NUMERICAL INTEGRATION ON GRAPHS: WHERE TO SAMPLE AND HOW TO WEIGH

GEORGE C. LINDERMAN AND STEFAN STEINERBERGER

ABSTRACT. Let $G = (V, E, w)$ be a finite, connected graph with weighted edges. We are interested in the problem of finding a subset $W \subset V$ of vertices and weights a_w such that

$$\frac{1}{|V|} \sum_{v \in V} f(v) \sim \sum_{w \in W} a_w f(w)$$

for functions $f : V \to \mathbb{R}$ that are "smooth" with respect to the geometry of the graph; here $\sim$ indicates that we want the right-hand side to be as close to the left-hand side as possible. The main application comprises problems where f is known to vary smoothly over the underlying graph but is expensive to evaluate on even a single vertex. We prove an inequality showing that the integration problem can be rewritten as a geometric problem ("the optimal packing of heat balls"). We discuss how one would construct approximate solutions of the heat ball packing problem; numerical examples demonstrate the efficiency of the method.

1. INTRODUCTION

The purpose of this paper is to report on a very general idea in the context of sampling on graphs. A graph for us will always be comprised of a finite set of n vertices V, a set of edges $E \subset V \times V$ which are equipped with weights: we use $\omega_{xy} > 0$ to denote the weight of the edge between the vertices $x, y \in V$. We will only deal with connected graphs. For simplicity of notation and exposition, we will assume that the vertices themselves are unweighted (or, equivalently, all have weight 1).

In a very real sense, the problem of "sampling" a function on a finite graph is a generalization of the problem of sampling a function $f : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^d$ since any bounded domain Ω can always be approximated by a graph by taking

$$V = \left\{ x \in \mathbb{Z}^d : \lambda x \in \Omega \right\},$$

where $\lambda > 0$ is a small parameter and

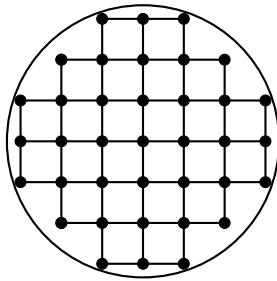$$E = \left\{ (x, y) \in V \times V : \|x - y\|_{\ell^1} = 1 \right\}.$$

FIGURE 1. The Integration Problem on continuous domains is a special case of the Integration Problem on finite graphs. However, there are many more graphs and they can be quite a bit more complicated than the grid.

Figure 1 shows an example: the unit disk with $\lambda = 3/10$. We can summarize the main problem as follows:

> **Classical Numerical Integration** tries to find the average value of a function $f : \Omega \to \mathbb{R}$ by sampling in specific points using specific weights.
>
> **Numerical Integration on Graphs** tries to find the average value of a function $f : V \to \mathbb{R}$ by sampling in specific vertices using specific weights.

Since we can approximate every domain $\Omega \subset \mathbb{R}^d$ by a finite graph, *any reasonable numerical integration technique needs to correspond to a reasonable numerical integration technique in Euclidean space.* However, since graphs are more general, even the description of such techniques will require a very different setup. Moreover, problems in numerical integration only make sense within a certain smoothness class: without any assumptions on the regularity of functions, numerical integration is impossible since the behavior of a function in a finite set of points may have little to no implications for the behavior of that function in other points. The same problem dominates finite graphs: we need to somehow specify smoothness properties of the function $f : V \to \mathbb{R}$; otherwise the question of numerical integration on a graph is not meaningful.

Problem (Quadrature)**.** If we are allowed to sample in a set $W \subset V$ of $|W| = k$ vertices, which set of vertices W and which weights a_w should we pick so that

$$\frac{1}{|V|} \sum_{v \in V} f(v) \sim \sum_{w \in W} a_w f(w)$$

for functions $f : V \to \mathbb{R}$ that are "smooth with respect to the geometry" of G? Here, $\sim$ indicates that we want the right-hand side to be as close to the left-hand side as possible. More precisely, how should we pick the set of vertices $W \subset V$ and the weights a_w such that

$$\max_{f \in X} \left| \frac{1}{|V|} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \to \min,$$

where $X \subset L^2(V)$ denotes a subspace of functions $f : V \to \mathbb{R}$ (the space of "smooth" functions that is yet to be defined)?

We emphasize that, just as in the continuous case, there may be more than one way of defining a smoothness class; we will define one further below but emphasize that many others are conceivable.
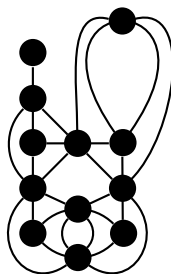

FIGURE 2. Suppose this is a social network: people who are similar to each other are connected by an edge. I would like to discover whether people prefer spaghetti or pizza and assume that my existing knowledge of similarity extends to tastes in food: I am only allowed to ask 3 people (because I have to pay them and have a limited budget). Who do I ask?

1.1. **Example applications.** At first glance, the question seems to not even make a lot of sense: since the graph is finite, one can simply compute the true average of f by summing over all n vertices. This question is only interesting whenever sampling f in a vertex is difficult or expensive (see Figure 2). We believe that in the light of new challenges in data science, the question is far from artificial and of *profound* relevance as opposed to merely an artificial extension of the classical numerical integration problem. A toy example is the following (that was relevant in work of the first author [17] and partially inspired this paper): suppose we have a medical database of $n \gg 1$ people containing all sorts of information and are interested in the average blood pressure (information not contained in the database) of those n people. Actually going out and measuring the blood pressure of all n people would take a very long time. However, it is known that blood pressure is strongly correlated with some of the factors we have on file (say, age, weight, and smoking habits) and weakly connected or unconnected to others (say, eye color). We can then build a weighted graph on n vertices where the weight on the edge connecting two people depends on how similar they are with regard to relevant factors – the hope is that blood pressure, as a function on the graph, is then smoothly varying. Which of the, say, $n/1000$ people should we do a blood pressure measurement on so that the sample average is representative of the global average? Is there a way to pick them in a way that decreases the expected error over a pure random selection? It is likely possible to make stronger statements if one restricts to certain classes of graphs and functions, and this could be of substantial interest. Our paper will only address the most general case.

1.2. **Related results.** Our paper is a companion paper to [31] (itself inspired by [30]) dealing with a problem in spectral graph theory that discusses, in the language of numerical integration, the analogue of t-designs on finite graphs. However, the

focus of [31] is more of an algebraic type and does not discuss the problem of numerical integration on generic graphs (moreover, the approach in [31] is unlikely to work on graphs that do not exhibit a great degree of structure). We extend some of the ideas from [30, 31] to sampling, prove an inequality bounding the integration error in terms of the geometry of the sampling points, and give several examples. As discussed above, any reasonable approach to numerical integration on graphs has to necessarily yield a corresponding reasonable scheme on Euclidean domains. The approach we discuss in this paper has been shown to have a natural analogue in Euclidean space in work of J. Lu, M. Sachs, and the second author [18] (where, at least on the torus, it improves on classical constructions with respect to a Fourier-analytic way of measuring the integration error). We also emphasize a paper of Pesenson, Pesenson, and Führ [26], which deals with cubature formulas on combinatorial graphs by enforcing exactness on bandlimited functions and a paper of Pesenson and Geller [25] for the continuous analogue.

