Lecture 22: Solving PDEs with Kernels

We will consider a modern application of kernel methods for numerical solution of differential equations, in particular second order elliptic PDEs (see Ch 6 of Evans)

$$\begin{cases} \mathcal{P}\, u(\underline{x}) = f(\underline{x}), & \underline{x} \in \Omega \\ u(\underline{x}) = 0 & \underline{x} \in \partial\Omega \end{cases}$$

where $\Omega \subset \mathbb{R}^d$ is assumed to be a convex domain with smooth bdry & $\mathcal{P}$ is a differential operator of

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

where $a: \Omega \to \mathbb{R}$, $b: \Omega \to \mathbb{R}^d$, $c: \Omega \to \mathbb{R}$ are smooth functions & it holds that $a^- \leq a(\underline{x}) \leq a^+$ for some constants $a^{\pm} > 0$ so that $\mathcal{P}$ is uniformly elliptic.

These assumptions are nice because they ensure that the PDE is well-posed & has a unique sol^n. In fact, we can show that

$$\mathcal{P}^{-1}: H^s \longrightarrow H_0^{s+2}, \quad \text{for } s \geq -1$$

is continuous. ↩ Dirichlet bdry cond.

Basically, these assumptions ensure that the PDE is nice & amenable to numerical approximation!

Let us assume $f \in H^s(\Omega)$ with $s > d/2$ so that by Sobolev embedding we have that f is continuous & so is $u \in H^{s+2}(\Omega)$. Our goal in this lecture is to derive a simple, meshless collocation scheme for solving this PDE.

22.1 Collocation methods

Consider the above setting. Then collocation methods for solving the PDE are broadly defined as families of methods that parameterize the solution u as

$$u^n(\underline{x}) = \sum_{j=1}^{n} \alpha_j \psi_j(\underline{x})$$

for an appropriate set of functions ψ_j, eg. polynomials or trigonometric bases, & then solve the system of equations

$$\begin{cases} P(u^n)(\underline{x}_i) = \sum_{j=1}^{n} \alpha_j P\psi_j(\underline{x}_i) = f(\underline{x}_i), & \underline{x}_i \in \Omega \\ u^n(\underline{x}_i) = 0 & \underline{x}_i \in \partial\Omega \end{cases}$$

where $X = \{\underline{x}_1, \dots, \underline{x}_m\} \subset \overline{\Omega}$ is a set of collocation pts. Putting everything together we get a sys. of eqns for α_j's.

(assuming the ψ_j satisfy $\psi_j(\underline{x}) = 0$ on $\partial\Omega$)

$$A\underline{\alpha} = \underline{f}$$

where $\underline{f} = (f_1, \dots, f_m)$, $f_i = f(\underline{x}_i)$, &

$A_{ij} = \psi_j(\underline{x}_i)$. Clearly when $m=n$ & A is invertible then we can uniquely solve for $\underline{\alpha}$.

(see Trefethen "Spectral Methods in MATLAB")

22.2 Kernel Collocation Methods

Observe that assuming the form $u^n = \sum_{j=1}^n \alpha_j \psi_j$ in the formulation of the collocation method implies the existence of an underlying kernel

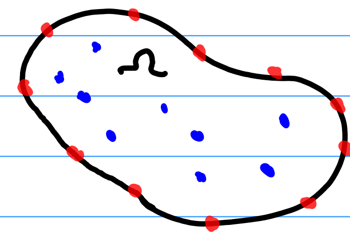
$$K(\underline{x}, \underline{y}) = \sum_{j=1}^n \psi_j(\underline{x})\psi_j(\underline{y})$$

such that u^n belongs to its RKHS. But this perspective is rarely employed in the lit. on collocation methods because we always aim to setup the $\underline{x}_i$ & the ψ_j so that A is invertible.

Let us consider a kernel $K: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ & a set of collocation pts $X = \{\underline{x}_1, \dots, \underline{x}_M\} \subset \overline{\Omega}$

such that $\{\underline{x}_1, \dots, \underline{x}_m\} \subset \Omega$ & $\{\underline{x}_{m+1}, \dots, \underline{x}_M\} \subset \partial\Omega$.

We then pose the problem



$$
(P1) \qquad u^n = \begin{cases} \underset{u \in \mathcal{H}_K}{\operatorname{argmin}} \ \|u\|_K \\ \text{subject to} \quad P(u)(\underline{x}_i) = f(\underline{x}_i), & i = 1, \dots, m, \\ \qquad\qquad\quad u(\underline{x}_i) = 0 & i = m+1, \dots, M. \end{cases}
$$

Put simply, we wish to find the minimum norm element of $\mathcal{H}_K$ that satisfies the PDE & bc's at the collocation pts. X.

