# From Mass-Spring Systems to Spectral Graph Neural Networks

Nguyen Ngoc Khanh

May 20, 2025

## 1 Mass-Spring System

### 1.1 The two particles system

Consider a spring follows Hook's law. Let two particles $i$ and $j$ connected by a spring located at $x_i$ and $x_j$ respectively and $e_{ij} = \frac{x_j - x_i}{||x_j - x_i||_2}$ be the direction from $x_i$ to $x_j$ then the force that $i$ affects $j$ can be represented as:

$$F_{ij} = -k(||x_j - x_i)||_2 - L)e_{ij} = -k(x_j - x_i) + kLe_{ij} \tag{1}$$

where $k$ is a positive real number, the characteristic of the spring and $L$ is the initial length the of spring. The magnitude of the force is proportional to the displacement from the initial distance between two particles.

Let two particles connected by a spring sit in an Euclidean space such that the particles can freely move on a particular $z$ axis. At the initial condition, the two particles are located at $x_i$ and $x_j$ and the spring is at its length ($||x_j - x_i||_2 = L$) (no force).

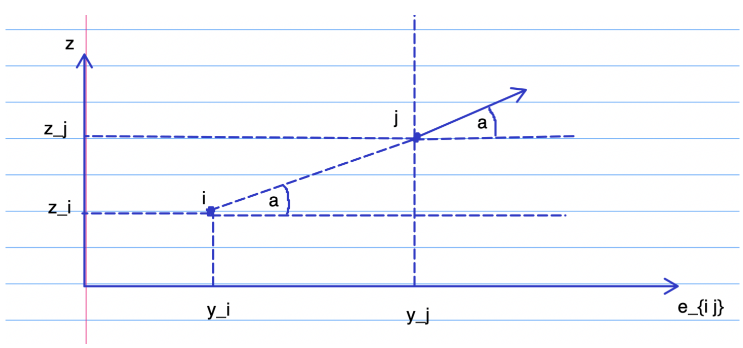


Figure 1: subspace of $e_{ij}$ and $z$

Since, $i$ and $j$ can only move in the $z$ axis, we can rewrite

$$x_i = x_i^{(0)} + z_i z$$
$$x_j = x_j^{(0)} + z_j z$$

Where $x_i^{(0)}$ and $x_j^{(0)}$ are the initial positions of $i$ and $j$, $z_i$ and $z_j$ are the displacements on the $z$ axis. Hence, the projected force on the $z$ axis can be written as

$$F_{ij} \cdot z = (-k(x_j - x_i) + kLe_{ij}) \cdot z = (-k(x_j^{(0)} - x_i^{(0)}) + kLe_{ij}) \cdot z + (-k(z_j - z_i)z) \cdot z \tag{2}$$

The first term is the dot product of the initial force with the $z$ direction which is essential zero since there is no force at the beginning. Hence, the projected force on the $z$ axis can be written as

$$F_{ij} \cdot z = -k(z_j - z_i) \tag{3}$$

The projected force on the $z$ axis linearly depends on the corresponding displacement.

### 1.2 The n particles system

Let $n$ particles with the same weight $m$ on an an Euclidean space that can freely move on a particular $z$ axis. Some of them are connected by springs of the same characteristic $k$ which is denoted by a undirected unweighted graph $G = (V, E)$. A particular node $i$ is affected by all of its neighbours where the projected force on $i$ is

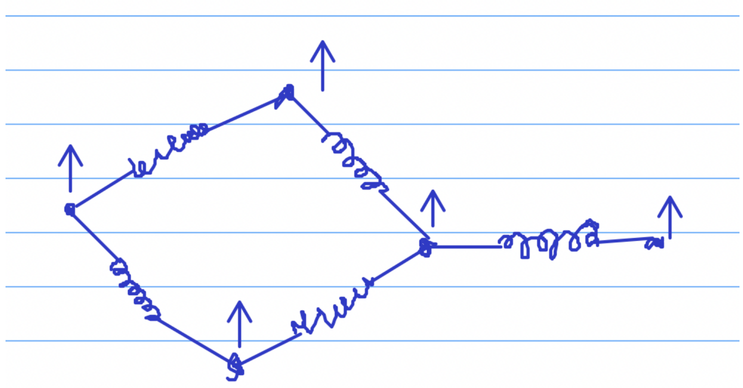

Figure 2: 4 particles system

$$F_i \cdot z = -k \sum_{e_{ji} \in E} z_i - z_j = -k(d_i z_i - \sum_{e_{ji} \in E} z_j) \tag{4}$$

Where $d_i$ denotes the degree of the node $i$. Newton's Second Law of Motion:

$$a_i \cdot z = \frac{F_i}{m} \cdot z$$
$$\ddot{z_i} = -\frac{k}{m}(d_i z_i - \sum_{e_{ji} \in E} z_j)$$

We can rewrite in the matrix form

$$\ddot{z} = -\frac{k}{m}(D - A)z = -\frac{k}{m}Lz \tag{5}$$

Where $z = (z_1, z_2, ..., z_n)^T$, $A$ is the adjacency matrix of $G$ and $D$ is the degree matrix of $A$ (diagonal matrix where each entry equals to the corresponding degree of the node). $L = D - A$ is called Laplacian matrix of $A$.