2. Setup and main results

2.1. **Formal setup.** We will now make the notion of smoothness precise. If f has no particular structure, then there is little hope of being able to achieve anything at all. However, a graph does induce a natural notion of smoothness: we want the function to vary little between nodes that are "well connected" (e.g., connected by edges with large weight or have many neighbors that connect to each other). These are nodes that are "very similar", and hence we expect the function to vary less between these nodes than between nodes that are less similar. We now introduce a notion of a Laplacian L on a graph: more precisely, we define a linear operator that maps functions $f : V \to \mathbb{R}$ to functions $Lf : V \to \mathbb{R}$ and is given as a local averaging operation. Our notion of a Laplacian deviates slightly from more classical notions and could be of independent interest (see Section 2.4 for a more detailed discussion). Let A denote the (weighted) adjacency matrix of G,

$$A_{ij} = \begin{cases} w(e_{ij}) & \text{if } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where $w(e_{ij}) \geq 0$ is the weight of the edge $e_{ij} = e_{ji}$ connecting vertices i and j. A is a symmetric matrix. For simplicity, we assume that there are no loops, so the diagonal entries of A are zero. We introduce the maximum sum of any of the rows of this symmetric matrix (coinciding, by symmetry, with the maximum sum of any of the columns) and use it to define a normalized adjacency matrix: more precisely, we have

$$d_{\max} = \max_{1 \leq i \leq n} \sum_{j=1}^{n} A_{ij} \qquad \text{and set} \qquad A' = \frac{1}{d_{\max}} A.$$

Finally, we introduce the (diagonal) degree matrix D' associated to the renormalized adjacency matrix and use it to define a Laplacian: we set

$$D'_{ii} = \frac{1}{d_{\max}} \sum_{j=1}^{n} A_{ij} \quad \text{and define the Laplacian as} \quad L = A' - D'.$$

We will never work directly with the Laplacian: our main object of interest is the associated diffusion process whose generator is given by

$$P = L + \mathrm{Id}_{n \times n},$$

where $\text{Id}_{n \times n}$ is the identity matrix of size n. P is a symmetric stochastic matrix and represents a lazy random walk where the probability of "staying put" depends on the vertex (as opposed to being, say, 0.5 as in the classical lazy random walk). We denote the eigenvalues of P, which are merely the eigenvalues of the Laplacian L shifted by 1, by $\lambda_1, \dots, \lambda_n$. Since P is a stochastic matrix, we have $|\lambda_i| \leq 1$. The eigenvectors whose eigenvalues are close to 0 "diffuse quickly" and are thus the natural high-frequency objects on the graph. This motivates an ordering of eigenvalues from low frequency to high frequency:

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n \geq 0.$$

We denote the corresponding orthogonal eigenvectors of P, which clearly coincide with the eigenvectors of the Laplacian L, by $\phi_1, \dots, \phi_n \in \mathbb{R}^n$ (normalized in $L_2(V)$, $\phi_1 = 1/\sqrt{n}$ is the constant vector). We define a function space X_λ, the canonical analogue of trigonometric polynomials on the torus $\mathbb{T}^d$ or spherical harmonics on the sphere $\mathbb{S}^d$, via

$$X_\lambda = \left\{ f : V \to \mathbb{R} \,\middle|\, f = \sum_{|\lambda_k| \geq \lambda} \langle f, \phi_k \rangle_{\ell^2} \right\} \quad \text{with norm} \quad \|f\|_{X_\lambda}^2 = \sum_{|\lambda_k| \geq \lambda} |\langle f, \phi_k \rangle_{\ell^2}|^2,$$

where $0 \leq \lambda \leq 1$ is a parameter controlling the degree of smoothness. If $\lambda > \mu$, then $X_\lambda \subset X_\mu$. Moreover, $X_0 = \{f : V \to \mathbb{R}\}$ contains all functions while, at least on generic graphs, X_1 contains only the constant functions (depends on whether the induced random walk is ergodic; this is not important for our purposes). The norm is just the classical L^2-norm on the subspace. In more classical terms, we are simply considering the L^2-space obtained via a Littlewood-Paley projection. This can be considered as the discrete analogue of the Paley-Wiener space of band-limited functions. To the best of our knowledge this space was first introduced in the discrete setting by Pesenson [21]. We also refer to the emerging theory around this notion in [11, 12, 19, 20, 22–24, 33–35].

This function space X_λ is natural: if the graph G approximates a torus $\mathbb{T}^d$ (being close to a grid graph), then this function space will indeed approximate trigonometric functions: graph Laplacians are known to converge spectrally when graphs approximate an underlying manifold [28]. If G is close to a discretization of a sphere $\mathbb{S}^d$, then the space X_λ approximates the space of low-degree spherical harmonics. However, as is not surprising, the precise definition of "smoothness" of the function is crucial here, and different notions of smoothness will lead to different versions of the numerical integration problem. We state this as an explicit

Open Problem. Investigate other notions of smoothness for functions $f : V \to \mathbb{R}$ and the arising Numerical Integration problem.

While our particular definition of smoothness has the advantage of naturally reducing to a spectral perspective on numerical integration of continuous functions, it is quite conceivable that many other notions may exist and have various types of advantages and disadvantages. We believe this to be a natural and fascinating question that could lead to rather interesting questions.

2.2. **Main result.** Our main result bounds the integration error in terms of $\|f\|_{X_\lambda}$ and a purely geometric quantity (explained in detail below) formulated in terms of the quadrature points and independent of f. This has the nice effect of multiplicatively separating the size of the function $\|f\|_{X_\lambda}$ and a quantity that can be

interpreted as the "quality" of the quadrature scheme. We use, for $v \in V$, the notation δ_v to denote the function $\delta_v : V \to \mathbb{R}$ for which $\delta_v(w) = \delta_{vw}$, where δ is the Kronecker delta function (i.e., $\delta_v(w) = 1$ if $w = v$ and $\delta_v(w) = 0$ otherwise).

Theorem 1 (Main result)**.** *Let $W \subset V$ be equipped with weights a_w summing to 1. Then, for all $\ell \in \mathbb{N}$ and all $0 < \lambda < 1$,*

$$\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \leq \frac{\|f\|_{X_\lambda}}{\lambda^\ell} \left(\left\| (\mathrm{Id}_{n \times n} + L)^\ell \sum_{w \in W} a_w \delta_w \right\|_{L^2}^2 - \frac{1}{n} \right)^{\frac{1}{2}}$$
$$\forall \, f \in X_\lambda.$$

The proof shows that the inequality is essentially close to sharp in various settings (see below). If one has access to the eigenfunctions of the Laplacian (eigenvectors of the Laplacian matrix), then it is possible to use a stronger inequality discussed below in Section 2.5. The advantage of the main theorem is that the upper bound can be computed without knowing anything about the eigenfunctions of the Laplacian. The result has a certain philosophical similarity to the Koksma-Hlawka inequality: its simplest setting states that for functions $f : [0,1]^d \to \mathbb{R}$,

$$\left| \int_{[0,1]^d} f(x)dx - \frac{1}{N} \sum_{k=1}^{N} f(x_k) \right| \leq V(f) D_N^*(x_k),$$

where $V(f)$ is the variation of the function in the sense of Hardy and Krause and D_N^* denotes the star-discrepancy of the set of points $(x_k)_{k=1}^{N}$. We refer to the standard textbooks [7, 8, 15] for a definition of these terms: the main point is that the error is the *product* of an error coming from a notion of smoothness of the function $V(f)$ and a notion of regularity of the sampling points $D_N^*(x_k)$. If one knows nothing about the function except the degree with which it oscillates, it is nonetheless wise to minimize the error that is within one's control and to try to make $D_N^*(x_k)$ as small as possible. Our main result is of a very similar flavor, the error being *multiplicative*. As in the case of the Koksma-Hlawka inequality, the proof shows that the result is close to sharp, which has the advantage of defining a notion of *discrepancy* (regularity of sampling points and weights) on finite graphs. The study of low-discrepancy sets on the torus has been of classical interest (see [7, 8, 15]) and has intimate connections to analytic number theory, combinatorics, and harmonic analysis. We believe that the notion of low-discrepancy sets on graphs might be an interesting avenue of further research.

