Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università degli Studi di Padova

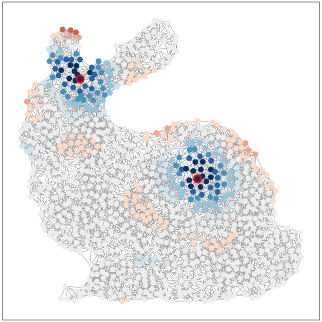
A Harmonic Journey trough Graph Wavelets

2. Spectral Graph Wavelets

Wolfgang Erb
University of Padova

Frontiers and Applications of Approximation Theory
Modern Perspectives for Young Researchers

March 16-20, 2026
Lucian Blaga University of Sibiu, Romania



wolfgang.erb@unipd.it

Outline of this talk

1. Spectral Wavelets on Graphs
   - Graph Fourier Transform, convolution and generalized translation on graphs
   - Construction of spectral wavelets on graphs
   - Reconstruction formula for spectral wavelet coefficients
2. Krylov subspace methods for the calculation of spectral wavelets
   - Krylov subspace methods for the approximation of the graph convolution
   - Setting up window functions for spectral scaling functions and wavelets
3. Diffusion Wavelets
   - Idea of wavelets based on diffusion operators.
   - How to numerically calculate diffusion wavelets.

Recapitulation: the Graph Fourier Transform

We write the eigendecomposition of the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ as

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^*,$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the eigenvalues of $\mathbf{L}$ (increasingly ordered) and the unitary matrix $\mathbf{U} = (u_1, u_2, \ldots, u_n)$ the eigenvectors.

Then, the Graph Fourier transform of x is defined as

$$\hat{x} = \mathbf{U}^* x, \quad \text{with } k\text{-th. entry} \quad \hat{x}_k = u_k^* x = \langle x, u_k \rangle.$$

The inverse Fourier transform is correspondingly given as

$$x = \mathbf{U}\hat{x}.$$

Example: the bunny graph

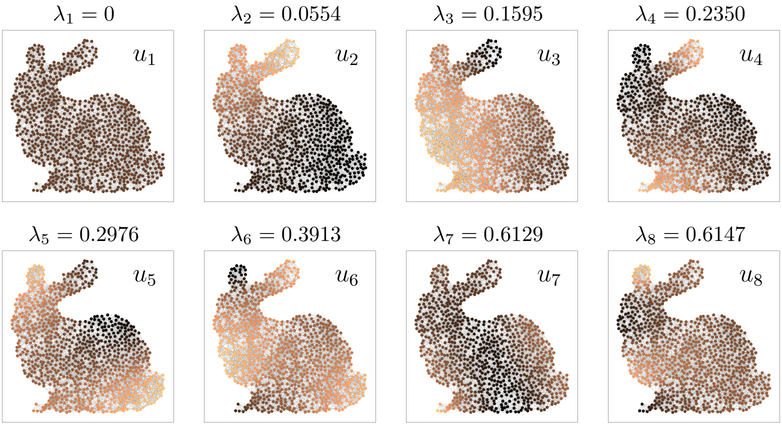


Figure 1: The first 8 eigenfunctions of the graph Laplacian $\mathbf{L}$ on the bunny graph.

Convolution on graphs

Convolution in $\mathbb{R}$

$$(x * y)(s) = \int_{\mathbb{R}} x(t)y(s-t)\mathrm{d}t.$$

In the Fourier domain:

$$\widehat{(x * y)}(\omega) = \hat{x}(\omega)\hat{y}(\omega)$$

Graph convolution

No translation available
Idea: define convolution via graph Fourier transform

$$\widehat{(x * y)}_k = \hat{x}_k \hat{y}_k$$

We define the graph convolution as

$$y * x := \mathbf{U}\mathbf{M}_{\hat{y}}\hat{x} = \mathbf{U}\mathbf{M}_{\hat{y}}\mathbf{U}^* x, \quad \text{where } \mathbf{M}_{\hat{y}} = \mathrm{diag}(\hat{y}_1, \ldots \hat{y}_n).$$

Further, we define the convolution matrix $\mathbf{C}_y \in \mathbb{R}^{n \times n}$ as

$$\mathbf{C}_y = \mathbf{U}\mathbf{M}_{\hat{y}}\mathbf{U}^*.$$

Note: the graph convolution depends on the choice of the basis u_k.

Generalized translation on graphs

Translation in $\mathbb{R}$

In the weak sense, we have

$$x(t+s) = (x * \delta_s)(t) = \int_{\mathbb{R}} \hat{x}(\omega) e^{-2\pi \imath \omega s} e^{-2\pi \imath \omega t} \mathrm{d}\omega$$

Graph translation

We can define a generalized translation as

$$\mathbf{C}_x \delta_{\mathrm{v}} = (x * \delta_{\mathrm{v}}) = \mathbf{U}\mathbf{M}_{\hat{x}}\mathbf{U}^* \delta_{\mathrm{v}}$$

Warnings:

- In general, no group structure for the generalized translation
- Depends on the choice of the basis elements u_k.

Spectral Graph Wavelet Transform (SGWT)

The spectral graph wavelet transform (SGWT) is determined by filters acting on the spectra of a signal x.