We are seeking the mode of oscillation of the system. A mode of oscillation is a particular frequency where all particles oscillate at the same frequency. At that frequency, the differential equation for each particle must be in the form:

$$\ddot{z_i} = -\omega^2 z_i \tag{6}$$

Where $\omega$ is the frequency. In the matrix form:

$$\ddot{z} = -\omega^2 z \tag{7}$$

From 5 and 7, we have

$$Lz = \frac{m}{k}\omega^2 z \tag{8}$$

From 8, the oscillation mode frequencies are equivalent to the eigenvalues of the Laplacian matrix, and the initial condition to achieve each of the frequencies is the corresponding eigenvector.

Since $L$ is real symmetric, by the Spectral Theorem, it has an eigenbasis. Furthermore, $L$ is positive semi definite, then all of its eigenvalues are positive, hence the frequencies make sense.

For an arbitrary initial condition, since the system is linear, we can decompose the displacement $z$ into the eigenbasis of the Laplacian matrix then solve each of the component individually.

# 2 Graph Laplacian Basis

Recall that, the eigen decomposition of $L$ is as follow:

$$L = U\Lambda U^T = U\Lambda U^{-1} \tag{9}$$

Where each column vector in $U$ is a normalized eigenvector.

Analogous to Fourier Transform, the eigenvalues of Laplacian matrix can serve as the frequency and the eigenbasis is corresponding to the Fourier basis.

Let $x \in \mathbb{R}^n$ be a graph signal on $G = (V, E)$ where each component of $x$ is a real number corresponding to a node in $G$.

The convolution in spatial domain is equivalent to multiplication in spectral domain. Define the convolution operation as:

$$y(x) = U(U^T w \odot U^T x) \tag{10}$$

Where $w \in \mathbb{R}^n$ is called filter or kernel. Define $W = diag(U^T w)$ be the diagonal matrix whose entries are the entries of $U^T w$, we can rewrite 10 as

$$y(x) = (UWU^T)x \tag{11}$$

## 2.1 ChebNet

Let $\mathcal{L}$ be the normalized laplacian matrix.

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{12}$$

The decomposition of $\mathcal{L}$

$$\mathcal{L} = U \Lambda U^T = U \Lambda U^{-1} \tag{13}$$

**Theorem 1** *(Chung [**?**]) All eigenvalues of $\mathcal{L}$ are in the interval* $[0, 2]$.

ChebNet [**?**] approximate the diagonal matrix $W$ using Chebyshev polynomials as the orthogonal basis in the polynomial subspace of the vector space of all functions $f : [-1, +1] \to \mathbb{R}$ with respect to the inner product.

$$\int_{-1}^{+1} f(x) g(x) \frac{dx}{\sqrt{1 - x^2}} \tag{14}$$

Chebyshev polynomials of the first kind:

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x) \tag{15}$$

Where $x \in [-1, +1]$, $T_0(x) = 1$ and $T_1(x) = x$. Let $f_i : [-1, +1] \to \mathbb{R}$ is an arbitrary function such that $f_i(\tilde{\lambda_i}) = w_i$ where $\tilde{\lambda_i} = \frac{2\lambda_i}{\lambda_{\max}} - 1 \in [-1, +1]$, $\lambda_{\max}$ is the largest eigenvalue. We want to project $f_i$ into the subspace with the orthogonal basis of the first $K$ terms of Chebyshev polynomials of the first kind. We can write $f_i$ as

$$\hat{f}_i(t) = \sum_{k=0}^{K-1} \theta_{ki} T_k(t) \tag{16}$$

Hence, $w_i$ is approximated as

$$\hat{w}_i = \hat{f}_i(\tilde{\lambda}_i) = \sum_{k=0}^{K-1} \theta_{ki} T_k(\tilde{\lambda}_i) \tag{17}$$

Matrix form of the approximation on $W$:

$$\hat{W} = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}) \tag{18}$$

Where $\tilde{\Lambda}$ is the diagonal matrix of $\tilde{\lambda}_i$. Moreover,

$$U\hat{W}U^T = \sum_{k=0}^{K-1} \theta_k U T_k(\tilde{\Lambda}) U^T = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathcal{L}}) \tag{19}$$

Where $\tilde{\mathcal{L}} = \frac{2\mathcal{L}}{\lambda_{\max}} - I$. It is a great exercise to prove the Chebyshev recurrence for $\tilde{\mathcal{L}}$:

$$T_{n+1}(\tilde{\mathcal{L}}) = 2\tilde{\mathcal{L}} T_n(\tilde{\mathcal{L}}) - T_{n-1}(\tilde{\mathcal{L}}) \tag{20}$$

Finally, The convolution operation is

$$y(x) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathcal{L}}) x \tag{21}$$

The construction of ChebNet avoids decomposing the matrix $L$ as compare to 10.

### 2.2 Graph Convolutional Network

Similar to ChebNet, GCN [**?**] limits $K = 2$ and sets $\lambda_{\max} = 2$ hence $\tilde{\mathcal{L}} = \mathcal{L} - I$.

$$\begin{aligned} U\hat{W}U^T &= \theta_0 T_0(\tilde{\mathcal{L}}) + \theta_1 T_1(\tilde{\mathcal{L}}) \\ &= \theta_0 I + \theta_1 \tilde{\mathcal{L}} \\ &= \theta_0 I - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \end{aligned}$$

The convolution operation is

$$y(x) = \theta_0 x - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \tag{22}$$

The notes here is greatly inspired by [**?**].