance at all nodes, i.e., $G_i > 0$ for all nodes i . Such an assumption is not only physically realistic; it also guarantees that for all $\alpha \in \mathbb{R}$, the matrix $\mathcal{L}(\alpha)$ is invertible. The invertibility for the $\alpha=0$ case is a consequence of Corollary 1 proved below.
- In this work, we are interested in the weakly nonlinear regime where $\varepsilon\|a\|$ is sufficiently small such that the perturbative method converges. As the nondimensional constant $\varepsilon\|a\|$ is increased beyond the breakdown point of the perturbative method, direct numerical solutions of the equations of motion reveal subharmonic oscillations, and eventually, chaotic oscillations.
- The fact that the Fourier transform yields the steady-state solution has been explained in our earlier work [7]. By fixing an arbitrary set of initial conditions and using the Laplace transform to derive the full solution, one can show that after the decay of transients, the part of the solution that remains is precisely what we obtain using the Fourier transform. This also explains why it was not necessary for us to specify initial conditions for (4) in our derivation above—the initial conditions only influence the decaying transient part of the solution.

Iterative Algorithm

The perturbative method developed above shows us that the solution $V(t)$ is a sum of harmonics where the fundamental frequency is given by the input frequency ω . This implies that the steady-state solution $V(t)$ is periodic with period $T=2\pi/\omega$. This observation leads us to ask whether it is possible to directly solve for the Fourier coefficients of $V(t)$ without first expanding in powers of ε . In this section, we develop a fixed point iteration scheme that accomplishes this task.

First, we integrate both sides of (8) from $t=0$ to $t=T$ to derive

$$\Delta \int_0^T V(t) dt = 0. \tag{20}$$

We show below that as long as the network contains at least one half-edge, Δ is invertible. Hence (20) implies

$$\int_0^T V(t) dt = 0. \tag{21}$$

This means there is no zero/DC mode present in $V(t)$, motivating the Fourier series expansion

$$V(t) = \sum_{k=1}^{\infty} \alpha_k e^{ik\omega t} + \text{c.c.} \tag{22}$$

In order to compute the solution, we truncate at $k=M$, leading to an approximation $\underline{V} \approx V$:

$$\underline{V}(t) = \sum_{k=1}^M \alpha_k e^{ik\omega t} + \text{c.c.} \tag{23}$$

Using orthogonality we derive

$$\alpha_k = \frac{1}{T} \int_0^T e^{-ik\omega t} \underline{V}(t) dt.$$

Using the T -periodicity of V and integration by parts, we have

$$\frac{1}{T} \int_0^T e^{-ik\omega t} \dot{\underline{V}}(t) dt = ik\omega \alpha_k.$$

To simplify notation, we combine (1) and (5) and write $V_{\text{in}} = w e^{i\omega t} + \text{c.c.}$ where

$$w = B[\text{diag}(L)]^{-1} P^T a. \tag{24}$$

Now let $\delta_{m,n}$ denote the Kronecker delta function which equals 1 if $m=n$, and 0 otherwise. We multiply both sides of (8) by $e^{-ik\omega t}$, integrate from $t=0$ to $t=T$, and finally divide by T to obtain

$$\mathcal{L}(k\omega)\alpha_k = w\delta_{k,1} + \bar{w}\delta_{k,-1} + \frac{\varepsilon C^0 (ik\omega)^2}{2} \widetilde{V}^2_k, \tag{25}$$

where \mathcal{L} was defined in (15) and

$$\widetilde{V}^2_k = \frac{1}{T} \int_0^T e^{-ik\omega t} \underline{V}^2 dt. \tag{26}$$

Because the form of the nonlinearity is simple, we can insert (23) into (26) and derive

$$\widetilde{V}^2_k = \sum_{\ell=-M}^M \alpha_\ell \alpha_{k-\ell}, \tag{27}$$

with the understanding that $\alpha_0 = 0$, $\alpha_{-j} = \bar{\alpha}_j$ for $j > 0$, and $\alpha_j = 0$ for $|j| > M$. We insert (27) into (25) and obtain

$$\mathcal{L}(k\omega)\alpha_k = w\delta_{k,1} + \bar{w}\delta_{k,-1} - \frac{\varepsilon C^0 k^2 \omega^2}{2} \sum_{\ell=-M}^M \alpha_\ell \alpha_{k-\ell}.$$

We convert this into an iterative scheme in a natural way. Let $\alpha_k^{(j)}$ denote the j -th iterate, and assume that α_k terms appearing on

the left-hand side are at iteration $j + 1$, while those appearing on the right-hand side are at iteration j . Let $\mathbf{A}^{(j)}$ denote the $N \times M$ complex matrix whose k -th column is $\alpha_k^{(j)}$. Then the scheme is

$$\mathbf{A}^{(j+1)} = F^M(\mathbf{A}^{(j)}) \tag{28}$$

where the k -th column of the matrix $F^M(\mathbf{A}^{(j)})$ is

$$F_k^M(\mathbf{A}^{(j)}) = [\mathcal{L}(k\omega)]^{-1} \left(w\delta_{k,1} - \frac{\varepsilon C^0 k^2 \omega^2}{2} \sum_{\ell=-M}^M \alpha_\ell^{(j)} \alpha_{k-\ell}^{(j)} \right). \tag{29}$$

Here we assume $1 \leq k \leq M$, which is also why we have deleted the second Kronecker delta from the right-hand side.

Starting at $\mathbf{A}^{(0)}$, we iterate forward using (28), stopping the computation when $\|F(\mathbf{A}^{(j)}) - \mathbf{A}^{(j)}\|$ is below a specified tolerance. Note that in our implementation of F , we precompute and store the LU factorization for the M matrices $\{\mathcal{L}(k\omega)\}_{k=1}^M$, since this part of the computation of the right-hand side of (29) does not change from one iteration to the next.

Again, we have derived the algorithm but have not proven its convergence. Instead, we will demonstrate empirically that the algorithm converges using several numerical tests.

Inverse Problem

In this section, we consider the inverse problem of finding a set of inductances such that Δ , the Laplacian defined by (6), achieves a desired spectrum. Before describing an algorithm to solve this inverse problem, we review basic spectral properties of Δ .

Lemma 1. *Assume all inductances are positive. Then Δ as defined in (6) is symmetric positive semidefinite, and all its eigenvalues must be nonnegative.*

Proof. Let $[\text{diag}(L)]^{-1/2}$ be the diagonal matrix whose (i,i) -th element on the diagonal is $L_i^{-1/2}$, for $1 \leq i \leq e$. Since $L_i > 0$, the matrix $\Delta^{1/2} = B[\text{diag}(L)]^{-1/2}$ is real. Then $\Delta = \Delta^{1/2} (\Delta^{1/2})^T$, and

$$\text{for any } v \in \mathbb{R}^N, \text{ we have } v^T \Delta v = \left[(\Delta^{1/2})^T v \right]^T (\Delta^{1/2})^T v \geq 0.$$

Let $\{\lambda_i\}_{i=1}^N$ denote the spectrum of Δ , with eigenvalues arranged in nondecreasing order: $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. The above argument shows that $\lambda_1 \geq 0$. We can be more precise about this: if there are no half-edges, then $\lambda_1 = 0$, while the presence of at least one half-edge causes $\lambda_1 > 0$.

