A Numerical Method for Nonlinear Analysis on the Sierpinski Gasket

Northern Arizona University REU Program 2012*

Robert J. Ravier

Department of Mathematics
Cornell University
Ithaca, NY

April 14, 2013

*This paper was funded by the NSF

Abstract

We seek solutions of the nonlinear equation $\Delta u + f(u) = 0$ on the Sierpinski Gasket (abbreviated SG), where Δ is the usual Kigami Laplacian. A result due to Strichartz suggests that solutions of this equation can be uniformly approximated by solutions of $\Delta_m^A u + f_m(u) = 0$, where Δ_m^A is the Laplacian on the m'th level cell graph approximation of SG. We'll start with a basic introduction to analysis on SG and some new results concerning its cell graph approximations. With this background, we'll detail a numerical algorithm to perform a complete nonlinear analysis on SG. Using $f(u) = su + u^3$ as an example, where s is a bifurcation parameter, we'll discuss some current results and explain how the highly symmetric nature of SG leads to difficulties in performing such nonlinear analysis.

1 Introduction

Let u be a function on the Sierpinski Gasket (known hereforth as SG) such that $\Delta u = -f$. Here, Δ is the usual Kigami Laplacian. Assuming arbitrary boundary conditions, in the case that the above equation is linear, a unique solution exists, and is given by:

$$u(x) = \int_{SG} G(x,y) f(y) d\mu(y) + h(x). \tag{1.1}$$

Here, $G(x,y)$ is the Green's function on SG, and h is the harmonic function satisfying $u|_{\partial SG} = h|_{\partial SG}$ (See ([1]) for the derivation).

Consider the case where f is nonlinear. The formula above then yields an implicit relation which u must uniquely satisfy, but such a relation does not allow us to obtain a solution. We are thus forced to consider other methods, be they analytic or numeric, in order to do nonlinear analysis on SG.

In this paper, we apply the Gradient Newton Galerkin Algorithm of Neuberger and Swift ([2]) to analyze the nonlinear problem with $f(u) = su + u^3$, where s is a bifurcation parameter. We hope to use the complexities of this particular example, explained further in the paper, to give insight into the most difficult aspect of nonlinear analysis on SG and other self-similar fractals: defining a notion of equivalence of solutions.

2 Preliminaries for SG

2.1 Definition and Construction

Let q_1, q_2, q_3 denote the vertices of an equilateral triangle. For the purposes of this paper, q_1 will be the left vertex, q_2 will be the upper vertex, and q_3 will be the right vertex. Define $F_i : \mathbb{R}^2 \to \mathbb{R}^2$ by

$$F_i(x) = \frac{1}{2}(x - q_i) + q_i. \tag{2.1}$$

for $i = 1, 2, 3$. SG is defined to be the unique nonempty compact set satisfying

$$SG = \bigcup_{i=1}^{3} F_i(SG). \tag{2.1}$$

We define a word of length n, $(w_1, ..., w_n)$ to simply be an element of $\mathbb{Z}_3^n$. Then, we say that $F_w = F_{w_n} \circ ... \circ F_{w_1}$. If T is the unit equilateral triangle, the m'th level approximation of SG is $\bigcup_{|w|=m} F_w(T)$, where $|w|$ is the number of components in the word w.

With this approximation, we can discuss two sets of graph approximations of SG. The first, the dyadic point graph, is the most intuitive. Given the m'th level approximation of SG, the vertices of the dyadic point graph are the vertices of the triangles of the m'th level approximation, and the edges are the edges on the m'th level approximation

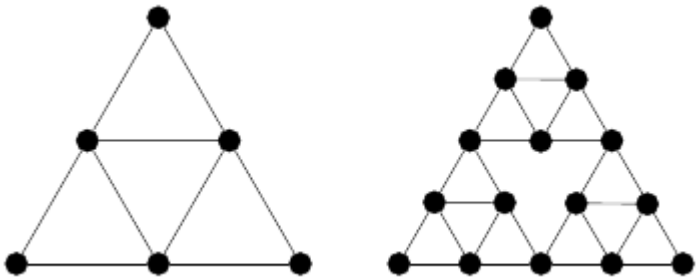

Figure 1: The dyadic point graphs corresponding to the level 1 and level 2 approximations of SG.

The other graph approximation we can use is the m'th level cell graph, which we will denote Γ_m. The vertices of this graph represent the right-side-up triangles of the m'th order approximation of SG. An edge between two vertices indicates that two triangles share a corner in common. It's important to note that no vertices or edges are preserved when going from Γ_m to Γ_{m+1}. This is because the triangles in the level m approximation of SG split into three separate triangles in the level $m+1$ approximation, so no triangles are preserved in subsequent approximations.

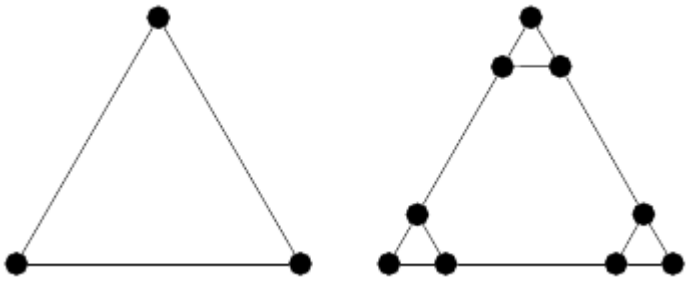

Figure 2: The cell graphs corresponding to the level 1 and level 2 approximations of SG.

