Progress Report

Wei Yuan OH

October 2024

1 Parametric Boundary Value Problem

We consider the model parametric elliptic boundary value problem.

$$\begin{cases} -\nabla \cdot (a\nabla u) = f & \text{in } D, \\ u = 0 & \text{on } \partial D. \end{cases} \tag{1.1}$$

To derive the variational formulation of (1.1) without the parameter y, we multiply (1.1) with a test function $v(x)$ and take the integral over D to get

$$-\int_D \nabla \cdot (a(x)\nabla u(x)) \cdot v dx = \int_D f(x) v dx \tag{1.2}$$

Using the Divergence Theorem,

$$-\int_D \nabla \cdot (a(x)\nabla u(x)) \cdot v dx = \int_{\partial D} v a(x) \nabla u + \int_D a(x) \nabla u \nabla u dx \tag{1.3}$$

The boundary term vanishes because $v = 0$ on ∂D.
This is set in the space $V := H_0^1(D)$, which we endow with the $\bar{a}$-dependent scalar product:

$$(u, v)_V := \int_D \bar{a}(x) \nabla u \cdot \nabla v(x)\, dx = \int_D f(x) v(x) dx \tag{1.4}$$

and the norm $\| \cdot \|_V$. More generally, for any measurable subset $G \subset D$, we define the semi-definite form $(\cdot, \cdot)_{V,G}$ and the corresponding seminorm $| \cdot |_{V,G}$ by restricting the integral in (1.2) to G. Our goal is to find $u(x)$ such that (1.4) holds.

2 Uniform Coefficients

Through the generalised Polynomial Chaos approach, the coefficient a is permitted to depend on a sequence of scalar random parameters y_m, hence we have

$$a(y,x) = \bar{a}(x) + \sum_{m=1}^{\infty} y_m a_m(x), \quad x \in D, \tag{2.1}$$

with $\bar{a}$, $a_m \in W^{1,\infty}(D)$ and $|y_m| \leq 1$, which entails $y := (y_m)_{m=1}^{\infty} \in \Gamma := [-1,1]^{\infty}$. Therefore, the initial problem in (1.1) becomes

$$\begin{cases} -\nabla \cdot (a(y,x)\nabla u) = f & \text{in } D, \\ \qquad\qquad\qquad\quad u = 0 & \text{on } \partial D. \end{cases} \tag{2.2}$$

For all $m \in \mathbb{N}$, let π_m be a symmetric Borel probability measure on $[-1,1]$, i.e., π_m is invariant under the transformation $y_m \mapsto -y_m$. Although each y can be distributed differently, we assume for simplicity that for every m, y is uniformly distributed in $[-1,1]$. Then

$$\pi := \bigotimes_{m=1}^{\infty} \pi_m \tag{2.3}$$

is a probability measure on Γ with the corresponding product sigma algebra. The weak formulation of (1.1) is to find $u \in L^2_\pi(\Gamma; V)$ such that

$$\begin{aligned} \int_\Gamma (A(y)u(y), v(y))_V \, d\pi(y) &= \int_\Gamma \int_D a(x,y)\nabla u(x,y) \cdot \nabla v(x,y)\, dx\, d\pi(y) \\ &= \int_\Gamma \int_D f(x) v(y,x)\, dx\, d\pi(y), \quad \forall v \in L^2_\pi(\Gamma; V). \end{aligned} \tag{2.4}$$

2.1 Tensorized Legendre Polynomials

Assume that each random variable y_m is uniformly distributed on the interval $[-1,1]$. Let Γ denote the probability space of the random vector $y = (y_1, y_2, \ldots, y_m, \ldots)$. Consider the Hilbert space $L^2(\Gamma)$ of square-integrable functions with respect to the uniform probability measure. An orthonormal basis of this space is given by the tensorized Legendre polynomials.

Let $\mathcal{F}$ be the multi-index set, and let $\mu = (\mu_1, \mu_2, \ldots, \mu_m, \ldots)$ represent a multi-index with $\mu_i \in \mathbb{N} \cup \{0\}$.

The family of tensorized Legendre polynomials is defined as

$$P_\mu(y) = P_{\mu_1}(y_1)P_{\mu_2}(y_2)\cdots P_{\mu_m}(y_m)\cdots \tag{2.5}$$

where $P_n(\cdot)$ denotes the one-dimensional Legendre polynomial of degree n.