For this, we require two window functions:

$$h : [0, \infty) \to \mathbb{R}, \quad h(0) > 0, \quad \lim_{x \to \infty} h(x) = 0$$

for the spectral scaling operator, and

$$g : [0, \infty) \to \mathbb{R}, \quad g(0) = 0, \quad \lim_{x \to \infty} g(x) = 0$$

for the spectral wavelets. The functions h and g act as low-pass and band-bass filters on the spectrum of $x \in \mathcal{L}(V)$.

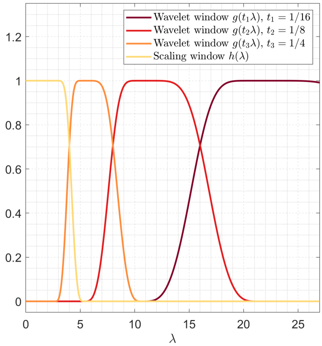
Definition 1

Based on the window functions g and h, and a scaling parameter $t > 0$, we define the spectral scaling operator $\mathbf{T}_h$ and the spectral wavelet operator $\mathbf{T}_{g,t}$ for a graph signal x in the frequency domain as

$$(\widehat{\mathbf{T}_h x})_k := h(\lambda_k)\hat{x}_k, \quad \text{and} \quad (\widehat{\mathbf{T}_{g,t} x})_k := g(t\lambda_k)\hat{x}_k. \tag{1}$$

- The function h for $\mathbf{T}_h x$ is selected such that the low-frequency content of the signal x is approximated;
- The higher frequency parts of x are represented by the wavelet components $\mathbf{T}_{g,t} x$ for different choices of the scaling parameter t.
- As g is defined on $[0, \infty)$, the scaling $t > 0$ allows to localize the filter g in different parts of the spectrum of $\mathbf{L}$.

Example of window functions



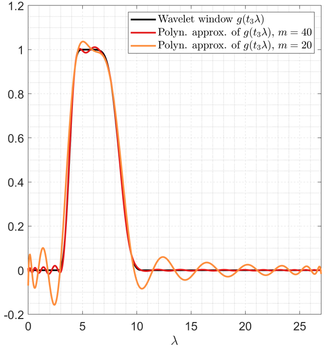


Figure 2: Left: shifts $g(t_j\lambda)$ of the Meyer wavelet window g and the Meyer scaling function h on the spectrum $[0, \lambda_n]$ of the graph Laplacian $\mathbf{L}$ of the bunny graph. Right: approximation of the wavelet window g with polynomials of degree m.

Applying the inverse GFT to (1), we obtain the following representations of the operators $\mathbf{T}_h$ and $\mathbf{T}_{g,t}$ in the vertex domain:

$$(\mathbf{T}_h x)(\mathrm{v}) = \sum_{k=1}^{n} h(\lambda_k)\hat{x}_k u_k(\mathrm{v}), \quad (\mathbf{T}_{g,t} x)(\mathrm{v}) = \sum_{k=1}^{n} g(t\lambda_k)\hat{x}_k u_k(\mathrm{v}). \quad (2)$$

Both operations can be considered as convolutional filters applied to the signal x. In particular, in matrix-vector form, they can be written as

$$\mathbf{T}_h x = \mathbf{U} h(\mathbf{\Lambda}) \mathbf{U}^* x = h(\mathbf{L}) x, \qquad \mathbf{T}_{g,t} x = \mathbf{U} g(t\mathbf{\Lambda}) \mathbf{U}^* x = g(t\mathbf{L}) x.$$

Definition 2

The spectral graph scaling functions ϕ_i as well as the spectral graph wavelets $\psi_{t,i}$ are defined through the application of $\mathbf{T}_h$ and $\mathbf{T}_{g,t}$ to the canonical basis δ_{v_i}, localized on the vertices v_i of the graph. We define

$$\phi_i(\mathrm{v}) := \mathbf{T}_h \delta_{\mathrm{v}_i}(\mathrm{v}), \quad \psi_{t,i}(\mathrm{v}) := \mathbf{T}_{g,t} \delta_{\mathrm{v}_i}(\mathrm{v}). \tag{3}$$

The corresponding expansions in the Fourier basis read as

$$\phi_i(\mathrm{v}) = \sum_{k=1}^{n} h(\lambda_k) u_k(\mathrm{v}_i) u_k(\mathrm{v}), \quad \psi_{t,i}(\mathrm{v}) = \sum_{k=1}^{n} g(t\lambda_k) u_k(\mathrm{v}_i) u_k(\mathrm{v}). \tag{4}$$

Note: the scaling function $\phi_i = \mathbf{T}_h \delta_{\mathrm{v}_i}$ can be regarded as generalized translation of the signal $\mathbf{U}h$ with the node v_i. Typically ϕ_i and $\psi_{t,i}$ will be localized around the node v_i.

Examples of spectral graph wavelets

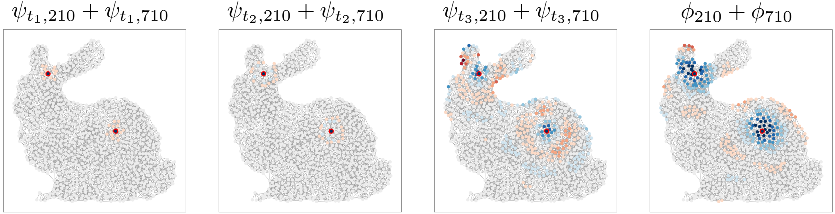


Figure 3: Illustration of spectral graph wavelets $\psi_{t,i}(\mathrm{v})$ and the scaling functions $\phi_i(\mathrm{v})$ on the bunny graph. The spectral wavelets are based on the Meyer scaling function h and the Meyer window function g given in Fig. 2. The red rings indicate the center nodes v_{210} and v_{710} of the wavelets and scaling functions.