In addition, when working with Γ_m, we assume that all function values are the average values of functions on the smallest triangles on the dyadic point graphs. Specifically, if a, b, c are values on the vertices of a small triangle on the dyadic point graph, then the function value on Γ_m for the vertex corresponding to the triangle is $\frac{a+b+c}{3}$. We also assume that if x is the value of a vertex on Γ_m, and that vertex splits into three vertices in Γ_{m+1} with values d, e, f respectively, then the values must satisfy

$$x = \frac{d+e+f}{3}. \tag{2.2}$$

.

2.2 The Laplacian

We define the renormalized graph energy of a function u on the level m dyadic point graph by

$$E_m(u) = \frac{3}{2}5^m \sum_{x \sim y} (u(x) - u(y))^2, \tag{2.3}$$

and we can similiarly define the bilinear form on the same level dyadic point graph by

$$E_m(u,v) = \frac{3}{2}5^m \sum_{x\sim y}(u(x)-u(y))(v(x)-v(y)). \tag{2.4}$$

We can then define the energy of a function on SG (and the corresponding bilinear form) by taking the limit as $m \to \infty$. A function on SG that has finite energy is said to be an element of dom$E$. It is not hard to prove the following:

Theorem 1. *The set dom$E$ modulo the constant functions is a Hilbert space with respect to $E(u,v)$.*

With this notion of an inner product, we can then define the Laplacian on SG by the following:

Definition Let $u \in$ dom$E$. Then $\Delta u = f$ for f continous if

$$E(u,v) = -\int_{SG} fvd\mu \tag{2.5}$$

for all $v \in$ dom$E$ that are zero on the boundary of SG.

Note that if we replace SG with the unit interval I, where the dyadic point graphs correspond to P_m, the m'th level path, dom$E$ corresponds with H where H is defined as usual. A quick exercise in integration by parts will then show that f corresponds with u'', so we can reasonably say that our definition of the Laplacian in terms of integrals is indeed the same as the usual definition. Our definition has the added benefit of ignoring the problem of a lack of smooth coordinates.

Nevertheless, we naturally think of derivatives in terms of difference quotients, so it would be more intuitive to have a definition of the Laplacian in terms of a difference quotient. We define the renormalized graph Laplacian of a function u on Γ_m at vertex x to be

$$\Delta_m u(x) = \frac{3}{2}5^m \sum_{x\sim y}(u(y)-u(x)). \tag{2.6}$$

Note that we can find the Laplacian of the entire function on Γ_m by writing u as a column vector, which is possible as the function space of Γ_m is clearly isomorphic to $\mathbb{R}^{3^m}$. The Laplacian of u is then Lu, where $L = D^T D$, where D is the difference matrix of $Gamma_m$.

With this familiar notion of a derivative in mind, it's natural to ask whether we can use it to get the Laplacian as defined by Equation (2.5). The following suggests yes.

Theorem 2. *Let $\Delta u = f$. Then $\Delta_m u$ converges uniformly to f. Conversely, if u is integrable, and $\Delta_m u$ converges uniformly to a continuous function f, then Δu exists and equals f.*

For a proof of this, see ([3]). Note that the theorem assumes, unlike us, that the graph Laplacian isn't defined at the corners. However, this doesn't matter in the limit, so we can keep our condition without worry. This theorem suggests that we can approximate the Laplacian of SG by taking m large enough.

3 The Algorithm

3.1 The Energy Functional

The results of Theorem 1 suggest that solutions of the equation $-\Delta u + f(u) = 0$ on SG can be approximated by solutions of $-Lu + f_m(u) = 0$ on Γ_m for m large enough. This means that we can approximate solutions of the nonlinear problem on SG by looking at solutions of the nonlinear problem on Γ_m. Therefore, solving the nonlinear problem on Γ_m will approximate solving the nonlinear problem on SG. To solve the nonlinear problem on Γ_m, we consider the functional $J : \mathbb{R}^{3^m} \to \mathbb{R}$ defined by

$$J(u) = \frac{1}{2} Du \cdot Du - \sum_{i=1}^{3^m} F(u_i). \tag{3.1}$$

where F denotes the primitive of f and D is the previously mentioned adjacency matrix. Note that as Γ_m has 3^m vertices, its corresponding function space is isomorphic to $\mathbb{R}^{3^m}$ in the obvious way, which justifies our choice of domain (change note to Section 2?). A simple calculation yields:

$$J'(u)(v) = -(-Lu + f(u)) \cdot v. \tag{3.2}$$

This computation immediately yields the following theorem:

Theorem 1. *Let $u \in \mathbb{R}^{3^m}$. Then u is a critical point of $J : \mathbb{R}^{3^m} \to \mathbb{R}$ if and only if u is a solution of $-Lu + f(u) = 0$*

This theorem tells us that all solutions of the nonlinear problem on Γ_m can be found by applying a root finding algorithm to J'.

3.2 The GNGA and Variants

Our algorithm of choice is the Galerkin Newton Gradient Algorithm (hereforth abbreviated as GNGA) of Neuberger and Swift. The algorithm is essentially little more than multivariable Newton's method on J'. Recall that for $x \in \mathbb{R}^n$, Newton's method finds a solution of $F(x) = 0$ with an initial guess x_0 and the following iteration:

$$x_{n+1} = x_n - \mathbf{D}F(x_n)^{-1}F(x_n). \tag{3.3}$$

Newton's method is guaranteed to converge to a root provided that the initial guess is within some sufficiently small neighborhood of said root. In the case of the GNGA, F corresponds to J', and the derivative matrix $\mathbf{D}F$ corresponds to the Hessian matrix of J.

