Intro to UQ Assignment 4
Joy Tolia - 1103478

1 Introduction

$$\mathbb{E}\,[1]$$

For this assignment I will be reviewing a paper written by P. G. Constantine, D. F. Gleich and G. Iaccarino called A factorization of the spectral Galerkin system parameterized matrix equations: derivation and applications. The paper looks at the following problem:

- Given $s \in \mathcal{S}$ a set of input parameters in a d-dimensional tensor product parameter space:

  $$\mathcal{S} = \mathcal{S}_1 \otimes \cdots \otimes \mathcal{S}_d$$

  Where $\mathcal{S}_i$ may be bounded or unbounded. Take a bounded, separable, positive weight function on the parameter space $\omega : \mathcal{S} \to \mathbb{R}_+$, where:

  $$\omega(s) = \omega_1\,(s_1) \cdots \omega_d\,(s_d)$$

  Given an $N \times N$ matrix valued function $A(s)$ which we assume is invertible for all $s \in \mathcal{S}$:

  $$A : \mathcal{S} \to \mathbb{R}^{N \times N}$$

  And given an $N \times 1$ vectored valued function $b(s)$ where each component of $b(s)$ is square integrable with respect to ω:

  $$b : \mathcal{S} \to \mathbb{R}^{N \times 1}$$

  Then we are trying a find an $N \times 1$ vector valued function $x(s)$ which satisfies:

  $$A(s)x(s) = b(s), \quad s \in \mathcal{S} \tag{1}$$

- This kind of parameterized matrix problem comes up as an intermediate step when computing approximate solutions to a complex problem with multiple input parameters. They appear in many different areas like differential equations with random inputs, electronic circuit design, image deblurring models and ranking methods for nodes in a graph.

- After finding an approximate solution is made which is cheaper to evaluate than the true solution, we can use mean and variance of the approximate solution to represent estimates of the true solution.

- This is where we will use the concepts we learnt in lectures about spectral methods. We will use a series of orthonormal polynomial polynomial basis functions to approximate the solution $x(s)$. The basis will be over $L^2\,(\mathcal{S}, \omega; \mathbb{R})$.

- We use here what we learnt about multi variate orthonormal polynomials and use multi index notation:

  $$\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$$

  And we define the basis polynomial as:

  $$\pi_\alpha = \pi_{\alpha_1}\,(s_1) \cdots \pi_{\alpha_d}\,(s_d) \tag{2}$$

Where $\pi_{\alpha_i}(s_i)$ is a univariate orthonormal polynomial of degree $\alpha_i \in \mathbb{N}_0$. The orthogonality is defined by the weight function $\omega_i(s_i)$. Then we have:

$$\int_{\mathcal{S}} \pi_\alpha(s)\,\pi_\beta(s)\,\omega(s)\,ds = \delta_{\alpha\beta}$$

Where $\alpha = \beta$ if $\alpha_i = \beta_i$ for all $i = 1, \ldots, d$.

- For a given index set $\mathcal{I} \subset \mathbb{N}_0^d$ with size $|\mathcal{I}| < \infty$. We can use the following polynomial approximation:

$$\begin{aligned} x(s) &= \sum_{\alpha\in\mathcal{I}} \mathbf{x}_\alpha \pi_\alpha(s) \\ &= \underbrace{\begin{pmatrix} \cdots + (\mathbf{x}_\alpha)_1\,\pi_\alpha(s) + \cdots \\ \vdots \\ \cdots + (\mathbf{x}_\alpha)_N\,\pi_\alpha(s) + \cdots \end{pmatrix}}_{N\times 1}, \quad \alpha \in \mathcal{I} \\ &= \underbrace{\begin{pmatrix} & \vdots & \\ \cdots & \mathbf{x}_\alpha & \cdots \\ & \vdots & \end{pmatrix}}_{N\times|\mathcal{I}|} \underbrace{\begin{pmatrix} \vdots \\ \pi_\alpha(s) \\ \vdots \end{pmatrix}}_{|\mathcal{I}|\times 1}, \quad \alpha \in \mathcal{I} \\ &= \mathbf{X}\boldsymbol{\pi}(s) \end{aligned}$$

Where $\mathbf{x}_\alpha \in \mathbb{R}^{N\times 1}$ is the vector of coefficients of the series, corresponding to $\pi_\alpha(s)$ and $(\mathbf{x}_\alpha)_i$ is the i-th component of $\mathbf{x}_\alpha$ for $i = 1, \ldots, N$. The matrix $\mathbf{X} \in \mathbb{R}^{N\times|\mathcal{I}|}$ has columns $\mathbf{x}_\alpha$ and the parameterized vector $\boldsymbol{\pi}(s) \in \mathbb{R}^{|\mathcal{I}|\times 1}$ contains the basis polynomials. The goal of the approximation method is to find the unknown coefficients $\mathbf{X}$.

