Improving robustness of the FEAST algorithm and solving eigenvalue problems from graphene nanoribbons

Martin Galgon[1], **Lukas Krämer**[1,*], **Bruno Lang**[1], **Andreas Alvermann**[2], **Holger Fehske**[2], and **Andreas Pieper**[2]

[1] Bergische Universität Wuppertal, 42097 Wuppertal, Germany
[2] Ernst-Moritz-Arndt-Universität Greifswald, Institute of Physics, Felix-Hausdorff-Str. 6, 17489 Greifswald, Germany

We consider the FEAST eigensolver, introduced by Polizzi in 2009 [5]. We describe an improvement concerning the reliability of the algorithm and discuss an application in the solution of eigenvalue problems from graphene modeling.

1 Introduction

The FEAST algorithm [5] is an algorithm aimed at solving generalized eigenvalue problems $\mathsf{AX} = \mathsf{BX\Lambda}$, where A, B are $n \times n$ matrices. In [4] we presented a short analysis of FEAST involving this eigenvalue problem. In the present work we focus on the real symmetric eigenvalue problem $\mathsf{AX} = \mathsf{X\Lambda}$, where $\mathsf{A} = \mathsf{A}^\mathsf{T}$, $\mathsf{X}^\mathsf{T}\mathsf{X} = \mathsf{I}$ and $\mathsf{\Lambda}$ is a diagonal matrix consisting of real eigenvalues. These eigenvalues are sought in a given interval $I_\lambda = [\underline{\lambda}, \overline{\lambda}]$. The FEAST method aims at solving this particular eigenvalue problem by performing Rayleigh–Ritz [6]. The involved subspace is spanned by

$$\mathsf{U} := \frac{1}{2\pi\mathbf{i}} \int_{\mathcal{C}} (z\mathsf{I} - \mathsf{A})^{-1} \mathsf{Y} \mathrm{d}z, \tag{1}$$

where Y is a starting base with $\widetilde{m} \leq n$ columns. The curve $\mathcal{C}$ is a closed curve in $\mathbb{C}$, supposed to encircle I_λ. The integral is typically approximated via numerical integration.

2 Counting eigenvalues

In [4], several minor shortcomings of FEAST were identified. One was the problem of choosing the initial dimension of the search space, i. e., the number of columns of Y in (1). Since this number must be larger or equal to the number of A's eigenvalues in I_λ, this amounts to counting (or at least estimating the number of) those eigenvalues. Several methods for counting eigenvalues based on the use of (1) have been proposed in [2, 3, 7]. In our experience, the most reliable one has proven to be based on certain singular values. An analysis has been given in [7]. For an SVD of U, $\mathsf{U} = \mathsf{W\Sigma V}^\mathsf{T}$, it can be shown that the number of singular values (i. e., diagonal entries of $\mathsf{\Sigma}$) larger or equal than $1/2$ is a very good estimation for the number of eigenvalues of A inside I_λ. In FEAST, the computation of $\mathsf{B}_\mathsf{U} := \mathsf{U}^\mathsf{T}\mathsf{U}$ is necessary and we have $\mathsf{B}_\mathsf{U} = \mathsf{V\Sigma}^\mathsf{T}\mathsf{\Sigma V}^\mathsf{T}$, meaning the number of eigenvalues inside I_λ can well be approximated by the number of singular values $\geq 1/4$ of B_U. Letting q denote this number (or a number slightly larger), we can set $\mathsf{U}' := \mathsf{U} \cdot \mathsf{V}(:, 1:q)$, yielding a basis of the desired eigenspace.

3 Graphene modeling

The example matrices considered here arise from the modeling of graphene tubes and ribbons in the tight-binding approach [1]. They are band matrices with dimension WL and bandwidth $\leq 4\min\{W, L\}+5$ for a graphene sample of size $W \times L$. Constant off-diagonal elements correspond to electron hopping between neighboring carbon atoms. The main diagonal contains random entries $A_{ii} \in [-\gamma, \gamma]$ accounting for on-site disorder potentials. The spectrum of these matrices is contained in the interval $[-3-\gamma, 3+\gamma]$. For physics applications one is interested in the inner eigenvalues and eigenvectors close to the center of the spectrum because they determine the electronic properties of the graphene samples such as the electrical conductivity. Examples of eigenvalue distributions from graphene modeling are shown in Figs. 1, 2.

4 Numerical results

We applied the FEAST algorithm with the presented eigenvalue counting method to three eigenvalue problems arising in graphene modeling, each of size approximately 1.2×10^6. A parallel variant of the method has been implemented in C using MPI. For the numerical integration we used Gauß–Legendre integration with 8 integration points. This leads to a numerical

* Corresponding author: Email lkraemer@math.uni-wuppertal.de, phone +49 202 439 2913, fax +49 202 439 2912

Matrix	$\underline{\lambda}$	$\overline{\lambda}$	est.	found	n_{it}	tol	r_{min}	r_{max}
Graphene I	-0.046	0.046	535	535	7	10^{-8}	$4 \cdot 10^{-14}$	$5 \cdot 10^{-11}$
Graphene II (ℓ)	-0.2085	-0.195	494	494	3	10^{-8}	$4 \cdot 10^{-12}$	$6 \cdot 10^{-11}$
Graphene II (r)	0.14	0.205	263	263	4	10^{-8}	$1 \cdot 10^{-10}$	$1 \cdot 10^{-9}$
Graphene III	-0.035	0.035	394	394	3	10^{-8}	$2 \cdot 10^{-11}$	$5 \cdot 10^{-10}$

Table 1: Numerical results. The columns contain (in this order): interval boundaries, estimated and found number of eigenpairs in the interval, number of FEAST iterations, residual tolerance, minimum and maximum eigenpair residual $\|\mathsf{Ax} - \mathsf{x}\lambda\|$ for eigenvector x and eigenvalue $\lambda \in I_\lambda$.