We can limit the number of numerical differentiations and integrations needed to do the GNGA as follows. Pick an orthonormal eigenbasis $\{\psi_j\}_{j=1}^{3^m}$ of $\mathbb{R}^{3^m}$ with respect to the linear transformation L. Then, with $u = \sum_{i=1}^{3^m} a_i\psi_i$, the k'th component of $J'(u)$ is given by

$$J'(u)_k = a_k\lambda_k - f(u) \cdot \psi_k \tag{3.3}$$

where λ_k is the eigenvalue corresponding to ψ_k. Such a choice of basis also gives that the jk'th compoent of the Hessian matrix is

$$h(u)_{jk} = \lambda_j\delta_{jk} - \text{diag}(f'(u))\psi_j \cdot \psi_k, \tag{3.4}$$

where f' is the derivative of f with respect to u.

We can use a modified version of the GNGA in order to compute bifurcation diagrams for the difference equation. The tangent augmented GNGA (dubbed the tGNGA) is a predictor-corrector continuation method: given a point on a solution branch, we use a linear approximation to obtain a vector close to a root, and then use a constraint to correct our prediction in order to get a root. We can estimate the tangent vector at the current point $p_c = (a_c, s)$ (ie the coefficient vector corresponding to the basis with the current value of s affixed) by finding the previous point p_o in the continuation and letting the unit tangent $v = \frac{p_c - p_o}{||p_c - p_o||}$. The predicted point is then $p_g = p_c + \delta v$, where δ is some scalar denoting the speed at which

we move along the branch. The correction constraint is $\kappa = (p - p_g) \cdot v = 0$, ie the solution lies on the space normal to the tangent vector. This is guaranteed by setting $v = (\nabla_a \kappa(a, s), \frac{\partial \kappa}{\partial s}(a, s))$. With this constraint, we find the new solution by solving both the initial equation for search direction in Newton's method as well as this constraint. If we let δ be small enough, we can compute a branch of solutions by this tGNGA.

3.3 Processing Bifurcations

Te signature of u is the number of negative eigenvalues of the $Hessian$ of J. In the case that we assume that solutions of the difference equation are nondegenerate, then the signature equals the Morse Index, or the number of "down" directions at the critical point of J. If the Morse Index changes while traveling along a branch, then there must be a bifurcation point along the branch, ie a point on the branch where it splits from one branch into multiple other branches.

If the Morse Index at p_o is k and the Morse Index at p_c is $k + d$, then we know that a bifurcation point occurs somewhere between p_o and p_c where the Hessian is not invertible, ie where the r'th eigenvalue, where the eigenvalues are sorted in ascending order, is 0. Here, $r = k + \lceil \frac{d}{2} \rceil$. In order to find the bifurcation oint, we apply the secant method. To do this, we let $p_0 = p_o, p_1 = p_c$, and let β_0 and β_1 be the r'th eigenvalue of the Hessian at p_0 and p_1. We can find the bifurcation point via iteration. To do this, we let our guess point $p_g = p_i - \frac{(p_i - p_{i-1})\beta_i}{\beta_i - \beta_{i-1}}$, and then use the tGNGA to correct the guess. We keep iterating until we get a point with the r'th eigenvalue sufficiently close to zero.

Once we find the bifurcation point, we need to choose the directions to search for branches. The current method simply involves testing to see which direction gives the most stable result (by stable, we mean least likely to jump off to a far away branch) and thus testing that directions. The possible directions that this method chooses are simply the direction of the basis vectors.

4 Construction of an Eigenbasis on Γ_m

To use the GNGA with a minimal number of numerical operations, we require an orthonormal eigenbasis. We can proceed to do this by applying Gram-Schmidt to a known basis. While suites such as LAPACK can be used to numerically compute an eigenbasis, we cannot control the resulting structure of the output. This section

of the paper focuses on a construction of a highly localized basis. By localized, we simply mean a basis with small support.

To the best of our knowledge, all of the information in this section has not been published.

4.1 Spectral Decimation

Before we can give an explicit construction of the basis, we need some additional tools that will allow us to transition between eigenfunctions on Γ_m and eigenfunctions on higher level graphs. For the vertex graph approximations of SG, such a process exists and is known as spectral decimation. For the vertex graph, spectral decimation gives a set of formulae which allows you to take an eigenfunction on $Gamma_m$ and extend it into at most two eigenfunctions on Γ_{m+1}, each with different eigenvalues. Using spectral decimation, we can obtain the Dirichlet and Neumann spectra of SG in the limit. Spectral decimation is also used in the construction of the eigenbasis for the vertex graphs.

Fortunately, an analogous set of formulae are available for Γ_m.

Theorem 1. *Let u be an eigenfunction on Γ_m with eigenvalue λ_m. Then, u can be extended to at most two eigenfunctions on Γ_{m+1} with eigenvalues λ_{m+1}^1 and λ_{m+1}^2. Furthermore, for each λ_{m+1}^k, the corresponding extension is unique.*

Given that the proof is noninstructive in the sense that it involves playing around with equations in an unenlightend manner, we will skip the complete derivation. However, we'll outline how to get the formulas given a few notational definitions. The derivation of the formulae will clearly show existence and uniqueness up to eigenvalues.

Consider the figure below. The picture on the left details a general subgraph of the interior of Γ_m for $m > 1$ (The case for $m = 1$ can be obtained by simply deleting the point labeled W and its corresponding edge connecting it to the triangular group of vertices). We extend it to a corresponding general subgraph of Γ_{m+1} in the picture adjacent to it.

