Universiteit Antwerpen

Faculteit Wetenschappen
Departement Fysica

Electronic properties of strained graphene and supercritical charge centers

Elektronische eigenschappen van uitgerekt grafeen en superkritische ladingscenters

Proefschrift voorgelegd tot het behalen
van de graad van doctor in de wetenschappen
aan de Universiteit Antwerpen te verdedigen door

Dean Moldovan

Promotor
Prof. Dr. Francois Peeters
Co-promotor
Dr. Massoud Ramezani Masir

Antwerpen, 2016

Members of the jury

Chairman

Prof. Dr. Michiel Wouters, University of Antwerp, Belgium

Supervisors

Prof. Dr. Francois Peeters, University of Antwerp, Belgium
Dr. Massoud Ramezani Masir, University of Texas at Austin, USA

Members

Prof. Dr. Pierre Van Mechelen, University of Antwerp, Belgium
Prof. Dr. Johan Verbeeck, University of Antwerp, Belgium
Prof. Dr. Eva Andrei, Rutgers University, Piscataway, USA
Prof. Dr. Chris Van Haesendonck, KU Leuven, Belgium

Contact information

Dean Moldovan
Groenenborgerlaan 171 U208
2020 Antwerpen
Belgium
dean.m8@gmail.com

“ I have made this letter longer than usual
because I lack the time to make it short. ”

Blaise Pascal

Acknowledgements

I would like to express my appreciation to my supervisor Prof. Francois Peeters for giving me this great opportunity and his continuous guidance and constructive criticism over the years. My deep thanks go to Dr. Massoud Ramezani Masir for sharing his knowledge and insights. I am also grateful for his patience, encouragement and friendly support.

I would also like to acknowledge the very fruitful collaboration with the experimental group at Rutgers University. My thanks go to Prof. Eva Andrei, Dr. Jinhai Mao and Dr. Yuhang Jiang for the insightful discussions and helping me gain a better understanding of the experimental side.

I am indebted to my friends and colleagues who have supported me over the years. I am sincerely grateful for all the stimulating discussions which helped further my research and understanding, but also for all the fun we have had and the nice moments which we have shared.

Last but not the least, I would like to thank my family for their strong support throughout writing this thesis and my life in general.

This work was supported by the European Science Foundation (ESF) under the EUROCORES Program EuroGRAPHENE within the project CONGRAN, the Flemish Science Foundation (FWO-Vl) and the Methusalem Funding of the Flemish Government.

List of abbreviations

ac Armchair

AC Atomic collapse

BZ Brillouin zone

CPU Central processing unit

DOS Density of states

DP Dirac point

GPU Graphics processing unit

KPM Kernel polynomial method

LDOS Local density of states

LL Landau level

QED Quantum electrodynamics

SpMV Sparse matrix-vector multiplication

STS Scanning tunneling spectroscopy

STM Scanning tunneling microscopy

WGM Whispering gallery mode

zz Zigzag

Contents

CHAPTER 1

Introduction

The field of microelectronics has been advancing at an incredible pace for more than half a century. Its evolution has always been driven by scaling down device sizes. Smaller sizes mean that more components can fit in the same area which permits the design of ever more complex circuits. Every couple of years, advancements in semiconductor manufacturing technology enable the creation of smaller transistors thereby doubling the number of logical circuits which can fit on a chip.

Intel co-founder Gordon Moore noticed this trend as early as 1965 and made predictions which would later become known as Moore's Law [1]. His original prediction held that the number of transistors on a chip would double yearly for at least the next 10 years. In the 1970's, the forecast was revised to doubling every 2 years and it has held steady for the following 40 years [2]. Like clockwork, it has fueled the exponential growth of the semiconductor industry. Along with the growth rate, Moore speculated about the impact on everyday life: in a very short time span advanced computing devices would be transformed from rare luxuries into common items.

The exponential growth curve is extremely powerful. Figure 1.0.1 shows the raw silicon chip (also known as a *die*) of a modern processor (typically found in personal computers), with transistors numbering in billions, compared to the Intel 4004, the first monolithic CPU which consisted of just 2300 transistors. Over the last 45 years the transistor count has increased by 6 orders of magnitude which has enabled the design of far more intricate circuits. While the downward scaling of transistor dimensions is the main

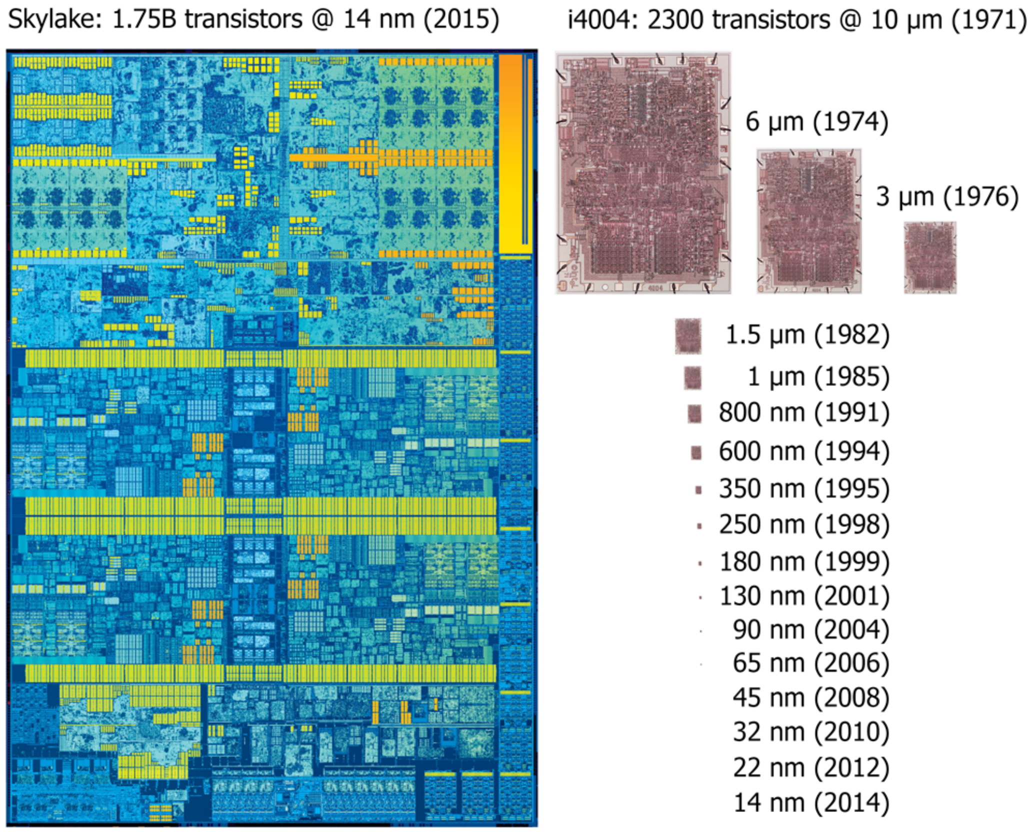


Fig. 1.0.1: Semiconductor transistor scaling over time. Images of a modern quad-core Intel "Skylake" chip (surface area 122 mm^2) and Intel 4004, the first monolithic CPU (12 mm^2). If the i4004 were produced on newer process nodes, it would shrink as presented here (assuming perfect scaling). The images were adapted from *www.intel.com*.

driving force, it is interesting to note that chip sizes have also increased over the years. The Intel "Skylake" chip presented here has a surface area of 122 mm^2 which is considerably larger than the 12 mm^2 of the i4004 (the chip images presented in Fig. 1.0.1 are to scale). The i4004 was produced on a 10 μm process node. The length number typically indicates the size of the smallest feature which can be created by a given manufacturing process. When a new process node is introduced, new chip architectures typically take advantage by increasing the number of transistors while keeping roughly the same surface area. However, if the i4004 had been continually shrunk without any circuit changes, the scaling over the years would look approximately like Fig. 1.0.1. Notice that the i4004 practically disappears from view by the 65 nm node. At 14 nm, it is truly insignificant compared to a modern chip.