- Typically we use:

$$\mathcal{I} = \mathcal{I}_n = \left\{\alpha \in \mathbb{N}_0^d : \alpha_1 + \cdots + \alpha_d \le n\right\}$$

We have that $|\mathcal{I}_n| = \binom{n+d}{n}$, which grows quickly with $d > 1$. We have the following drawbacks:

1. The method of computing the coefficients involves solving a linear system of size $N|\mathcal{I}| \times N|\mathcal{I}|$, which can be really big even for a small number of parameters (6 to 10) and low order polynomials (degree < 5).
2. Another drawback of the Galerkin method is its limited ability to take advantage of existing solvers for the problem $A(\lambda)\,x(\lambda) = b(\lambda)$ given a parameter point $\lambda \in \mathcal{S}$. Other methods such as pseudospectral and collocation methods have distinct advantages from the point of view of code reuse and rapid implementation.

To get around these drawbacks, we replace the integration in the Galerkin method by a multivariate quadrature rule. We derive a decomposition of the linear system of equations. We get the following advantages:

1. The decomposition means we can compute the Galerkin coefficients using only the evaluations of $A(\lambda)$ and $b(\lambda)$ for quadrature nodes $\lambda \in \mathcal{S}$.

2. We only need matrix-vector multiplications as in Krylov-based iterative methods which we will look at later. These methods take full advantage of the sparsity of the parameterized system resulting in memory reduced requirements. By sparse matrices, we mean matrices with very few non-zero elements. There are very memory efficient methods of storing these matrices which we can use.
3. The decomposition gives us an intuition on how to use a preconditioner on the Galerkin system from work already done on this area.
4. The decomposition also gets bounds on the eigenvalues of the Galerkin system provided A is symmetric.

2 Derivation and Decomposition

In this section we do some of the derivation and then talk about the decomposition. We will use the index set $\mathcal{I}$ to denote the index set for the basis orthonormal multivariate polynomials. We will use the index set $\mathcal{J}$ to denote the index set for the points/weights in quadrature rule.

- We will use the bracket notation $\langle \cdot \rangle$ to denote a discrete integral with respect to the weight function ω, so for a function $f : \mathcal{S} \to \mathbb{R}$:

$$\int_{\mathcal{S}} f(s)\,\omega(s) = \sum_{\beta \in \mathcal{J}} f\left(\lambda_\beta\right) \nu_\beta =: \langle f \rangle$$

  where $\lambda_\beta = \left(\lambda_{\beta_1} + \cdots + \lambda_{\beta_d}\right) \in \mathcal{S}$ and $\left\{\left(\lambda_\beta, \nu_\beta\right)\right\}_{\beta \in \mathcal{J}}$ define a multivariate quadrature rule, where ν_β are weights.

- Let us now look at the Galerkin method we will be using to compute the coefficients of the polynomial model $\mathbf{X}\boldsymbol{\pi}(s)$. Define the residual:

$$r(y, s) = A(s)\,y(s) - b(s)$$

  Let $x_g(s)$ be the Galerkin approximation. Denote the i-th component of the residual by $r_i\left(x_g, s\right)$.The Galerkin method requires that each component of the residual be orthogonal to the approximation space defined by the span of π_α for $\alpha \in \mathcal{I}$:

$$\left\langle r_i\left(x_g\right)\pi_\alpha \right\rangle = 0, \quad i = 1, \ldots, N, \quad \alpha \in \mathcal{I}$$