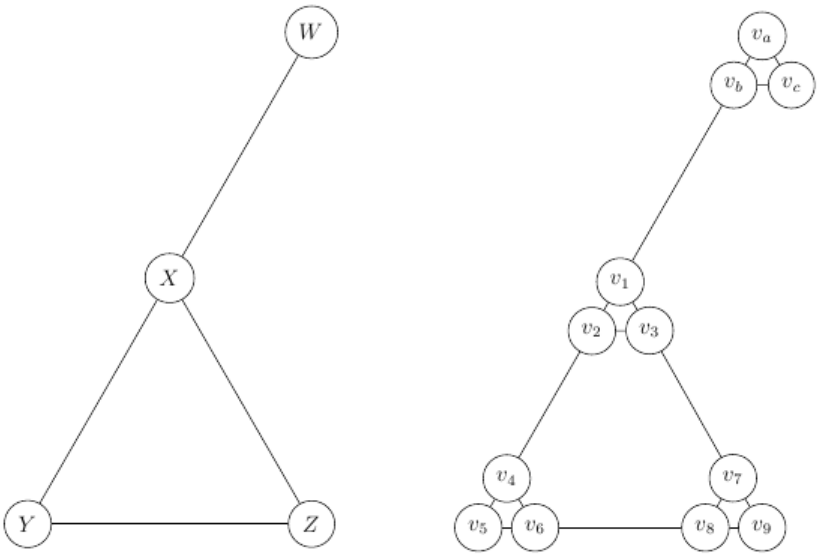

Figure 3: On the left, a general subgraph of Γ_m centered for an interior vertex X. On the right, that subgraph extended down to Γ_{m+1}.

Let $u(P)$ denote the value of u at vertex P. Assume that u is an eigenfunction with eigenvalue λ_m on Γ_m. So, for cell X, we have the equation

$$(3 - \lambda_m)u(X) = u(W) + u(Y) + u(Z). \tag{4.1}$$

Now, assume that u extends to an eigenfunction on Γ_{m+1}. This means that, for vertex v_1,

$$3(3 - \lambda_{m+1}u(v_1) = u(v_2) + u(v_3) + u(v_b) \tag{4.2}$$

and similarly for every vertex except possibly for v_5, v_9, and v_a, as one of them might be boundary vertices. By playing around with these equations for known interior points, and using the mean value property (2.2) of the cell graph, we obtain

$$u(v_1) = \frac{3(4 - \lambda_{m+1})u(X) + 3u(W)}{(3 - \lambda_{m+1})(5 - \lambda_{m+1})} \tag{4.3}$$

and similarly for all of the other interior vertices. In other words, the value of the extended eigenfunction on a vertex is a function of the vertex's parent, the (different) parent of the nearest neighboring vertex, and the eigenvalue of the extended function. The $(4 - \lambda_{m+1}$ term acts as a weighting factor, which makes sense as the value of the function at the parent cell should affect the value more.

To extend an eigenfunction on the boundary, we see in the above figure that if Y is a boundary vertex on Γ_m, then v_5 is a boundary vertex on Γ_{m+1}. To extend u to an eigenfunction on Γ_{m+1}, we first observe that such an eigenfunction would satisfy

$$(2 - \lambda_{m+1})u(v_5) = u(v_4) + u(v_6). \tag{4.4}$$

We then add $u(v_5)$ to both sides and apply (2.something) to get

$$u(v_5) = \frac{3u(Y)}{3 - \lambda_{m+1}}. \tag{4.5}$$

This is consistent with (4.3) as every vertex adjacent to v_5 shares its parent cell, so the only factors that should matter are the current eigenvalue and the value of the function on the parent cell.

All that remains is to figure out what the λ_{m+1}^k are. Again, there is no special method needed to do this. By playing around the equations for interior points, and using (4.1), we get the relation

$$\lambda_m = \lambda_{m+1}(5 - \lambda_{m+1}), \tag{4.6}$$

which has solutions

$$\lambda_{m+1} = \frac{5 \pm \sqrt{25 - 4\lambda_m}}{2}. \tag{4.7}$$

The "at most two" in the theorem comes from (4.3) and (4.5). The equation (4.7) produces two new eigenvalues via the quadratic formula as usual. However, if one of these eigenvalues happens to be 3 or 5, the equations for continuation in the interior are no longer valid, so we cannot extend to eigenfunctions in these cases.

4.2 The Basis

With spectral decimation in hand, we now have all of the tools that we need in order to produce the basis. Our construction is by recursion.

The basis for Γ_1 is easily computed by inspection. It consists of a constant function, a nonconstant function, and a rotation of the nonconstant function. The constant function has eigenvalue 0, and the other two have eigenvalue 3.