Lemma 2. *Let $\mathcal{H} = H(N, e)$ be a connected graph with N nodes, e edges, and zero half-edges. For a particular orientation of the graph, let B denote the signed incidence matrix. Then $\text{rank}(B) = N - 1$.*

Proof. Let r be any integer from 1 to $N - 1$. Consider any subset S of r vertices of the graph. Take the sum of the rows of the incidence matrix corresponding to the elements of S . This sum cannot be zero; if it were, there would be no path connecting S to the complement S^c and the resulting graph would not be connected. Hence the sum of these rows must contain a nonzero entry. As the same would be true if we considered linear combinations of the rows corresponding to S , we conclude that any subset of at most $N - 1$ rows must be linearly independent. At the same time, if we take the sum of all the rows we get a zero row, because each column contains precisely one $+1$ and one -1 .

Lemma 3. *Let $\mathcal{H}' = H(N, e, f)$ be a connected graph with N nodes, e edges, and $f > 0$ half-edges. For a particular orientation of the graph, let B' denote the signed incidence matrix. Then $\text{rank}(B') = N$.*

Proof. Without loss of generality, we can assume that the $N \times (e+f)$ incidence matrix B' is organized such that the first e columns correspond to full edges, while columns $e+1, \dots, e+f$ correspond to half-edges. Now choose any j such that $1 \leq j \leq f$, and examine column $e+j$ of B' . Let k be the unique row in which this column contains ± 1 . Since row k of B is the only row that contains an entry in column $e+j$, row k is linearly independent from the other $N - 1$ rows of B . By Lemma 2, the submatrix of B consisting of all rows other than row k has rank $N - 1$. Including row k increases the rank by one, yielding a rank N matrix.

Lemma 4. *For a connected graph $\mathcal{H}' = H(N, e, f)$ with N nodes, e edges, and $f > 0$ half-edges, let Δ be the edge-weighted graph Laplacian defined in (6). Assume all inductances are positive. Then $\text{rank}(\Delta) = N$.*

Proof. The $(e+f) \times (e+f)$ diagonal matrix $[\text{diag}(L)]^{-1}$ has rank $e+f > N$. Let B be the signed incidence matrix for a particular orientation of \mathcal{H}' . By Lemma 3, $\text{rank}(B) = N$, implying $\text{rank}(B[\text{diag}(L)]^{-1/2}) = N$, which implies $\text{rank}(\Delta) = \text{rank}[(B[\text{diag}(L)]^{-1/2})(B[\text{diag}(L)]^{-1/2})^T] = N$.

Corollary 1. *Let \mathcal{H}' , Δ and the inductances satisfy the hypotheses of Lemma 4. Then Δ is symmetric positive definite and all eigenvalues of Δ are positive, i.e., $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$.*

Proof. Combine Lemmas 1 and 4.

We now describe an algorithm that quantifies how we must change the vector of inductances L in order to make Δ have a desired set of eigenvalues. In what follows, we assume we work with a system that satisfies the hypotheses of Corollary 1.

For $n \leq N$, let $\lambda^* = (\lambda_1^*, \dots, \lambda_n^*)^T$ denote a vector of desired eigenvalues satisfying

$$0 < \lambda_1^* \leq \lambda_2^* \leq \dots \leq \lambda_n^*.$$

We treat the vector of inductances L as a variable, and let $\lambda(L)$ denote the sorted vector of eigenvalues of the graph Laplacian Δ defined in (6). Since Δ is symmetric, it possesses an orthonormal basis of eigenvectors. We assume that $v_j(L)$ is the normalized eigenvector corresponding to $\lambda_j(L)$.

Now let $\mathcal{F} : \mathbb{R}^{e+f} \rightarrow \mathbb{R}^n$ be the function

$$\mathcal{F}(L) = \begin{bmatrix} \lambda_1(L) \\ \lambda_2(L) \\ \vdots \\ \lambda_n(L) \end{bmatrix} - \begin{bmatrix} \lambda_1^* \\ \lambda_2^* \\ \vdots \\ \lambda_n^* \end{bmatrix}. \tag{30}$$

We now apply a version of Newton's method to find a zero of this function. To use Newton's method we will need to compute

Table 3. Eigenvalue gaps for random graphs.

	$N = 25$	$N = 75$	$N = 125$	$N = 175$
Barabási-Albert (BA)	0.6408	0.3561	0.3155	0.2850
Watts-Strogatz (WS)	1.4180	1.3255	1.2970	1.2758
Erdős-Rényi (ER)	1.5469	8.6936	17.7061	26.5297

For each of three types of random graphs, we vary the number of nodes N and record the first eigenvalue gap $\lambda_2 - \lambda_1$. The displayed results have been averaged over 200 realizations.

doi:10.1371/journal.pone.0078009.t003

the Jacobian $J(\mathcal{F}(L))$. Let primes denote differentiation with respect to L_k . To form the Jacobian we need to find

$$\lambda'_j := \frac{\partial}{\partial L_k} \lambda_j(L).$$

We proceed by implicit differentiation, starting from the eigenvector equation

$$\Delta v_j(L) = B[\text{diag}(L)]^{-1} B^T v_j(L) = \lambda_j(L) v_j(L).$$

Differentiating both sides with respect to L_k , and omitting the dependence on L , we obtain

$$B([\text{diag}(L)]^{-1})' B^T v_j + B[\text{diag}(L)]^{-1} B^T v'_j = \lambda'_j v_j + \lambda_j v'_j. \quad (31)$$

Since Δ is symmetric,

$$v_j^T B[\text{diag}(L)]^{-1} B^T = (B[\text{diag}(L)]^{-1} B^T v_j)^T = (\lambda_j v_j)^T = v_j^T \lambda_j. \quad (32)$$

Multiplying (34) on the left by v_j^T and using (35) together with $(v_j)^T v_j = 1$, we obtain

$$v_j^T B([\text{diag}(L)]^{-1})' B^T v_j = \lambda'_j, \quad (33)$$

where

$$([\text{diag}(L)]^{-1})' := \frac{\partial}{\partial L_k} ([\text{diag}(L)]^{-1}) = \left[0, 0, \dots, -\frac{1}{L_k^2}, \dots, 0 \right]^T.$$

Using λ'_j we can compute the entries of the Jacobian matrix and the corresponding Newton's method with pseudoinverse becomes

$$L^{(i+1)} = L^{(i)} - [J(\mathcal{F}(L))]^\dagger \mathcal{F}(L) \quad (34)$$

where \dagger denotes the Moore-Penrose pseudoinverse.

Using (34) as shown might produce inductances such that the ratio of the largest to smallest inductance is too large. In order to avoid these large variations, we constrain $\epsilon^{-1} \leq L_i \leq \epsilon$. We incorporate these constraints using an active set approach, replacing \mathcal{F} by the function $\mathcal{G}^i : \mathbb{R}^{e+f+m} \rightarrow \mathbb{R}^{n+m}$, where i denotes the iteration number and m denotes the number of constraints violated by $L^{(i)}$. Let \mathcal{Q}^\pm denote the functions

$$\mathcal{Q}^+(x) = \begin{cases} \frac{1}{2}x^2 & x > 0 \\ 0 & x \leq 0, \end{cases} \quad \text{and} \quad \mathcal{Q}^-(x) = \begin{cases} 0 & x \geq 0 \\ \frac{1}{2}x^2 & x < 0. \end{cases} \quad (35)$$

For every constraint p violated from below, we set $\mathcal{G}_p^i(L) = \mathcal{Q}^-(L_p - \epsilon^{-1})$. For every constraint q violated from above, we set $\mathcal{G}_q^i(L) = \mathcal{Q}^+(L_q - \epsilon)$. Since the \mathcal{Q}^\pm functions are continuously differentiable, it is easy to compute the Jacobian

