Lecture 23: Kernel PDE solvers Continued

Last time we saw that our representer formula for Generalized interpolation

$$u^n = \begin{cases} \arg\min \|u\|_{\mathcal{H}} \\ \text{s.t.} \quad \varphi(u) = \underline{Y} \end{cases}$$

$$u^n = K(\,\cdot\,,\varphi)K(\varphi,\varphi)^{-1}\underline{Y}$$

can be directly applied to obtain a collocation solver for liner PDEs

$$P(u)(\underline{x}) = -\operatorname{div} a(x)\nabla u(\underline{x}) + b(\underline{x})^T\nabla u(\underline{x}) + c(\underline{x})u(\underline{x})$$

$$\begin{cases} P(u) = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases} \xrightarrow{\text{approx}} \begin{cases} P(u)(\underline{x}_i) = f(\underline{x}_i), & i=1,\dots,m \\ u(\underline{x}_i) = 0, & i=m+1,\dots,M \end{cases}$$

simply by setting

$$\varphi_i := u \longmapsto P(u)(\underline{x}_i), \quad i=1,\dots,m$$

$$\varphi_i := u \longmapsto u(\underline{x}_i), \quad i=m+1,\dots,M.$$

At this point the main implementation detail

23 | Computational Details

of the method lies in constructing the vector field $K(\cdot,\varphi)$ & the matrix $K(\varphi,\varphi)$. Recall that

$$K(\cdot,\varphi) = (K(\cdot,\varphi_1), \dots, K(\cdot,\varphi_M))$$

for $i=1,\dots,m$ we have,

$$K(\underline{x},\varphi_i) = \varphi_i(K(\underline{x},\cdot))$$

$$= P(K(\underline{x},\cdot))(\underline{x}_i)$$

$$= -\operatorname{div}_{\underline{z}}\, a(\underline{z}) \nabla_{\underline{z}} K(\underline{x},\underline{z})\Big|_{\underline{z}=\underline{x}_i}$$

$$+ u(\underline{z})^T \nabla_{\underline{z}} K(\underline{x},\underline{z})\Big|_{\underline{z}=\underline{x}_i} + c(\underline{x}_i) K(\underline{x},\underline{x}_i)$$

for $i=m+1,\dots,M$ we simply have

$$K(\underline{x},\varphi_i) = K(\underline{x},\underline{x}_i)$$

The matrix $K(\varphi,\varphi)$ is in turn defined via the vector field $K(\cdot,\varphi)$ as

$$K(\varphi,\varphi)_{ij} = \varphi_i(K(\cdot,\varphi_j))$$

$$K(\varphi,\varphi) = \begin{array}{c|c} & \overbrace{\qquad}^{m} \quad \overbrace{\qquad}^{M-m} \\ \begin{array}{c} m\{ \\ M-m\{ \end{array} & \left[\begin{array}{c|c} K^{11} & K^{12} \\ \hline K^{21} & K^{22} \end{array}\right] \end{array}$$

$K^{12} = K^{21}$ due to symm.

- The K^{22} block is most familiar

$$K^{22} \ni K(\varphi,\varphi)_{ij} = \delta_{x_i}\left(K(\cdot, \delta_{\underline{x}_j})\right)$$

corresponds to bdry pts.

$$= K(\underline{x}_i, \underline{x}_j) \quad \text{for } i,j \in \{m+1, \dots, M\}.$$

- The K^{12} block is also easy to see since

$$K^{12} \ni K(\varphi,\varphi)_{ij} = \delta_{\underline{x}_i}\left(K(\cdot, \delta_{\underline{x}_j} \circ P)\right), \quad \begin{array}{l} i = m+1, \dots, M \\ j = 1, \dots, m \end{array}$$

$$= -\mathrm{div}_{\underline{z}}\, a(\underline{z}) \nabla_{\underline{z}} K(\underline{x}_i, \underline{z})\Big|_{\underline{z}=\underline{x}_j}$$

$$+ b(\underline{z})^T \nabla_{\underline{z}} K(\underline{x}_i, \underline{z})\Big|_{\underline{z}=\underline{x}_j} + c(\underline{x}_i) K(\underline{x}_i, \underline{x}_j),$$