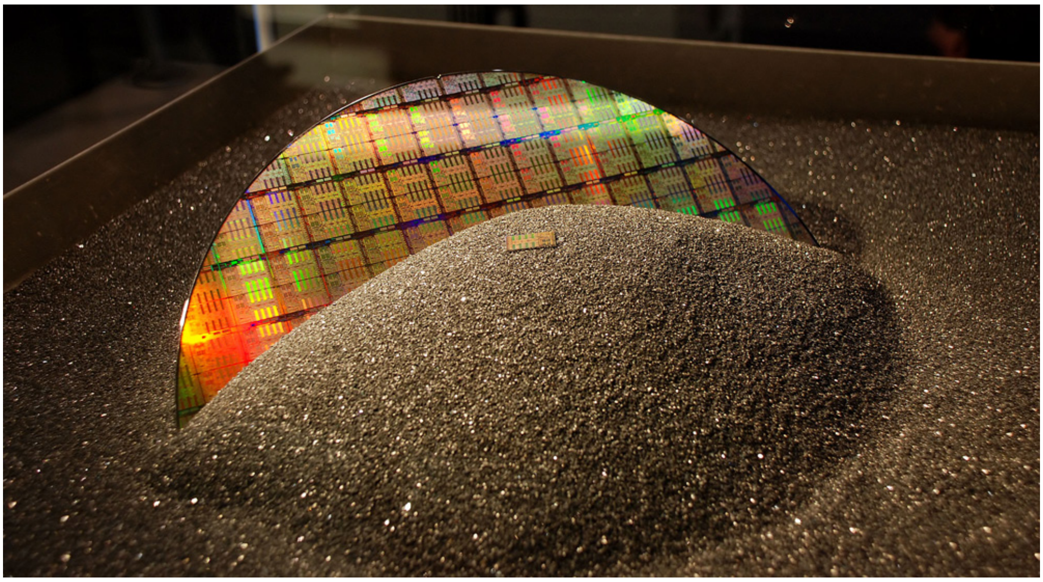


Fig. 1.0.2: A silicon chip wafer in sand [3]. An individual chip is also referred to as a *die* because it is obtained by *dicing* (cutting) the wafer into square pieces.

Scaling down from micrometer to nanometer length has been both advantageous and challenging. The nanometer range brings with it new quantum effects which must be taken into account for devices to function correctly. Even without considering the electronic properties, a quick visual inspection reveals some interesting small-size effects. As seen in Fig. 1.0.1, the old micrometer scale chip has a color which is mostly due to the reflection of its material. On the other hand, the modern chip has surface features in the nanometer range which bring it below the diffraction limit. In combination with refraction, the effect gives the modern chip a far more colorful appearance. It is even more apparent for an entire wafer of chips, as in Fig. 1.0.2, where a wide color spectrum is visible.

In the past, Moore's Law was maintained through a combination of smaller transistors and bigger chips. Viable surface area dimensions have since topped out at a few hundred mm^2, which made transistor scaling the only diving force. The introduction of Intel's Tick-Tock production model 10 years ago marked the start of an especially strict two year cycle [4]. This model called for the introduction of a new manufacturing process every other year (a "tick"). At that time an existing chip would be shrunk to a new process node, thus reducing chip size and power consumption. The year in between (a "tock") would be used to introduce a new chip architecture using the existing manufacturing process.

The Tick-Tock model was implemented on the 65 nm process node. Ten years and five process shrinks later, the strategy has begun to falter at the 14 nm node. The Tick-Tock model has now been retired, replaced by the new Process-Architecture-Optimization model [5]. It extends the time between new process shrinks to 3 years (the “process” stage of the new model). The intermediate years are used for the introduction of a new architecture and its later refinement on the same process node (the “architecture” and “optimization” stages). The prolonged cycle is a result of the ever-increasing complexity of developing smaller nodes.

To start with, difficulties arise from the manufacturing process itself: photolithography. Light is used to transfer the chip circuit design from masks onto the substrate. Reducing the size of the features requires light sources with ever-smaller wavelengths. Currently, excimer lasers in the deep ultraviolet range (193 nm) are common. In order to continue the drive to smaller feature sizes, photolithography will need to tap into the extreme ultraviolet range of the spectrum (13.5 nm) but finding a reliable light source has proven difficult.

Even after the manufacturing process has been perfected, the produced devices will run into more fundamental obstacles. Shrinking transistors from micrometer to nanometer dimensions introduces new challenges as quantum-mechanical effects become more apparent. Even after overcoming that, interatomic distances themselves become a barrier to continued scaling. Transistors are already at the limit where certain features have the thickness of only a handful of atoms [6]. With that, leakage current becomes a more prominent problem: electrons are harder to confine due to quantum tunnelling.

The present semiconductor roadmap is set to continue the march of Moore’s Law for at least another ten years, up to the 5 nm process node. Meanwhile, the search for an alternative approach to electronics is well underway. Instead of trying to overcome quantum effects and atomically thin structures, they are being embraced. New materials also bring new approaches for electron confinement.

1.1 Carbon flatland

The simplest and most abundant chemical elements have often proven to be incredibly versatile. Silicon is the eighth most common element in the universe. On Earth, it’s readily available in dust and sand. It’s not the best

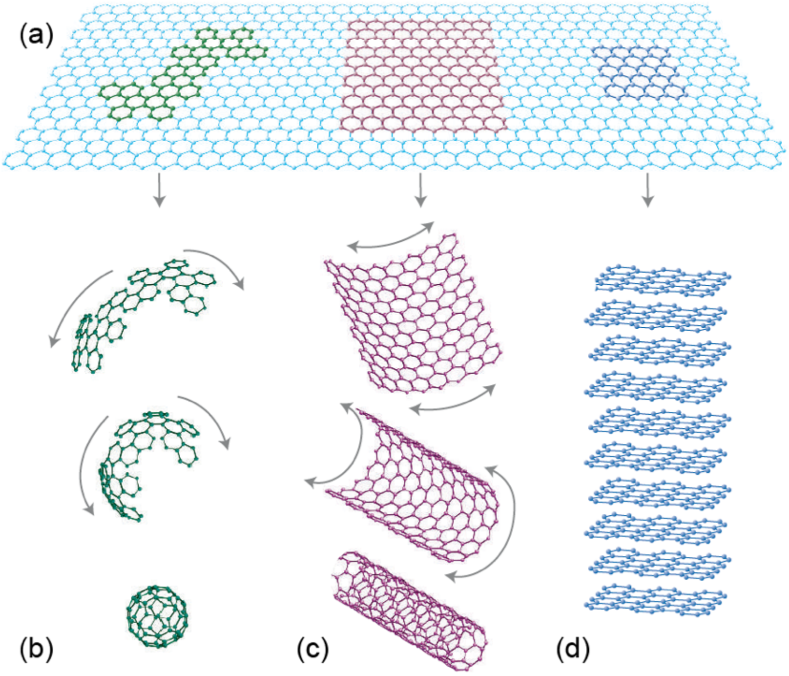


Fig. 1.1.1: Carbon allotropes: (a) graphene, (b) fullerene, (c) nanotube and (d) graphite. Adapted from [7].

semiconductor, but its abundance has made it the cornerstone of the rise of large-scale integrated circuits. Carbon is another very common element with many allotropes which are being considered to augment or completely replace the role of silicon. One could say that modern silicon-based semiconductor technology is made out of sand. Then perhaps it would not be too bizarre to imagine a future where technology is built out of pencil shavings.