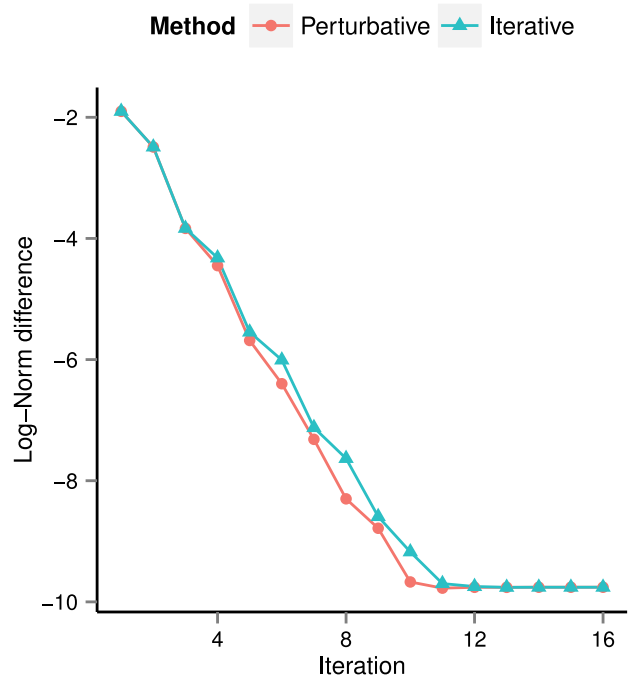


Figure 5. Error between perturbative/iterative solutions and reference solution. The reference solution has been computed via numerical time integration. We plot the log of the error as a function of the number of iterations. As shown in Figures 6 and 7 together with Tables 4 and 5, the perturbative/iterative solutions are more accurate than the reference solution. This explains why, in the above plot, the perturbative and iterative solutions do not converge to the reference solution.

doi:10.1371/journal.pone.0078009.g005

$J(\mathcal{G}^i(L))$ and then apply the algorithm

$$L^{(i+1)} = L^{(i)} - [J(\mathcal{G}^i(L))]^\dagger (\mathcal{G}^i(L)). \quad (36)$$

Algorithm (36) can be used to alter all the eigenvalues of the system if $n=N$ and $\lambda^* \in \mathbb{R}^N$. Alternatively, one can set $n=2$ and only request the two smallest eigenvalues to be changed to λ_1^* and λ_2^* , respectively.

In the next section we show that altering the lowest eigenvalue λ_1 is enough to cause higher energy transfer to the higher modes. To show, we will use (39) to change λ_1 to some desired value, keeping λ_2 constant. We note that since we do not constrain $\lambda_3, \dots, \lambda_N$, they can change as a result of altering L , but $\lambda_2 \leq \lambda_j$ for $j \in \{3, \dots, N\}$ will be maintained.

Table 4. Comparison of the three solutions using the fixed point error metric (40).

Scheme	$\max_{1 \leq k \leq M} \ \alpha_k - F_k^M(\mathbf{A})\ _\infty$
Numerical	2.035×10^{-11}
Perturbative	3.4321×10^{-16}
Iterative	2.7144×10^{-16}

doi:10.1371/journal.pone.0078009.t004

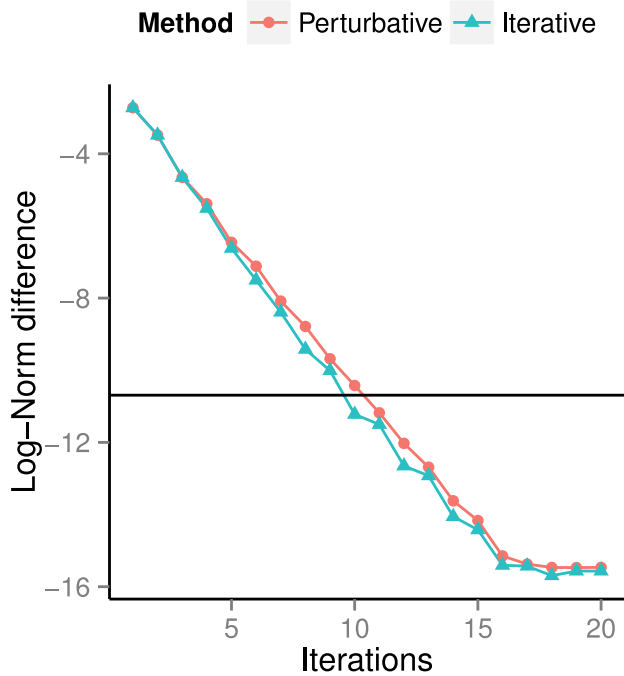


Figure 6. Log of the fixed point error (40) of the perturbative/iterative solutions after i iterations. Up to iteration 16, both curves are close to linear with slopes of -2.1407 (perturbative) and -2.2326 (iterative), indicating exponential convergence of both methods. Note that only 10 iterations are required to reach error values corresponding to that of the numerical integrator’s solution.
doi:10.1371/journal.pone.0078009.g006

For all applications of this inverse problem algorithm described in the next section, we use (36) with the initial conditions $L^{(0)} = [1, 1, \dots, 1]^T$ and the constraint violation parameter $\epsilon = 10^3$.

Gap Tuning: Methodology

How does the steady-state voltage in the nonlinear circuit change as a function of the gap between the first two eigenvalues of the graph Laplacian Δ ? In this section, we address this question by combining the perturbative/iterative algorithms with the inverse problem algorithm. We describe numerical experiments designed to test the effect of closing the graph Laplacian’s first eigenvalue gap on the circuit’s ability to (a) transfer more energy to higher harmonics, and (b) generate higher-amplitude output signals.

We conduct our numerical experiments on three types of random graphs, all generated using the NetworkX package [39]:

- Barabási-Albert (BA), a preferential attachment model with one parameter, m , the number of edges to draw between each new node and existing nodes [40]. We set $m=3$ in our experiments.
- Watts-Strogatz (WS), a small world model with two parameters, k , the number of nearest neighbor nodes to which each node is initially connected, and p , the probability of rewiring each edge [41]. In our experiments, we set $k=5$ and $p=0.3$.
- Erdős-Rényi (ER), a classical model in which edges are drawn independently with uniform probability p [42]. In our experiments, we set $p=0.25$.

When we produce realizations of any of these graphs, we accept only those graphs that are connected. Suppose we have used one

of these three models to generate a connected, random graph with N nodes. To make this a concrete circuit problem, we set $C_i^0 = 1$ for all nodes i , and $L_j = 1$ for all edges j . We fix the nonlinearity parameter $\epsilon = 0.5$. We select $N/10$ nodes uniformly at random, and attach half-edges to these nodes with inductance $L_j = 1$. For each node i , we set the conductance $G_i = 0.15$ for the BA and WS graphs, and $G_i = 0.5$ for the ER graphs. This selection will be explained in more detail below.

With these parameters set, we have enough information to compute the graph Laplacian Δ defined by (6). As we did before, let $\lambda_1, \dots, \lambda_N$ denote the eigenvalues of Δ sorted in increasing order. We set the forcing frequency $\omega = \sqrt{\lambda_2}$. Since this value is a resonant frequency of the linear, undamped system we expect it lies close to a resonance for the nonlinear, damped system. The type of forcing we consider is $A \sin \omega t$, a special case of (1) with $a = A/(2i)$.

With this setup, we use both the perturbative method and the iterative method to compute the steady-state solution $V(t)$. For the perturbative method, we solve up to order 9, and for the iterative method, we solve using 20 modes. This means that the iterative scheme captures ten modes— 11ω through 20ω —that are not captured by the perturbative scheme. We compare the two solutions as a check for whether the number of modes we have considered is sufficient. In all tests, we find that there is no significant difference between the solutions, implying that the first 10 harmonics— ω through 10ω —are sufficient to resolve the solution.

Since $\epsilon = 0.5$, the capacitance model (7) is valid only for $V_i < 2$. For all computed solutions, we check that the maximum voltage across all nodes in one cycle satisfies this constraint.

