Project

Instructions

This project will count for 1/3 of the final grade for the course. You are expected to complete this work individually, and you will need to upload your work on the Moodle page by Sunday December 29th, 23:59. The submission of a project will result in a debriefing oral exam, after which your project grade will be determined. You are ***not*** *expected to submit a report for the project. You will only need to submit your code and the plots that you find relevant. For the oral exam, you should be able to answer the questions in the project and you should be able to comment your code, implementation choices and outputs of your code.*

The project has to be written in Julia and it will be tested with Julia v.1.11.

1 Random Schrödinger operator

In this project, the eigenfunctions associated to the lowest eigenvalues of $-\Delta + V$ with homogeneous Dirichlet boundary conditions on $(0,1)^d$, with $d = 1$ or 2. The operator V is a multiplication operator by a random function that will be defined below.

We want to compute the eigenpairs using a finite-difference approximation.

1.1 One-dimensional case

In this case, for $d = 1$, the Laplacian operator $-\Delta$ is approximated by the tridiagonal matrix $\mathcal{L} \in \mathbb{R}^{N\times N}$ with $\mathcal{L}_{ii} = 2$ for $1 \leq i \leq N$ and $\mathcal{L}_{i,i+1} = \mathcal{L}_{i+1,i} = -1$.

The random multiplication operator will be taken as the diagonal matrix $v \in \mathbb{R}^{N\times N}$ such that

$$v_{ii} = \begin{cases} 1 \text{ with probability } \dfrac{1}{2} \\ \dfrac{1}{N^2} \text{ with probability } \dfrac{1}{2}. \end{cases} \tag{1}$$

1. Write a function `rand_schrodinger_1d` that returns a `SymTridiagonal` matrix implementing $\mathcal{L} + v$.

2. For $N = 1000$, compute the solution of $(\mathcal{L} + v)x = \mathbf{1}$ where $\mathbf{1} \in \mathbb{R}^N$ is the vector filled with ones.

3. Compute the lowest eigenvalue of $\mathcal{L} + v$ using the inverse power method. The matrix inversion must be done using the conjugate-gradient method, with a tight enough tolerance.

4. Implement the following deflation algorithm and compute the 5 lowest eigenvalues and eigenvectors of $\mathcal{L} + v$.

   Warning: the matrix inversion has to be done with the conjugate gradient method, with an appropriate tolerance.

Algorithm 1 Deflation algorithm

---

function DEFLATION$(A, x, \varepsilon_{\mathrm{tol}}, n_{\mathrm{eig}})$
  $\Lambda =$ `zeros`(n_{eig})
  $Y =$ `zeros`(n_{eig}, N^d) # N^d size of x
  for n in $1{:}n_{\mathrm{eig}}$ **do**
    $\Lambda[n], Y[n,:] =$ `PowerMethod`$(A^{-1} - \sum_{i=1}^{n-1} \frac{1}{\lambda_i} y_i y_i^*, x, \varepsilon_{\mathrm{tol}})$ # $\lambda_i = \Lambda[i], y_i = Y[i,:]$
  end for
  return (Λ, Y)
end function

---

5. Plot the norm of $y_m - y_1$ and $|\lambda^{(m)} - \lambda_1|$ where $\lambda^{(m)}, y^{(m)}$ are the iterates of the inverse power method where (λ_1, y_1) satisfy $(\mathcal{L} + v)y_1 = \lambda_1 y_1$ for the lowest eigenvalue of $\mathcal{L} + v$. Compare this plot with $\left(\frac{\lambda_1}{\lambda_2}\right)^k$.

6. Plot on the same graph $x/\|x\|$ solution to $(\mathcal{L} + v)x = \mathbf{1}$ and the 5 eigenvectors associated to the lowest eigenvalues of $\mathcal{L} + v$ using Algorithm 1.

1.2 Two-dimensional case

In the two-dimensional case ($d = 2$), we discretise the 2D-Laplacian operator $-\Delta = -\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2}$ using finite-differences. We want to compute the eigenpairs associated to the lowest eigenvalues of $-\Delta + V$ with homogeneous Dirichlet boundary conditions where V is a random potential, which will be defined below.

The Laplacian is approximated by the 5-point stencil finite-difference. The corresponding matrix is $\mathcal{L}^{(2)} \in \mathbb{R}^{N^2 \times N^2}$ defined by

$$\mathcal{L}^{(2)} = \mathrm{id}_N \otimes \mathcal{L} + \mathcal{L} \otimes \mathrm{id}_N,$$

where $\otimes$ is the Kronecker product given by $(A \otimes B)_{i_1 i_2; j_1 j_2} = A_{i_1, j_1} B_{i_2, j_2}$ for all $1 \leq i_1, i_2, j_1, j_2 \leq N$.

The matrix vector product $\mathcal{L}^{(2)}x$ for $x \in \mathbb{R}^{N\times N}$ can be efficiently performed by realising that

$$\mathcal{L}^{(2)}x = \mathcal{L}x + x\mathcal{L}$$

i.e. for all $1 \leq j_1, j_2 \leq N$

$$(\mathcal{L}^{(2)}x)_{j_1 j_2} = \sum_{k=1}^{N} \mathcal{L}_{j_1 k} x_{k j_2} + \sum_{k=1}^{N} x_{j_2 k} \mathcal{L}_{k j_1}.$$

The discretised random potential v is taken of the form $v = v_1 \otimes v_2$ where v_1 and v_2 are of the form (1). This means that the matrix-vector product is also efficient and for any $x \in \mathbb{R}^{N\times N}$ is given by for all $1 \leq j_1, j_2 \leq N$

$$(vx)_{j_1 j_2} = (v_1)_{j_1} (v_2)_{j_2} x_{j_1 j_2}.$$

7. Implement a function `Hmatvec(L,v,w,x)` where

   - `L` is the one-dimensional finite-difference Laplacian
   - `v` and `w` is a potential given by (1)
   - `x`$\in \mathbb{R}^{N\times N}$

   that returns `y`$\in \mathbb{R}^{N\times N}$ the result of the matrix-vector product $(\mathcal{L}^{(2)} + v)x$.

8. For $N \in \{50, 100, 200\}$ compute the solution to $(\mathcal{L}^{(2)} + v)x = \mathbf{1}$ where $\mathbf{1}$ is the vector filled with 1 using the conjugate gradient algorithm. Plot the solution using `heatmap` from the package `Plots`.

9. Compute the 5 lowest eigenvalues of $\mathcal{L}^{(2)} + v$ using the deflation algorithm 1. Again the matrix inversion has to be done with the conjugate gradient algorithm.

10. Compare $x/\|x\|$ solution to $(\mathcal{L}^{(2)} + v)x = \mathbf{1}$ and the eigenvectors computed in the previous question.