The core of a pencil is made up of graphite, a material which consists of stacked layers of carbon. Within a single layer, carbon atoms are held in place by strong covalent bonds arranged in a hexagonal lattice. Multiple layers are held together by relatively weak van der Waals forces which makes layer separation easy. This is exactly why pencils work so well: friction between the tip and a paper sheet causes flakes of graphite to break off and leave marks on the paper.

The fact that layers of graphite can be pealed off easily has been known for a long time, however the idea to isolate just a single layer, effectively a 2D material, remained just a theoretical curiosity for decades. The consensus was that strictly 2D crystals could not exist. Peierls and Landau pointed out that thermal fluctuations of atoms in low-dimensional lattices would be comparable to interatomic distances, thus rendering 2D crystals unstable at any finite temperature [8, 9]. The argument was further backed by experimental observations which showed the melting temperature of thin films rapidly decreases with thickness [10]. Given the solid evidence that 2D materials should not exist, it was a bit of a surprise when a stable single-layer of graphite was successfully extracted in 2004 at Manchester University [11].

Graphene is the name given to a one-atom-thick layer of carbon atoms arranged in a hexagonal lattice. It was theoretically considered to be a purely two-dimensional material, but as it turned out, graphene is not completely flat. While it is a single-layer of atoms, a free standing graphene sheet is actually slightly rippled instead of flat [12]. This ensures enough stability to overcome thermal fluctuations and reconciles with the earlier theories about the instability of thin films [13]. The simplicity of the original method of extracting graphene further outlines its stability. A single layer can be isolated using micromechanical exfoliation, more commonly know as the "Scotch tape method". It involves repeated peeling of graphite layers with an adhesive and deposition on a silicon substrate. This is made possible by the weak van der Waals bonds which connect graphite layers. In fact, it is quite possible that a look under the microscope of some pencil shavings, left over from writing, would reveal a few small sheets of single-layer graphene.

Although it was the last to be discovered, graphene represents the base building block for most other carbon allotropes (see Fig. 1.1.1). As such, it has been studied theoretically for a long time, mainly as the starting point for the further study of the 3D allotropes. Even the name "graphene" was coined before the material itself was experimentally observed [14]. The band structure was calculated in 1947 [15], while the Landau level spectrum was studied in the 1950's [16]. The peculiar nature of the zero Landau level was explored theoretically in the 1980's [17] and proved to be responsible for the later experimental observation of the anomalous integer quantum Hall effect [18, 19].

Having access to the actual material confirmed many of the theoretical predictions, including some very interesting electrical properties. At low energy the charge carriers in graphene behave like massless Dirac fermions [18] which move with a Fermi velocity of about 1/300 the speed of light with a linear spectrum [20]. Apart from the confirmation of earlier theories, the experiments produced continuous graphene flakes of high quality which give rise to room-temperature ballistic transport [21, 22]. Electrons in graphene can travel distances on the order of micrometer without scattering. Combined with excellent mechanical strength [23], this makes graphene a very promising material for future electronic devices.

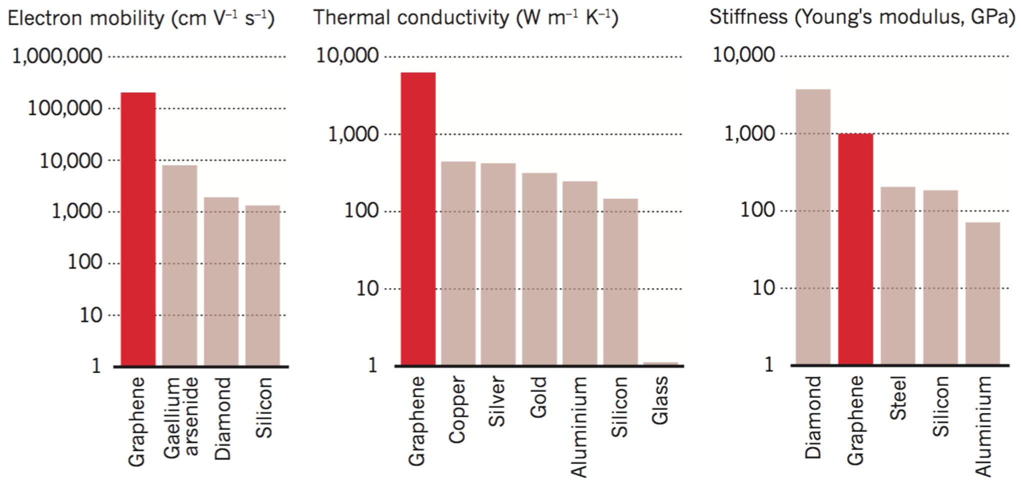


Fig. 1.2.1: The electronic, thermal and mechanical properties of graphene compared to other materials at room temperature. Note the logarithmic scale. Adapted from [24].

1.2 Graphene nanoelectronics

Graphene has many properties which make it favorable for use in electronic devices. Figure 1.2.1 shows a comparison of the electronic, thermal and mechanical properties of graphene and other materials. Charge carriers in graphene show very high mobility with no backscattering which could make it suitable for extremely fast devices. In this regard, it's similar to carbon nanotubes, but its 2D structure makes it more compatible with existing devices. Graphene absorbs only 2.3% of visible light. Combined with its great mechanical properties, this makes it a good candidate for transparent flexible electronics like screens or solar cells.

Because it lacks a band gap, graphene is not a semiconductor in the classical sense. It is sometimes also referred to as a semimetal. While it has excellent electronic properties, its gapless band structure has a downside: it makes the confinement of electrons a difficult proposition. Klein tunneling allows electrons to pass right through conventional barriers [25, 26], thus making it difficult to switch off electric current. This is a problem for traditional electronic devices, like transistors, which must be equally good at suppressing current as they are at conducting it. This makes the study of electron confinement in graphene an interesting topic. One way of changing the electronic properties of a material is through the application of mechanical deformations: strain engineering.

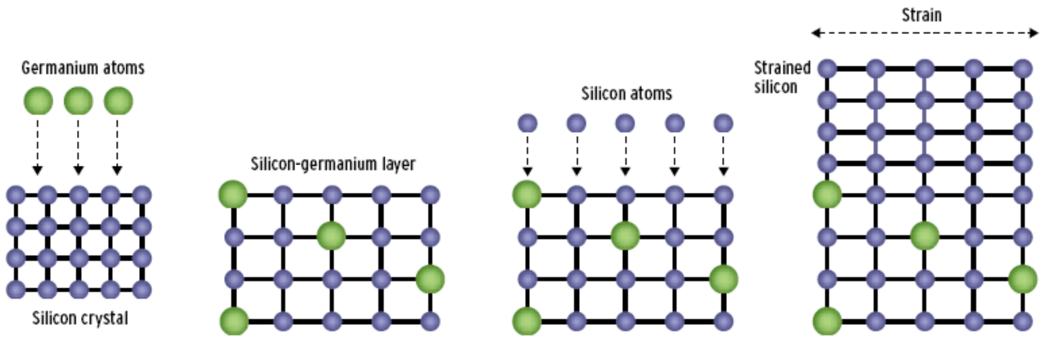


Fig. 1.2.2: Illustration of strained silicon. Two materials with mismatched lattice spacings are grown on top of one another. The interface strains the crystal lattice, increasing the inter-atomic distance, thus improving carrier mobility [27].

