Approximating PDE solutions: L2 Optimality, Galerkin Method, Petrov-Galerkin, Collocation Methods, & Stochastic Galerkin

Conor Rowan

Spring 2024

1 L2 Optimality

Consider the following boundary value problem

$$\frac{\partial^2 u}{\partial x^2} + u = b(x), \quad u(0) = u(1) = 0$$

The distributed force $b(x)$ is an input, assumed to be known continuously, that drives the displacement u. Solving this equation can be treated as an approximation problem: we want to find $u(x)$ such that its second derivative plus itself is as close to $b(x)$ as possible. It would be possible to do this in a continuous way using the calculus of variations. Our objective is naturally defined as

$$\Pi\Big(u(x)\Big) = \frac{1}{2}\int_0^1 \left(\frac{\partial^2 u}{\partial x^2} + u - b\right)^2 dx$$

We want to find a minimum of the loss function in terms of the displacement $u(x)$. Using the calculus of variations, we can write the condition for a minimum as

$$\delta\Pi = 0 = \int_0^1 (u_{xx} + u - b)\delta u_{xx} + (u_{xx} + u - b)\delta u dx$$

To obtain a governing equation, we can integrate by parts two derivatives off the variation δu. There are no boundary terms because we have two Dirichlet boundaries. Noting that the variation is arbitrary, we obtain the following as a condition for a minimum:

$$\frac{\partial^2}{\partial x^2}(u_{xx} + u - b) + u_{xx} + u - b = 0$$

This is a useless route to take. Our method for solving a second order problem has turned it into a fourth order problem. The issues lies in trying to stay

continuous. We need to build the solution up from a finite set of parameters. Just like vectors can be approximated in terms of a given basis by choosing coefficients, we can specify a basis for approximating functions. The displacement can be approximated with

$$u(x) \approx \sum_i u_i f_i(x)$$

where $f_i(x)$ is a set of functions that are either given or chosen to construct the approximation. Note that they must respect our boundary conditions $u(0) = u(1) = 0$. Returning to the expression for the error, we can plug in this approximation to obtain

$$\Pi\Big(u_1, u_2, \dots\Big) = \frac{1}{2}\int_0^1 \left(\sum_i u_i\left(\frac{\partial^2 f_i}{\partial x^2} + f_i\right) - b(x)\right)^2 dx$$

We now only have a finite number of unknowns $u_1, u_2, \dots, u_N$ in the form of coefficients on the basis functions. We can minimize this loss function with the standard multivariate calculus technique of setting its gradient to zero. Doing so yields

$$\frac{\partial \Pi}{\partial u_j} = \int_0^1 \left(\sum_i u_i\left(\frac{\partial^2 f_i}{\partial x^2} + f_i\right) - b(x)\right)\left(\frac{\partial^2 f_j}{\partial x^2} + f_j\right)dx = 0$$

$$\implies \sum_i u_i \int_0^1 \left(\frac{\partial^2 f_i}{\partial x^2} + f_i\right)\left(\frac{\partial^2 f_j}{\partial x^2} + f_j\right)dx = \int_0^1 b(x)\left(\frac{\partial^2 f_j}{\partial x^2} + f_j\right)dx$$

Because the basis functions f_i are known, these integrals can actually be carried out. This leads to a linear system which can be used to solve for the unknown displacement coefficients:

$$K_{ji}u_i = F_j \implies \underline{u} = \underline{\underline{K}}^{-1}\underline{F}$$

For the given governing equation and boundary conditions, and with these definitions of the "stiffness matrix" and "force vector"

$$K_{ji} := \int_0^1 \left(\frac{\partial^2 f_i}{\partial x^2} + f_i\right)\left(\frac{\partial^2 f_j}{\partial x^2} + f_j\right)dx$$

$$F_j := \int_0^1 b(x)\left(\frac{\partial^2 f_j}{\partial x^2} + f_j\right)dx$$

the displacement coefficients are chosen to minimize the total squared error with the distributed force. For this reason, we call this method L2 optimal, because it minimizes the L2 loss. It possible to give a rather elegant physical interpretation of the governing equations which arise from this method. First, let us define the general boundary value problem

$$\mathcal{L}\Big(u(x)\Big) = b(x), \quad u(0) = u(1) = 0$$

where $\mathcal{L}$ is a generic linear differential operator, i.e. a stand-in for some unspecified differential equation for the displacement u. We approximate the displacement u with the same basis expansion, and because the operator is linear, it acts only on the basis functions whereas the displacement coefficients can be factored out. Thus, the error is

$$\Pi(u_1, u_2, \ldots, u_N) = \frac{1}{2}\int_0^1 \left(\sum_i u_i \mathcal{L}\Big(f_i(x)\Big) - b(x)\right)^2 dx$$