One quantity of interest in our simulations is the amount of energy in the higher harmonics. Let Ψ be an $N \times M$ complex matrix such that the j -th column of Ψ contains the Fourier coefficients of the $+j\omega$ mode over all N nodes. Here j goes from 1 to M , the maximum mode to which the solution is computed. We then define

$$\kappa_{\text{pre}} = \frac{1}{N} \sum_{n=1}^N \frac{\|(\Psi_{n2}, \Psi_{n3}, \dots, \Psi_{nM})^T\|_2}{\|(\Psi_{n1}, \Psi_{n2}, \dots, \Psi_{nM})^T\|_2}, \tag{37}$$

the fraction of energy in modes $+2\omega$ and higher, averaged over all nodes. We also compute

$$V_{\text{pre}} = \max_{1 \leq i \leq N} \max_{0 \leq t \leq T} |V_i(t)|, \tag{38}$$

the maximum magnitude voltage produced anywhere in the circuit during one period. For both κ and V , the subscript “pre” denotes that these quantities have been computed before we change L to manipulate the eigenvalues of Δ .

Having computed κ_{pre} , we now study how this fraction changes when we reduce the gap between the first two eigenvalues of Δ . For a fixed $\delta \in (0, 1)$, we set $\lambda_1^* = \delta \lambda_2$ and $\lambda_2^* = \lambda_2$, and then apply the inverse problem algorithm.

Using (36), we solve for a vector of inductances L^* such that the first two eigenvalues of Δ are given by λ_1^* and λ_2^* . When we iterate forward using (36), if we find that $\|\mathcal{G}^j(L)\|_2 \geq 10^{-12}$ after 200 iterations, we generate a new random graph and restart the experiment.

We recompute the graph Laplacian Δ using the new vector inductances L^* , and again apply the perturbative and iterative algorithms to solve for the steady-state solution $V(t)$. Using this

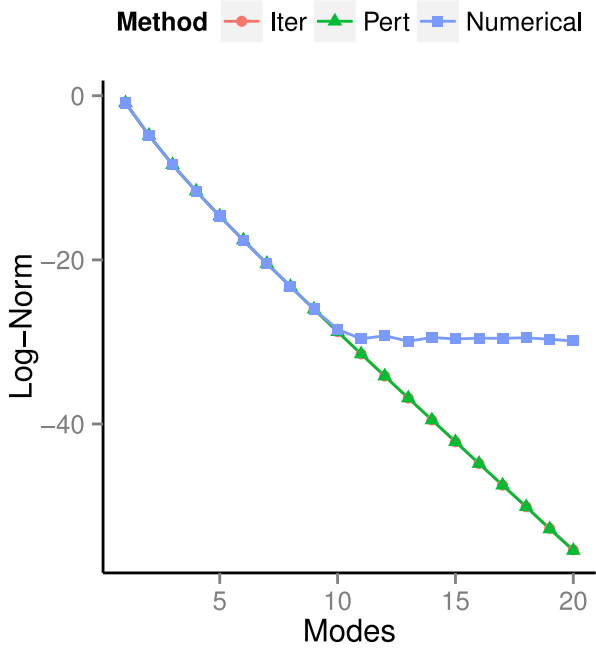


Figure 7. Decay of Fourier coefficients. We plot $\log \|x_k\|$ versus k to illustrate the decay of Fourier coefficients for the three methods. The iterative and perturbative curves coincide and are nearly linear with slope -2.8004 ; the exponential decay of these Fourier coefficients is consistent with theory. The time integrator’s Fourier coefficients do not decay after mode 10, violating the theoretical decay rate. doi:10.1371/journal.pone.0078009.g007

solution, we compute the energy in the higher harmonics using the right-hand side of (37), now labeling this average fraction as κ_{post} . We also compute the right-hand side of (38) and label this quantity as V_{post} .

Let us now describe how we choose the particular values of the conductance G_i and the eigenvalue fraction δ . In Table 3, we tabulate $\lambda_2 - \lambda_1$, the gap between the second and first eigenvalue for each of the three types of random graphs described above. The eigenvalue gaps we present are averaged over 200 simulations for each of four graph sizes: $N \in \{25, 75, 125, 175\}$.

We observe that the eigenvalue gaps for the BA and WS graphs do not change appreciably as a function of N , while for ER graphs, the gaps grow quickly as a function of N . Our choice of δ is guided by these results. For BA and WS graphs, we choose $\delta \in [0.5, 0.99]$. For ER graphs, we choose $\delta \in [0.25, 0.75]$.

When we solve for the steady-state voltages on these three types of graphs, we also notice a difference. For ER graphs, the maximum voltage grows quickly as a function of N , while for BA and WS graphs, the same phenomenon does not occur. To counteract the large growth of maximum voltages for large graph sizes, we set the conductance G_i to the larger value of 0.5 for ER graphs, causing more energy to dissipate through resistors. For BA and WS graphs, we set G_i to 0.15.

Results and Discussion

Comparison of Steady-State Algorithms

In this section, we compare steady-state solutions computed by numerical integration against the solutions computed using the perturbative and iterative methods derived earlier.

For the tests described in this section, the domain is a 20×20 square lattice with $N = 400$ nodes. Nodes along the left and

bottom boundaries of the lattice are driven by input forcing. The input provided is $0.03 \sin(\omega t)$ with $\omega = 1$. For the capacitance model (7), we set $C^0 = 1$ and $\epsilon = 0.5$. For each edge j , we set $L_j = 1$. The conductance G_i is set to 0.01 at all points except for the top-right corner, where it is set to 1.0.

To compare the results of the perturbative and iterative methods against the numerical integrator, we will need to obtain the steady-state solution using the numerical integrator. To do this, we start at $t = 0$ and numerically integrate the first-order system (2–3) forward in time for 1500 cycles. The ODE solver uses the Dormand-Prince (*dopri5*) method with relative and absolute tolerances equal to 10^{-10} and 10^{-12} , respectively. For the parameters given above, this number of cycles is sufficient so that, from one cycle to the next, the change in the solution is on the order of the relative tolerance of the numerical integrator. Hence we take the solution over the last cycle to be the steady-state solution.

As a preliminary check, we directly compare the three steady-state solutions. We treat the solution obtained from numerical time integration as a reference solution $z^{\text{ref}}(t)$. Let $z^{(i)}(t)$ denote either the perturbative or iterative solution after i iterations—for the perturbative method, the iteration count is defined as the largest mode number present in the solution. Let T be the period of the steady-state solution, and for an integer $\tau > 0$, consider the equispaced discretization of the interval $[0, T]$ given by $\{t_k = kT/\tau\}_{k=1}^{\tau}$. For each iteration i , we evaluate both the perturbative/iterative and reference solution on this equispaced grid with $\tau = 64$ points, and we compute the error

$$E(i) = \left[\frac{1}{\tau} \sum_{j=1}^N \sum_{k=1}^{\tau} \left(z_j^{(i)}(t_k) - z_j^{\text{ref}}(t_k) \right)^2 \right]^{1/2}. \quad (39)$$

In Figure 5 we have plotted $\log_{10} E(i)$ as a function of the iteration i . While both methods initially tend towards the reference solution, we see from Figure 5 that the error does not drop below 10^{-9} . In the following subsections, we provide evidence that the reference solution is less accurate than the solutions computed using the perturbative/iterative methods. This explains why the error in Figure 5 does not go to zero, i.e., why the perturbative/iterative methods will not converge to the solution produced by time integration.

Our first tests concern the Fourier coefficients of the computed solutions. In what follows, we use \mathbf{A} to denote the vector of Fourier series coefficients associated with a steady-state solution computed using any of the three methods discussed above.