Calculation of the spectral wavelet coefficients

The wavelet coefficients of a signal x are computed as the inner products

$$\langle x, \psi_{t,i} \rangle$$

between the signal x and the spectral wavelets $\psi_{t,i}$. By the orthonormality of the Fourier basis $\{u_1, \ldots, u_n\}$ and the symmetry of $\mathbf{T}_{g,t}$, the wavelet coefficients can be written as

$$\langle x, \psi_{t,i} \rangle = \langle \mathbf{T}_{g,t} x, \delta_{\mathrm{v}_i} \rangle = (\mathbf{T}_{g,t} x)(\mathrm{v}_i) = \sum_{k=1}^{n} g(t\lambda_k) \hat{x}_k u_k(\mathrm{v}_i). \tag{5}$$

Similarly, for the coefficients of the spectral scaling functions we get

$$\langle x, \phi_i \rangle = \langle \mathbf{T}_h x, \delta_{\mathrm{v}_i} \rangle = (\mathbf{T}_h x)(\mathrm{v}_i) = \sum_{k=1}^{n} h(\lambda_k) \hat{x}_k u_k(\mathrm{v}_i). \tag{6}$$

Reconstruction from spectral wavelet coefficients

Let ϕ_i and $\psi_{t_j,i}$ be defined via the window functions h and g, and a finite set of scaling parameters

$$0 < t_1 < \cdots < t_J.$$

As a redundant set of elements for the decomposition of x, we consider

$$\mathcal{F} = \{\mathbf{T}_h \delta_{\mathrm{v}_i} \,:\, i \in \{1,\ldots,n\}\} \cup \left\{\mathbf{T}_{g,t_j} \delta_{\mathrm{v}_i} \,:\, i \in \{1,\ldots,n\}, j \in \{1,\ldots,J\}\right\}.$$

Based on this system, we are able to analyse x in terms of the coefficients

$$\left\{\langle x, \phi_i \rangle \,:\, i \in \{1,\ldots,n\}\right\} \cup \left\{\langle x, \psi_{t_j,i} \rangle \,:\, i = \{1,\ldots,n\},\, j \in \{1,\ldots,J\}\right\}.$$

In order to guarantee recovery of the signal x from these coefficients, it is essential that the function

$$G(\lambda) = h^2(\lambda) + \sum_{j=1}^{J} g(t_j\lambda)^2 > 0$$

is strictly positive on the spectrum of the graph Laplacian.

Theorem 3 (Part 1: frame bounds)

Given the window functions h and g, and a finite set of scales $0 < t_1 < \cdots < t_J$, we assume that the function $G(\lambda) > 0$ is strictly positive on the interval $[0, \lambda_n]$. Then, the set

$$\mathcal{F} = \{\phi_i \,:\, i \in \{1, \ldots, n\}\} \cup \left\{\psi_{t_j,i} \,:\, i \in \{1, \ldots, n\}, j \in \{1, \ldots, J\}\right\}$$

forms a frame in $\mathcal{L}^2(V)$ by satisfying the two inequalities

$$A\|x\|_2^2 \leq \sum_{i=1}^{n} \left(|\langle x, \phi_i \rangle|^2 + \sum_{j=1}^{J} |\langle x, \psi_{t_j,i} \rangle|^2 \right) \leq B\|x\|_2$$

with the frame bounds

$$A = \min_{\lambda \in [0,\lambda_n]} G(\lambda), \quad B = \max_{\lambda \in [0,\lambda_n]} G(\lambda).$$

Theorem 3 (Part 2: reconstruction formula)

Further, if we define the canonical dual scaling functions and wavelets as

$$\tilde{\phi}_i = \mathbf{T}_{1/G} \mathbf{T}_h \delta_{\mathrm{v}_i} = \mathbf{T}_{h/G} \delta_{\mathrm{v}_i}, \quad \tilde{\psi}_{t_j,i}(\mathrm{v}) = \mathbf{T}_{1/G} \mathbf{T}_{g,t} \delta_{\mathrm{v}_i},$$

the signal x can be recovered from the spectral wavelet coefficients as

$$x(\mathrm{v}) = \sum_{i=1}^{n} \langle x, \phi_i \rangle \tilde{\phi}_i(\mathrm{v}) + \sum_{i=1}^{n} \sum_{j=1}^{J} \langle x, \psi_{t_j,i} \rangle \tilde{\psi}_{t_j,i}(\mathrm{v}).$$

Note: The operator $\mathbf{T}_G$ defined as

$$\mathbf{T}_G x(\mathrm{v}) = \sum_{i=1}^{n} \langle x, \phi_i \rangle \phi_i(\mathrm{v}) + \sum_{i=1}^{n} \sum_{j=1}^{J} \langle x, \psi_{t_j,i} \rangle \psi_i(\mathrm{v})$$

is referred to as frame operator.