Strain has long been used to enhance the electrical properties of semiconductors. It is an integral part of modern silicon-based field-effect transistors. Germanium generally has superior electronic properties compared to silicon, especially when it comes to carrier mobility. However, silicon is the preferred material due to its abundance and established manufacturing facilities. Nevertheless, it's still possible to augment silicon with a bit of germanium. The goal is to stretch out the silicon lattice (increase the inter-atomic spacing), which leads to improvement in carrier mobility. This is illustrated in Fig. 1.2.2. The lattice mismatch between silicon and germanium is actually too large to simply grow one material on top of the other. Instead, a silicon substrate is doped with germanium. The resulting SiGe has a lattice spacing which is 1% larger than regular Si. Pure silicon is then grown on top of the SiGe substrate matching the underlying lattice as it is being deposited, thus creating strained silicon.

Strained silicon may be created in ways other than the one just described, but the goal is usually the same: the improvement of carrier mobility. The process is extremely beneficial which is why it is used in the manufacturing of nearly all semiconductor processors today, from supercomputers to mobile devices. Most modern processes apply so called "dual-stress liners" which apply both tensile and compressive strain specifically targeting n-type or p-type transistors to boost electron or hole mobility. Overall, the benefits of strain engineering are significant: silicon strain of less than 1% results in 20-40% improvement of transistor performance [28]. Silicon itself is quite elastic and able to sustain strain up to 5%. However, this pales in comparison to graphene which has been shown to sustain elastic strain up to a maximum

Fig. 1.2.3: "It would take an elephant, balanced on a pencil, to break through a sheet of graphene the thickness of Saran Wrap." [29]

of 25% [23]. This presents significant leeway for strain engineering.

Of course, the goal for graphene would be different than traditional semiconductors. Instead of improving the already excellent charge mobility, the main motivation for strain in graphene is finding new ways of confining electrons. It has been shown that strain can be used to open a band gap in graphene [30]. This would enable the use of traditional electric barriers. However, the amount of strain required to achieve this is near the limits of graphene's elastic range which makes this impractical, if not impossible. Fortunately, even without opening a band gap, strain in graphene offers interesting ways of controlling the behavior of electrons.

It is well established that magnetic fields can control the motion of electrons in a material, but the realization of strong non-homogeneous magnetic fields can be difficult. Graphene offers an interesting alternative which will be explored in this thesis: in the presence of mechanical strain, electrons in graphene behave as if they are inside a magnetic field [31]. This strain-induced effect, called the *pseudo-magnetic field*, can be used to great effect thanks to the very high strain-tolerance of graphene. Achievable pseudo-magnetic fields were measured on the order of hundreds of Tesla [32–34], which is significantly higher than real magnetic fields achievable in the laboratory. This pseudo-magnetic field also leads to a pseudo-quantum Hall effect [35]. Landau quantization of the electronic spectrum was observed by scanning tunneling microscopy. With such large strain-induced pseudo-magnetic fields, strain engineering becomes a favorable technique of controlling the electronic properties of graphene [36].

The two-dimensional nature of graphene also enables alternative ways of applying strain. As described earlier for classical semiconductors, lattice mismatches may be used to apply strain along the graphene plain. However, the atomically thin structure makes out-of-plane deformations an interesting approach. This can be achieved by applying a force perpendicular to the graphene plane, either mechanically using a tip or gas pressure [37, 38]. It was shown experimentally that an external nonuniform electric field is able to induce local deformations [39], thus a pseudo-magnetic field could even be induced through an electric field.

1.3 Small-scale relativity

From the point of view of fundamental physics, graphene is quite interesting because of the peculiar nature of its electrons. The band structure does not resemble that of massive particles usually governed by the Schrödinger equation. Conventional semiconductors host electrons which can be described by a quadratic dispersion relation tied to an effective mass. In contrast, the low-energy excitations in graphene obey a linear energy-momentum relation governed by the Dirac equation.

The linear dispersion is analogous to photons with the difference that Dirac fermions in graphene move at Fermi velocity which is about 300 times slower than the speed of light. Effectively, electrons in graphene behave like ultra-relativistic massless particles. This brings it into an interesting realm where quantum mechanics intersect with special relativity. Experiments in this field have usually been difficult due to very high energy requirements. However, the exotic behavior of electrons in graphene has presented an opportunity to test out some fundamental theories at a smaller scale. Essentially, graphene serves as a condensed matter emulator of relativistic quantum mechanics.

Even before it was experimentally isolated, graphene was considered as an analog of $(2+1)$ dimensional electrodynamics [17, 41, 42]. Examples of relativistic properties include the Klein paradox [25, 26] and Zitterbewegung [43].

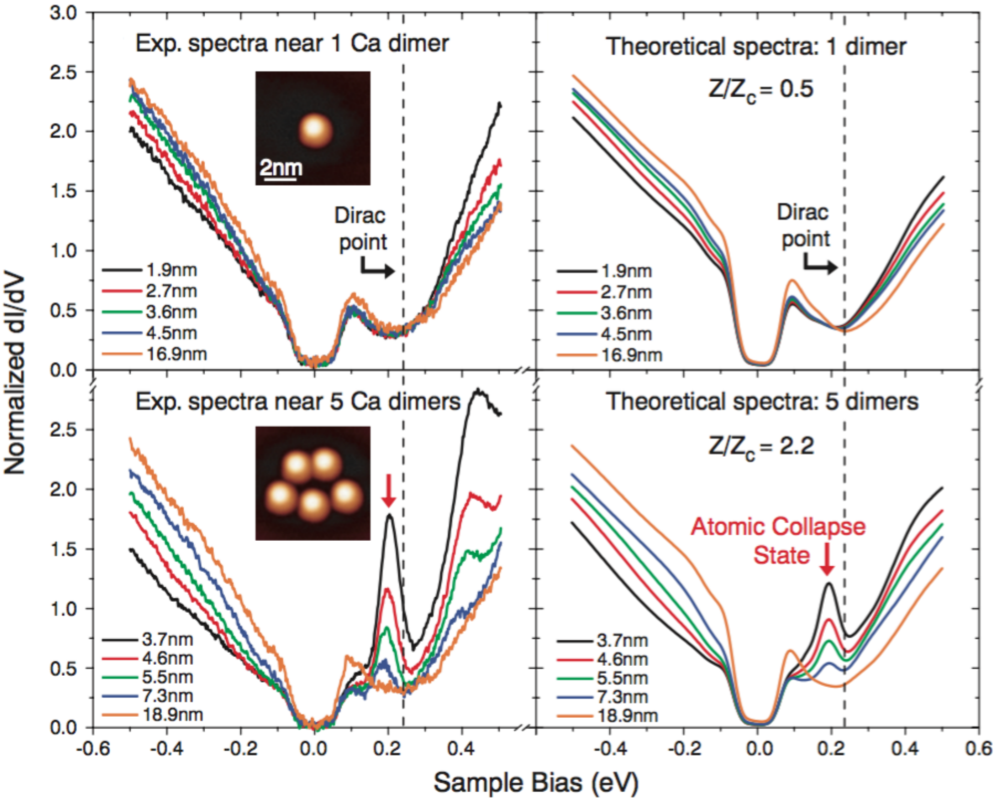


Fig. 1.3.1: Experimental signature of atomic collapse in graphene. (left) STM measurements and (right) calculations of the differential conductance dI/dV (local density of states) for graphene with 1 and 5 charge centers: Ca dimers as shown in the inset. The various curves correspond to different distances from the center of the charge. Adapted from Wang et al. [40]

The detection of the anomalous integer quantum Hall effect served as the definitive demonstration of the relativistic nature of carriers as well as the signature of single-layer graphene [18, 19].

