Image Processing with Optimally Designed Parabolic Partial Differential Equations

Qiuyi Wu

Advised by Dr. Xing Qiu

Department of Biostatistics and Computational Biology

Feb 2, 2024

Motivation and Background

Motivation: Image Denoising Analysis

Original

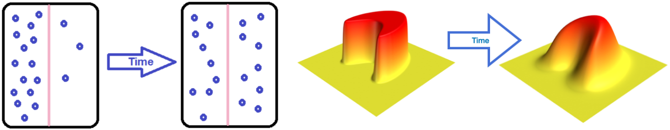
Denoised

Heat Equation in Image Processing

The diffusion equation often arises from physical phenomena. It is usually presented in a general form

$$\partial_t u(\mathbf{x}, t) = \nabla \cdot (g(\mathbf{u}, \mathbf{x}, t) \nabla u(\mathbf{x}, t))$$

- $u(\mathbf{x}, t)$ is a spatio-temporal function, e.g., a temperature field that changes in time
- Diffusivity $g(\mathbf{u}, \mathbf{x}, t)$ depends on the model
- If g depends on u: e.g.: Perona-Malik equation (nonlinear)
- If $g \equiv 1$: heat equation (linear)



Heat Equation & Kernel Smoothing

Consider the proverbial rod of infinite length. Denote temperature of the rod at a point x and time t by $\phi(x,t)$. If the initial temperature is described by a function f(x), the temperature at (x,t) is determined by the heat equation:

$$\phi_t(x,t) = \phi_{xx}(x,t)$$

with initial condition $\phi(x,0) = f(x)$

The solution of the heat equation, i.e. Strauss (1992)

$$\phi(x,t) = \frac{1}{\sqrt{4\pi t}} \int_{-\infty}^{\infty} f(z) \exp\left(-\frac{(x-z)^2}{4t}\right) dz$$

Heat Equation & Kernel Smoothing

The solution of the heat equation

$$\phi(x,t) = \frac{1}{\sqrt{4\pi t}} \int_{-\infty}^{\infty} f(z) \exp\left(-\frac{(x-z)^2}{4t}\right) dz$$

Gaussian kernel density estimate

$$\phi(x,t) = \frac{1}{\sqrt{4\pi t}n} \sum_{i=1}^{n} f(x_i) \exp\left(-\frac{(x-X_i)^2}{4t}\right)$$

This solution is immediately recognizable as a Gaussian kernel density estimate with the value $\sqrt{2t}$ playing the role of the bandwidth. As time passes and the heat diffuses through the rod, its distribution at time t is precisely given by a Gaussian kernel density estimate. This diffusion interpretation of kernel smoothing is conventional in the imaging literature where partial differential equation methods are commonplace.

Connection Between Generalized KS & Evolutionary PDE

Evolutionary PDE & Kernel Smoothing

The optimal smoothing parameters converges to zero when $n \to \infty$. Consequently, we can use infinitesimal techniques to establish an *approximate equivalence* between a noncentral, varying bandwidth Gaussian kernel smoothing and a corresponding general linear parabolic PDE for $t \approx 0$.

Functional kernel smoothing

$$\Psi(u_0, \boldsymbol{\delta}, s\Sigma_0)(\mathbf{x}) := \int_\Omega K(\mathbf{x}, \mathbf{y}|\boldsymbol{\delta}, s\Sigma_0) u_0(\mathbf{y}) d\mathbf{y}$$

Here $K(\cdot, \cdot)$ is a varying bandwidth Gaussian kernel with differentiable mean field $\boldsymbol{\delta}$ and differentiable covariance field $\Sigma(\mathbf{x}) = s(\mathbf{x})\Sigma_0(\mathbf{x})$, assuming that $|\Sigma_0(\mathbf{x})| \equiv 1$ for all $\mathbf{x} \in \Omega$.

The input data here is considered as a spatial function, not a set of n discrete observations.

Connections between PDE and Kernel Smoothing

Theorem (Asymptotic equivalence between parabolic partial differential equations and general kernel smoothing)

Let $\Psi(u_0, \boldsymbol{\delta}, s\Sigma_0)(\mathbf{x})$ *be the results of applying a general* *Gaussian kernel smoothing* *to* $u_0(\mathbf{x})$. *Let* L *be an elliptical operator and* $e^{tL}u_0(\mathbf{x})$ *be the solution of* *parabolic PDE* $\partial_t u = Lu$ *observed at time* t *with initial condition* $u_0(\mathbf{x})$. *For* $s, t \approx 0$, *we have*

$$e^{tL}u_0 = \Psi(u_0, \boldsymbol{\delta}, s\Sigma_0) + O(t^2)$$

if

$$\begin{cases} t = s/2, \quad A = \Sigma_0 + \boldsymbol{\delta}\boldsymbol{\delta}', \quad \mathbf{b} = \frac{2\delta}{s}. \\ s = 2t, \quad \boldsymbol{\delta} = t\mathbf{b}, \quad \Sigma_0 = A - t^2\mathbf{b}\mathbf{b}'. \end{cases}$$

Empirical Equivalence

Discretization of Kernel Smoothing

$$\int_{\Omega} K(\mathbf{x}, \mathbf{y}) u_0(\mathbf{y}) d\mathbf{y} = \frac{\sum_{\mathbf{i}} K_n(\mathbf{x}, \mathbf{y}_i) \mathbf{Y}_i}{\sum_{\mathbf{i}} K_n(\mathbf{x}, \mathbf{y}_i)} + O(n^{-1}).$$

The discretization of kernel smoothing, which involves integral/summation operations, demonstrates robustness to measurement errors.

Discretization of PDE

The discretization of PDE faces challenges due to the numerical estimation of partial derivatives.

- As $n \to \infty$, the numerical approximation of partial derivatives may fail to converge to the corresponding smooth function (FDM).
- Need to transform PDE into an integral equation (FEM).