Proof of Theorem 3

For a given graph signal x, we consider the coefficients $\langle x, \phi_i \rangle$ and $\langle x, \psi_{t_j,i} \rangle$. Using the identity (5), the Pythagorean theorem yields

$$\sum_{i=1}^{n} |\langle x, \psi_{t_j,i} \rangle|^2 = \sum_{i=1}^{n} \left| \sum_{k=1}^{n} g(t_j \lambda_k) \hat{x}_k u_k(\mathrm{v}_i) \right|^2 \tag{7}$$

$$= \left\| \sum_{k=1}^{n} g(t_j \lambda_k) \hat{x}_k u_k \right\|_2^2 = \sum_{k=1}^{n} |g(t\lambda_k)|^2 |\hat{x}_k|^2.$$

Analogously, by (6) we obtain for the scaling functions the identity

$$\sum_{i=1}^{n} |\langle x, \phi_i \rangle|^2 = \sum_{k=1}^{n} |h(\lambda_k)|^2 |\hat{x}_k|^2. \tag{8}$$

Proof of Theorem 3

Combining (7) and (8), we get

$$\sum_{i=1}^{n}\left(|\langle x,\phi_i\rangle|^2+\sum_{j=1}^{J}|\langle x,\psi_{t_j,i}\rangle|^2\right)=\sum_{k=1}^{n}\left(|h(\lambda_k)|^2+\sum_{j=1}^{J}|g(t_j\lambda_k)|^2\right)|\hat{x}_k|^2$$
$$=\sum_{k=1}^{n}G(\lambda_k)|\hat{x}_k|^2.$$

Then, by definition of the constants A and B, we get the inequalities

$$A\sum_{k=1}^{n}|\hat{x}_k|^2\leq\sum_{i=1}^{n}\left(|\langle x,\phi_i\rangle|^2+\sum_{j=1}^{J}|\langle x,\psi_{t_j,i}\rangle|^2\right)\leq B\sum_{k=1}^{n}|\hat{x}_k|^2. \quad (9)$$

The Pythagorean theorem $\|x\|_2^2=\sum_{k=1}^{n}|\hat{x}_k|^2$ implies now the statement for the frame bounds.

Proof of Theorem 3

Also the inversion formula can be shown directly. With the identities (4) and (5), and the orthonormality of the basis functions u_k, we get

$$\begin{aligned}\sum_{i=1}^{n}\langle x,\psi_{t_j,i}\rangle\tilde{\psi}_i(\mathrm{v}) &= \sum_{i=1}^{n}\sum_{k=1}^{n} g(t_j\lambda_k)\hat{x}_k u_k(\mathrm{v}_i)\sum_{k'=1}^{n}\frac{g(t_j\lambda_{k'})}{G(\lambda_k)}u_{k'}(\mathrm{v}_i)u_{k'}(\mathrm{v}) \\ &= \sum_{k=1}^{n}\hat{x}_k\frac{g(t\lambda_k)^2}{G(\lambda_k)}u_k(\mathrm{v}).\end{aligned} \quad (10)$$

In the same way, (4) and (6) provide for the scaling function the identity

$$\sum_{i=1}^{n}\langle x,\phi_i\rangle\tilde{\phi}_i(\mathrm{v}) = \sum_{k=1}^{n}\hat{x}_k\frac{h(\lambda_k)^2}{G(\lambda_k)}u_k(\mathrm{v}). \quad (11)$$

Combining (10) and (11) and the inverse GFT, we finally get

$$\sum_{i=1}^{n}\langle x,\phi_i\rangle\tilde{\phi}_i(\mathrm{v}) + \sum_{i=1}^{n}\sum_{j=1}^{J}\langle x,\psi_{t_j,i}\rangle\tilde{\psi}_i(\mathrm{v}) = \sum_{k=1}^{n}\hat{x}_k u_k(\mathrm{v}) = x(\mathrm{v}).$$

□

Numerical calculation of the SGWT

Similar to the GFT, the SGWT can be written compactly in matrix form:

$$c_{t_j} = \begin{bmatrix} \langle x, \psi_{t_j,1} \rangle \\ \vdots \\ \langle x, \psi_{t_j,n} \rangle \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{g,t_j} x(\mathrm{v}_1) \\ \vdots \\ \mathbf{T}_{g,t_j} x(\mathrm{v}_n) \end{bmatrix} = \mathbf{T}_{g,t_j} x, \quad j \in \{1, \ldots, J\}.$$

Also the coefficients of the scaling function can be expressed as

$$d = \begin{bmatrix} \langle x, \phi_1 \rangle \\ \vdots \\ \langle x, \phi_n \rangle \end{bmatrix} = \begin{bmatrix} \mathbf{T}_h x(\mathrm{v}_1) \\ \vdots \\ \mathbf{T}_h x(\mathrm{v}_n) \end{bmatrix} = \mathbf{T}_h x.$$

Further, for given c_{t_j} and d, the reconstruction formula is given as

$$x = \mathbf{T}_{h/G} d + \sum_{j=1}^{J} \mathbf{T}_{1/G} \mathbf{T}_{g,t_j} c_{t_j}.$$

In particular, all involved operators are convolution operators on the graph.