This thesis studies the effect of strong electric fields in graphene and their ability to localize electrons. While regular electric potentials only have a marginal effect due to Klein tunneling, the situation changes in the presence of strong charge centers which can tap into the supercritical regime. When the critical value is exceeded, quasi-bound states are formed. This effect is analogous to a longstanding prediction of quantum electrodynamics (QED) which states that extremely heavy atoms (with charge $Z > 170$) will undergo a process called *atomic collapse*. Electronic states sink into the positron continuum triggering the spontaneous creation of electron-positron pairs [44–

46]. The extremely high charge requirements have prevented the observation of this phenomenon notwithstanding several attempts based on collisions of heavy ions [47, 48].

Graphene has enabled the same physics to be observed at a much lower scale ($Z \approx 1$) by depositing impurities on the surface of graphene [40]. This artificial nucleus had enough charge to trigger collapse, therefore trapping electrons in a sequence of quasi-bound states. Experimentally, the supercritical state was observed as a resonance in the local density of states close to the charge center (see Fig. 1.3.1). While interesting from a fundamental point of view, the collapse states also present an alternative way to localize electrons in graphene. In addition, graphene experiments allow for further study by varying other parameters such as the back-gate voltage, defects or the number of layers. The results of this exploration have an effect on more than just theoretical QED, but also the understanding of confinement and transport in graphene.

1.4 Simulating physics

The principal job of theoretical physics is to faithfully model natural phenomena. A good model is able to explain previously observed behavior and make predictions which can be verified by future observations. Using the knowledge of fundamental particle interactions and equations of motion, the behavior of a system can be predicted with great accuracy. However, actually solving these fundamental equations for a realistically large number of particles becomes extremely complicated. A workaround can be found by designing a sufficiently simplified model, which can be solved analytically while still retaining the desired behavior. However, that approach may fail when studying the interplay of phenomena which compete on similar energy scales or when the interaction of many degrees of freedom must be considered. In such cases, numerical methods present the only solution.

In the field of condensed matter theory, analytical methods can easily model infinitely large systems by approximating bulk properties. If the description of only a handful of atoms is required, first principal methods offer very good accuracy. A middle ground is found in the form of the tight-binding method which approximates some features but still gives a very detailed description of the system. It can be used to model systems on the order of millions of atoms. This presents a large scale numeric problem. Over the last few decades, Moore's Law has ensured that computer performance has increased up to the point that such problems can be solved. Moreover, new computing devices such as GPUs help greatly in cases where the problem can be decomposed into many parallel tasks. Nevertheless, algorithm efficiency is still incredibly important. Fast computational hardware is wasted if the code being executed does not take full advantage of all the available resources.

Code reusability is another important requirement. Instead of writing (or copy/pasting) code for each new problem, the approach can be generic and applicable to a whole class of problems. Specializations can be made as needed while still maintaining a generic core. Part of the problem of writing generic algorithms for physics simulations is how to effectively translate the language of physics into computer code. Pure computation is usually straightforward, but sometimes setting up the problem can be the more challenging part, especially when many potentially interacting parameters are in play. The abstractions that work for physics on paper may not map nicely to code on the screen. In order to produce good generic code for physics, we must develop new abstractions specifically tuned for the computer.

In the course of this thesis, a software framework was developed in order to aid the numerical work. While it started as specialized code for graphene, it has since evolved into a versatile numerical package for arbitrary tight-binding models. It is now available as an open-source project[*]. The framework greatly simplifies the process of constructing models while still being generic and applicable to any tight-binding system. By taking care of the numerical details, the framework frees the user to concentrate on the physics: the quantum properties of the model. The code package also contains multiple computational tools with an accent on performance. In order to visualize and make sense of the results, data post-processing and plotting functions are also included.

1.5 Outline

Graphene has many superior properties compared to traditional semiconductors, however the lack of a band gap makes electron confinement a challenge. Good conductivity does not matter if the current cannot be turned off as needed. Klein tunneling renders traditional electric barriers ineffective, thus an alternative approach is needed. To that end, this thesis explores ways of controlling the electronic properties of graphene using mechanical strain as

[*] https://github.com/dean0x7d/pybinding

well as supercritical electric fields. The properties are investigated mainly using the numerical tight-binding approach for which a code framework was developed.

Chapter 1 introduces the main topics of interest in this thesis.

Chapter 2 reviews the tight-binding model and its application to graphene. The low energy approximation of graphene is examined in order to derive the effective Dirac Hamiltonian which governs the behavior of relativistic massless fermions.

Chapter 3 is devoted to the kernel polynomial method. Based on Chebyshev polynomial expansion, it is a very efficient numerical tool which we shall use for computation throughout this thesis.

Chapter 4 presents the main ideas behind the code framework which was developed to aid the numerical work in this thesis. We also develop specific tweaks which can be made to improve the performance of the kernel polynomial method both for CPUs and GPUs.

Chapter 5 is devoted to the pseudo-magnetic field which arises in graphene as a result of mechanical strain. We investigate several models of the strain-induced field and suggest corrections which should be applied depending on the strain intensity and geometry.

In **Chapter 6** we study the specific case of out-of-plain strain due to a Gaussian-shaped bump in a graphene flake. This geometry generates a very distinct pseudo-magnetic field with six-fold symmetry. The shape confines electrons along the armchair directions of graphene, while the zigzag directions carry valley-polarized current.

Chapter 7 examines in-plane triaxial strain in bilayer graphene. The generated pseudo-magnetic field is uniform with a peculiar side-effect: the zero Landau level is missing in only one of the layers. Effectively, this means that low-energy electrons are localizes to a single layer. The affected layer can be switched by rotating the strain direction by 60°.

Chapter 8 introduces the phenomena of atomic collapse in graphene. We study the effect in the presence of supercritically charged artificial nuclei which can be constructed using external impurities or charged vacancies. The vacancy is of particular interest due to the persistence of the induced charge and additional effects introduced by broken sublattice symmetry.

Chapter 9 studies the atomic-collapse effect induced solely by the sharp tip of a scanning tunneling microscope. By adjusting the backgate voltage, the induced charge can be continuously tuned from the subcritical to the supercritical regime. The transition is accompanied by whispering-gallery modes, a phenomena originating in optics, which presents an alternate confinement mechanism compared to collapse.

In **Chapter 10** we investigate supercritical charge centers in the presence of a magnetic field. We show that the atomic collapse effect persists even after the field is activated. The previously observed collapse resonances are replaced with a series of Landau level anti-crossings as the new signature of collapse. The Landau levels also display anomalous scaling with regard to magnetic field strength in the region of the collapse.

Chapter 11 considers the creation of artificial "collapsing" molecules made up of supercritical nuclei.

Finally, in **Chapter 12** we briefly summarize the results of this thesis and discuss future outlook.

Appendix A contains a brief overview of the most important features of the tight-binding code framework, called *Pybinding*, which was developed along with this thesis. A user guide is presented with general workflow information and several code examples.

Part I

Theoretical framework and numerical methods

CHAPTER 2

The tight-binding model of graphene

The tight-binding model is an approximate approach of calculating the electronic band structure of solids using a basis of localized atomic orbitals. Historically, it has also been referred to as the linear combination of atomic orbitals (LCAO) or Bloch method. Slater and Koster provided a systematic procedure for the formulation of tight-binding Hamiltonians using the famous *Slater-Koster* table of hopping integrals [49].

The approach does not require first principals calculations, but instead simply uses parameterized matrix elements, whose values are chosen such that it reproduces the experimental results or the outcome of first-principles calculations. In contrast to ab initio calculations which are restricted to relatively small computational unit cells, the tight-binding model can scale to large system sizes on the order of millions of atoms.