Given this basis, we can represent the solution $u(x, y)$ of the stochastic problem as a polynomial chaos expansion:

$$u(x, y) = \sum_{\mu\in\mathcal{F}} u_\mu P_\mu(y). \tag{2.6}$$

Each one-dimensional Legendre polynomial $P_n(y_m)$ satisfies the following recursion.For constants β_n, the recurrence can be expressed as:

$$\beta_n P_n(y_m) = y_m P_{n-1}(y_m) - \beta_{n-1}P_{n-2}(y_m) \tag{2.7}$$

Multiplying y_m to a given polynomial $P_\mu(y)$, we have

$$y_m P_\mu(y) = \beta_{\mu_m+1}P_{\mu+\epsilon_m}(y) + \beta_{\mu_m}P_{\mu-\epsilon_m}(y) \tag{2.8}$$

This relation expresses the multiplication by y_m in terms of neighboring terms in the multi-index space, which is crucial for building and manipulating polynomial chaos expansions. (can add proof if required)

2.2 Galerkin Method and Projections

We consider a set $\Lambda \subset \mathcal{F}$. For $\mu \in \Lambda$, we define

$$V_N := \{v(y, x) = \sum_{\mu} v_\mu P_\mu(y); v_\mu \in V_\mu\} \subset L^2_\pi(\Gamma; V) \tag{2.9}$$

We can express a test function v as such

$$v(x, y) = \sum_{\nu} v_\nu P_\nu(y) \tag{2.10}$$

Using (2.6) and (2.9), the weak formulation in (2.4) becomes

$$\int_{\Gamma}\int_{D} a(x,y)\nabla u(x,y)\cdot\nabla v(x,y)\,dx\,d\pi(y) = \sum_{\mu}\sum_{\nu}\int_{\Gamma}\int_{D} a(y,x)\cdot\nabla u_{\mu}\cdot\nabla v_{\nu}P_{\mu}(y)P_{\nu}(y)\,dx\,d\pi(y) \tag{2.11}$$

Using (2.1), we reformulate (2.11) as:

$$\sum_{\mu}\sum_{\nu}\int_{D}\left(\bar{a}(x)\left(\int_{\Gamma}P_{\mu}(y)P_{\nu}(y)\,d\pi(y)\right)+\sum_{m=1}^{\infty}a_m(x)\left(\int_{\Gamma}y_mP_{\mu}(y)P_{\nu}(y)\,d\pi(y)\right)\right)\nabla u_{\mu}\cdot\nabla v_{\nu}\,dx. \tag{2.12}$$

To simplify this, we introduce the Kronecker delta δ_{mn}, which equals 1 only when $\mu_m = \nu_n$ (i.e., the m-th index of μ matches the n-th index of ν), by the orthonormality of Legendre polynomials. Using this, (2.12) becomes:

$$\sum_{\mu}\sum_{\nu}\int_{D}\left(\bar{a}(x)\delta_{mn}+\sum_{m=1}^{\infty}a_m(x)\left(\int_{\Gamma}y_mP_{\mu}(y)P_{\nu}(y)\,d\pi(y)\right)\right)\nabla u_{\mu}\cdot\nabla v_{\nu}\,dx. \tag{2.13}$$

Here, δ_{mn} ensures that the integral of $P_{\mu}(y)P_{\nu}(y)$ is nonzero only when the indices of μ and ν match appropriately. Furthermore, we define:

$$\int_{\Gamma}y_mP_{\mu}(y)P_{\nu}(y)\,d\pi(y) = A_{mn}.$$

Next, we expand $u_{\mu}(x)$ and $v_{\nu}(x)$ as:

$$u_{\mu}(x) = \sum_{i}c_i\phi_i(x) \quad \text{and} \quad v_{\nu}(x) = \sum_{j}d_j\phi_j(x), \tag{2.14}$$

where $\phi_i(x)$ and $\phi_j(x)$ are basis functions, and c_i, d_j are the corresponding coefficients.
Substituting these expansions into the equation, we obtain:

$$\begin{aligned}\sum_{m}\sum_{n}\sum_{i}\sum_{j}\left(\int_{D}\left(\bar{a}(x)\delta_{mn}+\sum_{m=1}^{\infty}a_m(x)\cdot A_{mn}\right)\frac{\partial\phi_i}{\partial x}\cdot\frac{\partial\phi_j}{\partial x}\,dx\right)c_{in}d_{jm}\\ = \int_{D}f(x)\left(\sum_{i}d_{i0}\phi_i(x)\right)\,dx.\end{aligned} \tag{2.15}$$

We define the matrix $S_{(in)(jm)}$ as:

$$S_{(in)(jm)} = \int_D \left(\bar{a}(x)\delta_{mn} + a_m(x) \cdot A_{mn}\right) \frac{\partial \phi_i}{\partial x} \cdot \frac{\partial \phi_j}{\partial x}\, dx, \tag{2.16}$$

and the vector F_{in} as:

$$F_{in} = \int_D f(x)\phi_i(x)\, dx. \tag{2.17}$$

Therefore, the system of equations can be written in matrix form as:

$$Sc = F, \tag{2.18}$$

where$c$ is the vector of coefficients.