Efficient calculation of the graph convolution

If we approximate the filter function h with a polynomial p_m of degree m on the spectrum of $\mathbf{L}$, the filtered signal $h * x$ can be calculated without the usage of the spectral decomposition.

$$h(\Lambda) \approx \sum_{k=0}^{m} \alpha_k \lambda_k^k = p_m(\lambda_k), \quad m \leq n.$$

Then, we can approximate the filtered signal as

$$\begin{aligned} h * x &= \mathbf{U}\,\mathrm{diag}(h(\lambda_1), \ldots, h(\lambda_n))\mathbf{U}^* x \\ &\approx \mathbf{U}\,\mathrm{diag}(p_m(\lambda_1), \ldots, p_m(\lambda_n))\mathbf{U}^* x \\ &= p_m(\mathbf{U}\,\mathrm{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{U}^*) x \\ &= p_m(\mathbf{L}) x = \sum_{k=0}^{m} \alpha_k \mathbf{L}^k x, \end{aligned}$$

i.e., the filtered signal $p_m(\mathbf{L})x$ is an element of the Krylov space $\mathcal{K}^{(m+1)}(\mathbf{L}, x)$. To calculate it, only matrix-vector products are necessary.

Design of filter polynomials in $\mathcal{K}^{(m+1)}(\mathbf{L}, x)$

Strategy for the construction of the filters:

- Construct the polynomial $p_m(\lambda)$ such that it approximates well the filter function on the spectral domain (i.e. on the eigenvalues of $\mathbf{L}$).
- Use a "good" basis to represent the polynomial. The monomial basis λ^k, $k \in \{0, \ldots, m\}$ is in general not that well suited for this purpose, better use a properly dilated and shifted Chebyshev basis $\tilde{T}_k(\lambda)$.

Calculate the filtered signal as

$$f * x = p_m(\mathbf{L})x = \sum_{k=0}^{m} \beta_k \tilde{T}_k(\mathbf{L})x.$$

Approximation with Chebyshev polynomials

The Chebyshev polynomials $\{T_k(t)\}$ on $[-1, 1]$ satisfy

$$T_0(t) = 1, \quad T_1(t) = t, \quad T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t), \quad k \in \mathbb{N},\ k \geq 2.$$

To apply them for $\mathbf{L}$, the Chebyshev polynomials must be shifted and rescaled to the interval $[0, \lambda_n]$. This is achieved by

$$\tilde{T}_k(t) = T_k\left(\frac{2}{\lambda_n}t - 1\right), \quad k \in \mathbb{N}_0,$$

so that any polynomial $p_m(\lambda)$ can be decomposed as

$$p_m(\lambda) = \sum_{k=0}^{m} \beta_k \tilde{T}_k(\lambda),$$

with some coefficients $\beta_k \in \mathbb{R}$, $k \in \{0, \ldots, m\}$.

Calculation of the polynomials $p_m(\lambda)$

A simple way to construct $p_m(\lambda)$ is by interpolation:
Compute $p_m(\lambda)$ so that the interpolation conditions

$$p_m(t_j) = h(t_j), \quad j = \{0, \ldots, m\},$$

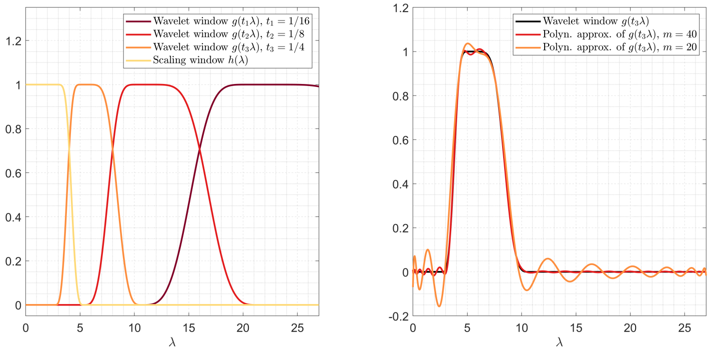
are satisfied, where

$$t_j = \frac{\lambda_n}{2}\left(\cos\left(\frac{2j+1}{2m+2}\pi\right) + 1\right), \qquad j = \{0, \ldots, m\},$$

are the shifted and dilated roots of the Chebyshev polynomial $T_{m+1}(\lambda)$.

- Using these nodes, the expansion coefficients β_k can be calculated efficiently from the samples $h(t_j)$ using a discrete cosine transform.
- The so calculated coefficients β_k are independent of the signal x and can be reused in calculations.

Approximation of wavelet window functions



If the window functions h and g have large derivatives, the degree m of the approximating polynomial $p_m(\lambda)$ has to be large in order to obtain a reasonable accuracy. Also, Gibbs-type effects might occur.

Alternative: Krylov subspace methods based on the Lanczos iteration.

Example: Meyer scaling functions

Define the transition function $\nu(x)$ as

$$\nu(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ x^4(35 - 84x + 70x^2 - 20x^3) & \text{if } 0 < x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

Then, the Meyer scaling function h is defined by the smooth low-pass filter

$$h(\lambda) = \begin{cases} 1, & \lambda \leq \frac{2}{3t_J}, \\ \cos\left(\frac{\pi}{2}\nu\left(\frac{3t_J}{2}\lambda - 1\right)\right), & \frac{2}{3t_J} < \lambda < \frac{4}{3t_J}, \\ 0, & \lambda \geq \frac{4}{3t_J}, \end{cases}$$

where $t_J > 0$ is the largest scaling parameter.
The scaling window h is supported in $[0, \frac{4}{3t_J}]$.