The condition for a minimum of the error in terms of the displacement approximation is

$$\frac{\partial \Pi}{\partial u_j} = \int_0^1 \left(\sum_i u_i \mathcal{L}\Big(f_i(x)\Big) - b(x)\right)\mathcal{L}\Big(f_j(x)\Big)dx = 0$$

The first term in the parentheses is the error of the approximation. This measures to what extent the approximation matches the body force. Call this error $e(\underline{u})$. If we think of the integral as a generalized dot product $\int_0^1 fgdx = \langle f, g\rangle$, this equation states that

$$\Big\langle e(\underline{u}), \mathcal{L}(f_j)\Big\rangle = 0$$

This indicates that the best choice of displacement coefficients is when the error is orthogonal to the each basis functions passed through the linear operator. So there is some orthogonality principle at work here, though this is still not totally clear. It can be clarified by considering the analogous problem in the context of linear algebra. Let's say we have a point $\underline{p} = (p_1, p_2)$ that we want to approximate in terms of a given basis vector $\underline{v} = [v_1, v_2]^T$. We have a single degree of freedom s that scales the basis vector. The condition for an optimal approximation is

$$\frac{\partial}{\partial s}\left(\frac{1}{2}||s\underline{v} - \underline{p}||^2\right) = 0$$

$$\implies (s\underline{v} - \underline{p}) \cdot \underline{v} = \underline{e}(s) \cdot \underline{v} = 0$$

where $\underline{e}$ is once again the error of the approximation. This equation states that the coefficient s is determined by finding the point in the approximation space defined by the basis $\underline{v}$ such that the error is orthogonal to the approximation space. This means that there is nowhere to move within the approximation space that reduces the error. This is shown in this plot. Note that if one were asked to move the slider for s to find the best approximation, one is likely to discover this orthogonality principle intuitively. This is not quite equivalent to case of approximating a solution to the different equation, because there an operator

showed up that does not appear here. We can easily remedy this discrepancy. The discrete version of any continuous operator is a matrix. An operator takes in a function and spits out another function, whereas a matrix takes in a vector and spits out another vector. The problem in linear algebra which is directly analogous to the differential equation is minimizing the following function

$$\Pi(s) = \frac{1}{2}||s\underline{\underline{A}}\underline{v} - \underline{p}||^2$$

We want to approximate the point with a given basis vector but only after that basis vector is acted on by a matrix. We can think of the action of this matrix as rotating and stretching the basis vector $\underline{v}$, but because the approximation space spans all scalar multiples of $\underline{\underline{A}}\underline{v}$, only the effect of rotation is important. What is the optimality principle here? It is straightforward to compute the minimal error approximation as

$$\frac{\partial \Pi}{\partial s} = (sA_{ij}v_j - p_i)A_{ik}v_k = \underline{e}(s) \cdot \underline{\underline{A}}\underline{v} = 0$$

The condition for an optimal approximation is now that the error is orthogonal to the rotated basis vector. If we look at the expression for the total error, we see that this is because the approximation is effectively happening in the space of $\underline{\underline{A}}\underline{v}$. In other words, if we define a new basis $\underline{\tilde{v}} = \underline{\underline{A}}\underline{v}$, we can apply the first optimality criteria (no matrix operator) to get

$$\underline{e}(s) \cdot \underline{\tilde{v}} = 0$$

In other words, we want the error not be perpendicular to the basis per se, rather the space in which $\underline{p}$ can be approximated. This is the same with the differential equation. Because we have

$$\mathcal{L}(u) = b$$

we don't actually approximate b in the space of our basis expansion of the displacement $u \approx \sum_i u_i f_i$, rather the "rotated" basis $\mathcal{L}(f_i(x))$. We can restate the condition for a minimum error approximation of the solution to the differential equation:

$$\Big\langle e(\underline{u}), \mathcal{L}(f_j) \Big\rangle = 0$$

Hopefully it is now clear what is going on. The displacement coefficients are chosen such that the approximation is error is perpendicular to each component of the "effective" basis $\mathcal{L}(f_j)$. Note that when one actually carries out this process, the problem reduces entirely to linear algebra (by integrating the set of basis functions). Thus, it is not a stretch to make use of linear algebra to build intuition for what is going on here.