Weak Formulation of Diffusion Equations

We want to find a solution trajectory $u(\mathbf{x}, t)$, such that at every $t \in [0, T]$, $u(\mathbf{x}, t) \in H_0(\Omega)$, and

$$\begin{cases} \frac{\partial u(\mathbf{x},t)}{\partial t} = Lu(\mathbf{x}, t) := div\left(g(\mathbf{x})\nabla u(\mathbf{x}, t)\right), & \mathbf{x} \in \Omega, t \in (0, T]. \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, t = 0. \end{cases}$$

One weak formulation is stated as below. For an arbitrary test function $v(\mathbf{x})$,

$$\int_\Omega \frac{\partial u}{\partial t} v = \int_\Omega \mathrm{div}\left(g\nabla u\right) v = \underbrace{\int_{\partial\Omega} v \left\langle g\nabla u,\ \vec{n}\right\rangle}_{\text{boundary term}} - \int_\Omega (\nabla u)^T g \nabla v.$$

e.g.: for 1D case, $\Omega = [0, 1], g = 1$:

$$\int_0^1 u''(x)v(x)dx = u'(x)v(x)|_0^1 - \int_0^1 u'(x)v'(x)dx$$

Basis Representation

$$u(\mathbf{x}, t) = \sum_{j=1}^{\infty} a_j(t)\tilde{\phi}_j(\mathbf{x}) = \mathbf{a}^T(t)\tilde{\boldsymbol{\phi}}(\mathbf{x}).$$

$$v(\mathbf{x}) = \sum_{j=1}^{\infty} b_j\tilde{\phi}_j(\mathbf{x}) = \mathbf{b}^T\tilde{\boldsymbol{\phi}}(\mathbf{x}).$$

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \sum_{j=1}^{\infty} c_j(t)\tilde{\phi}_j(\mathbf{x}) \approx \mathbf{c}^T(t)\tilde{\boldsymbol{\phi}}(\mathbf{x}).$$

Diffusivity field $g(\mathbf{x})$ and weighted stiffness matrix $K \in M_{B\times B}$

$$g(\mathbf{x}) \approx \sum_{l=1}^{\tilde{B}} G_{\cdot\cdot,l}\phi_l(\mathbf{x}).$$

$$K_{jj'} := \int_\Omega \nabla\tilde{\phi}_j(\mathbf{x})^T g(\mathbf{x})\nabla\tilde{\phi}_{j'}(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{d}\sum_{i'=1}^{d}\sum_{l=1}^{B} G_{ii',l}\int_\Omega \phi_l(\mathbf{x})\cdot\frac{\partial\tilde{\phi}_j(\mathbf{x})}{\partial x_{i'}}\cdot\frac{\partial\tilde{\phi}_{j'}(\mathbf{x})}{\partial x_i}d\mathbf{x}.$$

Weak Formulation of Diffusion Equations

A direct approach is based on the *complete* Green's formula:

$$\int_\Omega \frac{\partial u}{\partial t} v = \int_\Omega \operatorname{div}(g\nabla u)\, v = \underbrace{\int_{\partial\Omega} v \langle g\nabla u,\ \vec{n}\rangle}_{\text{boundary term}} - \int_\Omega (\nabla u)^T g \nabla v.$$

Using the complete piecewise linear basis, it leads to the following matrix equation

$$\forall \mathbf{b} \in \mathbb{R}^{B+2}, \quad \mathbf{b}^T J\mathbf{c}(t) = \mathbf{b}^T \partial K\, \mathbf{a}(t) - \mathbf{b}^T K\mathbf{a}(t) = \mathbf{b}^T (\partial K - K)\, \mathbf{a}(t).$$

Simulation

Simulation: 1D cases

We conduct the 3 simulations to study the equivalence between kernel smoothing and diffusion partial differential equation.

1. The true spatial function is $u(x) = 2\sin(4x)$ defined on $\Omega = [0, 1]$.
2. A total number of $n = 201$ discrete data are observed on an equi-distance grid on $[0, 1]$ with $i.i.d$ noise sampled from a standard normal distribution.

$$U_i = u(x_i) + \epsilon_i, \quad x_i = (i-1)\delta x, \quad \epsilon_i \sim N(0, 1)$$

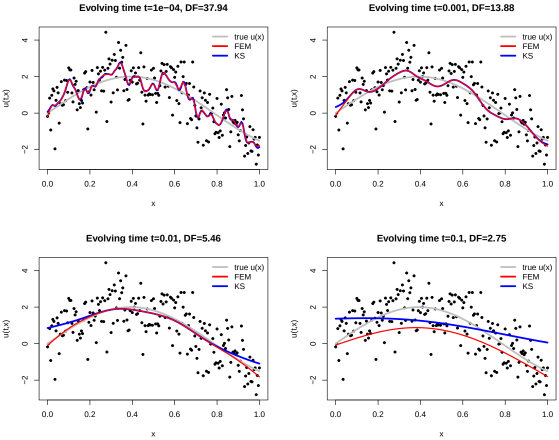
3. Four choices of evolving times: $t \in \{0.0001, 0.001, 0.01, 0.1\}$.

Simulation: 1D cases

1. For Simulation I(a):
   - Complete basis system: 101 piecewise linear functions defined on a partition of [0, 1] with 100 sub-intervals with equal length.
   - Constant diffusivity $g(x) \equiv \frac{1}{2}$ is used in the diffusion PDE.
2. For Simulation I(b):
   - Complete Basis system: 6 piecewise linear functions defined on the following partition of Ω: [0, 1/6, 1/3, 1/2, 3/4, 1].
   - Constant diffusivity $g(x) \equiv \frac{1}{2}$ is used in the diffusion PDE.
3. For Simulation I(c):
   - Complete basis system: 101 piecewise linear functions defined on a partition of [0, 1] with 100 sub-intervals with equal length.
   - Non-constant diffusivity $g(x) = \max(1 - 10(x - 0.4)^2, 0.1)$ is used.

