

Exercise Sheet No. 4

“Computational Condensed Matter Theory”

1 Arnoldi method

In preparation for upcoming problems we implement and study methods to construct an orthonormal basis of the Krylov subspace $\mathcal{K}_m(\mathbf{v})$,

$$\mathcal{K}_m(\mathbf{v}) = \text{span} \{ \mathbf{v}; A\mathbf{v}; \dots; A^{m-1}\mathbf{v} \}.$$

a) Classical Gram-Schmidt Orthogonalization

Write a function which takes as arguments a set of vectors and performs a classical Gram-Schmidt orthogonalization (Alg. 1) of these vectors.

Now consider a Hamiltonian as described on exercise sheet 3 in exercise 1a). Using the starting vector $\mathbf{v} = (1, 1, \dots, 1)^T$, construct a basis $\{ \mathbf{v}; \hat{H}\mathbf{v}; \dots; \hat{H}^{m-1}\mathbf{v} \}$ for $m = 20$ explicitly. Use the Gram-Schmidt method to explicitly calculate an orthonormal basis set $\{ \mathbf{v}_i \}$. Calculate the matrix elements $H_{ij}^a = \mathbf{v}_i^T \hat{H} \mathbf{v}_j$ and diagonalize the matrix H^a .

b) Modified Gram-Schmidt Orthogonalization

Write a function analogous to the previous exercise that implements the modified Gram-Schmidt orthogonalization (Alg. 2). Use this new function to calculate a new orthonormal basis $\{ \mathbf{v}_i \}$, the corresponding matrix elements H_{ij}^b and the eigenvalues of H^b . Use the same initial basis $\{ \mathbf{v}; \hat{H}\mathbf{v}; \dots; \hat{H}^{m-1}\mathbf{v} \}$ as in a).

c) Arnoldi Method

Now implement the Arnoldi method to obtain a third basis: Write a function which takes as arguments a matrix A , an initial vector $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|$, and the size of the Krylov subspace m . Using the algorithm introduced in the lecture the function should return the upper Hessenberg matrix \bar{H} and the matrix $V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$.

Using this function, calculate V_m and \bar{H} for the Hamiltonian of exercise sheet 3 exercise 1a) $\mathbf{v} = (1, 1, 1, 1, \dots, 1)^T$ and $m = 20$. Calculate the spectrum of the matrix H^c , which is obtained from \bar{H} by eliminating the last row.

d) Comparison of Methods

Test the orthogonality of the vectors $\{ \mathbf{v}_i \}$ for all three given methods and compare. For the Arnoldi method, examine the off-tridiagonal elements of the matrix H^c . Compare the spectra of H^a , H^b , and H^c to the spectrum of the full Hamiltonian \hat{H} . Discuss your results. Why do people prefer modified Gram-Schmidt over classical Gram-Schmidt for orthonormalization? Why do people prefer the Arnoldi method of exercise c) to the explicit construction of the Krylov space done in exercises a) and b)?

2 Lanczos Method

The Lanczos method for symmetric matrices as introduced in the lecture, enables the calculation of selected eigenvalues of huge systems not treatable by full diagonalization. Consider the Hamiltonian for a 1d chain with on-site disorder from exercise sheet 3, but construct it as a sparse matrix for a system size $L = 100\,000$. Implement and apply the Lanczos method to calculate the lowest lying eigenvalue and eigenvector for both H and H^2 . Terminate if the eigenvalue differs by less than 10^{-5} (absolute value) between two iterations. Visualize the eigenvectors. Can one trust these results?

Algorithm 1: Classical Gram-Schmidt Orthogonalization

1. **Input:** Set of vectors \mathbf{u}_i , $i \in \{1, \dots, m\}$
2. **Output:** $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$
3. $\mathbf{v}_1 \leftarrow \mathbf{u}_1 / \|\mathbf{u}_1\|$
4. For $j = 2, \dots, m$ Do
 - (a) $\mathbf{w} \leftarrow \mathbf{u}_j$
 - (b) For $k = 1, \dots, j - 1$ Do
 - i. $\mathbf{w} \leftarrow \mathbf{w} - (\mathbf{v}_k^\dagger \mathbf{u}_j) \mathbf{v}_k$
 - (c) End-For
 - (d) $\mathbf{v}_j \leftarrow \mathbf{w} / \|\mathbf{w}\|$
5. End-For

Algorithm 2: Modified Gram-Schmidt Orthogonalization

1. **Input:** Set of vectors \mathbf{u}_i , $i \in \{1, \dots, m\}$
2. **Output:** $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$
3. $\mathbf{v}_1 \leftarrow \mathbf{u}_1 / \|\mathbf{u}_1\|$
4. For $j = 2, \dots, m$ Do
 - (a) $\mathbf{w} \leftarrow \mathbf{u}_j$
 - (b) For $k = 1, \dots, j - 1$ Do
 - i. $\mathbf{w} \leftarrow \mathbf{w} - (\mathbf{v}_k^\dagger \mathbf{w}) \mathbf{v}_k$
 - (c) End-For
 - (d) $\mathbf{v}_j \leftarrow \mathbf{w} / \|\mathbf{w}\|$
5. End-For

Useful functions:

- numpy: dot, shape
- numpy.linalg: norm
- scipy.sparse: diags