2 Galerkin Method

The L2 optimal technique derived above is not a common technique for solving differential equations. The reason is that the formation of the linear system is rather complicated. A more common technique is the "Galerkin projection," which treats an optimal approximation as a set of coefficients for which the error is perpendicular to the basis functions/vectors, not the basis functions/vectors passed through the operator/matrix. See this plot for a demonstration. Find the point for which the error is perpendicular to the effective approximation space $\underline{\underline{A}}\underline{v}$. Then find the point for which the error is perpendicular to the basis vector. Note that these are not the same, and the discrepancy grows larger the more the operator $\underline{\underline{A}}$ rotates the basis. The governing equation for a Galerkin method is

$$\left\langle e(\underline{u}), f_j \right\rangle = 0$$

For the generic linear differential equation, this is equivalent to

$$\int_0^1 \left(\sum_i u_i \mathcal{L}(f_i) - b(x) \right) f_j(x) dx = 0$$

and for our particular differential equation, it is

$$\sum_i u_i \int_0^1 \left(f_i f_j - \frac{\partial f_i}{\partial x} \frac{\partial f_j}{\partial x} \right) dx = \int_0^1 b f_j dx$$

where we have used integration by parts to transfer one derivative off f_i and onto f_j. There are no boundary terms due to the fact that the basis functions are zero on the boundary. This is a linear system with

$$K_{ij} := \int_0^1 \left(f_i f_j - \frac{\partial f_i}{\partial x} \frac{\partial f_j}{\partial x} \right) dx$$

$$F_j := \int_0^1 b f_j dx$$

The Galerkin projection is the most common method for solving differential equations because it leads to much simpler linear systems with lower orders of differentiation. *Note that it is not clear why this method works if it not derived from an explicit minimization principle, as was the case for the L2 optimal method.* But evidently it does work. Disparities between the L2 optimal method and the Galerkin projection will be briefly explored later. Comparing the L2 and Galerkin stiffness matrices for this particular equation does make clear that this method is quite a bit simpler.

2.1 Petrov-Galerkin Method

It gets even weirder. Another approach is to treat the condition for an optimal approximation as

$$\Big\langle e(\underline{u}), g_j \Big\rangle = 0$$

where the $g_j(x)$ are a set of functions different from those used to approximate the displacement! Referencing linear algebra again, this is like saying that the error is normal to a distinct set of vectors than that of the basis. Note that one particular Petrov-Galerkin method would be using the functions

$$g_j = \mathcal{L}(f_j)$$

this recovering the L2 method, but this is not required. This condition for a Petrov-Galerkin optimal approximation can be written as

$$\int_0^1 \left(\sum_i u_i \mathcal{L}(f_i) - b(x) \right) g_j(x) dx = 0$$

and in the case of our particular equation, it is

$$\sum_i u_i \int_0^1 \left(f_i g_j - \frac{\partial f_i}{\partial x} \frac{\partial g_j}{\partial x} \right) dx = \int_0^1 b g_j dx$$

This leads to a linear system with the following stiffness matrix and force vector:

$$K_{ij} := \int_0^1 \left(f_i g_j - \frac{\partial f_i}{\partial x} \frac{\partial g_j}{\partial x} \right) dx$$

$$F_j := \int_0^1 b g_j dx$$

Note that the stiffness matrix is no longer symmetric, which can make computing its inverse more costly for large systems.

