

# Galerkin Methods for PDEs

Aleksandar Ivanov

January 3, 2021

## 1 Problem Statement

Use the Galerkin method to find the coefficient

$$C = \frac{32}{\pi} \iint_{\mathcal{D}} u(\xi, \phi) \xi \, d\xi \, d\phi \quad (1)$$

in the calculation of the flux

$$\Phi = C \frac{pS^2}{8\pi\eta} \quad (2)$$

for a semicircular pipe, where  $\xi = r/R$  is the dimensionless radius and  $u = v\eta/pR^2$  and the system is described by the Navier-Stokes equation in its uniform, incompressible, laminar flow driven by a constant pressure difference form

$$\nabla^2 v = -\frac{p}{\eta}. \quad (3)$$

How is the accuracy of the method affected by the number of basis functions used.

## 2 Setup

We will be solving the Navier-Stokes equation (eq. (3)) in its dimensionless form

$$\nabla^2 u(\xi, \phi) = -1, \quad (4)$$

on the domain

$$\mathcal{D} = \left\{ (\xi \cos \phi, \xi \sin \phi) \in \mathbb{R}^2 \mid \xi \in [0, 1] \phi \in [0, \pi] \right\}. \quad (5)$$

The boundary conditions are obviously Dirichlet BCs

$$u(\partial\mathcal{D}) = 0 \quad \Leftrightarrow \quad u(1, \phi) = u(\xi, 0) = u(\xi, \pi) = 0. \quad (6)$$

The analytical solution of eq. (4) is given as an expansion over the basis of eigenfunctions of the Laplacian in this particular geometrical setup

$$u(\xi, \phi) = \sum_{m=0}^{\infty} \sum_{n=1}^{\infty} c_{(m,n)} J_{2m+1}(\zeta_{2m+1,n} \xi) \times \sin((2m+1)\phi), \quad (7)$$

where  $\zeta_{2m+1,n}$  is the  $n$ -th zero of the  $(2m+1)$ -st Bessel  $J$  function and  $c_{(m,n)}$  are the expansion coefficient of  $-1$  in the eigenbasis

$$c_{(m,n)} = (-1) \frac{\int_0^1 J_{2m+1}(\zeta_{2m+1,n} \xi) \xi \, d\xi}{\|J_{2m+1}(\zeta_{2m+1,n} \xi)\|} \times \frac{\int_0^\pi \sin((2m+1)\phi) \, d\phi}{\|\sin((2m+1)\phi)\|} \quad (8)$$

The idea of the Galerkin method [3] is to choose a finite subset of a basis set of vectors of the vector space of functions which satisfy the boundary conditions, i.e. a guess

$$\tilde{u}(\xi, \phi) = \sum_{k=1}^N c_k \Psi_k(\xi, \phi), \quad (9)$$

and with them minimize the residual

$$\varepsilon(\xi, \phi) = \nabla^2 \tilde{u}(\xi, \phi) + 1, \quad (10)$$

by making it lie in the perpendicular subspace

$$\varepsilon \in \text{span}\{\Psi_1, \dots, \Psi_N\}_\perp, \quad (11)$$

where perpendicularity is defined under the inner product

$$(\Psi_i, \Psi_j) = \int_{\mathcal{D}} \Psi_i \Psi_j \, dS. \quad (12)$$

So by choosing a proper set of functions we can get an approximation for the actual solution  $u$ .

It should be noted that the eigenfunctions can be indexed by a multi-index  $k = (k_1, \dots, k_f)$  too. This wouldn't change anything except to replace the sum in eq. (9) with a multi-sum. This is important since our problem is two-dimensional and is thus indexed by a 2-index  $(m, n)$  as in eq. (7).

To satisfy the condition from eq. (11) we take the inner product of eq. (10) with some trial function  $\Psi_i$

$$\begin{aligned} 0 &= \sum_{k=0}^N c_k (\nabla^2 \Psi_k, \Psi_i) + (1, \Psi_i) \\ &= - \sum_{k=0}^N c_k (\nabla \Psi_k, \nabla \Psi_i) + (1, \Psi_i), \end{aligned} \quad (13)$$

where the last equality holds because of integration by parts and the Dirichlet BCs, and the product in the inner product is interpreted as the scalar product of normal two-dimensional  $\mathbb{R}^2$  vectors.

This we can define the symmetric matrix and vector

$$A_{ij} = -(\nabla \Psi_i, \nabla \Psi_j), \quad b = -(1, \Psi_i) \quad (14)$$

and solve the system of linear equations

$$A\mathbf{c} = \mathbf{b}, \quad (15)$$

where  $\mathbf{c}$  is the vector of the coefficients  $c_k$ .