Fixed Point Error. Suppose that $V(t)$ is an exact T -periodic steady-state solution of (4). If we were to expand this solution in a Fourier series as in (22), the resulting (infinite) coefficient vector \mathbf{A} would satisfy $\mathbf{A} = F_k^\omega(\mathbf{A})$ for all k , with F as in (29).

Table 5. Comparison of the three solutions’ preservation of the energy balance (44).

Scheme	$\left \int_0^T I^T W dt - \int_0^T V^T G V dt \right $
Numerical	5.7841×10^{-12}
Perturbative	2.1684×10^{-18}
Iterative	1.2576×10^{-18}

doi:10.1371/journal.pone.0078009.t005

Table 6. Timing results for three frequency response algorithms.

Scheme	Time I (to achieve comparable error)	Time II (to achieve $O(10^{-16})$ error)
Numerical	483.2126 s	N/A
Perturbative	1.71024 s	7.29115 s
Iterative	0.68374 s	0.89012 s

For the numerical method, Time I records the time required to integrate forward by 1500 cycles and obtain a solution with fixed point error $\approx 2 \times 10^{-11}$ as in Table 4. For the perturbative/iterative methods, Time I records the time required to compute 10 iterations, resulting in a fixed point error comparable to that of the numerical method—see the crossing of the curves with the black horizontal line in Figure 6. For the perturbative/iterative methods, we also record under Time II the time required to achieve the $O(10^{-16})$ errors as in Table 4. All times are averaged over 10 runs. doi:10.1371/journal.pone.0078009.t006

In both the perturbative and iterative methods, what we seek is a finite-mode truncation of the exact solution. For the iterative method we fix $M = 20$ so that the highest mode has frequency 20ω . In the perturbative method we solve, we solve up to order 19, which implies that the highest mode in the solution has frequency 20ω .

Combining the ideas of the last two paragraphs, it is natural to use

$$\mathcal{E}_M(\mathbf{A}) = \max_{1 \leq k \leq M} \|\alpha_k - F_k^M(\mathbf{A})\|_\infty \quad (40)$$

as an error metric for the M -mode truncation of the exact solution. In Table 4, we record (40) for solutions computed using the perturbative, iterative, and numerical integration methods. Note that the iterative and perturbative methods directly provide us with the Fourier coefficients necessary for this calculation. We compute the Fourier coefficients of the numerical integrator’s solution using the FFT. Table 4 shows that the perturbative and

iterative solutions are about five orders of magnitude closer to being fixed points of F_k^M than the solution obtained from numerical integration.

For the perturbative and iterative methods, let us examine how the fixed point error (40) decreases as a function of iteration count. In Figure 6, we plot $\log \mathcal{E}_M(\mathbf{A}^{(i)})$ versus the iteration number i . Here $\mathbf{A}^{(i)}$ is the vector of Fourier coefficients for the solution computed after only i iterations. The plot shows that, for both the perturbative and iterative methods, approximately 10 iterations are required to match the fixed point error of the solution computed using time integration. The error of this latter solution, taken from Table 4, is represented on Figure 6 by a horizontal black line.

Figure 6 also shows that the perturbative and iterative methods converge exponentially in the number of iterations. From iteration 1 until iteration 16, fitting lines of best fit to the perturbative and iterative error curves results in slopes of -2.1407 and -2.2326 , respectively. For both methods, this can be approximated by

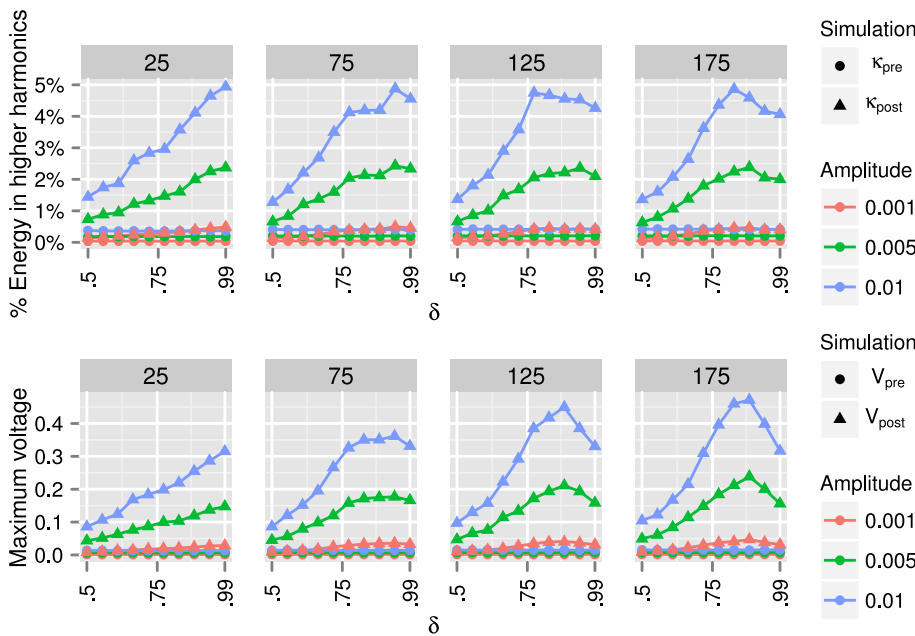


Figure 8. Barabási-Albert random graph results. From left to right, we present results for Barabási-Albert random graphs with $N = 25, 75, 125,$ and 175 nodes. For each graph, we use algorithm (36) to modify the inductances L such that the ratio of the smallest to the second smallest eigenvalue is δ . We use *pre* and *post* to denote, respectively, the graphs before and after algorithm (36) is applied. By shrinking the gap between the first two eigenvalues, the energy transferred to higher harmonics (37) can be increased from approximately $\kappa_{pre} \approx 0.5\%$ to $\kappa_{post} \approx 5\%$ (for all graph sizes), and the maximum voltage (38) can be increased from $V_{pre} < 0.05$ volts to $V_{post} \in [0.3, 0.5]$ volts (depending on the graph size). We also note that for larger graphs, choosing $\delta = 1$ (i.e., no gap between the first two eigenvalues) does not yield optimal behavior. doi:10.1371/journal.pone.0078009.g008