2.3. **Geometric interpretation: Heat ball packing.** Altogether, this suggests that we should use the quantity on the right-hand side in the inequality in Theorem 1, depending only on the set W, the weights a_w, and the free parameter ℓ (but not on the function f) as a guideline for how to construct the quadrature rule. This motivates studying the minimization problem

$$\min_{\substack{W \subset V \\ |W|=k}} \min_{a_w} \left\| (\mathrm{Id}_{n \times n} + L)^\ell \sum_{w \in W} a_w \delta_w \right\|_{L^2}^2 \qquad \text{subject to } \sum_{w \in W} a_w = 1.$$

Note that for practical applications we do not necessarily need to find the precise minimizer of the solution; indeed, we just want the quantity to be as small as possible. Of course, from a mathematical point of view, it is very desirable to

actually find global minima and study their properties. The quantity $(\mathrm{Id}_{n \times n} + L)^{\ell} \delta_w$ is the probability distribution of a random walker starting in w after ℓ jumps, which lends itself to a geometric interpretation.

> **Guideline.** If we manage to find a placement of vertices W with the property that the random walks, weighted by a_w, overlap very little, then we have found a good quadrature rule on the graph $G = (V, E)$.

We observe that this can be reinterpreted as a "packing problem for heat balls". This principle has already appeared naturally in the continuous setting of Riemannian manifolds in work of the second author [30]. It is even meaningful if we can only sample in a single vertex: we should pick the most central vertex and that is equivalent to heat diffusing quickly to many other points. In the ideal setting with perfectly distributed heat balls all the weights would be identical (as can be seen in examples with lots of symmetry; see [31]). We summarize that

(1) it is desirable to construct $W \subset V$ equipped with weights a_w such that random walks, starting in $w \in W$ and weighted by a_w, intersect each other as little as possible.

(2) if we are *not* allowed to choose W, we can still use the procedure above to find weights that yield a better result than weighting each vertex equally.

We emphasize that we do not attempt to solve the heat ball packing problem here, nor do we expect it to be easily solvable at this level of generality. The main contribution of this paper is to introduce the heat ball packing problem as a fundamental issue with implications for sampling.

Problem. How does one find effective heat ball packing configurations quickly? Are there algorithms leading to effective almost-minimizing configurations? What theoretical guarantees can be proven?

If the minimization problem over heat balls is similar to classical problems in potential theory (see also Section 3.4), where one places points in such a way that

$$\sum_{i,j=1}^{n} \frac{1}{d(x_i, x_j)^{\alpha}} \to \min,$$

where $d(\cdot, \cdot)$ is the distance on a manifold (the case of ball packing corresponding to the limit $\alpha \to \infty$), then this would suggest that there will be many local minimizers that are not global: this has been empirically observed in the continuous setting (see, e.g., [9]). Moreover, if the continuous analogies continue to hold, then the actual numerical value of minimizing configuration can be expected to be quite close to that of almost-minimizers, which is good news: there are many almost-minimizers (thus easy to find) and they are almost as good as the global minimizer. At this point, however, this is pure speculation.

2.4. **Another interpretation.** There is a nice bit of intuition coming from a more abstract perspective: the reason why the above method works is that the heat propagator $e^{t\Delta}$ is self-adjoint. More precisely, let us assume (M, g) is a compact manifold normalized to $\mathrm{vol}(M) = 1$ and let μ be a probability measure (our quadrature rule) on it. Smooth functions f have the property that they do not change substantially if we apply the heat propagator for a short time $t > 0$ (this is one way of quantifying

smoothness), and therefore

$$\langle f, \mu \rangle \sim \langle e^{t\Delta} f, \mu \rangle = \langle f, e^{t\Delta} \mu \rangle.$$

Here, the $\sim$ is hiding an error term that depends on the smoothness of the function f and the size of time t. Various ways of making it precise lead to various different elementary results that we do not further pursue here. In order for this quadrature rule to be effective, we want $e^{t\Delta}\mu$ to be close to the Lebesgue measure dx. However, the heat propagator preserves the L^1-mass and thus

$$1 = \|\mu\| = \|e^{t\Delta}\mu\|_{L^1} \leq \|e^{t\Delta}\mu\|_{L^2}.$$

The Cauchy-Schwarz inequality is only sharp if $e^{t\Delta}\mu$ coincides with the constant function, and thus minimizing $\|e^{t\Delta}\mu\|_{L^2}$ is a natural way to obtain good quadrature rules. If μ is a weighted sum of Dirac measures, then $e^{t\Delta}$ turns this, roughly, into a sum of Gaussians ("heat balls"), and finding locations to minimize the L^2-norm becomes, essentially, a packing problem. We observe that the argument does not single out L^2 and minimizing $\|e^{t\Delta}\mu\|_{L^p}$ for $p > 1$ will lead to a very similar phenomenon. It remains to be seen whether there is any advantage to that perspective since L^2 is usually easiest to deal with in practice.

2.5. **The Laplacian: A second method.** We believe that our notion of Laplacian could be useful in practice (see also [16]). It combines the desirable properties of

  (1) having a symmetric matrix (and thus orthogonal eigenvectors),
  (2) inducing a diffusion operator (that is, a time-discrete heat semigroup generated by $\text{Id} + L$) that preserves the mean value of the function,
  (3) and the propagator $\text{Id} + L$ having only nonnegative entries.

The Kirchhoff matrix $L_1 = D - A$ has the first two properties but not the third; the normalized Laplacian $L_2 = \text{Id}_{n \times n} - D^{-1/2} A D^{-1/2}$ does not induce a diffusion operator that preserves the mean value of the function. We believe that, for this reason alone, our notion of a Laplacian could be useful in other settings as well. If we can compute eigenvectors, then there is a way of approaching the problem directly.

Proposition. *Let $W \subset V$ be equipped with weights a_w summing to 1. Then, for all $0 < \lambda < 1$,*

$$\sup_{\substack{f \in X_\lambda \\ f \neq 0}} \frac{1}{\|f\|_{X_\lambda}} \left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| = \left\| \frac{1}{n} - \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda}.$$

This statement follows easily from L^2-duality and will be obtained in the proof of the Theorem as a by-product. The result is true in general, but, of course, one cannot compute the quantity on the right-hand side (or, more generally, the size of $\|f\|_{X_\lambda}$) unless one has access to the eigenvectors of the Laplacian. If one has indeed access to either the Laplacian eigenfunctions or at least some of the leading eigenvectors $\phi_1, \ldots, \phi_k$ where $k \ll n$ is chosen so that $\lambda_k \sim \lambda$ (simpler to obtain in practice), then optimizing the functional in $W \subset V$ and weights a_w is essentially equivalent to finding good quadrature points. This simple observation is very effective in practice; we refer to numerical examples below.