Computing our wanted coefficient  $C$  through the matrix form we get

$$C = -\frac{32}{\pi} \mathbf{b} \cdot \mathbf{c} \quad (16)$$

We will try out the method with the following choice of trial vectors, inspired by the analytical solution but still a simplification of it:

$$\Psi_{(m,n)}(\xi, \phi) = \xi^{2m+1} (1 - \xi)^n \sin((2m+1)\phi), \quad (17)$$

where  $m \in [0, M-1]$  and  $n \in [1, N]$ . These have the property that

$$(\Psi_{(m',n')}, \Psi_{(m,n)}) = \delta_{m'm} \frac{\pi}{2} B(n+n'+1, 4m+4), \quad (18)$$

where  $B$  is the beta function. This, in turn, gives

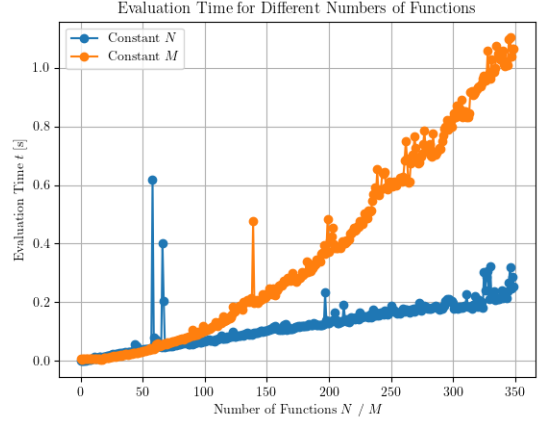


Figure 1: Evaluation times for our implementation holding either  $M = 50$  or  $N = 50$  constant.

the matrix formulation as

$$\begin{aligned} A_{(m',n')(m,n)} &= -\delta_{m'm} \frac{\pi}{2} \frac{nn'(4m+3)}{n+n'+4m+2} \\ &\quad \times B(n+n'-1, 4m+3) \end{aligned} \quad (19)$$

$$b_{(m,n)} = -\frac{2}{2m+1} B(n+1, 2m+3). \quad (20)$$

Since the angular eigenfunctions are orthogonal, the stacked system decomposes into block diagonal form, meaning that we can solve it separately for every  $m$  and do not have to deal with a huge  $MN \times MN$  matrix.

### 3 Results and Discussion

Firstly, we will test the speed of our implementation of the Galerkin method; we expect to see a linear trend as a function of  $M$  since we are solving the equations for each  $m$  separately. As a function of  $N$ , we expect more of a quadratic dependence since each time we are solving an  $N \times N$  system of equations in matrix form. There might be a slight speed up if the  $N \times N$  matrix had a special form that the equation solver was checking for, however our  $N \times N$  matrix doesn't have any special form.

This is borne out by the results shown in fig. 1; the  $M$  dependence is almost exactly linear, while the  $N$  dependence looks to be quadratic. A further confirmation that the  $N$  dependence is a power can be seen in fig. 2.

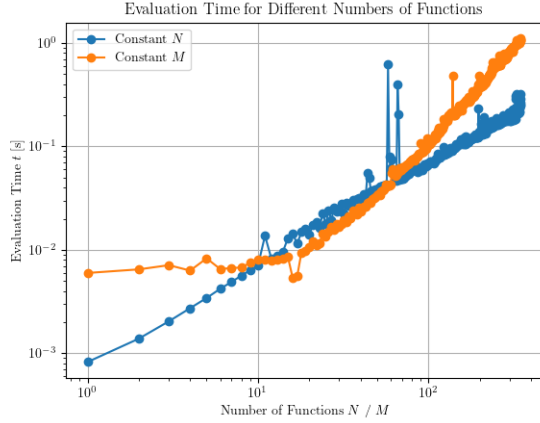


Figure 2: Evaluation times for our implementation holding either  $M = 50$  or  $N = 50$  constant, shown in a log scale.

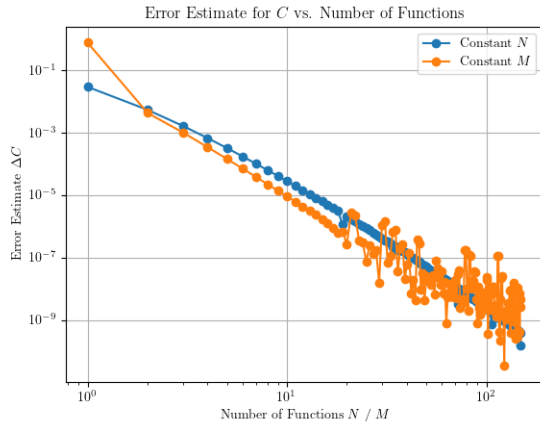


Figure 3: Error estimate for  $C$  as a function of number of trial functions, holding either  $M = 30$  or  $N = 30$  constant, shown in a log scale.