Using matrix notation:

$$\begin{aligned}
\left\langle r\left(x_g\right)\boldsymbol{\pi}^T\right\rangle &= \underbrace{\begin{pmatrix} \cdots & \left\langle r_1\left(x_g\right)\pi_\alpha\right\rangle & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \left\langle r_N\left(x_g\right)\pi_\alpha\right\rangle & \cdots \end{pmatrix}}_{N\times|\mathcal{I}|}, \quad \alpha\in\mathcal{I} \\
&= \underbrace{\begin{pmatrix} \cdots & \left\langle\left(A\left(x_g\right)-b\right)_1\pi_\alpha\right\rangle & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \left\langle\left(A\left(x_g\right)-b\right)_N\pi_\alpha\right\rangle & \cdots \end{pmatrix}}_{N\times|\mathcal{I}|}, \quad \alpha\in\mathcal{I} \\
&= \left\langle\left(Ax_g-b\right)\boldsymbol{\pi}^T\right\rangle \\
&= \mathbf{0}
\end{aligned}$$

Making the substitution $x_g(s) = \mathbf{X}\boldsymbol{\pi}(s)$, we get:

$$\left\langle A\mathbf{X}\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle = \left\langle b\boldsymbol{\pi}^T\right\rangle$$

Let us quickly look at vector notation. Suppose the matrix $Z\in\mathbb{R}^{n\times m}$ is made of columns $z_i\in\mathbb{R}^{n\times 1}$ where $i=1,\ldots,m$. Then we have:

$$\operatorname{vec}(Z) = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} \in \mathbb{R}^{nm\times 1}$$

Suppose the matrix $Y\in\mathbb{R}^{p\times q}$ is made of the elements $y_{ij}\in\mathbb{R}$ with $i=1,\ldots,p$ and $j=1,\ldots,q$ then we have the following notation:

$$Y\otimes Z = \begin{pmatrix} y_{11}Z & \cdots & y_{1q}Z \\ \vdots & \vdots & \vdots \\ y_{p1}Z & \cdots & y_{pq}Z \end{pmatrix} \in \mathbb{R}^{pn\times qm}$$

Where each $y_{ij}Z\in\mathbb{R}^{n\times m}$ is a matrix for $i=1,\ldots,p$ and $j=1,\ldots,q$. Then we have an equivalent statement to $\left\langle A\mathbf{X}\boldsymbol{\pi}\boldsymbol{\pi}^T\right\rangle = \left\langle b\boldsymbol{\pi}^T\right\rangle$:

$$\left\langle\boldsymbol{\pi}\boldsymbol{\pi}^T\otimes A\right\rangle\mathbf{x} = \langle\boldsymbol{\pi}\otimes b\rangle \tag{3}$$

Where $\mathbf{x}=\operatorname{vec}(\mathbf{X})\in\mathbb{R}^{N|\mathcal{I}|\times 1}$ is a constant vector. The constant matrix $\left\langle\boldsymbol{\pi}\boldsymbol{\pi}^T\otimes A\right\rangle\in\mathbb{R}^{N|\mathcal{I}|\times N|\mathcal{I}|}$ has a distinct block structure; the α,β block is equal to $\left\langle\pi_\alpha\pi_\beta A\right\rangle\in\mathbb{R}^{N\times N}$ for multi-indices $\alpha,\beta\in\mathcal{I}$. Finally the constant vector $\langle\boldsymbol{\pi}\otimes b\rangle\in\mathbb{R}^{N|\mathcal{I}|\times 1}$ has the α block as $\langle\pi_\alpha b\rangle\in\mathbb{R}^{N\times 1}$ for the multi index $\alpha\in\mathcal{I}$.

- Once we get the form $\left\langle\boldsymbol{\pi}\boldsymbol{\pi}^T\otimes A\right\rangle$ we have a decomposition for the matrix which is written as a theorem in the publication.

Theorem. *Let $\left\{\left(\lambda_\beta,\nu_\beta\right)\right\}$ with $\beta\in\mathcal{J}$ be a multivariate quadrature rule. The matrix $\left\langle\boldsymbol{\pi}\boldsymbol{\pi}^T\otimes A\right\rangle$ can be decomposed as*

$$\left\langle\boldsymbol{\pi}\boldsymbol{\pi}^T\otimes A\right\rangle = (\mathbf{Q}\otimes\mathbf{I})\,A\,(\boldsymbol{\lambda})\,(\mathbf{Q}\otimes\mathbf{I})^T$$

Where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the $N \times N$ identity matrix, and $\mathbf{Q} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{J}|}$ is a matrix of size $|\mathcal{I}| \times |\mathcal{J}|$- one row for each basis polynomial and one column for each point in the quadrature rule. The matrix $A(\boldsymbol{\lambda}) \in \mathbb{R}^{N|\mathcal{J}|\times N|\mathcal{J}|}$ is a block diagonal matrix of size $N|\mathcal{J}| \times N|\mathcal{J}|$ where each non-zero block is $A\left(\lambda_\beta\right)$ for $\beta \in \mathcal{J}$.