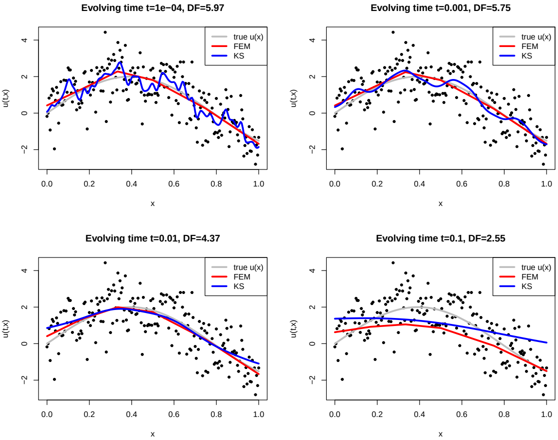
Simulation I(a)

101 piecewise linear functions; $g(x) \equiv \frac{1}{2}$.



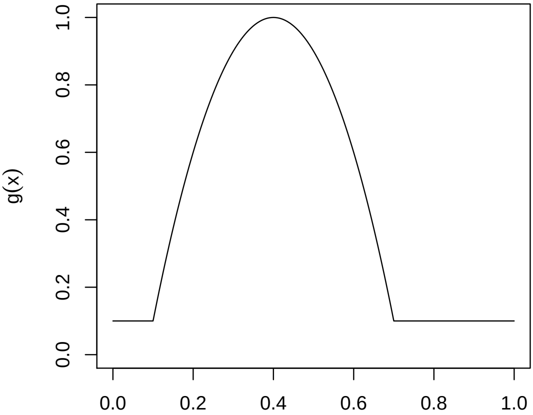
Simulation I(b)

6 piecewise linear functions; $g(x) \equiv \frac{1}{2}$.



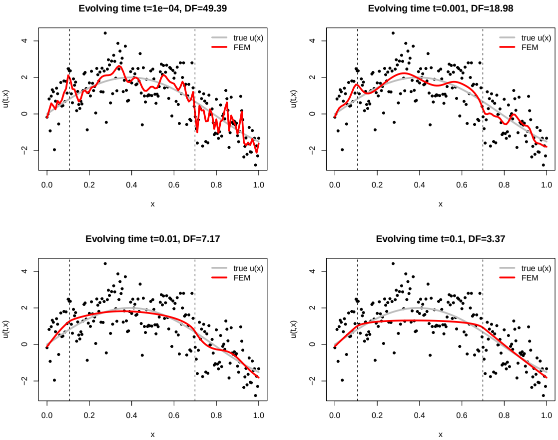
Simulation I(c)

Non-constant diffusivity $g(x) = \max(1 - 10(x - 0.4)^2, 0.1)$



Simulation I(c)

101 piecewise linear functions; $g(x) = \max(1 - 10(x - 0.4)^2, 0.1)$.



Simulation: 2D cases

We conduct 3 simulations to study in 2D. For the first two simulations:

1. Four choices of evolving times: $t \in \{0.0001, 0.001, 0.01, 0.1\}$.
2.
    1. Simulation II(a):
        - The true spatial function is $Z = \sin(5X)\cos(5Y)$ on $[0,1] \times [0,1]$.
        - $n = 30$ and $n = 50$ discrete data on an equi-distance grid

$$U_{ij} = Z(x_i, y_j) + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, 0.1). \tag{1}$$

        - 15 and 25 piecewise linear functions on the two directions.
    2. Simulation II(b):
        - The true spatial function is $Z = \mathcal{N}_x(3, 1, 0.5) + \mathcal{N}_y(3, 1, 0.5)$ defined on $\Omega = [0,6] \times [0,6]$.
        - $n = 100$ discrete data on an equi-distance grid on the two directions

$$U_{ij} = Z(x_i, y_j) + \epsilon_{ij} \quad \epsilon_{ij} \sim N(0, 0.01). \tag{2}$$

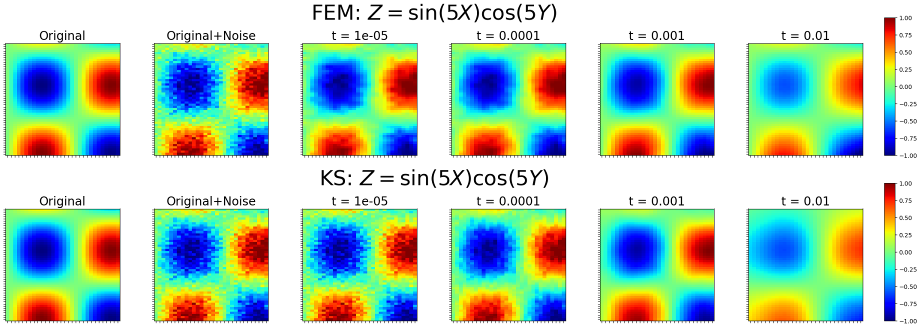
        - On both directions, 50 piecewise linear functions

Simulation: 2D cases

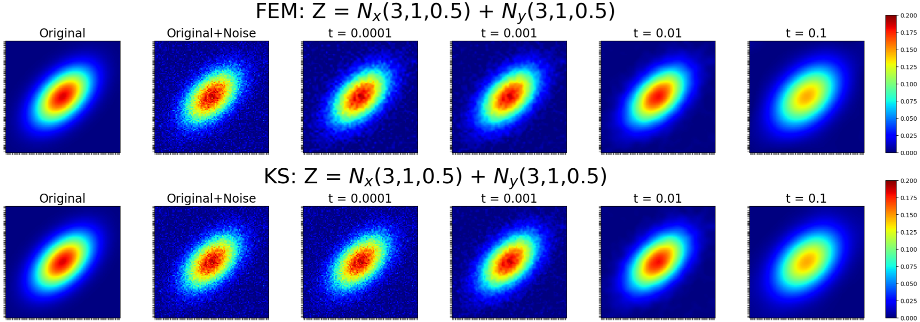
For Simulation II(c), we use Handwritten Digits based on NIST.

1. 1700 images of digits (0 through 9) consist of 8×8 pixels.
2. Evolving times: $t \in [0, 0.3]$.
3. We use constant diffusivity, $g(x) \equiv 1$, in the diffusion PDE.
4. We add two different noise levels to the original digits $\sigma_\epsilon^2 = 0.15, 0.3$.
5. LogitBoost classification algorithm is applied on the diffused digit images to exam how accurate the classification task performs after applying our diffusion method.