3 Relation to weak form

The "strong form" of a generic boundary value problem is

$$\mathcal{L}(u) = b(x)$$

A "weak form" of this equation is obtained by integrating against an arbitrary "test" function w:

$$\int_0^1 \mathcal{L}(u) w dx = \int_0^1 b w dx \implies \int_0^1 \Big(\mathcal{L}(u(x)) - b(x) \Big) w(x) dx = 0$$

This is yet another error orthogonality principle. If we discretize the solution u in the usual way, the weak form of the governing equation be can be expressed as

$$\Big\langle e(\underline{u}), w(x) \Big\rangle = 0$$

where $u_1, \ldots, u_N$ are the displacement coefficients. This states that the error is orthogonal to the test function. In theory, the test function is totally arbitrary but in practice we need to look at a finite set. All of the methods discussed above can be recovered by making particular choices of the set of test functions. For the L2 optimal method, we choose test functions which are the basis for $u(x)$ acted on by the differential operator. For a Galerkin method, the test functions are simply the basis for the displacement approximation. For a Petrov-Galerkin method, they are some other set of functions with potentially no relation to the displacement approximation. We can either think of the weak form as a different name for minimizing the approximation error, or error minimization as a nice property of the weak form.

4 Relation to collocation methods

A collocation method discretizes the displacement and minimizes the total error of the strong form evaluated at given "collocation" points. The loss for a collocation method is

$$\Pi(\underline{u}) = \frac{1}{2} \sum_j \Big(\sum_i u_i \mathcal{L}(f_i)(x_j) - b(x_j) \Big)^2$$

The index j refers to collocation points. The idea is that if this error is minimized, the governing equation is satisfied at the given points. We can minimize the error in terms of the displacement coefficients with

$$\frac{\partial \Pi}{\partial u_k} = \sum_j \Big(\sum_i u_i \mathcal{L}(f_i)(x_j) - b(x_j) \Big) \mathcal{L}(f_k)(x_j) = 0$$

$$\implies \sum_i u_i \left(\sum_j \mathcal{L}(f_i)(x_j) \mathcal{L}(f_k)(x_j) \right) = \sum_j b(x_j) \mathcal{L}(f_k)(x_j)$$

This looks exactly like the L2 optimal method except that integrals are approximated by sums. If we compute the integrals in the L2 optimal method numerically on a uniform grid, the collocation method will be equivalent if collocation points are taken as the integration points. The collocation method is essentially a less carefully integrated L2 optimal method. It is less careful because there is nothing that says collocation points should be evenly spaced, but if they are not evenly spaced their relative contributions to the integral this method approximates will not be accounted for. It is interesting to see that all these methods are closely connected. In fact, I have explored elsewhere that the

finite difference method can be viewed as a special case of a Petrov-Galerkin method!

5 Numerical comparisons

We numerically solve the 1D boundary value problem we have been discussing. The displacement is approximated with

$$u(x) = \sum_{i=1}^{N} u_i \sin(i\pi x)$$

and for the Petrov-Galerkin method we use the bizarre and made-up set of shape functions

$$g_j(x) = \sin(\pi x) \exp\left(-10\left(x - \frac{j-1}{N-1}\right)^2\right)$$

See Figures 1-7 for comparison of the different methods for a few different distributed forces and basis sizes. The stiffness matrices and force vectors are all formed using MATLAB's built-in integration tool. It is quite surprising how well all these seemingly disparate methods agree with each other. In particular, of all the distributed forces that I looked at, a typical difference between the displacement coefficients computed with the L2 and Galerkin methods is less than 1E-15!

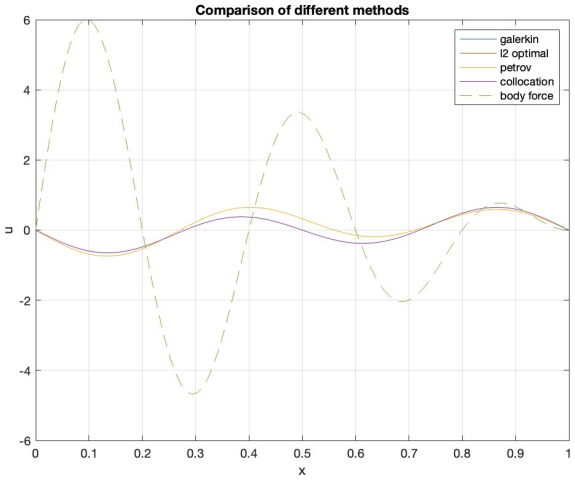

Figure 1: When only a few basis functions are used in the approximation ($N = 4$), the Petrov-Galerkin method disagrees with the other methods. The collocation method uses 100 points and a uniformly spaced grid.