3 Error Estimation and Mesh Refinements

let $\mathcal{T}_k$ be partition for u_k

$$\epsilon_k^2(u_k, T) = h_T^2 ||r(x)||_{L^2(T)}^2 + h_T ||j(x)||_{L^2(\partial T \setminus \partial D)}^2$$

$$r(x) = f(x) + \frac{d}{dx}(a(x)\frac{du_k(x)}{dx})$$

in T

$$||r(x)||_{L^2(T)}^2 = \int_T r(x)dx$$

$$j(x) = n^+ \cdot a(x)\frac{du_k}{dx}|_T + n^- \cdot a(x)\frac{du_k(x)}{dx}|_T^-$$

Algorithm 1: Compute Elementwise Error List

```
Data: Mesh $\mathcal{M} = [x_0, x_1, \ldots, x_m]$, Nodal values $\mathbf{u}_h = [u_0, u_1, \ldots, u_m]$, Error tolerance $\epsilon$.
Result: List of elementwise errors ErrorList = $[\epsilon_1^2, \epsilon_2^2, \ldots, \epsilon_m^2]$.
Initialize: ErrorList ← [];
for $i \leftarrow 0$ to $m - 1$ do
    Compute Mesh Size:
    $h_T \leftarrow \mathcal{M}[i+1] - \mathcal{M}[i]$;
    Compute Residual $r(x)$ over Element $T_i$:
    $r_T \leftarrow$ element_residual_l2($\mathcal{M}[i], \mathcal{M}[i+1], \mathcal{M}, \mathbf{u}_h$);
    Compute Jump $j(x)$ at Node $x_{i+1}$:
    $j_T \leftarrow$ j($\mathcal{M}, \mathbf{u}_h, i+1$);
    Calculate Error Estimator for $T_i$:
    $\epsilon_T^2 \leftarrow h_T^2 \cdot r_T + h_T \cdot j_T^2$;
    Append to Error List:
    ErrorList.append($\epsilon_T^2$);
return ErrorList;
```

4 Numerical Experiments

4.1 One-Dimensional Deterministic Elliptic Problem

We examine the following one-dimensional deterministic elliptic problem, characterized by the coefficient function and its exact solution:

$$a(x) = 2 + \sin(x), \quad u(x) = x(1 - x). \tag{4.1}$$

In this formulation, $a(x)$ represents the spatially varying coefficient, and $u(x)$ is the exact solution to the elliptic partial differential equation under consideration. Our objective is to approximate the solution $u(x)$ under two distinct mesh configurations:

1. **Uniform Mesh:** A mesh with equally spaced elements across the domain.

2. **Adaptive Mesh:** A non-uniform mesh that selectively refines elements exhibiting significant errors.

The goal is to ensure that the weak formulation, as presented in (2.11), is satisfied under both mesh conditions.

Algorithm 2: Adaptive Mesh Refinement Loop

```
Data: Initial mesh $\mathcal{M}_0 = [0, 0.5, 1]$, Error tolerance $\epsilon$.
Result: Final mesh $\mathcal{M}$, solution coefficients $\mathbf{c}_{\text{sol}}$, energy norm enorm.
Initialize: $\mathcal{M} \leftarrow \mathcal{M}_0$;
while True do
    Solve System
    $(\mathbf{c}_{\text{sol}}, \mathbf{f}_{\text{sol}}) \leftarrow$ solve_scF_once$(n, m, \mathcal{M})$;
    Assemble Nodal Values
    nodal $\leftarrow$ assemble_nodal_values$(\mathbf{c}_{\text{sol}})$;
    Estimate Errors
    errors $\leftarrow$ sum_of_error_list$(\mathcal{M}, \text{nodal})$;
    Mark Elements for Refinement
    elements_to_refine $\leftarrow$ element_selection$(\mathcal{M}, \text{errors}, \epsilon)$;
    if elements_to_refine is empty then
        Terminate Loop;
    else
        Refine Mesh
        $\mathcal{M} \leftarrow$ element_refinement$(\mathcal{M}, \text{elements\_to\_refine})$;
        Print $\mathcal{M}$;
Compute Final Results
approximation $\leftarrow$ approx_new$(\mathbf{c}_{\text{sol}}, \mathbf{f}_{\text{sol}})$;
enorm $\leftarrow$ energy_norm(approximation);
return $\mathcal{M}, \mathbf{c}_{\text{sol}}$, enorm;
```

4.2 Final Mesh and Solution Coefficients

An adaptive mesh refinement procedure with an epsilon threshold of 0.1 was employed, yielding the final mesh points and their corresponding approximations. These approximated solutions were subsequently plotted alongside the exact solution for comparison.