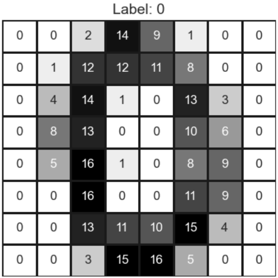
Simulation II(a)



Simulation II(b)

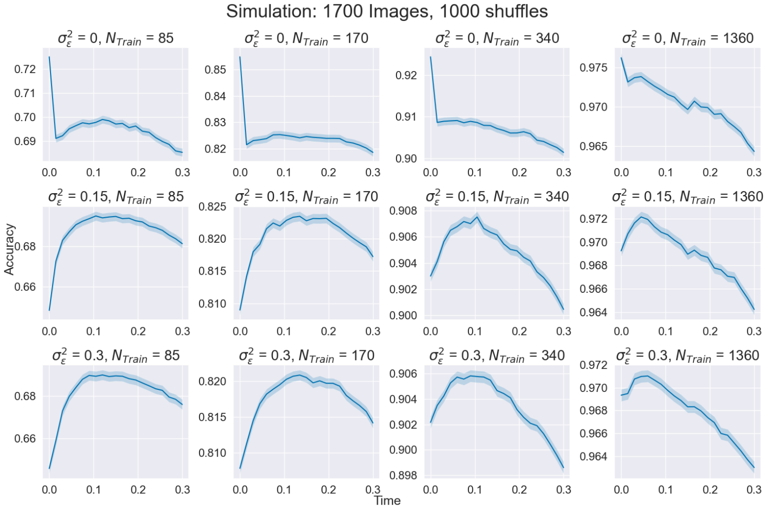


Simulation II(c)

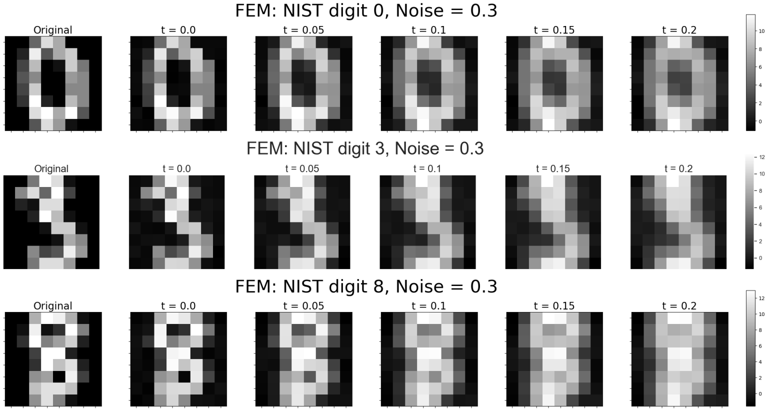


- 1700 NIST 8×8 digit images, 1000 shuffles
- Noise level (σ^2): 0, 0.15, 0.3
- # of training images: 85 (5%), 170 (10%), 340 (20%), 1360 (80%)

Simulation II(c)

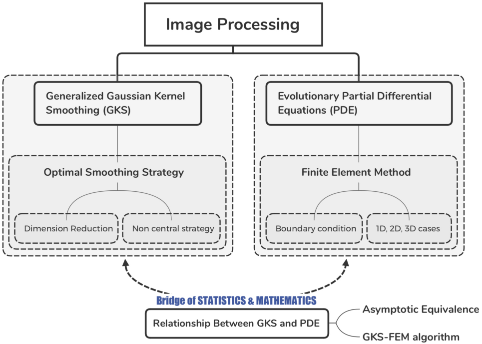


Simulation II(c)



Conclusion

Key Takeaways



- We establish mathematical equivalence between general kernel smoothing and a general parabolic partial differential equations
- In mathematics, we are given a PDE to study its properties. In statistics, we estimate the PDE system to achieve some goal.
- Compared with traditional numerical methods in PDE such as the finite difference method (FDM), FEM is known to be faster and numerically more stable.

Reference

1. LeqinWu, Xing Qiu, Ya-xiang Yuan, and HulinWu, Parameter estimation and variable selection for big systems of linear ordinary differential equations: A matrix-based approach, Journal of the American Statistical Association 114 (2019), no. 526, 657–667.
2. Consagra, William, Arun Venkataraman, and Xing Qiu. "Efficient Multidimensional Functional Data Analysis Using Marginal Product Basis Systems." arXiv preprint arXiv:2107.14728 (2021).
3. Karvonen, Toni, Fehmi Cirak, and Mark Girolami. "Error analysis for a statistical finite element method." arXiv preprint arXiv:2201.07543 (2022).

Appendix

Future Perspectives: Methodology

Higher Dimensional Scenario

- Extend our method to 3D and higher dimensional situations in the numerical implementation via Tensor Representation:

$$e^{tA} = \left(T^{(D)} \otimes \cdots \otimes T^{(1)}\right)\left(e^{t\Lambda^{(D)}} \otimes \cdots \otimes e^{t\Lambda^{(1)}}\right)\left({T^{(D)}}^{-1} \otimes \cdots \otimes {T^{(1)}}^{-1}\right)$$

- Truncate the eigenvalues to retain only the first few largest eigenvalues to improve the computational efficiency.