In the simple empirical scheme, a tight-binding Hamiltonian is constructed by closely matching the physical system [50]. Atomic orbitals are bound to lattice sites and electrons are allowed to hop between them as determined by the hopping integrals. Alternatively, the tight-binding model can also be produced as a rigorous approximation of density functional theory [51, 52]. It can also provide the basis for the construction of many-body theories such as the Hubbard model and the Anderson impurity model [53, 54].

Even withing the single-particle approximation, the tight-binding model is applicable to a wide variety of systems and phenomena in quantum physics (classical semiconductors, superconductivity, topological insulators, etc.). In the current work, we shall use this model to describe the π bands of carbon atoms in graphene.

2.1 The tight-binding model

If we consider electrons to be tightly localized around each atomic position, the probability to find an electron on a neighboring atom is very small. Therefore we can expand the Bloch wave function of the crystal by a linear combination of local functions,

$$\Psi_{k,n}(\vec{r}) = \frac{1}{\sqrt{N}} \sum_{\vec{R}} \Phi_n(\vec{R}, \vec{r}) \mathrm{e}^{i\vec{k}\cdot\vec{r}}, \tag{2.1.1}$$

where $\vec{R}$ is the real-space lattice vector. The functions $\Phi_n(\vec{R}, \vec{r})$ are called Wannier functions and N is the number of unit cells in the crystal. In general a unit cell may consist of multiple atoms where the sublattices are labeled by μ, and each atom may have several orbital degrees of freedom α and spins s. The individual wavefunction are then $\phi_{\alpha,s}(\vec{r} - \vec{\delta}_\mu - \vec{R})$, where $\vec{\delta}_\mu$ is the offset of sublattice μ within the unit cell $\vec{R}$. The resulting on-site Bloch wave function is,

$$\Psi_{k,n}(\vec{r}) = \frac{1}{\sqrt{N}} \sum_{\vec{R}} \sum_{\mu,\alpha,s} \phi_{\alpha,s}(\vec{r} - \vec{\delta}_\mu - \vec{R}) \mathrm{e}^{i\vec{k}\cdot\vec{r}}, \tag{2.1.2}$$

which obeys the Bloch theorem

$$\Psi_{k,n}(\vec{r} + \vec{R}') = \mathrm{e}^{i\vec{k}\cdot\vec{R}'} \Psi_{k,n}(\vec{r}). \tag{2.1.3}$$

In order to find the Hamiltonian matrix element we start with the Schrödinger equation,

$$H\Psi_{k,n}(\vec{r}) = E\Psi_{k,n}(\vec{r}). \tag{2.1.4}$$

The crystal wave function $\Psi_k(\vec{r})$ can be expanded in the basis of the on-site Bloch wave functions,

$$\Psi_k(\vec{r}) = \sum_i c_{k,i} \Psi_{k,i}(\vec{r}). \tag{2.1.5}$$

Plugging this expansion into the Schrödinger equation and using the orthogonality of the Bloch wave functions we obtain,

$$\sum_{i,j} c^*_{k,j} c_{k,i} \left[\Psi^*_{k,j}(\vec{r}) H \Psi_{k,i}(\vec{r}) - E_k \Psi^*_{k,j}(\vec{r}) \Psi_{k,i}(\vec{r}) \right] = 0. \tag{2.1.6}$$

The Hamiltonian matrix element can be defined as,

$$H_{i,j}(\vec{k}) = \frac{1}{N}\sum_{\vec{R},\vec{R}'} \mathrm{e}^{i\vec{k}\cdot(\vec{R}-\vec{R}')} \int d\vec{r}\phi_j^*(\vec{r}-\vec{R})H\phi_i(\vec{r}-\vec{R}) \tag{2.1.7}$$

and the overlap matrix elements are defined as,

$$S_{i,j}(\vec{k}) = \frac{1}{N}\sum_{\vec{R},\vec{R}'} \mathrm{e}^{i\vec{k}\cdot(\vec{R}-\vec{R}')} \int d\vec{r}\phi_j^*(\vec{r}-\vec{R})\phi_i(\vec{r}-\vec{R}). \tag{2.1.8}$$

In general, atomic orbitals centered at different sites do not have to be orthogonal, thus the overlap integral may not be zero. However, this is usually a very small value which can be discarded in most cases. Here, we shall consider the purely orthogonal tight-binding model.

Substituting Eqs. (2.1.7) and (2.1.8) in Eq. (2.1.6), we find the energy spectrum for fixed $\vec{k}$ as

$$E_k = \frac{\sum_{i,j} H_{i,j}(\vec{k})c_{k,j}^* c_{k,i}}{\sum_{i,j} S_{i,j}(\vec{k})c_{k,j}^* c_{k,i}}. \tag{2.1.9}$$

Minimizing the energy using $\partial E_k/\partial c_{k,i}^*$, we obtain the secular equation as

$$\sum_i \left[H_{i,j}(\vec{k}) - E_k S_{i,j}(\vec{k})\right] c_{k,i} = 0. \tag{2.1.10}$$

2.2 Graphene and the continuum limit

Graphene is a single layer of sp^2 hybridized carbon atoms arranged in a honeycomb lattice, as illustrated in Fig 2.2.1(a). A unit cell consists of two atoms, denoted A and B. The positions of all sites of sublattice $\mu = A, B$ is described by the discrete translation:

$$\vec{R}_\mu(n_1, n_2) = n_1\vec{a}_1 + n_2\vec{a}_2 + \vec{\delta}_\mu. \tag{2.2.1}$$

where n_1 and n_2 are integers. The lattice vectors of a triangular Bravais lattice are

$$\vec{a}_1 = a\begin{bmatrix}1\\0\end{bmatrix}, \qquad \vec{a}_2 = \frac{a}{2}\begin{bmatrix}1\\\sqrt{3}\end{bmatrix}, \tag{2.2.2}$$

where $a = \sqrt{3}a_{cc}$ is the unit length and $a_{cc} = 0.142$ nm is the carbon-carbon distance. The vectors $\vec{\delta}_A = [0,0]^T$ and $\vec{\delta}_B = [0, -a_{cc}]^T$ define the offset of a carbon atom within the unit cell.

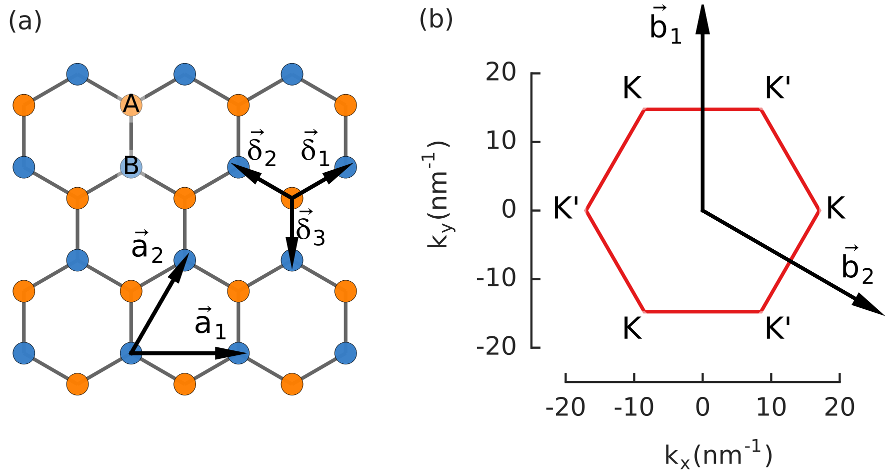


Fig. 2.2.1: (a) Illustration of monolayer graphene with sublattices A and B, lattice vectors $\vec{a}_i$ and nearest neighbor vectors $\vec{\delta}_i$. (b) The Brillouin zone with reciprocal lattice vectors $\vec{b}_i$.

