Some Numerical Methods

Conor Rowan

December 14, 2023

1 Collocation Methods

Collocations methods are a simple way to approximately solve linear and non-linear partial differential equations. The idea is to discretize the solution, and find the coefficients of the discretization such that the partial differential equation is satisfied at a finite number of "collocation" points. These methods are very simple to implement, though there are some minor differences between the linear and non-linear cases.

1.1 Linear Problems

Consider a linear scalar PDE on the quantity u in two spatial dimensions:

$$\mathcal{L}\Big(u(x,y)\Big) = f(x,y)$$

The first thing that we do is discretize the solution in terms of known spatial shape functions Ψ_i:

$$u(x,y) = \sum_{i=1}^{N} u_i \Psi_i(x,y)$$

With this discretization, the strong form of the PDE is

$$\sum_{i=1}^{N} u_i \mathcal{L}\Big(\Psi_i(x,y)\Big) = f(x,y)$$

Next, we choose a set of C collocation points at which we want the PDE to be satisfied. We can denote this set of points as $[x_j, y_j]_{j=1}^{C}$. We want to determine the coefficients u_i such that the PDE is satisfied at all of the collocation points. This condition reads

$$\sum_{i=1}^{N} u_i \mathcal{L}\Big(\Psi_i(x_j,y_j)\Big) = f(x_j,y_j) \quad \text{for } j = 1,\dots,C$$

This can be written in matrix form as

$$\begin{bmatrix} \mathcal{L}\Big(\Psi_1(x_1,y_1)\Big) & \dots & \mathcal{L}\Big(\Psi_N(x_1,y_1)\Big) \\ \mathcal{L}\Big(\Psi_1(x_2,y_2)\Big) & \dots & \mathcal{L}\Big(\Psi_N(x_2,y_2)\Big) \\ \vdots & & \vdots \\ \mathcal{L}\Big(\Psi_1(x_C,y_C)\Big) & \dots & \mathcal{L}\Big(\Psi_N(x_C,y_C)\Big) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f(x_1,y_1) \\ f(x_2,y_2) \\ \vdots \\ f(x_C,y_C) \end{bmatrix}$$

If $N = C$, this is a square matrix and the system can be solved exactly. In most situations, the number of collocations points will be greater than the size of approximation of u, so that the system will be overdetermined ($C > N$). In this, a pseudo-inverse of the matrix multiplying the unknown coefficients can be used. For example, the Moore-Penrose pseudoinverse minimizes the squared error between the left and right hand sides of the overdetermined equation. To give an example of this method, we can solve the Euler-Tricomi equation given by

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

Apparently, this equation shows up in the study of transonic fluid flow. Here, we choose it as an example simply because it is a bit unusual! For simplicity, we will solve this problem on a square domain of unit side length with zero Dirichlet boundary conditions. Thus, a 2D fourier sine series is a good choice of basis. The solution is approximated as

$$u(x,y) = \sum_i \sum_j \hat{u}_{ij} \sin(i\pi x)\sin(j\pi y)$$

It is straightforward to collapse this double sum into a single sum to match the presentation above. Thus, the discretization is

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

where Ψ_i is some product of sines in the x and y directions. We use the forcing $f(x,y) = x\sin(2\pi x)$ and randomly sample collocation points within the domain. See the figures for results. When the number of collocation points is small, the solution is extremely inaccurate, but increasing the sampled points leads to convergence in the solution. The size of the approximation is $N = 25$, meaning that there are sines in each coordinate direction up to $\sin(5\pi x)$.

1.2 Non-linear Problems

Consider a non-linear PDE of the form

$$\mathcal{N}\Big(u(x,y)\Big) = f(x,y)$$

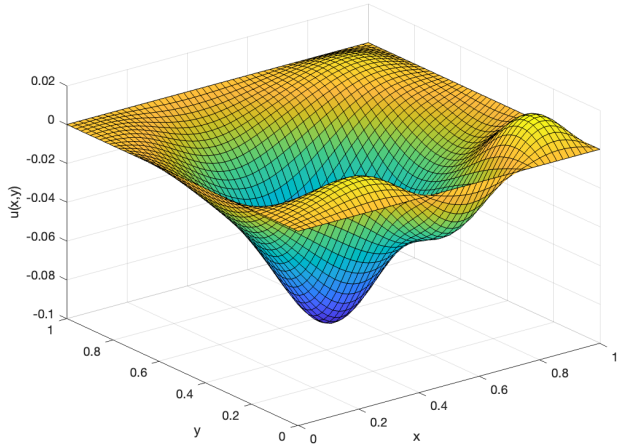

Figure 1: Solution of Euler-Tricomi equation with 25 randomly sampled collocation points.