Future Perspectives: Methodology

Adaptive Smoothing

- Improve the algorithm to allows non-constant diffusivity.
  1. When $g(\mathbf{x})$ is diagonal matrix for all $\mathbf{x} \in \Omega$. In this case, $G_{\cdot\cdot,j}$ is always diagonal, and we have
     $K_{jj'} = \sum_{d=1}^{D} \sum_{l=1}^{\tilde{B}} G_{dd,l} \int_\Omega \tilde{\phi}_l(\mathbf{x}) \cdot \frac{\partial \phi_j(\mathbf{x})}{\partial x_d} \cdot \frac{\partial \phi_{j'}(\mathbf{x})}{\partial x_d} d\mathbf{x}$
  2. When $g(\mathbf{x})$ is scalar for all $\mathbf{x} \in \Omega$. In this case, $G_{\cdot\cdot,l} = G_l I_D$, $G_l \in \mathbb{R}$,
     $K_{jj'} = \sum_{l=1}^{\tilde{B}} G_l \int_\Omega \tilde{\phi}_l(\mathbf{x}) \cdot \sum_{d=1}^{D} \frac{\partial \phi_j(\mathbf{x})}{\partial x_d} \cdot \frac{\partial \phi_{j'}(\mathbf{x})}{\partial x_d} d\mathbf{x}$
  3. When $g(\mathbf{x}) := G$, which is a constant matrix that does not depend on $\mathbf{x}$:
     $K_{jj'} := \sum_{d=1}^{D} \sum_{d'=1}^{D} G_{dd'} \int_\Omega \frac{\partial \phi_j(\mathbf{x})}{\partial x_{d'}} \cdot \frac{\partial \phi_{j'}(\mathbf{x})}{\partial x_d} d\mathbf{x}$
  4. When $g(\mathbf{x}) \equiv G \cdot I_d$, where $G \in \mathbb{R}$ is a constant that does not depend on $\mathbf{x}$:
     $K_{jj'} := G \int_\Omega \sum_{d=1}^{D} \frac{\partial \phi_j(\mathbf{x})}{\partial x_d} \cdot \frac{\partial \phi_{j'}(\mathbf{x})}{\partial x_d} d\mathbf{x}, \qquad K = G \langle \nabla\phi,\ \nabla\phi \rangle_{L^2}$
- Allow the drift term in the current algorithm to achieve an optimal smoothing strategy.

Future Perspectives: Methodology

Optimal Smoothing Parameter Selection

- Generalized Cross-Validation (GCV) technique to select the optimal diffusion time parameter

$$\mathrm{GCV}(t|A, U, \Phi) = \frac{\mathrm{RSS}(t|A, U, \Phi)/n}{\left(1 - \frac{\mathrm{DF}(t|A)}{n}\right)^2} = n \cdot \frac{\mathrm{RSS}(t|A, U, \Phi)}{(n - \mathrm{DF}(t|A))^2}$$

We may use a data-driven way to select optimal t based on minimizing GCV.

Future Perspectives: Simulation & Application

Application 1: Constant Diffusivity Smoothing for TBI Study

- Adopt a previous TBI study by applying smoothing strategy with constant diffusivity, smoothing parameter selected via GCV, followed by the techniques proposed in the referenced study.
- The goal of this two-step process is to assess if this combined approach leads to enhanced results.

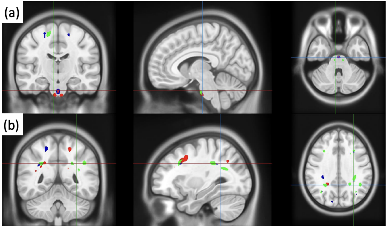


Figure 1: TBI study: two cross sections of the brain, with ROIs related to the identified eigenfunctions displayed in blue, red and green.

Future Perspectives: Simulation & Application

Application 2: Optimal Smoothing with HCP Training Data

- Derive the best smoothing parameters for anisotropic and non-constant smoothing strategy based on the mean and variance function computed from 900 subjects, then apply it to TBI data.
- The goal is to find the optimal non-constant diffusivity field by training a large dataset and then applying this optimal smoothing approach to real-life problems.

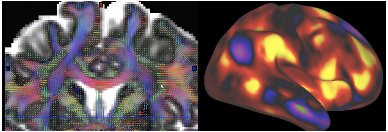


Figure 2: HCP Young Adult Study

Connections between PDE and Kernel Smoothing

Proof.

When $t, s \approx 0$, using "Taylor expansion":

$$\begin{aligned} e^{tL}u_0 &= u_0 + tLu_0 + O(t^{-2}) \approx u_0 + \operatorname{tr}\left(A\nabla^2 u\right) + b'\nabla u \\ &= u_0 + t\left(\sum_{i,j=1}^{d} a^{ij}\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{j=1}^{d} b^j \frac{\partial}{\partial x_j}\right)u_0. \\ \Psi(u_0, \Sigma) &= \int_\Omega K(\mathbf{x}, \mathbf{y})u_0(\mathbf{y})d\mathbf{y} \\ &= u_0(\mathbf{x}) + \nabla u_0(\mathbf{x})'\boldsymbol{\delta} + \frac{s}{2}tr\left(\left[\Sigma_0 + \frac{\boldsymbol{\delta}\boldsymbol{\delta}'}{s}\right]H_{\mathbf{x}}\right) + o(r^2) \\ &\approx u_0 + \frac{s}{2}\left(\sum_{i,j}\left[\Sigma_{0,ij} + \frac{\delta_i\delta_j}{s}\right]\frac{\partial^2}{\partial x_i \partial x_j} + \sum_i \frac{2\delta_i}{s}\frac{\partial}{\partial x_i}\right)u_0. \end{aligned}$$

□

Optimal Kernel Smoothing Strategy

To minimize MSE

$$\arg\min_{\{\delta_n,\Sigma\}} MSE = \{\delta_n^*=H_{\mathbf{x}}^{-1}\left(-\frac{c_n(s_n^*)^{-d/2}}{2Bias(\mathbf{x})}\cdot\nabla\sigma_\epsilon^2(\mathbf{x})-\nabla u^*(\mathbf{x})\right),$$

$$\Sigma^*=\frac{V(\mathbf{x})}{2Bias(\mathbf{x})}H_{\mathbf{x}}^{-1}=\frac{c_n(s_n^*)^{-d/2}\left(\sigma_\epsilon^2(\mathbf{x})+\nabla\sigma_\epsilon^2(\mathbf{x})'\boldsymbol{\delta}_n^*\right)}{2\nabla u^*(\mathbf{x})'\boldsymbol{\delta}_n^*+tr\left(H_{\mathbf{x}}\Sigma^*\right)+(\boldsymbol{\delta}_n^*)'H_{\mathbf{x}}\boldsymbol{\delta}_n^*}H_{\mathbf{x}}^{-1}\}$$

$$=\{\delta_{\mathbf{n}}^* = \kappa u + \nu_0, \quad \Sigma^* = s_n^*|H_x|^{1/d}\cdot H_x^{-1}\}$$

$$\kappa := -\frac{c_n(s_n^*)^{-d/2}}{2\mathrm{Bias}(x)}, \qquad u = H_x^{-1}\nabla\sigma_\epsilon^2(x), \qquad v_0 = H_x^{-1}\nabla u^*(x).$$

δ_n^*: implies that delta has only one degree of freedom because it lives on a one-dimensional affine subspace.