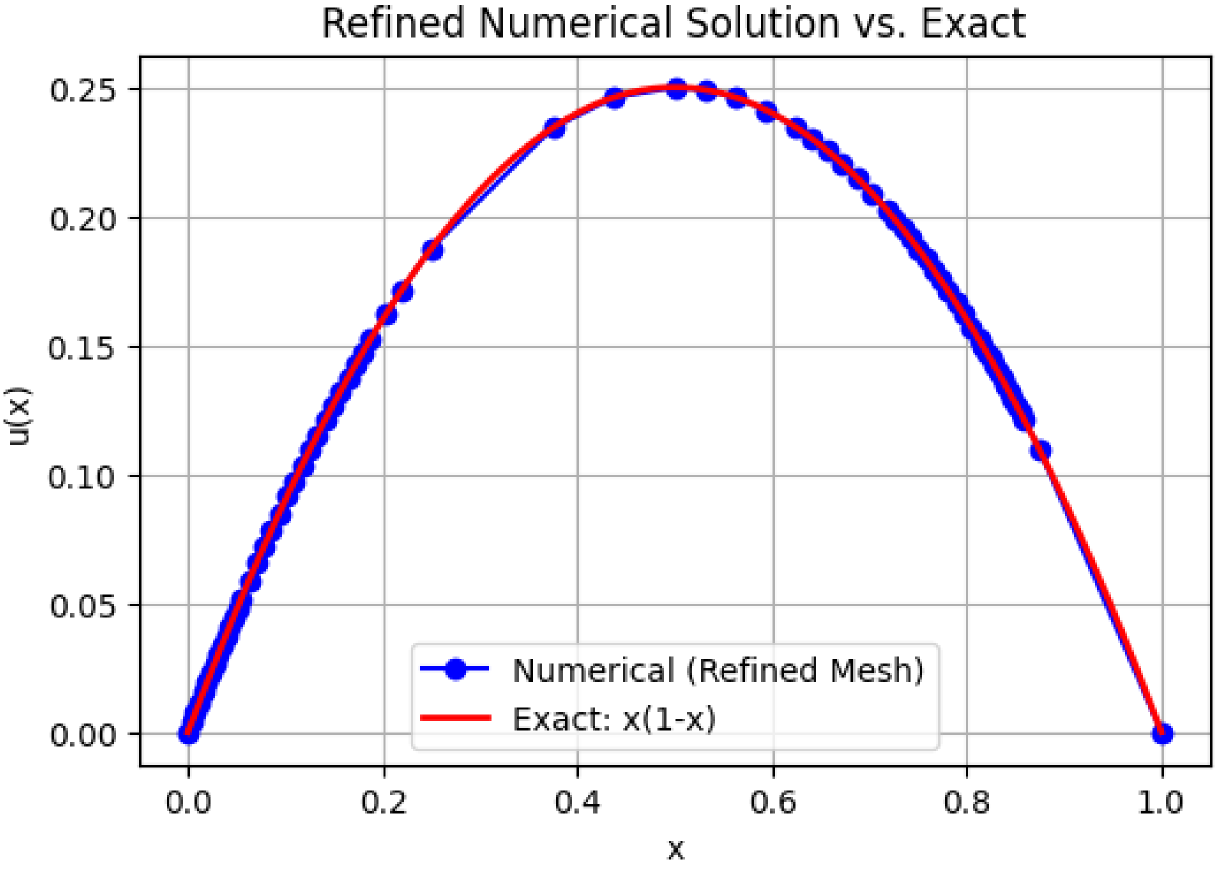

Figure 1: Figure 1: Comparison of Numerical Solution with Exact Solution

The energy norm of the final approximation is calculated to quantify the solution's accuracy:

$$\int_D f(x)u(x)dx - \int_D f(x)u^l(x)dx = 0.00388935312372196 \tag{4.2}$$

This low energy norm indicates a high level of accuracy in the numerical solution obtained through adaptive mesh refinement.