- The K^{22} block is most involved

$$K^{22} \ni K(\varphi,\varphi)_{ij} = \delta_{\underline{x}_i} \circ P\left(K(\cdot, \delta_{\underline{x}_j} \circ P)\right), \quad \begin{array}{l} i = 1, \dots, m \\ j = 1, \dots, m \end{array}$$

$$= +\mathrm{div}_{\underline{w}}\, a(\underline{w}) \nabla_{\underline{w}} \left(\mathrm{div}_{\underline{z}}\, a(\underline{z}) \nabla_{\underline{z}} K(\underline{w}, \underline{z})\right)\Big|_{(\underline{w},\underline{z}) = (\underline{x}_i, \underline{x}_j)}$$

$$+ b(\underline{w})^T \nabla_{\underline{w}} \left(b(\underline{z})^T \nabla_{\underline{z}} K(\underline{w}, \underline{z})\right)\Big|_{(\underline{w},\underline{z}) = (\underline{x}_i, \underline{x}_j)}$$

$$+ c(\underline{x}_i)\, c(\underline{x}_j) K(\underline{x}_i, \underline{x}_j)$$

Observe, the expressions can get complicated quickly! but there's a lot of structure in them & they can be automated easily e.g. using autograd or symbolic calculations.

Once the $K(\varphi, \varphi)$ matrix is computed most of our computational effort at this point will be spent on computing $\underline{\alpha} = K(\varphi,\varphi)^{-1} \underline{Y}$ as is often the case with kernel methods.

Additionally the matrix $K(\varphi,\varphi)$ can be ill-conditioned & often needs to be regularized appropriately.

$$K(\varphi,\varphi) \leftarrow K(\varphi,\varphi) + \lambda R$$

where $R = \mathrm{Diag}(K(\varphi,\varphi))$ is the diagonal part of $K(\varphi,\varphi)$. The reason for choosing this nugget instead of λI is that the different blocks K^{11} & K^{22} can have very diff. scales!

consider $K(x,y) = \exp\left(-\frac{|x-y|^2}{2\gamma^2}\right) \approx O(1)$

while $\partial_x K(x,y) = -\frac{|x-y|}{2\gamma^2} K(x,y) \approx O\left(\frac{1}{\gamma^2}\right)$

$\vdots$

23.2 Choice of Kernels & the Matérn Class

Our formulation so far is indep. beyond the choice of K beyond asking for suff. regularity so that the entries of $K(\varphi,\varphi)$ are well defined. For second order PDEs this amounts to $K(\underline{x},\underline{y})$ to be twice cont. diff. in each argument.

- But for different PDEs, certain K would be more suitable than others. In fact, the possible choice would be to let K be the Green's function of the PDE! in this case the RKHS of k will automatically satisfy the bc's. But this choice is not practical because if we knew the GF then we wouldn't need to solve the equation to begin with.

- Certain choices, such as polynomial kernels would not be the best because we would not expect the solution to be a polynomial.

- RBF kernel would be fine since it is flexible but since its RKHS is very smooth it may lead to bad conditioning & overly smooth solutions.

Recall, standard theory of elliptic PDEs tells us that the solution map $\mathcal{P}^{-1}: H^s(\Omega) \to H_0^{s+2}(\Omega)$. That is, if we know the regularity of f then we have a good idea of the regularity of sol^n, in the Sobolev sense.

The family of kernels that is naturally adapted to such regularity classes is the Matérn kernel:

$$K(\underline{x},\underline{y}) = k(\|\underline{x}-\underline{y}\|), \quad k(t) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\,\frac{t}{\gamma}\right)K_\nu\left(\sqrt{2\nu}\,\frac{t}{\gamma}\right)$$

where Γ is the Gamma function & K_ν is the modified Bessel function of the second kind. ν is a smoothness index & γ is the lengthscale.

Special Cases:

- for $\nu = 1/2$ $\quad K(t) = \exp\left(-\frac{t}{\gamma}\right)$ (Laplace kernel)
- for $\nu = 3/2$ $\quad K(t) = \left(1 + \frac{\sqrt{3}t}{\gamma}\right)\exp\left(-\frac{\sqrt{3}t}{\gamma}\right)$
- for $\nu = 5/2$ $\quad K(t) = \left(1 + \frac{\sqrt{5}t}{\gamma} + \frac{5t^2}{3\gamma^2}\right)\exp\left(-\frac{\sqrt{5}t}{\gamma}\right)$
- as $\nu \to \infty$ $\quad K(t) \to \exp\left(-\frac{t^2}{2\gamma^2}\right)$ (RBF)

The reason that Matérn kernels are desirable for PDEs & function approx. is that their RKHS is equivalent to Sobolev spaces (see Kanagawa et al. "Gaussian Processes & kernel methods: A review on Connections & equivalences"

Thm: Let $K(\underline{x},\underline{y}) = K_{\nu}(\|\underline{x}-\underline{y}\|)$ be a Matern kernel with with index ν such that $s = \nu + d/2$ is an integer. Then the RKHS of K is norm equivalent with the Sobolev space $H^s(\mathbb{R}^d)$ ie,

$$C_1\|u\|_{H^s} \leq \|u\|_K \leq C_2\|u\|_{H^s}, \quad C_1, C_2 \geq 0.$$