Σ^*: As a result, there is only one unknown parameter (s0) in sigma that needs to be determined from the gradient equations.

Degrees of Freedom

Inspired by the degrees of freedom defined for ridge regression and roughness penalized splines, we define the degrees of freedom associated with the weak formulation of diffusion equation as

$$\mathrm{DF}(t|A) := \mathrm{tr}\left(P_t\right) = \mathrm{tr}\left(e^{t\tilde{A}}\left(\tilde{\Phi}^T\tilde{\Phi}\right)^{-1}\tilde{\Phi}^T\tilde{\Phi}\right) = \mathrm{tr}\left(e^{t\tilde{A}}\right) = \mathrm{tr}\left(e^{t\Lambda}\right) = \sum_{j=1}^{B+2} e^{t\lambda_j}$$

Notes

1. $\mathrm{DF}(t|A)$ not only depends on the diffusion equation (L), but also the implementation of the numerical solver. Specifically, it depends on the basis system used in the solver, which has great influence on J and K and in turn A.
2. Number of basis functions $(B+2)$ determines the upper bound of the degrees of freedom:
$$\mathrm{DF}(0, A) = B + 2, \qquad \mathrm{DF}(t|A) \leqslant B + 2.$$
3. When $t \to \infty$ DF monotonically decreases to 2:
$$\mathrm{DF}(t|A) < \mathrm{DF}(s, A), \quad \text{for all } t > s. \qquad \lim_{t\to\infty} \mathrm{DF}(t|A) = 2.$$

General Linear Parabolic PDE

Let $\Omega \subset \mathbb{R}^d$ be a compact spatial domain, let $[0, T]$ be a time interval. A general linear parabolic partial differential equation defined on $\Omega \times [0, T]$ is

$$\begin{cases} \dfrac{\partial u(t,\mathbf{x})}{\partial t} = L\big(u(t,\mathbf{x})\big), \quad t > 0, \\ u(0,\mathbf{x}) = u_0(\mathbf{x}). \end{cases} \tag{3}$$

Here $L(\cdot) : W^{2,2}(\Omega) \to L^2(\Omega)$ is an *linear differential operator*

$$Lu = \sum_{i,j=1}^{d} a^{ij}(t,\mathbf{x}) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^{d} b^j(t,\mathbf{x}) \frac{\partial u}{\partial x_j} + c(t,\mathbf{x})u + f(t,\mathbf{x}).$$

$$\sum_{i,j} a^{ij}(t,\mathbf{x}) \xi_i \xi_j > 0, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^d, \boldsymbol{\xi} \neq 0_d.$$

General Linear Parabolic PDE

A popular family of parabolic PDE used in image analysis is the **general heat flow** with diffusivity field $g(\mathbf{x}) \in C^2(\Omega)$:

$$Lu(\mathbf{x}, t) = \operatorname{div}\left(g(x)\nabla u(x, t)\right) = \sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\left[g^{ij}(x)\frac{\partial u}{\partial x_j}\right]$$

Note

- $g(\cdot)$ does not depend on ∇u
- $g(\cdot)$ does not have to be a scalar matrix
- The larger the $g(\cdot)$, the quicker the smoothing

General Parabolic PDE

In this study, we will focus on a more flexible family of elliptical operators $L(\cdot)$ with arbitrary first and second order terms but no $c(t, \mathbf{x})$ and $f(t, \mathbf{x})$.

$$
\begin{aligned}
Lu = \operatorname{tr}\left(A\nabla^2 u\right) + b'\nabla u = \sum_{i,j=1}^{d} a^{ij}(t,x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^{d} b^j(t,x)\frac{\partial u}{\partial x_j}. \\
A = A(t,\mathbf{x}) = \left[a^{ij}(t,\mathbf{x})\right], \\
\mathbf{b} = \mathbf{b}(t,\mathbf{x}) = \left(b^j(t,\mathbf{x})\right), \\
\nabla^2 = \left[\frac{\partial^2}{\partial x_i \partial x_j}\right].
\end{aligned}
\tag{4}
$$

Green's Function and Smoothing Kernel

$$Lu = \sum_{i,j=1}^{d} a^{ij}(t,\mathbf{x}) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{j=1}^{d} b^j(t,\mathbf{x}) \frac{\partial u}{\partial x_j}$$

The **fundamental solutions** of the equation above is the (generalized) solution with initial condition $\delta(\mathbf{x}-\mathbf{y})$, which is a Dirac-delta function centered at $\mathbf{y} \in \Omega$. We denote this solution trajectory by $K(\mathbf{x},\mathbf{y},t)$. The most important property of the Dirac-delta function is

$$\int \delta(\mathbf{x}-\mathbf{y}) u_0(\mathbf{y}) d\mathbf{y} = u_0(\mathbf{x}). \tag{5}$$

In other words, $\delta(\mathbf{x}-\mathbf{y})$ is a family of initial conditions indexed by $\mathbf{y}$; $K(\mathbf{x},\mathbf{y},t)$ are the corresponding solution trajectories.