3. How to use the theorem

The result discussed above has the nice effect of cleanly separating the problem of numerical integration on a graph from the actual graph structure: the geometry of the graph is encoded implicitly in the geometry of the random walk. This has the nice effect of providing a uniform treatment but, as a downside, does not provide an immediate method on how to proceed in particular instances. The purpose of this section is to comment on various aspects of the problem and discuss approaches. Recall that our main result can be written as

$$
\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \leq \|f\|_{X_\lambda} \min_{\ell \in \mathbb{N}} \frac{1}{\lambda^\ell} \left(\left\| (\mathrm{Id}_{n \times n} + L)^\ell \sum_{w \in W} a_w \delta_w \right\|_{L^2}^2 - \frac{1}{n} \right)^{\frac{1}{2}}
$$
$$
\forall\, f \in X_\lambda.
$$

3.1. **The parameters λ and ℓ.** A priori we have no knowledge about the degree of smoothness of the function and have no control over it. The parameter ℓ, on the other hand, is quite important and has a nontrivial impact on the minimizing energy configurations for the quantity on the right-hand side. In practice, we have to choose ℓ without knowing λ (fixing an allowed number of sample points implicitly fixes a scale for λ). We propose the following basic heuristic.

> **Heuristic.** If ℓ is too small, then there is not enough diffusion and heat balls interact strongly with themselves. If ℓ is too large, the exponentially increasing weight $\lambda^{-\ell}$ is too large. There is an intermediate regime when heat balls start to interact with nearby heat balls.

The heuristic is accurate if the graph is locally (at the scale of typical distance between elements of W) close to Euclidean (in the sense that it is given as the discretization of a Euclidean domain as discussed in the Introduction). On general graphs, the optimal scale of ℓ might be more nontrivial to estimate: we have observed that, in practice, there is a wide range of ℓ yielding good results.

3.2. **The placement of W.** Naturally, to avoid intersections of random walkers, we want to place the elements of W as far from each other as possible. In particular, if the graph is close to having a Euclidean structure, we would expect fairly equispaced points to do well. A method that was used in [31] is to start with a random set of k vertices $\{v_1, v_2, \ldots, v_k\}$ and compute the

$$
\text{total mutual distance} = \sum_{i,j=1}^{k} d(v_i, v_j),
$$

where d is a metric on $G = (V, E)$. The algorithm then goes through all the vertices and checks whether moving one of them to a neighboring vertex increases the total mutual distance and, if so, moves the vertex. This is repeated as long as possible. The simple numerical examples in [31] all have edges with equal weight, and the standard combinatorial graph distance can be used. Which notion of "total mutual distance" leads to the best result could strongly depend on the type of graph under consideration. There is a fairly natural reason why an algorithm of this type has the ability to successfully produce sets of vertices that are very well spread out. We quickly return to the sphere $\mathbb{S}^d$, where a particularly spectacular justification exists.

Let σ be the normalized measure on $\mathbb{S}^d$. Then, for any set $X = \{x_1, \ldots, x_n\} \subset \mathbb{S}^d$, we have Stolarsky's invariance principle [1, 32]

$$\int_{\mathbb{S}^d} \int_{\mathbb{S}^d} \|x - y\| d\sigma(x) d\sigma(y) - \frac{1}{n^2} \sum_{i,j=1}^{n} \|x_i - x_j\| = c_d \left[D_{L^2, \mathrm{cap}}(X) \right]^2,$$

where the quantity on the right is the L^2-based spherical cap discrepancy and c_d is a constant only depending on the dimension. The L^2-based spherical cap discrepancy is a measure that has been studied in its own right: if the points are evenly distributed, then it is small. This may be a somewhat peculiar case. However, it is not difficult to see that on fairly generic manifolds functionals along the lines of

$$\sum_{i,j=1}^{n} \frac{1}{d(x_i, x_j)^\alpha} \to \min \qquad \text{or} \qquad \sum_{i,j=1}^{n} e^{-\alpha d(x_i, x_j)} \to \min$$

converge to the uniform distributions if the number of points is large. Moreover, and this is particularly useful, these types of functionals tend to produce minimal energy configurations that only weakly depend on the functional being used. On two-dimensional manifolds, the hexagonal lattice seems to be particularly universal (see [2]). We do not know what kind of interaction functional is optimal on graphs. In practice, one would like to have fast and reliable algorithms that scale well, and this seems like a problem of substantial interest.

3.3. **The weights a_w.** Once we are given a set $W \subset V$ and a parameter ℓ, the optimization of the weights is completely straightforward. Observe that

$$\left\| (\mathrm{Id}_{n \times n} + L)^\ell \sum_{w \in W} a_w \delta_w \right\|_{L^2}^2 = \sum_{w_1, w_2 \in W} a_{w_1} a_{w_2} \left\langle (\mathrm{Id}_{n \times n} + L)^\ell \delta_{w_1}, (\mathrm{Id}_{n \times n} + L)^\ell \delta_{w_2} \right\rangle.$$

This is merely a quadratic form indexed by a $|W| \times |W|$ matrix; we thus need to solve the semidefinite program

$$\sum_{w_1, w_2 \in W} a_{w_1} a_{w_2} \left\langle (\mathrm{Id}_{n \times n} + L)^\ell \delta_{w_1}, (\mathrm{Id}_{n \times n} + L)^\ell \delta_{w_2} \right\rangle \to \min$$

$$\text{subject to } \sum_{w \in W} a_w = 1.$$

These weights a_w play an important role in further fine-tuning an existing set of vertices $W \subset V$. This is the reason why minimizing the functional can be used to find appropriate weights for *any* given set of vertices W: if two vertices in W happen to be somewhat close to each other, then the quadratic form will take this into account when distributing the weights. Conversely, if one of the vertices in W is surprisingly isolated from the other vertices in W, then the quadratic form will increase the weight assigned to that vertex. This is exactly how things should be: points that are oversampling a region in the graph should be given a smaller weight, whereas isolated vertices cover a wider range and are, correspondingly, more important. We refer to Section 4 for numerical examples illustrating this point.

3.4. **Related results.** Sampling on graphs is a fundamental problem, and a variety of approaches have been discussed in the literature [13]. Sampling is usually done for the purpose of compression or visualization and not numerical integration (in particular, vertices are usually not equipped with weights).
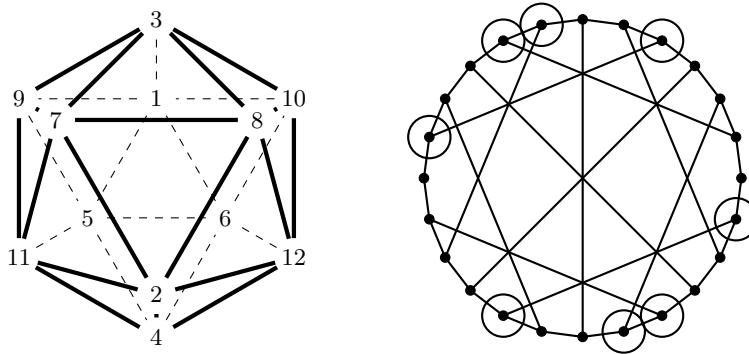

FIGURE 3. (Left) The icosahedron integrates all polynomials up to degree 5 on $\mathbb{S}^2$ exactly (this space is 36-dimensional). (Right) a subset of 8 vertices integrates 21 of 24 eigenvectors of the McGee graph exactly. Such examples require extraordinary amounts of symmetry and are not generic (from [31]).

