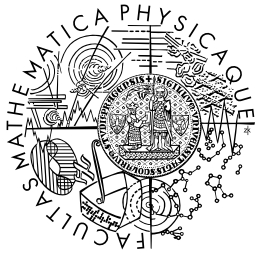


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Martina Šarmanová

Mathematical modeling of vibrational dynamics in electron scattering from molecule.

Institute of Theoretical Physics

Supervisor of the master thesis: doc. RNDr. Martin Čížek, Ph.D.

Study programme: Mathematical Modelling in Physics and Technology

Study branch: Mathematics

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

First and foremost, I would like to express my deepest gratitude to my supervisor, doc. RNDr. Martin Čížek, Ph.D., for ever encouraging and motivating guidance. This project would not have been possible without his patient guidance that enabled me to develop an understanding of the subject. I would also like to thank prof. Ing. Miroslav Tůma, CSc. and prof. Ing. Zdeněk Strakoš, DrSc. for invaluable assistance and insights that helped me finalize the project.

Title: Mathematical modeling of vibrational dynamics in electron scattering from molecule.

Author: Martina Šarmanová

Institute: Institute of Theoretical Physics

Supervisor: doc. RNDr. Martin Čížek, Ph.D., Institute of Theoretical Physics

Abstract: In this work, we study low-energy electron-molecule collisions. To understand these processes, it is necessary to explain the phenomena appearing in so called 2D electron energy-loss spectra. These spectra are very different for different molecules, which we still cannot explain satisfactorily. Therefore we would like to gain deeper understanding through mathematical modeling of the problem. The collision of an electron with a molecule can be mathematically formulated in the language of partial integro-differential equations. The discretization converts the problem to a system of linear algebraic equations with a complex symmetric matrix. The matrix of this system is also sparse and for this reason we believe that the use of iterative methods to solve the resulting system is a suitable choice. However, as we were convinced when testing the convergence rate of the Krylov subspace methods for the model with two degrees of freedom (which we dealt with in the bachelor thesis), iterative methods suffer from slow convergence. This motivated us to try using preconditioning which is considered to be crucial for the reliability of iterative techniques across the literature. Our main goal in this work was to find a suitable preconditioning technique for Krylov subspace methods, which would ensure their faster convergence.

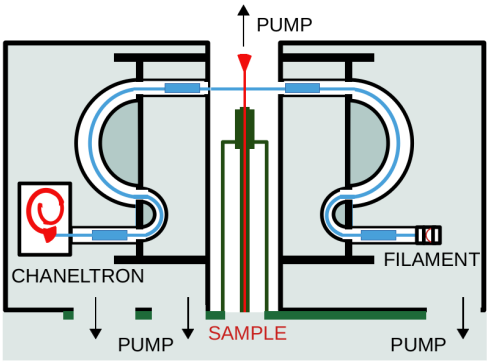
Keywords: Krylov subspace methods, preconditioning, Energy-loss spectrum

Contents

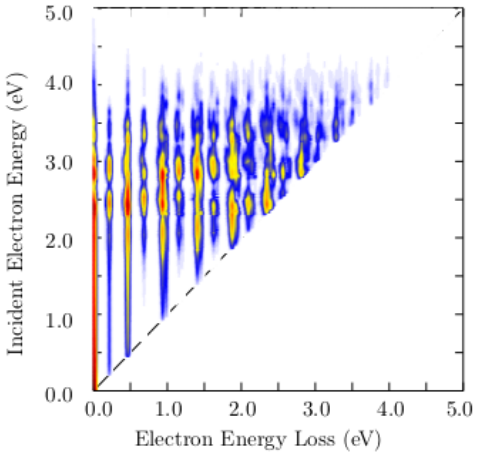
Introduction

The motivation for this work is the study of the process of a low-energy electron-molecule collisions and especially of the formation and the decay of resonances, i. e. short-lived negative ions. Such resonances occur both spontaneously, for instance in the atmosphere of Earth, but similar processes are also used in various technological fields, science and medicine. The electron-molecule collision phenomena play an important role in investigation of plasmas, upper athomophere and synthesis of ozone, biological systems, processes in comets etc.

We can get a good insight into the process of electron-molecule collision by analyzing the so-called 2D vibrational spectra (an example of such a spectrum is shown in the Figure 1b). It is a plot of integral cross section, which characterizes the probability of excitation of a molecule to a certain state, as a function of electron energy loss and incident electron energy. Such spectra can for example be measured using an experiment designed by Michael Allan, who built an optimized measuring apparatus (a simplified drawing of this device is shown in the Figure 1a). In fact, history of measurement of the energy-loss spectra con-



(a) A simplified diagram of spectrometer.



(b) Example of 2D electron energy-loss spectrum.

tains a significant Czechoslovak footprint. Jiří Schulz, who later emigrated from Czechoslovakia, discovered in the 1960s the presence of resonances in atomic and molecular processes that were known from nuclear physics until then. His student Michael Allan (also a Czech emigrant who worked at the Swiss University in Friborg) focused on improving the apparatus for measuring low-energy scattering cross section. After his retirement, he donated this measuring device to his student, who transferred it to the J. Heyrovský Institute of Physical Chemistry in Prague, where it is currently located.

The problem is that it can take weeks to accumulate enough data which we obtain as outputs of modern crossed-beam electron-molecule collision experiments. Moreover, the resulting spectra are very different for different molecules and we still cannot explain a lot of phenomena that appear in them. For this reason, we would like to model these processes mathematically. The collision of an electron

with a molecule can be mathematically formulated within the quantum scattering theory, which tells us how to calculate the cross sections needed to construct a 2D electron energy-loss spectrum. The task that ultimately needs to be solved is to find the *wave function* $\psi \in L_2\left(\mathbb{R}^d\right)$ that satisfies the integro-differential equation, which is derived from the stationary Schrödinger equation. We discretize this equation by expansion of the function ψ into a suitably chosen basis of spaces $L_2\left(\mathbb{R}^d\right)$. This converts the problem to a system of linear algebraic equations with a complex and symmetric and sparse matrix. The dimension of matrix and the number of its nonzero elements are strongly dependent on the number of degrees of freedom of the task. So far, the case has been well studied for one degree of freedom that can be efficiently (numerically) solved.

In the bachelor thesis Šarmanová [2020] we dealt with a model involving two degrees of freedom, which we took from Estrada et al. [1986]. To solve this system, we tested a total of six Krylov subspace methods, of which the COCG method proved to be the best. However, thanks to the special block structure of the matrix, it was possible to solve the problem by a specially designed direct method, which by its efficiency surpassed all iterative methods.

In this work we want to deal with a more complex and general model involving three vibrational degrees of freedom. For this reason, we have suggested three test models that capture various typical properties of the real molecules. As we know, the dimension of the system of linear equations that needs to be solved increases rapidly with the number of vibrational degrees of freedom. However, the matrix of this system is still sparse and in addition has a special block structure. Moreover, vector multiplication by this matrix can be effectively implemented without storing this matrix explicitly in memory. Thanks to this, we believe that the use of iterative methods to solve the resulting system is a suitable choice. However, the unpreconditioned iterative methods often suffer from slow convergence, which we have been convinced of when testing the convergence of the Krylov subspace methods for the model with two degrees of freedom. Although the tested iterative methods converged in most cases, they used a large number of iterations, which led to a long computational time. This phenomenon is so common that preconditioning has become a natural part of iterative solvers in most of the practical aplications. Unfortunately, in most cases, it is not clear how to construct an efficient preconditioning, because different approaches may help in different applications.

Our main goal is to find an efficient (iterative) approach for solving systems of linear algebraic equations arising from the models involving three degrees of freedom. Above all, however, we would like to focus on finding a suitable preconditioning technique for Krylov subspace methods, which would ensure their faster convergence.

Let us now briefly summarize the structure of the following chapters. We devote the first chapter to the motivation and formulation of the partial integro-differential equation that represents a mathematical model of electron-molecule scattering process. In the second chapter, we introduce the discretization of the problem and derive a system of linear algebraic equations. In the third chapter, we recall two basic approaches to solving systems of linear equations and the idea of preconditioning. Then we present a search of recently published literature that focus on specific preconditioning techniques. The fourth chapter focuses on the

description of test models as well as the specific properties of linear systems that we solve. The core of the whole work is the fifth chapter, which is devoted to numerical experiments.

1. Motivation

Let us have a neutral molecule M that consists of N atoms and has a total energy $E(\boldsymbol{\nu})$. Vector $\boldsymbol{\nu} \in \mathbb{N}_0^d$ in this expression indicates the quantum vibrational state of the molecule M. The index d stands for the number of vibrational degrees of freedom. Note that the positions of N atoms can generally be described using $3N$ position coordinates. Due to the fact that the displacement and the rotation of the overall system can be treated separately, we can only describe it using $3N-6$ vibrational degrees of freedom. We want to study a molecule that consists of three atoms and therefore $d = 3$ in our case. We deal with the electron-molecule collisions and especially with process of vibrational excitation

$$\mathrm{e}^- + \mathrm{M}(\boldsymbol{\nu}_\mathrm{i}) \to \mathrm{M}^- \to \mathrm{e}^- + \mathrm{M}(\boldsymbol{\nu}_\mathrm{f}). \tag{1.1}$$

We assume, that the electron e^- with an incident energy ε is captured by the molecule M to form a short-lived ion. Then the electron detaches from the ion, but it can leave with a different final energy. However, the total energy E of the system is conserved, i.e.

$$\varepsilon + E(\boldsymbol{\nu}_\mathrm{i}) = E = \varepsilon_\mathrm{f} + E(\boldsymbol{\nu}_\mathrm{f}). \tag{1.2}$$

Usual way to study microscopic systems in physics is to find the quantity called *scattering cross section*, in this case vibrational excitation cross section $\sigma_{\boldsymbol{\nu}_\mathrm{f}\leftarrow\boldsymbol{\nu}_\mathrm{i}}(\varepsilon)$. This quantity characterizes the probability of excitation of the molecule with initial energy given by a quantum number $\boldsymbol{\nu}_\mathrm{i}$ to the final state given by quantum number $\boldsymbol{\nu}_\mathrm{f}$ due to an electron-molecule collision. In our model we will always consider only one possible initial state of the molecule and that is the ground state $\boldsymbol{\nu}_\mathrm{i} = (0,0,0)$. We can calculate the scattering cross section using the following formula

$$\sigma_{\boldsymbol{\nu}_\mathrm{f}\leftarrow\boldsymbol{\nu}_\mathrm{i}}(\varepsilon) = \frac{(2\pi)^3}{4E}\left| f(\varepsilon_\mathrm{f}) \int\limits_{\mathbb{R}^2} \phi_{\boldsymbol{\nu}_\mathrm{f}}(\mathbf{q})\psi(\mathbf{q})\mathrm{d}\mathbf{q}\right|^2 . \tag{1.3}$$

We see, that $\sigma_{\boldsymbol{\nu}_\mathrm{f}\leftarrow\boldsymbol{\nu}_\mathrm{i}}(\varepsilon)$ depends on both the incident electron energy ε and the final energy ε_f electron has after the collision. Function $f(\varepsilon_\mathrm{f})$ represents quantum amplitude for detachment of e^- with energy ε_f. $\phi_{\boldsymbol{\nu}_\mathrm{f}}(\mathbf{q})$ is then a vibrational wave function of the target neutral molecule given by d-dimensional harmonic oscillator model. Importantly, in addition to the these known functions, formula 1.3 contains also the *wave function* $\psi(\mathbf{q})$ of the temporarily formed ion M^-.

Our goal is to construct a 2D energy-loss spectrum that is a plot of the cross section given by formula 1.3 as a function of incident electron energy ε (whose values we consider in the interval $(0;5)$) end energy-loss ΔE which is defined

$$\Delta E = \varepsilon - \varepsilon_\mathrm{f} = E(\boldsymbol{\nu}_\mathrm{i}) - E(\boldsymbol{\nu}_\mathrm{f}). \tag{1.4}$$

An example of such spectrum is in Figure 1.1. So all we have to do is to find the wave function $\psi : \mathbb{R}^3 \to \mathbb{C}$. We know from the quantum theory, that every wave function is a quadraticaly integrable complex function, which satisfies the Schrödinger equation. In our case it is a linear partial integro-differential equation of the form

$$\left[\widehat{E}\,(\varepsilon) - \left(\widehat{\mathcal{H}}_0 + \widehat{E}_\mathrm{d}\right) - \widehat{F}\,(\varepsilon)\right]\psi(\mathbf{q}) = \varphi(\mathbf{q}). \tag{1.5}$$

This equation parametrically depends on the electron energy ε. By solving this equation for a specific energy ε, we can obtain (using the formula 1.3) a section of the vibrational energy-loss spectrum (see the Figure 1.1). As a consequence,

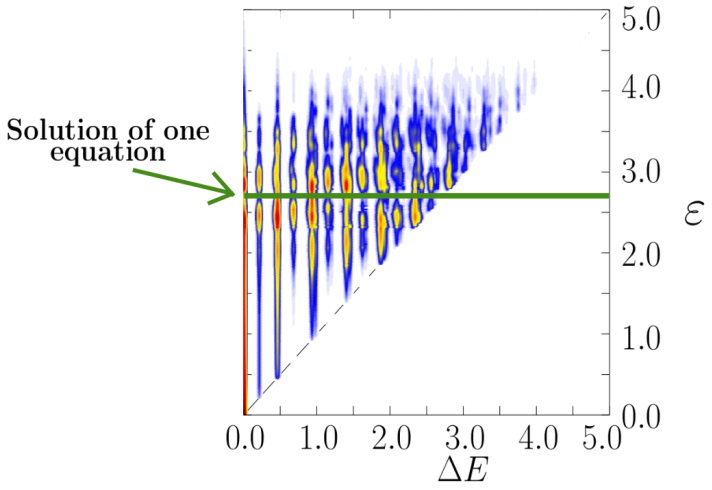


Figure 1.1: A 2D energy-loss spectrum.

we need to solve the equation 1.5 approximately 500 times for different $\varepsilon \in (0; 5)$ in order to be able to construct the 2D energy-loss spectrum. In the following section we will describe in more details the individual terms in equation 1.5.

1.1 Description of the equation

We want to solve a linear partial integro-differential equation 1.5 for the unknown wave function

$$\psi : \mathbb{R}^3 \to \mathbb{C}, \psi \in L^2(\mathbb{R}^3).$$

We will now explain the individual terms of this equation. First and foremost, operator $\widehat{E}$ simply mutiply ψ by energy. It depends on the electron energy ε and is defined as a multiplication by the initial energy of the system

$$\widehat{E}(\varepsilon)\,\psi(\mathbf{q}) = \left[\varepsilon + \underbrace{\frac{\omega_B}{2} + \frac{\omega_S}{2} + \frac{\omega_A}{2}}_{\text{neutral molecule energy}}\right]\psi(\mathbf{q}). \tag{1.6}$$

Parameters $\omega_B > 0$, $\omega_S > 0$ and $\omega_A > 0$ stand for neutral molecule oscillation frequencies in different vibrational degrees of freedom. Moreover, the energy $E_{\boldsymbol{\nu}_i} = \frac{1}{2}(\omega_B + \omega_S + \omega_A)$ represents a ground state energy of the three dimensional quantum harmonic oscilator that is described by the stationary Schrödinger equation

$$\widehat{\mathcal{H}_0}\phi_{\mathbf{n}}(\mathbf{q}) = E_{\mathbf{n}}\phi_{\mathbf{n}}(\mathbf{q}) \quad \mathbf{n} \in \mathbb{N}_0^3. \tag{1.7}$$

It means that $E_{\mathbf{0}}$ is the smallest eigenvalue of the Hamiltonian operator $\widehat{\mathcal{H}_0}$. This operator is the second term in equation 1.5 and is defined

$$\hat{\mathcal{H}}_0\psi(\mathbf{q}) = -\frac{\omega_B}{2}\frac{\partial^2\psi(\mathbf{q})}{\partial q_B^2} - \frac{\omega_S}{2}\frac{\partial^2\psi(\mathbf{q})}{\partial q_S^2} - \frac{\omega_A}{2}\frac{\partial^2\psi(\mathbf{q})}{\partial q_A^2} + \left(\frac{\omega_B}{2}q_B^2 + \frac{\omega_S}{2}q_S^2 + \frac{\omega_A}{2}q_A^2\right)\psi(\mathbf{q}). \tag{1.8}$$

It overall means that the motions of the nuclei of atoms in a neutral molecule are governed by the harmonic potential. The next term in our equation is called the electron afinity $\widehat{E}_{\mathrm{d}}$

$$\widehat{E}_{\mathrm{d}}\psi(\mathbf{q}) = \left(\mathbf{q}^{\mathrm{T}} \cdot \mathbb{M}\mathbf{q} + \boldsymbol{\lambda} \cdot \mathbf{q} + \epsilon_{\mathrm{d}}\right)\psi(\mathbf{q}), \tag{1.9}$$

where $\mathbb{M} \in \mathbb{R}^{3\times3}_{\mathrm{sym}}$, $\boldsymbol{\lambda} \in \mathbb{R}^3$ and $\epsilon_{\mathrm{d}} \in \mathbb{R}$. This term describes the potential of the molecular ion that is created by trapping an electron on a neutral molecule and we include it up to the second order in $\mathbf{q}$. Finally the 'level shift' operator $\widehat{F}$ is nonlocal and we can express it using an integral

$$\widehat{F}\psi(\mathbf{q}) = \int_{\mathbb{R}^3} \mathcal{F}(E, \mathbf{q}, \mathbf{q}')\psi(\mathbf{q}')\,\mathrm{d}\mathbf{q}', \tag{1.10}$$

with an integral kernel formed by the function $\mathcal{F}$. We want the ion in our model to be short-lived and the function $\mathcal{F}$ describes the possibility of electron detachment from the anion forming the neutral molecule again. In particular it holds

$$\mathcal{F}(E, \mathbf{q}, \mathbf{q}') = \sum_{\mathbf{n}} \phi_{\mathbf{n}}(\mathbf{q}) \left[\lim_{\xi\to0^+} \int_0^\infty \frac{g(\mathbf{q})\varepsilon^\alpha \mathrm{e}^{-\beta\varepsilon} g(\mathbf{q}')}{E - E_{\mathbf{n}} + i\xi - \epsilon}\mathrm{d}\epsilon\right] \phi_{\mathbf{n}}(\mathbf{q}'). \tag{1.11}$$

Function $\phi_{\mathbf{n}}(\mathbf{q})$ in this formula is an eigenfunction of the Hamiltonian operator $\widehat{\mathcal{H}_0}$ which means that it solves the equation 1.7 and describes the possible vibrational states of the resulting neutral molecule. Function $g(\mathbf{q})$ is approximated by a polynomial of the degree at most two, in our case we consider for simplicity $g(\mathbf{q}) = const.$, which is convinient to take $const. = \sqrt{a/(2\pi)}$. The right-hand side $\varphi(\mathbf{q})$ of the equation characterizes the possibility of capturing an electron on a molecule. It is defined in the following way

$$\varphi(\mathbf{q}) = V_{d\varepsilon}(\mathbf{q})\phi_{\mathbf{n}_\mathrm{i}}, \tag{1.12}$$

where

$$V_{d\varepsilon}(\mathbf{q}) = g(\mathbf{q})\varepsilon^{\alpha/2}\mathrm{e}^{-\beta\varepsilon/2}. \tag{1.13}$$

The form of this function in the models represents the known (see Domcke [1991]) limit of energy dependence for $\varepsilon \to 0$, exponential cut-off for $\varepsilon \to \infty$ and a possibility of dependence on coordinates $\mathbf{q}$. Note also that the electron capture and detachment are represented by the same amplitude. Integrand in 1.11 is therefore proportional to square of 1.13.

2. Discretization

Since we wish to solve the equation 1.5 numericaly, we need to define the relevant discretized problem. First and foremost, let us remind the assumption we have for the wave function. We know from quantum theory that every wave function is quadratically integrable, i.e. $\psi \in L^2(\mathbb{R}^3)$. Before we show how to derive a system of linear algebraic equaions that represents a discretization of the equation 1.5, we introduce notation for inner product, which we use throughout this chapter.

Definition 1. *Let us define an inner product as a map* $\langle\cdot|\cdot\rangle : L^2(\mathbb{R}^3)\times L^2(\mathbb{R}^3) \to \mathbb{C}$ *for functions* $\phi, \psi \in L_2(\mathbb{R}^3)$ *by the following formula:*

$$\langle\phi|\psi\rangle = \int_{\mathbb{R}^3} \overline{\phi(\boldsymbol{q})}\psi(\boldsymbol{q})\mathrm{d}\boldsymbol{q}. \tag{2.1}$$

It is a known fact (see for instance Hall [2013]) that $L^2(\mathbb{R}^3)$ together with the above defined scalar product forms a Hilbert space. Moreover, the functions $\{\phi_{\mathbf{n}}\}_{\mathbf{n}\in\mathbb{N}_0^3}$, where $\phi_{\mathbf{n}}$ are defined in equation 1.7, form an orthonormal basis of this space (which is shown in Formánek [2004]). In our notation it means that $\forall \mathbf{n}, \mathbf{n'} \in \mathbb{N}_0^3$ it holds $\langle\phi_{\mathbf{n}}|\phi_{\mathbf{n'}}\rangle = \delta_{\mathbf{n},\mathbf{n'}}$. We will use this basis formed by the eigenstates of the operator $\widehat{\mathcal{H}}_0$ to discretize our integro-differential equation by the spectral method. The reason is that this choice of the basis ensures a small filling of the matrix with complex numbers. In addition the different choice of discretization basis would lead to the need to calculate the operator inversion $\left[E - \widehat{\mathcal{H}}_0 + i\xi - \varepsilon\right]^{-1}$.

For every $\psi \in L^2(\mathbb{R}^3)$ there exist complex coefficients $\{\alpha_{\mathbf{n}}\}_{\mathbf{n}\in\mathbb{N}_0^3} \subset \mathbb{C}$ so that it holds

$$\psi = \sum_{\mathbf{n}\in\mathbb{N}_0^3} \alpha_{\mathbf{n}}\phi_{\mathbf{n}}. \tag{2.2}$$

Let $\widehat{A}$ be a linear map from $L^2(\mathbb{R}^3)$ to $L^2(\mathbb{R}^3)$. By $\langle\phi|\,\widehat{A}\,|\psi\rangle$ we understand the inner product $\langle\phi|\,\widehat{A}\,\psi\rangle$.

Let us now define an approximation $\Psi_{\mathbf{N}}$ of the wave function ψ. The index $\mathbf{N} = (N_B; N_S; N_A)$ determines the number of basis functions considered. In particular it means

$$\Psi_{\mathbf{N}} = \sum_{\mathbf{n}=(0;0;0)}^{(N_B;N_S;N_A)} \alpha_{\mathbf{n}}\phi_{\mathbf{n}}, \tag{2.3}$$

where $\alpha_{\mathbf{n}} \in \mathbb{C}$ are unknown coefficients. Let us first multiply our equation 1.5 by 'test function' $\overline{\phi_{\mathbf{n}}}$ where $\mathbf{n} \in \mathbb{N}_0^3$ and integrate it over $\mathbb{R}^3$. In the language of inner product we get

$$\left\langle \phi_{\mathbf{n}} \middle| \left[\widehat{E}\,(\varepsilon) - \left(\widehat{\mathcal{H}}_0 + \widehat{E}_{\mathrm{d}}\right) - \widehat{F}\,(\varepsilon)\right]\psi \right\rangle = \langle\phi_{\mathbf{n}}\,|V_{d\varepsilon}(\mathbf{q})\,\phi_{\mathbf{n}_\mathrm{i}}\rangle\,. \tag{2.4}$$

Let us now substitute the approximation $\Psi_{\mathbf{N}}$ to this equation

$$\left\langle \phi_{\mathbf{n}} \middle| \left[\widehat{E}\,(\varepsilon) - \left(\widehat{\mathcal{H}}_0 + \widehat{E}_{\mathrm{d}}\right) - \widehat{F}\,(\varepsilon)\right] \middle| \sum_{\mathbf{n'}=(0;0;0)}^{(N_B;N_S;N_A)} \alpha_{\mathbf{n'}}\phi_{\mathbf{n'}} \right\rangle = \langle\phi_{\mathbf{n}}\,|V_{d\varepsilon}(\mathbf{q})|\,\phi_{\mathbf{n}_\mathrm{i}}\rangle\,. \tag{2.5}$$

Using the linearity of the inner product we derive the system of $N_B \cdot N_S \cdot N_A$ linear algebraic equations for $\alpha_{\mathbf{n}}$

$$\sum_{\mathbf{n'}=(0;0;0)}^{(N_B;N_S;N_A)} \left\langle \phi_{\mathbf{n}} \left| \left[\widehat{E}\left(\varepsilon\right) - \left(\widehat{\mathcal{H}}_0 + \widehat{E}_{\mathrm{d}} \right) - \widehat{F}\left(\varepsilon\right) \right] \right| \phi_{\mathbf{n'}} \right\rangle \alpha_{\mathbf{n'}} = \left\langle \phi_{\mathbf{n}} \left| V_{d\varepsilon}(\mathbf{q}) \right| \phi_{\mathbf{n}_{\mathrm{i}}} \right\rangle . \quad (2.6)$$

This allows us to define elements of matrix $\mathbb{A}_{\mathcal{P}}\left(\varepsilon\right)$

$$\mathbb{A}_{\mathcal{P}}\left(\varepsilon\right)\left(\mathbf{n};\mathbf{n}'\right) \equiv \left\langle \phi_{\mathbf{n}} \left| \left[\widehat{E}\left(\varepsilon\right) - \left(\widehat{\mathcal{H}}_0 + \widehat{E}_{\mathrm{d}} \right) - \widehat{F}\left(\varepsilon\right) \right] \right| \phi_{\mathbf{n'}} \right\rangle \quad (2.7)$$

$$= \underbrace{\left\langle \phi_{\mathbf{n}} \left| \hat{E}_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \mathbb{E}_{\mathcal{P}}(\mathbf{n};\mathbf{n}')} - \underbrace{\left\langle \phi_{\mathbf{n}} \left| \left(\hat{\mathcal{H}}_0 + \hat{E}_d \right)_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \mathbb{H}_{\mathcal{P}}(\mathbf{n};\mathbf{n}')} - \underbrace{\left\langle \phi_{\mathbf{n}} \left| \hat{F}_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \mathbb{F}_{\mathcal{P}}(\mathbf{n};\mathbf{n}')}.$$

This matrix $\mathbb{A}_{\mathcal{P}}\left(\varepsilon\right)$ depends on the incident electron energy ε and a set of parameters $\mathcal{P} = \left\{ \overrightarrow{\omega}, \overrightarrow{\lambda}, \mathbb{M}, E_d, a, b, \alpha \right\}$ which is indicated by the subscript $\mathcal{P}$. Matrix $\mathbb{A}_{\mathcal{P}}\left(\varepsilon\right)$ is then the sum of three terms

$$\mathbb{A}_{\mathcal{P}}(\varepsilon) = \mathbb{E}_{\mathcal{P}}\left(\varepsilon\right) - \mathbb{H}_{\mathcal{P}} - \mathbb{F}_{\mathcal{P}}\left(\varepsilon\right). \quad (2.8)$$

In the following sections we derive elements of these three matrices.

2.1 Matrix $\mathbb{E}_{\mathcal{P}}(\varepsilon)$

Elements of the matrix $\mathbb{E}_{\mathcal{P}}(\varepsilon)$ are defined

$$\mathbb{E}_{\mathcal{P}}(\varepsilon)\left(\mathbf{n};\mathbf{n}'\right) = \left\langle \phi_{\mathbf{n}} \left| \hat{E}_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle = \left\langle \phi_{\mathbf{n}} \left| \varepsilon + \frac{\omega_B}{2} + \frac{\omega_S}{2} + \frac{\omega_A}{2} \right| \phi_{\mathbf{n}'} \right\rangle \quad (2.9)$$

$$= \left[\varepsilon + \frac{\omega_B}{2} + \frac{\omega_S}{2} + \frac{\omega_A}{2} \right] \delta_{n_B,n'_B}\, \delta_{n_S,n'_S}\, \delta_{n_A,n'_A}.$$

So overall we can write

$$\mathbb{E}_{\mathcal{P}}(\varepsilon) = \left[\varepsilon + \frac{\omega_B}{2} + \frac{\omega_S}{2} + \frac{\omega_A}{2} \right] \mathbb{I}_{N_B \cdot N_S \cdot N_A}. \quad (2.10)$$

$\mathbb{E}_{\mathcal{P}}(\varepsilon)$ is a real diagonal matrix with positive numbers on the main diagonal and therefore it is a positive definite matrix.

2.2 Matrix $\mathbb{H}_{\mathcal{P}}$

The matrix $\mathbb{H}_{\mathcal{P}}$ is created by discretization of the sum of operators $\left(\hat{\mathcal{H}}_0 + \hat{E}_{\mathrm{d}} \right)_{\mathcal{P}}$. Let us remind the definitions of these operators

$$\left(\hat{\mathcal{H}}_0 \right)_{\mathcal{P}} \psi(\mathbf{q}) = -\frac{\omega_B}{2} \frac{\partial^2 \psi(\mathbf{q})}{\partial q_B^2} - \frac{\omega_S}{2} \frac{\partial^2 \psi(\mathbf{q})}{\partial q_S^2} - \frac{\omega_A}{2} \frac{\partial^2 \psi(\mathbf{q})}{\partial q_A^2} \quad (2.11)$$

$$+ \left(\frac{\omega_B}{2} q_B^2 + \frac{\omega_S}{2} q_S^2 + \frac{\omega_A}{2} q_A^2 \right) \psi(\mathbf{q})$$

$$\left(\hat{E}_{\mathrm{d}} \right)_{\mathcal{P}} = \left(\mathbf{q}^T \mathbb{M} \mathbf{q} + \overrightarrow{\lambda}\, \mathbf{q} + \epsilon_{\mathrm{d}} \right) \psi(\mathbf{q}) \quad (2.12)$$

We define elements of matrix $\mathbb{H}_{\mathcal{P}}$

$$\mathbb{H}_{\mathcal{P}}\left(\mathbf{n};\mathbf{n}'\right) = \left\langle \phi_{\mathbf{n}} \left| \left(\hat{\mathcal{H}}_0 + \hat{E}_{\mathrm{d}}\right)_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle \tag{2.13}$$

$$= \left\langle \phi_{\mathbf{n}} \left| \left(\hat{\mathcal{H}}_0\right)_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle + \left\langle \phi_{\mathbf{n}} \left| \left(\mathbf{q}^T \mathbb{M} \mathbf{q} + \overrightarrow{\lambda}\mathbf{q} + \epsilon_{\mathrm{d}}\right)_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle$$

$$= \underbrace{\left\langle \phi_{\mathbf{n}} \left| \left(\hat{\mathcal{H}}_0\right)_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \mathbb{H}_{0,\mathcal{P}}(\mathbf{n};\mathbf{n}')} + \underbrace{\left\langle \phi_{\mathbf{n}} \left| \left(\lambda_B q_B + \lambda_s q_S + \lambda_A q_A\right) \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \Lambda_{\mathcal{P}}(\mathbf{n};\mathbf{n}')}$$

$$+ \underbrace{\left\langle \phi_{\mathbf{n}} \left| \left(\mathbb{M}_{BB} q_B^2 + \mathbb{M}_{SS} q_S^2 + \mathbb{M}_{AA} q_A^2\right) \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \Xi_{\mathcal{P}}(\mathbf{n};\mathbf{n}')}$$

$$+ \underbrace{\left\langle \phi_{\mathbf{n}} \left| \left(2\mathbb{M}_{BS} q_B q_S + 2\mathbb{M}_{BA} q_B q_A + 2\mathbb{M}_{SA} q_S q_A\right)_{\mathcal{P}} \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \Upsilon_{\mathcal{P}}(\mathbf{n};\mathbf{n}')} + \underbrace{\left\langle \phi_{\mathbf{n}} \left| \epsilon_{\mathrm{d}} \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \epsilon_{\mathrm{d}} \mathbb{I}}$$

So in the end we can write the matrix $\mathbb{H}_{\mathcal{P}}$ as the sum of several matrices

$$\mathbb{H}_{\mathcal{P}} = \mathbb{H}_{0,\mathcal{P}} + \Lambda_{\mathcal{P}} + \Xi_{\mathcal{P}} + \Upsilon_{\mathcal{P}} + \epsilon_{\mathrm{d}}\mathbb{I}, \tag{2.14}$$

the description of which is dealt with in the next sections.

2.2.1 Matrix $\mathbb{H}_{0,\mathcal{P}}$

Matrix $\mathbb{H}_{0,\mathcal{P}}$ is created by discretization of the operator $\left(\hat{\mathcal{H}}_0\right)_{\mathcal{P}}$, which represents the Hamiltonian operator for a three-dimensional linear harmonic oscillator. Therefore it holds

$$\hat{\mathcal{H}}_0 \left| \phi_{\mathbf{n}'} \right\rangle = E_{n'} \left| \phi_{\mathbf{n}'} \right\rangle, \tag{2.15}$$

since we have chosen the basis functions $\{\phi_{\mathbf{n}'}\}_{\mathbf{n}'\in\mathbb{N}}$ as the eigenfunctions of this operator $\left(\hat{\mathcal{H}}_0\right)_{\mathcal{P}}$. Hence the elements of matrix $\mathbb{H}_{0,\mathcal{P}}$ are defined

$$\mathbb{H}_{0,\mathcal{P}}\left(\mathbf{n};\mathbf{n}'\right) = \left\langle \phi_{\mathbf{n}} \left| \hat{\mathcal{H}}_0 \right| \phi_{\mathbf{n}'} \right\rangle = \left\langle \phi_{\mathbf{n}} \left| E_{n'} \right| \phi_{\mathbf{n}'} \right\rangle = E_{n'} \left\langle \phi_{\mathbf{n}} \mid \phi_{\mathbf{n}'} \right\rangle \tag{2.16}$$

$$= \left[\omega_B \left(n'_B + \frac{1}{2}\right) + \omega_S \left(n'_S + \frac{1}{2}\right) + \omega_A \left(n'_A + \frac{1}{2}\right)\right] \delta_{n_B,n'_B}\, \delta_{n_S,n'_S}\, \delta_{n_A,n'_A}.$$

$$= \left[\omega_B \left(n'_B + \frac{1}{2}\right) \delta_{n_B,n'_B}\right] \delta_{n_S,n'_S}\, \delta_{n_A,n'_A} + \delta_{n_B,n'_B} \left[\omega_S \left(n'_S + \frac{1}{2}\right) \delta_{n_S,n'_S}\right] \delta_{n_A,n'_A}$$

$$+ \delta_{n_B,n'_B}\, \delta_{n_S,n'_S} \left[\omega_A \left(n'_A + \frac{1}{2}\right) \delta_{n_A,n'_A}\right].$$

Let us define an auxiliary matrix

$$\mathbb{E}_N(n,n') = \left(n' + \frac{1}{2}\right)\delta_{n,n'} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1+\frac{1}{2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & (n'-1)+\frac{1}{2} \end{pmatrix}. \tag{2.17}$$

Matrix $\mathbb{H}_{0,\mathcal{P}}$ can be arranged in several ways. We will mark them with three letters, B, S and A, which indicate the order of the vibrational degrees of freedom. We can for example write

$$\mathbb{H}_{0,\mathcal{P}}(ASB) = (\omega_B \mathbb{E}_{N_B}) \otimes \mathbb{I}_{N_S} \otimes \mathbb{I}_{N_A} + \mathbb{I}_{N_B} \otimes (\omega_S \mathbb{E}_{N_S}) \otimes \mathbb{I}_{N_A} + \mathbb{I}_{N_B} \otimes \mathbb{I}_{N_S} \otimes (\omega_A \mathbb{E}_{N_A}) \tag{2.18}$$

or

$$\mathbb{H}_{0,\mathcal{P}}(ABS) = (\omega_S \mathbb{E}_{N_S}) \otimes \mathbb{I}_{N_B} \otimes \mathbb{I}_{N_A} + \mathbb{I}_{N_S} \otimes (\omega_B \mathbb{E}_{N_B}) \otimes \mathbb{I}_{N_A} + \mathbb{I}_{N_S} \otimes \mathbb{I}_{N_B} \otimes (\omega_A \mathbb{E}_{N_A}) . \tag{2.19}$$

It applies to all configurations of matrix $\mathbb{H}_{0,\mathcal{P}}$, that it is real diagonal matrix with positive elements on main diagonal and therefore it is a positive definite matrix.

2.2.2 Matrix $\Lambda_{\mathcal{P}}$

Let us move on to the description of the elements of the matrix $\Lambda_{\mathcal{P}}$. We divide this matrix even further into the sum of three matrices.

$$\Lambda_{\mathcal{P}}(\mathbf{n};\mathbf{n}') = \langle \phi_{\mathbf{n}} | \lambda_B q_B + \lambda_s q_S + \lambda_A q_A | \phi_{\mathbf{n}'} \rangle \tag{2.20}$$

$$= \underbrace{\lambda_B \langle \phi_{\mathbf{n}} | q_B | \phi_{\mathbf{n}'} \rangle}_{\equiv \Lambda_B(\mathbf{n};\mathbf{n}')} + \underbrace{\lambda_S \langle \phi_{\mathbf{n}} | q_S | \phi_{\mathbf{n}'} \rangle}_{\equiv \Lambda_S(\mathbf{n};\mathbf{n}')} + \underbrace{\lambda_A \langle \phi_{\mathbf{n}} | q_A | \phi_{\mathbf{n}'} \rangle}_{\equiv \Lambda_A(\mathbf{n};\mathbf{n}')}$$

We first derive the elements of the matrix Λ_B.

$$\Lambda_B(\mathbf{n};\mathbf{n}') = \lambda_B \langle \phi_{\mathbf{n}} | q_B | \phi_{\mathbf{n}'} \rangle = \lambda_B \int_{\mathbb{R}^3} \overline{\phi_{n_B} \phi_{n_S} \phi_{n_A}} q_B \phi_{n'_B} \phi_{n'_S} \phi_{n'_A} \mathrm{d}\mathbf{q} \tag{2.21}$$

$$= \lambda_B \cdot \left(\int_{\mathbb{R}} \overline{\phi_{n_B}} q_B \phi_{n'_B} \mathrm{d}q_B \right) \cdot \underbrace{\left(\int_{\mathbb{R}} \overline{\phi_{n_S}} \phi_{n'_S} \mathrm{d}q_S \right)}_{\delta_{n_S,n'_S}} \cdot \underbrace{\left(\int_{\mathbb{R}} \overline{\phi_{n_A}} \phi_{n'_A} \mathrm{d}q_A \right)}_{\delta_{n_A,n'_A}}$$

For the evaluation of the first integral, we use the fact that $\phi_{\mathbf{n}}$ are eigenstates of the operator $\hat{\mathcal{H}}_0$, i. e. eigenstates of the quantum harmonic oscilator. Using a standard computational procedure (see for instance Formánek [2004]) can be shown that it holds for these functions

$$\int_{\mathbb{R}} \overline{\phi_{n_B}} q_B \phi_{n'_B} \mathrm{d}q_B = \frac{1}{\sqrt{2}} \int_{\mathbb{R}} \overline{\phi_{n_B}} \sqrt{n'_B} \phi_{n'_B - 1} + \overline{\phi_{n_B}} \sqrt{n'_B + 1} \phi_{n'_B + 1} \mathrm{d}q_B. \tag{2.22}$$

Therefore we see that elements of matrix Λ_B are equal to

$$\Lambda_B(\mathbf{n};\mathbf{n}') = \lambda_B \cdot \left[\sqrt{\frac{n'_B}{2}} \delta_{n_B,n'_B-1} + \sqrt{\frac{n'_B+1}{2}} \delta_{n_B,n'_B+1} \right] \cdot \delta_{n_S,n'_S} \, \delta_{n_A,n'_A}. \tag{2.23}$$

Similarly we can derive $\Lambda_S(\mathbf{n};\mathbf{n}')$ and $\Lambda_A(\mathbf{n};\mathbf{n}')$

$$\Lambda_S(\mathbf{n};\mathbf{n}') = \delta_{n_B,n'_B} \cdot \lambda_S \left[\sqrt{\frac{n'_S}{2}} \delta_{n_S,n'_S-1} + \sqrt{\frac{n'_S+1}{2}} \delta_{n_S,n'_S+1} \right] \cdot \delta_{n_A,n'_A} \tag{2.24}$$

$$\Lambda_A(\mathbf{n};\mathbf{n}') = \delta_{n_B,n'_B} \cdot \delta_{n_S,n'_S} \cdot \lambda_A \cdot \left[\sqrt{\frac{n'_A}{2}} \delta_{n_A,n'_A-1} + \sqrt{\frac{n'_A+1}{2}} \delta_{n_A,n'_A+1} \right] \tag{2.25}$$

Let us define auxiliary matrix $\mathbb{Q}_N$ such that for $n, n' \in \{0 \cdots N-1\}$:

$$\mathbb{Q}_N(n;n') = \sqrt{\frac{n'}{2}} \delta_{n,n'-1} + \sqrt{\frac{n'+1}{2}} \delta_{n,n'+1}, \tag{2.26}$$

which means

$$\mathbb{Q}_N = \begin{pmatrix} 0 & \sqrt{\frac{1}{2}} & 0 & \cdots & 0 \\ \sqrt{\frac{1}{2}} & 0 & \sqrt{\frac{2}{2}} & \cdots & 0 \\ 0 & \sqrt{\frac{2}{2}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \sqrt{\frac{N-1}{2}} \\ 0 & 0 & 0 & \sqrt{\frac{N-1}{2}} & 0 \end{pmatrix}. \tag{2.27}$$

We can again arange matrices Λ_B, Λ_S and Λ_A in several ways based on different orders of vibration dimensions. We will give an example here

$$\Lambda_{\mathcal{P}}(ASB) = \underbrace{(\lambda_B \mathbb{Q}_{N_B}) \otimes \mathbb{I}_{N_S} \otimes \mathbb{I}_{N_A}}_{\Lambda_B(ASB)} + \underbrace{\mathbb{I}_{N_B} \otimes (\lambda_S \mathbb{Q}_{N_S}) \otimes \mathbb{I}_{N_A}}_{\Lambda_S(ASB)} + \underbrace{\mathbb{I}_{N_B} \otimes \mathbb{I}_{N_S} \otimes (\lambda_A \mathbb{Q}_{N_A})}_{\Lambda_A(ASB)} \tag{2.28}$$

or

$$\Lambda_{\mathcal{P}}(ABS) = \underbrace{(\lambda_S \mathbb{Q}_{N_S}) \otimes \mathbb{I}_{N_B} \otimes \mathbb{I}_{N_A}}_{\Lambda_B(ABS)} + \underbrace{\mathbb{I}_{N_S} \otimes (\lambda_B \mathbb{Q}_{N_B}) \otimes \mathbb{I}_{N_A}}_{\Lambda_S(ABS)} + \underbrace{\mathbb{I}_{N_S} \otimes \mathbb{I}_{N_B} \otimes (\lambda_A \mathbb{Q}_{N_A})}_{\Lambda_A(ABS)}. \tag{2.29}$$

Matrix $\mathbb{Q}_N$ is real and symmetric. Since it is also Jacobi matrix, we know, that it has N real and distinct eigenvalues. Moreover we can see, that $\det(\lambda\mathbb{I} - \mathbb{Q}_N)$ is an odd function for N odd and even function for N even. As a consequence, it holds $\lambda \in \mathrm{sp}(\mathbb{Q}_N) \implies -\lambda \in \mathrm{sp}(\mathbb{Q}_N)$ and therefore we can say, that $\mathbb{Q}_N$ is neither positive definite nor negative definite.