The reciprocal lattice vectors satisfy the relation $\vec{a}_i \cdot \vec{b}_j = 2\pi\delta_{i,j}$ and are given by

$$\vec{b}_1 = \frac{2\pi}{3a_{cc}} \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \qquad \vec{b}_2 = \frac{2\pi}{3a_{cc}} \begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix}. \tag{2.2.3}$$

The Brillouin zone is hexagonal with vertices in the K-points, as can be seen in Fig. 2.2.1(b). We can distinguish two sets of inequivalent points, K and K', because they cannot be connected by reciprocal lattice vectors. Two of the points are located on the k_x-axis with positions

$$\vec{K} = \frac{2\pi}{3a_{cc}} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \vec{K'} = \frac{2\pi}{3a_{cc}} \begin{bmatrix} -1 \\ 0 \end{bmatrix}. \tag{2.2.4}$$

The electron states around high-symmetry K and K' points, which represents the valley degeneracy of graphene. The two points are connected by time reversal symmetry.

The carbon atoms in graphene are sp^2 hybridized. Three of the four electrons in the outer-shell form strong σ bonds. These in-plane covalent bonds responsible for the structural properties of graphene, in particular its stiffness. The remaining electrons occupy the p_z orbital perpendicular to the graphene plane. They are free to hop between neighboring sites, giving rise to the low-energy π bands. Since there is only one electron per p_z orbital for each carbon atom, the Fermi level lies at zero energy. Therefore, to model the

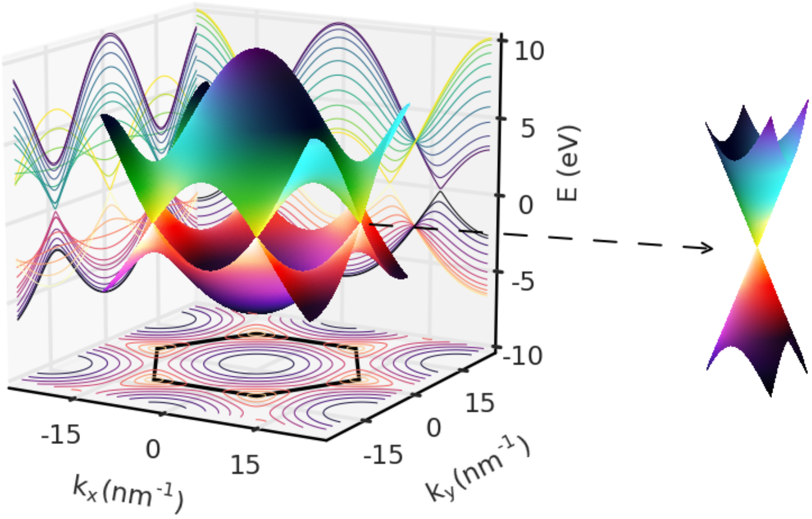


Fig. 2.2.2: Energy dispersion of graphene, as obtained by the next-nearest neighbor tight-binding method with $t = -2.8$ eV and $t_{nn} = 0.1$ eV. The black hexagon in the bottom projection indicates the outline of the first Brillouin zone. The valence and conduction bands touch at the vertices of the hexagon (K points). A close-up view of the linear dispersion around a K point is shown on the right.

electrical properties, it is enough to consider the nearest-neighbor interaction of p_z orbitals.

Given the normalized wavefunction $\varphi(\vec{R}_{A,B})$ for an isolated p_z orbital at carbon atoms A and B, the tight-binding Hamiltonian can be written down as,

$$H = t \sum_{\vec{R}_{A,B}} |\varphi(\vec{R}_A)\rangle \langle\varphi(\vec{R}_B)| + H.c., \tag{2.2.5}$$

where $t = -\langle\varphi_A|H|\varphi_B\rangle \approx -2.8$ eV is the hopping integral between nearest neighbors. In graphene, the overlap matrix is small and usually neglected $S = \langle\varphi_A|\varphi_B\rangle \approx 0$. Because both sublattices A and B host carbon atoms their onsite energy is equal, and it is convenient to set it to zero: $\langle\varphi_A|H|\varphi_A\rangle = \langle\varphi_B|H|\varphi_B\rangle = 0$. The Bloch wavefunction or sublattice $c = A, B$ is given by

$$\Psi_\mu(\vec{k}) = \frac{1}{\sqrt{N}} \sum_{\vec{R}_\mu} \mathrm{e}^{i\vec{k}\cdot\vec{R}_\mu} \varphi_\mu(\vec{R}_\mu), \tag{2.2.6}$$

where N is the total number of unit cells. In terms of the two-component wave function $\Psi = [\Psi_A, \Psi_B]^T$, the Hamiltonian can be expressed as,

$$H(\vec{k}) = \begin{bmatrix} 0 & h(\vec{k}) \\ h^*(\vec{k}) & 0 \end{bmatrix}, \tag{2.2.7}$$

where

$$h(\vec{k}) = t \sum_i \mathrm{e}^{i\vec{k}\cdot\vec{\delta}_i}. \tag{2.2.8}$$

Solving the Schrödinger equation $H\Psi = E\Psi$ as $\det(H - EI) = 0$, one obtains the dispersion relation, as was deduced by Wallace in 1947 [15]:

$$E(\vec{k}) = \pm t\sqrt{3 + 2\cos(k_x a) + 4\cos(k_x a/2)\cos(\sqrt{3}k_y a/2)}. \tag{2.2.9}$$

Taking the limit close to a K-points of the Brillouin zone, $\vec{k} = \vec{K} + \vec{q}$ with $|q| << |K|$, Eq. (2.2.8) can be simplified to $h(\vec{q}) = \hbar v_F(q_x \pm iq_y)$ and thus the effective low energy Hamiltonian is

$$H_{K,K'}(\vec{q}) = \hbar v_F \begin{bmatrix} 0 & q_x \pm iq_y \\ q_x \mp iq_y & 0 \end{bmatrix}, \tag{2.2.10}$$

where $v_F = 3a_{cc}t/(2\hbar)$ is the Fermi velocity. The dispersion relation now reveals the photon-like linear spectrum of low energy excitations in graphene,

$$E(\vec{q}) = \pm\hbar v_F|\vec{q}|. \tag{2.2.11}$$

Using terser notation, Eq. (2.2.10) can be expressed as

$$H(\vec{q}) = \eta\hbar v_F(q_x\sigma_x + \eta q_y\sigma_y), \tag{2.2.12}$$

where σ_x and σ_y are Pauli matrices and $\eta = \pm 1$ corresponds to the K and K' points.

An object is said to have *chirality* if it cannot be mapped to its mirror image using only rotations and translations. This is quite common for everyday objects like gloves, shoes, etc.

In graphene we can define the helicity operator as the projection of the momentum operator along the pseudo-spin direction,

$$h_{\vec{q}} = \vec{\sigma}\frac{\vec{q}}{|\vec{q}|}. \tag{2.2.13}$$

The helicity eigenvalue $h = \pm 1$ is a good quantum number for energy close to the K and K' points. For massless particles chirality is the same as helicity. Electrons in graphene due to the linear spectrum close to the Dirac points have well defined chirality. Since the helicity operator commutes with the Hamiltonian, it's possible to find a common basis of eigenstates, where helicity is a conserved quantity. It is possible to present the Hamiltonian using the helicity operator,

$$H = \eta \hbar v_F |\vec{q}| h_{\vec{q}}. \tag{2.2.14}$$

Here the dispersion relation becomes $E(\vec{q}) = \eta h \hbar v_F |\vec{q}|$. From here it's clear that the band index $\alpha = \pm 1$ can be determined as $\alpha = \eta h$.