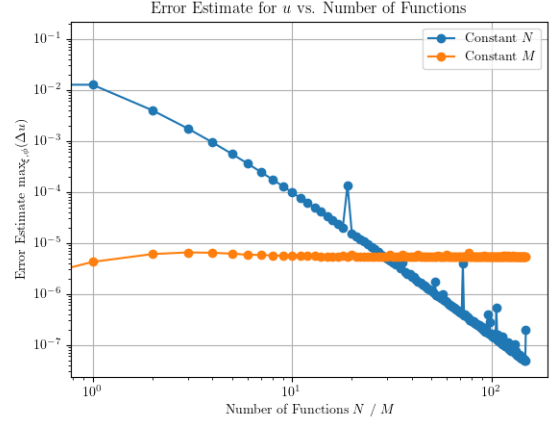


Figure 4: Error estimate for  $u$  as a function of number of trial functions, holding either  $M = 30$  or  $N = 30$  constant, shown in a log scale.

The next thing to check is what the error of the method is and how it converges. As a built-in error estimate, we will use the absolute value of a single term in the series we're summing. At this point however, even though we've been working with abstract function and their series, to actually see the result and errors, we need to set up a grid on our domain and calculate the values at those grid points.

The grid we will use is a polar coordinates grid with 500 points in the radial and angular directions, with the angular points being weighed more towards the outer edge, to get a better covering of the domain.

We will measure the error in two ways. One way will be by looking at the error in the coefficient  $C$  that we're trying to calculate, while the second will be by looking at the maximum over the grid points of each consecutive trial function. In each of these cases the estimate will obviously be an underestimate of the error, but since we mathematically know that the series converges to the right answer as long as our trial set converges to a basis (which it does), then we can at least be sure that we are getting the correct behavior if our error estimate converges to 0.

The results of this are shown in figs. 3 and 4. Figure 3 shows the described error estimate for  $C$  in a log scale. We see that the error goes to zero exponentially, independent of what parameter we change and what parameter we keep constant be-

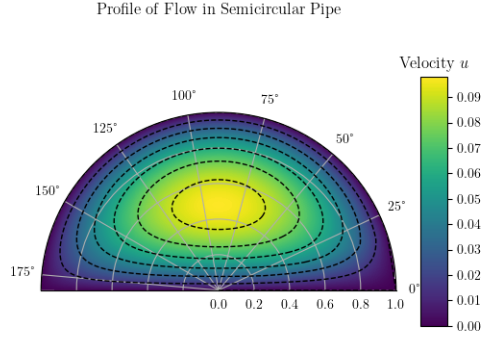


Figure 5: Velocity  $u$  flow profile for the semicircular pipe.

tween  $M$  and  $N$ . Figure 4, on the other hand, shows the maximal error in the terms of the series themselves. We see that in the  $m$ -series the terms start out with low error, but they plateau around an error of  $10^{-5}$ . This is because the first term in the  $m$ -series  $\sin(\phi)$  gets the major contribution to the solution right and all the following terms are just small corrections, however it's very hard to lower the error substantially since the corrections just get more and more oscillatory. In the case of the  $n$ -series, the behavior is totally different. It starts off very inaccurate but then it exponentially decreases. This makes sense since in that case the first term is not particularly accurate since we guessed that part completely, but all the successive terms aren't oscillatory, instead being just finer and finer corrections to the leading term.

Finally, we can plot the result from calculating  $u$  in this way. Figure 5 shows a contour plot of the velocity  $u$ . We can see that as required the velocity at the boundary is 0 and as expected it is maximal somewhere near the middle. Also drawn are some surfaces of equal velocity as dashed lines. This result agrees with the one in [1]. A more detailed analysis of a related problem can be found in [2].

From the same calculations we also get the numerical value of the coefficient  $C$  as

$$C = 0.75772189 \pm 10^{-8} \quad (21)$$

## References

- [1] Rajai Alassar. "Hagen-Poiseuille flow in tubes of semi-circular cross-sections". In: May 2011, pp. 1–2. DOI: 10.1109/ICMSAO.2011.5775524.
- [2] O.A. Elsamni, A.A. Abbasy, and O.A. El-Masry. "Developing laminar flow in curved semi-circular ducts". In: *Alexandria Engineering Journal* 58.1 (2019), pp. 1–8. ISSN: 1110-0168. DOI: <https://doi.org/10.1016/j.aej.2018.03.013>. URL: <http://www.sciencedirect.com/science/article/pii/S1110016819300055>.
- [3] Borut Paul Kerševan. *Reševanje PDE z metodo Galerkina*. Slovene. [Last accessed: 03.01.2021]. URL: [https://ucilnica.fmf.uni-lj.si/pluginfile.php/92621/mod\\_resource/content/3/11-naloga-PDE-Poisson-Galerkin.pdf](https://ucilnica.fmf.uni-lj.si/pluginfile.php/92621/mod_resource/content/3/11-naloga-PDE-Poisson-Galerkin.pdf).