Matrix $\Lambda_B(ASB)$ is real, symmetric matrix defined

$$\Lambda_B(BSA) = \lambda_B \cdot \mathbb{I}_{N_A} \otimes \mathbb{I}_{N_S} \otimes \mathbb{Q}_{N_B}. \tag{2.30}$$

It has following structure

$$\Lambda_B(BSA) = \lambda_B \cdot \underbrace{\begin{pmatrix} \mathbb{Q}_{N_B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbb{Q}_{N_B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & \mathbb{Q}_{N_B} \end{pmatrix}}_{N_S \cdot N_A \text{ blocks}}. \tag{2.31}$$

Since spectrum of $\Lambda_B(BSA)$ is $\mathrm{sp}(\Lambda_B(BSA)) = \{\lambda_B\lambda \mid \lambda \in \mathrm{sp}(\mathbb{Q}_{N_B})\}$, we can say that $\Lambda_B(BSA)$ is neither positive definite nor negative definite.

Let us now focus on the properties of the second matrix, i. e. $\Lambda_S(BSA)$. We know that it holds

$$\Lambda_S(BSA) = \lambda_A \cdot \mathbb{I}_{N_A} \otimes [\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}] \tag{2.32}$$

Let us look at the properties of the matrix created by the Kronecker product of matrices $\mathbb{Q}_{N_S}$ and $\mathbb{I}_{N_B}$. This product creates a matrix that has the following

structure

$$(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}) = \underbrace{\begin{pmatrix} \mathbf{0} & \sqrt{\frac{1}{2}}\mathbb{I}_{N_B} & \mathbf{0} & \cdots & \mathbf{0} \\ \sqrt{\frac{1}{2}}\mathbb{I}_{N_B} & \mathbf{0} & \sqrt{\frac{2}{2}}\mathbb{I}_{N_B} & \cdots & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{2}{2}}\mathbb{I}_{N_B} & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \sqrt{\frac{N_S-1}{2}}\mathbb{I}_{N_B} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sqrt{\frac{N_S-1}{2}}\mathbb{I}_{N_B} & \mathbf{0} \end{pmatrix}}_{N_S \text{ blocks of size } N_B}. \quad (2.33)$$

Matrix $(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B})$ is indefinite, which we can easily prove from the properties of the spectrum. Let α is an eigenvalue of $\mathbb{Q}_{N_S}$ with corresponding eigenvector $\mathbf{a} = (a_1, a_2, \cdots, a_{N_S})^T$ and $\mathbf{b} \in \mathbb{C}^{N_B}$, then

$$(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B})(\mathbf{a} \otimes \mathbf{b}) = (\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}) \begin{pmatrix} a_1\mathbf{b} \\ a_2\mathbf{b} \\ \vdots \\ a_{N_S}\mathbf{b} \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{1}{2}}\mathbb{I}_{N_B} a_2\mathbf{b} \\ \sqrt{\frac{1}{2}}\mathbb{I}_{N_B} a_1\mathbf{b} + \sqrt{\frac{2}{2}}\mathbb{I}_{N_B} a_3\mathbf{b} \\ \vdots \\ \sqrt{\frac{N_S-1}{2}}\mathbb{I}_{N_B} a_{N_S-1}\mathbf{b} \end{pmatrix} \quad (2.34)$$

$$= \begin{pmatrix} \sqrt{\frac{1}{2}}a_2\mathbf{b} \\ \left(\sqrt{\frac{1}{2}}a_1 + \sqrt{\frac{2}{2}}a_3\right)\mathbf{b} \\ \vdots \\ \sqrt{\frac{N_S-1}{2}}a_{N_S-1}\mathbf{b} \end{pmatrix} = (\mathbb{Q}_{N_S}\mathbf{a}) \otimes \mathbf{b} = (\alpha\,\mathbf{a}) \otimes \mathbf{b} = \alpha\,(\mathbf{a} \otimes \mathbf{b}), \quad (2.35)$$

which means that α is also an eigenvalue of $(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B})$. Because it holds $\mathrm{sp}(\mathbb{Q}_{N_S}) \subset \mathrm{sp}(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B})$, we know that matrix $(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B})$ is indefinite.

Finally, matrix $\Lambda_S(BSA)$ has following structure

$$\Lambda_S(BSA) = \lambda_S \cdot \underbrace{\begin{pmatrix} (\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & (\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}) \end{pmatrix}}_{N_A \text{ blocks of size } N_S \cdot N_B} \quad (2.36)$$

Since spectrum of $\Lambda_S(BSA)$ is $\mathrm{sp}(\Lambda_S(BSA)) = \{\lambda_S\lambda \mid \lambda \in \mathrm{sp}(\mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B})\}$, we can say that $\Lambda_S(BSA)$ is neither positive definite nor negative definite.

We can proceed similarly for the matrix $\Lambda_A(BSA)$, which is defined

$$\Lambda_A(BSA) = \lambda_A \cdot \mathbb{Q}_{N_A} \otimes \mathbb{I}_{N_S} \otimes \mathbb{I}_{N_B}. \quad (2.37)$$

which means

$$\Lambda_A(BSA) = \lambda_A \cdot \underbrace{\begin{pmatrix} \mathbf{0} & \sqrt{\frac{1}{2}}\mathbb{I}_{N_B \cdot N_S} & \mathbf{0} & \cdots & \mathbf{0} \\ \sqrt{\frac{1}{2}}\mathbb{I}_{N_B \cdot N_S} & \mathbf{0} & \sqrt{\frac{2}{2}}\mathbb{I}_{N_B \cdot N_S} & \cdots & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{2}{2}}\mathbb{I}_{N_B \cdot N_S} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \sqrt{\frac{N_A-1}{2}}\mathbb{I}_{N_B \cdot N_S} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sqrt{\frac{N_A-1}{2}}\mathbb{I}_{N_B \cdot N_S} & \mathbf{0} \end{pmatrix}}_{N_A \text{ blocks of size } N_B \cdot N_S} \quad (2.38)$$

We see that $\Lambda_A(BSA) = \lambda_A \cdot \mathbb{Q}_{N_A} \otimes \mathbb{I}_{N_S \cdot N_B}$. It means that $\Lambda_A(BSA)$ is real, symmetric matrix and as we have already proven, it is neither positive definite nor negative definite. For other arrangements of matrices (different orders of Kronecker products) we would proceed in the same way when proving their properties.

2.2.3 Matrix $\Xi_{\mathcal{P}}$

Let us move on to the description of the elements of the matrix $\Xi_{\mathcal{P}}$. We divide this matrix even further into the sum of three matrices.

$$\Xi_{\mathcal{P}}\left(\mathbf{n};\mathbf{n}'\right) = \left\langle \phi_{\mathbf{n}} \left| \lambda_B q_B^2 + \lambda_s q_S^2 + \lambda_A q_A^2 \right| \phi_{\mathbf{n}'} \right\rangle \tag{2.39}$$

$$= \underbrace{\mathbb{M}_{BB} \left\langle \phi_{\mathbf{n}} \left| q_B^2 \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \Xi_B(\mathbf{n};\mathbf{n}')} + \underbrace{\mathbb{M}_{SS} \left\langle \phi_{\mathbf{n}} \left| q_S^2 \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \Xi_S(\mathbf{n};\mathbf{n}')} + \underbrace{\mathbb{M}_{AA} \left\langle \phi_{\mathbf{n}} \left| q_A^2 \right| \phi_{\mathbf{n}'} \right\rangle}_{\equiv \Xi_A(\mathbf{n};\mathbf{n}')}$$

We firstly derive the elements of the matrix Ξ_B.

$$\Xi_B\left(\mathbf{n};\mathbf{n}'\right) = \mathbb{M}_{BB} \left\langle \phi_{\mathbf{n}} \left| q_B^2 \right| \phi_{\mathbf{n}'} \right\rangle = \mathbb{M}_{BB} \int\limits_{\mathbb{R}^3} \overline{\phi_{n_B} \phi_{n_S} \phi_{n_A}} q_B^2 \phi_{n'_B} \phi_{n'_S} \phi_{n'_A} \mathrm{d}\mathbf{q} \tag{2.40}$$

$$= \mathbb{M}_{BB} \underbrace{\left(\int\limits_{\mathbb{R}} \overline{\phi_{n_A}} \phi_{n'_A} \mathrm{d}q_S \right)}_{\delta_{n_A,n'_A}} \cdot \underbrace{\left(\int\limits_{\mathbb{R}} \overline{\phi_{n_S}} \phi_{n'_S} \mathrm{d}q_S \right)}_{\delta_{n_S,n'_S}} \cdot \left(\int\limits_{\mathbb{R}} \overline{\phi_{n_B}} q_B^2 \phi_{n'_B} \mathrm{d}q_B \right)$$

Again using the standard procedure in quantum theory we can derive the value of the third integral and we get a formula for the elements of matrix Ξ_B

$$\Xi_B\left(\mathbf{n};\mathbf{n}'\right) = \mathbb{M}_{BB}\, \delta_{n_A,n'_A}\, \delta_{n_S,n'_S} \cdot \left[\sqrt{\frac{n'_B \cdot (n'_B - 1)}{4}} \delta_{n_B,n'_B-2} + \frac{2n'_B + 1}{2} \delta_{n_B,n'_B} + \right. \tag{2.41}$$

$$\left. + \sqrt{\frac{(n'_B + 1)\,(n'_B + 2)}{4}} \delta_{n_B,n'_B+2} \right].$$

Let us define another auxiliary matrix $\mathbb{S}_N$ such that for $n, n' \in \{0 \cdots N-1\}$:

$$\mathbb{S}_N(n;n') = \sqrt{\frac{n' \cdot (n'-1)}{4}} \delta_{n,n'-2} + \frac{2n'+1}{2} \delta_{n,n'} + \sqrt{\frac{(n'+1)\,(n'+2)}{4}} \delta_{n,n'+2}. \tag{2.42}$$

It has a five-diagonal structure as follows

$$\mathbb{S}_N = \begin{pmatrix} \frac{1}{2} & 0 & \sqrt{\frac{2}{4}} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \frac{3}{2} & 0 & \sqrt{\frac{6}{4}} & 0 & 0 & \cdots & 0 \\ \sqrt{\frac{2}{4}} & 0 & \frac{5}{2} & 0 & \sqrt{\frac{12}{4}} & 0 & \cdots & 0 \\ 0 & \sqrt{\frac{6}{4}} & 0 & \frac{7}{2} & 0 & \sqrt{\frac{20}{4}} & & 0 \\ 0 & 0 & \sqrt{\frac{12}{4}} & 0 & \frac{9}{2} & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sqrt{\frac{20}{4}} & 0 & \frac{11}{2} & \ddots & \sqrt{\frac{(N-2)(N-1)}{4}} \\ \vdots & \vdots & \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{\frac{(N-2)(N-1)}{4}} & 0 & \frac{2N-1}{2} \end{pmatrix}. \tag{2.43}$$

We see that

$$\mathbb{S}_N = \mathbb{Q}_N \mathbb{Q}_N + \frac{n}{2} \mathbf{e}_N \mathbf{e}_N^T, \tag{2.44}$$

and therefore we have for arbitrary $\mathbf{v} \in \mathbb{R}^N$

$$\mathbf{v}^T \mathbb{S}_N \mathbf{v} = \mathbf{v}^T \left(\mathbb{Q}_N \mathbb{Q}_N + \frac{N}{2} \mathbf{e}_N \mathbf{e}_N^T \right) \mathbf{v} = \mathbf{v}^T \mathbb{Q}_N^T \mathbb{Q}_N \mathbf{v} + \frac{N}{2} v_N^2 = \| \mathbb{Q}_N \mathbf{v} \|_2^2 + \frac{N}{2} v_N^2 \geq 0,$$

hence $\mathbb{S}_N$ is positive semidefinite. Matrix Ξ_B, in turn, can be defined in several ways that differ in the order of the vibrational dimensions. This time, let us show only one example

$$\Xi_B(BSA) = \mathbb{M}_{BB} \cdot \mathbb{I}_{N_A} \otimes \mathbb{I}_{N_S} \otimes \mathbb{S}_{N_B}. \tag{2.45}$$

Matrix $\Xi_B(BSA)$ has again a block diagonal structure

$$\Xi_B(BSA) = \mathbb{M}_{BB} \cdot \underbrace{\begin{pmatrix} \mathbb{S}_{N_B} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbb{S}_{N_B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{S}_{N_B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbb{S}_{N_B} \end{pmatrix}}_{N_S \cdot N_A \text{ blocks}}. \tag{2.46}$$

We can see, that matrix $\Xi_B(BSA)$ is real, symmetric and negative semidefinite (parameter $\mathbb{M}_{BB} \leq 0$) which is due to the properties of the Kronecker product.

Let us now focus on matrix $\Xi_S(BSA)$ defined

$$\Xi_S(BSA) = \mathbb{M}_{SS} \cdot \mathbb{I}_{N_A} \otimes \left[\mathbb{S}_{N_S} \otimes \mathbb{I}_{N_B} \right]. \tag{2.47}$$

It follows from the properties of Kronecker product of matrices, that matrices $\mathbb{S}_{N_S}$ and $(\mathbb{S}_{N_S} \otimes \mathbb{I}_{N_B})$ have the same set of eigenvalues.

$$\Xi_S(BSA) = \mathbb{M}_{SS} \cdot \underbrace{\begin{pmatrix} (\mathbb{S}_{N_S} \otimes \mathbb{I}_{N_B}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\mathbb{S}_{N_S} \otimes \mathbb{I}_{N_B}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \\ \mathbf{0} & \mathbf{0} & & (\mathbb{S}_{N_S} \otimes \mathbb{I}_{N_B}) \end{pmatrix}}_{N_A \text{ blocks}} \tag{2.48}$$

As a consequence, matrix $\Xi_S(BSA)$ is symmetric and negative semidefinite (because parameter $\mathbb{M}_{SS} \leq 0$).

Finally, matrix $\Xi_A(BSA)$ is defined

$$\Xi_A(BSA) = \mathbb{M}_{AA} \cdot \mathbb{S}_{N_A} \otimes \mathbb{I}_{N_S} \otimes \mathbb{I}_{N_B}. \tag{2.49}$$

It can be shown similarly as in previous cases, that matrix $\Xi_A(BSA)$ is symmetric negative semidefinite (parameter $\mathbb{M}_{AA} \leq 0$), because it has the same eigenvalues as matrix $\mathbb{S}_{N_A}$.

2.2.4 Matrix $\Upsilon_{\mathcal{P}}$

Let us move on to the next matrix.

$$\Upsilon_{\mathcal{P}}\left(\mathbf{n};\mathbf{n}'\right)=\left\langle\phi_{\mathbf{n}}\left|\mathbb{M}_{BS}q_B\cdot q_S+\mathbb{M}_{BA}q_B\cdot q_A+\mathbb{M}_{SA}q_S\cdot q_A\right|\phi_{\mathbf{n}'}\right\rangle \tag{2.50}$$

$$=\underbrace{\mathbb{M}_{BS}\left\langle\phi_{\mathbf{n}}\left|q_Bq_S\right|\phi_{\mathbf{n}'}\right\rangle}_{\equiv\Upsilon_{BS}(\mathbf{n};\mathbf{n}')}+\underbrace{\mathbb{M}_{BA}\left\langle\phi_{\mathbf{n}}\left|q_Bq_A\right|\phi_{\mathbf{n}'}\right\rangle}_{\equiv\Upsilon_{BA}(\mathbf{n};\mathbf{n}')}+\underbrace{\mathbb{M}_{SA}\left\langle\phi_{\mathbf{n}}\left|q_Sq_A\right|\phi_{\mathbf{n}'}\right\rangle}_{\Upsilon_{SA}(\mathbf{n};\mathbf{n}')}$$

First, we derive the elements of the matrix Υ_{BS}

$$\Upsilon_{BS}\left(\mathbf{n};\mathbf{n}'\right)=\mathbb{M}_{BS}\left\langle\phi_{\mathbf{n}}\left|q_B\cdot q_S\right|\phi_{\mathbf{n}'}\right\rangle=\mathbb{M}_{BS}\int\limits_{\mathbb{R}^3}\overline{\phi_{\mathbf{n}}}q_B\cdot q_S\phi_{\mathbf{n}'}\mathrm{d}\mathbf{q} \tag{2.51}$$

$$=\mathbb{M}_{BS}\int\limits_{\mathbb{R}^3}\overline{\phi_{n_B}\phi_{n_S}\phi_{n_A}}q_B\cdot q_S\phi_{n'_B}\phi_{n'_S}\phi_{n'_A}\mathrm{d}\mathbf{q}$$

$$=\mathbb{M}_{BS}\left(\int\limits_{\mathbb{R}}\overline{\phi_{n_A}}\phi_{n'_A}\mathrm{d}q_A\right)\cdot\left(\int\limits_{\mathbb{R}}\overline{\phi_{n_S}}q_S\phi_{n'_S}\mathrm{d}q_S\right)\cdot\left(\int\limits_{\mathbb{R}}\overline{\phi_{n_B}}q_B\phi_{n'_B}\mathrm{d}q_B\right)$$

$$=\mathbb{M}_{BS}\,\delta_{n_A,n'_A}\cdot\mathbb{Q}_{N_S}\left(n_S;n'_S\right)\cdot\mathbb{Q}_{N_B}\left(n_B;n'_B\right) \tag{2.52}$$

$$=\mathbb{M}_{BS}\,\delta_{n_A,n'_A}\cdot\left[\sqrt{\frac{n'_Sn'_B}{4}}\delta_{n_S,n'_S-1}\delta_{n_B,n'_B-1}+\sqrt{\frac{(n'_S+1)n'_B}{4}}\delta_{n_S,n'_S+1}\delta_{n_B,n'_B-1}\right. \tag{2.53}$$

$$\left.+\sqrt{\frac{n'_S(n'_B+1)}{4}}\delta_{n_S,n'_S-1}\delta_{n_B,n'_B+1}+\sqrt{\frac{(n'_S+1)(n'_B+1)}{4}}\delta_{n_S,n'_S+1}\delta_{n_B,n'_B+1}\right]. \tag{2.54}$$

Similarly, we can write for matrices Υ_{BA} and Υ_{SA}

$$\Upsilon_{BA}\left(\mathbf{n};\mathbf{n}'\right)=\mathbb{M}_{BA}\,\mathbb{Q}_{N_A}\left(n_A;n'_A\right)\delta_{n_S,n'_S}\,\mathbb{Q}_{N_B}\left(n_B;n'_B\right). \tag{2.55}$$

$$\Upsilon_{SA}\left(\mathbf{n};\mathbf{n}'\right)=\mathbb{M}_{SA}\,\mathbb{Q}_{N_A}\left(n_A;n'_A\right)\cdot\mathbb{Q}_{N_S}\left(n_S;n'_S\right)\cdot\delta_{n_B,n'_B}. \tag{2.56}$$

Before we write a definition of matrix Υ_{BS}, we will look at the properties of Kronecker product of matrices $\mathbb{Q}_{N_S}$ and $\mathbb{Q}_{N_B}$.

$$\mathbb{Q}_{N_S}\otimes\mathbb{Q}_{N_B}=\underbrace{\begin{pmatrix}\mathbf{0}&\sqrt{\frac{1}{2}}\mathbb{Q}_{N_B}&\mathbf{0}&\cdots&\mathbf{0}\\ \sqrt{\frac{1}{2}}\mathbb{Q}_{N_B}&\mathbf{0}&\sqrt{\frac{2}{2}}\mathbb{Q}_{N_B}&\cdots&\mathbf{0}\\ \mathbf{0}&\sqrt{\frac{2}{2}}\mathbb{Q}_{N_B}&\mathbf{0}&\cdots&\mathbf{0}\\ \vdots&\vdots&\vdots&\ddots&\sqrt{\frac{N_S-1}{2}}\mathbb{Q}_{N_B}\\ \mathbf{0}&\mathbf{0}&\mathbf{0}&\sqrt{\frac{N_S-1}{2}}\mathbb{Q}_{N_B}&\mathbf{0}\end{pmatrix}}_{N_S\text{blocks of size}N_B} \tag{2.57}$$

Let α be an eigenvalue of matrix $\mathbb{Q}_{N_S}$ with eigenvector $\mathbf{a} = (a_1, a_2, \cdots, a_{N_S})^T$ and β be eigenvalue of matrix $\mathbb{Q}_{N_B}$ with eigenvector $\mathbf{b} = (b_1, b_2, \cdots, b_{N_S})^T$, then it holds for spectrum of $\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B}$

$$(\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B})\, \mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} \sqrt{\frac{1}{2}}\mathbb{Q}_{N_B} a_2 \mathbf{b} \\ \sqrt{\frac{1}{2}}\mathbb{Q}_{N_B} a_1 \mathbf{b} + \sqrt{\frac{2}{2}}\mathbb{Q}_{N_B} a_3 \mathbf{b} \\ \vdots \\ \sqrt{\frac{N_S-1}{2}}\mathbb{Q}_{N_B} a_{N_S} \mathbf{b} \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{1}{2}}\beta a_2 \mathbf{b} \\ \sqrt{\frac{1}{2}}\beta a_1 \mathbf{b} + \sqrt{\frac{2}{2}}\beta a_3 \mathbf{b} \\ \vdots \\ \sqrt{\frac{N_S-1}{2}}\beta a_{N_S} \mathbf{b} \end{pmatrix} \tag{2.58}$$

$$= \beta \begin{pmatrix} \sqrt{\frac{1}{2}} a_2 \mathbf{b} \\ \left(\sqrt{\frac{1}{2}} a_1 + \sqrt{\frac{2}{2}} a_3\right) \mathbf{b} \\ \vdots \\ \sqrt{\frac{N_S-1}{2}} a_{N_S} \mathbf{b} \end{pmatrix} = \beta\, (\mathbb{Q}_{N_S} \mathbf{a}) \otimes \mathbf{b} = \beta\alpha\, (\mathbf{a} \otimes \mathbf{b})\,. \tag{2.59}$$

As a consequence, matrix $(\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B})$ is indefinite, because we know, that eigenvalues α and β of matrix $\mathbb{Q}_N$ are both positive and negative.

We can define matrix $\Upsilon_{BS}(BSA)$

$$\Upsilon_{BS}(BSA) = \mathbb{M}_{BS} \cdot \mathbb{I}_{N_A} \otimes (\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B})\,, \tag{2.60}$$

which has the following block diagonal structure

$$\Upsilon_{BS} = \mathbb{M}_{BS} \cdot \underbrace{\begin{pmatrix} (\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B}) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & (\mathbb{Q}_{N_S} \otimes \mathbb{Q}_{N_B}) \end{pmatrix}}_{N_A \text{ blocks of size } N_S \cdot N_B}. \tag{2.61}$$

Therefore we can say, that matrix $\Upsilon_{BS}(BSA)$ is symmetric and indefinite.

We can also define matrix $\Upsilon_{BA}(BSA)$

$$\Upsilon_{BA}(BSA) = \mathbb{M}_{BA}\, (\mathbb{Q}_{N_A} \otimes \mathbb{I}_{N_S}) \otimes \mathbb{Q}_{N_B}. \tag{2.62}$$

Since we already know properties of spectra of matrices $(\mathbb{Q}_{N_A} \otimes \mathbb{I}_{N_S})$ and $\mathbb{Q}_{N_B}$, we can say that matrix $\Upsilon_{BA}(BSA)$ is neither positive nor negative definite.

Similarly, we can define matrix $\Upsilon_{SA}(BSA)$

$$\Upsilon_{SA}(BSA) = \mathbb{M}_{SA} \cdot \mathbb{Q}_{N_A} \otimes \mathbb{Q}_{N_S} \otimes \mathbb{I}_{N_B}. \tag{2.63}$$

Matrix $\Upsilon_{SA}(BSA)$ is again a Kronecker product of matrices that are indefinite, so $\Upsilon_{SA}(BSA)$ is indefinite too.

2.2.5 Matrix $\mathbb{F}_{\mathcal{P}}(\varepsilon)$

Matrix $\mathbb{F}_{\mathcal{P}}$ is diagonal and it holds

$$\mathbb{F}_{\mathcal{P}}(\varepsilon)\, (\mathbf{n}; \mathbf{n}') = \left[\Delta_{\mathcal{P}}(E - E_{\boldsymbol{n}}) + \frac{i}{2}\Gamma_{\mathcal{P}}(E - E_{\boldsymbol{n}})\right] \delta_{n_A, n'_A} \delta_{n_S, n'_S} \delta_{n_B, n'_B}, \tag{2.64}$$

where E is the energy of the system

$$E = \varepsilon + E_{\mathbf{n}_i} \tag{2.65}$$

and $\mathbf{n}_\mathrm{i} = (0; 0; 0)$. It means that $E - E_{\boldsymbol{n}}$ can be replaced with

$$E - E_{\boldsymbol{n}} = \varepsilon + E_{\mathbf{n}_i} - E_{\mathbf{n}} = \varepsilon - \omega_B n_B - \omega_S n_S - \omega_A n_A. \tag{2.66}$$

Function $\Gamma_\mathcal{P}$ is defined

$$\Gamma_\mathcal{P}(x) = \begin{cases} ax^\alpha e^{-bx} & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases} \tag{2.67}$$

where a, b and α are parameters of model. Function $\Delta_\mathcal{P}$ is defined

$$\Delta_\mathcal{P}(x) = \mathrm{p.\,v.} \int_0^\infty \frac{f(\tau)^2}{x - \tau} \mathrm{d}\tau, \tag{2.68}$$

where

$$f(x) = \sqrt{\frac{\Gamma(x)}{2\pi}}. \tag{2.69}$$

We see, that it also holds $f(\varepsilon) = V_{d\varepsilon}$. Note that 2.68 can be calculated analytically or numerically if needed. Matrix $\mathbb{F}_\mathcal{P}$ is complex and diagonal. Therefore it is neither positive nor negative definite.

3. Linear system solvers

There are two basic approaches to solving systems of linear equations. The first group consists of so-called direct methods, most of which are based on decomposition of a matrix into the product of two matrices that can be easily inverted. Direct methods find an exact solution in a finite, known number of steps. However, another common feature of these methods is that until we complete the calculation, we have no idea about its solution. This can be a disadvantage, especially in various applications where we do not need an exact solution, but on the contrary we would like to obtain (if possible in as few steps as possible) its approximation.

The second large group of linear system solvers is formed by iterative methods. These are based on the construction of a sequence of solution approximations. At the same time, we hope that in as few steps as possible we achieve a sufficiently good approximation of the solution. This number of iterations is unknown in advance. In this work we deal with two iterative methods, both of which belong to the group of the Krylov subspace methods (the theory of Krylov subspace methods can be found, for example, in books Saad [2003] or Tebbens et al. [2012]).

In the bachelor thesis Šarmanová [2020] we have tested both Krylov subspace iterative methods and direct methods. The iterative methods often suffered from slow convergence which was a consequence of a large number of used steps. In this way, we became convinced of the need to use so-called preconditioning. In fact, the problem of a large number of iterations is unfortunatelly very common in iterative methods across applications and the use of preconditioning has become a natural part of iterative solvers.

3.1 Idea of preconditioning

A little vaguely defined, a preconditioning is any modification of a system (non-changing solution) that improves its properties in such a way that it can be more easily solved using iterative methods. Consider our linear algebraic system

$$\mathbb{A}_{\mathcal{P}}(\epsilon)\mathbf{x} = \mathbf{b}. \tag{3.1}$$

In this section we will use the shorter notation $\mathbb{A}_{\mathcal{P}}$ for matrix $\mathbb{A}_{\mathcal{P}}(\epsilon)$. We distinguish three basic ways (described in detail in Saad [2003]) of preconditioning.

1. *Left-preconditioned linear system*

$$\mathbb{K}^{-1}\mathbb{A}_{\mathcal{P}}\mathbf{x} = \mathbb{K}^{-1}\mathbf{b} \tag{3.2}$$

2. *Right-preconditioned linear system*

$$\mathbb{A}_{\mathcal{P}}\mathbb{K}^{-1}\mathbf{y} = \mathbf{b}, \qquad \mathbf{y} = \mathbb{K}\mathbf{x} \tag{3.3}$$

3. *Split preconditioning of the linear system*

$$\mathbb{M}_1^{-1}\mathbb{A}_{\mathcal{P}}\mathbb{M}_2^{-1}\mathbf{y} = \mathbb{M}_1^{-1}\mathbf{b}, \qquad \mathbf{y} = \mathbb{M}_2\mathbf{x}, \quad \mathbb{K} = \mathbb{M}_1\mathbb{M}_2 \tag{3.4}$$

The goal of preconditioning is to find suitable matrix $\mathbb{K}$ or matrices $\mathbb{M}_1$ and $\mathbb{M}_2$ so that the transformed system of equations has better properties in terms of convergence of iterative method. Usually we try to make $\mathbb{A}_{\mathcal{P}} \approx \mathbb{K} = \mathbb{M}_1\mathbb{M}_2$, where $\mathbb{M}_1$ and $\mathbb{M}_2$ are easily invertible.

The main problem is to find the matrices $\mathbb{M}_1$ and $\mathbb{M}_2$. Unfortunately, it is indeed not easy to determine what properties a preconditioned system should have so that iterative methods converge quickly. The rate of convergence is often associated with the spectral properties of the matrix. Note that by spectral information we mean the whole spectral decomposition of a matrix, not just a set of its eigenvalues. In the following text we will briefly describe two selected iterative methods - COCG and GMRES. In doing so, we will consider their preconditioned versions.

Before we focus on the individual methods, let us recall the basic idea (a detailed description of this topic can be found in Liesen and Strakoš [2013] or Saad [2003]) of the projection methods, to which the two method we have chosen belong. Let us have an arbitrary initial approximation $\mathbf{x}_0$ of the solution $\mathbf{x}$. A projection method is based on construction of a sequence of approximations $\mathbf{x}_k, k \in \{1, 2, \cdots\}$ so that

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{S}_k, \quad \mathbf{r}_k \perp \mathcal{C}_k. \tag{3.5}$$

Here $\mathbf{r}_k = A\mathbf{x}_k - \mathbf{b}$, where $\mathcal{S}_k$ is so called search space and $\mathcal{C}_k$ is denoted a constraint space. $\mathcal{S}_k$ and $\mathcal{C}_k$ are spaces of dimension k and by the specific choice of these spaces, we define the individual methods. Specifically, Krylov methods use Krylov subspaces to construct a sequence of solution approximations.

Definition 2. *Let $\mathbb{A} \in \mathbb{C}^{n\times n}$ and $\boldsymbol{v} \in \mathbb{C}^n$. The space generated by vectors $\boldsymbol{v}, \mathbb{A}\boldsymbol{v}, \mathbb{A}^2\boldsymbol{v}, \cdots, \mathbb{A}^{k-1}\boldsymbol{v}$ for $k \leq n$ is called k-th Krylov subspace $\mathcal{K}_k(\mathbb{A}, \boldsymbol{v})$, i. e.*

$$\mathcal{K}_k(\mathbb{A}, \boldsymbol{v}) = \operatorname{span}\left\{\boldsymbol{v}, \mathbb{A}\boldsymbol{v}, \cdots, \mathbb{A}^{k-1}\boldsymbol{v}\right\}.$$

Number $d \equiv d(\mathbb{A}, \boldsymbol{v})$ is called a grade of $\boldsymbol{v}$ with respect to $\mathbb{A}$ and denotes the maximum dimension of the Krylov subspace, i. e.

$$\mathcal{K}_1(\mathbb{A}, \boldsymbol{v}) \subset \mathcal{K}_2(\mathbb{A}, \boldsymbol{v}) \subset \cdots \subset \mathcal{K}_d(\mathbb{A}, \boldsymbol{v}) = \mathcal{K}_{d+1}(\mathbb{A}, \boldsymbol{v}) = \cdots = \mathcal{K}_n(\mathbb{A}, \boldsymbol{v}).$$

3.2 COCG method

The first method we deal with in this work is called *Conjugate orthogonal conjugate gradient method*. We have chosen this method because (in the bachelor thesis Šarmanová [2020]) it used to prove to be the most efficient for the given type of problem, which is related to the system we are now solving. The method was introduced by van der Vorst and Melissen [1990]. We have described the COCG method in detail in the bachelor thesis, so we only recall its main idea here.

The COCG method is created especially for complex symmetric matrices. As we know thanks to the Faber-Manteuffel theorem (the theorem together with other related results are discussed in details in Liesen and Strakoš [2013]), it is not possible for general matrices to define an algorithm which uses short-term

recurrences and is based on optimal Krylov subspace projection. The COCG method circumvents this principle by considering the so-called *conjugate product* instead of the standard scalar product. It is defined as follows.

Definition 3. *Let* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^n$. *The conjugate product* $[\cdot;\cdot] : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ *is defined by*

$$[\boldsymbol{a}; \boldsymbol{b}] = \langle \overline{\boldsymbol{a}} | \boldsymbol{b} \rangle, \tag{3.6}$$

where $\langle \cdot | \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ *stands for standard scalar product. We say that vectors* $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^n$ *are conjugate orthogonal, if*

$$[\boldsymbol{a}; \boldsymbol{b}] = 0.$$

The COCG method is a projection method determined by the choice of search space $\mathcal{S}_k$ and constrained space $\mathcal{C}_k$:

$$\mathcal{S}_k = \mathcal{C}_k = \mathcal{K}_k \left(\overline{\mathbb{A}}_{\mathcal{P}}; \overline{\mathbf{r}}_0 \right). \tag{3.7}$$

In order to generate the base of Krylov subspace using short recurrences, we consider the conjugate orthogonality of base vectors instead of classical orthogonality. This leads to an algorithm, which is an analogy of the conjugate gradient method, in which, however, we replace the standard scalar product with the conjugate product. Moreover, let $\mathbb{K}$ be a preconditioner, which can be written in form $\mathbb{K} = \mathbb{L}\mathbb{L}^T$. The following pseudocode shows how the COCG method works.

Conjugate orthogonal conjugate gradient method

We set $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
$\mathbf{p}_{-1} = 0, \beta_{-1} = 0$
$\mathbf{w}_0 = \mathbb{K}^{-1}\mathbf{r}_0$

for $k = 0, 1, 2, \cdots$:

1. $\mathbf{p}_k = \mathbf{w}_k + \beta_{k-1}\mathbf{p}_{k-1}$
2. $\alpha_k = \dfrac{[\mathbf{r}_k; \mathbf{w}_k]}{[A\mathbf{p}_k; \mathbf{p}_k]}$
3. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
4. $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$
5. $\mathbf{w}_{k+1} = \mathbb{K}^{-1}\mathbf{r}_{k+1}$
6. $\beta_k = \dfrac{[\mathbf{r}_{k+1}; \mathbf{w}_{k+1}]}{[\mathbf{r}_k; \mathbf{w}_k]}$

3.3 GMRES method

The generalized minimal residual method (GMRES) is an iterative method, which is used to solve generally non-symmetric systems of linear equations. It is a known method that was developed by Saad and Schultz [1986] and its analysis is described, for example, in Liesen and Strakoš [2013]. Its preconditioned version is described in detail for example in Saad [2003].

The GMRES method is a projection method determined by the choice of search space $\mathcal{S}_k$ and constrained space $\mathcal{C}_k$:

$$\mathcal{S}_k = \mathcal{K}_k \left(\mathbb{A}_{\mathcal{P}}; \mathbf{r}_0 \right), \quad \mathcal{C}_k = \mathbb{A}_{\mathcal{P}} \mathcal{K}_k \left(\mathbb{A}_{\mathcal{P}}; \mathbf{r}_0 \right), \tag{3.8}$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbb{A}_{\mathcal{P}}\mathbf{x}_0$ and $\mathbf{x}_0$ is an initial guess of the solution. The method is designed so that the approximation of the solution is chosen in each step so that the optimality property

$$\|\mathbf{r}_k\| = \min_{\mathbf{z}\in\mathbf{x}_0+\mathcal{K}_k(\mathbb{A}_{\mathcal{P}};\mathbf{r}_0)} \|\mathbf{b} - \mathbb{A}_{\mathcal{P}}\mathbf{z}\| \tag{3.9}$$

is satisfied. We have more options for GMRES preconditioning than with the COCG method, because it is not necessary for the preconditioned system to be symmetric. The following pseudocode shows the left-preconditioned GMRES method.

The Left-preconditined GMRES

We set $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbb{K}^{-1}(\mathbf{b} - A\mathbf{x}_0)$

$\mathbf{v}_0 = \mathbf{r}_0/\|\mathbf{r}_0\|$

Until we have a sufficiently accurate solution:

1. $\mathbf{w}_k = \mathbb{K}^{-1}A\mathbf{v}_{k-1}$
2. Using the Arnoldi algorithm, we find the vector $\mathbf{v}_k$ and update matrix H_k
3. We solve the minimization problem $\min_{y\in\mathbb{C}^k} \|\|\mathbf{r}_0\|\mathbf{e}_1 - H_k\mathbf{y}\|$
4. $\mathbf{x}_k = \mathbf{x}_0 + V_k\mathbf{y}$

The convergence analysis of the GMRES method has been studied in detail in Liesen and Strakoš [2013]. We will only mention two important results here. But before that, let us recall two important terms that we will use later.

Definition 4. *Let $\mathbb{A}_{\mathcal{P}} \in \mathbb{C}^{N\times N}$. We define a spectral norm of matrix $\mathbb{A}_{\mathcal{P}}$:*

$$\|\mathbb{A}_{\mathcal{P}}\|_2 = \sqrt{\lambda_{\max}\left(\mathbb{A}_{\mathcal{P}}^H\mathbb{A}_{\mathcal{P}}\right)},$$

where $\lambda_{\max}\left(\mathbb{A}_{\mathcal{P}}^H\mathbb{A}_{\mathcal{P}}\right)$ denotes the largest eigenvalue of matrix $\mathbb{A}_{\mathcal{P}}^H\mathbb{A}_{\mathcal{P}}$.

Definition 5. *Let $\mathbb{A}_{\mathcal{P}} \in \mathbb{C}^{N\times N}$ be a non-singular matrix. We define a condition number of matrix $\mathbb{A}_{\mathcal{P}}$:*

$$\kappa_2\left(\mathbb{A}_{\mathcal{P}}(\epsilon)\right) = \|\mathbb{A}_{\mathcal{P}}(\epsilon)\|_2 \left\|\mathbb{A}_{\mathcal{P}}^{-1}(\epsilon)\right\|_2.$$

The first convergence result that we take from Liesen and Strakoš [2013] gives us the GMRES (worst-case) convergence bound.

Theorem 1. *Consider a linear system $\mathbb{A}_{\mathcal{P}}\boldsymbol{x} = \boldsymbol{b}$ where $\mathbb{A}_{\mathcal{P}} \in \mathbb{C}^{N\times N}$ is non-singular and denote $\boldsymbol{r}_0 = \boldsymbol{b} - \mathbb{A}_{\mathcal{P}}\boldsymbol{x}_0$ an initial residual with grade d with respect to $\mathbb{A}_{\mathcal{P}}$. Let $\mathbb{A}_{\mathcal{P}} = \mathbb{X}\mathbb{J}\mathbb{X}^{-1}$, $\mathbb{J} = \mathrm{diag}(J_1, \cdots, J_m)$ be a Jordan decomposition of $\mathbb{A}_{\mathcal{P}}$. Then the Eukleidian residual norm in GMRES method satisfies*

$$\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{r}_0\|} \leq \kappa_2(\mathbb{X}) \min_{\substack{\phi(0)=1 \\ deg(\phi)\leq k}} \max_{1\leq i\leq m} \|\phi(J_i)\|.$$

Let us note, that if $\mathbb{A}_{\mathcal{P}}$ is normal (and therefore $\kappa_2(\mathbb{X}) = 1$), we can obtain an estimate of the convergence rate of GMRES using the eigenvalues. On the other hand, for general matrices $\mathbb{A}_{\mathcal{P}}$ for which $\kappa_2(\mathbb{X}) > 1$ this theorem usually does not give us much information about the actual behaviour of the method.

Finally, let us mention another important result to which we will refer many times in the work. The idea of the statement, which first appeared in Greenbaum et al. [1996], is that the set of eigenvalues alone does not tell us anything about the speed of convergence of GMRES. We take this theorem from Liesen and Strakoš [2013].

Theorem 2. *For any N positive numbers $f_0 \geq f_1 \geq \cdots \geq f_{N-1} > 0$ and any N nonzero complex numbers $\lambda_1, \cdots, \lambda_N$, not necesarily distinct, there exists a matrix $\mathbb{A}_{\mathcal{P}} \in \mathbb{C}^{N\times N}$ with eigenvalues $\lambda_1, \cdots, \lambda_N$ and a vector $\boldsymbol{b} \in \mathbb{C}^N$ with $\|\boldsymbol{b}\| = f_0$ so that GMRES applied to $\mathbb{A}_{\mathcal{P}}\boldsymbol{x} = \boldsymbol{b}$ with $\boldsymbol{x}_0 = 0$ has the residual norms $\|\boldsymbol{r}_n\| = f_n$, $n = 0, 1, \cdots, N-1$.*

We see, that if we want to describe the convergence of the GMRES method, it is indeed not enough to just look at the set of eigenvalues alone, we always need to know more information. However, as we will see later, this fact is very often ignored in the literature.

3.4 Preconditioning techniques

In the following text, we introduce some preconditioning techniques. Some of the methods are known and commonly used, while others have only recently been published and are designed for specific complex symmetric matrices. Let us start this section with a quote from the book by Saad [2003]

> '*Finding a good preconditioner to solve a given sparse linear system is often viewed as a combination of art and science. Theoretical results are rare and some methods work surprisingly well, often despite expectations.*'

We know that our matrix $\mathbb{A}_{\mathcal{P}}$ is banded and also symmetric. This leads us to the idea that we should choose a symmetric banded preconditioning matrix $\mathbb{K}$. Banded matrices form a rich source of preconditioners since they can approximate matrix $\mathbb{A}_{\mathcal{P}}$ well and at the same time it is quite cheap to decompose them. A special case is formed by diagonal matrices. They should be the first choice provided the matix $\mathbb{A}_{\mathcal{P}}$ is diagonaly dominant.

The problem is that even though we have a sparse matrix, the resulting factors can be dense. In case of banded matrix, the situation is a bit more optimistic since the fill-in elements can only appear within the nonzero band of the original matrix. This principle motivates us to take a different commonly used approach to creating preconditioners which is to perform an *incomplete factorization* (Saad [2003]). There are two basic ways to proceed. Firstly, we can define a pattern P of nonzero elements (it is usually composed of the positions of nonzero elements in matrix $\mathbb{A}_{\mathcal{P}}$) and subsequently, during the construction of the decomposition count only the elements that belong to this pattern. On the other hand, we can

define a level of fill l. This concept is based on idea that we will count only the 'significant' elements of decomposition that are larger than this given limit l. It means that dropping of fill-in elements depends on l.

3.4.1 Splitting preconditioners

Let us now consider a different approach of construction of preconditioner. We have our system of linear algebraic equations

$$\mathbb{A}_{\mathcal{P}}(\epsilon)\mathbf{x} = \mathbf{b}, \tag{3.10}$$

and it can be rewritten using real and imaginary part as follows

$$\left(\mathbb{B}_{\mathcal{P}}(\epsilon) + i\,\mathbb{D}_{\mathcal{P}}(\epsilon)\right)\left(\mathbf{y} + i\,\mathbf{z}\right) = \mathbf{b}, \tag{3.11}$$

where $\mathbb{B}_{\mathcal{P}}(\epsilon) = \mathrm{Re}\left(\mathbb{A}_{\mathcal{P}}(\epsilon)\right)$ and $\mathbb{D}_{\mathcal{P}}(\epsilon) = \mathrm{Im}\left(\mathbb{A}_{\mathcal{P}}(\epsilon)\right)$ are real, symmetric matrices, and $\mathbf{x} = \mathbf{y} + i\,\mathbf{z}$. In the following parts we will for clarity use $\mathbb{B}_{\mathcal{P}}$, resp. $\mathbb{D}_{\mathcal{P}}$, instead of $\mathbb{B}_{\mathcal{P}}(\epsilon)$, resp. $\mathbb{D}_{\mathcal{P}}(\epsilon)$. In the following we will mention a few preconditioning techniques based on this splitting of matrix $\mathbb{A}_{\mathcal{P}}$.