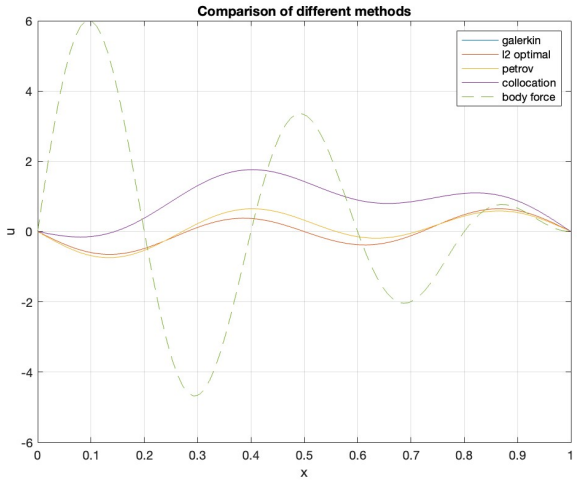

Figure 2: For $N = 4$ we try using Monte Carlo integration with a few hundred points to form the stiffness and force vectors for the collocation method. Collocation points are sampled randomly from the interval $[0, 1]$. This leads to integration error which causes the solution to be inaccurate.

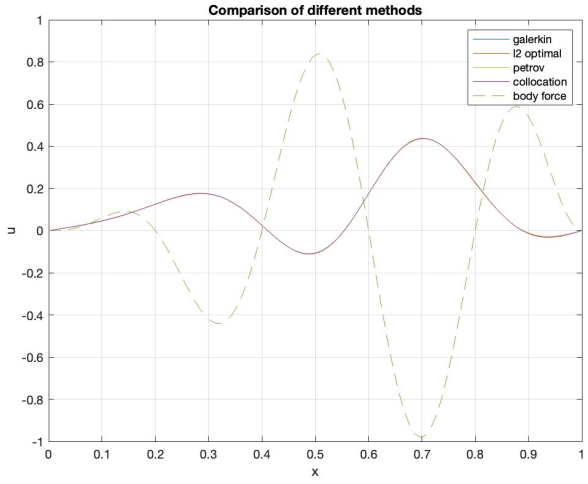

Figure 3: With $N = 8$ and uniformly spaced collocation points, all methods are in good agreement.

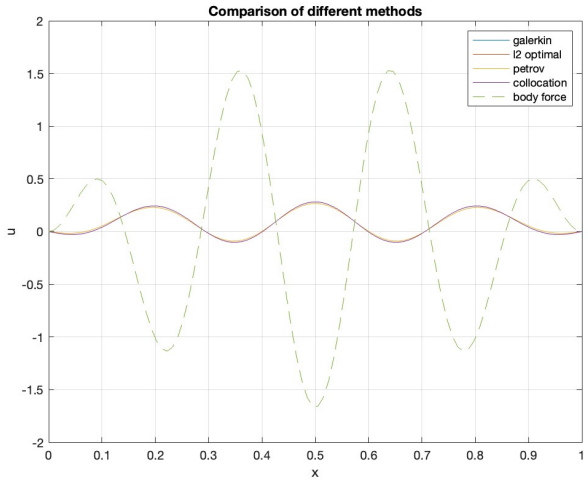

Figure 4: For $N = 8$ and uniformly spaced collocation points, all methods are in good agreement. We do see some slight deviations from the other methods on the part of the Petrov-Galerkin method.

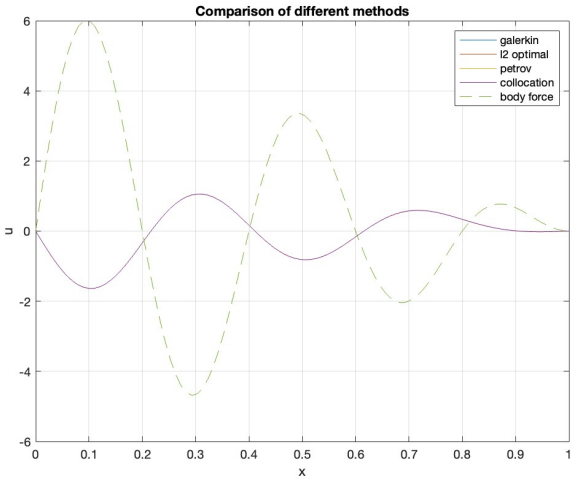

Figure 5: All methods agree for $N = 8$ and uniformly spaced collocation points.