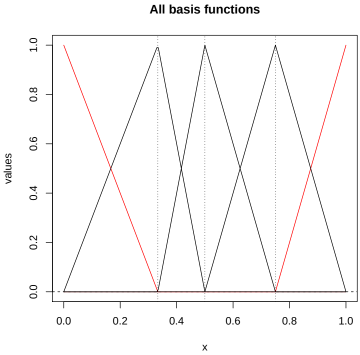
Green's Function and Smoothing Kernel

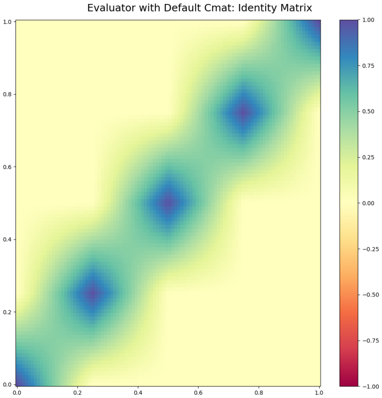
Now let's assume that an initial condition function $u_0(\cdot)$ is a family of **linear** coefficients (i.e. $a(\mathbf{y})$). Using the linearity and green's function, the solution can be represented by the following convolution (kernel smoothing)

$$u(t, \mathbf{x}) = \int K(\mathbf{x}, \mathbf{y}, t) u_0(\mathbf{y}) d\mathbf{y}. \tag{6}$$

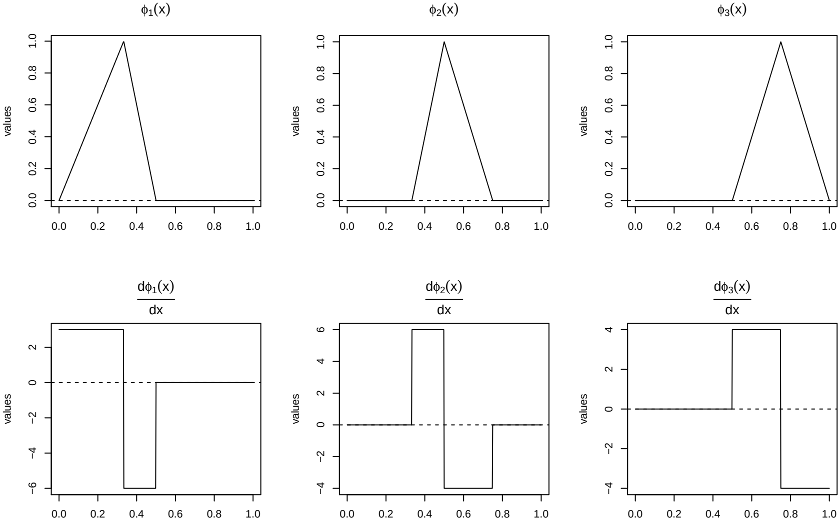
Unfortunately, in general (other than the classical heat equation) we do not know the analytic form of $K(\mathbf{x}, \mathbf{y}, t)$. We will need to derive an approximate kernel for it.