Our approach seems to be very different from anything that has been proposed. The closest construction seems to be a line of research using biased random walks [10, 14, 27] based on the idea of sending random walkers and accepting the vertices they traverse with certain biased probabilities as sampling points. In contrast, we select points so that random walkers starting there avoid each other. Other results seem related in spirit [3]. Our approach is motivated by a recent approach [30] to the study of spherical t-designs of Delsarte, Goethals, and Seidel [6] (and Sobolev [29]). These are sets of points defined on $\mathbb{S}^d$ with the property that they integrate a large number of low-degree polynomials exactly (we refer to a survey of Brauchart and Grabner) [4]. The second author recently extended some of these results to weighted points and general manifolds [30]. These ideas are shown to be an effective source of good quadrature points in the Euclidean setting in a paper of Lu, Sachs, and the second author [18].

The second author recently proved [31] a generalized Delsarte-Goethals-Seidel bound for graph designs (the analogue of spherical t-designs on combinatorial graphs). The main condition in that paper is algebraic (exact integration of a certain number of Laplacian eigenvectors) as opposed to quantitative (small integration error). [31] shows a number of truly remarkable quadrature rules on highly structured graphs that were found by numerical search: one of these rules is depicted in Figure 3 and manages with only 8 evaluations to integrate 21 out of a total of 24 eigenvectors exactly. However, these examples are very nongeneric, the consequence of strong underlying symmetries and not likely to be representative of what can be achieved in a typical setting.

4. NUMERICAL EXAMPLES

4.1. **Importance of weights.** We start with a toy example shown in Figure 4 to illustrate the importance of the weights to counterbalance bad geometric distributions. We assume we are given two clusters and an uneven distribution of six sampling points: five end up in one cluster, while the other cluster contains a total of five points. We construct a graph based on nearest neighbor distances weighted with a Gaussian kernel. The spreading of heat is rather uneven if all points are given equal weight. Conversely, by adjusting the weight to minimize the L^2-norm (constraint to summing to 1 and being nonnegative), a much more balanced distribution is achieved.
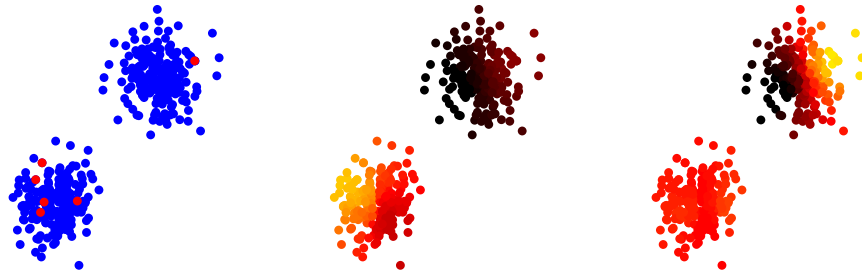

FIGURE 4. Six points selected unevenly in two clusters (left), the heat flow emanating from weighting them all evenly (middle), and the optimal coefficients (0.11, 0.24, 0.05, 0.00, 0.18, and 0.43) for the heat ball packing problem (right).

The weights show that, in particular, one point is given weight 0 and another point is given a rather small weight (0.05). This is to counterbalance the clustering of points. The isolated point is given almost half the weight (0.43). We see that the heat distribution in the second cluster is still highly uneven: this shows that it would be preferable to pick another point, since the single sampling point is actually quite far from the center of the cluster; if it was closer to the center, it would have received an even larger weight.
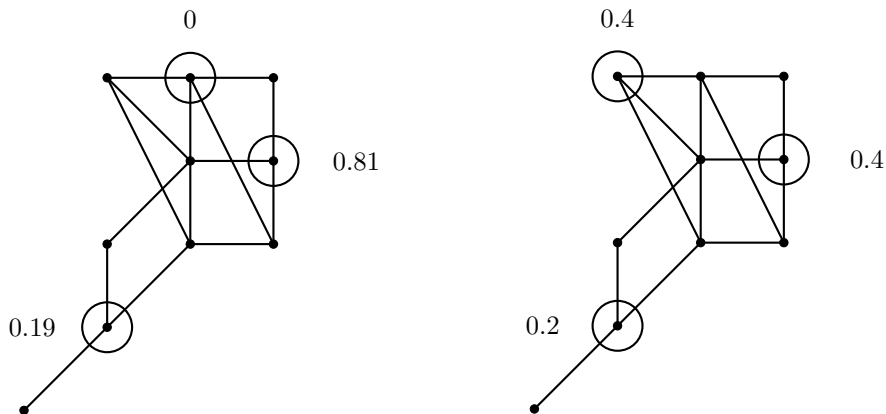

FIGURE 5. Two optimal configurations for $\ell = 3$ on 3 vertices.

Figure 5 shows an example on a small graph with 10 vertices and 15 edges. More precisely, we optimize $\|(\mathrm{Id}_{10\times 10} + L)^3 \sum_w a_w \delta_w\|_{L^2}$ over all sets W with three vertices. In the first example, we see that there is one very central node that distributes well throughout the network; another weight is actually set to 0. This is a consequence of constraining the optimization to nonnegative weights $a_w \geq 0$. This constraint is not required by the Theorem but makes the optimization easier and is well-motivated by classical methods in numerical integration. If we move the point that was assigned weight 0, then the weight splits evenly (the value of the functional barely changes).

4.2. **MNIST.** Our explicit example is as follows: we consider the data set MNIST, a collection of handwritten digits represented as 28×28 pixels. For simplicity, we only consider the subset comprised of the digits 0 and 1, resulting in a total of 12,665 images. The problem is to determine the proportion of elements in the set that are handwritten digits that are 1's (which is 0.53). This ties in to our example in the beginning: suppose we did not know the precise proportion of 1's and the data is unlabeled. Labeling the data is expensive: the function evaluation would be one human being looking at a picture and labeling it, which is costly. However,
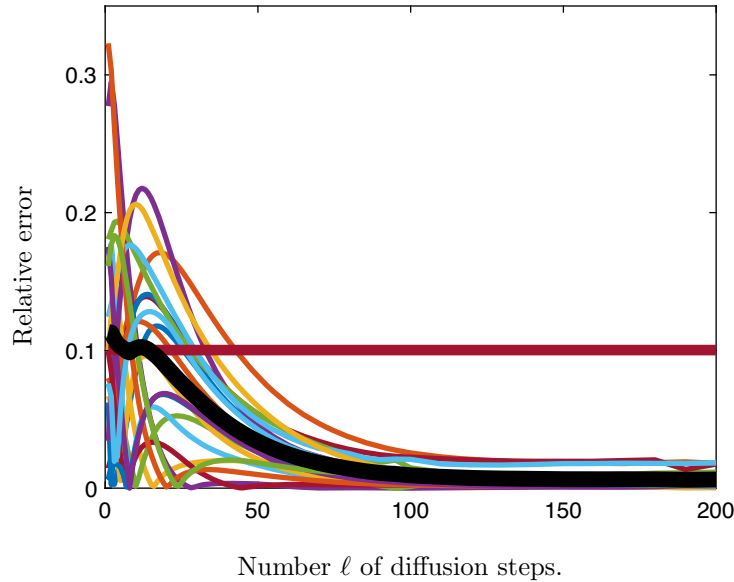

Number ℓ of diffusion steps.

