

# Graph Diffusion

Stefano Fochesatto

April 3, 2023

## 1 Introduction

Throughout our course we have considered the heat equation for many examples; simple problems with square boundaries and straight forward discretization schemes. The goal of this project is to study the more general, diffusion equation with graphs as the domain.

This problem can be thought of as a continuous version of the network flow problems discussed in optimization. Suppose there exists some flow, in most examples we've seen that flow represents heat and in the context of optimization goal was to move flow from one vertex to another in an underlying network as efficiently as possible. Here our goal is to model the state of this flow as a closed system which respects conservation laws.

## 2 Continuum Model [or Continuum Problem]

Unsurprisingly, the diffusion equation when applied to a graph looks very familiar. Let  $\alpha$  be the diffusivity constant,  $\phi(t)$  is flow state vector at each vertex with respect to time, and  $L$  is the aptly named Laplacian matrix of the graph, our problem is stated as [3],

$$\frac{\partial \phi}{\partial t} = -\alpha L \phi(t).$$

The Laplacian matrix of a graph is defined as  $L = D - A$  where  $D$  is the diagonal degree matrix of the graph, and  $A$  is the adjacency matrix[1]. As of now we will only consider simple, weighted, undirected graphs, so  $L$  will always be a symmetric matrix.

A significant portion of this section will be devoted to deriving this equation from, Newton's law of cooling. For example, when applied to a single pair of vertices on the graph  $(i, j)$  we get that the rate of flow leaving  $i$  towards  $j$  is,

$$\frac{\partial \phi_{ij}}{\partial t} = \alpha \omega_{ij} (\phi_i - \phi_j).$$

Again here  $\phi_i$  represents flow at vertex  $i$  and  $\omega_{ij}$  represents the weight along the edge between  $i$  and  $j$ . Summing across all vertices in the neighborhood of  $i$ , we get a differential equation which describes the flow at a vertex  $i$ ,

$$\frac{\partial \phi_i}{\partial t} = - \sum_{j \in \mathcal{N}(i)} \alpha \omega_{ij} (\phi_i - \phi_j)$$

Describing this equation for each vertex in the graph generates a system of coupled ordinary differential equations, and we'll see later that this system can be written neatly by way of the Laplacian representation of the graph.

In this section we will also derive the analytical solution for this problem[2],

$$\phi(t) = \sum_{i=1}^n v_i^t e^{-\alpha \lambda_i t} v_i.$$

where  $\lambda_i$ ,  $v_i$  are the eigenvalues and eigenvectors of the Laplacian matrix.

### 3 Numerical Scheme(s)

As stated this problem is a system of ODE IVP, and therefore an application of any of the ODE IVP schemes discussed throughout the course would be suitable. Implementation would be very straight forward, essentially all we would have to do is implement a function which evaluates the left hand side:  $-\alpha L\phi(t)$ , then apply an ODE IVP scheme.

I'd like to proceed with some sort of truncation error analysis, so a simpler scheme like explicit trapezoid or midpoint seems more likely.

### 4 Analysis

Since we have access to an analytical solution, an experimental analysis of truncation error for this class of problems would be straight forward.

Beyond that I'd like to explore which class of graphs cause our scheme to be unstable, or ill-posed.

I would also like to compare the continuous flow described by this equation to the discrete flow described by network simplex.

### 5 Results

### References

- [1] Naumann, U.& Schenk, O. (2012). *Combinatorial Scientific Computing*. CRC Press.
- [2] Leonid Zhukov (Director). (2015, March 31). *Network Analysis. Lecture 11. Diffusion and random walks on graphs*. <https://www.youtube.com/watch?v=reTP7XAJztU>
- [3] Cory Simon. (n.d.). *Diffusion on graphs*. Retrieved April 3, 2023, from [https://simonensemble.github.io/pluto.nbs/graph\\_diffusion\\_blog.jl.html](https://simonensemble.github.io/pluto.nbs/graph_diffusion_blog.jl.html)

### A Something suitable for an appendix

I'll have a bit of graph theory terminology here.