Basis Function for 1D and 2D

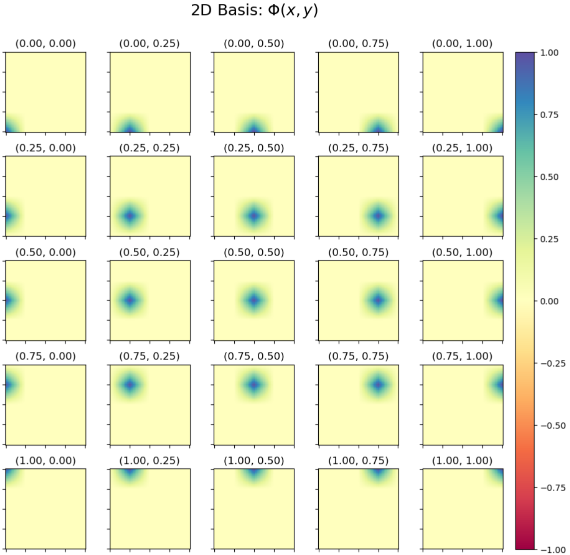




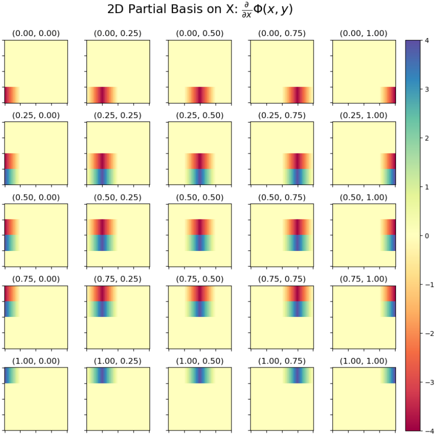
1D Basis Function & Their Derivatives

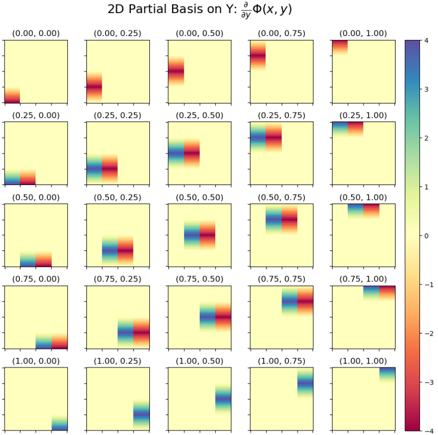


2D Basis Function & Their Derivatives

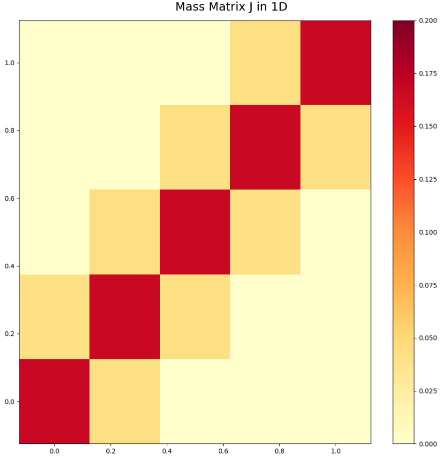


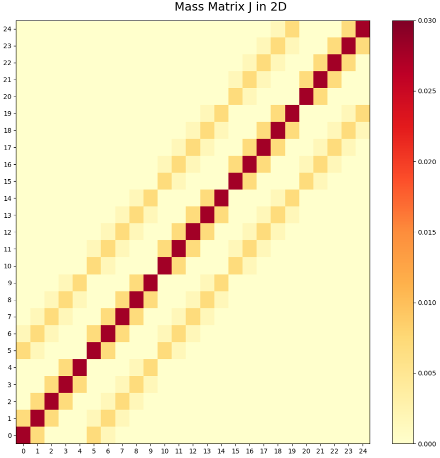
2D Basis Function & Their Derivatives



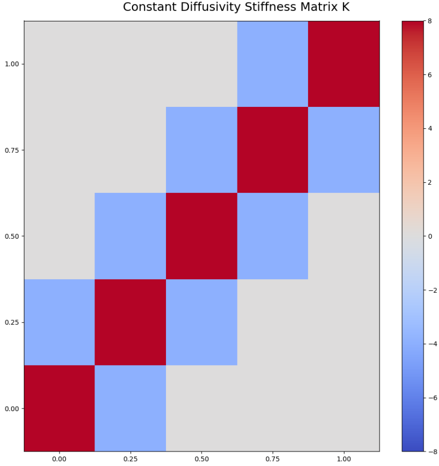


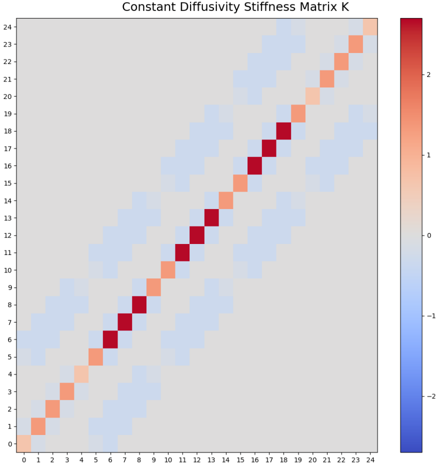
Inner Product Matrix J for 1D & 2D



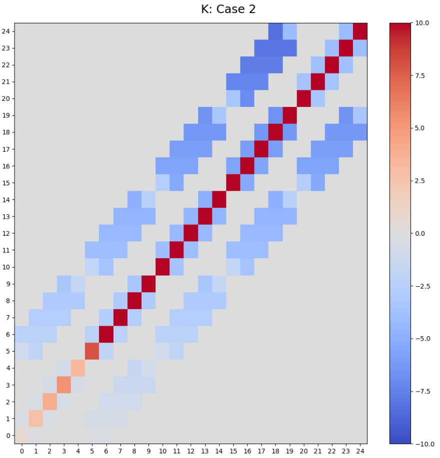


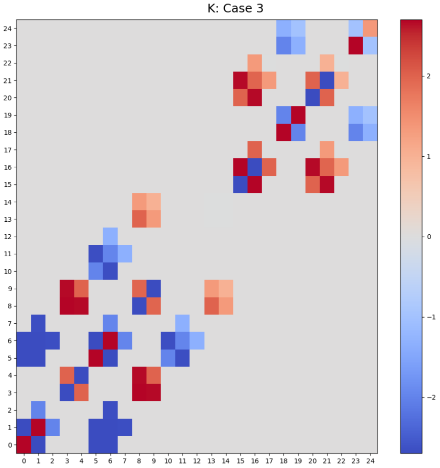
Stiffness Matrix K for 1D & 2D





2D Stiffness Matrix K with non-constant diffusivity





Computation for 2D Stiffness Matrix

The 2D unweighted stiffness matrix is

$$
\begin{aligned}
K_{\mathbf{j},\mathbf{l}} &:= \int_\Omega \langle \nabla \phi_{\mathbf{j}}(x_1, x_2)),\ \nabla \phi_{\mathbf{l}}(x_1, x_2) \rangle_{\mathbb{R}^2}\, dx_1 dx_2 \\
&= \int_0^1 \phi_{j_1}^{(1)'}(x_1)\phi_{l_1}^{(1)'}(x_1)\mathrm{d}x_1 \int_0^1 \phi_{j_2}^{(2)}(x_2)\phi_{l_2}^{(2)}(x_2)\mathrm{d}x_2 \\
&\quad + \int_0^1 \phi_{j_1}^{(1)}(x_1)\phi_{l_1}^{(1)}(x_1)\mathrm{d}x_1 \int_0^1 \phi_{j_2}^{(2)'}(x_2)\phi_{l_2}^{(2)'}(x_2)\mathrm{d}x_2 \\
&= K_{j_1 l_1}^{(1)} J_{j_2 l_2}^{(2)} + J_{j_1 l_1}^{(1)} K_{j_2 l_2}^{(2)}. \\
K &= J^{(2)} \otimes K^{(1)} + K^{(2)} \otimes J^{(1)}.
\end{aligned}
$$

Computation for 2D Boundary Effect Matrix

The 2D unweighted boundary effect matrix is

$$
\begin{aligned}
\partial K_{\mathbf{j},\mathbf{l}} &= \int_{\partial\Omega} \langle \nabla\phi_{\mathbf{l}}(\mathbf{x}),\ \vec{n}(\mathbf{x})\rangle_{\mathbb{R}^2}\, \phi_{\mathbf{j}}(\mathbf{x}) d\mathbf{x} \\
&= \int_{\partial\Omega_1} n(x_1)\phi_{l_1}^{(1)'}(x_1)\phi_{j_1}^{(1)}(x_1)\mathrm{d}x_1 \cdot \int_{\Omega_1} \phi_{l_2}^{(2)}(x_2)\phi_{j_2}^{(2)}(x_2)\mathrm{d}x_2 \\
&+ \int_{\partial\Omega_2} n(x_2)\phi_{l_2}^{(2)'}(x_2)\phi_{j_2}^{(2)}(x_2)\mathrm{d}x_2 \cdot \int_{\Omega_2} \phi_{l_1}^{(1)}(x_1)\phi_{j_1}^{(1)}(x_1)\mathrm{d}x_1 \\
&= \partial K_{j_1,l_1}^{(1)} J_{l_2,j_2} + \partial K_{j_2,l_2}^{(2)} J_{l_1,j_1}. \\
\partial K &= J^{(2)} \otimes \partial K^{(1)} + \partial K^{(2)} \otimes J^{(1)}.
\end{aligned}
$$