6 Stochastic Galerkin and Principle of Minimum Expected Potential Energy

The minimum expectation of the potential energy associated with a partial differential equation (PDE) corresponds to a solution in the stochastic and physical

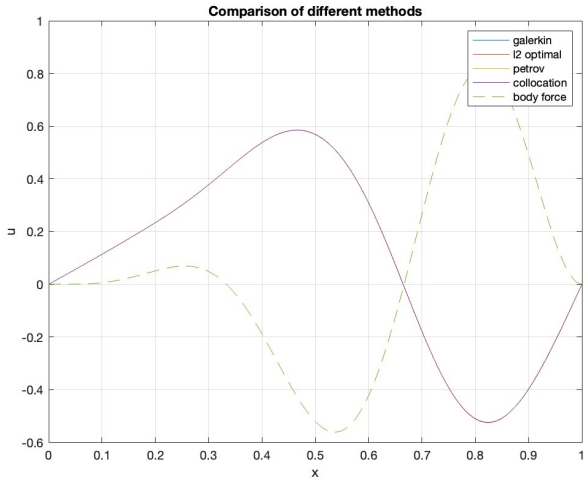

Figure 6: All methods agree for $N = 8$ and uniformly spaced collocation points, even with only 20 collocation points.

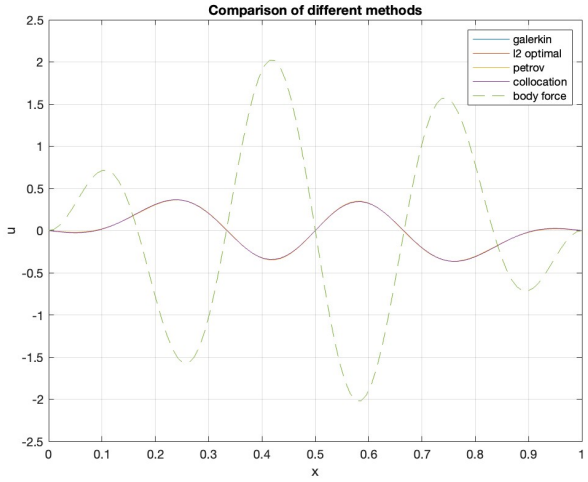

Figure 7: The collocation method only works with Monte Carlo integration when the number of points is very large. Here we use 1000 integration points sampled randomly from a uniform distribution.

space. This is a technique for performing uncertainty quantification on PDE's with variational principles. The more common Polynomial Chaos Expansion (PCE) relies on the stochastic Galerkin method, which computes a Galerkin op-

timal approximation first in the stochastic space, and then in the physical space. This is usually accomplished with a multiplicative (tensor product) decomposition of the solution's dependence on the spatial coordinate(s) and parameter(s). In our case, we will assume that that the dependence of the solution on the physical coordinates and parameters is captured with a single set of basis functions. With this setup, we can show the equivalence of the stochastic Galerkin method with the principle of minimum expected potential energy for elliptic PDE's. The stochastic Galerkin method is a simple extension of the approximation techniques we have outlined above to a continuous variable representing a parameter appearing in the partial differential equation. For example, if the modulus of a material depends on a parameter y, then the solution u has continuous dependence on this parameter just as it does on the spatial coordinates. Assuming that the parameter y is a random variable, the stochastic Galerkin method treats the approximation in the "parameter space" just as it does the approximation in physical space. Consider a simple 1D boundary value problem (BVP)

$$\kappa(x,y)\frac{\partial^2 u}{\partial x^2} + f(x) = 0, \quad u(0) = u(1) = 0$$