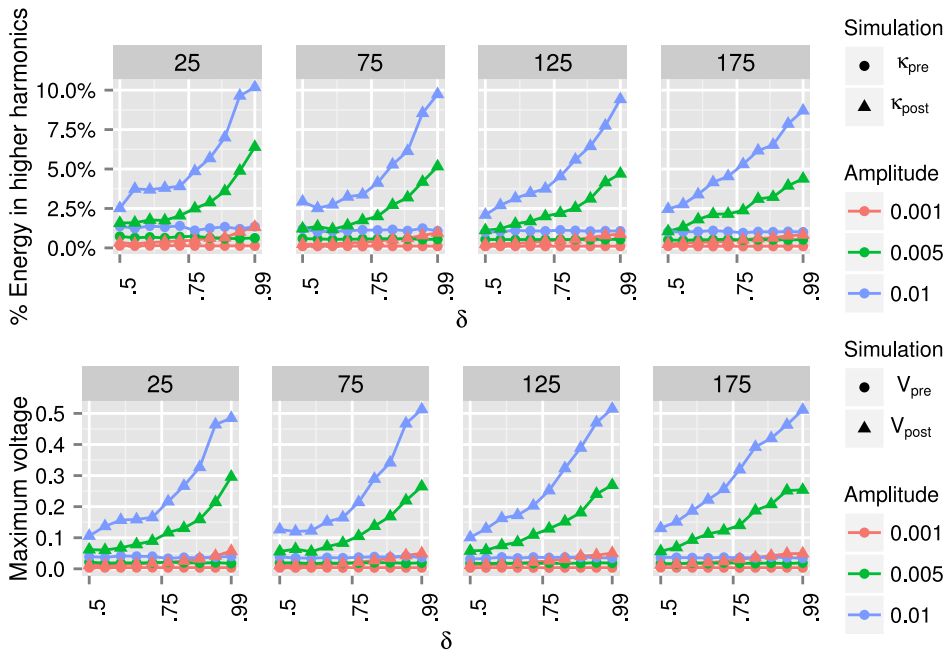


Figure 9. Watts-Strogatz random graph results. From left to right, we present results for Watts-Strogatz random graphs with $N=25, 75, 125,$ and 175 nodes. For each graph, we use algorithm (36) to modify the inductances L such that the ratio of the smallest to the second smallest eigenvalue is δ . We use *pre* and *post* to denote, respectively, the graphs before and after algorithm (36) is applied. By shrinking the gap between the first two eigenvalues, the energy transferred to higher harmonics (37) can be increased from $\kappa_{pre} \approx 1\%$ to $\kappa_{post} \geq 8.75\%$ (for all graph sizes), and the maximum voltage (38) can be increased from $V_{pre} \approx 0.05$ volts to $V_{post} \approx 0.5$ volts (for all graph sizes). The values of κ_{post} for Watts-Strogatz graphs are about twice as large as the values of κ_{post} for Barabási-Albert graphs in Figure 8. doi:10.1371/journal.pone.0078009.g009

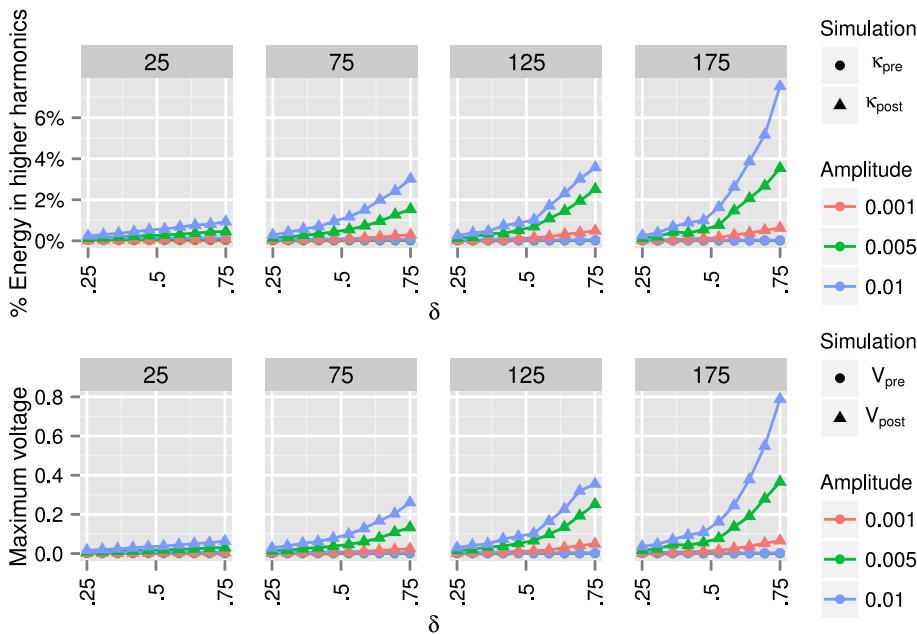


Figure 10. Erdős-Rényi random graph results. From left to right, we present results for Erdős-Rényi random graphs with $N=25, 75, 125,$ and 175 nodes. For each graph, we use algorithm (36) to modify the inductances L such that the ratio of the smallest to the second smallest eigenvalue is δ . We use *pre* and *post* to denote, respectively, the graphs before and after algorithm (36) is applied. By shrinking the gap between the first two eigenvalues, the energy transferred to higher harmonics (37) can be increased to $\kappa_{post} \in [1, 8]\%$ (depending on the graph size), and the maximum voltage (38) can be increased to $V_{post} \in [0.1, 0.8]$ (depending on the graph size). The results for Erdős-Rényi graphs are much more strongly dependent on the number of nodes N than the results shown in Figures 8 or 9. Note that the peak voltages for the $N=175$ graphs with forcing amplitude 0.01 are the largest voltages for any graphs considered in this paper. We can increase the peak voltages for smaller graphs by choosing a smaller value of the conductance than $G_i=0.5$ (for all nodes i), the value used to compute the results in this figure. doi:10.1371/journal.pone.0078009.g010

writing $\mathcal{E}_M(\mathbf{A}^{(i)}) \sim e^{-2i}$. After 16 iterations, the error has approached machine epsilon, and both curves level off before reaching the final values shown in Table 4.

Decay Rate. Suppose we write the first-order system (2–3) in the form $\dot{\mathbf{z}} = \mathbf{F}(\mathbf{z}, t)$, with $\mathbf{z} = (I, V)^T$. Then on the open set $\mathcal{D} = \{I \in \mathbb{R}^{e+f}, V \in \mathbb{R}^N, t \in \mathbb{R} \mid |V_i| < \epsilon^{-1}, 1 \leq i \leq N\}$, the vector field $\mathbf{F} : \mathcal{D} \rightarrow \mathbb{R}^{e+f+N}$ is C^∞ , i.e., \mathbf{F} is j times continuously differentiable for any integer $j \geq 1$. By the standard existence/uniqueness theorem for ordinary differential equations, it follows that wherever the solution $\mathbf{z}(t) = (I(t), V(t))^T$ exists, it must also be C^∞ in t .

The above observation enables us to test the decay of the Fourier coefficients of all three solutions. For if the steady-state solution $V(t)$ is C^∞ in t , then the Fourier series coefficients of V must satisfy the following decay property:

$$\text{For all } \ell \geq 1, \text{ there exists } C_\ell \text{ such that } \|\alpha_k\| \leq C_\ell k^{-\ell}. \quad (41)$$

To examine the decay of the Fourier coefficients for the three computed solutions, we plot $\log \|\alpha_k\|$ versus k in Figure 7. For the perturbative and iterative solutions, the curves on the plot coincide and are nearly linear with slope -2.8004 . This implies that $\|\alpha_k\| \sim e^{-2.8k}$, which is sufficient to satisfy the theoretical decay rate given by (41).

The Fourier coefficients obtained from the numerical integrator’s solution, on the other hand, do not decay at all beyond mode 10. This violates the theoretical decay rate (41) even for $\ell = 1$.

Energy Conservation. Next we test the energy conservation properties of the three computed solutions. We proceed to derive an energy balance equation. Because our capacitors are voltage-dependent, the charge Q and voltage V are related via $dQ = C(V)dV$, which implies

$$\frac{dQ}{dt} = C(V) \frac{dV}{dt}.$$

Using this in (3) together with (2), we obtain

$$I^T L \frac{dI}{dt} + V^T \frac{dQ}{dt} = I^T W - V^T G V. \quad (42)$$

Let $\mathbf{M}(t)$ be the total energy stored in the magnetic fields of all inductors at time t . Then

$$\frac{d\mathbf{M}}{dt} := \frac{d}{dt} \left[\frac{1}{2} \sum_{i=1}^{e+f} L_i I_i^2 \right] = \sum_{i=1}^{e+f} I_i L_i \frac{dI_i}{dt} = I^T L \frac{dI}{dt},$$

the first term on the left-hand side of (42). Let $\mathbf{E}(t)$ be the total energy stored in the electric fields of all capacitors at time t . Then

$$\frac{d\mathbf{E}}{dt} := \frac{d}{dt} \left[\sum_{i=1}^N \int_0^{Q_i(t)} V_i(q) dq \right] = \sum_{i=1}^N V_i \frac{dQ_i}{dt} = V^T \frac{dQ}{dt},$$

the second term on the left hand side of (42). Hence (42) reads:

$$\frac{d}{dt} (\mathbf{M}(t) + \mathbf{E}(t)) = I^T W - V^T G V. \quad (43)$$

If the system has reached a T -periodic steady state, then $I(t)$, $V(t)$, $\mathbf{M}(t)$, and $\mathbf{E}(t)$ will all be T -periodic. Therefore, integrating (43) in t from $t=0$ to $t=T$, we find that the left-hand side vanishes. The remaining terms yield the following energy balance equation:

$$\int_0^T I^T W dt = \int_0^T V^T G V dt. \quad (44)$$

The left-hand side is the energy pumped into the system over one cycle, while the right-hand side denotes the energy dissipated through resistors, again over one cycle.