Let us start with a preconditioning matrix $\mathbb{R}_1(\alpha)$ for complex symmetric indefinite system introduced by Zhang and Dai [2015]. It is called the PSHNS preconditioner and is defined

$$\mathbb{R}_1(\alpha) = \frac{1}{2\alpha}\left(\alpha\mathbb{B}_{\mathcal{P}} + i\,\mathbb{I}\right)\left(\alpha\mathbb{D}_{\mathcal{P}} + \mathbb{I}\right), \tag{3.12}$$

where α is a positive number. Authors also mention, that if we assume that $\mathbb{B}_{\mathcal{P}}$ is indefinite matrix and $\mathbb{D}_{\mathcal{P}}$ is symmetric and positive definite, eigenvalues of the preconditioned matrix are clustered around the point 1 int the complex plane. They state, that this property is desirable, since it can lead to fast convergence rate of the preconditioned GMRES method. On the other hand, we already know from theorem 2 that the distribution of eigenvalues does not determine the speed of convergence of the GMRES method.

Another way to transform a set of linear equations is to use 'normal' equations, which we can obtain by multiplying our system by $\mathbb{B}_{\mathcal{P}}$. This approach was used by Wu [2015], who defined a preconditioning matrix

$$\mathbb{R}_2(\alpha) = \left(\alpha\mathbb{I} + i\,\mathbb{B}_{\mathcal{P}}\right)\left(\alpha\mathbb{D}_{\mathcal{P}} + \mathbb{B}_{\mathcal{P}}^2\right). \tag{3.13}$$

where $\mathbb{B}_{\mathcal{P}}$ is indefinite matrix and $\mathbb{D}_{\mathcal{P}}$ is symmetric and positive definite, for the transformed system of normal equations

$$\left(\mathbb{B}_{\mathcal{P}}^2 + i\,\mathbb{B}_{\mathcal{P}}\mathbb{D}_{\mathcal{P}}\right)\mathbf{x} = \mathbb{B}_{\mathcal{P}}\mathbf{b}. \tag{3.14}$$

Under these assumptions (if $\mathbb{B}_{\mathcal{P}}$ is also non-singular, which is not mentioned by the author, but is probably assumed by definition) $\mathbb{B}_{\mathcal{P}}^2$ is positive definite. It allows us to use the conjugate gradient method for solving this preconditioned system of linear equations. Authors also prove that the eigenvalues λ of the preconditioned matrix $\mathbb{R}_2^{-1}\mathbb{B}_{\mathcal{P}}\mathbb{A}_{\mathcal{P}}$ satisfy $|1 - |\lambda|| < 1$. They consider it desirable, that the number of distinct eigenvalues (or at least clusters) is small, because then the GMRES method teminates in small number of steps. However, this

connection between eigenvalues and the rate of convergence has been refuted (Carson and Strakoš [2020]), we cannot assess the convergence properties of the GMRES method solely on the basis of knowledge of eigenvalues.

Alternatively, we can multiply both sides of (3.10) by $i\,\mathbb{D}_{\mathcal{P}}$ and obtain

$$\left(i\,\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}} - \mathbb{D}_{\mathcal{P}}^2\right)\mathbf{x} = i\,\mathbb{D}_{\mathcal{P}}\mathbf{b}. \tag{3.15}$$

Pourbagher and Salkuyeh [2018] introduced a modification of the preconditioner defined in Wu [2015] for this system of 'normal' equations

$$\mathbb{R}_3(\alpha) = \frac{1}{2\alpha i}\left(\alpha\mathbb{I} + \mathbb{D}_{\mathcal{P}}\right)\left(i\,\alpha\mathbb{B}_{\mathcal{P}} - \mathbb{D}_{\mathcal{P}}^2\right). \tag{3.16}$$

However, this approach is unusable for us, because 3.15 is a singular system in our case. Let us better imagine that matrix $\mathbb{B}_{\mathcal{P}}$ in (3.10) can be written in form

$$\mathbb{B}_{\mathcal{P}} = -\mathbb{M} + \mathbb{K} \tag{3.17}$$

and matrices $\mathbb{M}, \mathbb{K}$ and $\mathbb{D}_{\mathcal{P}}$ are real, symmetric and positive definite. Let α be a positive number. Assuming these properties, Wu and Li [2017] introduced a preconditioning matrix for 3.10 defined

$$\mathbb{R}_4(\alpha) = \frac{1+i}{2\alpha}\left(\alpha\mathbb{I} + \mathbb{K}\right)\left(\alpha\mathbb{I} + \mathbb{D}_{\mathcal{P}} + i\mathbb{M}\right). \tag{3.18}$$

If the above mentioned assumptions are satisfied, then it can be shown, that all eigenvalues of the preconditioned matrix $\mathbb{R}_4^{-1}\mathbb{A}_{\mathcal{P}}$ statisfy $|1 - |\lambda|| < 1$. However, as we know, this feature does not necesarily guarantee an improvement in the convergence of iterative methods.

Finally, let $\mathbb{V} \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Cui et al. [2021] introduced a preconditioning matrix for system 3.10

$$\mathbb{R}_5(\mathbb{V}, \alpha) = \frac{1+i}{2\alpha}\left(\alpha\mathbb{V} + \mathbb{K}\right)\mathbb{V}^{-1}\left(\alpha\mathbb{V} + \mathbb{D}_{\mathcal{P}} + i\mathbb{M}\right). \tag{3.19}$$

In particular, if $\mathbb{V} = \mathbb{K}$, then

$$\mathbb{R}_5(\alpha) = \frac{(1+i)(\alpha+1)}{2\alpha}\left(\alpha\mathbb{K} + \mathbb{D}_{\mathcal{P}} + i\mathbb{M}\right). \tag{3.20}$$

Authors again prove that if matrices $\mathbb{M}, \mathbb{K}$ and $\mathbb{D}_{\mathcal{P}}$ are real, symmetric and positive definite, the eigenvalues of the preconditioned system are clustered around the point 1 in the complex plane. In our case, the problem is that the real and imaginary part of our matrix does not meet the requirements. First and foremost, we know, that our matrix $\mathbb{D}_{\mathcal{P}}$ is only positive semi-definite. Moreover, we are not able to prove that matrix $\mathbb{B}_{\mathcal{P}}$ can be written in the form 3.17.

Let us have our system of linear algebraic equations (3.10) with $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$ being symmetric matrices. Let us further assume that

$$-\mathbb{B}_{\mathcal{P}} \preceq \mathbb{D}_{\mathcal{P}} \prec \mathbb{B}_{\mathcal{P}}, \tag{3.21}$$

where the expression $A \preceq B$ means that the matrix $A - B$ is symmetric negative semidefinite and $A \prec B$ means that the matrix $A - B$ is symmetric negative

definite. Let $\mathbb{V} \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix and α be a positive number. Xu [2013] defined a preconditioner

$$\mathbb{R}_6(\mathbb{V};\alpha) = \frac{1}{2\alpha}(\alpha\mathbb{V} + \mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P})\,\mathbb{V}^{-1}(\alpha\mathbb{V} + \mathbb{B}_\mathcal{P} + \mathbb{D}_\mathcal{P}). \tag{3.22}$$

Moreover, if $\mathbb{V} = \mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P}$, the preconditioner becomes

$$\mathbb{R}_6(\alpha) = \frac{\alpha+1}{2\alpha}(\alpha(\mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P}) + \mathbb{B}_\mathcal{P} + \mathbb{D}_\mathcal{P}). \tag{3.23}$$

Authors metioned, that the eigenvalues of matrix $\mathbb{R}_6^{-1}(\alpha)\mathbb{A}_\mathcal{P}$ are clustered within the complex disc centered in 1 and with radius $\sqrt{\alpha^2+1}/(\alpha+1)$. Let us again note that the theorem 2 tells us, that the distribution of eigenvalues alone does not imply any information about the rate of convergence of GMRES method. Similar results can be derived for case

$$\mathbb{D}_\mathcal{P} \preceq \mathbb{B}_\mathcal{P} \prec -\mathbb{D}_\mathcal{P}$$

for which the author defined different preconditioner

$$\mathbb{R}_7(\mathbb{V};\alpha) = \frac{1}{2\alpha}(\alpha\mathbb{V} - \mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P})\,\mathbb{V}^{-1}(\alpha\mathbb{V} + \mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P}) \tag{3.24}$$

and assuming $\mathbb{V} = -\mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P}$, the preconditioner becomes

$$\mathbb{R}_7(\alpha) = \frac{\alpha+1}{2\alpha}(\alpha(-\mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P}) + \mathbb{B}_\mathcal{P} - \mathbb{D}_\mathcal{P}). \tag{3.25}$$

Eigenvalues of matrix $\mathbb{R}_7^{-1}(\alpha)\mathbb{A}_\mathcal{P}$ are again clustered within the complex disc centered in 1 and with radius $\sqrt{\alpha^2+1}/(\alpha+1)$. Although the distribution of eigenvalues of the preconditioned matrix may not mean anything for convergence of iterative methods, the authors were able to use the method for solving Helmholtz equation discretized by the finite element method.

3.4.2 Block two-by-two real systems

System of linear algebraic equations (3.10) can be written in real block two-by-two form

$$\begin{pmatrix} \mathbb{B}_\mathcal{P} & -\mathbb{D}_\mathcal{P} \\ \mathbb{D}_\mathcal{P} & \mathbb{B}_\mathcal{P} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{x} = \mathbf{y} + i\,\mathbf{z}. \tag{3.26}$$

Let us assume that matrices $\mathbb{B}_\mathcal{P}$ and $\mathbb{D}_\mathcal{P}$ are symmetric and at least one of them is non-singular. Assuming these properties Liao and Zhang [2017] constructed block multiplicative preconditioner for the system 3.26. First, the authors of this paper argued that if $\mathbb{B}_\mathcal{P}$ is non-singular we can write

$$\begin{bmatrix} \mathbb{B}_\mathcal{P} & -\mathbb{D}_\mathcal{P} \\ \mathbb{D}_\mathcal{P} & \mathbb{B}_\mathcal{P} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\alpha}\mathbb{B}_\mathcal{P} \end{bmatrix} \begin{bmatrix} \mathbb{I} & -\frac{1}{\alpha}\mathbb{D}_\mathcal{P} \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{B}_\mathcal{P} + \frac{1}{\alpha}\mathbb{D}_\mathcal{P}\mathbb{B}_\mathcal{P}^{-1}\mathbb{D}_\mathcal{P} & \mathbf{0} \\ \mathbf{0} & \alpha\mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \frac{1}{\alpha}\mathbb{B}_\mathcal{P}^{-1}\mathbb{D}_\mathcal{P} & \mathbb{I} \end{bmatrix}. \tag{3.27}$$

Although this equality is actually wrong, the authors used it to construct the preconditioner. They did it by removing some terms from this product in order

to simplify the cost of computation in each iteration. They defined the preconditioning matrix $\mathbb{P}_{BM}(\alpha)$

$$\mathbb{P}_{BM}(\alpha) = \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\alpha}\mathbb{B}_{\mathcal{P}} \end{bmatrix} \begin{bmatrix} \mathbb{I} & -\frac{1}{\alpha}\mathbb{D}_{\mathcal{P}} \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{B}_{\mathcal{P}} & \mathbf{0} \\ \mathbf{0} & \alpha\mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \frac{1}{\alpha}\mathbb{D}_{\mathcal{P}} & \mathbb{I} \end{bmatrix} = \begin{bmatrix} \mathbb{B}_{\mathcal{P}} - \frac{1}{\alpha}\mathbb{D}_{\mathcal{P}}^2 & -\mathbb{D}_{\mathcal{P}} \\ \frac{1}{\alpha}\mathbb{B}_{\mathcal{P}}\mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}} \end{bmatrix}. \tag{3.28}$$

System (3.10) can be alternatively written in real block two-by-two form

$$\begin{pmatrix} \mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}} \\ -\mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{b} \end{pmatrix}, \quad \mathbf{x} = \mathbf{y} + i\,\mathbf{z}. \tag{3.29}$$

which is advantageous if $\mathbb{D}_{\mathcal{P}}$ dominates over $\mathbb{B}_{\mathcal{P}}$. Liao and Zhang [2017] have also defined a preconditioner for the system 3.29

$$\mathbb{P}_{VBM}(\alpha) = \begin{bmatrix} \mathbb{D}_{\mathcal{P}} - \frac{1}{\alpha}\mathbb{B}_{\mathcal{P}}^2 & -\mathbb{B}_{\mathcal{P}} \\ -\frac{1}{\alpha}\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}} \end{bmatrix}. \tag{3.30}$$

However, we will not deal with this case, because our matrix does not meet the given assumption.

Let us focus on a different approach introduced by Liang and Zhang [2019]. Authors considered symmetric indefinite matrices $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$ satisfying

$$-\mathbb{D}_{\mathcal{P}} \prec \mathbb{B}_{\mathcal{P}} \preceq \mathbb{D}_{\mathcal{P}}, \tag{3.31}$$

where $\mathbb{D}_{\mathcal{P}} \preceq \mathbb{B}_{\mathcal{P}}$ means that $\mathbb{B}_{\mathcal{P}} - \mathbb{D}_{\mathcal{P}}$ is symmetric positive semidefinite and $-\mathbb{D}_{\mathcal{P}} \prec \mathbb{B}_{\mathcal{P}}$ means that $\mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}}$ is symmetric positive definite. We can transform the system of equations 3.26 as follows

$$\begin{pmatrix} \mathbb{I} & \mathbb{I} \\ -\mathbb{I} & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{B}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}} \\ \mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbb{I} & \mathbb{I} \\ -\mathbb{I} & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \tag{3.32}$$

which means that we get

$$\begin{pmatrix} \mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}} & -(\mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}}) \\ \mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}. \tag{3.33}$$

The reason is that we obtain a system of equations that has a positive definite matrix on the diagonal and a positive semidefinite matrix outside the diagonal. For this transformed system of linear equations we can use the following preconditioners

$$\mathbb{P}_{ABT-I}(\alpha) = \begin{pmatrix} \mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}} + \alpha\,(\mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}}) & \mathbf{0} \\ (1+\alpha^2)\,(\mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}}) & \mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}} + \alpha\,(\mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}}) \end{pmatrix} \tag{3.34}$$

and

$$\mathbb{P}_{CTR-I}(\alpha) = \begin{pmatrix} \mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}} & -(\mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}}) \\ \mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}} & \alpha^2\,(\mathbb{B}_{\mathcal{P}} + \mathbb{D}_{\mathcal{P}}) + 2\alpha\,(\mathbb{D}_{\mathcal{P}} - \mathbb{B}_{\mathcal{P}}) \end{pmatrix}. \tag{3.35}$$

Similarly, it is possible to define preconditioners in case $-\mathbb{B}_{\mathcal{P}} \prec \mathbb{D}_{\mathcal{P}} \preceq \mathbb{B}_{\mathcal{P}}$.

The following two methods were introduced in 2013. First of them, described in Bai et al. [2013a], requires an assumption, that both matrices $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$ from

equation (3.26) are positive semi-definite and at least one of them is positive definite. We know that for our matrices $\mathbb{B}_\mathcal{P}$ and $\mathbb{D}_\mathcal{P}$ these constraints are far too strong, but we can try using the transformation 3.33 and in that case, a milder condition 3.31 would suffice. With these assumptions, authors derive PMHSS method and also a preconditioner based on splitting

$$\mathbb{A}_N = \begin{pmatrix} \mathbb{B}_\mathcal{P} & -\mathbb{D}_\mathcal{P} \\ \mathbb{D}_\mathcal{P} & \mathbb{B}_\mathcal{P} \end{pmatrix} = \mathbb{P}_7(\alpha) - \mathbb{G}(\alpha), \tag{3.36}$$

where

$$\mathbb{P}_1(\alpha) = (\alpha + 1) \cdot \mathbb{P}(\alpha) \begin{pmatrix} \alpha\mathbb{B}_\mathcal{P} + \mathbb{D}_\mathcal{P} & \mathbf{0} \\ \mathbf{0} & \alpha\mathbb{B}_\mathcal{P} + \mathbb{D}_\mathcal{P} \end{pmatrix} \tag{3.37}$$

is so called preconditioning matrix and

$$\mathbb{P}(\alpha) = \frac{1}{2\alpha} \begin{pmatrix} \mathbb{I} & -\mathbb{I} \\ \mathbb{I} & \mathbb{I} \end{pmatrix}. \tag{3.38}$$

The second method introduced by Bai et al. [2013b] follows the one just mentioned. Authors consider system in form 3.26 with $\mathbb{B}_\mathcal{P}$ and $\mathbb{D}_\mathcal{P}$ positive semi-definite and at least one of them is positive definite and further assume

$$\text{null}\,(\mathbb{B}_\mathcal{P}) \cap \text{null}\,(\mathbb{D}_\mathcal{P}) = \{0\}. \tag{3.39}$$

The preconditioning matrix $\mathbb{P}_1(\alpha)$ derived in Bai et al. [2013a] is generaly non-symmetric and therefore it is mainly usable for nonsymmetric Krylov subspace methods such as GMRES. Therefore authors introduce simplified additive block diagonal preconditioning matrix

$$\mathbb{P}_2(\alpha) = \begin{pmatrix} \alpha\mathbb{B}_\mathcal{P} + \mathbb{D}_\mathcal{P} & \mathbf{0} \\ \mathbf{0} & \alpha\mathbb{B}_\mathcal{P} + \mathbb{D}_\mathcal{P} \end{pmatrix} \in \mathbb{R}^{2N \times 2N}, \tag{3.40}$$

which is symmetric and is obtained from $\mathbb{P}_2(\alpha)$ by omiting $\mathbb{P}(\alpha)$. It can be used to preconditon the symmetric matrix

$$\mathbb{A}_S = \begin{pmatrix} \mathbb{B}_\mathcal{P} & \mathbb{D}_\mathcal{P} \\ \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P} \end{pmatrix} \tag{3.41}$$

that is another alternative formulation of a system 3.10 of linear equations using a block two-by-two real linear system. Authors then for example show for $\alpha = 1$ that the eigenvalues of the preconditioned matrix $\mathbb{P}_2^{-1}(1)\mathbb{A}_S$ are clustered within the set $\left[-1; -\frac{\sqrt{2}}{2}\right] \cup \left[\frac{\sqrt{2}}{2}; 1\right]$, however, the eigenvalues of the preconditioned matrix $\mathbb{P}_1^{-1}(1)\mathbb{A}_N$ are clustered within the set $(0; 1] \times [-1; 1]$. Let us recall at this point that, due to the validity of theorem 2, this alone is not a sufficient argument for the speed of convergence of the GMRES method.

Let us move on to other two methods of preconditioning introduced and discussed by Liao and Zhang [2019]). First of them is a variant of the preconditioner described earlier in Bai et al. [2013b]. They assume a linear system of form (3.26) where $\mathbb{B}_\mathcal{P}$ is symmetric and positive definite and $\mathbb{D}_\mathcal{P}$ is symmetric positive semidefinite. Firstly, they take preconditioning matrix defined in the following way

$$\mathbb{P}_{GBD}(\alpha) = \begin{pmatrix} \mathbb{B}_\mathcal{P} + \alpha\mathbb{D}_\mathcal{P} & \mathbf{0} \\ \mathbf{0} & \mathbb{B}_\mathcal{P} + \alpha\mathbb{D}_\mathcal{P} \end{pmatrix}. \tag{3.42}$$

Preconditioned system is than

$$\mathbb{P}_{GBD}^{-1}(\alpha)\begin{pmatrix}\mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & -\mathbb{B}_{\mathcal{P}}\end{pmatrix}=\begin{pmatrix}\mathbb{H} & \frac{1}{\alpha}\left(\mathbb{I}-\mathbb{H}\right)\\ \frac{1}{\alpha}\left(\mathbb{I}-\mathbb{H}\right) & -\mathbb{H}\end{pmatrix},\quad \mathbb{H}=\left(\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}\right)^{-1}\mathbb{B}_{\mathcal{P}}. \tag{3.43}$$

Second preconditioning matrix they introduce is

$$\mathbb{P}_{BT}(\alpha)=\begin{pmatrix}\mathbb{B}_{\mathcal{P}} & \mathbf{0}\\ \mathbb{D}_{\mathcal{P}} & -\left(\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}\right)\mathbb{B}_{\mathcal{P}}^{-1}\left(\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}\right)\end{pmatrix} \tag{3.44}$$

Preconditioned system is than

$$\mathbb{P}_{BT}^{-1}(\alpha)\begin{pmatrix}\mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & -\mathbb{B}_{\mathcal{P}}\end{pmatrix}=\begin{pmatrix}\mathbb{I} & \mathbb{B}_{\mathcal{P}}^{-1}\mathbb{D}_{\mathcal{P}}\\ \mathbf{0} & -\mathbb{M}(\alpha)\end{pmatrix}, \tag{3.45}$$

where

$$\mathbb{M}(\alpha)=\left[\left(\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}\right)\mathbb{B}_{\mathcal{P}}^{-1}\left(\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}\right)\right]^{-1}\left(\mathbb{B}_{\mathcal{P}}+\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}}^{-1}\mathbb{D}_{\mathcal{P}}\right). \tag{3.46}$$

Authors then show that the sets of eigenalues of their preconditioned matrices are clustered and therefore they expect a good convergence of iterative methods. However as theorem 2 says, we cannot draw conclusions about the speed of convergence of GMRES from the distribution of eigenvalues alone.

Next preconditioning matrices were published by Liang and Zhang [2019]. Let us firstly assume a linear system of form (3.26) where $\mathbb{B}_{\mathcal{P}}$ is symmetric and positive definite and $\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}$ is nonsingular. Authors itroduce splitting

$$\begin{pmatrix}\mathbb{B}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}}\end{pmatrix}=\mathbb{P}_{ABT}(\alpha)-\mathbb{G}_{ABT}(\alpha), \tag{3.47}$$

where

$$\mathbb{P}_{ABT}(\alpha)=\begin{pmatrix}\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}} & \mathbf{0}\\ \left(1+\alpha^2\right)\mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}}+\alpha\mathbb{D}_{\mathcal{P}}\end{pmatrix},\quad \mathbb{G}_{ABT}(\alpha)=\begin{pmatrix}\alpha\mathbb{D}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}}\\ \alpha^2\mathbb{D}_{\mathcal{P}} & \alpha\mathbb{D}_{\mathcal{P}}\end{pmatrix}. \tag{3.48}$$

Subsequently they consider preconditioned system with matrix

$$\mathbb{P}_{ABT}^{-1}(\alpha)\begin{pmatrix}\mathbb{B}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}}\end{pmatrix} \tag{3.49}$$

and analyze its spectral properties. Further they assume, that both $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$ are symmetric positive semi-definite and in addition they satisfy

$$\operatorname{null}\left(\mathbb{B}_{\mathcal{P}}\right)\cap\operatorname{null}\left(\mathbb{D}_{\mathcal{P}}\right)=\{0\}.$$

Under these assumptions authors prove that all eigenvalues of preconditioned matrix 3.49 are located in interval $\left[\frac{1}{2};1\right]$.

Finally, let us assume that $\mathbb{B}_{\mathcal{P}}$ is positive definite matrix. Yuan and Zhang [2021] introduced a preconditioning technique with some $\alpha>0$:

$$\begin{bmatrix}\mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & -\mathbb{B}_{\mathcal{P}}\end{bmatrix}=\underbrace{\begin{bmatrix}\frac{1}{\alpha}I & \mathbf{0}\\ \left(\frac{1}{\alpha}+\alpha\right)\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}}^{-1} & -\alpha I\end{bmatrix}}_{\mathbb{P}_{NBT}(\alpha)}-\underbrace{\begin{bmatrix}\frac{1}{\alpha}I-\mathbb{B}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}}\\ \left(\frac{1}{\alpha}+\alpha\right)\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}}^{-1} & \mathbb{B}_{\mathcal{P}}-\alpha I\end{bmatrix}}_{\mathbb{R}_{NBT}(\alpha)}. \tag{3.50}$$

and the preconditioned matrix is in form

$$\mathbb{P}_{NBT}^{-1}\begin{bmatrix}\mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & -\mathbb{B}_{\mathcal{P}}\end{bmatrix}=\begin{bmatrix}\alpha\mathbb{B}_{\mathcal{P}} & \alpha\mathbb{D}_{\mathcal{P}}\\ \alpha\mathbb{D}_{\mathcal{P}} & \left(\frac{1}{\alpha}+\alpha\right)\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}}^{-1}+\frac{1}{\alpha}\mathbb{B}_{\mathcal{P}}\end{bmatrix}. \tag{3.51}$$

If $\mathbb{B}_{\mathcal{P}}$ is positive definite, this transformed system is also positive definite which is shown in article by construction of the block Cholesky factorization. Consequently, the preconditioned system can be solved by conjugate gradient method. Unfortunately in our case the assumption that $\mathbb{B}_{\mathcal{P}}$ is positive definite is too restrictive since we know that it does not hold for our matrix.

Let us now focus on preconditioners published by Benzi and Bertaccini [2008], who also considered a system in form 3.26. Firstly let $\mathbb{B}_{\mathcal{P}}$ 'dominate' $\mathbb{D}_{\mathcal{P}}$ (it means that for example that $\mathbb{D}_{\mathcal{P}}$ has small norm or rank in comparison to $\mathbb{B}_{\mathcal{P}}$). If this is the case, the authors recommend the use the following preconditioner

$$\mathbb{P}_3=\begin{pmatrix}\hat{\mathbb{B}}_{\mathcal{P}} & \mathbf{0}\\ \mathbf{0} & \hat{\mathbb{B}}_{\mathcal{P}}\end{pmatrix}, \tag{3.52}$$

where $\hat{\mathbb{B}}_{\mathcal{P}}$ is an approximation of $\mathbb{B}_{\mathcal{P}}$. If on the other hand $\mathbb{D}_{\mathcal{P}}$ is dominant, they offer two possible preconditioners, first of which is

$$\mathbb{P}_4(\alpha)=\begin{pmatrix}\alpha\mathbb{I} & -\hat{\mathbb{D}}_{\mathcal{P}}\\ \hat{\mathbb{D}}_{\mathcal{P}} & \alpha\mathbb{I}\end{pmatrix}, \tag{3.53}$$

with $\hat{\mathbb{D}}_{\mathcal{P}}\approx\mathbb{D}_{\mathcal{P}}$ and $\alpha>0$. These assumptions guarantee that $\mathbb{P}_4(\alpha)$ is invertible. Second mentioned preconditioning is

$$\mathbb{P}_5=\begin{pmatrix}\hat{\mathbb{B}}_{\mathcal{P}} & -\hat{\mathbb{D}}_{\mathcal{P}}\\ \hat{\mathbb{D}}_{\mathcal{P}} & \hat{\mathbb{B}}_{\mathcal{P}}\end{pmatrix}, \tag{3.54}$$

where $\hat{\mathbb{B}}_{\mathcal{P}}\approx\mathbb{B}_{\mathcal{P}}$, $\hat{\mathbb{D}}_{\mathcal{P}}\approx\mathbb{D}_{\mathcal{P}}$ and $\hat{\mathbb{B}}_{\mathcal{P}}+i\hat{\mathbb{D}}_{\mathcal{P}}$ is easily invertible. Another possible preconditioner the authors mention is

$$\mathbb{P}_6=\begin{pmatrix}\mathbb{B}_{\mathcal{P}}+\alpha\mathbb{I} & \mathbf{0}\\ \mathbf{0} & \mathbb{B}_{\mathcal{P}}+\alpha\mathbb{I}\end{pmatrix}\begin{pmatrix}\alpha\mathbb{I} & -\mathbb{D}_{\mathcal{P}}\\ \mathbb{D}_{\mathcal{P}} & \alpha\mathbb{I}\end{pmatrix}. \tag{3.55}$$

Subsequently they speak about block triangular preconditioners

$$\mathbb{P}_7=\begin{pmatrix}\hat{\mathbb{B}}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}}\\ \mathbf{0} & \hat{\mathbb{S}}\end{pmatrix}. \tag{3.56}$$

Block $\hat{\mathbb{S}}$ can be chosen in different ways, for instance $\hat{\mathbb{S}}=\mathbb{B}_{\mathcal{P}}+\mathbb{D}_{\mathcal{P}}\mathbb{B}_{\mathcal{P}}^{-1}\mathbb{D}_{\mathcal{P}}$. We will assume that both approximation $\hat{\mathbb{B}}_{\mathcal{P}}$ and matrix $\hat{\mathbb{S}}$ are invertible. The authors mention case when $\hat{\mathbb{S}}=\hat{\mathbb{A}}$, so we have

$$\mathbb{P}_8=\begin{pmatrix}\hat{\mathbb{B}}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}}\\ \mathbf{0} & \hat{\mathbb{B}}_{\mathcal{P}}\end{pmatrix}. \tag{3.57}$$

System of linear algebraic equations (3.10) can be equivalently written in real block two-by-two form

$$\begin{pmatrix}\mathbb{D}_{\mathcal{P}} & -\mathbb{B}_{\mathcal{P}}\\ \mathbb{B}_{\mathcal{P}} & \mathbb{D}_{\mathcal{P}}\end{pmatrix}\begin{pmatrix}\mathbf{z}\\ \mathbf{y}\end{pmatrix}=\begin{pmatrix}\mathbf{-b}\\ \mathbf{0}\end{pmatrix}. \tag{3.58}$$

Real and non-symmetric system of $2n \times 2n$ linear equations satisfying these properties is studied in many recentely published works.

Zhang and Dai [2017] introduced two preconditioners for the system (3.58)

$$\mathbb{P}_9(\alpha) = \frac{1}{\alpha}\begin{bmatrix} \alpha\mathbb{I} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} \end{bmatrix}\begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & 0 \\ 0 & \alpha\mathbb{I} \end{bmatrix} = \begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P}\left(\mathbb{I} + \frac{1}{\alpha}\mathbb{D}_\mathcal{P}\right) & \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} \end{bmatrix} \tag{3.59}$$

and

$$\mathbb{P}_{10}(\alpha) = \frac{1}{\alpha}\begin{bmatrix} \alpha\mathbb{I} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \mathbb{D}_\mathcal{P} \end{bmatrix}\begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & 0 \\ 0 & \alpha\mathbb{I} \end{bmatrix} = \begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P}\left(\mathbb{I} + \frac{1}{\alpha}\mathbb{D}_\mathcal{P}\right) & \mathbb{D}_\mathcal{P} \end{bmatrix} \tag{3.60}$$

and parameter α should balance the two parts and usually is assumed to be positive. Authors also claim that the preconditioner $\mathbb{P}_{10}(\alpha)$ is much closer to the original matrix of the system (3.58). Under additional assumptions that matrix $\mathbb{B}_\mathcal{P}$ is indefinite and matrix $\mathbb{D}_\mathcal{P}$ is symmetric and positive definite, they prove that the eigenvalues of the preconditioned system locate within the interval $(0; 1]$. Unfortunately, this result is not beneficial for us since we know, that distribution of eigenvalues do not determine the speed of convergence of iterative methods. Moreover, our matrices $\mathbb{B}_\mathcal{P}$ and $\mathbb{D}_\mathcal{P}$ do not satisfy the given assumptions. Let us move on to other methods. Zhang et al. [2018] defined a preconditioner

$$\mathbb{P}_{11}(\alpha) = \begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P}\left(\mathbb{I} + \frac{1}{\alpha}\mathbb{D}_\mathcal{P}\right) \\ \mathbb{B}_\mathcal{P} & \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} \end{bmatrix} \tag{3.61}$$

and its improved version

$$\mathbb{P}_{IB}(\alpha, \beta) = \begin{bmatrix} \mathbb{D}_\mathcal{P} & -\frac{1}{\alpha}\mathbb{B}_\mathcal{P}\left(\beta\mathbb{I} + \mathbb{D}_\mathcal{P}\right) \\ \mathbb{B}_\mathcal{P} & \beta\mathbb{I} + \mathbb{D}_\mathcal{P} \end{bmatrix}. \tag{3.62}$$

As in the previous case authors analyze the eigenvalue distribution of the preconditioned system under the assumptions that matrix $\mathbb{B}_\mathcal{P}$ is indefinite and matrix $\mathbb{D}_\mathcal{P}$ is symmetric and positive definite. They claim that the convergence behavior relates closely to the eigenvalue distribution of the preconditioned matrix and conclude, that 'the preconditioned matrix they introduced has well-clustered eigenvalue distribution'. Another preconditioning matrices are defined also in Shen and Shi [2018]. They further assume

$$\text{null}\left(\mathbb{B}_\mathcal{P}\right) \cap \text{null}\left(\mathbb{D}_\mathcal{P}\right) = \{0\} \tag{3.63}$$

and $i = \sqrt{-1}$ is not a generalized eigenvalue of the matrix pair $(\mathbb{B}_\mathcal{P}, \mathbb{D}_\mathcal{P})$. These assumptions ensure that the matrix $\mathbb{B}_\mathcal{P} + i\,\mathbb{D}_\mathcal{P}$ is non-singular. Authors then define first preconditioner

$$\mathbb{P}_{HSS} = \frac{1}{2\alpha}\begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & \mathbf{0} \\ \mathbf{0} & \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} \end{bmatrix}\begin{bmatrix} \alpha\mathbb{I} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \alpha\mathbb{I} \end{bmatrix}. \tag{3.64}$$

After that they however point out that preconditioner $\mathbb{P}_{10}(\alpha)$ published in Zhang and Dai [2017] is better. Finally they present an improved variant of *HSS* preconditioner which should work better for system (3.58)

$$\mathbb{P}_{VHSS} = \frac{1}{2\alpha}\begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & \mathbf{0} \\ \mathbf{0} & 2\alpha\mathbb{I} \end{bmatrix}\begin{bmatrix} \alpha\mathbb{I} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \mathbb{D}_\mathcal{P} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & -\frac{1}{\alpha}\left(\alpha\mathbb{I} + \mathbb{D}_\mathcal{P}\right)\mathbb{B}_\mathcal{P} \\ 2\mathbb{B}_\mathcal{P} & 2\mathbb{D}_\mathcal{P} \end{bmatrix}. \tag{3.65}$$

Fan et al. [2019] published a different preconditioning matrices for system (3.58) with more general properties (they assume that the matrix $\mathbb{D}_\mathcal{P}$ can be non-symmetric and positive definite). Firstly, they define the preconditioner

$$\mathbb{P}_{12}(\alpha) = \begin{bmatrix} \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P} - \frac{1}{\alpha}\mathbb{D}_\mathcal{P}\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \alpha\mathbb{I} + \mathbb{D}_\mathcal{P} - \frac{1}{\alpha}\mathbb{B}_\mathcal{P}^2 \end{bmatrix}. \tag{3.66}$$

After that they point out, that this matrix is not a good approximation of the system (3.58). Therefore they introduce an improved preconditioner

$$\mathbb{P}_{13}(\alpha, \beta) = \begin{bmatrix} \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P} - \frac{1}{\alpha}\mathbb{D}_\mathcal{P}\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \beta\mathbb{I} + \mathbb{D}_\mathcal{P} \end{bmatrix}. \tag{3.67}$$

Further, Axelsson et al. [2016] took a slightly different approach and considered a block two-by-two matrix in form

$$\begin{pmatrix} A & -bB_2 \\ aB_1 & A \end{pmatrix} \tag{3.68}$$

where A, B_i are square matrices and A is positive semidefinite. They define a preconditioning matrix in form

$$\mathbb{P} = \begin{pmatrix} A & -bB_2 \\ aB_1 & A + \sqrt{ab}(B_1 + B_2) \end{pmatrix}. \tag{3.69}$$

After that, they show that if $B_2 = B_1^T = 1$, $B + B^T$ is positive semidefinite and

$$\text{null}\,(\mathbb{B}_\mathcal{P}) \cap \text{null}\,(\mathbb{D}_\mathcal{P}) = \{0\},$$

then all eigenvalues of matrix

$$\mathbb{P}^{-1} \begin{pmatrix} A & -bB^T \\ aB & A \end{pmatrix} \tag{3.70}$$

are contained in interval $\left[\frac{1}{2}; 1\right]$. If we take $A = \mathbb{D}_\mathcal{P}$, $B_1 = B_2 = \mathbb{B}_\mathcal{P}$ and $a = b = 1$, we obtain a preconditioning matrix

$$\mathbb{P}_{14} = \begin{pmatrix} \mathbb{D}_\mathcal{P} & -\mathbb{B}_\mathcal{P} \\ \mathbb{B}_\mathcal{P} & \mathbb{D}_\mathcal{P} + 2 \cdot \mathbb{B}_\mathcal{P} \end{pmatrix}. \tag{3.71}$$

Finally, Zheng et al. [2021] invented another method for preconditioning of our system of linear algebraic equations (3.10) with $\mathbb{B}_\mathcal{P}$ being indefinite matrix and $\mathbb{D}_\mathcal{P}$ being symmetric and positive definite. Using the positive definiteness of $\mathbb{D}_\mathcal{P}$ the system (3.10) can be rewritten in the following way

$$(\mathbb{M} + i\sigma\mathbb{D}_\mathcal{P})\tilde{x} = \tilde{b}, \qquad \mathbb{M} = \mathbb{B}_\mathcal{P} + \epsilon\mathbb{D}_\mathcal{P}, \sigma = 1 + i\epsilon, \tag{3.72}$$

where ϵ is sufficiently large to ensure that $\mathbb{M}$ is symmetric possitive semi-definite. Then we can transform the equation to two-by-two system

$$\begin{bmatrix} \sigma\mathbb{D}_\mathcal{P} & -\mathbb{M} \\ \mathbb{M} & \sigma\mathbb{D}_\mathcal{P} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ i\tilde{x} \end{bmatrix} = \begin{bmatrix} -ib \\ b \end{bmatrix}. \tag{3.73}$$

Authors defined the following preconditioner

$$\mathbb{P}_{FS} = \begin{bmatrix} \sigma\mathbb{D}_\mathcal{P} & -\mathbb{M} \\ \mathbb{M} & \frac{|\sigma|}{\sigma}(|\sigma|\mathbb{D}_\mathcal{P} + 2\mathbb{M}) \end{bmatrix}, \qquad |\sigma| = \sqrt{1 + \epsilon^2}. \tag{3.74}$$

4. Test models

Let us have a system of linear algebraic equations

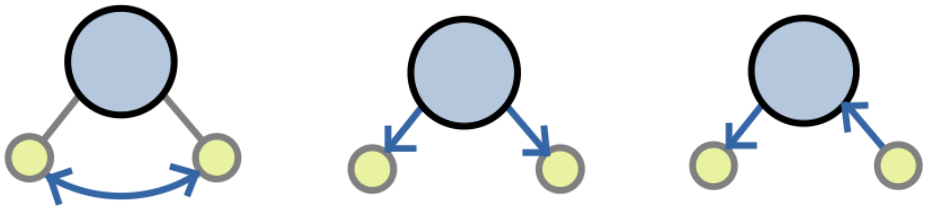
$$\mathbb{A}_{\mathcal{P}}(\varepsilon)\mathbf{x} = \mathbf{b}, \tag{4.1}$$

where $\mathbb{A}_{\mathcal{P}}(\varepsilon) \in \mathbb{C}^{N\times N}$ is symmetric matrix we derived in the previous chapter and $\mathbf{b} \in \mathbb{R}^N$ and $\mathbf{x} \in \mathbb{C}^N$ (let us remind that $N = N_B \cdot N_S \cdot N_A$). As we know, matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ can be written as a sum of three matrices

$$\mathbb{A}_{\mathcal{P}}(\varepsilon) = \mathbb{E}_{\mathcal{P}}(\varepsilon) - \mathbb{H}_{\mathcal{P}} - \mathbb{F}_{\mathcal{P}}(\varepsilon), \tag{4.2}$$

which depend on set of parameters $\mathcal{P} = \left\{\overrightarrow{\omega}, \overrightarrow{\lambda}, \mathbb{M}, E_d, a, b, \alpha\right\}$ and two of which also depend on incident electron energy ε. In the next section, we present three test models (i. e. particular sets $\mathcal{P}$) that we will later use for our numerical experiments. We will call the test models A, B and C. All of these models are (though relatively loosely) inspired by the vibrations of the water molecule and they include three vibrational degrees of freedom. The individual models then capture the various typical properties of the molecules.

In all the models we consider a molecule that performs three different vibrational movements. We call them bending motion, symmetric stretch and asymmetric stretch. Figure 4.1 illustrates how we can intuitively imagine the given movements. Each of these movements, which we will call vibrational modes, is



(a) Bending motion (b) Symmetric stretch (c) Asymmetric stretch

Figure 4.1: Illustration of different vibration modes

characterized by its frequency of oscillations. We call these frequencies ω_B, ω_S and ω_A and write $\overrightarrow{\omega} = (\omega_B, \omega_S, \omega_A)$. Note also that values of ω_B, ω_S and ω_A affect the size of the resulting matrix, i. e. the choice of N_B, N_S and N_A. These three numbers are chosen so that their ratio at least approximately corresponds to the ratio of the reciprocals of the parameters ω_B, ω_S and ω_A. Other parameters $\mathbb{M}$, $\overrightarrow{\lambda}$ and ϵ_{d} represent first three members of the Taylor expansion of the internal forces in the anion M^-. The parameter values are chosen in the range corresponding to the actual molecules and also respect the symmetry of the molecule. Finally, parameters a, b and α determine how metastable anion is formed by trapping an electron on the molecule or decays by electron detachment. In the following sections, we will give a few details about each model.

4.0.1 Model A

Let us first introduce model A. This model was designed to respect the fact that long-lived resonances may occur in molecules. We can imagine that it takes a long

time for an electron to detach from an anion M^-. This phenomenon is ensured mainly by parameters a, b and α. The table 4.1 contains a set of parameters that define model A. Figure 4.2 shows the electron energy-loss spectrum representing

$a = 1.00$	$\omega_B = 0.22$	$\lambda_B = 0.05$	M_{BB} = -0.05	$M_{BS} = 0.03$
$b = 1.00$	$\omega_S = 0.47$	$\lambda_S = 0.15$	M_{SS} = -0.10	$M_{BA} = 0.00$
$\alpha = 0.20$	$\omega_A = 0.49$	$\lambda_A = 0.00$	M_{AA} = -0.20	$M_{SA} = 0.00$

Table 4.1: Parameters of model A. Note that $\mathbb{M}$ is a symmetric matrix, so $M_{SB} = M_{BS}$, $M_{AS} = M_{SA}$ and $M_{AB} = M_{BA}$. Size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model A is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$).

model A. Figure 4.3 illustrates structure of matrix $\mathbb{A}_{\mathcal{P}}$ for model A and energy

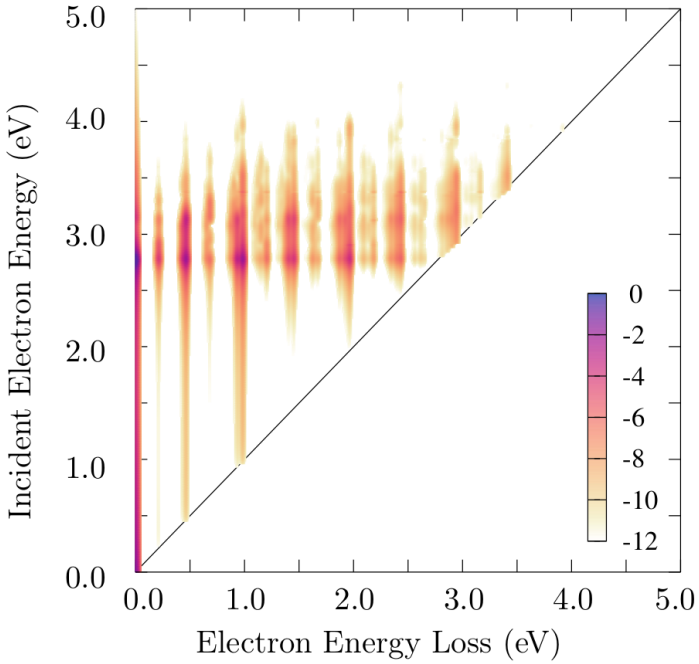


Figure 4.2: Energy-loss spectrum representing model A, i. e. plot of the integral cross section $\sigma_{\boldsymbol{\nu}_\mathrm{f} \leftarrow \boldsymbol{\nu}_\mathrm{i}}(\varepsilon)$ as a function of the incident electron energy ε and the electron energy loss. The integral cross section is ploted in a logaritmic scale.