FIGURE 6. Estimating digits in MNIST: the relative integration error for 20 different sets of points of size 50 and how it evolves as a function depending on ℓ (leading to different selection of weights). The average of these 20 curves is shown as the bold black line; sampling error for randomly chosen points is the red line.

these pictures are merely $\{0,1\}$-vectors in $\mathbb{R}^{784}$. As is commonly done, we reduce the dimensionality of the data by projecting onto its first ten principal components. We build a graph by connecting every element to its ten nearest neighbors (in Euclidean distance) weighted with a Gaussian kernel and then symmetrize by averaging the resulting adjacency matrix with its transpose. It is reasonable to assume that the

indicator function of 1's is smooth over a graph defined by that notion of distance: handwritten digits looking like a 1 should be close to other handwritten digits that look like 1. We then proceed as outlined above: we sample random points, move them iteratively so that they are far away from each other, and then adjust weights by solving the semidefinite program. In Figure 6, the result is plotted against the parameter ℓ and compared to uniform weights on random points (red); the picture shows 20 different sets of points, the evolution of their integration error depending on ℓ as well as their average (black). We observe that for the right parameter range of ℓ, the obtained numerical integration scheme performs much better, but the precise performance depends on the points chosen. This highlights the need for fast, stable, and guaranteed ways of approximately solving the heat ball packing problem.

4.3. **Using eigenvectors of the Laplacian.** This section studies the same example as above, estimating the proportion of handwritten digits "1" in MNIST, but assumes additionally that we are able to access the eigenvectors of the Laplacian associated to the largest few eigenvalues exactly. We set $\lambda = 0.994$ close to 1, leading to a space X_λ spanned by very few of the smoothest eigenvectors, sample random points, move them far apart, and make use of

$$\sup_{\substack{f \in X_\lambda \\ f \neq 0}} \frac{1}{\|f\|_{X_\lambda}} \left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| = \left\| -\frac{1}{n} + \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda}$$

to explicitly optimize the weights. Our method (blue) is shown to perform ex-
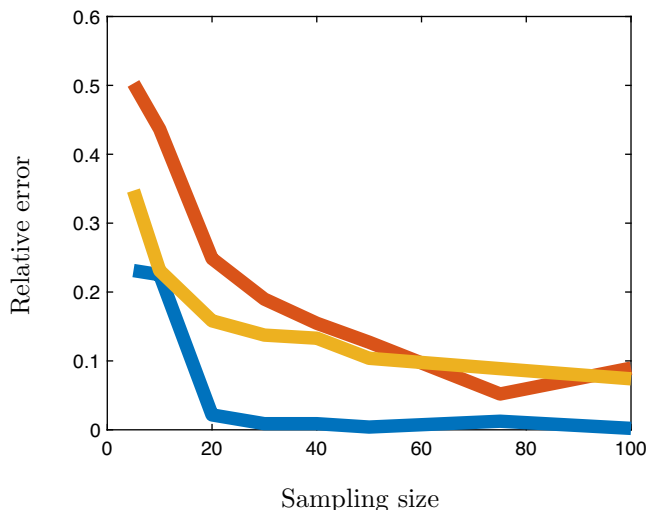

FIGURE 7. Numerical Integration with access to eigenvectors: our method (blue) compared to sampling in random points (yellow) and sampling in the location in which we move the points (red).

ceedingly well (Figure 7); the weights are crucial; sampling over the same points (red) is even worse than sampling over random points (yellow), which decays as (#size of subset)$^{-1/2}$ (averaged over 200 random samples). As mentioned above,

having direct access to the eigenvectors implies that the bound is often close to sharp. This is illustrated in the subsequent Figure 8, where we replace the indicator function of the "1"-digits in the subset of MNIST comprised of digits 0 and 1 by its mollification obtained from projecting onto the first 6 eigenvectors. We optimize in the weights and obtain

$$\left\| \frac{1}{n} - \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda} \qquad \text{as an upper bound (red)}$$

that is then compared to the error on the function f (blue). We observe that as soon as the sample size exceeds a certain limit, integration becomes exact. It is quite desirable to obtain a better understanding of the interplay of parameters involved: suppose we are given a set of k well-distributed points and optimize the weights so as to minimize $\left\| -n^{-1} + \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda}$. What is the interplay between k, the X_λ space, and the performance of the arising quadrature rule? Or, put differently, how does the integration error in an X_μ space depend on the space X_λ that was used to determine the weights? This question is clearly of great relevance in applications.
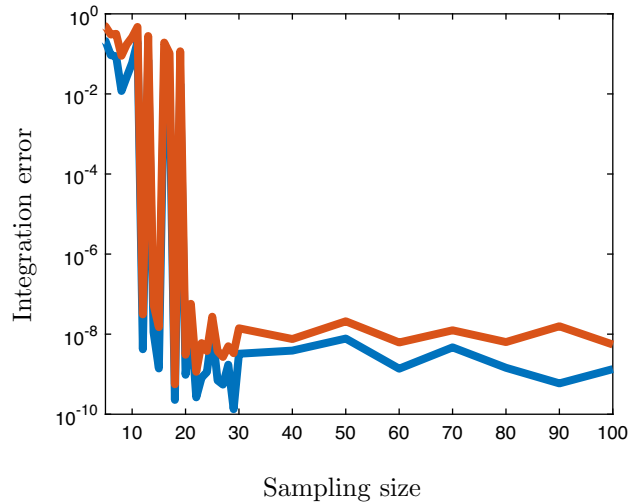

FIGURE 8. Error for the smoothed indicator function on the 1's (blue) and the theoretical upper bound on the integration error (red). It depends strongly on W and, after some initial fluctuation, settles down to essentially exact integration. The theoretical upper bound matches the performance on the particular instance.

The phenomenon, which seems to be generic and easily observed in most examples, is illustrated in Figure 9. We downsampled the MNIST dataset of digits "0" and "1" to consist of a total of 1000 points and constructed the graph as before. We then choose a subset of points W of size 100, increase their mutual distance, and fix them as quadrature points. Finally, we optimize their weights in three different X_λ spaces where λ is chosen such that the dimensions of the spaces are 10, 20, and 25 (i.e., they contain the first 10, 20, and 25 eigenfunctions, respectively). We then
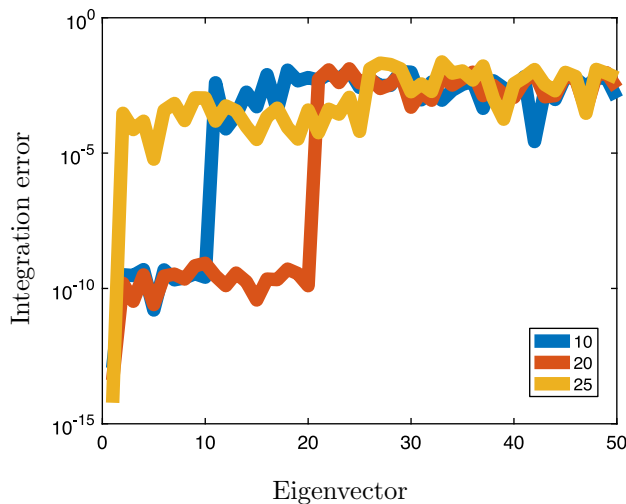
FIGURE 9. Integration error of three quadrature rules with weights fine-tuned in three different X_λ-spaces on the first 50 eigenfunctions.