Meyer wavelet functions

The Meyer wavelet window g is given as

$$g(\lambda) = \begin{cases} \sin\left(\frac{\pi}{2}\nu\left(\frac{3}{2}\lambda - 1\right)\right), & \frac{2}{3} < \lambda < \frac{4}{3}, \\ \cos\left(\frac{\pi}{2}\nu\left(\frac{3}{4}\lambda - 1\right)\right), & \frac{4}{3} < \lambda < \frac{8}{3}, \\ 0, & \text{otherwise.} \end{cases}$$

This function is supported in $[\frac{2}{3}, \frac{8}{3}]$. Defining the dyadic scales $t_j = 2^{j-J} t_J$, we then get the scaled windows $g(t_j \lambda) = g(2^{j-J} t_J \lambda)$ supported in the interval $[\frac{2^{J-j+1}}{3t_J}, \frac{2^{J-j+3}}{3t_J}]$. The Meyer construction is designed so that

$$G(\lambda) = h(\lambda)^2 + \sum_{j=1}^{J} g(2^{j-J} t_J \lambda)^2 = 1, \qquad \lambda \in \left[0, \frac{2^{J+1}}{3t_J}\right].$$

Diffusion Wavelets

On graphs, we don't have classical dilation and translation at disposition which are essential for the definition of wavelets.

In SGWT:

- Translation is replaced by generalized convolutional translation.
- Dilation via different scaling parameters t_j on the spectrum of $\mathbf{L}$.

In diffusion wavelets (DW):

- Dilation via repeated application of a diffusion operator $\mathbf{T}$.
- Translation via application of this operator to canonical basis.

Advantages and disadvantages of DW

- No calculation of the spectrum of $\mathbf{L}$ required, cost-efficient;
- Multiresolution analysis, the resulting wavelets are orthogonal;
- Pure numerical linear algebra in the computation of the wavelets, outcome depends on algorithmical parameters.

Diffusion operator

.

A diffusion operator $\mathbf{T}$ on G is defined as a linear operator $\mathbf{T} : \mathcal{L}(V) \to \mathcal{L}(V)$ (or equivalently as a matrix in $\mathbb{R}^{n \times n}$) with the following additional properties suitable to obtain diffusion wavelets:

(i) It is symmetric.

(ii) Its spectral norm is bounded by $\|\mathbf{T}\|_2 \leq 1$

(iii) It is a local operator with respect to the graph distance, meaning that $(\mathbf{T}x)(\mathrm{v}_i)$ depends only on values of x at vertices in a small neighborhood around v_i.

(iv) Increasing powers $\mathbf{T}^j$ have an increasingly smaller ϵ-rank.

ϵ-rank of a matrix

Definition 4

Let $\varepsilon > 0$. Given vectors $\{x_1, \ldots, x_k\} \subset \mathbb{R}^n$, their ε-span is the set

$$\operatorname{span}_\varepsilon\{x_1, \ldots, x_k\} = \left\{ x \in \mathbb{R}^n \colon \exists\, c_1, \ldots, c_k \text{ such that } \left\| x - \sum_{i=1}^{k} c_i x_i \right\|_2 < \varepsilon \right\}.$$

For a matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, its ε-rank is defined as

$$\operatorname{rank}_\varepsilon(\mathbf{A}) := \min \left\{ k \in \mathbb{N} : \begin{array}{l} \exists x_1, \ldots, x_k \in \mathbb{R}^n \text{ such that the columns of} \\ \mathbf{A} \text{ are in the } \varepsilon\text{-span of } \{x_1, \ldots, x_k\} \end{array} \right\}.$$

Idea: the ε-rank assumption for increasing powers of $\mathbf{T}$ reflects the idea that repeated application of $\mathbf{T}$ leads to a smoothing of a signal.

Examples: $\mathbf{T} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$, $\mathbf{T} = e^{-t\mathbf{L}}$.

General idea

Once the diffusion operator $\mathbf{T}$ is selected, the repeated application of $\mathbf{T}$ generates a sequence of sequentially smoother signals

$$x,\ \mathbf{T}x,\ \mathbf{T}^2x,\ \mathbf{T}^4x,\ \cdots,\ \mathbf{T}^{2^{J-1}}x.$$

Idea: the powers $\mathbf{T}^{2^{j-1}}x$ correspond to different diffusions of x, and make it possible to differentiate between different resolutions.

Diffusion wavelets are calculated numerically in terms of nested **scaling spaces** associated with the powers $\mathbf{T}^{2^{j-1}}$. For this, let

$$\mathcal{V}^{(0)} = \mathcal{L}^2(V),$$

and then calculate J increasingly smaller spaces

$$\mathcal{V}^{(0)} \supseteq \mathcal{V}^{(1)} \supseteq \mathcal{V}^{(2)} \supseteq \cdots \subseteq \mathcal{V}^{(J)}$$

such that the range of the operator $\mathbf{T}^{2^{j-1}}$ lies in the ε-span of the spaces $\mathcal{V}^{(j)}$. As the scale j increases, the dimensions of $\mathcal{V}^{(j)}$ decreases, providing a compact representation of the diffusion operators at coarser scales.