$\varepsilon = 2.5$, which is in the middle of the interval of energies we consider. Six different parts of the Figure 4.3 represent the natural reorderings of matrix $\mathbb{A}_{\mathcal{P}}$. These correspond to the different orders of the Kronecker products by which the matrix $\mathbb{A}_{\mathcal{P}}$ is formed. As we can see, magnitudes of diagonal elements in matrix $\mathbb{A}_{\mathcal{P}}$ are in most cases much bigger than other elements in the relevant rows.

In the Figure 4.4 we drew in a complex plane approximated eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$. For finding these eigenvalues we used LAPACK subroutine 'zgeev', which computes the eigenvalues and, optionally, the left and right eigenvectors for general complex matrices. For better idea, we also depicted unit circles in complex plane.

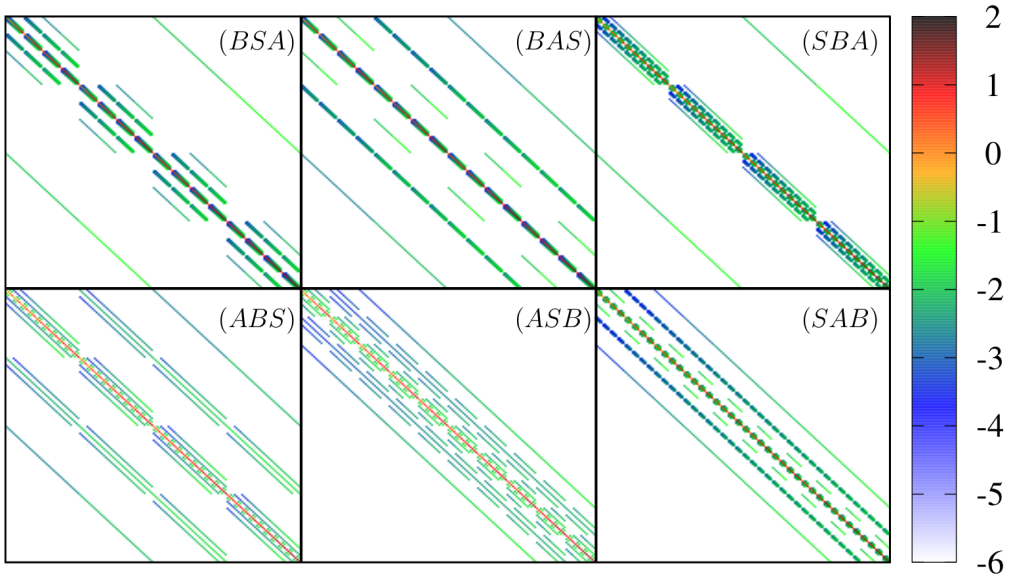


Figure 4.3: Structure of matrix $\mathbb{A}_{\mathcal{P}}$ for model A and selected energy $\varepsilon = 2.5$ and different orders of the Kronecker products. Ploted values of nonzero elements in matrices are logarithms of magnitudes of elements in matrix $\mathbb{A}_{\mathcal{P}}$. For better visibility of the matrix structure, the size of the basis is lower here, i. e. $N_B = 10$, $N_S = 4$, $N_A = 4$.

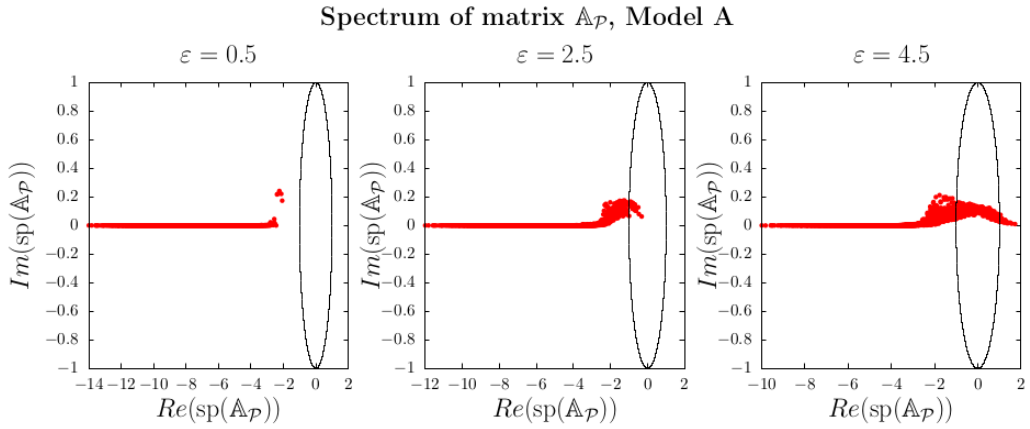


Figure 4.4: Eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$ are marked in red color. For approximating eigenvalues we used LAPACK routine 'zgeev'. Black curve represents unit circle in complex plane. Remind that the size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model A is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$).

4.0.2 Model B

We named the second model B. In contrast to model A, this model was designed to describe short-lived resonances. This means that the anion M^- formed by trapping an electron e^- on the molecule immediately decays. The table 4.2 contains a set of parameters that define model B. The fast decay of anion causes the forces $\hat{E}_{\mathrm{d}}$ to act for a short time and there is significantly lower energy loss. Figure 4.5 shows the electron energy-loss spectrum representing model B.

$a = 4.105$	$\omega_B = 0.22$	$\lambda_B = 0.09$	M_{BB} = -0.03	$M_{BS} = 0.03$
$b = 0.15$	$\omega_S = 0.47$	$\lambda_S = 0.50$	M_{SS} = -0.10	$M_{BA} = 0.00$
$\alpha = 0.20$	$\omega_A = 0.49$	$\lambda_A = 0.00$	M_{AA} = -0.20	$M_{SA} = 0.00$

Table 4.2: Parameters of model B. Note that $\mathbb{M}$ is a symmetric matrix, so $M_{SB} = M_{BS}$, $M_{AS} = M_{SA}$ and $M_{AB} = M_{BA}$. Size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model B is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$).

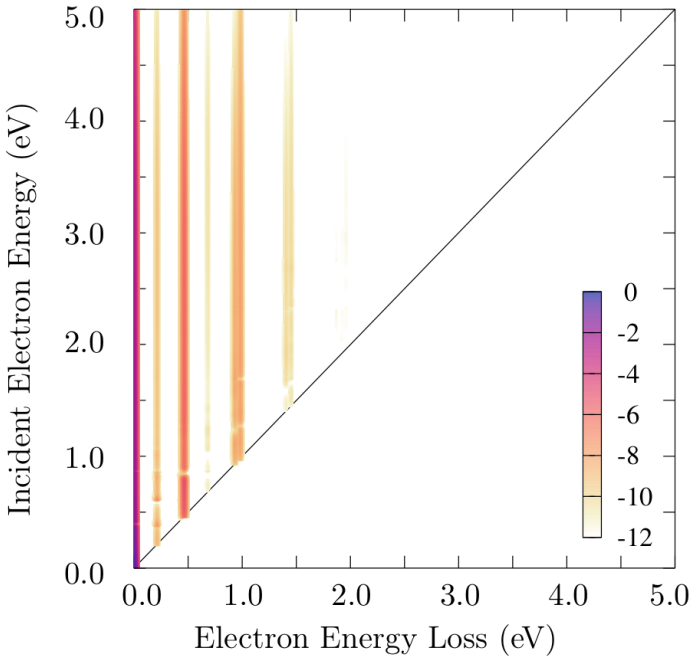


Figure 4.5: Energy-loss spectrum representing model B, i.e. plot of the integral cross section $\sigma_{\nu_f \leftarrow \nu_i}(\varepsilon)$ as a function of the incident electron energy ε and the electron energy loss. The integral cross section is ploted in a logaritmic scale.

Figure 4.6 illustrates structure of matrix $\mathbb{A}_{\mathcal{P}}$ for model B and energy $\varepsilon = 2.5$. As in the previous case, six different parts of the Figure 4.6 represent the natural reorderings of matrix $\mathbb{A}_{\mathcal{P}}$. These reorderings can also be understood as a consequence of different permutations of vibrational degrees of freedom of our model. In the Figure 4.7 we have again ploted eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$.

4.0.3 Model C

Finally, let us focus on model C, which differs significantly from the previous ones. This model is designed so that one vibration mode is much more pronounced than the others. The remaining vibrational degrees of freedom are only weakly coupled by quadratic terms M_{BS}, M_{BA} causing energy dissipation. Regarding the parameters a, b and α, which determine whether the resonance will be short-lived or long-lived, we leave them the same as in model B. The table 4.3 contains a set of parameters that define model C.

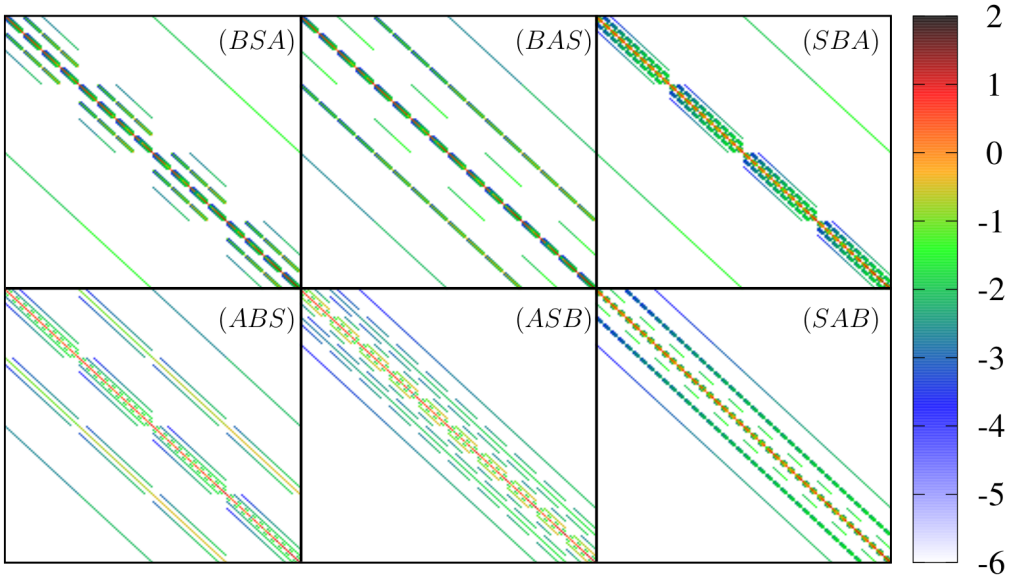


Figure 4.6: Structure of matrix $\mathbb{A}_{\mathcal{P}}$ for model B and selected energy $\varepsilon = 2.5$. Ploted values of nonzero elements in matrix are logarithms of magnitudes of elements in matrix $\mathbb{A}_{\mathcal{P}}$. For better visibility of the matrix structure, the size of the basis is lower here, i. e. $N_B = 10$, $N_S = 4$, $N_A = 4$.

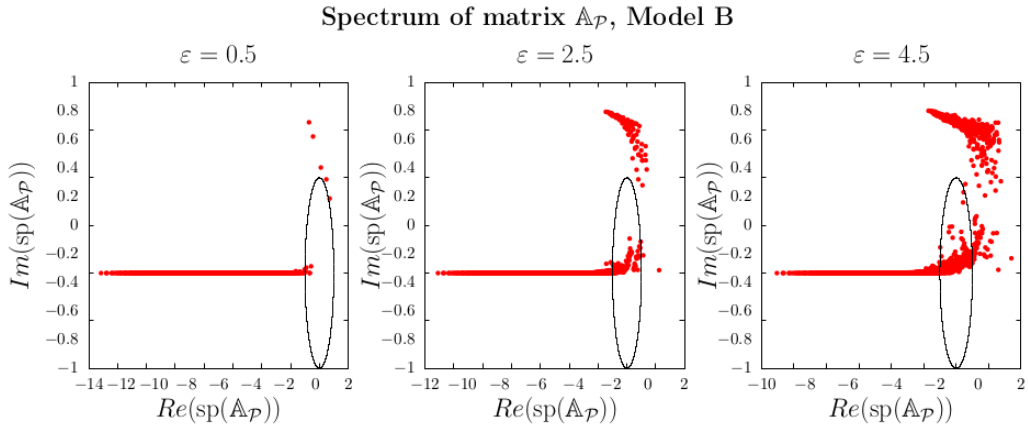


Figure 4.7: Eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$ are marked in red. For approximating eigenvalues we used LAPACK routine 'zgeev'. Black curve represents unit circle in complex plane. Remind that the size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model B is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$).

$a = 4.105$	$\omega_B = 0.07385$	$\lambda_B = 0.00$	$M_{BB} = 0.00$	$M_{BS} = 0.047$
$b = 0.15$	$\omega_S = 0.47$	$\lambda_S = 0.50$	$M_{SS} = -0.10$	$M_{BA} = 0.00$
$\alpha = 0.20$	$\omega_A = 0.0522$	$\lambda_A = 0.00$	$M_{AA} = 0.00$	$M_{SA} = 0.047$

Table 4.3: Parameters of model C. Note that $\mathbb{M}$ is a symmetric matrix, so $M_{SB} = M_{BS}$, $M_{AS} = M_{SA}$ and $M_{AB} = M_{BA}$. Size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model C is $N = 6750$ ($N_B = 30$, $N_S = 5$, $N_A = 45$).

Figure 4.9 illustrates structure of matrix $\mathbb{A}_{\mathcal{P}}$ for model C and energy $\varepsilon = 2.5$.

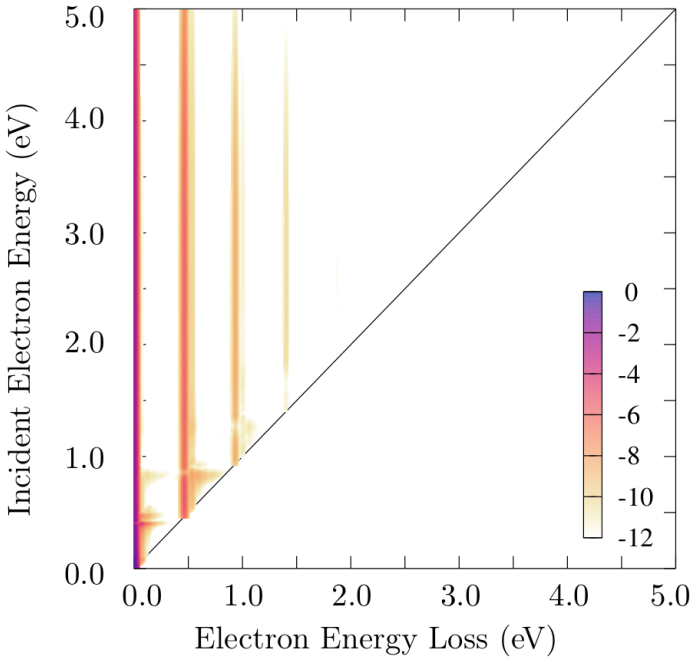


Figure 4.8: Energy-loss spectrum representing model C, i.e. plot of the integral cross section $\sigma_{\boldsymbol{\nu}_\mathrm{f}\leftarrow\boldsymbol{\nu}_\mathrm{i}}(\varepsilon)$ as a function of the incident electron energy ε and the electron energy loss. The integral cross section is ploted in a logaritmic scale.

As in the previous case, six different parts of the Figure 4.9 represent the natural reorderings of matrix $\mathbb{A}_\mathcal{P}$. We see that the structure of nonzero elements in matrix

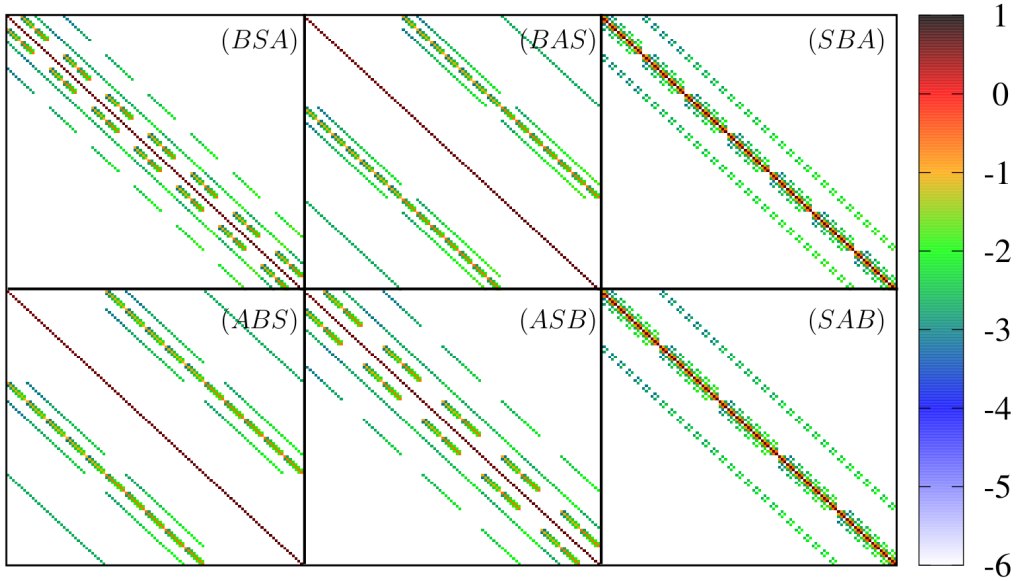


Figure 4.9: Structure of matrix $\mathbb{A}_\mathcal{P}$ for model C and selected energy $\varepsilon = 2.5$. Ploted values of nonzero elements in matrix are logarithms of magnitudes of elements in matrix $\mathbb{A}_\mathcal{P}$. For better visibility of the matrix structure, the size of the basis is lower here, i. e. $N_B = 5$, $N_S = 3$, $N_A = 7$.

ariring from model C differs more significantly from the previous two models. For comparison with previous models, we have again plotted the eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$.

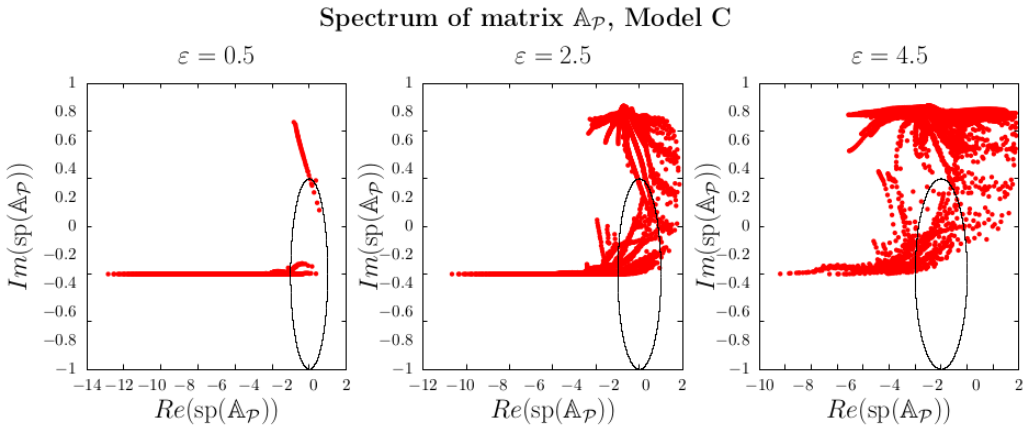


Figure 4.10: Eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$ are marked in red. For approximating eigenvalues we used LAPACK routine 'zgeev'. Black curve represents unit circle in complex plane. Remind that the size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model C is $N = 6750$ ($N_B = 30$, $N_S = 5$, $N_A = 45$).

4.1 Properties of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$

In this section, we will look at the basic properties of the matrices of systems of equations that we need to solve. Let us remind that we have to solve a lot of linear systems that differ from each other by the value of the electron energy ε. So we can write

$$\mathbb{A}_{\mathcal{P}}(\varepsilon)\mathbf{x} = \mathbf{b}, \tag{4.3}$$

where $\mathbb{A}_{\mathcal{P}}(\varepsilon) \in \mathbb{C}^{N\times N}$ is symmetric matrix we derived in the previous chapter and $\mathbf{b} \in \mathbb{R}^N$ and $\mathbf{x} \in \mathbb{C}^N$.

Naturally, the first feature we are interested in is whether the individual systems of linear equations are non-singular. Unfortunately, the fact is that our models depend on many parameters and are therefore too complex to be able to prove analytically that the matrices are invertible. For this reason, we decided to verify that they are non-singular numerically. We will use the fact that matrix $\mathbb{A}_{\mathcal{P}}$ is invertible if and only if $0 \notin \mathrm{sp}(\mathbb{A}_{\mathcal{P}})$ which is equivalent to $0 \notin \mathrm{sp}(\mathbb{A}_{\mathcal{P}}^H\mathbb{A}_{\mathcal{P}})$ (here $\mathbb{A}_{\mathcal{P}}^H$ denotes conjugate transpose of matrix $\mathbb{A}_{\mathcal{P}}$). Figure 4.11 shows the smallest eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ as a function of electon energy. We see, that all the eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}^H\mathbb{A}_{\mathcal{P}}$ are nonzero. For this reason we know that matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ is invertible for every ε that we use in the discretization of the energy interval.

In addition, for the sake of interest, we show in the Figure 4.12 the dependence of the condition number of the matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ on the electron energy.

Another property we want to determine for matrices is the so-called diagonal dominance.

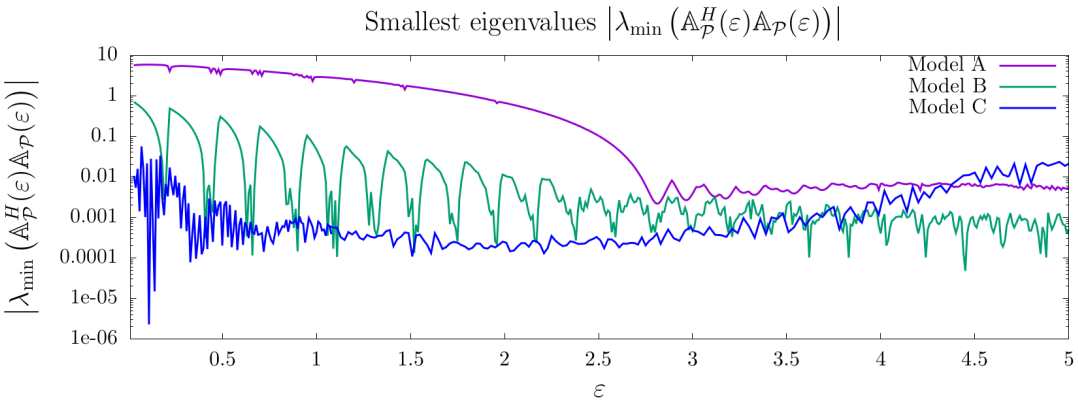


Figure 4.11: Eigenvalues of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ with the smallest absolute value for models A, B and C. For approximating eigenvalues we used LAPACK routine 'zgeev'.

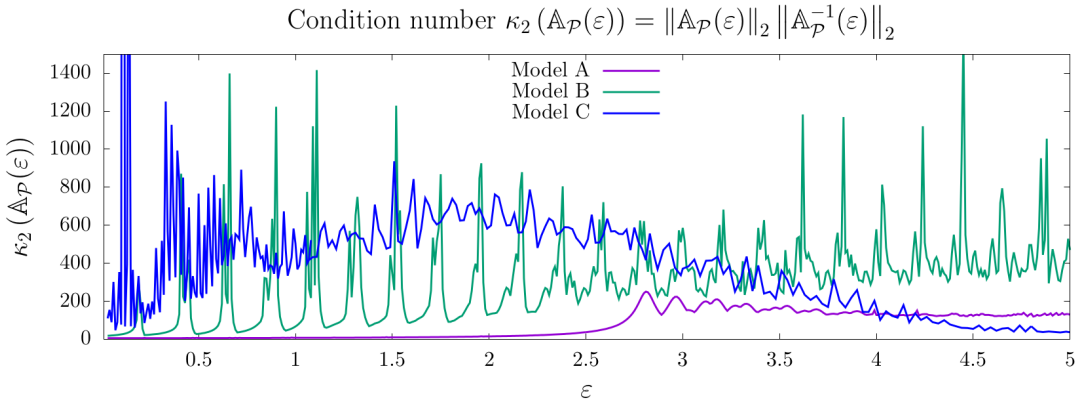


Figure 4.12: Condition number of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model A, B and C.

Definition 6. *A matrix $\mathbb{A}_{\mathcal{P}}$ is strictly diagonally dominant if*

$$|(\mathbb{A}_{\mathcal{P}})_{kk}| > \sum_{j \neq k} \left|(\mathbb{A}_{\mathcal{P}})_{jk}\right| \quad k = 1, \cdots, N.$$

Even though the connection between the convergence of Krylov subspace methods (COCG and GMRES in particular) and this property is not straightforward, it can be useful for us in constructing preconditioner. Let us define the auxiliary function $y(k)$

$$y(k) = |(\mathbb{A}_{\mathcal{P}})_{kk}| - \sum_{j \neq k} \left|(\mathbb{A}_{\mathcal{P}})_{jk}\right| \quad k = 1, \cdots, N. \tag{4.4}$$

It is clear, that if $y(k) > 0 \ \forall k = 1, \cdots, N$, where '$k$' is a column index of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$, then the matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ is strictly diagonally dominant. Figure 4.13 shows the function $y(k)$ for three selected energies ε for model A. We see, that for small energy the values of function $y(k)$ are indeed positive and therefore the matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ is strictly diagonally dominant. On the contrary this is not the case for high energies. We will not show these plots for the other models (i.e. B and C), because they look very similar. The following Figure 4.14 is more illustrative. It shows the minimal value $\min_k y(k)$ as a function of electron energy. This is

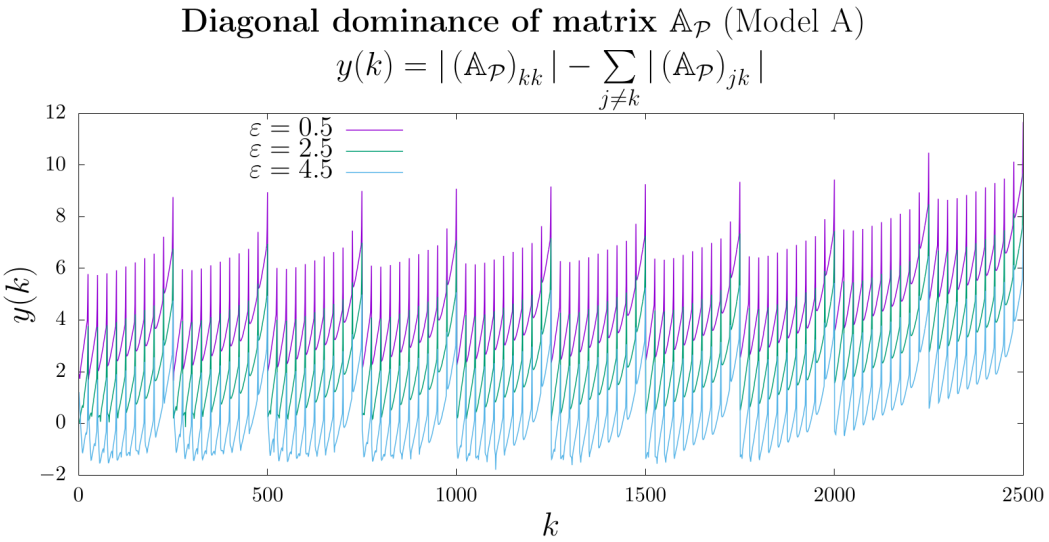


Figure 4.13: Plot of function $y(k)$ for three selected energies ε and model A.

sufficient information because we know that matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ is strictly diagonally dominant if and only if

$$\min_k y(k) > 0.$$

The figure shows that the matrices in models B and C are not diagonally dom-

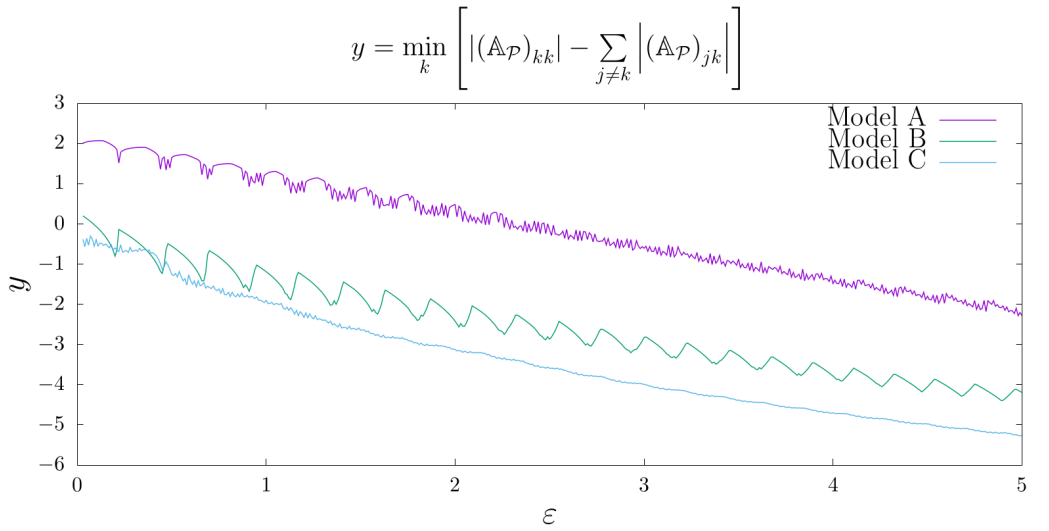


Figure 4.14: Diagonal dominance of matrices

inant practically without exception. The situation is slightly different for model A. However, even for this model, the matrices $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ are not diagonally dominant for all energies ε. For energies greater than (approximately) 2.5 eV, relation

$$\min_k y(k) < 0.$$

already applies and matrices $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ are not diagonally dominant.

Finally, let us find out if our matrices are normal or, better said, how much non-normal they are. We know that this property plays a role in the GMRES

convergence rate estimates. Let us again define an auxiliary function $z(\varepsilon)$

$$z(\varepsilon) = \left\| \mathbb{A}_{\mathcal{P}}^H(\varepsilon)\mathbb{A}_{\mathcal{P}}(\varepsilon) - \mathbb{A}_{\mathcal{P}}(\varepsilon)\mathbb{A}_{\mathcal{P}}^H(\varepsilon) \right\|_F / \left\| \mathbb{A}_{\mathcal{P}}(\varepsilon) \right\|_F^2 . \tag{4.5}$$

Assuming $\mathbb{A}_{\mathcal{P}}(\varepsilon) \neq \mathbf{0}$ it holds that matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ is normal if and only if $z(\varepsilon) = 0$. Figure 4.15 shows the plot of function $z(\varepsilon)$ for our three models. It is clear, that

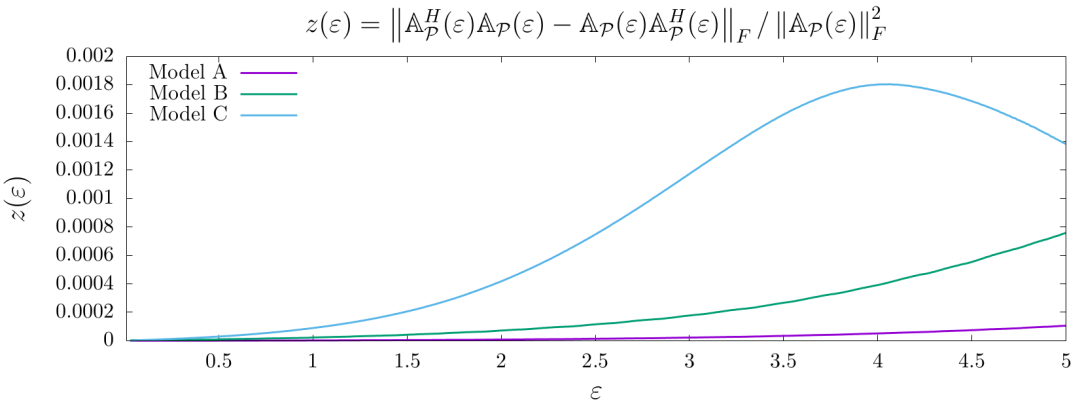


Figure 4.15: Values of $z(\varepsilon)$ as a function of electron energy for our three models.

none of the matrices in our models are normal. We have also tried calculating the condition number of matrices $\mathbb{X}(\varepsilon)$ formed by eigenvectors of matrices $\mathbb{A}_{\mathcal{P}}(\varepsilon)$, i.e. $\mathbb{A}_{\mathcal{P}}(\varepsilon) = \mathbb{X}(\varepsilon)\mathbb{J}(\varepsilon)\mathbb{X}^{-1}(\varepsilon)$. Surprisingly, although the matrices are not normal, the condition numbers $\kappa_2(\mathbb{X}(\varepsilon))$ all came out almost equal to one for most of the energies ε.

4.2 Properties of matrix $\mathbb{B}_{\mathcal{P}}$

We can write the matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ as a sum of real and imaginary part

$$\mathbb{A}_{\mathcal{P}}(\varepsilon) = \mathrm{Re}\left(\mathbb{A}_{\mathcal{P}}(\varepsilon)\right) + i\,\mathrm{Im}\left(\mathbb{A}_{\mathcal{P}}(\varepsilon)\right) \equiv \mathbb{B}_{\mathcal{P}}(\varepsilon) + i\,\mathbb{D}_{\mathcal{P}}(\varepsilon), \tag{4.6}$$

where the real part equals

$$\mathbb{B}_{\mathcal{P}}(\varepsilon) = \mathbb{E}_{\mathcal{P}}\left(\varepsilon\right) - \mathbb{H}_{\mathcal{P}} - \mathrm{Re}\left(\mathbb{F}_{\mathcal{P}}\left(\varepsilon\right)\right) \equiv \mathbb{E}_{\mathcal{P}}\left(\varepsilon\right) - \mathbb{H}_{\mathcal{P}} - \Delta_{\mathcal{P}}\left(\varepsilon\right) \tag{4.7}$$

and for imaginary part it holds

$$\mathbb{D}_{\mathcal{P}}(\varepsilon) = -\mathrm{Im}\left(\mathbb{F}_{\mathcal{P}}\right) \equiv \frac{1}{2}\Gamma_{\mathcal{P}}. \tag{4.8}$$

The real and imaginary parts of the matrix are used to construct the splitting preconditioners. Let us now briefly comment on some properties of these matrices. First and foremost, most of the authors of articles dealing with the construction of preconditioners assume, that the matrix $\mathbb{B}_{\mathcal{P}}(\varepsilon)$ is non-singular. Matrix

$$\mathbb{B}_{\mathcal{P}} \equiv \mathbb{B}_{\mathcal{P}}(\varepsilon) \equiv \mathrm{Re}\left(\mathbb{A}_{\mathcal{P}}(\varepsilon)\right) = \mathbb{E}_{\mathcal{P}}(\varepsilon) - \mathbb{H}_{\mathcal{P}} - \Delta_{\mathcal{P}}(\varepsilon) \tag{4.9}$$

is invertible if and only if $\mathbb{B}_{\mathcal{P}}$ has trivial kernel. If there exists a nonzero vector $\mathbf{v} \in \mathbb{C}^N$ such that

$$\left[\mathbb{E}_{\mathcal{P}} - \mathbb{H}_{\mathcal{P}} - \Delta_{\mathcal{P}}(\varepsilon)\right]\mathbf{v} = \mathbf{0}, \tag{4.10}$$

or equivalently

$$[\mathbb{H}_{\mathcal{P}} + \Delta_{\mathcal{P}}(\varepsilon)]_{\mathcal{P}} \mathbf{v} = \mathbb{E}_{\mathcal{P}}(\varepsilon)\mathbf{v} = E(\varepsilon)\mathbf{v}, \tag{4.11}$$

than $\mathbb{B}_{\mathcal{P}}$ is singular. This is equivalent to the fact that $E(\varepsilon) \in \mathrm{sp}\left(\mathbb{H}_{\mathcal{P}} + \Delta_{\mathcal{P}}(\varepsilon)\right)$. We know, that matrix $\mathbb{H}_{\mathcal{P}} + \Delta_{\mathcal{P}}(\varepsilon)$ is real and symmetric, so it has only real eigenvalues $\lambda_i(\epsilon)$, $i \in \{1, \cdots, N\}$ and

$$E(\varepsilon) = \varepsilon + \frac{1}{2}\left(\omega_B + \omega_S + \omega_A\right). \tag{4.12}$$

We can plot both $E(\varepsilon)$ and $\lambda_i(\varepsilon)$ as functions of ε and see if there exists ε such that $E(\varepsilon) = \lambda_i(\varepsilon)$ for some i. Figure 4.16, 4.17 and 4.18 show selected eigenvalues of matrix $\mathbb{H}_{\mathcal{P}}+\Delta_{\mathcal{P}}(\varepsilon)$ as functions of ε. Initial energy $E(\varepsilon)$ of the system is plotted with a red line. Eigenvalues are approximated with accuracy 10^{-2}. Although

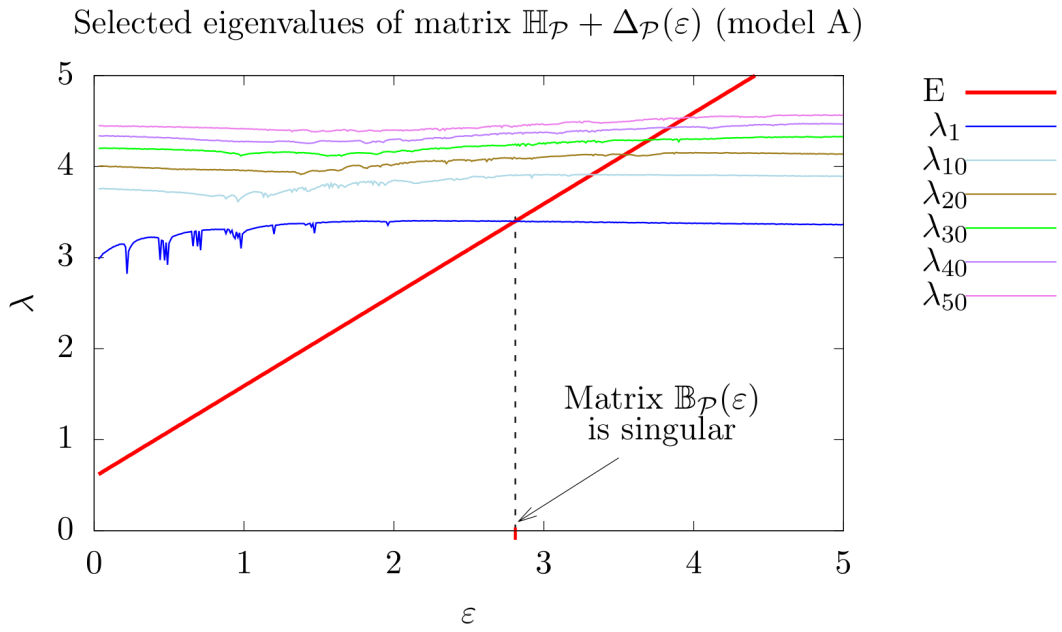


Figure 4.16: Eigenvalues of matrix $\mathbb{H}_{\mathcal{P}} + \Delta(\epsilon)$ for model A. Size of matrix is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$). Eigenvalues are approximated using LAPACK function 'dsyevx' with tolerance 10^{-2}. Initial energy $E(\varepsilon)$ of the system is plotted with a red line.

the probability that $\mathbb{B}_{\mathcal{P}}(\varepsilon)$ is singular for particular $\varepsilon \in [0; 5]$ is small (in exact arithmetics it equals zero), it is clear from all four plots, that matrix $\mathbb{B}_{\mathcal{P}}(\varepsilon)$ is close to singular for some energies ε. As a consequence, we generaly can not assume that $\mathbb{B}_{\mathcal{P}}$ is invertible, because we need to solve the system for aproximately 500 energies $\varepsilon \in [0; 5]$ and if we are unlucky, we can choose ε for which the matrix $\mathbb{B}_{\mathcal{P}}(\varepsilon)$ is singular.

4.3 Properties of matrix $\mathbb{D}_{\mathcal{P}}(\varepsilon)$

Matrix $\mathbb{D}_{\mathcal{P}}(\varepsilon)$ is an imaginary part of $\mathbb{A}_{\mathcal{P}}(\varepsilon)$

$$\mathbb{D}_{\mathcal{P}}(\varepsilon) = \mathrm{Im}\left(-\mathbb{F}_{\mathcal{P}}\left(a, b, \alpha, \epsilon\right)\right) \equiv \frac{1}{2}\Gamma_{\mathcal{P}}\left(E(\varepsilon) - E_{\mathbf{n}'}\right)\delta_{\mathbf{n},\mathbf{n}'}. \tag{4.13}$$

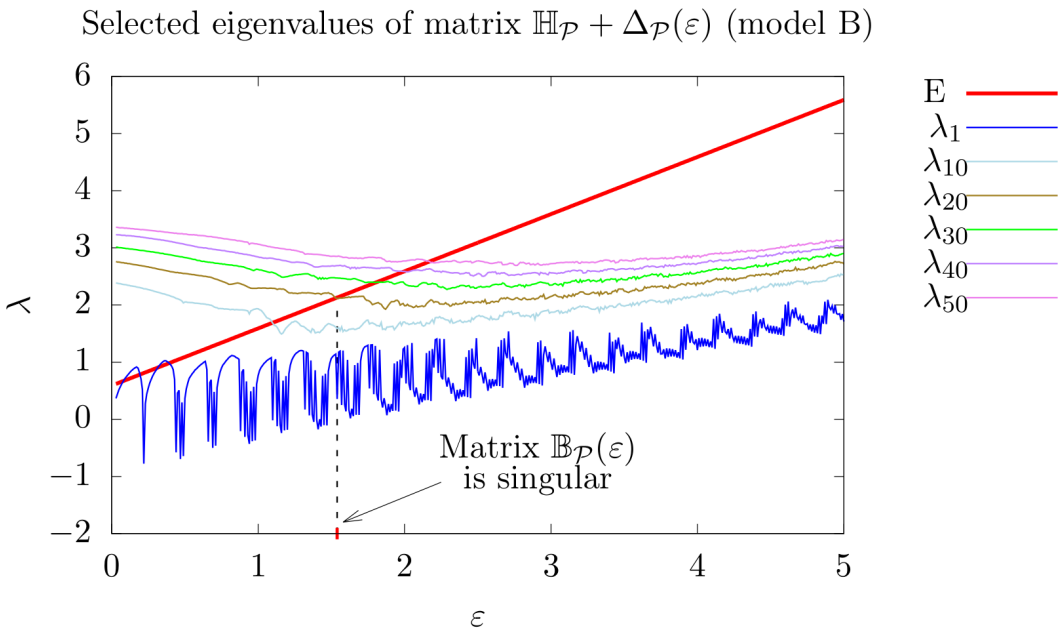


Figure 4.17: Eigenvalues of matrix $\mathbb{H}_{\mathcal{P}} + \Delta(\varepsilon)$ for model B. Size of matrix is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$). Eigenvalues are approximated using LAPACK function 'dsyevx' with tolerance 10^{-2}. Initial energy $E(\varepsilon)$ of the system is plotted with a red line.