- Let us look at what $(\mathbf{Q} \otimes \mathbf{I})$ and $A(\boldsymbol{\lambda})$ look like. First the following are the elements of $\mathbf{Q}$:

  $$q_{\alpha\beta} = \sqrt{\nu_\beta}\pi_\alpha\left(\lambda_\beta\right), \quad \alpha \in \mathcal{I}, \quad \beta \in \mathcal{J}$$

  We get the decomposition because using the quadrature rules we have the following:

  $$\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle = \sum_{\beta\in\mathcal{J}} \left[\boldsymbol{\pi}\left(\lambda_\beta\right)\boldsymbol{\pi}\left(\lambda_\beta\right)^T \otimes A\left(\lambda_\beta\right)\right]\nu_\beta$$

  Then we define the vectors:

  $$\mathbf{q}_\beta = \sqrt{\nu_\beta}\boldsymbol{\pi}\left(\lambda_\beta\right)$$

  Then we have:

  $$\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle = \sum_{\beta\in\mathcal{J}} \mathbf{q}_\beta\mathbf{q}_\beta^T \otimes A\left(\lambda_\beta\right)$$

  Where $\mathbf{q}_\beta$ form the columns for the matrix $\mathbf{Q}$. and this forms the final decomposition.

  $$(\mathbf{Q} \otimes \mathbf{I}) = \begin{pmatrix} & \vdots & \\ \cdots & q_{\alpha\beta}\mathbf{I} & \cdots \\ & \vdots & \end{pmatrix} \in \mathbb{R}^{N|\mathcal{I}|\times N|\mathcal{J}|}, \quad \alpha \in \mathcal{I}, \quad \beta \in \mathcal{J}$$

  Where $\mathbf{I} \in \mathbb{R}^{N\times N}$.

  $$A(\boldsymbol{\lambda}) = \begin{pmatrix} \ddots & & 0 \\ & A\left(\lambda_\beta\right) & \\ 0 & & \ddots \end{pmatrix} \in \mathbb{R}^{N|\mathcal{J}|\times N|\mathcal{J}|}, \quad \beta \in \mathcal{J}$$

  Where $A\left(\lambda_\beta\right) \in \mathbb{R}^{N\times N}$ for $\beta \in \mathcal{J}$. So finally $(\mathbf{Q} \otimes \mathbf{I})\, A(\boldsymbol{\lambda})\, (\mathbf{Q} \otimes \mathbf{I})^T \in \mathbb{R}^{N|\mathcal{I}|\times N|\mathcal{J}|}$.

- As $A(s)$ depends polynomially on $s \in \mathcal{S}$, we have that each integrand in the matrix $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ is a polynomial on $s \in \mathcal{S}$. Therefore, as we have learnt from lectures there exists a Gaussian quadrature rule so that we get the exact true Galerkin matrix at the end.

- We have for two rows $\mathbf{r}_\alpha, \mathbf{r}_\beta$ of the matrix $\mathbf{Q}$ for $\alpha, \beta \in \mathcal{I}$ the following relationship:

  $$\mathbf{r}_\alpha\mathbf{r}_\beta^T = \sum_{\gamma\in\mathcal{J}} \pi_\alpha\left(\lambda_\gamma\right)\pi_\beta\left(\lambda_\gamma\right)\nu_\gamma$$

  Similarly, from lectures if the quadrature rule is a tensor product Gaussian quadrature rule of sufficiently high order to compute the exact integrand then we get $\mathbf{Q}\mathbf{Q}^T = \mathbf{I} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$.

- We will now assume that the quadrature rule we use gives us the above property of $\mathbf{Q}$ being an orthogonal matrix. We also assume that $|\mathcal{I}| \leq |\mathcal{J}|$ otherwise the Galerkin matrix $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ will be rank deficient.