Example of diffusion operator acting on a unit vector

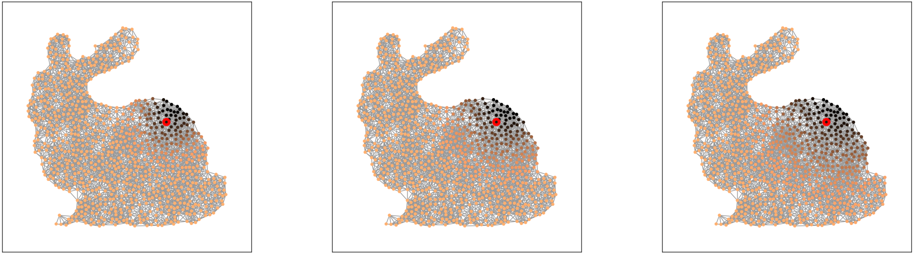


Figure 4: Illustration of the heat diffusion operator $\mathbf{T}^j = e^{-\frac{j}{2}\mathbf{L}}$ on the bunny graph for the scales $j \in \{1, 2, 4\}$ acting on a canonical basis vector. The red ring indicates the node at which the basis vector is localized.

Construction of diffusion scaling functions

We assume that the spaces $\mathcal{V}^{(j)}$ have dimension n_j and that a respective orthonormal basis $\{\phi_{j,k} \;:\; k \in \{1,\ldots,n_j\}\}$ of $\mathcal{V}^{(j)}$ is encoded in the columns of a matrix $\mathbf{\Phi}_j \in \mathbb{R}^{n \times n_j}$ satisfying $\mathbf{\Phi}_j^* \mathbf{\Phi}_j = \mathbf{I}_{n_j}$.

- Intialization: set $\mathbf{\Phi}_0 = \mathbf{I}_n$ and $n_0 = n$, i.e., the initial basis functions $\phi_{0,k} = \delta_{\mathrm{v}_k}$, $k \in \{1,\ldots,n\}$ are the canonical ONB of $\mathcal{L}^2(V)$.
- Step $j = 1$: consider the range of $\mathbf{T}_1 = \mathbf{T}^{2^0} = \mathbf{T}$. Using the QR decomposition with column pivoting, we can then extract an ONB

  $$\mathbf{Q}_1 \in \mathbb{R}^{n_0 \times n_1}$$

  such that the columns of $\mathbf{T}$ are in the ε-span of the space spanned by the columns of $\mathbf{Q}_1$. The columns of $\mathbf{\Phi}_1 = \mathbf{Q}_1$ form a basis of the space $\mathcal{V}^{(1)}$.

Construction of diffusion scaling functions

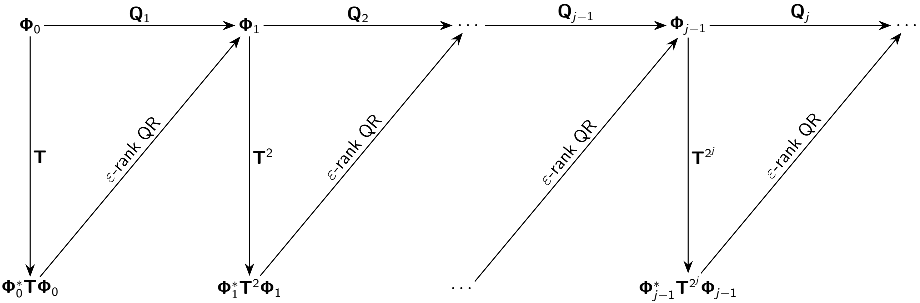
- Step $j-1 \to j$: assume that the basis $\mathbf{\Phi}_{j-1}$ and the matrix $\mathbf{T}_{j-1} \in \mathbb{R}^{n_{j-1} \times n_{j-1}}$ have been calculated and that $\mathbf{T}_{j-1}$ corresponds approximately to the representation $\mathbf{\Phi}_{j-1}^* \mathbf{T}^{2^{j-1}} \mathbf{\Phi}_{j-1}$ of the power $\mathbf{T}^{2^{j-1}}$ in the basis $\mathbf{\Phi}_{j-1}$ of $\mathcal{V}^{(j-1)}$. We consider now the square $\mathbf{T}_{j-1}^2$ of the matrix $\mathbf{T}_{j-1}$ with the idea that $\mathbf{T}_{j-1}^2$ is an approximate representation of $\mathbf{\Phi}_{j-1}^* \mathbf{T}^{2^j} \mathbf{\Phi}_{j-1}$ in the basis $\mathbf{\Phi}_{j-1}$. Using a QR decomposition with column pivoting, we can then extract a new ONB

$$\mathbf{Q}_j \in \mathbb{R}^{n_{j-1} \times n_j}$$

such that the columns of T_{j-1}^2 are in the ε-span of the space spanned by the columns of $\mathbf{Q}_j$. The scaling functions spanning $\mathcal{V}^{(j)}$ are then given as the columns of

$$\mathbf{\Phi}_j = \mathbf{\Phi}_{j-1} \mathbf{Q}_j.$$

Diagram of the numerical scheme for the diffusion wavelets



where we are using

$$\mathbf{T}_{j-1}^2 = (\mathbf{Q}_{j-1}^*\mathbf{T}_{j-2}^2\mathbf{Q}_{j-1})^2 \in \mathbb{R}^{n_{j-1}\times n_{j-1}}$$

as a substitute of

$$\mathbf{\Phi}_{j-1}^*\mathbf{T}^{2^j}\mathbf{\Phi}_{j-1}$$

The QR decomposition for the calculation of the basis $\mathbf{\Phi}_j$

The applied QR decomposition to obtain $\mathbf{\Phi}_j$ is a localized QR decomposition with a tolerance ε that removes directions associated with small singular values. The resulting basis functions $\phi_{j,k}$, $k \in \{1, \ldots, n_j\}$ encoded in the columns of $\mathbf{\Phi}_j$ are referred to as diffusion scaling functions.