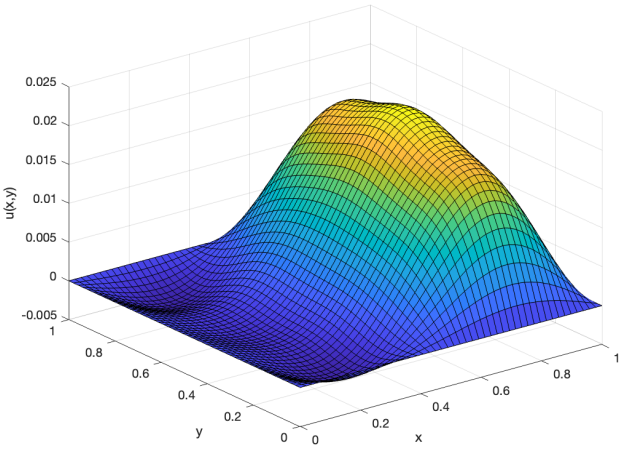

Figure 2: Solution of Euler-Tricomi equation with 100 randomly sampled collocation points.

We use the same discretization of the solution, namely

$$u(x,y) = \sum_{i=1}^{N} u_i \Psi_i(x,y)$$

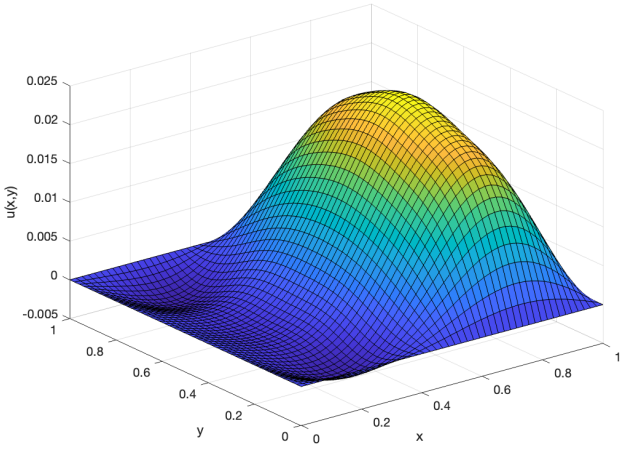

Figure 3: Solution of Euler-Tricomi equation with 400 randomly sampled collocation points.

but when we plug this into the governing equation, non-linearity prevents us from factoring out the unknown degrees of freedom u_i. Thus, the governing equation is

$$\mathcal{N}\left(\sum_{i=1}^{N} u_i \Psi_i(x,y)\right) = f(x,y)$$

In order to use a collocation method, we have to carry out an explicit optimization problem. We still enforce the governing equation at C collocation points, but instead of obtaining a matrix problem, we must minimize the loss defined by

$$\ell(\underline{u}) = \sum_{j=1}^{C} \left(\mathcal{N}\left(\sum_{i=1}^{N} u_i \Psi_i(x_j,y_j)\right) - f(x_j,y_j)\right)^2$$

Finding the vector of coefficients that minimizes this loss is equivalent to approximately satisfying the PDE at all of the collocation points. Constructing a minimization problem of this sort is probably easier than solving non-linear boundary value problems with the weak form of the governing equations, though one cannot be certain that these collocation methods lead to accurate solutions.

2 Fourier Transforms

We will explore using Fourier transforms to solve PDE's by looking at the 1D wave equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

The spatial Fourier transform of a function f is

$$\hat{f}(k,t) = \int_{-\infty}^{\infty} f(x,t) e^{-ikx} dx$$

Denoting the Fourier transform of a function as $\mathcal{F}$, it can be shown that Fourier transforms have a very useful property:

$$\mathcal{F}\left(\frac{\partial}{\partial x} f(x,t)\right) = ik\mathcal{F}(f(x,t)) = ik\hat{f}(k,t)$$

With this property, spatial derivatives disappear entirely in the frequency domain, transforming partial differential equations into ordinary differential equations. Because the Fourier transform does not see the time variable, time derivatives simply factor out of the transform. Thus, the wave equation in the frequency domain is

$$\frac{\partial^2 \hat{u}}{\partial t^2} = -c^2 k^2 \hat{u}(k,t)$$

This is now an ordinary differential equation in time for each frequency k. The solutions at different values of k are independent from one another. The general solution to this ODE is known analytically, and can be written as

$$\hat{u}(k,t) = \hat{F}(k) e^{-ickt} + \hat{G}(k) e^{ikct}$$

The constant amplitudes of the sinusoidal response can depend on k just as constants of integration depend on coefficients in an ordinary differential equation. The point is that this equation is simple to solve at a given frequency because the system response at a given frequency is decoupled from the responses at other frequencies. With an analytical representation of the solution in the frequency domain, we can simply take the inverse Fourier transform to get back into the spatial domain:

$$u(x,t) = \mathcal{F}^{-1}(\hat{u}(k,t)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left(\hat{F}(k) e^{-ickt} + \hat{G}(k) e^{ikct}\right) e^{ikx} dk$$