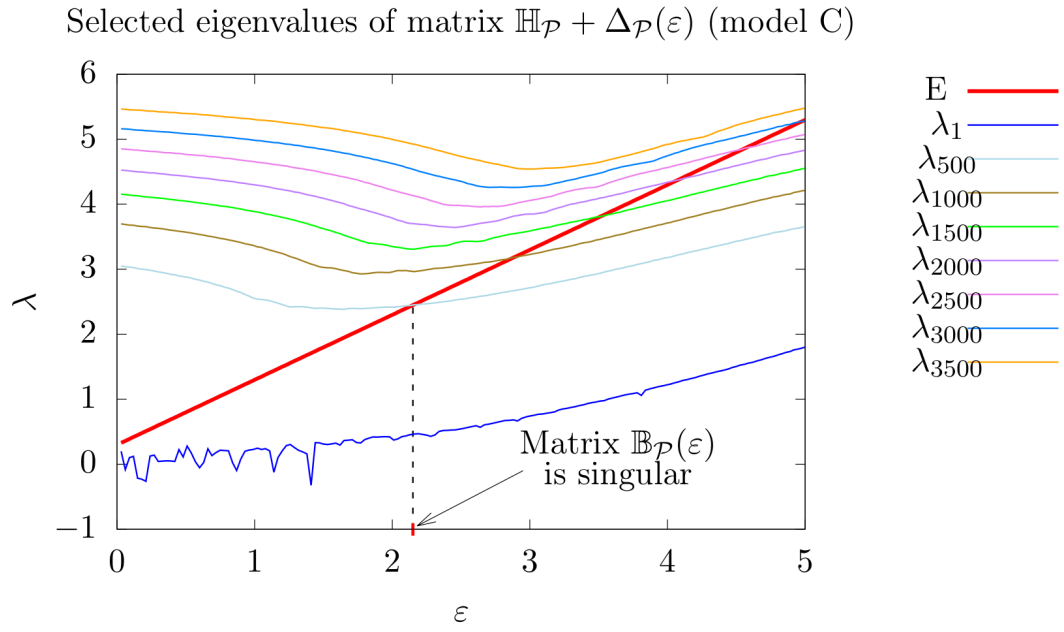


Figure 4.18: Eigenvalues of matrix $\mathbb{H}_{\mathcal{P}} + \Delta(\varepsilon)$ for model C. Size of matrix is $N = 6750$ ($N_B = 30$, $N_S = 5$, $N_A = 45$). Eigenvalues are approximated using LAPACK function 'dsyevx' with tolerance 10^{-2}. Initial energy $E(\varepsilon)$ of the system is plotted with a red line.

Let us remind that

$$E(\varepsilon) = \varepsilon + \frac{1}{2}\left(\omega_B + \omega_S + \omega_A\right). \tag{4.14}$$

and

$$E_{\mathbf{n}'} = \omega_B \left(n_B + \frac{1}{2} \right) + \omega_S \left(n_S + \frac{1}{2} \right) + \omega_A \left(n_A + \frac{1}{2} \right). \tag{4.15}$$

Function $\Gamma_{\mathcal{P}}$ is plotted in the Figure 4.19. We see, that almost all diagonal

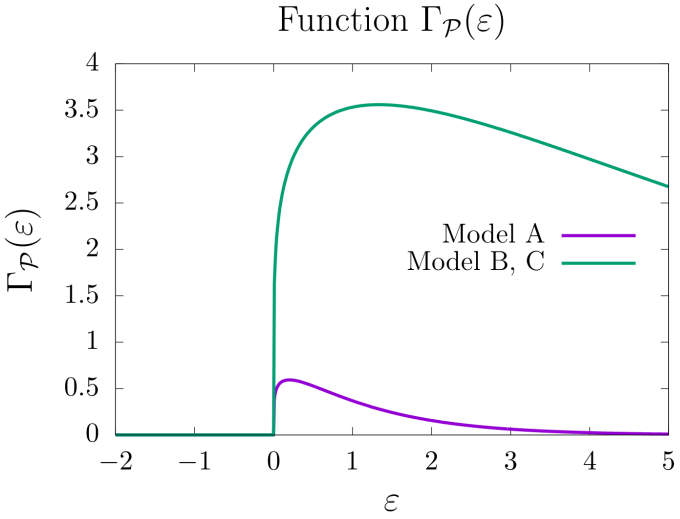


Figure 4.19: Function $\Gamma_{\mathcal{P}}$

values of matrix $\mathbb{D}_{\mathcal{P}}$ must be equal to zero, because the value of $(E(\varepsilon) - E_{\mathbf{n}'})$ is less than zero especially for small ε. For this reason, the matrix $\mathbb{D}_{\mathcal{P}}(\varepsilon)$ is real diagonal positive semidefinite (singular) matrix.

5. Numerical experiments

Before we start using preconditioners, we would like to know how the iterative methods without preconditioning work for our linear systems. We will use the same stopping criterion throughout this chapter. We consider the approximation $\mathbf{x}_k$ of the solution $\mathbf{x}$ of the system $\mathbb{A}_{\mathcal{P}}(\varepsilon)\mathbf{x} = \mathbf{b}$ to be sufficiently accurate if

$$\|\mathbf{r}_k\| \leq 10^{-6} \|\mathbf{r}_0\| \tag{5.1}$$

holds. Here $\mathbf{r}_k = \mathbf{b} - \mathbb{A}_{\mathcal{P}}(\varepsilon)\mathbf{x}_k$ is the k-th residual vector. The following sections focus on solving the linear systems arising from our three models.

Model A

Let us firstly focus on model A. Let us remind, that the size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model A, for which we test the behaviour of different iterative aproaches, is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$). In figure 5.1 we have ploted number of iterations for iterative methods COCG and GMRES as a functions of electron energy ε. We see that the number of iterations strongly depends on the energy

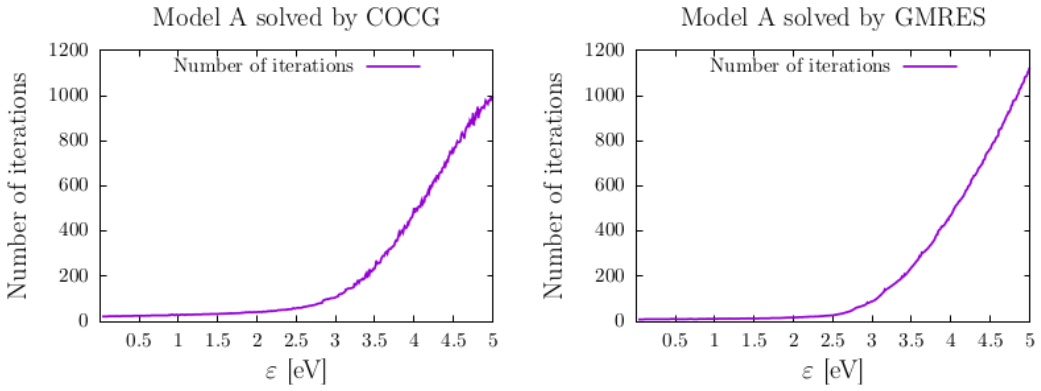


Figure 5.1: Plots of number of iterations as a function of electron energy ε for COCG and GMRES.

of the electron. In the Figure 5.2 we ploted convergence curves for three selected energies ε and methods COCG and GMRES. We also see that the number of iterations starts to increase sharply when the electron energy exceeds 2.5 eV. At first glance, one could say that this phenomenon is correlated with the fact that for approximately the same value of energy the matrix loses the property of diagonal dominance. Looking at the 2D electron energy-loss spectrum for model A, it is also clear that for energies greater than 2.5 eV the cross sections have much higher values, so this is a physically much more interesting area.

However, what is more important than the number of iterations is the time required to perform the calculation. In our case, we are interested in the time for which we are able to construct the 2D electron energy-loss spectrum, which means solving 500 systems of equations for different incident electron energies ε. We have measured that for model A the time for solving all 500 systems of linear equations using GMRES without preconditioning is $t = 7601.10$ s. For the

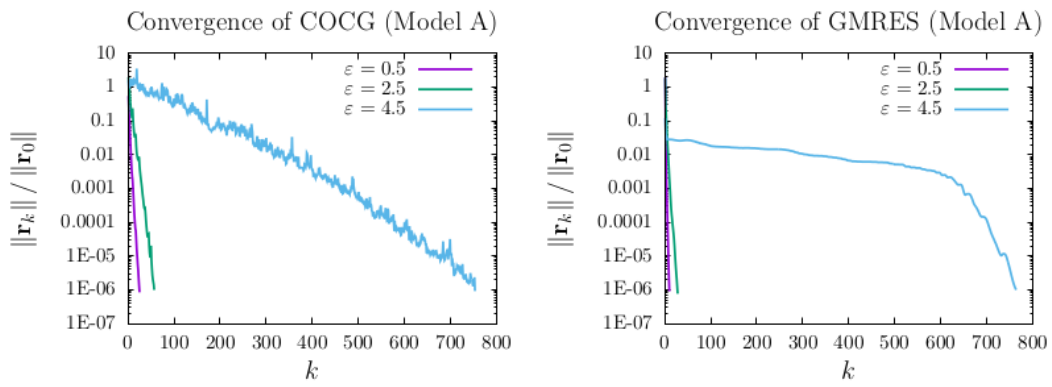


Figure 5.2: Plots of the relative norm of the residual $\|\mathbf{r}_k\| / \|\mathbf{r}_0\|$ as a function of the number of iterations.

COCG method, the calculation takes $t = 254.06$ s. Note that the calculation using GMRES method takes almost thirty times more time than for COCG, even though the number of iterations does not differ significantly. This is caused by the fact that GMRES uses long recurrences to construct the basis of the Krylov subspace and therefore its time complexity increases rapidly with the number of iterations. Therefore, we also believe that in the case of the GMRES method preconditioning can help much more significantly than in the case of the COCG method.

Model B

Now let us look at the results for model B. Similarly as for model A, the size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model B is $N = 2500$ ($N_B = 25$, $N_S = 10$, $N_A = 10$). In Figure 5.3 we have ploted number of iterations for iterative methods COCG and GMRES as a functions of electron energy ε. We see, that the number of iter-

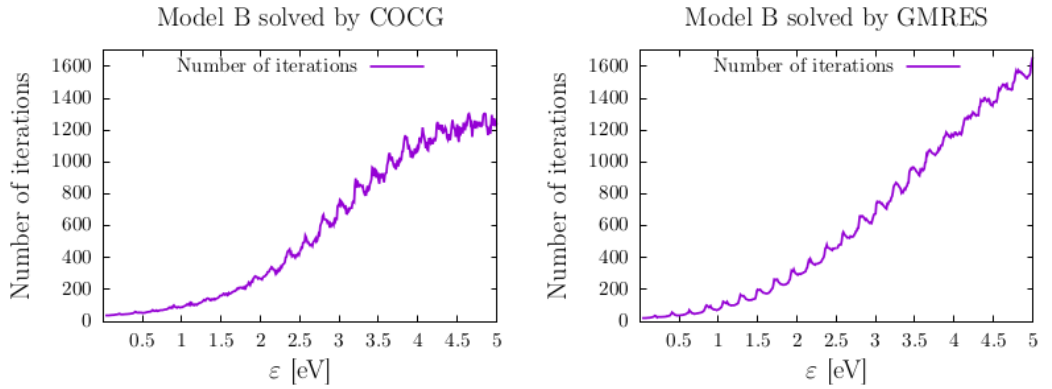


Figure 5.3: Plots of number of iterations as a function of electron energy ε for COCG and GMRES.

ations has a slightly different behavior from model A. We see that the function contains significant oscillations. Also with a little imagination, we could say that the number of iterations correlates with the condition number of the matrix (see Figure 4.12.) In the Figure 5.4 se ploted convergence curves for three selected

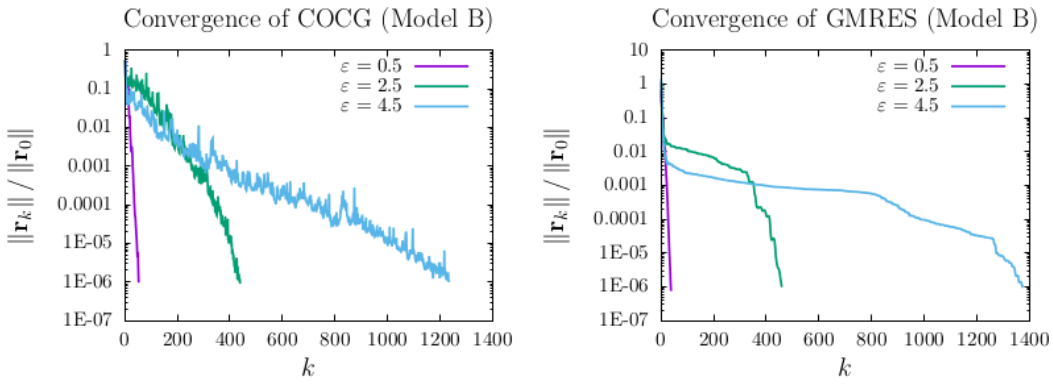


Figure 5.4: Plots of the relative norm of the residual $\|\mathbf{r}_k\| / \|\mathbf{r}_0\|$ as a function of the number of iterations.

energies ε and methods COCG and GMRES. We can see that the convergence curves for individual methods look quite different.While in the GMRES method the relative norms of the residual are a non-increasing function of k, in the COCG method this quantity oscillates significantly. However, this behavior can be assumed, because unlike the GMRES method, the COCG method is not based on the minimization of the residual in individual iterations.

Again, we would like to know the time for which we are able to construct a vibrational spectrum. We have found, that for model B it takes $t = 75495.15$ s to solve all 500 systems of linear equations using GMRES without preconditioning. This time corresponds to about 21 hours, which is a really long time for solving such a relatively small problem. For the COCG method, the calculation takes $t = 758.59$ s.

Model C

Finally let us look at the results for model C. The size of matrix $\mathbb{A}_{\mathcal{P}}(\varepsilon)$ for model C is $N = 6750$ ($N_B = 30$, $N_S = 5$, $N_A = 45$). In Figure 5.5 we have ploted number of iterations for iterative methods COCG and GMRES as a functions of electron energy ε.

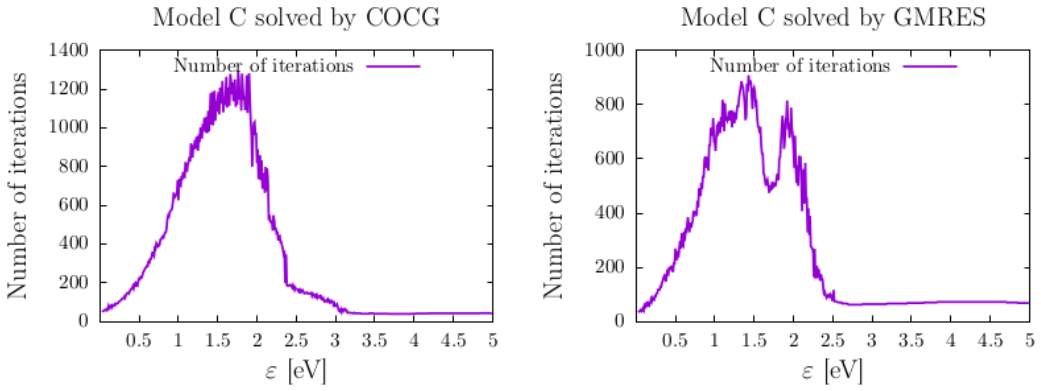


Figure 5.5: Plots of number of iterations as a function of electron energy ε for COCG and GMRES.

In the Figure 5.6 se ploted convergence curves for three selected energies ε and

methods COCG and GMRES. For model C, it is no longer true that the number

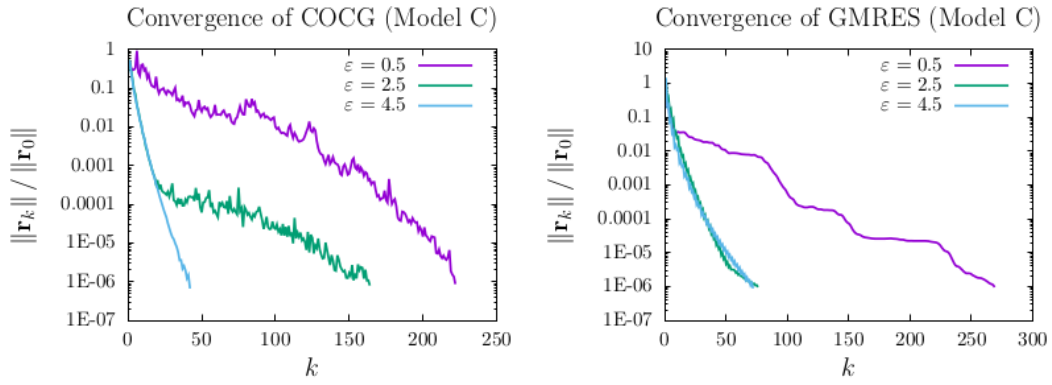


Figure 5.6: Plots of the relative norm of the residual $\|\mathbf{r}_k\| / \|\mathbf{r}_0\|$ as a function of the number of iterations.

of iterations increases with the electron energy. On the contrary, for $\varepsilon > 3.5$, both methods converge in less than 70 iterations. We would naturally like to know the time for which we are able to construct a vibrational spectrum. We have measured, that for model C it takes $t = 9374.98$ s to solve all 500 systems of linear equations using GMRES without preconditioning. For the COCG method, the calculation takes $t = 988.60$ s. Note that since we did not test the convergence of physical results with basis size, the basis used by us for testing the iterative methods may not be large enough in terms of physically relevant results.

5.1 Jacobi preconditioning

In this section we use the Jacobi preconditioning and its modifications, i. e. we use diagonal preconditioning matrix $\mathbb{K}$. The reason for choosing this structure of preconditioning matrix is that it is both easy to construct the inverse of this matrix and apply it to the vector. Therefore we expect, that using this method does not worsen the time complexity of iterative methods too much.

Let us discuss three diagonal preconditioners we consider the most natural for our system of equations. Firstly, let the preconditioning matrix $\mathbb{K}_1$ be defined simply as a diagonal of matrix $\mathbb{A}_\mathcal{P}$

$$(\mathbb{K}_1)_{i,j} = \begin{cases} (\mathbb{A}_\mathcal{P})_{i,j} & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \tag{5.2}$$

Secondly, let matrix $\mathbb{K}_2$ be defined as a diagonal matrix with absolute values of diagonal elements of matrix $\mathbb{A}_\mathcal{P}$ on the main diagonal

$$(\mathbb{K}_2)_{i,j} = \begin{cases} \left|(\mathbb{A}_\mathcal{P})_{i,j}\right| & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \tag{5.3}$$

The matrix $\mathbb{K}_2$ is in contrast to $\mathbb{K}_1$ real and positive definite. Finally, we define matrix $\mathbb{K}_3$ as a modification of $\mathbb{K}_2$, where we replace values which are close to zero with ones

$$(\mathbb{K}_3)_{i,j} = \begin{cases} \max\left\{\left|(\mathbb{A}_\mathcal{P})_{i,j}\right|; 1\right\} & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \tag{5.4}$$

The reason is that while using the preconditioning we do not want to amplify magnitudes of small elements in matrix $\mathbb{A}_{\mathcal{P}}$. We consider 'Cholesky' decomposition $\mathbb{K}_i = \mathbb{L}_{\mathrm{D}}\mathbb{L}_{\mathrm{D}}^T, i \in \{1;2;3\}$ with $\mathbb{L}_{\mathrm{D}} = \sqrt{\mathbb{K}}$. It is clear that the preconditioned matrix $\mathbb{L}_{\mathrm{D}}^{-1}\mathbb{A}_{\mathcal{P}}\mathbb{L}_{\mathrm{D}}^{-T}$ is symmetric, because

$$\left(\mathbb{L}_{\mathrm{D}}^{-1}\mathbb{A}_{\mathcal{P}}\mathbb{L}_{\mathrm{D}}^{-T}\right)^T = \left(\mathbb{L}_{\mathrm{D}}^{-T}\right)^T \left(\mathbb{A}_{\mathcal{P}}\right)^T \left(\mathbb{L}_{\mathrm{D}}^{-1}\right)^T = \mathbb{L}_{\mathrm{D}}^{-1} \left(\mathbb{A}_{\mathcal{P}}\right)^T \mathbb{L}_{\mathrm{D}}^{-T} = \mathbb{L}_{\mathrm{D}}^{-1}\mathbb{A}_{\mathcal{P}}\mathbb{L}_{\mathrm{D}}^{-T}. \quad (5.5)$$

Note that in practical implementation it is not necessary to construct the square root of the matrix $\mathbb{K}$, as it is enough to act on a suitable vector (see the algorithms in the section 3.2 and 3.3) with the matrix $\mathbb{K}^{-1}$. In the next sections, we test the effectiveness of the defined methods for our models.

Model A

We will first look at the results obtained for model A. We have tested the above mentioned preconditioners for two Krylov subspace methods - COCG and GMRES. In Figure 5.7 we compare the number of iterations used with and without preconditioning. Number of iterations are ploted as a function of electron energy ε. We can see, that for both iterative methods the preconditioning $\mathbb{K}_1$ reduces

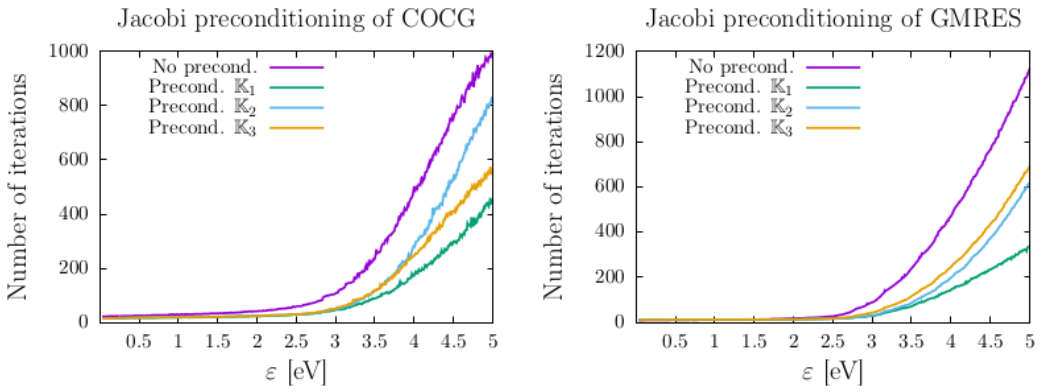


Figure 5.7: Plots of number of iterations as a function of electron energy ε for different Jacobi preconditioning matrices for model A.

the number of iterations the most. The other two preconditioners also reduce the number of iterations, but they are less efficient. We can also see, that the results for preconditioners $\mathbb{K}_2$ and $\mathbb{K}_3$ are almost the same for small energies. The reason is, that for these energies, the absolute values of the diagonal elements are almost always bigger than one. Absolute values of diagonal elements of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$ are for interest depicted in figure 5.8.

In the table 5.1 we can see measured times for COCG and GMRES with preconditioning $\mathbb{K}_1$, $\mathbb{K}_2$ and $\mathbb{K}_3$. The table shows that for GMRES the best diagonal preconditioning reduces the calculation time thirty times. On the other hand, the best improvement for method COCG is only three times. This is not surprising since the COCG alone (without preconditioning) is fairly efficient. The Figure 5.9 for a better idea of the efficiency shows the measured times for different diagonal preconditioning aproaches in comparison to the methods without preconditioning.

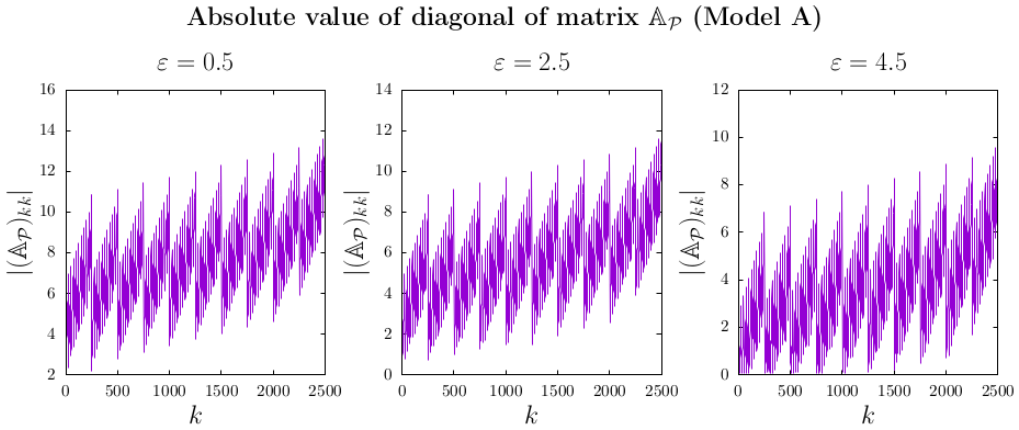


Figure 5.8: Absolute values of diagonal elements of matrix $\mathbb{A}_\mathcal{P}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$. Number k denotes row (and column) index in matrix $\mathbb{A}_\mathcal{P}$.

	Measured time [s]	
	COCG	GMRES
No preconditioning	254.06	7601.10
$\mathbb{K}_1$	86.69	230.05
$\mathbb{K}_2$	131.77	826.42
$\mathbb{K}_3$	126.92	1215.08

Table 5.1: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with $\mathbb{K}_1 - \mathbb{K}_3$ needed for construction of 2D electron energy-loss spectrum for model A.

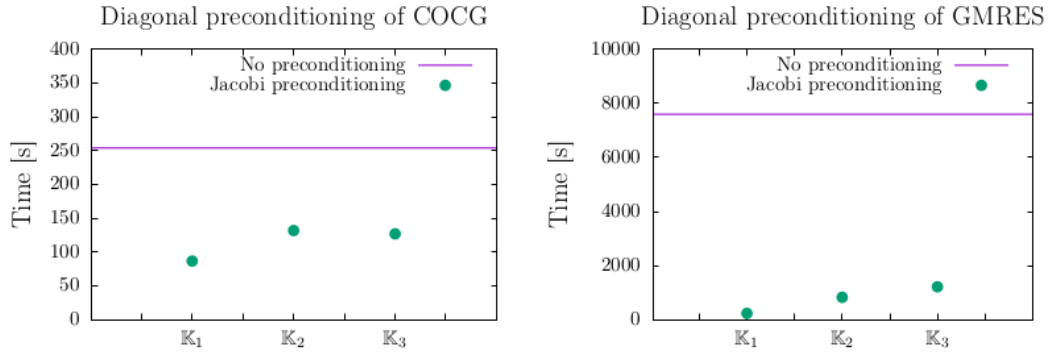


Figure 5.9: Plots of the measured times for different Jacobi preconditioning matrices for model A.

Model B

We have again tested the above mentioned preconditioners for two Krylov subspace methods. In the Figure 5.10 we compare the number of iterations used with and without preconditioning. Number of iterations are ploted as a function of electron energy ε. We can see, that in contrast to model A, in this case the Jacobi preconditioning does not always work well. In fact, for the method COCG, it sometimes increases the number of iterations instead of decreasing it.

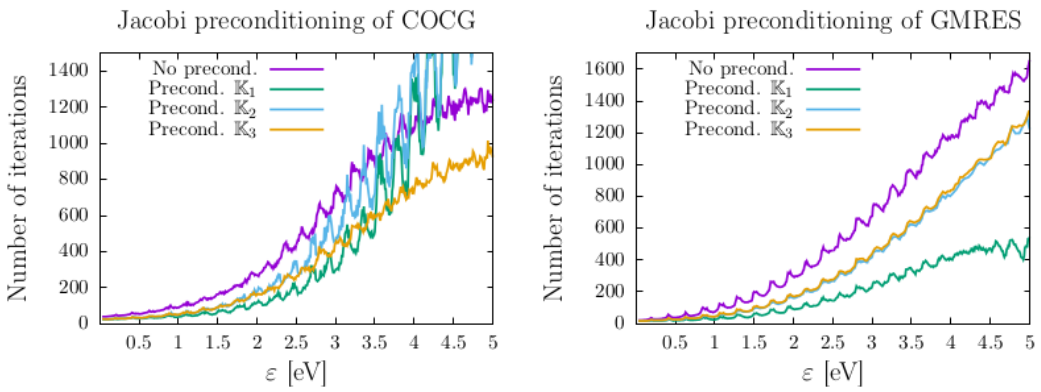


Figure 5.10: Plots of number of iterations as a function of electron energy ε for different Jacobi preconditioning matrices for model B.

Unlike model A, the matrix $\mathbb{K}_3$ appears to be the best for model B and COCG method. On the other hand, we can see that for GMRES, the most efficient diagonal preconditioner is again matrix $\mathbb{K}_1$. We again visualise the absolute values of diagonal elements of matrix $\mathbb{A}_\mathcal{P}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$ in Figure 5.11. In contrast to model A, the absolute values of the diagonal elements

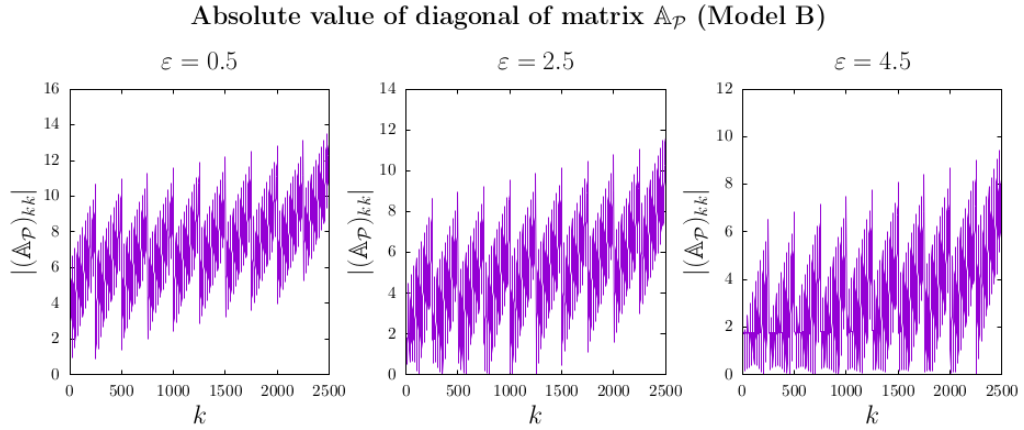


Figure 5.11: Absolute values of diagonal elements of matrix $\mathbb{A}_\mathcal{P}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$. Number k denotes row (and column) index in matrix $\mathbb{A}_\mathcal{P}$.

are often close to zero even for smaller energies. It means, that preconditioners $\mathbb{K}_2$ and $\mathbb{K}_3$ differ much more markedly.

In the table 5.2 we can see measured times for COCG and GMRES with preconditioning $\mathbb{K}_1$, $\mathbb{K}_2$ and $\mathbb{K}_3$. As we know and as can be seen from the table, it is particularly important to speed up the calculation using preconditioning for model B (especially for the GMRES method). In contrast to model A, here preconditioner $\mathbb{K}_3$ turns out to be the best for the COCG method, since it reduces the computation time more than twice. On the other hand, for the GMRES method, we achieved the greatest speedup with preconditioner $\mathbb{K}_1$. In this case, the calculation was 52 times faster. This is a good achievement for the diagonal preconditioner, considering that the matrix $\mathbb{A}_\mathcal{P}$ is not diagonally dominant for model B, and thus its diagonal cannot be assumed to approximate it well.

	Measured time [s]	
	COCG	GMRES
No preconditioning	758.59	75495.15
$\mathbb{K}_1$	429.41	1445.52
$\mathbb{K}_2$	472.48	18524.80
$\mathbb{K}_3$	306.44	20565.59

Table 5.2: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with $\mathbb{K}_1 - \mathbb{K}_3$ needed for construction of 2D electron energy-loss spectrum for model B.

The Figure 5.12 shows the measured times for model B for different diagonal preconditioniners in comparison to the methods without preconditioning.

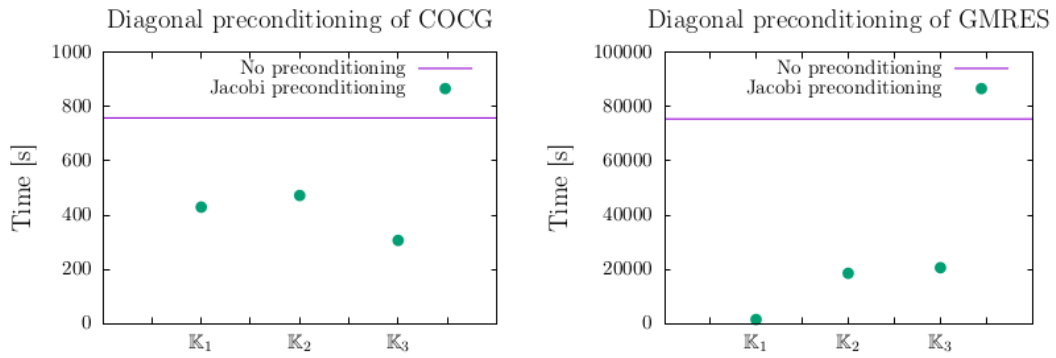


Figure 5.12: Plots of the measured times for different Jacobi preconditioning matrices for model B.

Model C

Finally, we have tested the the same preconditioners $\mathbb{K}_1$, $\mathbb{K}_2$ and $\mathbb{K}_3$ with COCG and GMRES for model C. In the Figure 5.13 we compare the number of iterations used with and without preconditioning. Number of iterations are ploted as a function of electron energy ε. We can see, that for method COCG, the Jacobi preconditioning does not work well in terms of the number of iterations for model C. Almost always the preconditioned iterative methods converge with more iterations than without using preconditioning at all. The only exception is preconditioner $\mathbb{K}_3$ which uses less iterations in comparison to COCG without preconditioning. Nevertheless, the improvement in the number of iterations is so insignificant that we cannot expect a reduction in computing time. On the other hand, we can see a significant reduction in the number of iterations of the GMRES method when using the preconditioner $\mathbb{K}_1$. Let us again look at the absolute values of diagonal elements of matrix $\mathbb{A}_{\mathcal{P}}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$. They are depicted in Figure 5.14.

In the table 5.3 we can see measured times for COCG and GMRES with preconditioning $\mathbb{K}_1$, $\mathbb{K}_2$ and $\mathbb{K}_3$. We have expected, that all three methods of preconditioning when used with COCG would increase the calculation time. However,

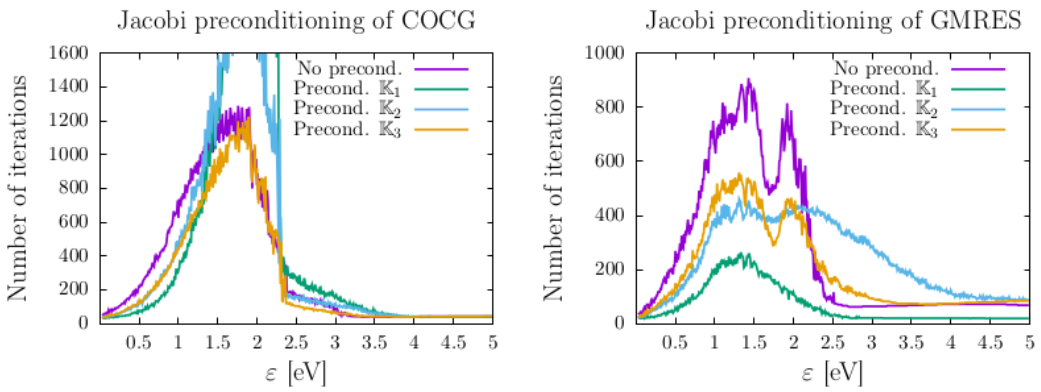


Figure 5.13: Plots of number of iterations as a function of electron energy ε for different Jacobi preconditioning matrices for model C.

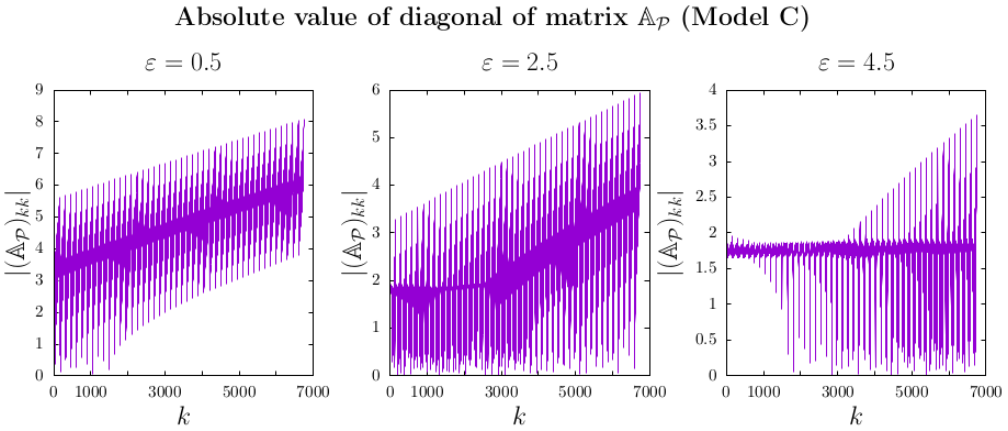


Figure 5.14: Absolute values of diagonal elements of matrix $\mathbb{A}_\mathcal{P}$ for three chosen energies $\varepsilon \in \{0.5, 2.5, 4.5\}$. Number k denotes row (and column) index in matrix $\mathbb{A}_\mathcal{P}$.

	Measured time [s]	
	COCG	GMRES
No preconditioning	988.60	9374.98
$\mathbb{K}_1$	982.27	427.37
$\mathbb{K}_2$	883.49	2955.38
$\mathbb{K}_3$	598.25	2624.70

Table 5.3: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with $\mathbb{K}_1 - \mathbb{K}_3$ needed for construction of 2D electon energy-loss spectrum for model C.

it can be seen from the measured time values that this is not true. Especially for the third preconditioner $\mathbb{K}_3$, the time has been significantly reduced. The situation is a much more optimistic for GMRES. For $\mathbb{K}_1$ the calcualtion takes aproximately twenty times less time. The Figure 5.15 shows the measured times for model C for different diagonal preconditioniners in comparison to the methods without preconditioning.

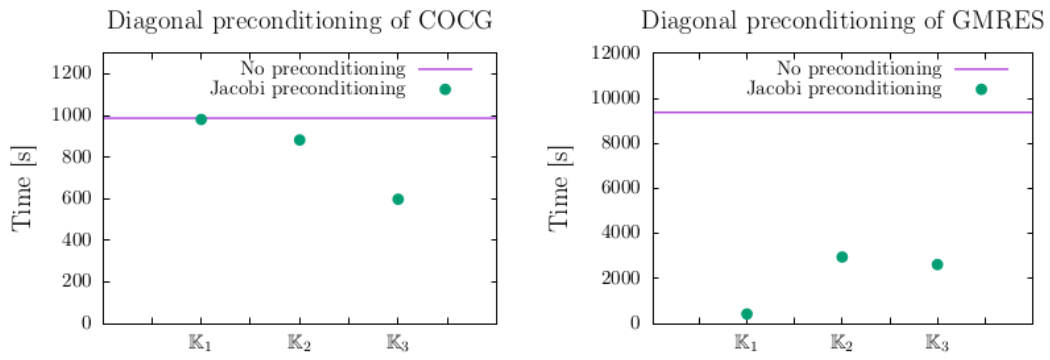


Figure 5.15: Plots of the measured times for different Jacobi preconditioning matrices for model C.

5.2 Block Jacobi preconditioning

Another possibility is to use for preconditioning a block extension of Jacobi preconitioner. Such extension can be more efficient and has a large potential to be highly useful at parallel computer architectures. Thanks to the block structure of our matrix, we believe, that this type of preconditioning is a natural choice for our problem. In order to visualize block Jacobi preconditioner, let us first describe the blocks defined by the physics of the problem that naturally imply the preconditioners. There are two straightforward ways to construct a block diagonal precondition in our case.

First six preconditioners are block diagonal matrices with 'small' blocks on the main diagonal (Figure 5.16 on the left). These blocks are five-diagonal matrices, so they can be decomposed to a product LDL^T easily. Let us define preconditioning matrix $\mathbb{K}_{BJ(S)}(BSA)$ in the following way:

$$\mathbb{K}_{BJ(S)}(BSA) = \left(\underbrace{\left[\mathbb{E}_{\mathcal{P}} - \mathbb{H}_{0,\mathcal{P}} - \Xi_S^{\mathrm{D}} - \Xi_A^{\mathrm{D}} - \epsilon_{\mathrm{d}}\mathbf{I} - \mathbb{F}_{\mathcal{P}}\right]}_{\text{Diagonal matrix}} - \Lambda_B - \Xi_B \right)(BSA), \tag{5.6}$$

where

$$\Lambda_B(BSA) = \lambda_B \cdot \mathbb{I}_{N_S \cdot N_A} \otimes \mathbb{Q}_{N_B} \tag{5.7}$$

is tridiagonal matrix and

$$\Xi_B(BSA) = \mathbb{M}_{BB} \cdot \mathbb{I}_{N_S \cdot N_A} \otimes \mathbb{S}_{N_B} \tag{5.8}$$

is five-diagonal matrix. Symbol Ξ_S^{D} stands for diagonal of matrix Ξ_S. Analogicaly, we can define other five preconditioning matrices $\mathbb{K}_{BJ(S)}(BAS)$, $\mathbb{K}_{BJ(S)}(SBA)$, $\mathbb{K}_{BJ(S)}(SAB)$, $\mathbb{K}_{BJ(S)}(ASB)$ and $\mathbb{K}_{BJ(S)}(ABS)$, which are also block diagonal matrices with five-diagonal blocks on the main diagonal. They are formed similarly as $\mathbb{K}_{BJ(S)}(BSA)$ only using a different order of Kronecker products of matrices.

Structure of matrix $\mathbb{A}_{\mathcal{P}}(BSA)$ also allows to use blocks with size $N_B \cdot N_S$ (as in Figure 5.16 in the middle). We define preconditioning matrix $\mathbb{K}_{BJ(L)}(BSA)$ in

the following way

$$\mathbb{K}_{BJ(L)}(BSA) = \left[\underbrace{\mathbb{E}_{\mathcal{P}} - \mathbb{H}_{0,\mathcal{P}} - \Xi_A^{\mathrm{D}} - \epsilon_{\mathrm{d}}\mathbf{I} - \mathbb{F}_{\mathcal{P}}}_{\text{Diagonal matrix}}\right](BSA) \tag{5.9}$$

$$+\,(-\Lambda_B - \Lambda_S - \Xi_B - \Xi_S - \Upsilon_{BS})\,(BSA).$$

$\mathbb{K}_{BJ(L)}(BSA)$ is then a block diagonal matrix with blocks of size $N_B \cdot N_S$. Clearly, we can again define also other five versions of this preconditioner ($\mathbb{K}_{BJ(L)}(BAS)$, $\mathbb{K}_{BJ(L)}(SBA)$, $\mathbb{K}_{BJ(L)}(SAB)$, $\mathbb{K}_{BJ(L)}(ASB)$ and $\mathbb{K}_{BJ(L)}(ABS)$) changing the order of Kronecker products by which the matrix $\mathbb{A}_{\mathcal{P}}$ is formed.