The QR-decomposition is performed in such a way that these functions are localized on the graph, with supports that grow as the scale increases. The new basis allows to compress the square matrix $\mathbf{T}_{j-1}^2$ even further, and we obtain as a new representation the matrix

$$\mathbf{T}_j = \mathbf{Q}_j^* \mathbf{T}_{j-1}^2 \mathbf{Q}_j \in \mathbb{R}^{n_j \times n_j}.$$

Example of diffusion scaling functions

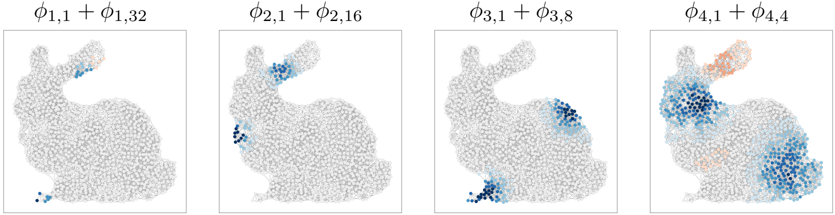


Figure 5: Illustration of diffusion scaling functions $\phi_{j,k}(\mathrm{v})$ on the bunny graph for different scales j and different positions k. The diffusion is based on the operator $\mathbf{T} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ (symmetric random walk on the graph).

Diffusion wavelet spaces

The diffusion wavelet spaces are defined as the orthogonal complements

$$\Delta\mathcal{V}^{(j)} = \mathcal{V}^{(j-1)} \ominus \mathcal{V}^{(j)}.$$

These spaces capture the detail information when passing from a scale $j-1$ to a coarser scale j. The orthonormal basis

$$\{\psi_{j,k} \;:\; k \in \{1,\ldots,n_{j-1}-n_j\}\}$$

of the detail space $\Delta\mathcal{V}^{(j)}$ is referred to as a diffusion wavelet basis.

The full diffusion wavelet decomposition can be written as

$$\mathcal{L}^2(V) = \mathcal{V}_J \oplus \bigoplus_{j=1}^{J} \Delta\mathcal{V}^{(j)},$$

where J denotes the coarsest scale considered.

Calculation of diffusion wavelets

To compute the diffusion wavelets, we require the orthogonal complement

$$\mathbf{Q}_j^{\perp} \in \mathbb{R}^{n_{j-1} \times (n_{j-1} - n_j)}$$

of the columns in the matrix $\mathbf{Q}_j^{\perp}$. These can, for instance, be obtained in an additional step when calculating $\mathbf{Q}_j^{\perp}$ via the QR decomposition. The diffusion wavelets $\psi_{j,k}$, $k \in \{1, \ldots, n_{j-1} - n_j\}$ at scale j are then given as the columns of the matrices

$$\mathbf{\Psi}_j = \mathbf{\Phi}_{j-1} \mathbf{Q}_j^{\perp}.$$

Algorithm 1: Generation of diffusion wavelets

Input: $\mathbf{\Phi}_0 = \mathbf{I}_n$, $n_0 = n$, number J of wavelets levels.

for $j = 1$ *to* J **do**

Calculate square matrix $\mathbf{M} = \mathbf{T}_{j-1}^2 \in \mathbb{R}^{n_{j-1} \times n_{j-1}}$ ($\mathbf{M} = \mathbf{T}$ if $j = 1$).
Calculate QR factorization of $\mathbf{M}$ (with column pivoting) such that the first n_j columns in the orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n_{j-1} \times n_{j-1}}$ ϵ-span the range of $\mathbf{M}$. Split the matrix $\mathbf{Q}$ in $\mathbf{Q}_j$ (the first n_j columns) and $\mathbf{Q}_j^{\perp}$ (the last $n_{j-1} - n_j$ columns).
Calculate the new representation

$$\mathbf{T}_j = \mathbf{Q}_j^* \mathbf{M} \mathbf{Q}_j \in \mathbb{R}^{n_j \times n_j}.$$

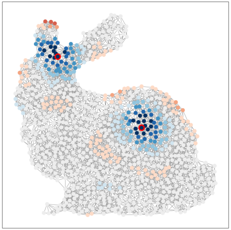
Calculate the new scaling functions and diffusion wavelets

$$\mathbf{\Phi}_j = \mathbf{\Phi}_{j-1} \mathbf{Q}_j, \quad \mathbf{\Psi}_j = \mathbf{\Phi}_{j-1} \mathbf{Q}_j^{\perp}.$$

Output: The diffusion scaling functions $\phi_{j,k}$ and the diffusion wavelets $\psi_{j,k}$ as columns of the matrices

$$\mathbf{\Phi}_j \in \mathbb{R}^{n \times n_j}, \quad \mathbf{\Psi}_j \in \mathbb{R}^{n \times (n_{j-1} - n_j)}, \; j \in \{1, \ldots, J\}.$$

Thanks a lot for your attention!



The main articles related to SGWT and DWs:

[1] Coifman, R., Maggioni, M.: Diffusion wavelets. Appl. Comput. Harmon. Anal. **21**, 53–94 (2006)

[2] Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Appl. Comput. Harmon. Anal. **30**(2), 129–150 (2011)

General overview:

[3] Ortega, A.: Introduction to Graph Signal Processing. Cambridge University Press, Cambridge (2022)