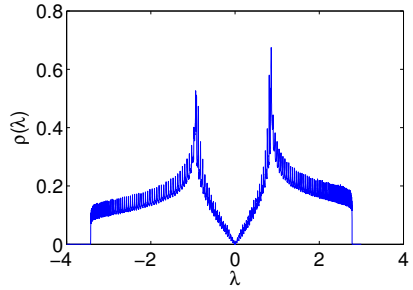

Fig. 1: Density of states $\rho(\lambda)$, i.e., the average number of eigenvalues as a function of the eigenvalues λ, computed with the kernel polynomial method [8]. The plot corresponds to Graphene III, normalized such that $\int \rho(\lambda) \mathrm{d}\lambda = 1$.

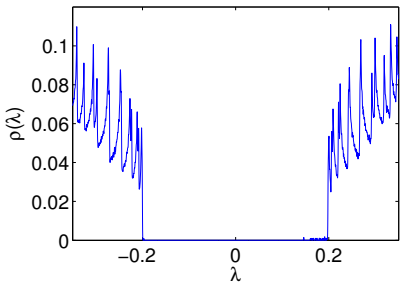

Fig. 2: Zoom into the interval $[-0.35, 0.35]$ for Graphene II. It can be seen that the spectrum has a gap around $\lambda = 0$.

approximation of the form $\mathsf{U} \approx \sum_{j=1}^{8} \omega_j (z_j \mathsf{I} - \mathsf{A})^{-1} \mathsf{Y}$, where z_j denote certain points on the contour $\mathcal{C}$ and ω_j are scalars. The matrices $(z_j \mathsf{I} - \mathsf{A})$ were banded with semi-bandwidth 248, having in average 12 nonzero entries per row. Thus, the linear systems $(z_j \mathsf{I} - \mathsf{A})\mathsf{V} = \mathsf{Y}$ were solved using a parallel banded solver. The results are shown in Table 1. As it can be seen, we sought eigenpairs with eigenvalue around zero. The residuals were required to be below $\mathrm{tol} = 10^{-8}$, in fact most of the residuals were much smaller. The matrix Graphene II had a gap in the spectrum around zero, hence it was reasonable to treat the parts to the left (ℓ) and right (r) of zero independently. The number of eigenvalues in I_λ was estimated with the SVD method from Sec. 2, while the actual subspace dimension was set to 1.25 times this number. When the estimated number of eigenpairs was computed, the algorithm was stopped. For the level of orthogonality $\max_{i \neq j} |\mathsf{x}_i^{\mathsf{T}} \mathsf{x}_j|$, where the eigenvalues belonging to $\mathsf{x}_i, \mathsf{x}_j$ resided in I_λ, we obtained values of order 10^{-15} to 10^{-13}. In the case Graphene II we obtained a level of orthogonality of order 10^{-11} between the eigenvectors belonging to the intervals on the left and right of zero. This is remarkable because, typically, independently computed eigenvectors show a worse level of orthogonality [4].

5 Conclusion

We applied FEAST to the eigenvalue problems arising in graphene modeling. Our implementation proved to be reliable, delivering high quality numerical results within a reliable number of iterations. For those problems having a gap around zero in the spectrum, the orthogonality issues were not serious even for independently computed eigenvectors.

Acknowledgements This work was supported by DFG within the priority programme 1648—Software for Exascale Computing (SPPEXA), project ESSEX (Equipping Sparse Solvers for Exascale).

References

[1] A. H. Castro Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov, and A.K. Geim, The electronic properties of graphene, Rev. Mod. Phys., **81**, 109–162, (2009).
[2] E. Di Napoli, E. Polizzi, and Y. Saad, Efficient estimation of eigenvalue counts in an interval, http://arxiv.org/abs/1308.4275, (2013).
[3] M. Galgon, L. Krämer, and B. Lang, Counting eigenvalues and improving the integration in the FEAST algorithm, Preprint BUW-IMACM 12/22, (2012).
[4] L. Krämer, E. Di Napoli, M. Galgon, B. Lang, and P. Bientinesi, Dissecting the FEAST algorithm for generalized eigenproblems, J. Comput. Appl. Math., **244**, 1–9, (2012).
[5] E. Polizzi, Density-matrix-based algorithm for solving eigenvalue problems, Phys. Rev. B, **79**, 115112, (2009).
[6] G. Stewart, Matrix Algorithms II: Eigensystems (SIAM, Philadelphia, PA, 2001).
[7] P. Tang and E. Polizzi, Subspace Iteration with Approximate Spectral Projection, http://arxiv.org/abs/1302.0432, (2013).
[8] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske, The kernel polynomial method, Rev. Mod. Phys., **78**, 275–306, (2006).