Sometimes, this integral can be computed analytically. But even if it must be carried out numerically, we have transformed the problem of solving a partial differential equation into computing a single numerical integral. This method may not be convenient for certain boundary conditions. As a final thought, it is interesting to consider how this analysis changes when the PDE is non-linear. Consider the following equation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u^2$$

Take its Fourier transform to obtain

$$\frac{\partial}{\partial t}\hat{u} = -k^2\hat{u} + \int u^2 e^{-ikx} dx$$

We can use the convolution property, which states that the Fourier transform of a product is a convolution of the Fourier transforms:

$$\frac{\partial}{\partial t}\hat{u} = -k^2\hat{u} + \int \hat{u}(k-k')n(k')dk'$$

Thus, we lose the fact that the ordinary differential equation for each frequency k is local. The solution at a fixed k depends on the whole spectrum of frequencies, thus no longer making this problem amenable to an analytical solution. This shows that non-linearities couple the frequency response of a system, and cause dynamics at different frequency levels to "mix" in some way.

3 Boundary Element Method

As we will sketch, boundary element methods require knowing the Green's function of the differential operator for a given problem. Thus, they are typically limited to linear problems for which Green's functions can be determined. Because only the boundary of the domain is meshed, the corresponding matrices in the discrete formulation are much smaller than traditional FEM. However, these matrices are dense and thus sparsity cannot be used to store them. The computational size of finite element problems tends to grow approximately linearly with the number of elements (size of discretization) due to the band structure of the stiffness matrix, but for boundary element problems which give rise to dense matrices, the growth is quadratic. Thus, boundary element methods are an interesting application of the various tools of mathematical mechanics (weak form solutions, green's functions, linear systems of discretized continuous problem, etc) but are not useful for large and/or complex problems. It could also be useful where the domain is very large compared to the boundary–it is a good choice on infinite or semi-infinite domains. Note that even linear problems which are anisotropic will not have readily available green's functions to used. Thus, BEM is good for efficient analysis of small and physically simple problems, perhaps where farfield boundary conditions need to be modeled exactly (no messy/inconvenient approximations).

3.1 Derivation

The so-called "fundamental solution" to Laplace's equation satisfies

$$\nabla^2\Phi = \Delta\Phi = \delta(x-\xi, y-\eta)$$

This is also called the Green's function. The function $\Phi(x, y, \xi, \eta)$ is the system's response to applied impulse at the point (ξ, η). For example, if the Laplace's equation is used to describe the displacement of a membrane over

some domain, the Green's function would provide the displacement field from a unit point load applied at (ξ, η). The fundamental solution to Laplace's equation is

$$\Phi(x, y, \xi, \eta) = \frac{1}{2\pi} \ln\Big(\sqrt{(\xi - x)^2 + (\eta - y)^2}\Big) = \frac{1}{4\pi} \ln\Big((\xi - x)^2 + (\eta - y)^2\Big)$$

It can be verifed (symbolic calculation in matlab) that when we take the Laplacian of this expression that it is zero when $(x, y) \neq (\xi, \eta)$ and that it is infinite when they coincide. The constant out front is used to normalize the magnitude of the integral of the delta function ie $\int \Delta\Phi d\Omega = 1$. Once we have the fundamental solution, we turn our attention to a particular problem. We are interested in the function $u(x, y)$ which satisfies homogeneous Laplace's equation over some domain Ω. We integrate the PDE against a test function w to obtain a weakened version of the problem:

$$\Delta u = 0 \rightarrow \int_\Omega w(x, y)\Delta u(x, y) d\Omega = 0$$

We want to integrate by parts to move derivatives off the solution u. From the product rule and the divergence theorem we can write

$$\int_\Omega \nabla \cdot (w\nabla u) d\Omega = \int_\Omega \nabla w \cdot \nabla u d\Omega + \int_\Omega w\nabla \cdot \nabla u d\Omega = \int_{\partial\Omega} w(\nabla u \cdot n) ds$$