where u is a displacement-like quantity, $x \in [0,1]$ is the spatial coordinate, and $y \in [-\infty, \infty]$ is a random parameter that determines the spatially varying material coefficient κ. Let's say that the statistics of the random parameter are described by a known density function $\rho(y)$. For simplicity, we are assuming that there is only one random parameter and one spatial dimension to ease the presentation, though the following considerations are general. Because the coefficient in the BVP depends on the parameter y, the solution u will as well. To perform uncertainty quantification, we need to understand how the solution $u(x)$ changes with the parameter y. Once we have a relationship $(x,y) \to u(x,y)$, we can compute statistical quantities of interest such as the mean and variance of the solution. We can discretize the solution in the "parameter space" of y in the same way that we discretize it in physical space. This could be done in the following way:

$$u(x,y) = \sum_i u_i(x)\Psi_i(y)$$

This is a tensor-product type decomposition analogous to how we discretize space-time PDE's, where the coefficients control the time dependence of spatial basis functions. This is how a typical Polynomial Chaos Expansion proceeds. When using neural networks to discretize a PDE, it is convenient to not distinguish between the parameters and spatial coordinates: they both influence the solution as inputs to the first layer of the network. More closely aligned with the neural network's "indifference" to parameters, we discretize the solution in physical and parameter space with a single set of basis functions:

$$u(x,y) = \sum_i u_i\Psi_i(x,y)$$

The solution $u(x, y)$ is simply some surface, and we can build it up in any way we wish. We take this approach because it is more closely aligned with how a neural network handles the spatial coordinates and parameters. The Galerkin method must account for the fact that y is a random variable. Instead of simply integrating against each element of the basis for the displacement approximation, we use the density of y to weight contributions to the weak form by their corresponding density. This is natural if we think of the density as "prioritizing" points in parameter space which are more likely to be observed. In other words, we typically have no notion of importance for points in the spatial domain, but in the parameter space, there are some regions with zero or almost zero probability. There is no sense in weighting these equivalently in forming the residual for the weak form. More concretely, it can be seen that integrating against the density is equivalent with Galerkin optimality for the expected error of the approximation. It is hopefully clear that in the presence of randomness, we would like to minimize the expected error as opposed to an unweighted error, as the expectation operation will tend to prioritize reducing error in regions which are frequently observed. The stochastic Galerkin projection for the 1D boundary value problem yields

$$\int_0^1 \int_{-\infty}^{\infty} \left(\kappa(y) \frac{\partial^2 u}{\partial x^2} + b \right) \Psi_j \rho(y) dy dx = 0$$

Plugging in the displacement approximation, integrating by parts, and noting that $\Psi_j(0, y) = \Psi_j(1, y) = 0$ for each j, we obtain the discrete standard weak form for the stochastic PDE:

$$\sum_i u_i \left(\int_0^1 \int_{-\infty}^{\infty} \kappa(y) \frac{\partial \Psi_i}{\partial x} \frac{\partial \Psi_j}{\partial x} \rho(y) dy dx \right) - \int_0^1 \int_{-\infty}^{\infty} b \Psi_j \rho(y) dy dx = 0$$

$$\implies \sum_i u_i \langle K_{ij} \rangle - \langle F_j \rangle = 0$$

where $\langle \cdot \rangle$ indicates the expected value taken with respect to the random parameter y. The notation K_{ij} and F_j is used for the usual definition of the stiffness matrix and force vector. We know that equations like this come from the gradients of quadratic energies of the form

$$\Pi = \frac{1}{2} \langle K_{ij} \rangle u_i u_j - \langle F_i \rangle u_i$$

In this case we have expected value operations on the stiffness matrix and force vector. It can be seen by "reversing the discretization" that the continuous form of the energy functional is

$$\Pi = \int_0^1 \int_{-\infty}^{\infty} \left(\frac{1}{2} \kappa(y) \left(\frac{\partial u}{\partial x} \right)^2 \rho(y) - bu\rho(y) \right) dy dx$$

Thus, the stochastic Galerkin method applied to the strong form of the governing equations is equivalent to minimizing the expectation of the potential energy. This is analogous to the equivalence in the deterministic case between the condition of minimal energy and the weak form of the governing equations. Thus, we see that the minimum expectation of the energy is simply a re-casting of the stochastic Galerkin method which, in our case, is especially convenient for conducting uncertainty quantification when the PDE is discretized with a neural network.