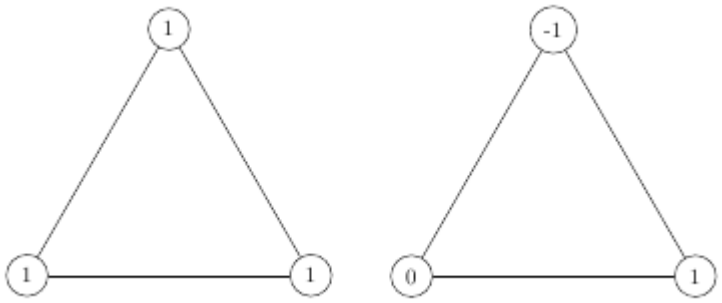

Figure 4: The basis of Γ_1 consists of the constant eigenfunction on the left and two rotations of the eigenfunction on the right

Now, consider Γ_2. We can create 5 basis elements by using the spectral decimation equations derived above to extend the three elements of the basis of Γ_1 down to Γ_2. Note that (4.7) implies that the constant eigenfunction can only bifurcate into eigenfunctions with eigenvalues 0 and 5. However, (4.3) shows that 5 is a forbidden eigenvalue, so the constant eigenfunction can only extend to the constant eigenfunction. The remaining elements of the basis are computed by inspection, and are listed below. The first type is a "battery chain" construction around the hexagon in the graph. Start at a point that lies on the hexagon and place a -1. Then go around the hexagon clockwise, alternating between placing 1 and -1 on vertices until every point on the hexagon is nonzero. The second type is constructed by placing a 2 at one boundary vertex, -1 at its adjacent vertices, -1 at the adjacent vertices of the boundary point's adjacent vertices, and 1 at the two remaining non-boundary vertices. The remaining elements of the basis consist of one function of the first type, and the three rotations of the second type.

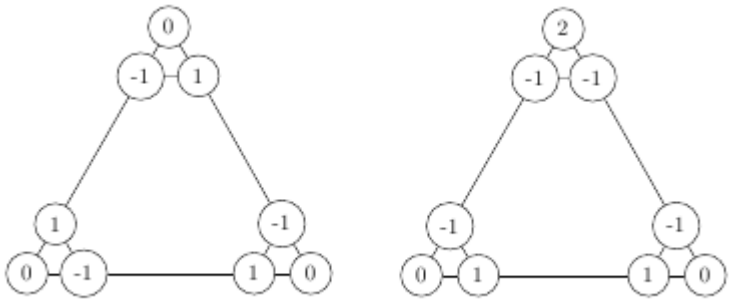

Figure 5: The two types of basis elements for Γ_2 that are not continued from Γ_1

In general, consider going from Γ_{m-1} to Γ_m. Using spectral decimation, we take the eigenbasis on Γ_{m-1} and extend it to a linear independent set in with cardinality $2 \cdot 3^{m-1} - 1$ on Γ_m, where every eigenfunction in the basis on Γ_{m-1} extends to two eigenfunctions on Γ_m except for the constant function, which only extends to the constant function.

We then look at the hexagons on Γ_m. Each hexagon corresponds to exactly one upside-down triangle on the graph of the dyadic points of SG. A simple argument shows that there are $1 + 3 + 3^2 + ... + 3^{m-2} = \frac{3^{m-1}-1}{2}$ hexagon cycles on Γ_m. We proceed as we did for the first type of non-extended eigenfunction on Γ_2: pick a hexagon and a vertex on it and assign it a value of -1, then continue clockwise around the hexagon, alternating between 1 and -1 until every vertex on the hexagon has a nonzero value. Do this for every hexagon on Γ_m to get $\frac{3^{m-1}-1}{2}$ eigenfunctions with eigenvalue 5.

We now considerthe eigenfunctions with eigenvalue 3. First, we take each vertex on the boundary of Γ_m and copy the appropriate rotation of the second type of non-extended eigenfunction on Γ_2 onto the corresponding copy of Γ_2 on the boundary of $Gamma_m$, with the 2 being placed on the vertex on the boundary. This yields three eigenfunctions. To get the remaining eigenfunctions, we consider the 3^{m-2} copies of Γ_2 in Γ_m. Take two adjacent copies of Γ_2, and consider the edge connecting them. Assign a value of 2 to each of the vertices on the edge, and then repeat the construction of the second type of nonextended eigenfunction on Γ_2 twice. Refer to the figure below for the specific construction. An inductive argument shows that there are $\frac{3^{m-1}-3}{2}$ eigenfunctions of this type, so we have a total of $\frac{3^{m-1}+3}{2}$ eigenfunctions at level m with eigenvalue 3.

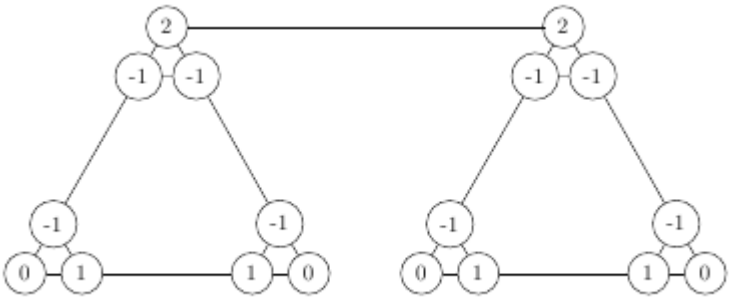

Figure 6: The type of eigenfunctions with eigenvalue 3 on level m that have no support on the boundary

Counting the eigenfunctions that we have thus far, we see we have

$$2 \cdot 3^{m-1} - 1 + \frac{3^{m-1} - 1}{2} + \frac{3^{m-1} + 3}{2} = 3^m.$$

However, we can have at most 3^m linear independent basis elements, so we have necessarily constructed the eigenbasis.

5 Numerical Implementation

5.1 Construction of Γ_m

As is common with SG, our construction is recursive. We start with the unit equilateral triangle, and recursively cut out smaller triangles until we get the desired level m approximation. In order to mathematically represent the graph Γ_m, each vertex is listed as an element of a Matlab structure array. Each element of the array consists of four components: the coordinates of the vertices making up the triangle, the edges of the triangle (represented by 1x2 arrays connecting the vertices), the neighboring triangles, and the ID of the triangle. The first three entries are self-explanatory. By the ID, we mean the word w corresponding to the appropriate fixed point map that maps the unit equilateral triangle to the cell of the m'th level approximation. The list of neighbors and the ID of the triangle are vital for our implementation of the basis, whereas the other two entries are used specifically for constructing contour plots.