Figure 5.16 describes the structure of the block Jacobi preconditioners defined above hierarchically. Diagonal blocks are banded symmetric matrices, so com-

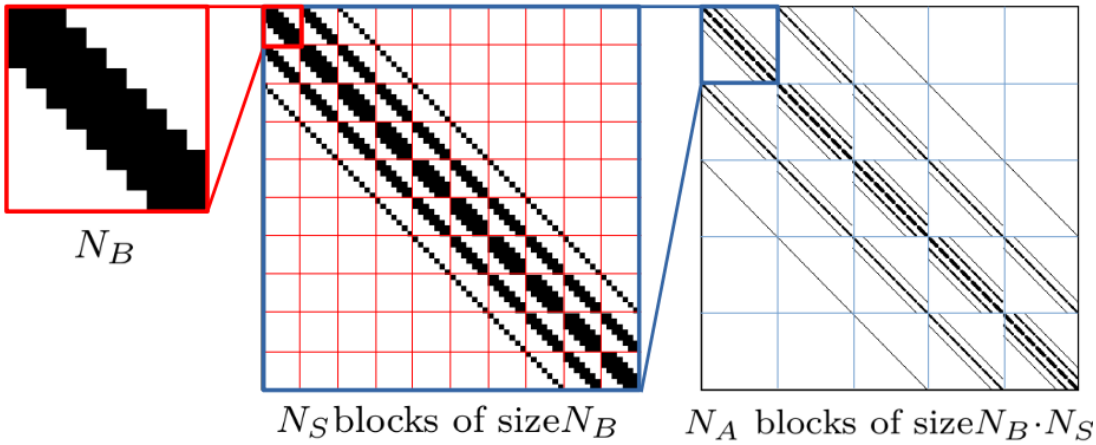


Figure 5.16: Figure shows the structure of matrix $\mathbb{A}_{\mathcal{P}}(BSA)$ (on the right), which is composed of $N_A \times N_A$ blocks of size $N_S \cdot N_B$ (in the middle). Each block is then composed of $N_S \times N_S$ blocks of size N_B (on the left).

puting their LDL^T decomposition is fairly cheap. Advantage of this approach is, that diagonal blocks are not coupled and calculating their decomposition can be easily parallelized. In the following sections we test our defined preconditioners on our models.

Model A

Let us firstly look at the results obtained for model A. We have tested the block Jacobi preconditioners for two Krylov subspace methods - COCG and GMRES. In the Figure 5.17 we can see number of iterations as a function of ε for first six block Jacobi preconditioners that are created using small blocks. We can see, that for method COCG, number of iterations decreases on average three or four times. The best result was achieved for preconditioner $\mathbb{K}_{BJ(S)}(ABS)$ and $\mathbb{K}_{BJ(S)}(ASB)$, both of which reduced the number of iterations more than five times. For GMRES the improvement in number of iterations is even better. As with the previous method, the greatest improvement in the number of iterations is achieved by using the same preconditioners $\mathbb{K}_{BJ(S)}(ABS)$ and $\mathbb{K}_{BJ(S)}(ASB)$. Now let us think about the reasons why some preconditioners work better. Let

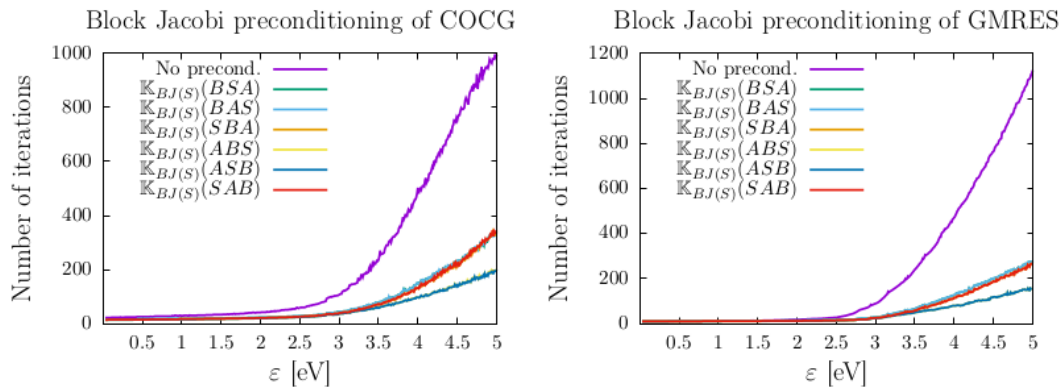


Figure 5.17: Plots of number of iterations as a function of electron energy ε for different block Jacobi preconditioning matrices with small blocks.

us use two examples to remind how the individual small block preconditioners actually differ. Preconditioners $\mathbb{K}_{BJ(S)}(BSA)$ and $\mathbb{K}_{BJ(S)}(ABS)$ are defined

$$\mathbb{K}_{BJ(S)}(BSA) = \left(\underbrace{\left[\mathbb{E}_{\mathcal{P}} - \mathbb{H}_{0,\mathcal{P}} - \Xi_S^{\mathrm{D}} - \Xi_A^{\mathrm{D}} - \epsilon_{\mathrm{d}}\mathbf{I} - \mathbb{F}_{\mathcal{P}}\right]}_{\text{Diagonal matrix}} - \Lambda_B - \Xi_B \right)(BSA), \tag{5.10}$$

$$\mathbb{K}_{BJ(S)}(ABS) = \left(\underbrace{\left[\mathbb{E}_{\mathcal{P}} - \mathbb{H}_{0,\mathcal{P}} - \Xi_S^{\mathrm{D}} - \Xi_B^{\mathrm{D}} - \epsilon_{\mathrm{d}}\mathbf{I} - \mathbb{F}_{\mathcal{P}}\right]}_{\text{Diagonal matrix}} - \Lambda_A - \Xi_A \right)(ABS). \tag{5.11}$$

It is clear, that both mentioned preconditioners have the same main diagonal and are formed by five diagonal blocks. Other diagonals in these blocks are determined by the matrices $\Lambda_B(BSA) - \Xi_B(BSA)$ in case of $\mathbb{K}_{BJ(S)}(BSA)$ and $\Lambda_A(ABS) - \Xi_A(ABS)$ in case of $\mathbb{K}_{BJ(S)}(ABS)$. So our idea is that the preconditioner $\mathbb{K}_{BJ(S)}(ABS)$ works better than the preconditioner $\mathbb{K}_{BJ(S)}(BSA)$ because the parameter M_{AA} in model A has a higher value than the parameter M_{BB}. This causes, that the matrix $\Xi_A(ABS)$ has significantly larger norm than matrix $\Xi_B(BSA)$ and therefore to put it a bit naively, one can say that matrix $\mathbb{K}_{BJ(S)}(ABS)$ approximates matrix $\mathbb{A}_{\mathcal{P}}$ better because it contains a more fundamental term. We can also notice that the numers of iterations for preconditioners that have the same vibrational dimension in the first place (for example matrices $\mathbb{K}_{BJ(S)}(ABS)$ and $\mathbb{K}_{BJ(S)}(ASB)$ have 'A' in the first place) are not different. Although these matrices have a different structure (they are rearranged differently) they contain the same amount of information relative to the rest of the matrix.

Figure 5.18 displays plots of number of iterations of COCG and GMRES using block Jacobi preconditioner with large blocks. As we expected, the block Jacobi preconditioning with large blocks reduces the number of iterations more than the block Jacobi preconditioning with small blocks, because the preconditioning matrix approximates the original system of linear equations better. The greatest improvement in the number of iterations is for both iterative method achieved by using four preconditioners $\mathbb{K}_{BJ(L)}(BAS)$, $\mathbb{K}_{BJ(L)}(ABS)$, $\mathbb{K}_{BJ(L)}(ASB)$ and

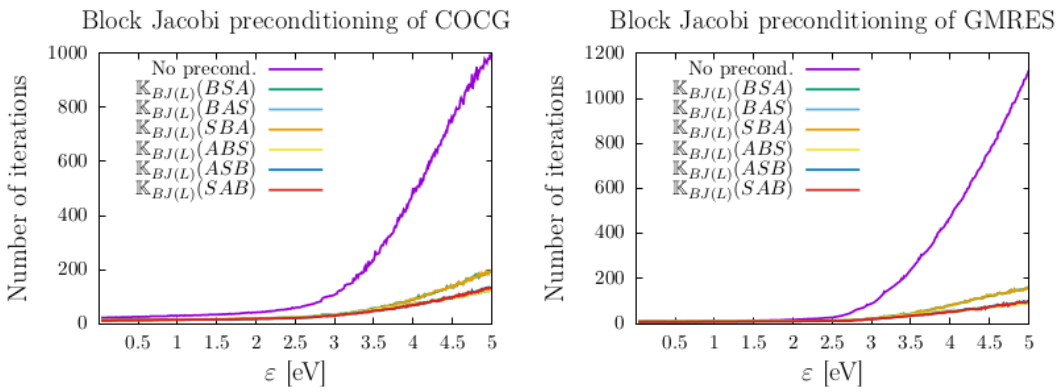


Figure 5.18: Plots of number of iterations as a function of electron energy ε for different block Jacobi preconditioning matrices with large blocks for model A.

$\mathbb{K}_{BJ(L)}(SAB)$. The remaining two methods are less effective. Let us comment on why we think it turned out that way, although in this case the situation is no longer so transparent. In this case, the preconditioner always corresponds to a combination of two vibrational dimensions, for example $\mathbb{K}_{BJ(L)}(BAS)$ containes the terms corresponding to 'B' and 'A'. The argument why some methods work better than others is based on the same reasoning here as in the previous case, only we have to take into account combinations of multiple terms. We can see that in the case of model A, the methods $\mathbb{K}_{BJ(L)}(BSA)$ and $\mathbb{K}_{BJ(L)}(SBA)$ that include dimensions 'B' and 'S' only performed the worst. However, this result is consistent with what we observed with small-block preconditioning.

In the table 5.4 we can see measured times for COCG and GMRES with different block Jacobi preconditioners. We see that for preconditions using small

	Time (small blocks) [s]		**Time (large blocks) [s]**	
	COCG	GMRES	COCG	GMRES
No preconditioning	254.06	7601.10	254.06	7601.10
BSA	85.55	177.82	98.61	139.83
BAS	85.52	178.38	83.52	102.58
SBA	71.68	166.19	90.63	131.75
ABS	58.49	108.24	71.96	95.33
ASB	45.89	86.21	48.18	71.78
SAB	65.42	177.31	45.56	70.30

Table 5.4: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with block Jacobi preconditioning needed for construction of 2D electron energy-loss spectrum for model A.

blocks, the order of the measured times coincides with what we would expect from the number of iterations. The Figure 5.19 shows the measured times for model A for different block diagonal preconditioniners in comparison to the methods without preconditioning. It means, that the fastest is preconditioning $\mathbb{K}_{BJ(S)}(ASB)$ that it speeds up the calculation more than five times. On the other hand, for GMRES, it is 88 times faster to use preconditioning matrix $\mathbb{K}_{BJ(S)}(ASB)$ in comparison to iterations without preconditioning.

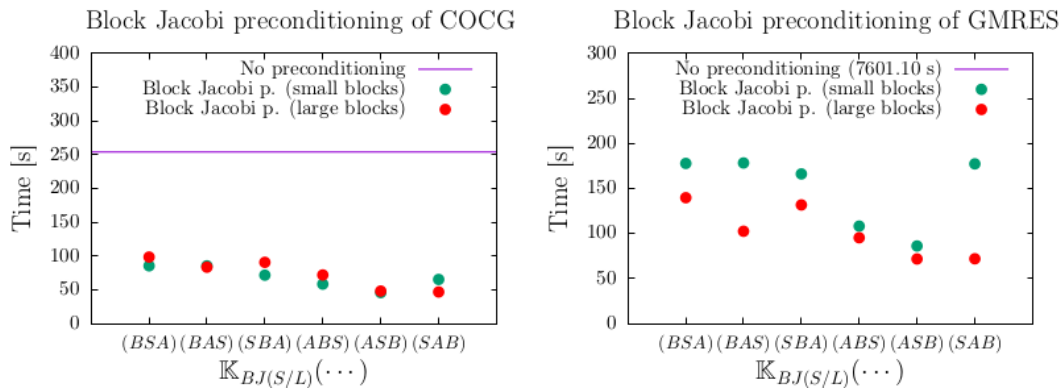


Figure 5.19: Plots of the measured times for different block Jacobi preconditioning matrices for model A.

The situation is more complicated with the second type of preconditioning which uses large blocks. In this case, the order of number of iterations for different methods does not correspond to measured times. The reason is, that size of blocks is different for every order of Kronecker products (this was true also for small blocks, but these were always five diagonal matrices and finding their decompositions was almost equaly difficult). From the time point of view, matrices $\mathbb{K}_{BJ(L)}(ASB)$ and $\mathbb{K}_{BJ(L)}(SAB)$ proved to be the most effective preconditioners for both methods COCG and GMRES. The reason why the computations in this case took less time than the other two preconditioners ($\mathbb{K}_{BJ(L)}(BAS)$ and $\mathbb{K}_{BJ(L)}(ABS)$), which use the same number of iterations, is that matrices $\mathbb{K}_{BJ(L)}(ASB)$ and $\mathbb{K}_{BJ(L)}(SAB)$ contain smaller blocks that need to be decomposed. Figure 5.20 illustrates how block sizes differ for different orders of Kronecker products.

Model B

We will now turn our attention to model B. In the Figure 5.21 we can see number of iterations as a function of ε for first six block Jacobi preconditioners that are created using small blocks. We see in the Figure 5.21, that block Jacobi preconditioners with small blocks do not always work efficiently with COCG. For some of them the number of iterations (especially for large energies ε) overcomes the number of iterations of COCG without preconditioning. Since this happens only on a small interval of energies, it does not necesarilly have to mean that the preconditioned methods will be less efficient in terms of time. The two exceptions are matrices $\mathbb{K}_{BJ(S)}(SBA)$ and $\mathbb{K}_{BJ(S)}(SAB)$, both of which reduce number of iterations for large energies approximately three times. The same preconditioners together with GMRES work even better. The best are again matrices $\mathbb{K}_{BJ(S)}(SBA)$ and $\mathbb{K}_{BJ(S)}(SAB)$, both of which use eight times less iterations than GMRES without preconditioning. It is clear that the vibrational dimension 'S' is the most important in the case of model B, although, similarly to model A, it holds that M_{AA} has a higher value than the parameter M_{AA}. In this case, however, the paremeter $\lambda_S = 0.5$ which strongly exceeds the value $\lambda_A = 0$ and this also apparently places the 'S' dimension in the role of the most important term.

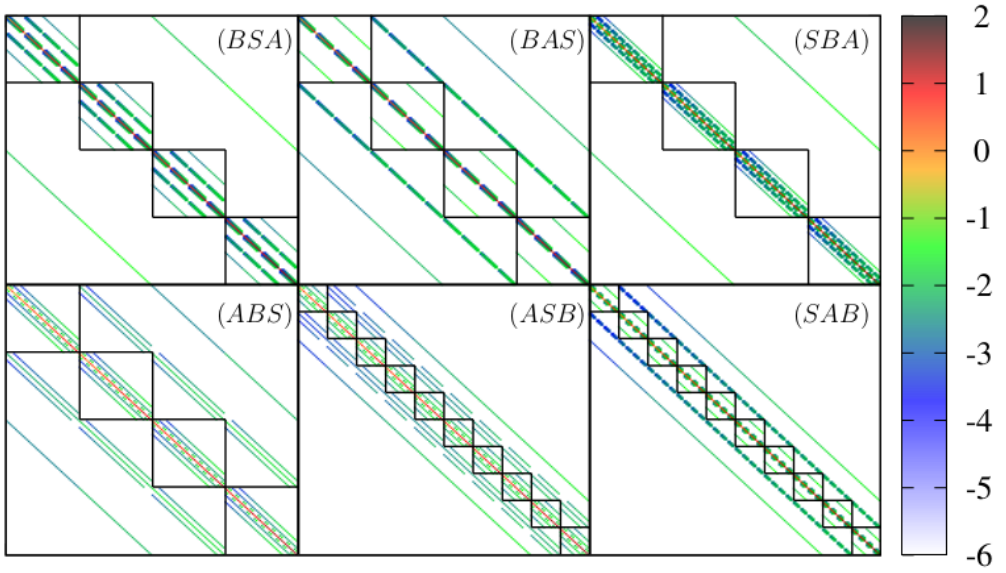


Figure 5.20: Figure shows an illustration of different types of block diagonal preconditioning matrices for model A. Ploted values of nonzero elements in matrices are logarithms of magnitudes of elements in matrix $\mathbb{A}_{\mathcal{P}}$.

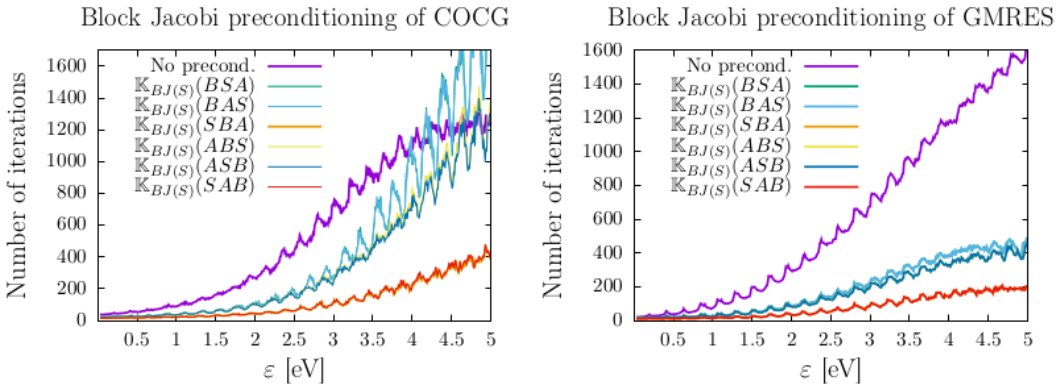


Figure 5.21: Plots of number of iterations as a function of electron energy ε for different block Jacobi preconditioning matrices with small blocks for model B.

Figure 5.22 shows number of iterations for COCG and GMRES with block Jacobi preconditioners with large blocks. As far as the number of iterations is concerned, in this case matrices $\mathbb{K}_{BJ(L)}(ASB)$ and $\mathbb{K}_{BJ(L)}(SAB)$ clearly won for both COCG and GMRES methods. In case of the GMRES method, both mentioned matrices reduce number of iterations of GMRES almost thirty times, which means that the methods usually converge in a few dozen iterations. This result was to be expected, as 'A' and 'S' is clearly the most striking combination of terms in model B.

In the table 5.5 we can see measured times for COCG and GMRES with different block Jacobi preconditioning matrices. The Figure 5.23 shows the measured times for model B for different block diagonal preconditioniners in comparison to the methods without preconditioning. We see that in terms of time, all tested block diagonal preconditioning techniques were advantageous for the

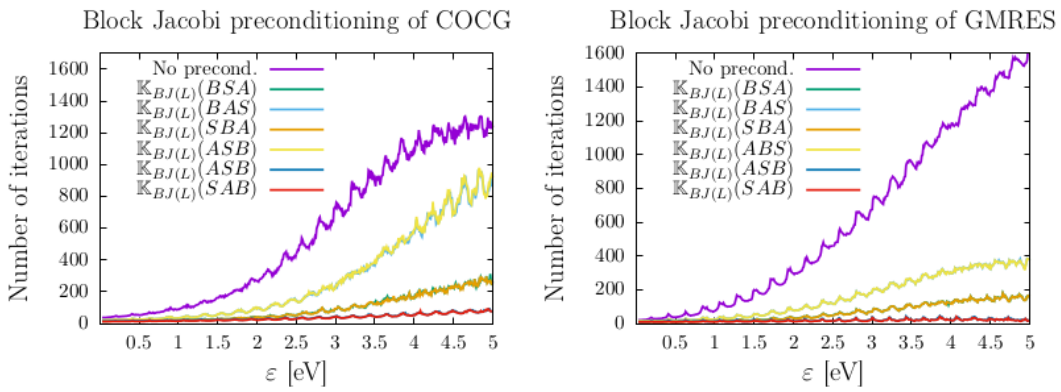


Figure 5.22: Plots of number of iterations as a function of electron energy ε for different block Jacobi preconditioning matrices with large blocks for model B.

	Time (small blocks) [s]		Time (large blocks) [s]	
	COCG	GMRES	COCG	GMRES
No preconditioning	758.59	75495.15	758.59	75495.15
BSA	402.17	1165.40	165.95	210.24
BAS	402.37	1184.14	455.93	836.07
SBA	115.99	190.69	150.39	199.63
ABS	313.57	996.93	411.22	817.87
ASB	292.79	879.30	47.40	55.10
SAB	109.18	200.52	46.07	54.96

Table 5.5: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with block Jacobi preconditioning needed for construction of 2D electron energy-loss spectrum for model B.

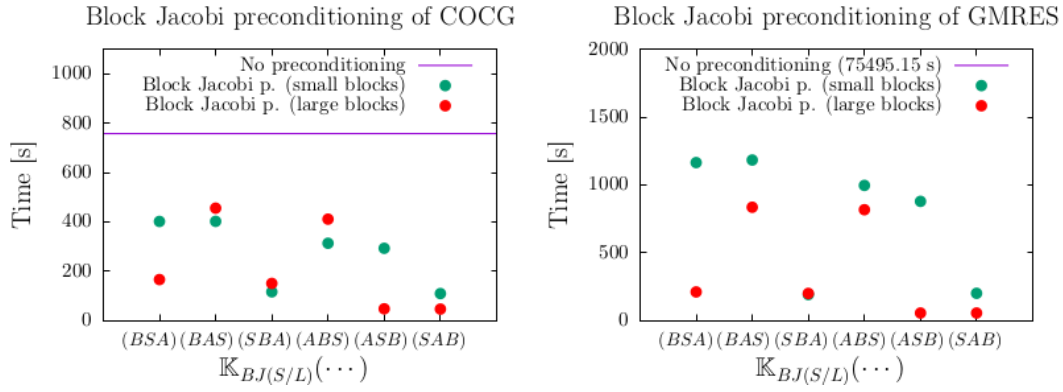


Figure 5.23: Plots of the measured times for different block Jacobi preconditioning matrices for model B.

COCG method. The best result is achieved for matrix $\mathbb{K}_{BJ(S)}(SAB)$, which shortens calculation time more than six times. However, we can see that matrix $\mathbb{K}_{BJ(S)}(SBA)$ did almost as well. Similarly it is clear, that any block Jacobi preconditioning is worth using for GMRES. As for preconditioning with small blocks, it is best to use the matrix $\mathbb{K}_{BJ(S)}(SBA)$, which shortens the computation time by 397 times.

For preconditioning with large blocks, the results are even better. In this case, matrix $\mathbb{K}_{BJ(L)}(SAB)$ proved to be the most effective preconditioner. Method COCG was speeded up sixteen times using this matrix and the method GMRES was even a 1372 times faster than without preconditioning. However, we can see that the matrix $\mathbb{K}_{BJ(S)}(ASB)$ did almost as well. Figure 5.24 again illustrates how block sizes differ for different orders of Kronecker products.

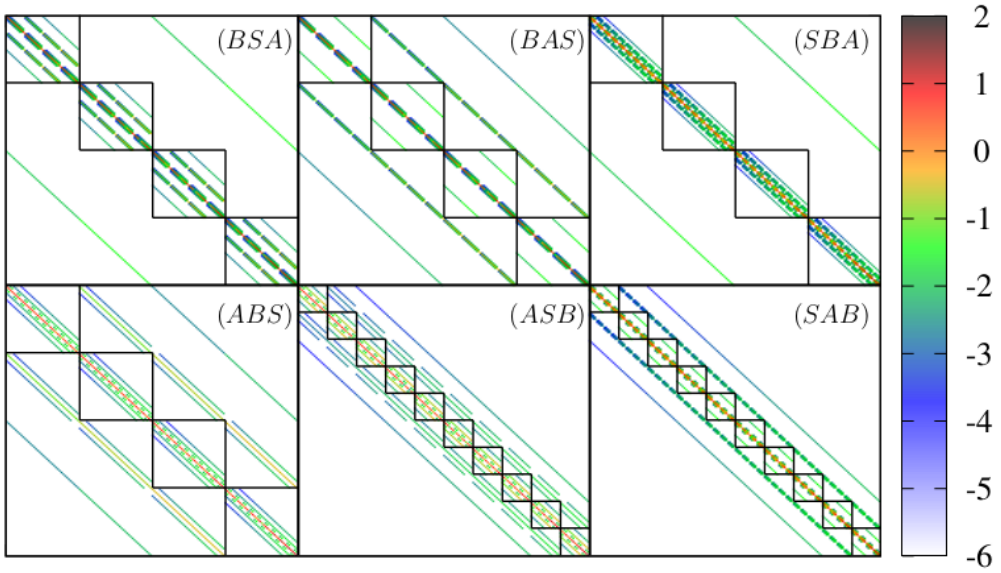


Figure 5.24: Figure shows an illustration of different types of block diagonal preconditioning matrices for model B. Ploted values of nonzero elements in matrices are logarithms of magnitudes of elements in matrix $\mathbb{A}_{\mathcal{P}}$.

Model C

Finally, let us focus on the results for model C. We have tested the block Jacobi preconditioners for two Krylov subspace methods - COCG and GMRES. In the Figure 5.25 we can see number of iterations as a function of ε for first six block Jacobi preconditioners that are created using small blocks. Except of preconditioning matrices $\mathbb{K}_{BJ(S)}(SBA)$ and $\mathbb{K}_{BJ(S)}(SAB)$ all preconditioners increase number of iterations of COCG many times (for some of the energies), so from the point of view of number of iterations it looks like it is more eficient to use the method without preconditioning. However, this result is expected and very natural given the parameter values in model C. The vibrational dimension 'S' is the only significant member of the model C. For GMRES, the situation is a bit more optimistic. All the preconditioners in this case considerably reduce the number of iterations. Preconditioning matrices $\mathbb{K}_{BJ(S)}(SBA)$ and $\mathbb{K}_{BJ(S)}(SAB)$ seem to work the best together with GMRES. Figure 5.26 shows plots of number of iterations for block Jacobi preconditioners with large blocks for model C. Unfortunatelly, some of the methods can not be used with COCG, because they again increase the number of iterations. However, there are also four preconditioners $\mathbb{K}_{BJ(L)}(BSA)$, $\mathbb{K}_{BJ(L)}(SBA)$, $\mathbb{K}_{BJ(L)}(ASB)$ and $\mathbb{K}_{BJ(L)}(SAB)$ that reduce the

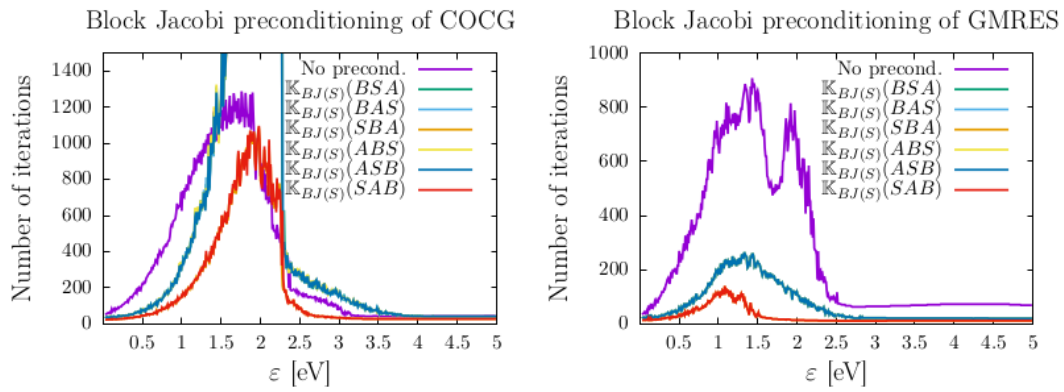


Figure 5.25: Plots of number of iterations as a function of electron energy ε for different block Jacobi preconditioning matrices for model C.

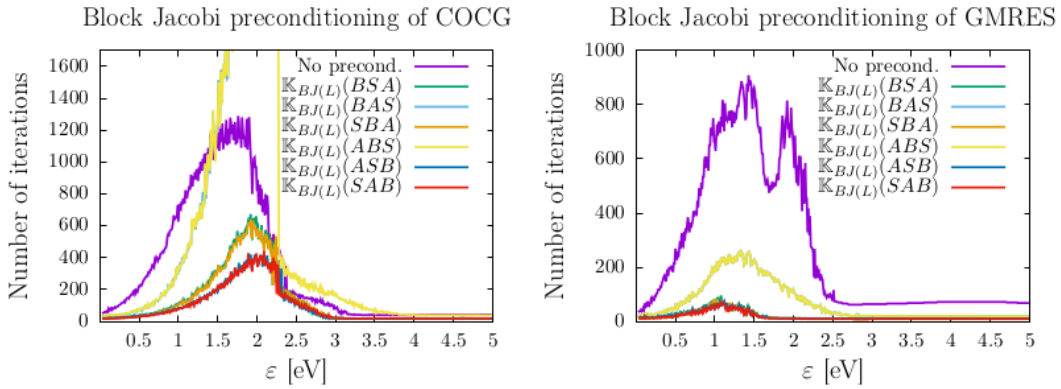


Figure 5.26: Plots of number of iterations as a function of electron energy ε for different block Jacobi preconditioning matrices for model C.

number of iterations. We can see that it is precisely those matrices that contain the vibrational dimension 'S', with the fact that those that contain it in the first place perform significantly better. We see that for GMRES, the situation is quite similar.

In the table 5.6 we can see measured times for COCG and GMRES with different block Jacobi preconditioning matrices. The Figure 5.27 shows the measured

	Time (small blocks) [s]		**Time (large blocks) [s]**	
	COCG	GMRES	COCG	GMRES
No preconditioning	988.60	9374.98	988.60	9374.98
BSA	1104.73	419.90	478.17	150.91
BAS	1369.39	655.88	6780.34	1597.26
SBA	432.94	130.16	397.83	118.10
ABS	1438.34	723.85	7135.06	1637.52
ASB	1222.26	495.34	477.29	204.55
SAB	470.52	180.66	367.46	153.27

Table 5.6: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with block Jacobi preconditioning needed for construction of 2D electron energy-loss spectrum for model C.

times for model C for different block diagonal preconditioniners in comparison to the methods without preconditioning. Unlike other models, we cannot claim that

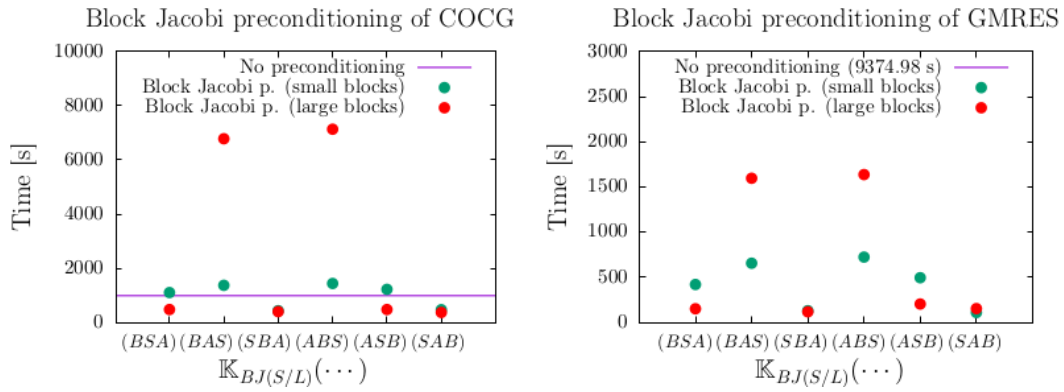


Figure 5.27: Plots of the measured times for different block Jacobi preconditioning matrices for model C.

using the block Jacobi preconditioning with COCG is beneficial for model C in any way. We obtained the best results for matrices $\mathbb{K}_{BJ(L)}(SBA)$ and $\mathbb{K}_{BJ(L)}(SAB)$. On the other hand, for GMRES the situation is different, because all preconditioners tested work faster with GMRES than GMRES alone. From the point of view of computation time, the best result was obtained for $\mathbb{K}_{BJ(L)}(SBA)$ for which the time was reduced almost eighty times for GMRES and was significantly faster than using $\mathbb{K}_{BJ(L)}(SAB)$. This may be because matrix $\mathbb{K}_{BJ(L)}(SBA)$ contains smaller blocks than matrix $\mathbb{K}_{BJ(L)}(SAB)$. Figure 5.28 again illustrates how block sizes differ for different orders of Kronecker products.

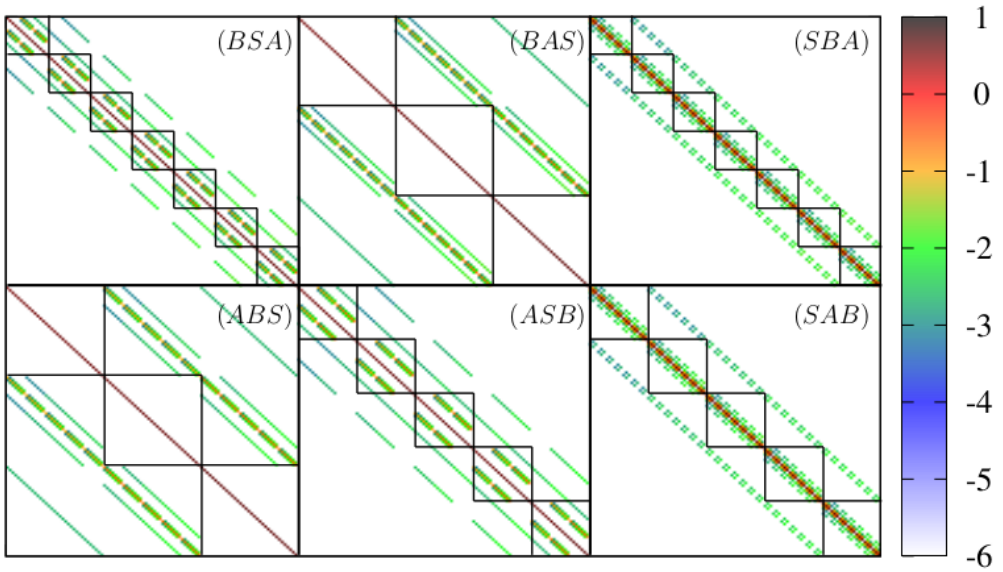


Figure 5.28: Figure shows an illustration of different types of block diagonal preconditioning matrices for model C. Ploted values of nonzero elements in matrices are logarithms of magnitudes of elements in matrix $\mathbb{A}_{\mathcal{P}}$.

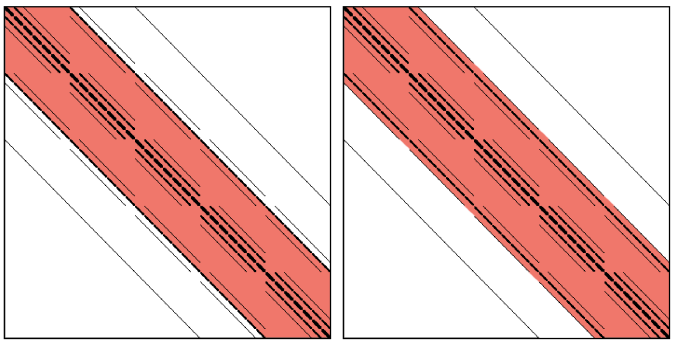
5.3 Preconditioning with banded matrix

In this part we will define banded preconditioning matrices. The number of diagonals we consider is motivated by the structure of the matrix. The reason we want to test preconditions using banded matrices is that they can approximate matrix $\mathbb{A}_{\mathcal{P}}$ even better than block diagonal matrices which we considered before. On the other hand, the construction and use of these preconditions can be much more time complex (in comparison to block diagonal matrices) even though it is generally quite cheap to decompose them.

First six preconditioners are banded matrices with 'small' number of diagonals (Figure 5.29 on the left). Let us define preconditioning matrix $\mathbb{K}_{B(2)}(BSA)$ in the following way:

$$\left(\mathbb{K}_{B(2)}(BSA)\right)_{i,j} = \begin{cases} (\mathbb{A}(BSA)_{\mathcal{P}})_{i,j} & \text{for } |i-j| \leq N_B \cdot N_S + 2 \\ 0 & \text{for } |i-j| > N_B \cdot N_S + 2 \end{cases} \tag{5.12}$$

Letters in parentheses in $\mathbb{A}_{\mathcal{P}}(BSA)$ stand for indication of a specific order of factors in Kronecker product which defines matrix $\mathbb{A}_{\mathcal{P}}$. Analogicaly, we can define other five preconditioning matrices $\mathbb{K}_{B(2)}(BAS)$, $\mathbb{K}_{B(2)}(SBA)$, $\mathbb{K}_{B(2)}(SAB)$, $\mathbb{K}_{B(2)}(ASB)$ and $\mathbb{K}_{B(2)}(ABS)$, which are also symmetric banded matrices. They are formed similarly as $\mathbb{K}_{B(2)}(BSA)$ only using a different order of Kronecker products of matrices. The structure of just defined matrices is displayed in the Figure 5.29.



(a) $\mathbb{K}_{B(2)}(BSA)$ (b) $\mathbb{K}_{B(3)}(BSA)$

Figure 5.29: Structure of banded preconditioning matrices.

Secondly we define other six preconditioning matrices with different numbers of nonzero diagonals.

$$\left(\mathbb{K}_{B(3)}(BSA)\right)_{i,j} = \begin{cases} (\mathbb{A}(BSA)_{\mathcal{P}})_{i,j} & \text{for } |i-j| \leq N_B \cdot (N_S + 2) \\ 0 & \text{for } |i-j| > N_B \cdot (N_S + 2) \end{cases} \tag{5.13}$$

All the banded preconditioners that we consider are symmetric matrices, so we can find their LDL^T decomposition. Unlike block diagonal matrices, however, we have to look for the decomposition of the whole matrix at once. This can make the construction of the precondition more complex and cause a calculation slowdown. In the following sections we test our defined preconditioners on our models.

Model A

As usually, let us firstly focus on the results for model A. We have tested different banded preconditioners for two Krylov subspace methods - COCG and GMRES. In the Figure 5.30 we can see number of iterations as a function of ε for first six banded preconditioners. As we can see from the graphs of the number of

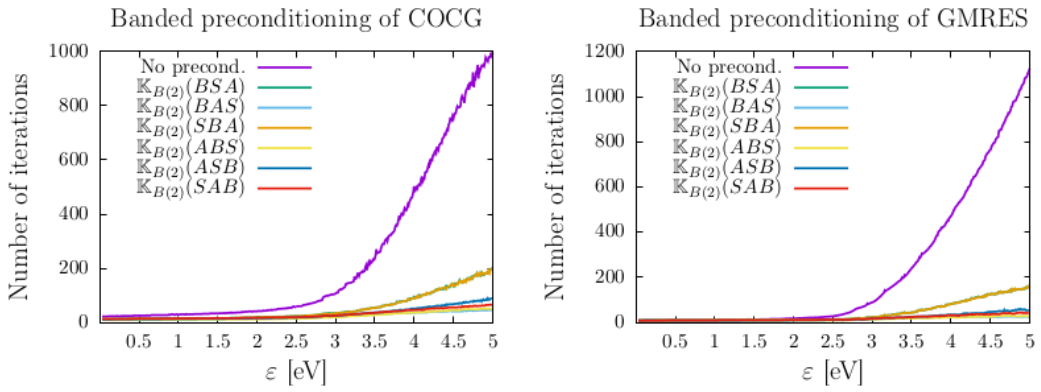


Figure 5.30: Plots of number of iterations as a function of electron energy ε for different banded preconditioning matrices with 'small' number of diagonals.

iterations, the band preconditioning reduces the number of iterations of both methods more than twice in comparison to the block diagonal preconditioning techniques and approximately ten times in comparison to the methods without preconditioning. This result is in line with expectations, as the band matrix defined earlier approximates the original system better than a diagonal of block diagonal matrix. In the next Figure 5.31 we can see number of iterations as a function of ε for other six banded preconditioners with more nonzero diagonals. As we can see, the numbers of iterations are a bit smaller in comparison to

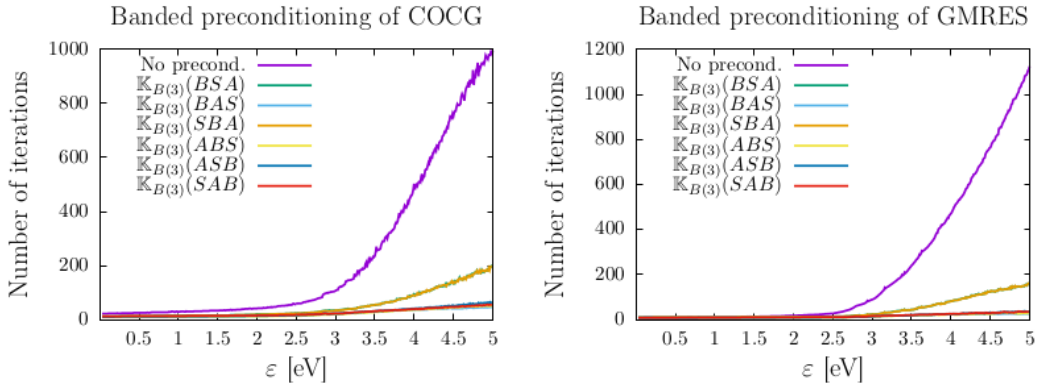


Figure 5.31: Plots of number of iterations as a function of electron energy ε for different banded preconditioning matrices with 'large' number of nonzero diagonals.

the previous preconditioning using a matrix with a lower number of diagonals. However, the difference is not significant.

In the table 5.7 we can see measured times for COCG and GMRES with different banded preconditioners. The Figure 5.32 shows the measured times for model

	Time (small) [s]		Time (large) [s]	
	COCG	GMRES	COCG	GMRES
No preconditioning	254.06	7601.10	254.06	7601.10
BSA	334.77	365.90	408.33	431.72
BAS	257.74	270.50	324.50	327.20
SBA	334.05	365.54	362.36	388.13
ABS	255.23	266.74	277.66	285.22
ASB	127.06	142.17	131.64	142.50
SAB	122.10	138.22	129.41	142.90

Table 5.7: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with banded preconditioning needed for construction of 2D electron energy-loss spectrum.

A for different banded preconditioniners in comparison to the methods without preconditioning. Let us first comment on the results for the COCG method. We

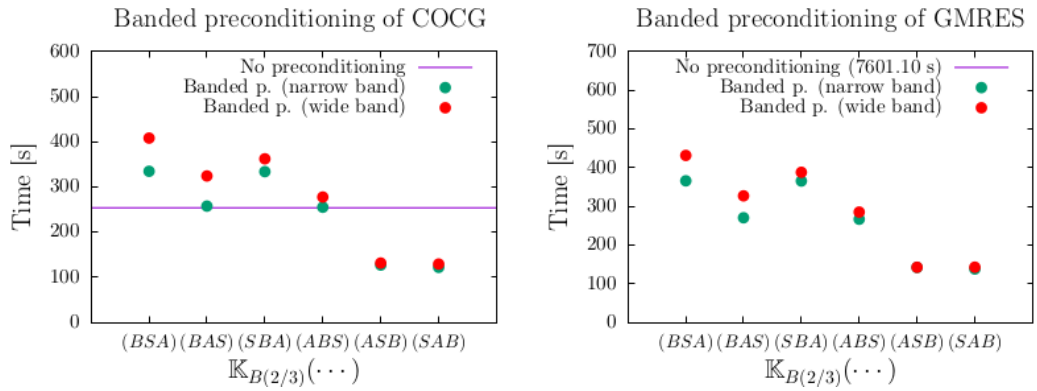


Figure 5.32: Plots of the measured times for different banded preconditioning matrices.

obtained the best result in terms of time for preconditioners $\mathbb{K}_{B(2)}(SAB)$ and $\mathbb{K}_{B(2)}(ASB)$. These results are comparable to $\mathbb{K}_{B(3)}(SAB)$ and $\mathbb{K}_{B(3)}(ASB)$. In general, however, we can say that block diagonal preconditioning is more efficient. The situation is much better with the GMRES method. In this case, it is the fastest to use the preconditioning matrix $\mathbb{K}_{B(2)}(SAB)$, which is more than 55 times more efficient than GMRES without preconditioning. Generally we can say, that it is worth using any of the tested preconditioning matrices with GMRES method. Nevertheless, none of them is better than the best result we obtained for block Jacobi preconditioning.