Table 5 shows the absolute difference between the left- and right-hand sides of (44), computed for each of the three methods. We find that for the perturbative and iterative methods’ energy balance errors are below machine epsilon. The numerical integrator yields an error approximately five orders of magnitude larger than that of the two other methods.

Computational Time. The results presented thus far indicate that whether we measure error using the fixed point error (40) or the violation of the energy balance (44), the solution obtained via numerical integration has errors that are approximately five orders of magnitude larger than that of the perturbative/iterative methods. The actual values of the errors committed by the numerical integrator in Tables 4 and 5, as well as the final error values for the curves in Figure 5, are close to the numerical integration relative and absolute tolerances of 10^{-10} and 10^{-12} , respectively. We hypothesize that, if computational time were not an issue, we could run the numerical integrator with smaller tolerances and obtain steady-state solutions that more closely match, in the same error metrics described above, the perturbative and iterative solutions.

As we now proceed to show, computational time is a major issue for the time integration method. In Table 6, we record the time required to compute steady-state solutions using the three methods. We see from the Time I column that to achieve the error of $\approx 2 \times 10^{-11}$ in Table 4, the numerical integrator requires over 483 seconds. We know from Figure 6 that the perturbative/iterative methods require 10 iterations to achieve approximately the same error as the time integrator; the remaining entries in the Time I column show that both the perturbative and iterative algorithms compute such a solution hundreds of times faster than the time integrator.

The Time II column in Table 6 records how long it takes the perturbative/iterative algorithms to achieve the errors recorded in Table 4. Observe that even if we run the perturbative/iterative algorithms all the way to full convergence, they are much faster than time integration. In this case, the time integrator is 542 (respectively, 66) times slower than the iterative (respectively, perturbative) algorithm.

Note that the perturbative and iterative algorithms were implemented in Python using the Numpy/Scipy packages. The *dopri5* implementation used for numerical time integration is the implementation provided by the `scipy.integrate.ode` module. All times reported are average times across 10 runs on the same machine.

Gap Tuning

For each $N \in \{25, 75, 125, 175\}$, and for each of 10 values δ chosen in an equispaced fashion from the intervals given above, we compute 100 runs of the complete procedure described above—see Gap Tuning: Methodology. For each such run, we compute pre/post values of κ and V for three values of the input forcing amplitude, which we take to be the same at all input nodes k : $A_k \in \{0.001, 0.005, 0.01\}$. These results for κ and V , averaged over the 100 runs, are plotted in Figures 8, 9, and 10.

Figure 8 shows the results for Barabási-Albert (BA) graphs. By shrinking the gap between the first two eigenvalues, the percentage of energy transferred to higher harmonics (37) can be increased by approximately one order of magnitude, for all graph sizes, while the maximum magnitude voltage (38) can be increased by a factor of 6 to 20, depending on the graph size. Note that for larger graphs, choosing $\delta = 1$, i.e., forcing the first two eigenvalues to coincide, does not yield optimal energy transfer to higher harmonics.

The results in Figure 9 for Watts-Strogatz (WS) graphs are similar to those for BA graphs. We again find that by shrinking the gap between the first two eigenvalues, the energy transferred to higher harmonics (37) can be increased. However, the values of κ_{post} for Watts-Strogatz graphs are about twice as large as the values of κ_{post} for Barabási-Albert graphs in Figure 8. For all graph sizes, tuning the eigenvalue gap can increase the percentage of energy transferred to higher harmonics by a factor of up to 8, while the maximum magnitude voltage can be increased by approximately one order of magnitude.

In Figure 10, we present the results for Erdős-Rényi graphs. The results again support the finding that by shrinking the gap between the first two eigenvalues, the circuit can transfer more energy to higher harmonics and boost the peak magnitude of output signals. Specifically, we see that the energy transferred to higher harmonics (37) can be increased to $\kappa_{\text{post}} \in [1, 8]\%$, and the maximum voltage (38) can be increased to $V_{\text{post}} \in [0.1, 0.8]$.

The results for Erdős-Rényi graphs are much more strongly dependent on the number of nodes N than the results shown in Figures 8 or 9. Note that the peak voltages for the $N = 175$ graphs with forcing amplitude 0.01 are the largest voltages for any graphs considered in this paper. We can increase the peak voltages for smaller graphs by choosing a smaller value of the conductance than $G_i = 0.5$ (for all nodes i), the value used to compute the results in Figure 10.

For all three types of graphs, both *pre* and *post* values of κ and V increase as a function of the input forcing amplitude.

Code

All code necessary to reproduce the above results have been posted as a public repository on GitHub, accessible at the following URL: <https://github.com/GarnetVaz/Nonlinear-electrical-oscillators>

We use Python together with the `numpy`, `scipy`, `matplotlib`, and `networkx` modules for all numerical computing. The code that generates Figures 8, 9, and 10 is set to utilize 10 processors using the open-source multiprocessing module. For plotting, we use R together with the `ggplot2`, `plyr`, and `reshape` packages. All languages, packages and modules used are open source.

Assuming all packages and modules have been correctly installed, one can reproduce all results by running the Python codes `numerical_comparison.py` and `graphmulti.py`. The latter code may require several hours to run. The Python codes will generate figures using R; the R codes we provide need not be run independently.

Further details on how to run the codes, including the specific versions of required packages and modules, are given in the README.md file at the URL given above.

The code that we provide can easily be modified to run simulations not described here. For example, one can compare the perturbative/iterative algorithms against numerical integration using graphs other than the 20×20 grid graph used above. One can also explore gap tuning results for random graphs with different parameters than the ones we have chosen.

Conclusion

For nonlinear electrical circuits on arbitrary connected graphs, we have developed two numerical methods to compute the steady-state voltage. Using both absolute metrics and relative comparisons with a solution obtained via direct numerical integration, we validated the new algorithms. The results show that for the same error tolerance, both the perturbative and iterative methods are orders of magnitude faster than the solution obtained by time stepping. Moreover, these methods are able to capture the behavior in high Fourier modes and converge to machine precision in a fixed point error metric.

In future work, we plan to apply the steady-state algorithms developed above to solve Maxwell's equations in nonlinear electromagnetic media [26]. This application makes use of the correspondence between the nonlinear electrical network and a finite volume discretization of Maxwell's equations on an unstructured mesh.

In order to enhance the nonlinearity-driven features of these circuits, we developed a Newton-like algorithm that alters the eigenvalues of a network's graph Laplacian. The algorithm leaves the topology of the network untouched, changing only the inductances, i.e., the edge weights. By applying the Newton-like algorithm to three types of random graphs, we showed that reducing the gap between the graph Laplacian's first two eigenvalues leads to enhanced nonlinear behavior. Comparing pre- and post-optimized circuits, it is evident that optimizing the eigenvalue gap significantly increases (i) energy transfer from the fundamental driving frequency to higher harmonics, and (ii) the maximum magnitude output voltage.