5.2 The Basis

In order to construct the basis on Γ_m, we first need a few conventions. The column vectors that represent the functions on Γ_m will be ordered by the lexicographical sorting of the cell IDs. For example, on Γ_3, the first entry will be the value of the function at cell 111, the second entry will be the value of the function at cell 112, etc. We define the parent ID of the ID w to be the truncation to the first $m-1$ entries of w. The notion of grandparent ID can be defined analogously.

For convenience, the non-spectral decimation eigenfunctions on Γ_1 and Γ_2 were hardcoded into the program. This was done so that our methods for constructing the eigenfunctions obtained by spectral decimation, as well as the eigenfunctions with eigenvalues 3 and 5 would not need any special exceptions written into them. The methods used to obtain these functions are listed below. Each construction assumes that the vector associated with the eigenfunction is initially zero.

Algorithm 1 Spectral Decimation

Compute two possible eigenvalues λ_1, λ_2
for Each eigenvalue that's not 3 or 5 **do**
  for Each $v \in \Gamma_m$ **do**
    Get both the ID and the parent ID
    $X \leftarrow$ value of the parent cell
    if v is on the boundary **then**
      $u(v) \leftarrow 3 * X/(3-\lambda_i)$
    else
      Find w, the neighbor of v with a different parent ID
      Find w's parent ID
      $Y \leftarrow$ value of w's parent cell
      $u(v) \leftarrow 3 * ((4-\lambda_i) * X + Y)/((3-\lambda_i)(5-\lambda_i))$
    end if
  end for
end for

Algorithm 2 3-Eigenfunctions

for Each $v \in \Gamma_m$ **do**
  if v has a neighbor with a different grandparent or v is a boundary vertex **then**
    $u(v) \leftarrow 2$
    $v_1, v_2 \leftarrow$ neighbors of v that have the same grandparent
    $u(v_1), u(v_2) \leftarrow -1$
    $w_1, w_2 \leftarrow$ the neighbors of v_1, v_2 such that $u(w_1) = u(w_2) = 0$.
    $w_1, w_2 \leftarrow -1$
    for i = 1:2 **do**
      for Each neighbor n of w_i **do**
        if $u(n) = 0$ **then**
          if The neighbors of n all have the same grandparent **then**
            $u(n) \leftarrow 1$
          end if
        end if
      end for
    end for
  end if
end for

Algorithm 3 5-Eigenfunctions

for i = 0:m-2 **do**
  baseArrayLength $\leftarrow 3^{m-i}$
  for j = 1:3^i **do**
    s $\leftarrow 1$
    baseArray $\leftarrow$ rows (1 + $j-1$ * baseArrayLength) to j*baseArrayLength of sorted IDs
    for Each row in baseArray **do**
      testIndex $\leftarrow (i+1)$'st index of row
      if the entries with index greater than $i+1$ are different from testIndex **then**
        val $\leftarrow$ testIndex
        Put the row in valArray
      end if
    end for
    Sort the three valArrays
    for Each valArray **do**
      Get sorted list V of vertices corresponding to IDs in valArray
      for j = 1:valArray **do**
        $v \leftarrow V(j)$
        $u(v) = (-1)^s$
        $s \leftarrow s+1$
      end for
      Export u to list of eigenfunctions
      Reset u to zero vector
    end for
  end for
end for

6 Results and Symmetry

6.1 Introduction

With the basis in place, we can run the GNGA on Γ_m in an effort to approximate solutions of the nonlinear equation on SG. However, when attempting to run the algorithm on Γ_3 and higher, we run into two critical problems hindering our ability to do nonlinear analysis.

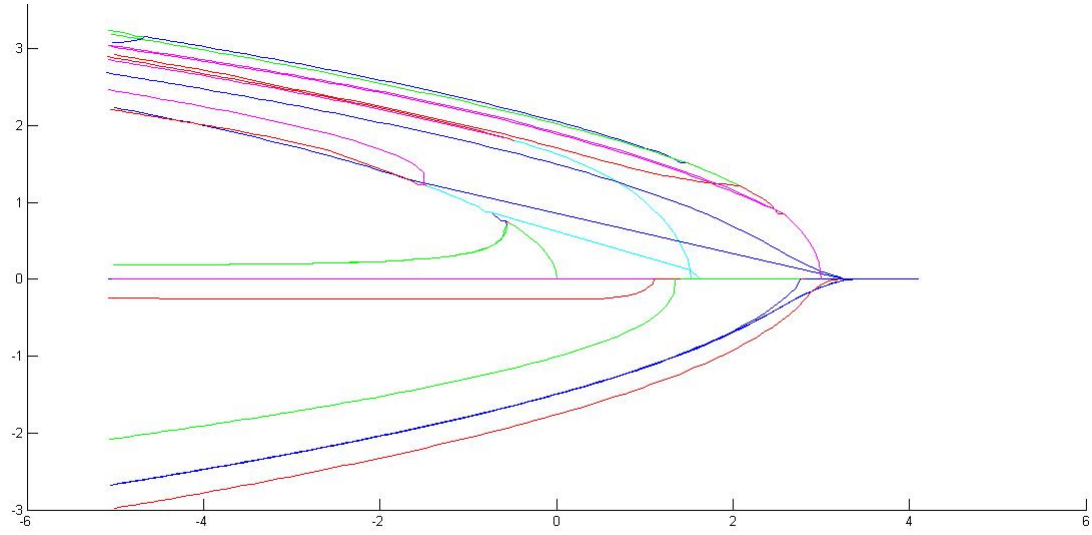

Figure 7: A partial bifurcation diagram on Γ_2. The plot is $u(1)$ vs. s, where $u(1)$ is the value of the function on the left boundary point.