- We get the following corollary from the theorem which yields bounds on the spectrum of the matrix $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ for symmetric A:

Corollary. *Suppose $A(s)$ is symmetric for all $s \in \mathcal{S}$. The eigenvalues of $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ satisfy the bounds:*

$$\min_{\beta \mathcal{J}} \left[\theta_{\min}\left(A\left(\lambda_\beta\right)\right)\right] \leq \theta\left(\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle\right) \leq \max_{\beta \mathcal{J}} \left[\theta_{\max}\left(A\left(\lambda_\beta\right)\right)\right]$$

where $\theta(X)$ denotes the eigenvalues of a matrix X, and $\theta_{\min}(X)$ and $\theta_{\max}(X)$ denote the smallest and largest eigenvalues of X, respectively.

- Notice, from this corollary that we $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ is positive definite if $A(s)$ is positive definite for all $s \in \mathcal{S}$.

3 Iterative Solvers

- To solve $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle \mathbf{x} = \langle \boldsymbol{\pi} \otimes b \rangle$, instead of inverting the matrix $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ which will have the same computational cost as a matrix-matrix multiplication. We want to use Krylov Based iterative methods. Suppose we are trying to solve

  $$Ax = b$$

  Then a Krylov based solver uses an iterative method to approximate x where x_i, the approximation to x for the i-th iteration, are in the Krylov subspace $K_n(A, b)$ defined by:

  $$K_n(A, b) = \text{span}\left\{b, Ab, \ldots, A^{n-1}b\right\}$$

  The Krylov based methods give us advantages because:

  1. We only need matrix-vector multiplication instead of matrix-matrix multiplication which is what would be needed to invert a matrix.
  2. We can use the spareness of our system to our advantage using Krylov iterative methods to increase memory efficiency.

- Going back to our system let us look at the iterative method for the matrix-vector multiplication. Given a vector $\mathbf{u} = \text{vec}(\mathbf{U})$, suppose we want to compute:

  $$\mathbf{v} = \text{vec}(\mathbf{V}) = (\mathbf{Q} \otimes \mathbf{I}) A(\boldsymbol{\lambda}) (\mathbf{Q} \otimes \mathbf{I})^T \mathbf{u}$$

  We can do this in 3 steps:

  1. $\mathbf{W} = \mathbf{U}\mathbf{Q}$. Let $\mathbf{w}_\beta$ be a column of $\mathbf{W}$ with $\beta \in \mathcal{J}$.
  2. For each β, $\mathbf{y}_\beta = A\left(\lambda_\beta\right)\mathbf{w}_\beta$. Define $\mathbf{Y}$ to be the matrix with columns $\mathbf{y}_\beta$.
  3. $\mathbf{V} = \mathbf{Y}\mathbf{Q}^T$.

- Steps 1 and 3 each require $N|\mathcal{I}||\mathcal{J}|$ multiplications. If a matrix-vector with $A(s)$ takes $O(N)$ operations due to its sparsity patterns then step 2 takes $O(N|\mathcal{J}|)$ operations. Another advantage is that this can be accomplished without any knowledge of the specific type of parameter dependence in $A(s)$ which means we have a reusable interface for this implementation.

4 Preconditioning

- In iterative methods, convergence and divergence of the method is affected by the condition number of the matrix. Given an invertible matrix A, the condition number of A, $\kappa(A)$ is defined by:

$$\kappa(A) = \left|\frac{\lambda_{\max}}{\lambda_{\min}}\right|$$

  Where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalue of A, respectively.

- The smaller the condition number, the more likely the iterative method will converge and the larger the condition number the more likely the iterative method will diverge [1]. Which also means we need less iterations to converge for a small condition number means better convergence rate. So given the problem of finding x such that $Ax = b$ but supposing A is ill conditioned, i.e. it has a large condition number. We can adjust the problem by introducing a matrix P which we construct which is easily invertible to get a new problem:

$$Cx = P^{-1}Ax = P^{-1}b = d$$

  Where $C = P^{-1}A$ is a lot better conditioned due to the construction of P which means our iterative method is more likely to converge and converges faster. We call the matrix P the preconditioner, it is usually very difficult to construct or find such a preconditioner.