plot the integration error of these three quadrature rules on the first 50 eigenfunctions. If we optimize the weights according to the X_λ space containing the first 10 eigenfunctions, then the first 10 eigenfunctions are essentially integrated exactly; the subsequent integration error is small. The same is true for optimization in the space containing the first 20 eigenfunctions. Then the behavior changes abruptly: if we optimize over the first 25 eigenfunctions, then the error on those 25 eigenfunctions is small ($\sim 10^{-4}$) and, as in the other examples, increases afterwards. This seems to be typical: for any given set $W \subset V$, there seems to be a range of X_λ-spaces such that optimizing parameters leads to exact integration in X_λ. Once their dimension exceeds a certain (sharp) threshold, the error is still small in X_λ but many orders of magnitude larger than before. This sharp phase transition could serve as another measure of quality of W that may be useful in judging the quality of algorithms finding W (the largest number of eigenfunctions that can be integrated exactly using W for some choice of weights, a measure already studied in [31]).

5. PROOF OF THE THEOREM

Proof. We write the integration error as the inner product of two vectors and decompose it as

$$\frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) = \sum_{v \in V} \left(\frac{1}{n} - \sum_{w \in W} a_w \delta_w(v) \right) f(v)$$
$$= \sum_{k=1}^{n} \left\langle \frac{1}{n} - \sum_{w \in W} a_w \delta_w, \phi_k \right\rangle \langle f, \phi_k \rangle.$$

Since $f \in X_\lambda$, we have that $\langle f, \phi_k \rangle = 0$ unless $|\lambda_k| \geq \lambda$. A simple application of the Cauchy-Schwarz inequality then shows that

$$\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \leq \left\| \frac{1}{n} - \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda} \|f\|_{X_\lambda}.$$

More precisely, L^2-duality implies that this step is not lossy since

$$\sup_{\substack{f \in X_\lambda \\ f \neq 0}} \frac{1}{\|f\|_{X_\lambda}} \left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| = \left\| \frac{1}{n} - \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda}.$$

For any function $g : V \to \mathbb{R}$, we have

$$\|g\|_{X_\lambda}^2 = \sum_{|\lambda_k| \geq \lambda} |\langle g, \phi_k \rangle|^2 \leq \frac{1}{\lambda^{2\ell}} \sum_{k=1}^{n} |\lambda_k|^{2\ell} |\langle g, \phi_k \rangle|^2$$

$$= \frac{1}{\lambda^{2\ell}} \left\| \sum_{k=1}^{n} \lambda_k^\ell \langle g, \phi_k \rangle \phi_k \right\|_{L^2}^2$$

$$= \frac{1}{\lambda^{2\ell}} \left\| (\mathrm{Id}_{n \times n} + L)^\ell g \right\|_{L^2}^2.$$

We observe that this inequality is also valid if $g \notin X_\lambda$ since every step is a valid bound from above and $\|\cdot\|_{X_\lambda}$ is defined on all functions as a seminorm (we note, however, that if $g \notin X_\lambda$, then the inequality will usually be far from sharp). We use this inequality for

$$g = \frac{1}{n} - \sum_{w \in W} a_w \delta_w$$

to conclude that

$$\left\| \frac{1}{n} - \sum_{w \in W} a_w \delta_w \right\|_{X_\lambda} \|f\|_{X_\lambda} \leq \frac{\|f\|_{X_\lambda}}{\lambda^\ell} \left\| (\mathrm{Id}_{n \times n} + L)^\ell \left(\frac{1}{n} - \sum_{w \in W} a_w \delta_w \right) \right\|_{L^2}.$$

We observe that $\mathrm{Id}_{n \times n} + L$ is the generator of the diffusion. It is a linear operator for which constant functions are invariants. This implies that

$$(\mathrm{Id}_{n \times n} + L)^\ell \left(\frac{1}{n} - \sum_{w \in W} a_w \delta_w \right) = \frac{1}{n} - (\mathrm{Id}_{n \times n} + L)^\ell \sum_{w \in W} a_w \delta_w.$$

We now show that the operator $(\mathrm{Id}_{n \times n} + L)$ preserves the average value of the function. It suffices to show that L maps the average value of a function to 0. We use the definition

$$L = A' - D' = \frac{1}{d_{\max}} (A - D).$$

It thus suffices to show that $A - D$ maps every function to a function with mean value 0. This follows from changing the order of summation

$$\sum_{i \in V} (Af)(i) = \sum_{i \in V} \sum_{j \in V} e_{ij} f(j) = \sum_{j \in V} \sum_{i \in V} e_{ij} f(j) = \sum_{i \in V} D_{ii} f(i) = \sum_{i \in V} (Df)(i).$$

This implies that if we normalize the weights so that constants are being integrated exactly, i.e.,

$$\sum_{w \in W} a_w = 1,$$

then the mean value of

$$(\mathrm{Id}_{n\times n} + L)^\ell \sum_{w\in W} a_w \delta_w \qquad \text{is exactly} \quad \frac{1}{n}.$$

Squaring out implies that

$$\left\| \frac{1}{n} - (\mathrm{Id}_{n\times n} + L)^\ell \sum_{w\in W} a_w \delta_w \right\|_{L^2}^2 = \frac{1}{n} + \left\| (\mathrm{Id}_{n\times n} + L)^\ell \sum_{w\in W} a_w \delta_w \right\|_{L^2}^2$$

$$- 2\left\langle \frac{1}{n}, (\mathrm{Id}_{n\times n} + L)^\ell \sum_{w\in W} a_w \delta_w \right\rangle$$

$$= \left\| (\mathrm{Id}_{n\times n} + L)^\ell \sum_{w\in W} a_w \delta_w \right\|_{L^2}^2 - \frac{1}{n}.$$

Altogether, we have shown that

$$\sup_{f\in X_\lambda} \left| \frac{1}{n} \sum_{v\in V} f(v) - \sum_{w\in W} f(w) \right| \le \frac{\|f\|_{X_\lambda}}{\lambda^\ell} \left(\left\| (\mathrm{Id}_{n\times n} + L)^\ell \sum_{w\in W} a_w \delta_w \right\|_{L^2}^2 - \frac{1}{n} \right)^{\frac{1}{2}}.$$

$\square$

Acknowledgment

References

[1] D. Bilyk, F. Dai, and R. Matzke, *The Stolarsky principle and energy optimization on the sphere*, Constr. Approx. **48** (2018), no. 1, 31–60, DOI 10.1007/s00365-017-9412-4. MR3825946

[2] X. Blanc and M. Lewin, *The crystallization conjecture: a review*, EMS Surv. Math. Sci. **2** (2015), no. 2, 225–306, DOI 10.4171/EMSS/13. MR3429164

[3] A. Bermanis, A. Averbuch, and R. R. Coifman, *Multiscale data sampling and function extension*, Appl. Comput. Harmon. Anal. **34** (2013), no. 1, 15–29, DOI 10.1016/j.acha.2012.03.002. MR2981331