When looking at the above bifurcation diagram, we see that some of the branch plots look fairly jagged. This stems from our method of predicting initial search direction. Recall from previous sections that we find the search direction at a bifurcation point by merely testing which of the basis directions seems to work the best. While this method works fine for bifurcation points with small changes in Morse Index, it is known to fail for high changes in Morse Index. Experiments on higher levels have shown bifurcation points have high jumps in Morse Index (for example, a bifurcation point on Γ_4 was shown to have a jump in Morse Index of 50). Because of this, we cannot expect reliable performance with our current

method of finding search direction, and the instability evident in the above bifurcation diagram supports this claim. Note that the instabilities are much more apparent for partial bifurcation diagrams for higher m. However, these issues have caused such bifurcation diagrams to be little more than bunches of squiggles, so such diagrams were omitted.

The high changes in Morse Index imply that bifurcation diagrams for even fairly low values of m will be quite complicated. Experiments on fairly good hardware have lasted for hours without any sign of termination. This is due to the large number of branches that are plotted and the large number of high multiplicity bifurcation points that are present in the diagrams.

The above suggests that our ability to do nonlinear analysis is somewhat limited by our current methods. To fix the first issue, we can employ a method known as the cGNGA ([4]). Our code currently contains an implementation of this, but at the moment, our current method of finding search directions is much less unstable, suggesting that our cGNGA implementation is bugged. Because of this, we cannot hope to process bifurcation points of high multiplicity at this time.

On the other hand, we can start to work towards ameliorating the issue of following a large number of branches. In order to do this, we need some notion of equivalence of solutions. We could then use such a notion in order to determine what branches to follow and which ones not to follow in our analysis. In order to find a proper definition of equivalence, we need to examine the possible symmetries on Γ_m. This task is quite difficult given the possibility for many local symmetries, not just symmetries with respect to $\mathbb{D}_3$. It doesn't help that our analysis of possible solution types is limited because of the lack of a working cGNGA; in order to get possible ideas for definitions of equivalence, we need to see examples of solutions to analyze their symmetries. However, because a large chunk of branches stem from bifurcation points of high multiplicity, we cannot guarantee accurate analysis.

We can, however, examine branches that bifurcate from points of low multiplicity with confidence that our methods are accurate. Specifically, we can examine two groups of branches. Recall that bifurcation points on the $u = 0$ branch (ie the $||u||$ axis) occur at each of the eigenvalues on Γ_m. The branches that stem from bifurcation points of multiplicity 1 or 2 are those obtained by taking the eigenfunctions on Γ_1 with $\lambda = 3$ and extending them down to eigenfunctions on Γ_m by spectral decimation, and taking the eigenfunction on Γ_2 with eigenvalue 5 and also extending down via spectral decimation. For the purposes of this paper, we restrict our analysis to the two smallest positive eigenvalues on Γ_m, which are obtained by taking the eigenvalue 3 on Γ_1 and extending it down via spectral

decimation, making sure that the eigenvalue gotten via (4.7) always comes from a minus sign.

6.2 Analysis of the Branches from the Smallest Positive Eigenvalues

Before we look at the branches, we first make some adjustments to the basis. We know from spectral decimation that there will be two elements of the basis corresponding to the smallest positive eigenvalue. We modify these elements by taking the even and odd projections corresponding to reflection about the vertex q_2, then replacing the current basis element with the projection with the highest norm.

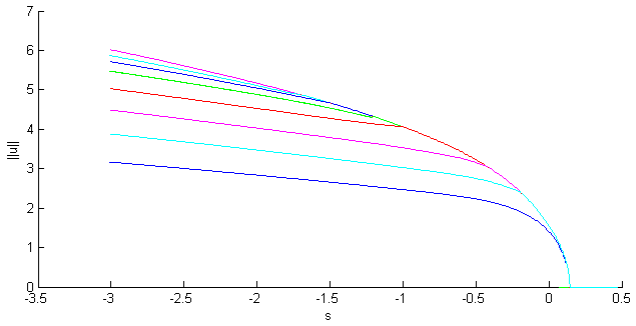

Figure 8: A bifurcation diagram on Γ_3 for the smallest positive eigenvalues. There are two main branches: the dark blue branch with MI 2, and the light blue branch (and daughters bifurcating off of it) with MI 3. The plot is ——u—— vs. s.

Consider the above bifurcation diagram for the smallest positive eigenvalues on Γ_3. As expected, there are two main branches that bifurcate from the trivial branch. The lower, dark blue branch is the one predicted by ([5]). This branch corresponds to the Morse Index 2 solution that changes sign exactly once. Solutions on this branch are odd. There are no secondary bifurcations on this branch.

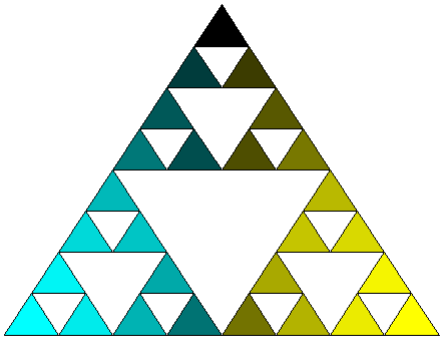

Figure 9: The CCN solution on Γ_3.