In both the perturbative and iterative algorithm, the only way in which the network's structure influences its frequency response is through the graph Laplacian matrix. In our experiments, we have tuned this graph Laplacian's first eigenvalue gap by holding the topology of the graph constant and altering the edge weights, i.e., inductances. What if we had instead tuned the gap by holding the edge weights constant and altering the topology of the graph? This would amount to altering the incidence matrix B instead of the inductance vector L . So long as both types of alterations result in the same graph Laplacian Δ , our results indicate that the nonlinear electrical network's functionality should be enhanced significantly. This, of course, leads to the question of whether it is possible to algorithmically alter the network topology to achieve a particular graph Laplacian matrix, an interesting avenue for further work.

Acknowledgments

The authors thank the UC Merced Open Access Fund Pilot for covering this article's open access publication costs.

Author Contributions

Conceived and designed the experiments: HSB. Performed the experiments: GJV HSB. Analyzed the data: HSB GJV. Wrote the paper: HSB GJV. Designed the software used in analysis: GJV HSB.

References

- Lilis GN, Park J, Lee WE, Li G, Bhat HS, et al. (2010) Harmonic generation using nonlinear LC lattices. *IEEE Transactions on Microwave Theory and Techniques* 58: 1713–1723.
- Momeni O, Afshari E (2011) A broadband mm-wave and terahertz traveling-wave frequency multiplier on CMOS. *IEEE Journal of Solid-State Circuits* 46: 2966–2976.
- Lee W, Adnan M, Momeni O, Afshari E (2012) A nonlinear lattice for high-amplitude picosecond pulse generation in CMOS. *IEEE Transactions on Microwave Theory and Techniques* 60: 370–380.
- Lee W, Afshari E (2012) A CMOS noise-squeezing amplifier. *IEEE Transactions on Microwave Theory and Techniques* 60: 329–339.
- Han R, Afshari E (2013) A 260GHz broadband source with 1.1mW continuous-wave radiated power and EIRP of 15.7dBm in 65nm CMOS. In: *Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2013 IEEE International. pp. 138–139.
- Han R, Afshari E (2013) A high-power broadband passive terahertz frequency doubler in CMOS. *IEEE Transactions on Microwave Theory and Techniques* 61: 1150–1160.
- Bhat HS, Lee W, Lilis GN, Afshari E (2010) Steady-state perturbative theory for nonlinear circuits. *Journal of Physics A: Mathematical and Theoretical* 43: 205101.
- Landauer R (1960) Shock waves in nonlinear transmission lines and their effect on parametric amplification. *IBM Journal of Research and Development* 4: 391–401.
- Landauer R (1960) Parametric amplification along nonlinear transmission lines. *Journal of Applied Physics* 31: 479–484.
- Hirota R, Suzuki K (1970) Studies on lattice solitons by using electrical networks. *Journal of the Physical Society of Japan* 28: 1366–1367.
- Case M, Kamegawa M, Yu R, Rodwell MJW, Franklin J (1991) Impulse compression using soliton effects in a monolithic GaAs circuit. *Applied Physics Letters* 58: 173–175.
- Case M, Carman E, Yu R, Rodwell MJW, Kamegawa M (1992) Picosecond duration, large amplitude impulse generation using electrical soliton effects. *Applied Physics Letters* 60: 3019–3021.
- Afshari E, Hajimiri A (2005) Nonlinear transmission lines for pulse shaping in silicon. *IEEE Journal of Solid-State Circuits* 40: 744–752.
- Ostrowskii LA, Papko VV, Stepanyants YA (1980) Solitons and nonlinear resonance in two-dimensional lattices. *Zhurnal Eksperimental'noi i Teoreticheskoi Fiziki* 78: 831–841.
- Stepanyants YA (1981) Experimental investigation of cylindrically diverging solitons in an electric lattice. *Wave Motion* 3: 335–341.
- Stepanyants YA (1983) Experimental study of “Cerenkov” radiation from solitons in twodimensional LC-lattices. *Radiophysics and Quantum Electronics* 26: 601–607.
- Afshari E, Bhat HS, Hajimiri A, Marsden JE (2005) Extremely wideband signal shaping using one- and two-dimensional nonuniform nonlinear transmission lines. *Journal of Applied Physics* 99: 054901.
- Bhat HS, Afshari E (2008) Nonlinear constructive interference in electrical lattices. *Physical Review E* 77: 066602.
- Bhat HS, Osting B (2009) The zone boundary mode in periodic nonlinear electrical lattices. *Physica D* 238: 1216–1228.
- Wyart M, Liang H, Kabla A, Mahadevan L (2008) Elasticity of floppy and stiff random networks. *Physical Review Letters* 101: 215501.
- Monthus C, Garel T (2011) Random elastic networks: a strong disorder renormalization approach. *Journal of Physics A: Mathematical and Theoretical* 44: 085001.
- Tighe BP (2012) Dynamic critical response in damped random spring networks. *Physical Review Letters* 109: 168303.
- DiDonna B, Lubensky T (2005) Nonaffine correlations in random elastic media. *Physical Review E* 72: 066619.
- Togashi Y, Mikhailov AS (2007) Nonlinear relaxation dynamics in elastic networks and design principles of molecular machines. *Proceedings of the National Academy of Sciences* 104: 8697–8702.
- Bhat HS, Osting B (2011) Kirchhoff's laws as a finite volume method for the planar Maxwell equations. *IEEE Transactions on Antennas and Propagation* 59: 3772–3779.
- Bhat HS, Vaz GJ (2013) Finite volume method for planar Maxwell equations in nonlinear media. In preparation.
- Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU (2006) Complex networks: Structure and dynamics. *Physics Reports* 424: 175–308.
- Kori H, Mikhailov A (2006) Strong effects of network architecture in the entrainment of coupled oscillator systems. *Physical Review E* 74: 066115.
- Park K, Huang L, Lai YC (2007) Desynchronization waves in small-world networks. *Physical Review E* 75: 026211.
- Arenas A, Diaz-Guilera A, Kurths J, Moreno Y, Zhou C (2008) Synchronization in complex networks. *Physics Reports* 469: 93–153.
- Jalili M, Rad A, Hasler M (2008) Enhancing synchronizability of weighted dynamical networks using betweenness centrality. *Physical Review E* 78: 016105.
- Donetti L, Hurtado PI, Muñoz MA (2005) Entangled networks, synchronization, and optimal network topology. *Physical Review Letters* 95: 188701.
- Brede M (2008) Locals vs. global synchronization in networks of non-identical Kuramoto oscillators. *The European Physical Journal B* 62: 87–94.
- Brede M (2008) Construction principles for highly synchronizable sparse directed networks. *Physics Letters A* 372: 5305–5308.
- Dadashi M, Barjasteh I, Jalili M (2010) Rewiring dynamical networks with prescribed degree distribution for enhancing synchronizability. *Chaos* 20: 043119–043119-6.
- Kelly D, Gottwald GA (2011) On the topology of synchrony optimized networks of a Kuramoto-model with non-identical oscillators. *Chaos* 21: 025110.
- Yanagita T, Mikhailov AS (2012) Design of oscillator networks with enhanced synchronization tolerance against noise. *Physical Review E* 85: 056206.
- Bhat HS, Osting B (2011) 2-D Inductor-capacitor lattice synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30: 1483–1492.
- Hagberg AA, Schult DA, Swart PJ (2008) Exploring network structure, dynamics, and function using NetworkX. In: *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA USA, pp. 11–15.
- Barabási AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286: 509–512.
- Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. *Nature* 393: 440–442.
- Erdős P, Rényi A (1960) On the evolution of random graphs. *Magyar Tud Akad Mat Kutató Int Közl* 5: 17–61.