[4] J. S. Brauchart and P. J. Grabner, *Distributing many points on spheres: minimal energy and designs*, J. Complexity **31** (2015), no. 3, 293–326, DOI 10.1016/j.jco.2015.02.003. MR3325677

[5] F. R. K. Chung, *Spectral graph theory*, CBMS Regional Conference Series in Mathematics, vol. 92, Published for the Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 1997. MR1421568

[6] P. Delsarte, J. M. Goethals, and J. J. Seidel, *Spherical codes and designs*, Geometriae Dedicata **6** (1977), no. 3, 363–388, DOI 10.1007/bf03187604. MR485471

[7] J. Dick and F. Pillichshammer, *Digital nets and sequences*, Discrepancy theory and quasi-Monte Carlo integration, Cambridge University Press, Cambridge, 2010. MR2683394

[8] M. Drmota and R. F. Tichy, *Sequences, discrepancies and applications*, Lecture Notes in Mathematics, vol. 1651, Springer-Verlag, Berlin, 1997. MR1470456

[9] T. Erber and G. M. Hockney, *Complex systems: equilibrium configurations of N equal charges on a sphere* ($2 \le N \le 112$), Advances in chemical physics, Vol. XCVIII, Adv. Chem. Phys., XCVIII, Wiley, New York, 1997, pp. 495–594. MR1238214

[10] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, *Walking in Facebook: A case study of unbiased sampling of osns*, INFOCOM, 2010 Proceedings IEEE, pp. 1–9, IEEE, 2010.

[11] V. Borodin, H. Snoussi, F. Hnaien, and N. Labadie, *Signal processing on graphs: Case of sampling in Paley–Wiener spaces*, Signal Processing **152** (2018), 130–140.

[12] K. Fujiwara, *Eigenvalues of Laplacians on a closed Riemannian manifold and its nets*, Proc. Amer. Math. Soc. **123** (1995), no. 8, 2585–2594, DOI 10.2307/2161293. MR1257106

[13] P. Hu and W. C. Lau, *A survey and taxonomy of graph sampling*, arXiv:1308.5865, 2013.

[14] L. Jin, Y. Chen, P. Hui, C. Ding, T. Wang, A. V. Vasilakos, B. Deng, and X. Li, *Albatross sampling: Robust and effective hybrid vertex sampling for social graphs*, Proceedings of the 3rd ACM International Workshop on MobiArch, pp. 11–16, ACM, 2011.

[15] L. Kuipers and H. Niederreiter, *Uniform distribution of sequences*, Pure and Applied Mathematics, Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, 1974. MR0419394

[16] G. C. Linderman and S. Steinerberger, *Clustering with t-SNE, provably*, SIAM J. Math. Data Sci. **1** (2019), no. 2, 313–332, DOI 10.1137/18M1216134. MR3955236

[17] J. Lu, Y. Lu, X. Wang, X. Li, G. Linderman, C. Wu, X. Cheng, L. Mu, H. Zhang, J. Liu, M. Su, H. Zhao, E. Spatz, J. Spertus, F. Masoudi, H. Krumholz and L. Jiang, *Prevalence, awareness, treatment, and control of hypertension in China: Data from 1.7 million adults in a population-based screening study* (China PEACE Million Persons Project), The Lancet 390, pp. 2549–2558, 2017.

[18] J. Lu, M. Sachs, and S. Steinerberger, *Quadrature Points via Heat Kernel Repulsion*, Constr. Approx. **51** (2020), no. 1, 27–48, DOI 10.1007/s00365-019-09471-4. MR4045735

[19] I. Pesenson, *Sampling of Paley-Wiener functions on stratified groups*, J. Fourier Anal. Appl. **4** (1998), no. 3, 271–281, DOI 10.1007/BF02476027. MR1650917

[20] I. Pesenson, *A sampling theorem on homogeneous manifolds*, Trans. Amer. Math. Soc. **352** (2000), no. 9, 4257–4269, DOI 10.1090/S0002-9947-00-02592-7. MR1707201

[21] I. Pesenson, *Sampling in Paley-Wiener spaces on combinatorial graphs*, Trans. Amer. Math. Soc. **360** (2008), no. 10, 5603–5627, DOI 10.1090/S0002-9947-08-04511-X. MR2415088

[22] I. Z. Pesenson, *Shannon sampling and weak Weyl's law on compact Riemannian manifolds*, Analysis and partial differential equations: perspectives from developing countries, Springer Proc. Math. Stat., vol. 275, Springer, Cham, 2019, pp. 207–218. MR3923340

[23] I. Pesenson, *Average sampling and average splines on combinatorial graphs*, arXiv:1901.08726, 2019.

[24] I. Pesenson, *Weighted sampling and weighted interpolation on combinatorial graphs*, arXiv:1905.02603, 2019.

[25] I. Z. Pesenson and D. Geller, *Cubature formulas and discrete Fourier transform on compact manifolds*, From Fourier analysis and number theory to Radon transforms and geometry, Dev. Math., vol. 28, Springer, New York, 2013, pp. 431–453, DOI 10.1007/978-1-4614-4075-8_21. MR2986970

[26] I. Z. Pesenson, M. Z. Pesenson, and H. Führ, *Cubature formulas on combinatorial graphs*, arXiv:1104.0963, 2011.

[27] A. H. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, and D. Stutzbach, *Respondent-driven sampling for characterizing unstructured overlays*, INFOCOM 2009, IEEE, 2009, pp. 2701–2705.

[28] A. Singer, *From graph to manifold Laplacian: the convergence rate*, Appl. Comput. Harmon. Anal. **21** (2006), no. 1, 128–134, DOI 10.1016/j.acha.2006.03.004. MR2238670

[29] S. L. Sobolev, *Cubature formulas on the sphere which are invariant under transformations of finite rotation groups* (Russian), Dokl. Akad. Nauk SSSR **146** (1962), 310–313. MR0141225

[30] S. Steinerberger, *Spectral limitations of quadrature rules and generalized spherical designs*, IMRN (to appear).

[31] S. Steinerberger, *Designs on graphs: Sampling, spectra, symmetries*, J. Graph Theory (to appear).

[32] K. B. Stolarsky, *Sums of distances between points on a sphere. II*, Proc. Amer. Math. Soc. **41** (1973), 575–582, DOI 10.2307/2039137. MR333995

[33] R. S. Strichartz, *Half sampling on bipartite graphs*, J. Fourier Anal. Appl. **22** (2016), no. 5, 1157–1173, DOI 10.1007/s00041-015-9452-8. MR3547716

[34] X. Wang, P. Liu, and Y. Gu, *Local-set-based graph signal reconstruction*, IEEE Trans. Signal Process. **63** (2015), no. 9, 2432–2444, DOI 10.1109/TSP.2015.2411217. MR3341774

[35] J. P. Ward, F. J. Narcowich, and J. D. Ward, *Interpolating splines on graphs for data science applications*, arXiv:1806.10695, 2018.

PROGRAM IN APPLIED MATHEMATICS, YALE UNIVERSITY, NEW HAVEN, CONNECTICUT 06511
Email address: george.linderman@yale.edu

DEPARTMENT OF MATHEMATICS, YALE UNIVERSITY, NEW HAVEN, CONNECTICUT 06511
Email address: stefan.steinerberger@yale.edu