Observe, we need to assume $\mathcal{H}_K$ is sufficiently regular so that $P(u)(\underline{x}_i)$ is well-defined, ie, $\Delta u \in C^0(\Omega)$ for all $u \in \mathcal{H}_K$, ie, $\mathcal{H}_K \subset C^2(\Omega)$.

We now observe that (P1) can be written in a familiar form, given $u \in C^2(\Omega)$ define

$$
\phi_i : C^2(\Omega) \longrightarrow \mathbb{R}, \qquad i = 1, \dots, m
$$

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

as well as, $\phi_i(u) = u(\underline{x}_i)$, $i = m+1, \dots, M$.

Observe that the $\phi_i \in \mathcal{H}_K^*$ under our assumptions. Thus, we can rewrite P1 as

$$u^n = \begin{cases} \operatorname{argmin} \|u\|_K \\ \text{s.t.} \quad \phi_i(u) = f_i, \quad i = 1, \dots, m \\ \qquad\quad \phi_i(u) = 0, \quad i = m+1, \dots, M. \end{cases}$$

Let $\underline{Y} = \begin{bmatrix} \underline{f} \\ \underline{0} \end{bmatrix} \in \mathbb{R}^M$ & $\varphi : \mathcal{H}_K \longrightarrow \mathbb{R}^M$ with $\varphi(u) = (\phi_1(u), \dots, \phi_M(u))$ & write

P2 $$u^n = \begin{cases} \operatorname{argmin} \|u\|_K \\ \text{s.t.} \quad \varphi(u) = \underline{Y} \end{cases}$$

which is precisely the generalized interpolation problem you encountered in HW3. Here you showed that assuming the ϕ_i are linearly independent then $\underline{u}^n$ is uniquely identified by the formula

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

where $K(\cdot, \varphi) : \mathbb{R}^d \to (\mathcal{H}_K)^{\otimes M}$ is the vector field

$$K(\cdot, \varphi) = (K(\cdot, \varphi)_1, \dots, K(\cdot, \varphi)_M)$$

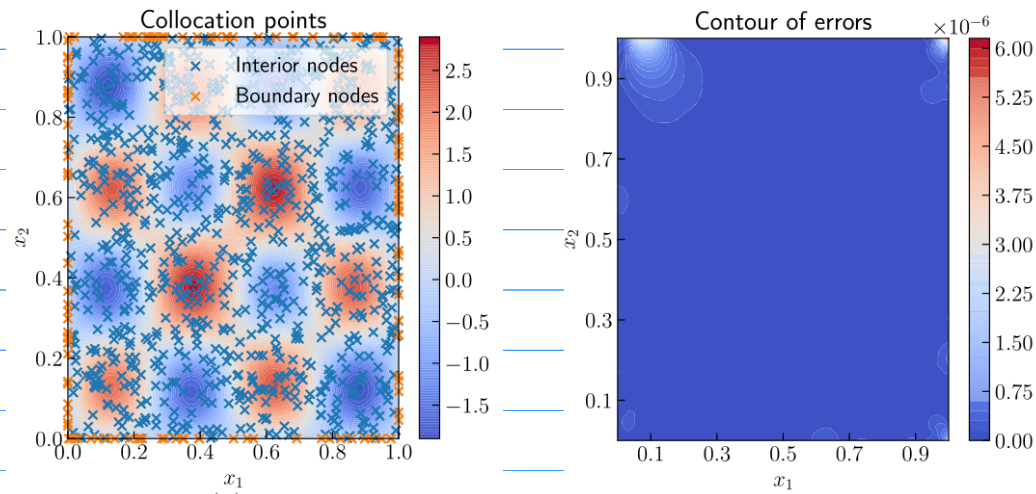
$$K(\underline{x}, \varphi)_i := \varphi_i(K(\underline{x}, \cdot)) \in \mathcal{H}_K$$

& the matrix $K(\varphi, \varphi) \in \mathbb{R}^{M \times M}$ has entries

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

Observe the simplicity of $(*)$ & the fact that it allows us to generalize collocation methods to kernels with inf. many features. Also, the method, at the level of implementation, is benign to location of the $\underline{x}_i$'s (ie, meshless) & dimension of Ω (size of the system depends only on M!).

All of the details of implementation lie in computing $K(\varphi,\varphi)^{-1}\underline{y}$ & in particular in forming the entries of this kernel matrix.



$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega \end{cases}$$

K - RBF kernel