Model B

Let us now see the results for model B. In the Figure 5.33 we can see number of iterations as a function of ε for first six banded preconditioners. As we can see from the graphs of the number of iterations, the band preconditioning reduces the number of iterations of both methods more than the block diagonal preconditioning. Especially for large energies, the number of iterations (for both COCG

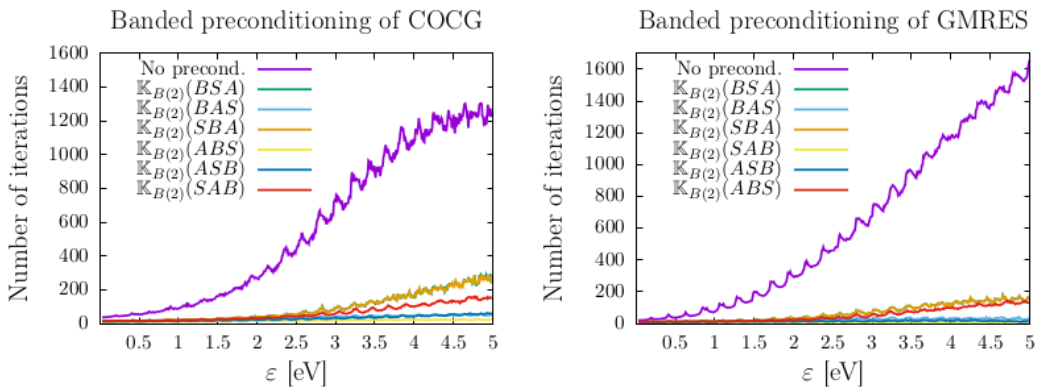


Figure 5.33: Plots of number of iterations as a function of electron energy ε for narrower banded preconditioning matrices.

and GMRES) with the use of banded preconditioning is more than three times smaller in comparison to the block Jacobi preconditioning. This result is again in line with expectations, as the band matrix defined earlier approximates the original system better than a diagonal of block diagonal matrix. In the next Figure 5.34 we can see number of iterations as a function of ε for other six banded preconditioners with more nonzero diagonals. The results are similar to the previous

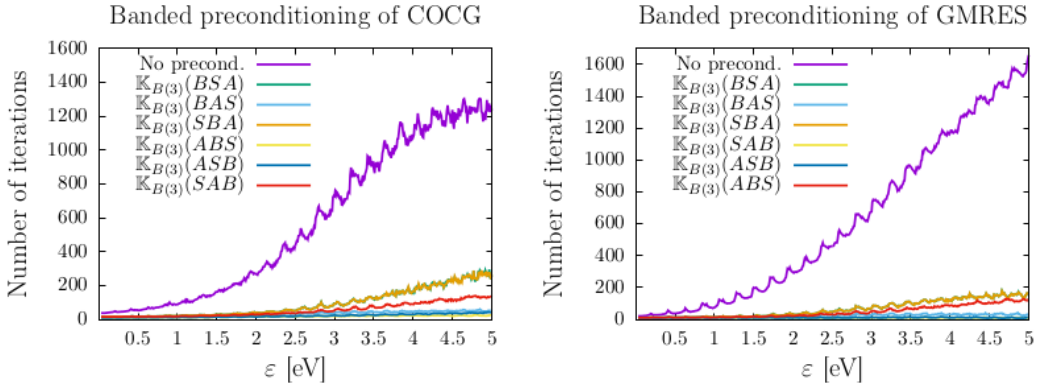


Figure 5.34: Plots of number of iterations as a function of electron energy ε for wider banded preconditioning matrices.

preconditioning using a matrix with a lower number of diagonals.

In the table 5.8 we can see measured times for COCG and GMRES with different banded preconditioning matrices. Let us first look at the results for the COCG method. We see that the calculation takes the shortest time when using the preconditioning of the matrix $\mathbb{K}_{B(3)}(ASB)$ (and also $\mathbb{K}_{B(2)}(ASB)$), although the lowest number of iterations is reached when the preconditioning of the matrix $\mathbb{K}_{B(2)}(ABS)$ (or $\mathbb{K}_{B(3)}(ABS)$) is used. This phenomenon is caused by the different number of matrix decomposition operations that arose using different orders of Kronecker products in matrix formation. A similar situation can be observed with the GMRES method. Note that in every case we used banded preconditioner for model B we have achieved a better time than without the use of preconditioning. The Figure 5.35 shows the measured times for model B for different banded preconditioniners in comparison to the methods without preconditioning. Unfor-

	Time (small) [s]		Time (large) [s]	
	COCG	GMRES	COCG	GMRES
No preconditioning	758.59	75495.15	758.59	75495.15
BSA	470.97	458.61	548.96	534.43
BAS	303.10	295.35	355.43	355.04
SBA	488.65	483.42	483.44	488.51
ABS	256.11	254.20	267.96	273.33
ASB	136.80	133.85	126.58	135.06
SAB	195.41	227.66	188.75	230.74

Table 5.8: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with banded preconditioning needed for construction of 2D electron energy-loss spectrum.

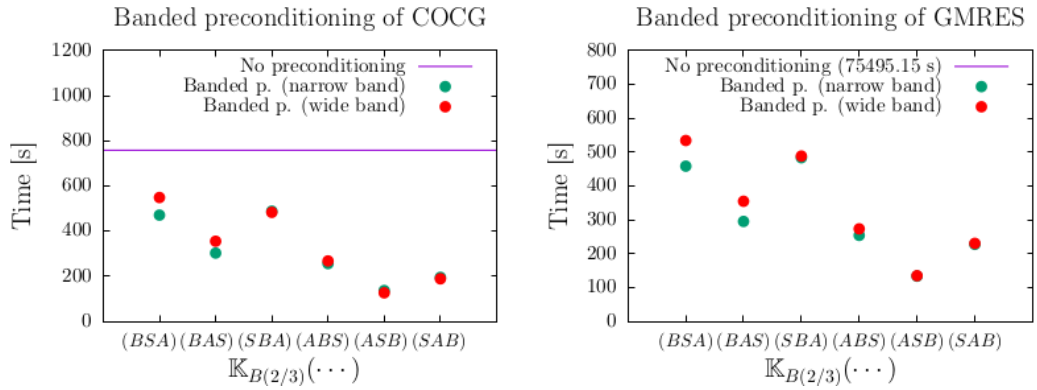


Figure 5.35: Plots of the measured times for different banded preconditioning matrices.

tunately, we were no longer able to overcome the times we achieved with block Jacobi method. Although iterative methods now converge with a lower number of iterations compared to block Jacobi method, the calculations take longer. This is due to the more complex construction of the LDL^T decomposition, which we do when creating a preconditioning. Nevetheless we can say, that (especially for GMRES) some of the techniques are worth using.

Model C

Finally, let us focus on the results for model C. We have tested different banded preconditioners for two Krylov subspace methods - COCG and GMRES. In the Figure 5.36 we can see number of iterations as a function of ε for first six narrower banded preconditioners. We can see that compared to the block Jacobi method, the number of iterations is significantly reduced for all banded preconditioners tested. This is true for both COCG and GMRES. In the Figure 5.37 we can see number of iterations as a function of ε for other six wider banded preconditioners (defined with more nonzero diagonals). Comparing results of both banded preconditioners we can see that there is a difference between the numbers of iterations. In this case, some preconditioning matrices even caused the solution to converge using only a few iterations. For instance, GMRES precondi-

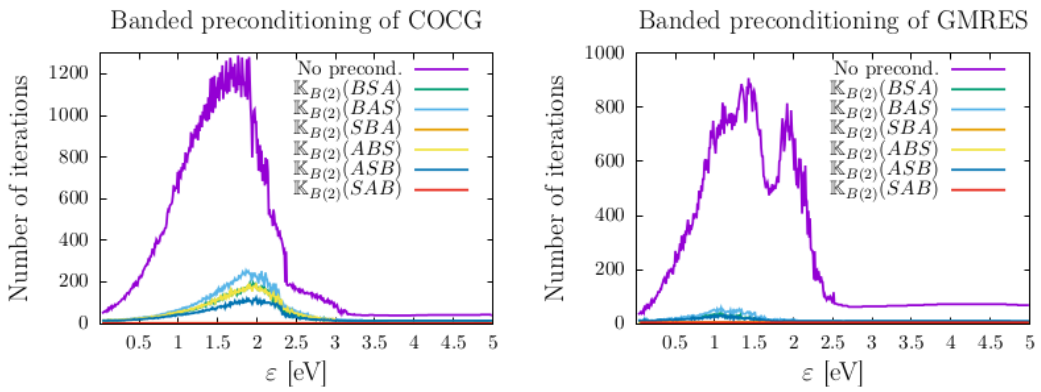


Figure 5.36: Plots of number of iterations as a function of electron energy ε for banded preconditioning matrices.

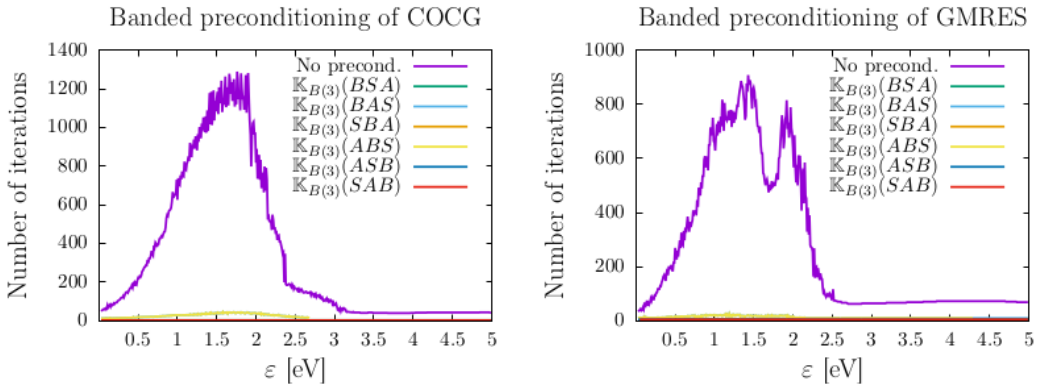


Figure 5.37: Plots of number of iterations as a function of electron energy ε for banded preconditioning matrices.

tioned with $\mathbb{K}_{B(3)}(SAB)$ converged in four iterations for every considered energy ε, but the truth is also that the zeroness of some parameters in the model C causes $\mathbb{A}_{\mathcal{P}} = \mathbb{K}_{B(3)}(SAB)$. This also applies to some other rearrangements of the preconditioning matrix. However, as usual, we are more interested in whether the preconditioning reduced the time required to solve all systems. In the table 5.9 we can see measured times for COCG and GMRES with different banded preconditioning matrices. The Figure 5.38 shows the measured times for model C for different banded preconditioniners in comparison to the methods without preconditioning. With a few exceptions, we have found that banded preconditioning is not a suitable choice for the model C. Regarding the COCG method, the best result was obtained using preconditioning $\mathbb{K}_{B(3)}(SBA)$. The same matrix is the most efficient banded preconditioner also for method GMRES. We can also see that for the preconditioners that were created from a matrix rearranged so that it has the vibrational dimension S in the last position, i. e. $\mathbb{K}_{B(3)}(BAS)$ and $\mathbb{K}_{B(3)}(ABS)$ (and also $\mathbb{K}_{B(2)}(BAS)$ and $\mathbb{K}_{B(2)}(ABS)$), the calculation times are huge. Nevertheless, for none of the iterative methods the band preconditioning exceeds the block diagonal preconditioning.

	Time (small) [s]		Time (large) [s]	
	COCG	GMRES	COCG	GMRES
No preconditioning	988.60	9374.98	988.60	9374.98
BSA	2224.55	1195.66	773.67	768.60
BAS	> 12 h	36534.55	> 12 h	35378.22
SBA	813.71	801.00	616.47	693.12
ABS	32142.85	31593.18	> 12 h	>12 h
ASB	5893.27	1453.56	1252.53	1157.25
SAB	879.92	850.15	895.71	866.75

Table 5.9: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with banded preconditioning needed for construction of 2D electron energy-loss spectrum.

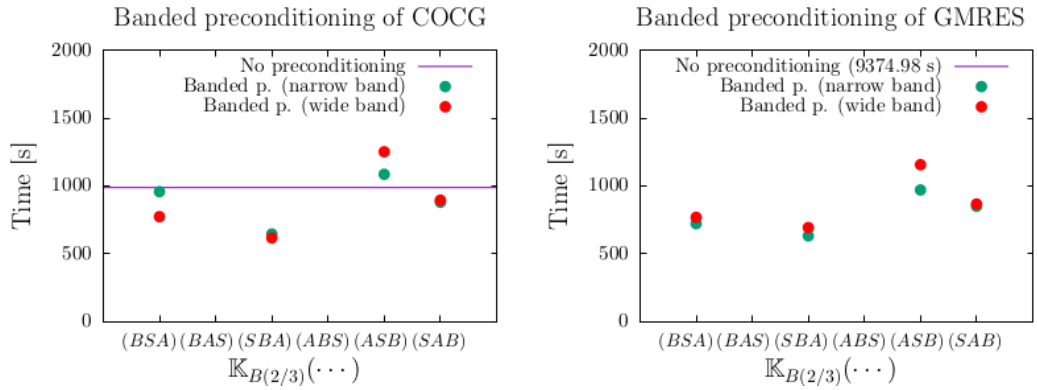


Figure 5.38: Plots of the measured times for different banded preconditioning matrices.

5.4 Incomplete factorization

In this section, we use preconditioners, that are created as an incomplete LDL^T factorization of banded matrices $\mathbb{K}_{B(3)}(\cdots)$. There are many possible fill-in strategies. Here we consider fill-in strategy that preserves structure of matrix $\mathbb{A}_{\mathcal{P}}$. It means, that we only compute $L(i,j)$ if $\mathbb{A}_{\mathcal{P}}(i,j) \neq 0$ for $i > j$. In the following parts we test our defined preconditioners on our models.

Model A

Plots of number of iterations as a function of electron energy ε for different banded preconditioning matrices with use of incomplete factorization is ploted in Figure 5.39. We see, that for method COCG, the number of iterations is decreased approximately five times in comparison to the number of iterations used without preconditioning. For GMRES, the number of iterations using predontitioning is usually six times lower.

In the table 5.10 we can see measured times for COCG and GMRES with banded preconditioner used with incomplete factorization. The Figure 5.40 shows the measured times for model A for different incomplete banded preconditioniners in comparison to the methods without preconditioning. We can see that with the

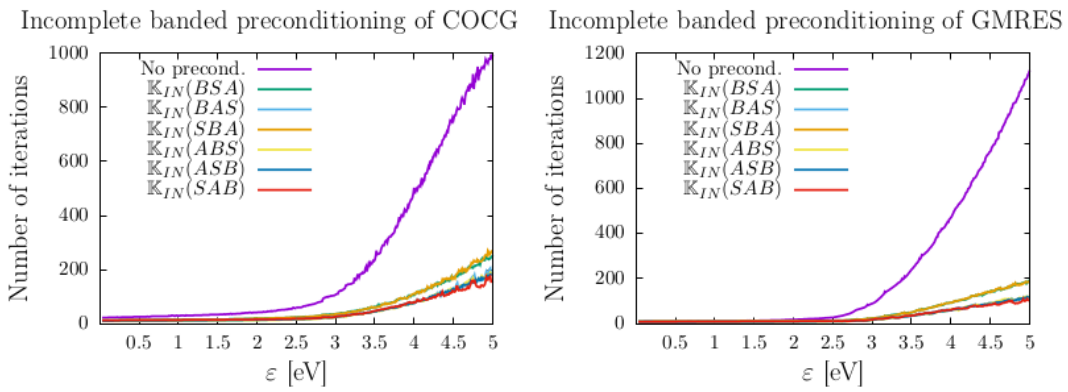


Figure 5.39: Plots of number of iterations as a function of electron energy ε for different banded preconditioning matrices with use of incomplete factorization.

	Time [s]	
	COCG	GMRES
No preconditioning	254.06	7601.10
BSA	264.91	291.09
BAS	215.09	215.34
SBA	254.39	285.70
ABS	198.49	205.99
ASB	145.47	161.62
SAB	141.42	157.28

Table 5.10: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with incomplete banded preconditioning needed for construction of 2D electron energy-loss spectrum.

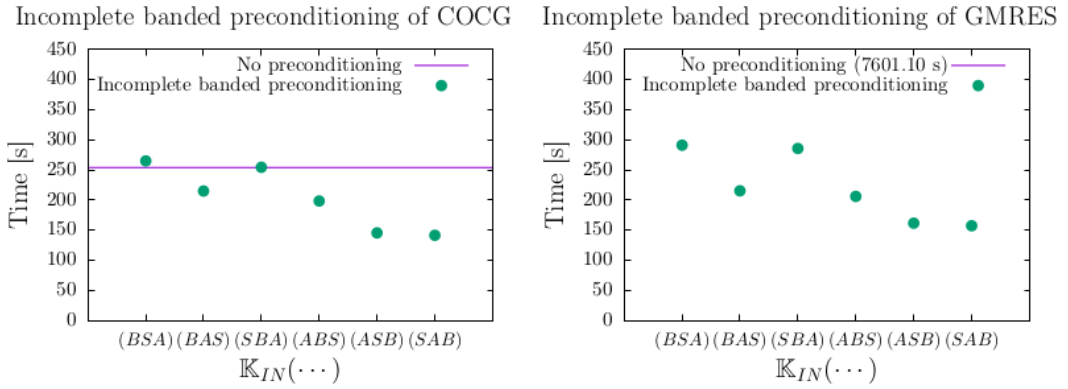


Figure 5.40: Plots of the measured times for incomplete banded preconditioner.

exception of two matrices ($\mathbb{K}_{IN}(ASB)$ and $\mathbb{K}_{IN}(SAB)$), we have improved in time compared to the banded preconditioning. On the other hand, in comparison to the block Jacobi preconditioning, the computational times are larger for the incomplete banded preconditioning.

Model B

For model B, we can see in the Figure 5.41 that used precondtioning matrices are not equally efficient for both iterative methods. On the one hand, preconditioning decreases number of iterations of GMRES aproximately four to eight times. On the other hand the same preconditioning reduces the number of iterations of COCG approximately twice.

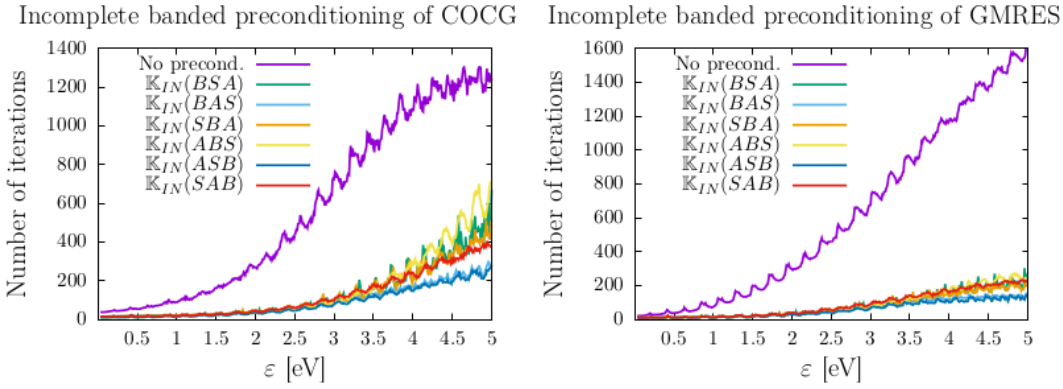


Figure 5.41: Plots of number of iterations as a function of electron energy ϵ for different banded preconditioning matrices with use of incomplete factorization.

In the table 5.11 we can see measured times for COCG and GMRES with incomplete banded preconditioners. The Figure 5.42 shows the measured times

	Time [s]	
	COCG	GMRES
No preconditioning	758.59	75495.15
BSA	493.55	472.52
BAS	366.68	361.34
SBA	453.91	438.81
ABS	522.41	506.37
ASB	222.24	240.55
SAB	299.39	389.62

Table 5.11: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with incomplete banded preconditioning needed for construction of 2D electron energy-loss spectrum.

for model B for different incomplete banded preconditioniners in comparison to the methods without preconditioning. Incomplete banded preconditioning is not a suitable choice for model B. As for the COCG method, the best result was obtained using preconditioning $\mathbb{K}_{IN}(ASB)$, but even in this case the computation was not faster than the block Jacobi preconditioned COCG. On the other hand, for GMRES method all the preconditioning matrices reduced computational time significantly. However, even here the result is not better than the one we got with the block Jacobi method.

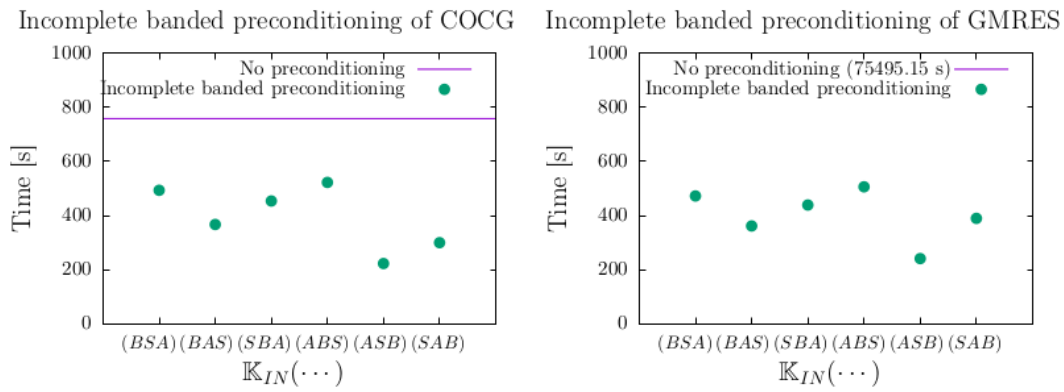


Figure 5.42: Plots of the measured times for incomplete banded preconditioner.

Model C

Finally, let us look at the results for the model C. Figure 5.43 shows plots of number of iterations for model C using preconditioners constructed by incomplete factorization. Regarding the COCG method, we see that the individual

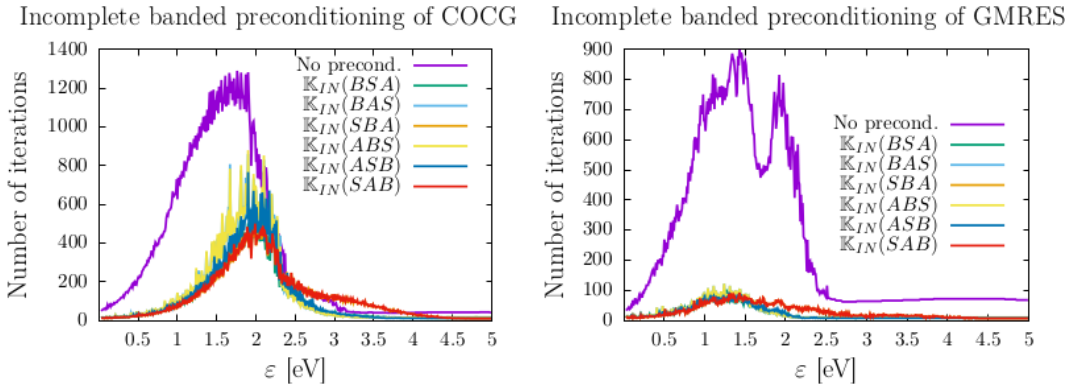


Figure 5.43: Plots of number of iterations as a function of electron energy ε for different banded preconditioning matrices with use of incomplete factorization.

preconditioners reduce the number of iterations for approximately half of the energies and increase it for the other half. On the other hand, all the preconditioners work eficiently with iterative method GMRES. All the preconditioners, although constructed using incomplete factorization, still reduce the number of iterations very significantly. As usual, the most important thing for us is the duration of the calculation. In the table 5.12 we can see measured times for COCG and GMRES with incomplete banded preconditioners. In terms of time, for the COCG method, we have only confirmed that the use of incomplete preconditioning is not advantageous for the model C. Unfortunatelly, neither of the incomplete banded preconditioners works efficiently for iterative method COCG, although the preconditioning techniques reduce number of iterations for some energies. On the contrary, all preconditions only increased the duration of the calculation. With the GMRES method, the situation is again the opposite. We see, that all the incomplete banded preconditioning methods worked here, and the best of them, matrix $\mathbb{K}_{IN}(BSA)$, reduced the time needed to solve the systems by 13 times. The Figure 5.44 shows the measured times for model C for different incomplete

	Time [s]	
	COCG	GMRES
No preconditioning	988.60	9374.98
BSA	1297.42	700.54
BAS	5641.74	1774.21
SBA	1256.90	720.92
ABS	6239.90	1887.94
ASB	1617.98	799.87
SAB	1465.36	832.36

Table 5.12: Table with measured time of solving all the linear systems with COCG and GMRES preconditioned with incomplete banded preconditioning needed for construction of 2D electron energy-loss spectrum.

banded preconditioniners in comparison to the methods without preconditioning.

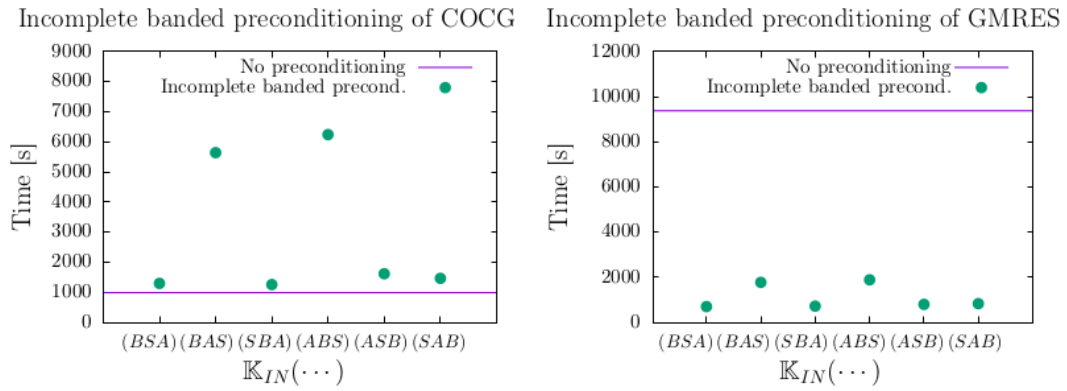


Figure 5.44: Plots of the measured times for incomplete banded preconditioner.

5.5 Splitting type methods

We have decided to try solving our systems of linear equations with some of the methods mentioned recently, even though we are not sure our system satisfies all the assumptions mentioned by authors. First and foremost, throuhout the literature is assumed that both real and imaginary parts of the system (matrices $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$) are non-singular. Although authors usually do not mention this property explicitely, they often use inverses of matrices $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$ while deriving some preconditioner of proving convergence results.

5.6 Zhang-Dai preconditioner

Firstly, we have chosen so called Zhang-Dai preconditioner. The main reason is that this preconditioner introduced by Zhang and Dai in 2015 (in Zhang and Dai [2015]) does not require too many special assumptions on matrix properties.

They consider a system

$$(\mathbb{B}_{\mathcal{P}} + i\,\mathbb{D}_{\mathcal{P}})\,(\mathbf{y} + i\,\mathbf{z}) = \mathbf{b}, \tag{5.14}$$

with $\mathbb{B}_{\mathcal{P}}$ being symmetric indefinite (and also non-singular, which is not explicitly mentioned but probably assumed) and $\mathbb{D}_{\mathcal{P}}$ being symmetric positive definite. They defined a preconditioner

$$\mathbb{R}_1(\alpha) = \frac{1}{2\alpha}\,(\alpha\mathbb{B}_{\mathcal{P}} + i\,\mathbb{I})\,(\alpha\mathbb{D}_{\mathcal{P}} + \mathbb{I})\,. \tag{5.15}$$

We have not tried this preconditioner with method COCG, because this method requires a symmetric preconditioning matrix, which $\mathbb{R}_1(\alpha)$ is not.

Model A

As usually, we firstly test model A. Figure 5.45 contains number of iterations as a function of electron energy ε for GMRES with Zhang-Dai preconditioner. We have used five different values of $\alpha \in \{2^0; 2^1; 2^2; 2^3; 2^4\} = \{1; 2; 4; 8; 16\}$.

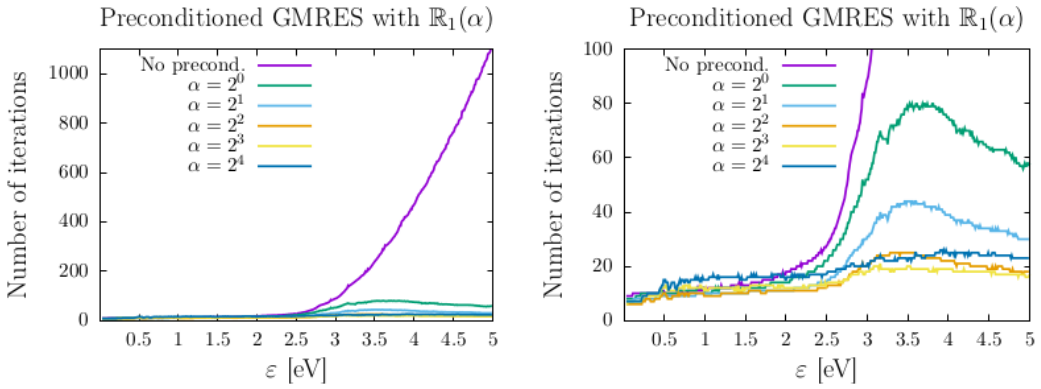


Figure 5.45: Plots of number of iterations as a function of electron energy ε for GMRES with Zhang-Dai preconditioner. The plot on the right contains the same data as plot on the left, but it has different range on vertical axis.

From the plots in Figure 5.45 we see, that using Zhang-Dai predonditioner significantly decreases the number of iterations. We can also see, that the number of iterations depends on the value of α. The best results are achieved for $\alpha = 8$. We know, that the number of iterations itself does not have to determine the efficiency of our method, we also have to take into account the computation time.

In the table 5.13 we can see measured times for GMRES with preconditioning $\mathbb{R}_1(\alpha)$ for different values of α. From this table, it is clear, that using GMRES preconditioned using Zhang-Dai method is indeed efficient, because it is (for $\alpha = 8$) more than 35 times faster thant GMRES without preconditioning. Nevertheless, this restult is still not very satisfactory, since we know, that matrix $\mathbb{R}_1(\alpha)$ has the same structure of non-zero elements as matrix $\mathbb{A}_{\mathcal{P}}$. It means, that the work we have to do in order to construct the preconditioner (mainly the LDL^T decomposition of matrix $\mathbb{B}_{\mathcal{P}}$) is always at least the same complicated as solving the original system of linear algebraic equations directly. For this reason, we have decided to modify the Zhang-Dai method by replacing the matrix $\mathbb{B}_{\mathcal{P}}$ in

$$\mathbb{R}_1(\alpha) = \frac{1}{2\alpha}\,(\alpha\mathbb{B}_{\mathcal{P}} + i\,\mathbb{I})\,(\alpha\mathbb{D}_{\mathcal{P}} + \mathbb{I})\,. \tag{5.16}$$

α	Measured time [s]
No preconditioning	7601.10
2^0	292.11
2^1	239.73
2^2	217.22
2^3	214.40
2^4	224.69

Table 5.13: Table with measured time of solving all the linear systems with GMRES preconditioned with $\mathbb{R}_1(\alpha)$ needed for construction of 2D electron energy-loss spectrum.

by sparser matrix $\tilde{\mathbb{K}}_{BJ(L)}(ASB) = \mathrm{Re}\left(\mathbb{K}_{BJ(L)}(ASB)\right)$ with $\mathbb{K}_{BJ(L)}(ASB)$ defined earlier in the section about the block Jacobi preconditioning. Let us remind that matrix $\mathbb{K}_{BJ(L)}(ASB)$ is a real, symmetric and block diagonal matrix, which means, that construction of its decomposition is much cheaper than the decomposition of matrix $\mathbb{B}_{\mathcal{P}}$. We define a preconditioner

$$\mathbb{R}_1^{BJ}(\alpha) = \frac{1}{2\alpha}\left(\alpha\tilde{\mathbb{K}}_{BJ(L)}(ASB) + i\,\mathbb{I}\right)(\alpha\mathbb{D}_{\mathcal{P}} + \mathbb{I})\,. \tag{5.17}$$

In the table 5.14 we can see measured times for GMRES with preconditioning $\mathbb{R}_1^{BJ}(\alpha)$ for some selected values of α. If we compare these results with the previous ones in table 5.13, we can see a considerable improvement. This time, we have got the best result for $\alpha = 4$, for which the time is almost 99 times smaller than for GMRES without preconditioning.

α	Measured time [s]
No preconditioning	7601.10
2^0	105.87
2^1	79.31
2^2	77.74
2^3	91.22
2^4	117.36

Table 5.14: Table with measured time of solving all the linear systems with GMRES preconditioned with $\mathbb{R}_1^{BJ}(\alpha)$ needed for construction of 2D electron energy-loss spectrum.

Figure 5.46 contains number of iterations as a function of electron energy ε for GMRES modified Zhang-Dai preconditioner $\mathbb{R}_1^{BJ}(\alpha)$. As expected, our modification of Zhang-Dai preconditioner increased the number of iterations comparing with the original preconditioner $\mathbb{R}_1(\alpha)$, which is clear from plots in Figure 5.46. Despite this fact, the time needed for solving the equations is much smaller and from this point of view the modified preconditioner is more efficient for our linear system.

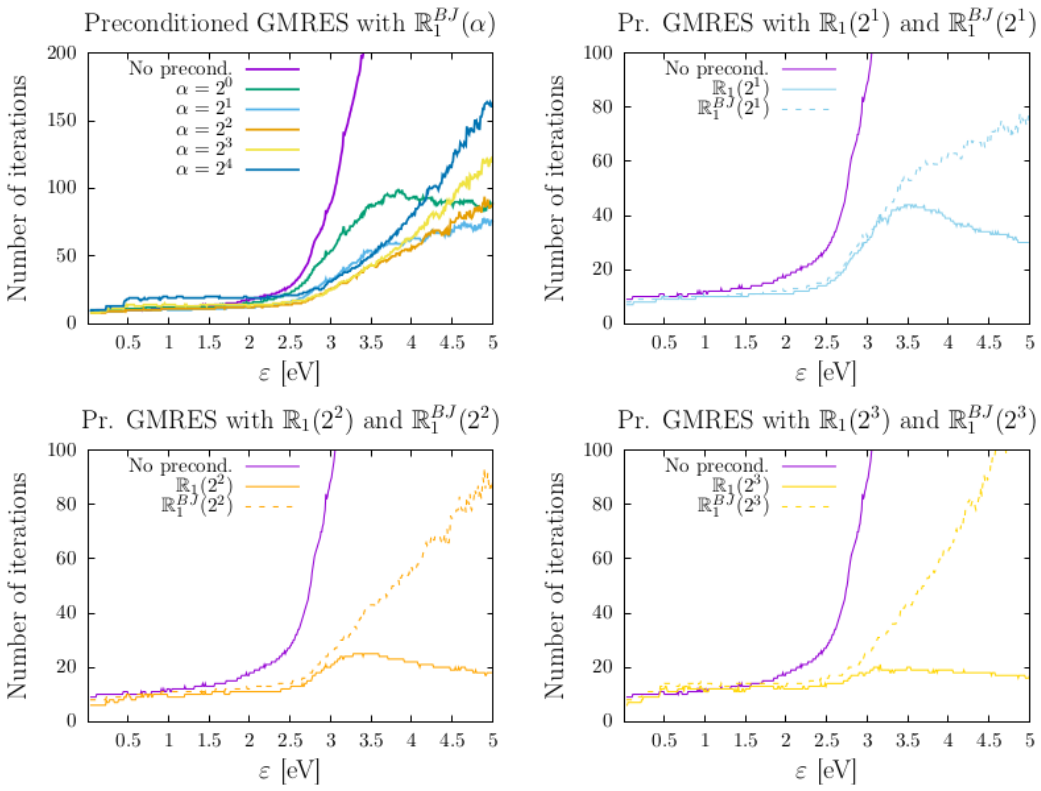


Figure 5.46: Plots of number of iterations as a function of electron energy ε for GMRES with modified Zhang-Dai preconditioner $\mathbb{R}_1^{BJ}(\alpha)$. Also plots of number of iterations for selected values of α for comparison of $\mathbb{R}_1(\alpha)$ and $\mathbb{R}_1^{BJ}(\alpha)$.

Model B

Secondly, we have tested convergence of preconditioned GMRES on model B. In the Figure 5.47 we can see plots of number of iterations as a function of electron energy ε for GMRES with Zhang-Dai preconditioner for model B. It

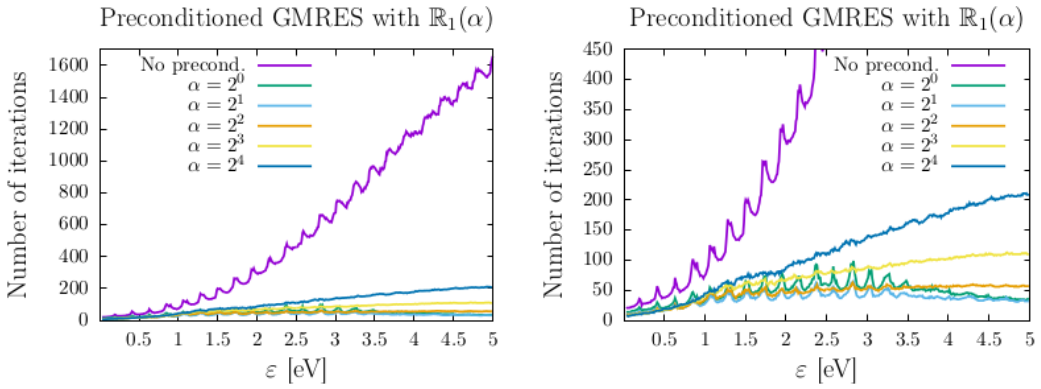


Figure 5.47: Plots of number of iterations as a function of electron energy ϵ for GMRES with Zhang-Dai preconditioner. The plot on the right contains the same data as plot on the left, but it has different range on vertical axis.

is clear, that the number of iterations used for solving the linear systems with preconditioned GMRES is much (more than ten times) smaller than the GMRES without preconditioning.

In the table 5.15 we can see measured times for GMRES with preconditioning $\mathbb{R}_1(\alpha)$ for different values of α. The measured times prove that using Zhang-

α	**Measured time [s]**
No preconditioning	75495.15
2^0	315.01
2^1	282.83
2^2	308.70
2^3	408.67
2^4	637.92

Table 5.15: Table with measured time of solving all the linear systems needed for construction of 2D electron energy-loss spectrum.

Dai preconditioner of GMRES method is more efficient than GMRES without preconditioning. This holds for all tested values of α. The best result is for $\alpha = 2$ for which the computation with Zhang-Dai preconditioning was more than 267 times faster than the GMRES alone.

We have also tried using our modification $\mathbb{R}_1^{BJ}(\alpha)$ of Zhang-Dai preconditioner. In the table 5.16 we can see measured times for GMRES with preconditioning $\mathbb{R}_1^{BJ}(\alpha)$ for the same values of α as usually. If we compare these results with the previous ones in table 5.16, we can again see an improvement. Similarly as for original Zhang-Dai preconditioner, we have got the best result for $\alpha = 2$. For this α, the computation is almost 779 times faster than for GMRES without preconditioning.

α	**Measured time [s]**
No preconditioning	75495.15
2^0	112.51
2^1	97.27
2^2	141.67
2^3	290.83
2^4	709.32

Table 5.16: Table with measured time of solving all the linear systems with GMRES preconditioned with $\mathbb{R}_1^{BJ}(\alpha)$ needed for construction of 2D electron energy-loss spectrum.

Figure 5.48 contains number of iterations as a function of electron energy ε for GMRES modified Zhang-Dai preconditioner $\mathbb{R}_1^{BJ}(\alpha)$. We can again see that our modification increased the number of iterations. Nevertheless this increase is not important in contrast to the improvement of method regarding the computation time.

Model C

Finally, we have also tested Zhang-Dai method for model C. In the Figure 5.49 we can see plots of number of iterations as a function of electron energy ε for GMRES with Zhang-Dai preconditioner for model C.

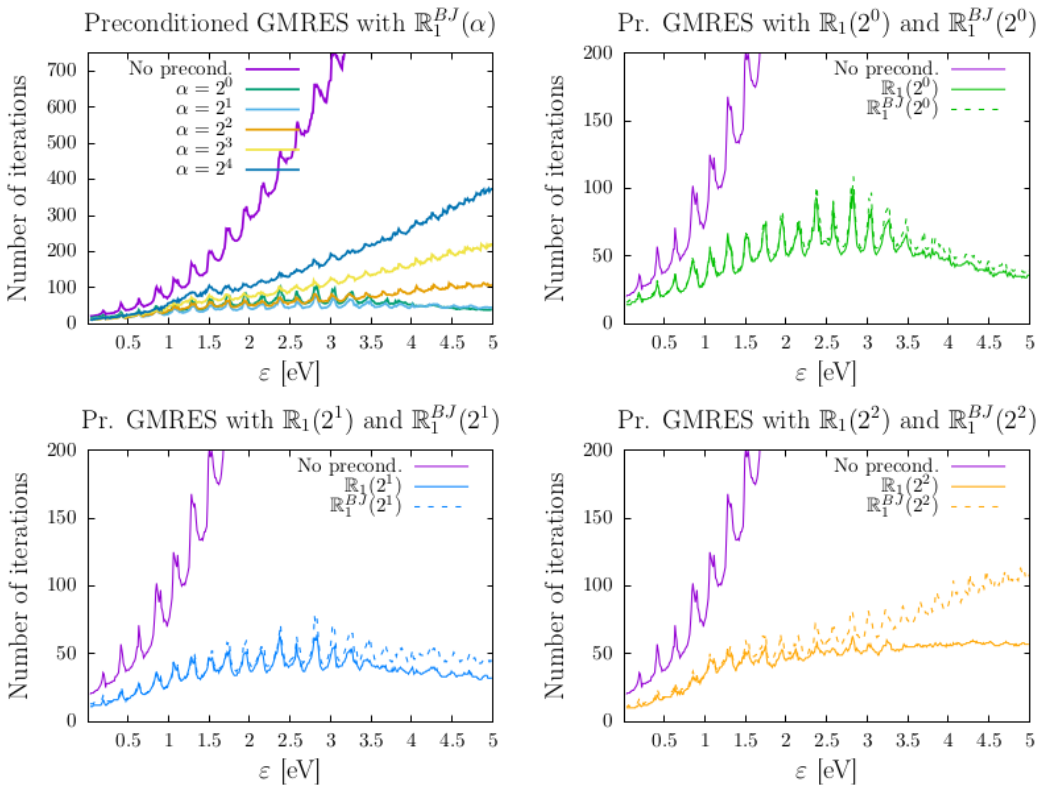


Figure 5.48: Plots of number of iterations as a function of electron energy ϵ for GMRES with modified Zhang-Dai preconditioner $\mathbb{R}_1^{BJ}(\alpha)$. Also plots of number of iterations for selected values of α for comparison of $\mathbb{R}_1(\alpha)$ and $\mathbb{R}_1^{BJ}(\alpha)$.

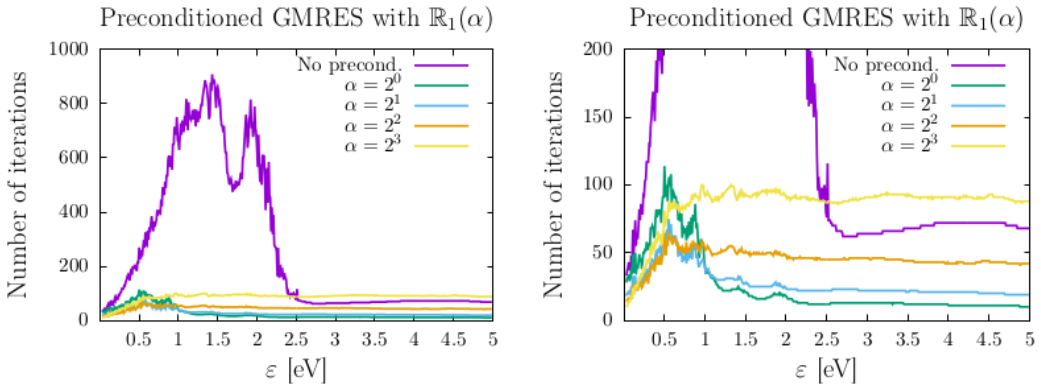


Figure 5.49: Plots of number of iterations as a function of electron energy ε for GMRES with Zhang-Dai preconditioner. The plot on the right contains the same data as plot on the left, but it has different range on vertical axis.