In order to understand the more complicated branch, we first make a definition.

Definition Let "$+$" denote evenness with respect to a $\mathbb{D}_3$ reflection, and let "$-$" denote oddness with respect to a $\mathbb{D}_3$ reflection. A *symmetry sequence of length* n on Γ_m is a list of n letters for n between 1 and m consisting of "$+$" and "$-$", where the k'st letter denotes the corresponding symmetry on each of the bottom $2^{k-1}\Gamma_{m-k}$ subgraphs.

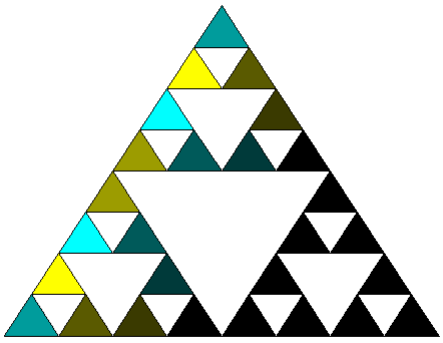

Figure 10: A solution on Γ_3 with symmetry sequence $(+,-)$.

With this definition in hand, we can look at the behavior of the other main branch. However, at the current moment, we have conflicting information. Our analysis on previously built C++ code suggests that the following branch structure: each secondary branch corresponds to the following pattern from the bottom: the first branch has symmetry sequence $(+)$, the second branch has trivial symmetry, the third branch has symmetry sequence $(+,+)$, the fourth branch has trivial symmetry. In general, every even numbered branch from the bottom has trivial symmetry, while every $2k+1$'s branch has a symmetry sequence of k pluses. This pattern continues until we reach the last branch, which has a symmetry sequence of m pluses. Our Matlab code suggest the branches all have symmetry sequences whose first terms are all pluses and last terms are either plus or minus. This discrepancy is likely due to our different methods of finding the search direction at bifurcation points, as the Matlab code's primary method of finding search direction is different from the usual cGNGA.

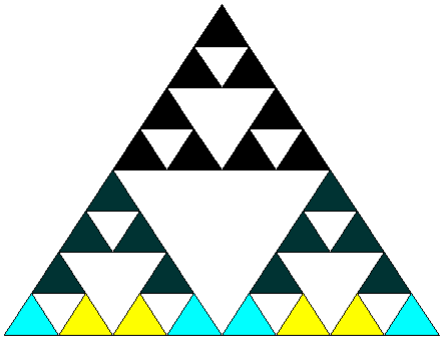

Figure 11: A solution on Γ_3 with symmetry sequence $(+,+)$. For the corresponding bifurcation diagram on the smallest positive eigenvalues of Γ_3, this would correspond to the third branch.

Nevertheless, our observations resulted in the following.

Theorem 3. *Consider the set of symmetry sequences for Γ_m that consist of "+" for all letters except possibly the last. Then each of the symmetry sequences corresponds to a subspace of the function space of Γ_m that is invariant under the map $-Lu + f(u)$ for f odd.*

Proof outline: Obvious for Γ_1. Assume this is true for Γ_{m-1}. Consider Γ_m; it has 3 Γ_{m-1} subgraphs. The invariance for each of the sequences on the edges not connecting the Γ_{m-1} subgraphs follows directly from the inductive assumption. It remains to check the behavior at the edges connecting the subgraphs. The invariance follows directly from the presence of the first letter in the sequence. Take care to remember that the spaces of trivial symmetry are automatically invariant.

Note that this result does not hold for any symmetry sequences that contain a "$-$" but do not terminate afterwards. This can be seen by comparing the Laplacians on the bottom row of Γ_m. Specifically, the two middle points will lack some symmetry (the symmetry in question depending on the sequence).

7 Conclusions

7.1 Further Inverstigation

It goes without saying that our investigation is far from complete. While we have a good idea as to what happens on the two branches corresponding to the smallest nonzero eigenvalue, we do not know what happens on the other branches. Analysis can be done on the branches stemming from the Γ_2 eigenfunction with eigenvalue 5 in a similar manner as the eigenvalues are simple. However, to get the other branches, the issues in the cGNGA must be fixed, as at the current moment we cannot guarantee accuracy. We know that other symmetry types that don't fit our symmetry sequence definition exist (for example, the Γ_2 eigenfunction with eigenvalue 5 id odd with respect to every reflection), but what types we should expect are currently unknown. When the cGNGA begins working properly, we will hopefully be able to analyze the symmetries in a timely manner.

We could also look into the idea of whether a function that satisfies a pde on Γ_m can be extended down to a function on Γ_{m+1} that satisfies the same pde. Methods for such an extension are not clear as the most obvious method to prove such a claim, namely the way with which we derived the spectral decimation formula, yields an implicit equation, which does us no good. If such methods exist, this would lead to an easy proof that the pde on SG has infinitely many solutions, but such an extension method may be little more than a pipedream at the current point.

The basis currently used is simply the basis constructed in section 4 after an application of Gram-Schmidt. Such a naive application will cause us to lose the localization we initially had in our eigenbasis. We could look for ways to improve

the orthonormalization process in order to maximize the localization of the basis.

7.2 Acknowledgements

The author would like to thank John Neuberger and James Swift for mentoring him this past summer, as well as for allowing him to use their base code for analysis on SG as well as assisting with the symmetry analysis. The author would also like to thank Robert Strichartz for past (and continued mentorship), and the NSF for funding his REU project.