The quantity $\nabla u \cdot n$ is called the normal derivative and can be written for brevity as $\frac{\partial u}{\partial n}$. Using these results, we can write

$$0 = \int w\nabla^2 u d\Omega = \int w\frac{\partial u}{\partial n} ds - \int \nabla w \cdot \nabla u d\Omega$$

Using similar logic, we can observe that

$$\int \nabla \cdot (u\nabla w) d\Omega = \int u\frac{\partial w}{\partial n} ds = \int \nabla w \cdot \nabla u d\Omega + \int u\nabla \cdot \nabla w d\Omega$$

Therefore,

$$-\int \nabla w \cdot \nabla u d\Omega = \int u\nabla \cdot \nabla w d\Omega - \int u\frac{\partial w}{\partial n} ds$$

The weak form integrated twice by parts is then

$$0 = \int_\Omega u\nabla^2 w d\Omega + \int_{\partial\Omega} w\frac{\partial u}{\partial n} ds - \int_{\partial\Omega} u\frac{\partial w}{\partial n} ds$$

Choose the (arbitrary) weighting function as the fundamental solution $w = \Phi(x, y, \xi, \eta)$ so that the weak form becomes

$$\int_\Omega u\delta(x - \xi, y - \eta) d\Omega = u(\xi, \eta) = -\int_{\partial\Omega} w\frac{\partial u}{\partial n} ds + \int_{\partial\Omega} u\frac{\partial w}{\partial n} ds$$

We have used the definition of the fundamental solution and the sifting property of the delta function. This is for a point (ξ, η) inside the domain, where the integral encompasses the non-zero part of the delta function. If the point (ξ, η) is outside the domain/region of integration, we get zero. There is still the case of the point lying on the boundary of the domain $\partial\Omega$. For smooth domains, it can be shown that $u(\xi, \eta)$ picks up a factor of $1/2$ when the point is on the boundary. The "governing equaion" for the boundary element method is then

$$c(\xi, \eta)u(\xi, \eta) = \int_{\partial\Omega} u\frac{\partial w}{\partial n}ds - \int_{\partial\Omega} w\frac{\partial u}{\partial n}ds$$

where $c(\xi, \eta) = 1/2$ for points on the boundary and $c(\xi, \eta) = 1$ for points inside the domain. In order to determine solutions within the domain, we first need to fully determine the function u and its normal derivative $\nabla u \cdot n$ on the boundary $\partial\Omega$. Thus for the time being we restrict attention to the boundary only. Either the normal derivative or the function value will be specified over the whole boundary (but never both simultaneously). The boundary will be chopped into N elements Γ_j. For simplicity, we will require that the function value and its flux are constant over each element, though this is not necessary. The value will be stored at a single node at the center of each element. Looking at element i on the boundary,

$$\frac{1}{2}u_i = \sum_{j=1}^{N}\left[\int_{\Gamma_j} u\frac{\partial w}{\partial n}ds\right] - \sum_{j=1}^{N}\left[\int_{\Gamma_j} w\frac{\partial u}{\partial n}ds\right]$$

We know, however, that the function values and fluxes are constant over each element. We also know that if we are looking at element i (via u_i) that the weight function corresponds to the position of that element/node. This means that the weight functions should be indexed along with the function value

$$\frac{1}{2}u_i = \sum_{j=1}^{N} u_j\left[\int_{\Gamma_j} \frac{\partial w_i}{\partial n}ds\right] - \sum_{j=1}^{N}\left(\frac{\partial u}{\partial n}\right)_j\left[\int_{\Gamma_j} w_i ds\right]$$

In other words, $u_i = u(\xi_i, \eta_i)$ which corresponds to $w_i = w(x, y, \xi_i, \eta_i)$. The weight function and its flux parameterized by the position of node i is integrated over the element Γ_j. The bracketed expressions from matrices so that the discretized system reads

$$\left(H_{ij} + \frac{1}{2}\delta_{ij}\right)u_j += G_{ij}\left(\frac{\partial u}{\partial n}\right)_j$$

$$H_{ij} := \int_{\Gamma_j} \nabla w(x, y, \xi_i, \eta_i) \cdot n_j ds$$

$$G_{ij} := \int_{\Gamma_j} w(x, y, \xi_i, \eta_i)ds$$

To be clear, $w = \Phi$ ie the weighting functions are the green's functions. Thus the BEM gives rise to a linear system quite similar to FEM. In this case, however, both the left and right hand side function value and flux vectors contain both knowns and unknowns (in general case). Thus, the system needs to be shuffled to transfer all known DOF's to one side and unknown DOF's to the other. If in a special case, only the flux or value was specified on the boundary we would not have this problem. Once the values of the function and the flux are known on the boundary, values of the function can be determined on the interior of the domain as well.