In the table 5.17 we can see measured times for GMRES with preconditioning $\mathbb{R}_1(\alpha)$ for different values of α. We see that for all tested values of α the Zhang-Dai preconditioner is worth using in case of model C. We have obtained the best result for $\alpha = 1$ for which the preconditioned calculation takes six times less time.

We have again tested our modification $\mathbb{R}_1^{BJ}(\alpha)$ of Zhang-Dai preconditioner on model C. However in this case we have used the matrix $\tilde{\mathbb{K}}_{BJ(L)}(SAB) = \mathrm{Re}\left(\mathbb{K}_{BJ(L)}(SAB)\right)$ indstead of $\tilde{\mathbb{K}}_{BJ(L)}(ASB)$, because this version worked better for model C. In the table 5.18 we can see measured times for GMRES with

α	Measured time [s]
No preconditioning	9374.98
2^0	1536.41
2^1	1547.31
2^2	1769.70
2^3	2336.80

Table 5.17: Table with measured time of solving all the linear systems needed for construction of 2D electron energy-loss spectrum.

preconditioning $\mathbb{R}_1^{BJ}(\alpha)$ for selected values of α. Comparing the new results

α	Measured time [s]
No preconditioning	9374.98
2^0	174.48
2^1	178.75
2^2	841.41
2^3	8821.63

Table 5.18: Table with measured time of solving all the linear systems with GMRES preconditioned with $\mathbb{R}_1^{BJ}(\alpha)$ needed for construction of 2D electron energy-loss spectrum.

with the previous ones in table 5.18, we can again see, that using $\mathbb{R}_1^{BJ}(\alpha)$ leads to faster computation than using Zhang-Dai preconditioner $\mathbb{R}_1(\alpha)$. The best result is achived for $\alpha = 1$. Figure 5.50 contains number of iterations as a function of electron energy ε for GMRES modified Zhang-Dai preconditioner $\mathbb{R}_1^{BJ}(\alpha)$. As usually, we can see an increase of number of iterations, which is the most prominent for $\alpha = 4$. On the other hand, for smaller values of α, the number of iteraions is almost the same for both preconditioning methods.

5.7 Liao-Zhang preconditioner

Secondly, we have decided to test Liao-Zhang preconditioner published in Liao and Zhang [2017]. Let us remind that in this case we consider a real block two-by-two linear system in form

$$\begin{pmatrix} \mathbb{B}_{\mathcal{P}} & -\mathbb{D}_{\mathcal{P}} \\ \mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \tag{5.18}$$

and the preconditioner is then in form

$$\mathbb{P}_{BM}(\alpha) = \begin{bmatrix} \mathbb{B}_{\mathcal{P}} - \frac{1}{\alpha}\mathbb{D}_{\mathcal{P}}^2 & -\mathbb{D}_{\mathcal{P}} \\ \frac{1}{\alpha}\mathbb{B}_{\mathcal{P}}\mathbb{D}_{\mathcal{P}} & \mathbb{B}_{\mathcal{P}} \end{bmatrix}. \tag{5.19}$$

Matrices $\mathbb{B}_{\mathcal{P}}$ and $\mathbb{D}_{\mathcal{P}}$ are assumed to be symmetric and at least one of them non-singular. Although we can not assume non-singularity of either, we still decided to try using Liao-Zhang preconditioner. We have again tested it for different values of α.

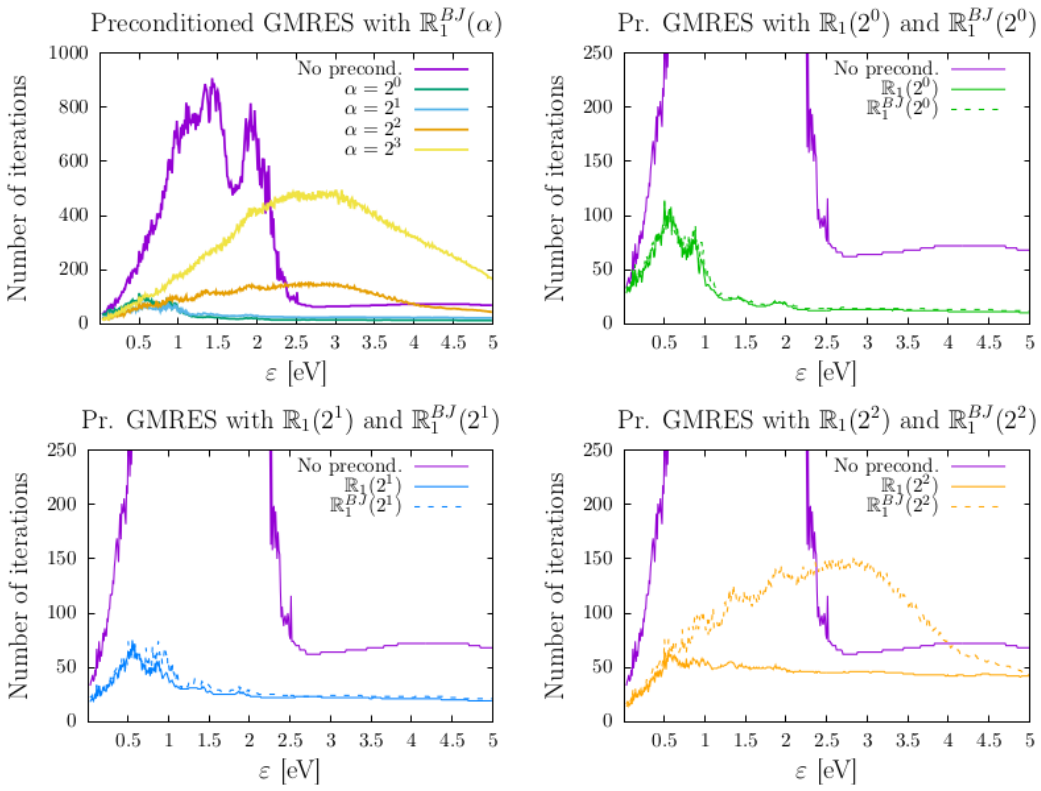


Figure 5.50: Plots of number of iterations as a function of electron energy ε for GMRES with modified Zhang-Dai preconditioner $\mathbb{R}_1^{BJ}(\alpha)$. Also plots of number of iterations for selected values of α for comparison of $\mathbb{R}_1(\alpha)$ and $\mathbb{R}_1^{BJ}(\alpha)$.

Model A

Let us firstly introduce the results for model A. In the Figure 5.51 we have plotted number of iterations for both GMRES with and without preconditioning.

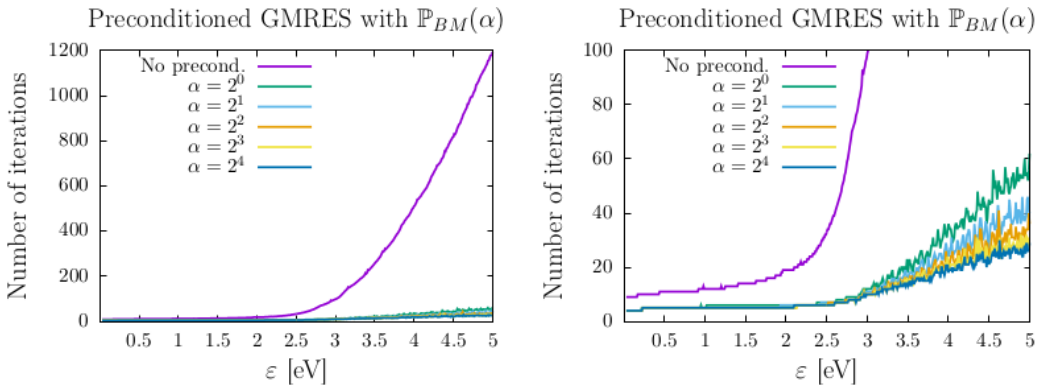


Figure 5.51: Plots of number of iterations as a function of electron energy ε for Liao-Zhang preconditioner. The plot on the right contains the same data as plot on the left, but it has different range on vertical axis.

We can see, that the number of iterations of GMRES with preconditioning is more than ten times smaller (for all tested values of α) than the number of GMRES itself. Secondly, we will again compare measured times, which is also very important aspect regarding the efficiency of preconditioner. The time needed

for solving all required linear systems with GMRES without preconditioning is $t = 4473.23$ s. In the table 5.19 we can see measured times for GMRES with preconditioning $\mathbb{P}_{BM}(\alpha)$ for different values of α. We can see that the precondi-

α	**Measured time** $[s]$
2^0	617.53
2^1	517.77
2^2	469.06
2^3	440.69
2^4	425.49

Table 5.19: Table with measured time of solving all the linear systems needed for construction of 2D vibrational spectrum.

tioned GMRES is much more efficient than the GMRES without preconditioning. The best result is for $\alpha = 16$.

Model B

Secondly, we have also tested Liao-Zhang preconditioner on model B. In the Figure 5.52 we have plotted number of iterations for both GMRES with and without preconditioning.

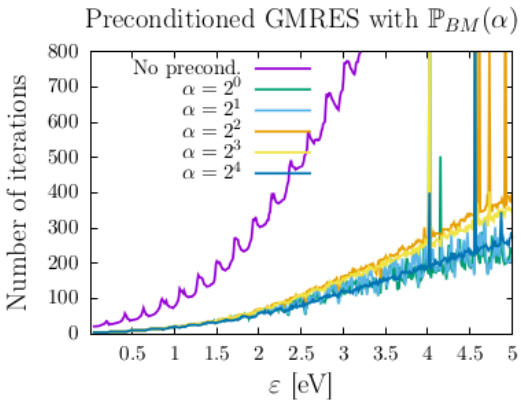


Figure 5.52: Plots of number of iterations as a function of electron energy ε for Liao-Zhang preconditioner.

We can see, that for most energies, the preconditioned GMRES needs small number of iterations for solving the system of linear equation. Nevertheless, there are also energies, for which it has difficulty converging at all (we can observe narrow peaks). As usually, we also have to compare times needed for solving the systems.

The time needed for solving all required linear systems with GMRES without preconditioning is $t = 26133.68$ s. In the table 5.20 we can see measured times for GMRES with preconditioning $\mathbb{P}_{BM}(\alpha)$ for different values of α. Measured times are for all values of α at least five times smaller than without preconditioning, which means, that this preconditioner is definitely worth using. However, we have to be carefull, because in some particular energies, we can get the result, which does not have required accuracy.

α	Measured time $[s]$
2^0	3611.61
2^1	3791.60
2^2	5227.48
2^3	4695.78
2^4	3572.21

Table 5.20: Table with measured time of solving all the linear systems needed for construction of 2D vibrational spectrum.

Model C

Finally, let us briefly comment on model C. In this case we were not successful at all. All we were able to count was the number of iterations without preconditioning, which took $t = 82106.60$ s (almost 23 hours). Since all tested preconditioners did not even reach half of the energy interval even in twice the time, we concluded that method Liao-Zhang is not suitable for model C. In the Figure 5.53 we have at least for interest plotted number of iterations for both GMRES with and without preconditioning.

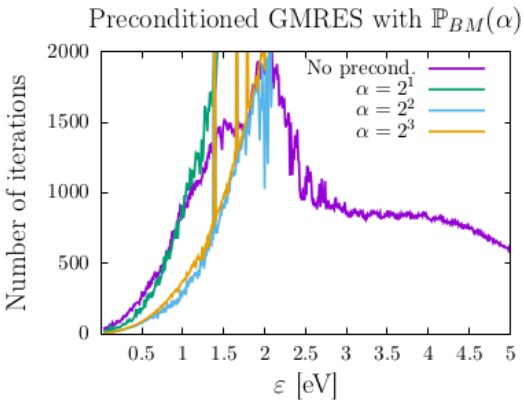


Figure 5.53: Plots of number of iterations as a function of electron energy ε for Liao-Zhang preconditioner.

5.8 Comparison of results

At the end of this chapter, let us compare the different types of preconditioning techniques. In the following sections, we will present graphs comparing individual types of preconditioning methods. The ploted data is the same as in the previous sections, but the goal is a comparison of different types of preconditioning techniques to each other. The graphs always contain the best results (in terms of time) from each type of preconditioning.

Model A

Let us firstly comment on the results for model A. Figures 5.54 and 5.55 show the comparison of different types of preconditioning techniques. It is clear from the

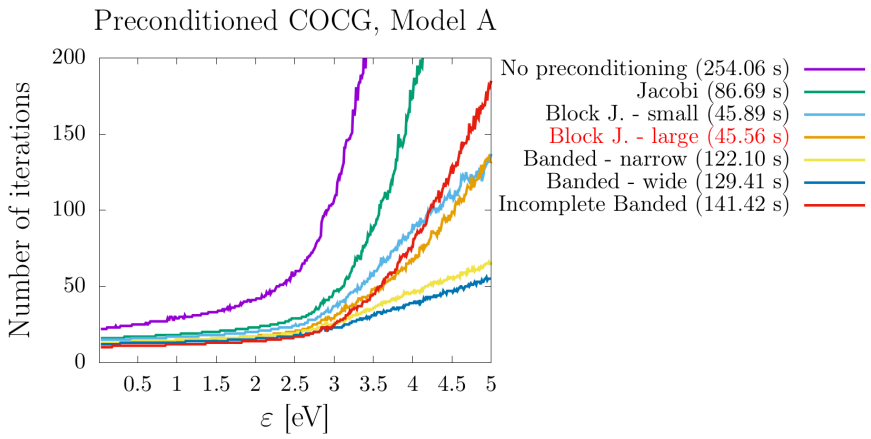


Figure 5.54: The graph plots the the number of iterations as a function of the electron energy for the preconditioned COCG method.

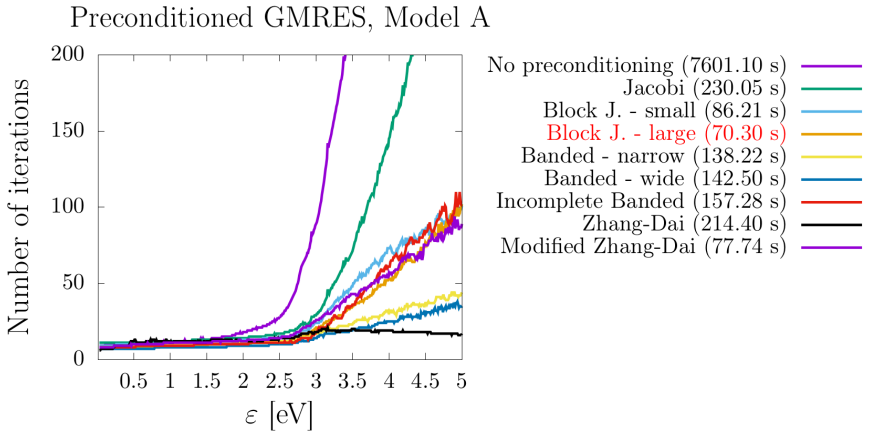


Figure 5.55: The graph plots the the number of iterations as a function of the electron energy for the preconditioned GMRES method.

Figures 5.54 and 5.55 that the most efficient preconditioning technique in terms of time is block Jacobi method. In the case of block Jacobi method we were able to accelerate the computation more than five times for COCG method and 108 times for GMRES. On the other hand, in the language of number of iterations, we obtained the best result for banded preconditioners (in case of method COCG) and Zhang-Dai preconditioner (in case of GMRES). This result is not surprising, since the construction of the block Jacobi preconditioner (LDL^T decomposition of block diagonal matrix in particular) requires much fewer operations than the banded matrix factorization. Interestingly, in the case of model A, the results for the two block Jacobi methods (using small and large blocks) did not differ much. This may be due to the fact that the off-diagonal matrix elements are small (even if the matrix is not diagonally dominant which is true for $\varepsilon > 2.5$). Note also that the best results almost without exception belonged to the rearrangement of the matrix (order of Kronecker products), which we denote as (SAB). Figures 5.56 and 5.57 include measured times for all preconditioning methods and various matrix rearrangements.

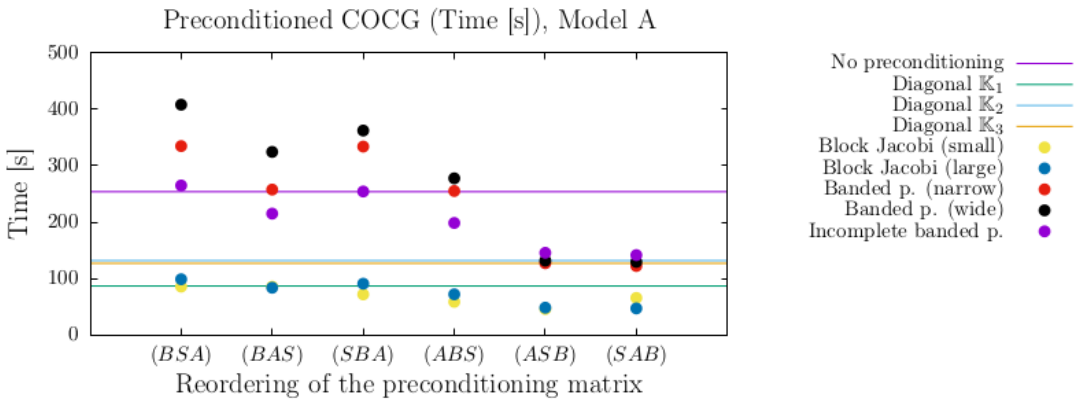


Figure 5.56: The graph plots the the number of iterations as a function of the electron energy for the preconditioned COCG method.

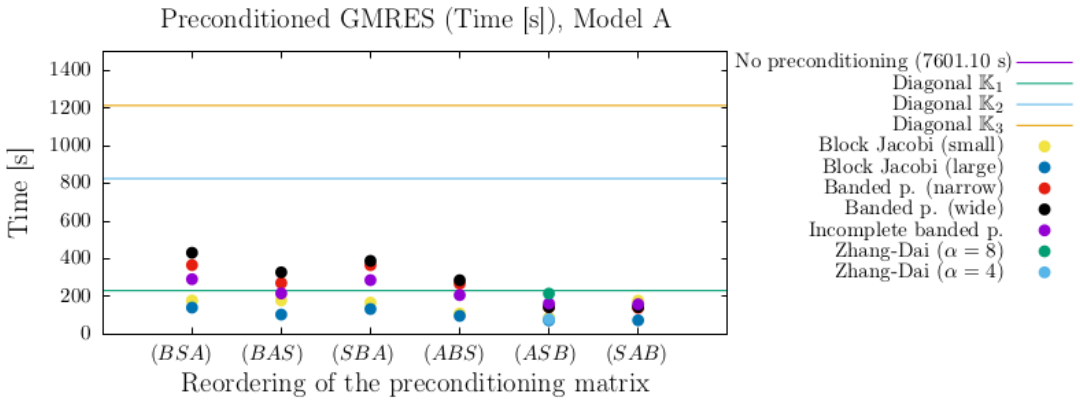


Figure 5.57: The graph plots the the number of iterations as a function of the electron energy for the preconditioned GMRES method.

Model B

For model B the results do not differ much from model A. Figures 5.58 and 5.59 show how the different types of preconditioning techniques compared to each other work for model B. It is clear from the Figures 5.58 and 5.59 that the most efficient preconditioning technique in terms of time is again block Jacobi method. In the case of block Jacobi method we were able to accelerate the computation more than 16 times for COCG method and 1398 times for GMRES. On the other hand, in the language of number of iterations, we obtained for both COCG and GMRES the best result for banded preconditioners. The best results were obtained for rearrangements (SAB) and (ASB) which happens because the terms in the matrix corresponding to the vibrational dimensions 'A' and 'S' are most prominent in the model B. Figures 5.60 and 5.61 include measured times for all preconditioning methods and various matrix rearrangements. In the case of preconditioned GMRES, we have achieved a really great time improvement for model B. We see, that all the tested methods are worth using from the point of wiev of time.

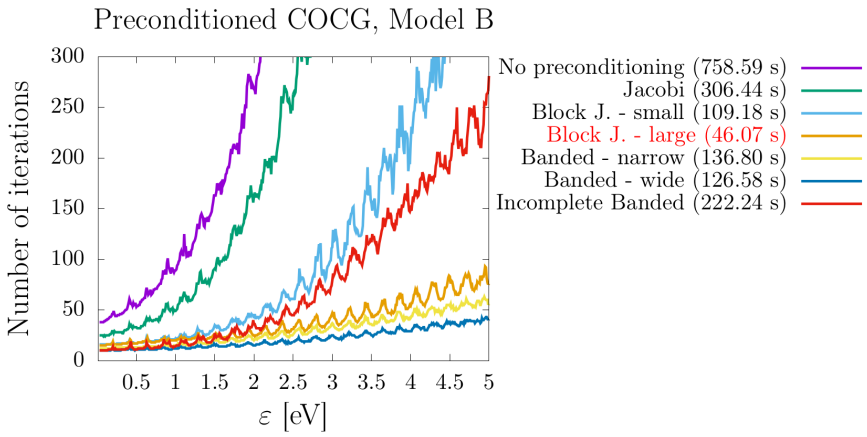


Figure 5.58: The graph plots the the number of iterations as a function of the electron energy for the preconditioned COCG method.

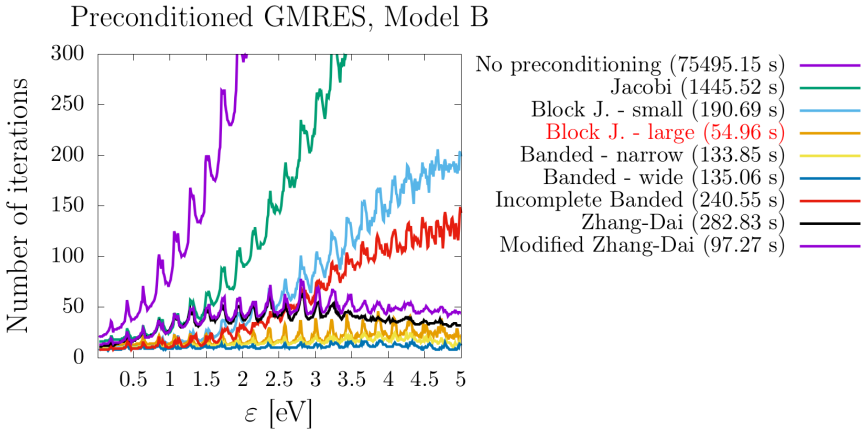


Figure 5.59: The graph plots the the number of iterations as a function of the electron energy for the preconditioned GMRES method.

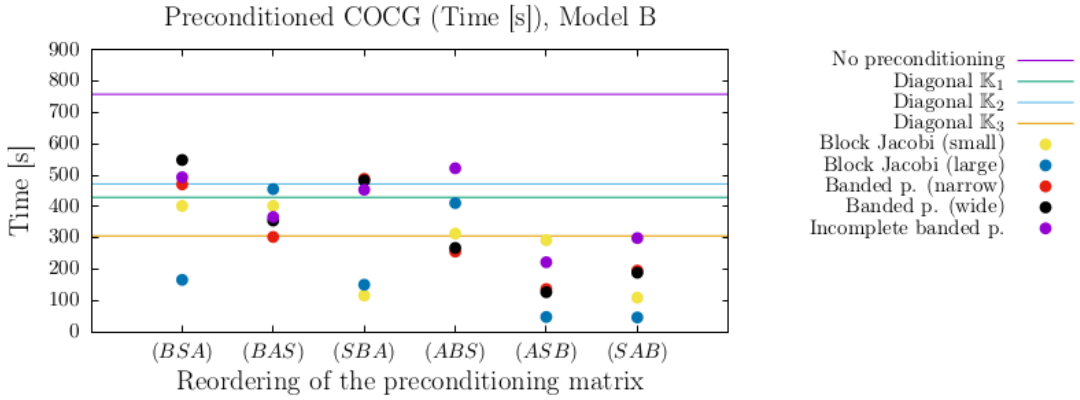


Figure 5.60: The graph plots the the number of iterations as a function of the electron energy for the preconditioned COCG method.

Model C

Finally, let us compare the results for model C. Figures 5.62 and 5.63 show the comparison of different types of preconditioning techniques. As usually, in the

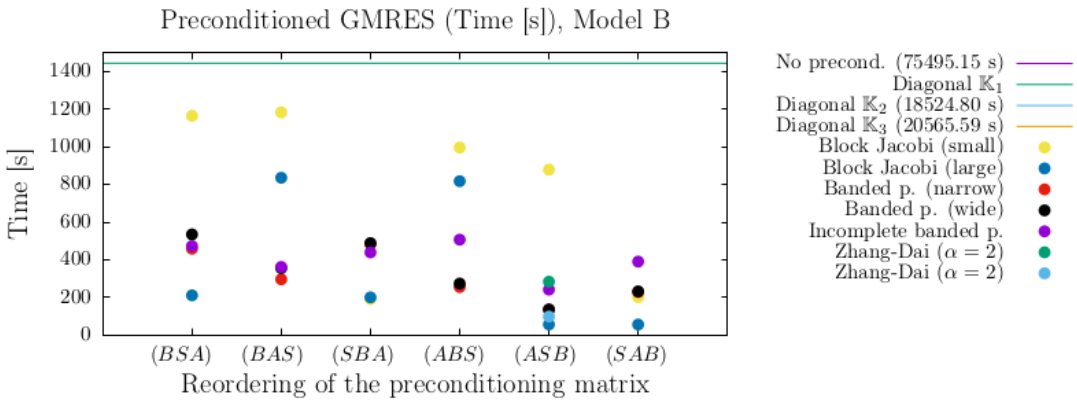


Figure 5.61: The graph plots the the number of iterations as a function of the electron energy for the preconditioned GMRES method.

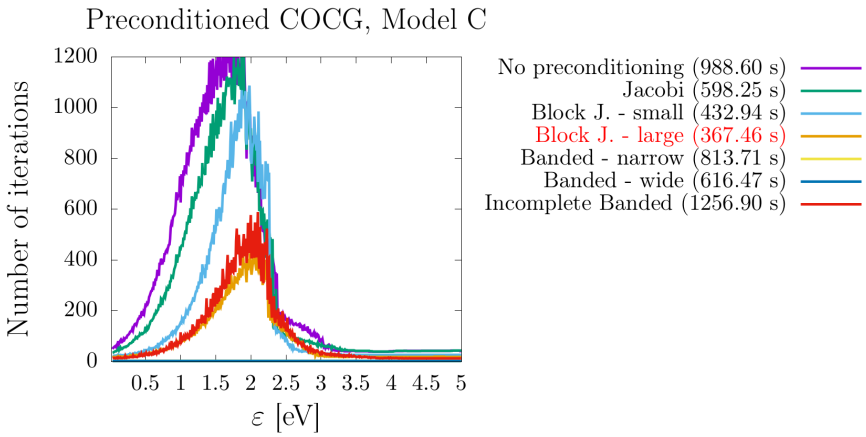


Figure 5.62: The graph plots the the number of iterations as a function of the electron energy for the preconditioned COCG method.

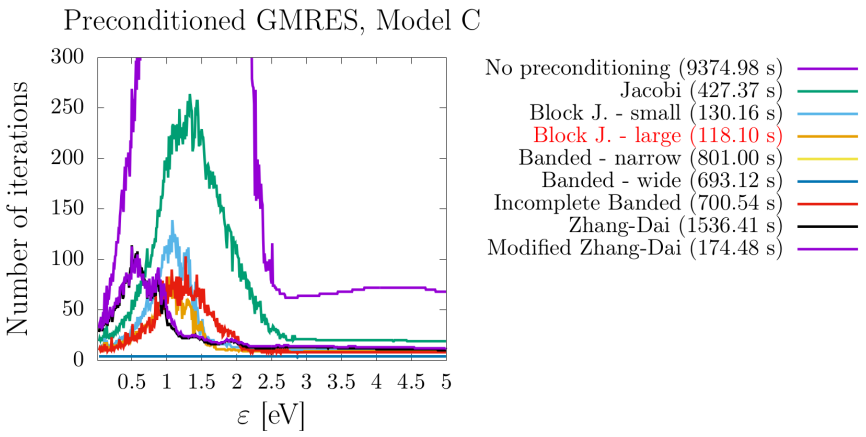


Figure 5.63: The graph plots the the number of iterations as a function of the electron energy for the preconditioned GMRES method.

Figures 5.62 and 5.63 we can see, that the most efficient preconditioner for both COCG and GMRES method is block Jacobi preconditioner. In the case of block Jacobi method we were for model C able to reduce computing time approximately twice for COCG method and almost 80 times for GMRES. For model C, the

version of the preconditioning with the vibration dimension 'S' in the first place proved to be the most effective, i. e. almost without exception, the (SBA) turned out to be the most effective arrangement.

Figures 5.60 and 5.61 include measured times for all preconditioning methods and various matrix rearrangements, i. e. specific orders of Kronecker products. It is clear from both figures that the preconditioners that do not contain the

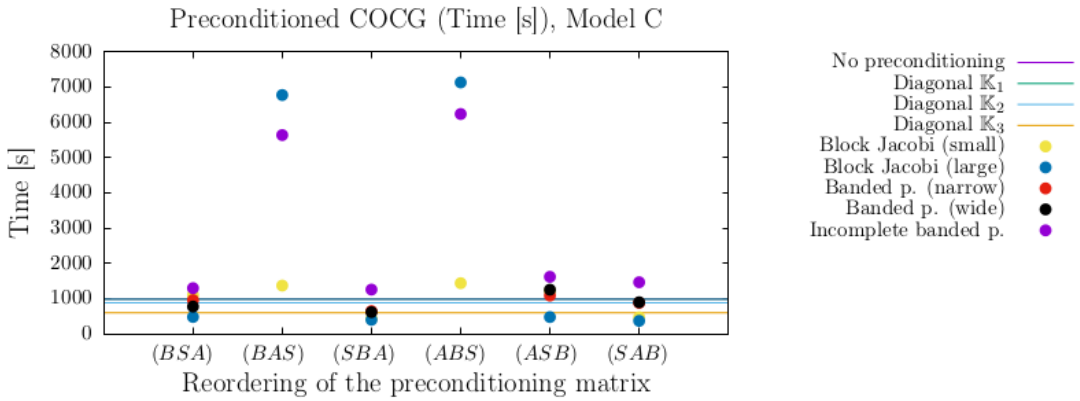


Figure 5.64: The graph plots the the number of iterations as a function of the electron energy for the preconditioned COCG method.

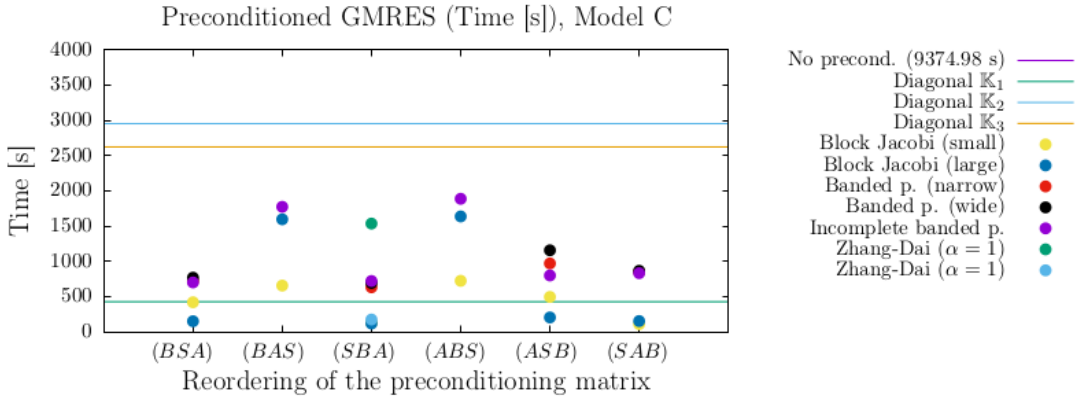


Figure 5.65: The graph plots the the number of iterations as a function of the electron energy for the preconditioned GMRES method.

vibrational dimension 'S' in one of the first two positions (i. e. (BAS) and (ABS)) perform much less efficiently than the other methods. An interesting fact is that the results of the GMRES method are even better than those of the COCG method. In the case of the block Jacobi preconditioner the computational time of GMRES method is reduced more than eighty times and the resulting algorithm even surpasses the best result obtained for COCG.

Conclusion

This work follows on from the bachelor thesis Šarmanová [2020] in which we dealt with comparison of different iterative methods for solving complex symmetric systems of algebraic equations. These arise from the discretization of integro-differential equation that represents a mathematical model of inelastic electron-molecule collision.

The description of low-energy electron-molecule collisions represents an interesting but also rather complicated problem, especially for more complex than diatomic molecules. To understand these processes, it is necessary to explain the phenomena appearing in 2D electron energy-loss spectra, which we obtain as outputs of modern crossed-beam electron-molecule collision experiments. For this reason, we would like to gain deeper understanding through mathematical modeling of the problem. The collision of an electron with a molecule can be mathematically defined within the quantum scattering theory, which tells us how to formulate a problem in the language of integro-differential equation. The discretization converts the problem to a system of linear algebraic equations with a complex and symmetric matrix. The dimension of matrix and the number of its nonzero elements are strongly dependent on the number of degrees of freedom of the model. So far, the case has been well studied for one or two degrees of freedom that can be efficiently (numerically) solved.

In this work we dealt with a more complex and general model involving three vibrational degrees of freedom. As we know, the dimension of the system of linear equations that needs to be solved increases exponentially with the number of vibrational degrees of freedom. Fortunately, the matrix of this system is still sparse and for this reason we believe that the use of iterative methods to solve the resulting system is a suitable choice. However, as we were convinced when testing the convergence rate of the Krylov subspace methods for the model with two degrees of freedom, iterative methods suffer from slow convergence. This fact can greatly limit us, because if we are not able to effectively solve the problem for a low number of degrees of freedom, we could hardly hope to succeed for more complex models. This motivated us to try using preconditioning which is considered to be crucial for the reliability of iterative techniques across the literature. Our main goal in this work was to find a suitable preconditioning technique for Krylov subspace methods, which would ensure their faster convergence. If we succeed in finding the precondition and manage to solve the given type of problem effectively, it can provide us with the possibility to model more complex molecules in the future, which will be the subject of further studies.

We devoted the first chapter to the brief description of the motivation. The main goal was to introduce the physical problem of electron-molecule collision and explain the origin of 2D electron energy-loss spectrum resulting from the modern experiments. We also explained how to construct vibrational spectra as a 2D plot of the integral cross section. Subsequently, we formulated the problem in the language of partial integro-differential equation and briefly described its individual terms.

In the second chapter, we introduced the discretization which converts the problem to a system of linear algebraic equations with a complex symmetric

matrix. We also derived the elements of this matrix.

In the third chapter, we recalled two basic approaches to solving systems of linear equations and we described two selected iterative methods - Conjugate orthogonal conjugate gradient method (COCG) and the Generalized minimal residual method (GMRES) and we also provided pseudocodes of their preconditioned versions. However, the main part of the third chapter was devoted to the idea of preconditioning. After that we introduced a few preconditioning techniques. Some of the methods we mentioned are known and commonly used, while others have only recently been published and are designed for specific complex symmetric matrices. Suprisingly, the authors often link the eigenvalues of the matrix to the convergence rate of iterative methods, although we know that the connection between them is not straightforward.

The fourth chapter was devoted to the description of the test models we have suggested. These models were designed to capture the various typical properties of the molecules and were qualitatively inspired by the water molecule. After that we investigated various properties of matrices that correspond to each model.

The core of the whole work is the fifth chapter, which is devoted to numerical experiments. We have implemented a program, which is used to solve our systems of linear equations with preconditioned iterative methods and calculate the energy-loss spectrum. We have chosen the programming language Fortran 90, which is still widely used in computational physics today. The advantage of this program is that it can be easily generalized, because vector multiplication by this matrix is implemented without the need to explicitly construct the matrix and keep it in memory. This allows us to take full advantage of the sparsity of the matrix. The program allows the use of 36 preconditioning techniques whose efficiency we have tested and compared.

For the COCG method, we have in general achieved the greatest improvement over time using the block Jacobi preconditioners. In case of model A, we were able to reduce the computational time more than five times. The difference between the preconditioned and unpreconditioned versions of COCG was the most significant for model B, for which the best preconditioner reduced the computational time more than sixteen times. Finally, for model C we were able to accelerate the computation approximately twice. On the other hand, in the terms of number of iterations, we have obtained the best results for banded preconditioners.

For the GMRES method we have similarly as for the COCG method got the best result in terms of time for block Jacobi preconditioning. In this case, however, we have to say that for all the models the preconditioning is really worth using. In particular, for model A the time reduced more than 108 times, for model B the improvement in terms of time is actually 1398 times and for model C the computation has accelerated approximately 80 times. In the terms of the number of iterations, we have in general again obtained the best results for banded preconditioners. However, this is in line with expectations, as the band matrices we used as preconditioners approximate our system of linear equations better than a diagonal or block diagonal matrix. The reason why the banded preconditioning was not more efficient in terms of time even though it reduced the number of iterations better, is that the construction of preconditioner (LDL^T decomposition of matrix) requires much more operations in comparison to decompositions of small diagonal blocks. It should be noted that it has also been shown that a

particular rearrangement of the matrix plays an absolutely essential role for the effectiveness of the preconditioning.

To summarize the whole work, let us say that we actually have found several preconditioning techniques that work well for all three of our models, each of which has a slightly different physical character. We are happy for this success, because an efficient solution of sparse complex symmetric systems of linear equations is a key ingredient in mathematical modeling electron-molecule collisions and therefore also understanding 2D energy-loss spectra.

Bibliography

Owe Axelsson, Shiraz Farouq, and Maya Neytcheva. Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems. *Numerical Algorithms*, 73:631–663, 2016.

Zhong-Zhi Bai, Michele Benzi, Fang Chen, and Zeng-Qi Wang. Preconditioned MHSS Iteration Methods for a Class of Block Two-by-Two Linear Systems with Applications to Distributed Control Problems. *IMA Journal of Numerical Analysis*, 33, 2013a.

Zhong-Zhi Bai, Fang Chen, and Zeng-Qi Wang. Additive block diagonal preconditioning for block two-by-two linear systems of skew-hamiltonian coefficient matrices. *Numerical Algorithms*, 62:655–675, 2013b.

Michele Benzi and Daniele Bertaccini. Block preconditioning of real-valued iterative algorithms for complex linear systems. *IMA Journal of Numerical Analysis*, 28:598–618, 2008.

Erin Carson and Zdeněk Strakoš. On the cost of iterative computations. *Philosophical transactions A*, 378, 2020.

Lu-Bin Cui, Xiao-Qing Zhang, and Yu-Tao Zheng. A preconditioner based on a splitting-type iteration method for solving complex symmetric indefinite linear systems. *Japan Journal of Industrial and Applied Mathematics*, 38:965–978, 2021.

Wolfgang Domcke. Theory of resonance and threshold effects in electron-molecule collisions: The projection-operator approach. *Physics Reports*, 208(2):97–188, 1991.

Estrada, Cederbaum, and Domcke. Vibronic coupling of short-lived electronic states. *The Journal of chemical physics*, 84(1):152–169, 1986.

Hong-Tao Fan, Yan-Jun Zhang, and Ya-Jing Li. A modified block preconditioner for complex nonsymmetric indefinite linear systems. *Applied Mathematics and Computation*, 358:455–467, 2019.

Jiří Formánek. *Úvod do kvantové teorie*. Druhé upravené a rozšířené vydání. Academia, Praha, 2004. ISBN 80-200-1176-5.

Anne Greenbaum, Vlastimil Pták, and Zdeněk Strakoš. Any nonincreasing convergence curve is possible for gmres. *SIAM J. Matrix Anal. Appl.*, 17:465–469, 1996.

Brian C Hall. *Quantum theory for mathematicians*. Springer, 2013.

Zhao-Zheng Liang and Guo-Feng Zhang. Robust additive block triangular preconditioners for block two-by-two linear systems. *Numerical Algorithms*, 82: 503–537, 2019.

Li Dan Liao and Guo Feng Zhang. Efficient Preconditioner and Iterative Method for Large Complex Symmetric Linear Algebraic Systems. *East Asian Journal on Applied Mathematics*, 7(3):530–547, 2017.

Li-Dan Liao and Guo-Feng Zhang. A note on block diagonal and block triangular preconditioners for complex symmetric linear systems. *Numerical Algorithms*, 80:1143–1154, 2019.

Jörg Liesen and Zdeněk Strakoš. *Krylov subspace methods : principles and analysis*. Numerical mathematics and scientific computation. Oxford University Press, Oxford, 1st ed. edition, 2013. ISBN 9780199655410.

Maeddeh Pourbagher and Davod Khojasteh Salkuyeh. On the solution of a class of complex symmetric linear systems. *Applied Mathematics Letters*, 76:14–20, 2018.

Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. *SIAM Journal on Scientific & Statistical Computing*, pages 856–869, 1986.

Yousef Saad. *Iterative methods for sparse linear systems*. Second edition. Society for Industrial and Applied Mathematics, 2003.

Martina Šarmanová. *Iteračvní výpočty vibrační dynamiky při rozptylu elektronu molekulou*. 2020.

Qin-Qin Shen and Quan Shi. A variant of the HSS preconditioner for complex symmetric indefinite linear systems. *Computers and Mathematics with Applications*, 75(3):850–863, 2018.

Duintjer Tebbens, Erik Jurjen, Iveta Hnětynková, Martin Plešinger, Zdeněk Strakoš, and Petr Tichý. *Analýza metod pro maticové výpočty: základní metody*. První vydání. Matfyzpress, Praha, 2012. ISBN 978-80-7378-201-6.

Henk A. van der Vorst and Jan Melissen. A Petrov-Galerkin type method for solving Ax=b, where A is symmetric complex. *IEEE Transactions on Magnetics*, 26(2):706–708, 1990.

Shi-Liang Wu. Several variants of the Hermitian and skew-Hermitian splitting method for a class of complex symmetric linear systems. *Numerical Linear Algebra with Applications*, 22:338–356, 2015.

Shi-Liang Wu and Cui-Xia Li. A splitting method for complex symmetric indefinite linear system. *Journal of Computational and Applied Mathematics*, 313: 343–354, 2017.

Wei-wei Xu. A generalization of preconditioned MHSS iteration method for complex symmetric indefinite linear systems. *Applied Mathematics and Computation*, 219(21):10510–10517, 2013.

Xiang Yuan and Nai-Min Zhang. On the preconditioned conjugate gradient method for complex symmetric systems. *Applied Mathematics Letters*, 120, 2021.

Jian-Hua Zhang and Hua Dai. A new block preconditioner for complex symmetric indefinite linear systems. *Numerical Algorithms*, 74(3):889–903, 2017.

Jianhua Zhang and Hua Dai. A new splitting preconditioner for the iterative solution of complex symmetric indefinite linear systems. *Applied Mathematics Letters*, 49:100–106, 2015.

Ju-Li Zhang, Hong-Tao Fan, and Chuan-Quing Gu. An improved block splitting preconditioner for complex symmetric indefinite linear systems. *Numerical Algorithms*, 77:451–478, 2018.

Zhong Zheng, Jing Chen, and Yue-Fen Chen. A fully structured preconditioner for a class of complex symmetric indefinite linear systems. *BIT Numerical Mathematics*, 2021.