- However the decomposition we have found makes this much easier for us. Suppose we have $\mathbf{P} \in \mathbb{R}^{N\times N}$ that is easily invertible. We can then construct a block diagonal preconditioner $\mathbf{I} \otimes \mathbf{P}^{-1} \in \mathbb{R}^{N|\mathcal{I}|\times N|\mathcal{I}|}$ where $\mathbf{I} \in \mathbb{R}^{|I|\times|I|}$. If we multiply on the left of $\left\langle \boldsymbol{\pi}\boldsymbol{\pi}^T \otimes A \right\rangle$ by our preconditioner then we get:

$$\left(\mathbf{I}_{|\mathcal{I}|} \otimes \mathbf{P}^{-1}\right)(\mathbf{Q} \otimes \mathbf{I}_N)\, A(\boldsymbol{\lambda})\, (\mathbf{Q} \otimes \mathbf{I}_N)^T = (\mathbf{Q} \otimes \mathbf{I}_N)\left(\mathbf{I}_{|\mathcal{J}|} \otimes \mathbf{P}^{-1}\right) A(\boldsymbol{\lambda})\, (\mathbf{Q} \otimes \mathbf{I}_N)^T$$

  Where $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, by the mixed product property and commutativity of the identity matrix, $\mathbf{I} \otimes \mathbf{P}^{-1}$ slips past $\mathbf{Q} \otimes \mathbf{I}$ to act directly on $A(\boldsymbol{\lambda})$. The non-zero blocks of the diagonal matrix $\left(\mathbf{I}_{|\mathcal{J}|} \otimes \mathbf{P}^{-1}\right) A(\boldsymbol{\lambda})$ are $\mathbf{P}^{-1}A\left(\lambda_\beta\right)$ for $\beta \in \mathcal{J}$. So we only have to choose one constant matrix $\mathbf{P}$ to affect the parameterized system of all quadrature points.

- A reasonable an popular choice is the mean $\mathbf{P} = \langle A \rangle$. The next part in the paper, they analyse different preconditioners for solution of the parametrised elliptic PDE with homogeneous Dirichlet boundary conditions. The following different ways were used to calculate the preconditioners:

  1. $\mathbf{P} = \mathbf{I}$ which means no preconditioning.
  2. $\mathbf{P} = A(s_r)$ where $s_r \in \mathcal{S}$ is randomly picked.
  3. $\mathbf{P} = A(s_{\max})$ where $A(s_{\max})$ is the matrix with the largest eigenvalue for $s \in \mathcal{S}$.
  4. $\mathbf{P} = A(s_{\min})$ where $A(s_{\min})$ is the matrix with the smallest eigenvalue for $s \in \mathcal{S}$.
  5. $\mathbf{P} = A(s_{\text{mid}})$ where s_{mid} is the midpoint of the domain $\mathcal{S}$.
  6. $\mathbf{P} = \langle A \rangle$.

  The results of this analysis was that the methods 5 and 6 got the convergence with the lowest number of iterations. Methods 2-6 converged much faster in terms of iterations than method 1. However the preconditioning made little or no difference to the iteration time. Which means if preconditioning takes slightly more time per iteration, it is worth it as the solution converges in much fewer iterations.

5 Summary

- In the paper, they show this method applied to heat transfer with uncertain material properties.
- We have looked at the system arising from a spectral Galerkin approximation of a vector value solution $x(s)$ to the parameterized matrix equation

$$A(s)\,x(s) = b(s)$$

- These problems come up in PDE models where the parameterized inputs or random inputs are discretized in space and a Galerkin projection with an orthonormal basis is used for approximation in the parameter space.
- We then derived the system we needed to solve and found a decomposition of the matrix involved in this system.
- This decomposition helped us get possible bounds on the spectrum of the matrix.
- We worked out that we could use Krylov based iterative methods to solve the system due to the spareness of the system.
- We also worked out that matrix-vector multiplications of Galerkin matrix can be computed with only the action of $A(s)$ on a vector point in the parameter space. This gave us a reusable interface for the implementation of the Galerkin method.
- Due to the decomposition we looked at the possibility of using and constructing precondtitioners to get better convergence. By doing a specific study we saw that the midpoint and mean method to find the preconditioner gave us the best results.

References

[1] A. Pyzara, B. Bylina and J. Bylina. ”The influence of a matrix condition number on iterative methods convergence”. ISBN 978-83-60810-22-4 pp. 459464