The *Klein paradox* represents the perfect tunneling of a relativistic electron through wide barriers. It is a well-known phenomenon originating from quantum electrodynamics [25]. It also applies to the low-energy excitation in graphene where it was first observed experimentally [55].

If an electron travels along the x-direction it will have $k_y = 0$. The velocity operator is given by the Heisenberg equation $v_x \approx -i[x, H]/\hbar = \sigma_x$ and the change in velocity is $dv_x/dt = -i[\sigma_x, H]/\hbar = 2\sigma_z k_y$. Because $k_y = 0$, the velocity is a constant and therefore backscattering is not possible.

CHAPTER 3

The kernel polynomial method

Exactly solving a tight-binding problem implies the diagonalization of the Hamiltonian matrix. This produces eigenvalues and eigenvectors required to calculate the desired physical properties. However, the computational resources used by diagonalization algorithms scale up rapidly with system size which makes it challenging to model realistically large systems. This has placed great importance on the development of more efficient exact algorithms as well as novel approximative methods.

When a Hamiltonian is represented by a dense matrix of dimension D, memory requirements scale as D^2 while the computation time required to solve for all the eigenvalues scales as D^3. This cubic scaling blows up quickly, even for moderately sized systems on the order of tens of thousands of lattice sites. In an effort to solve larger problems, the first obvious step is to switch to a sparse matrix representation of the Hamiltonian. Sparse matrices can be stored in various formats, but the common trait is that the memory requirements scale linearly with the number of non-zero elements N_{nz}. Tight-binding models are very well suited for this because $N_{nz} \approx mD$, where $m \ll D$, thus making a substantial improvements over the D^2 scaling of dense matrices. The number m is proportional to the average number of hoppings per lattice site.

Along with the storage format, the strategy of solving large matrices must also change. A good choice for sparse Hermitian matrices is the Lanczos algorithm [56]. Instead of solving for all the eigenvalues and eigenvectors, sparse diagonalization targets a small subset of size k. The Lanczos algorithm strictly requires that $k < D$ but for best performance $k \ll D$ is preferred.

The problem complexity scales with kN_{nz}, but the procedure suffers from numerical instabilities which require additional computational effort to overcome. All practical implementations must take this into account. One of the most widely used variations is the implicitly restarted Lanczos method [57] which is implemented in ARPACK [58].

An alternative sparse diagonalization scheme is offered by the FEAST algorithm [59]. It differs significantly from traditional techniques like the Lanczos procedure, instead taking its inspiration from the density-matrix representation and contour integration in quantum mechanics. When solving a series of eigenvalue problems which are close to one another, as is the case for band structure calculations, the results of the previous calculation can be used as the starting point for the next. The algorithm also features natural parallelism where different eigenvalues can be computed separately without overlap.

No matter which sparse diagonalization algorithm is used, they all produce exact solutions in the form of a small set of eigenvalues and eigenvectors. These are used to calculate the desired physical properties. A fundamentally different approach is to set aside the requirement for exact solutions (avoid diagonalization altogether) and instead use approximative methods to calculate the properties of interest. This chapter will present one such approach in the form of the kernel polynomial method (KPM) based on Chebyshev expansion.

One of the most desirable properties of KPM is computation time which scales linearly with sparse matrix size N_{nz}. The procedure is numerically very stable so no additional computation is wasted [60]. The approximative nature of the method presents an opportunity for additional performance tuning. Results may be computed very quickly at low accuracy to get an initial estimate for the problem at hand. Once final results are required, the accuracy can be increased at the cost of longer computation time.

KPM has been successfully applied to large quantum problems in solid-state physics [61, 62]. It can be used to approximate the local density of states within a tight-binding framework [63, 64]. Green's function can be efficiently expanded in a series of Chebyshev polynomials [65, 66]. It also allows for easy coupling to baths which are also described by Chebyshev expansions [67]. One of the great benefits of this method is that spatially dependent properties such as the local density of states or Green's function are calculated separately for each spatial position. This means that localized properties can be computed extremely quickly, but it also allows a trivial parallel implementation when the area of interest is large.

The domain of KPM can be seen as orthogonal to traditional diagonalization algorithms. Sparse diagonalization produces results for a very small energy range (eigenvalues) but does so for all positions simultaneously (eigenvectors). KPM computes only a single position at a time, but produces results for the entire energy spectrum at once. In this way, the two approaches complement each other nicely. For example, KPM may be used to quickly compute the local density of states for the entire energy spectrum. Once interesting features are identified in the LDOS, sparse diagonalization can be used to get the full spatial picture only at the energy positions of interest. This scheme is extremely beneficial since all sparse diagonalization algorithms function much more efficiently when only a small, highly targeted, set of eigenstates is required.

This orthogonal relationship of diagonalization and KPM just goes to demonstrate that there are no ultimate one-size-fits-all tools. Numerical tools should be picked specifically for the job at hand and it's even better if different approaches can be combined. Of course, that does require having a well stocked toolbox, where KPM has an important place.

3.1 Chebyshev expansion

In general, the expansion of a given function $f(x)$ in terms of a polynomial set $p_n(x)$ can be introduced as

$$f(x) = \sum_{n=0}^{\infty} \alpha_n p_n(x), \tag{3.1.1}$$

where the expansion coefficients α_n are proportional to the scalar product of f and p_n,

$$\alpha_n = \langle f|p_n\rangle \, h_n, \tag{3.1.2}$$

and $h_n = 1/\langle p_n|p_n\rangle$ denotes the inverse squared norm. The polynomial set $p_n(x)$ must fulfill the orthogonality relations

$$\langle p_n|p_m\rangle = \frac{\delta_{n,m}}{h_n}. \tag{3.1.3}$$

The scalar product $\langle f|g\rangle$ between functions $f(x)$ and $g(x)$ is defined on the interval $[a,b]$ with a positive weight $w(x)$,

$$\langle f|g\rangle = \int_a^b w(x)f(x)g(x)dx. \tag{3.1.4}$$

An expansion can be used to approximately reconstruct the desired function. It shifts the problem from the direct solution of $f(x)$ to the computation of expansion moments α_n. Given a good choice of the polynomial set $p_n(x)$, the moment computation can be implemented in a very efficient way.

All types of orthogonal polynomials can be used for the expansion, but Chebyshev polynomials stand out as a very good choice for numerical applications. They have good convergence properties and a close relation to the Fourier transform which allows the partial reuse of existing numerical tools. Furthermore, the recursive relationship between Chebyshev polynomials enable the development of a very efficient iterative routine for computing the expanded function.

Chebyshev polynomials

Chebyshev polynomials of the first kind are denoted by T_n, while the second kind are U_n. They can be expressed in terms of trigonometric functions as

$$T_n(x) = \cos[n \arccos(x)], \tag{3.1.5}$$

$$U_n(x) = \frac{\sin[(n+1)\arccos(x)]}{\sin[\arccos(x)]}. \tag{3.1.6}$$

From here it is quite apparent that they are only defined on the interval $[a, b] = [-1, 1]$. The two kinds of Chebyshev polynomials are distinguished by their weight functions, $w(x) = (\pi\sqrt{1-x^2})^{-1}$ and $w(x) = \pi\sqrt{1-x^2}$, for the first and second kind, respectively. Plugging the weight function into the scalar product (3.1.4) yields,

$$\langle f|g\rangle_1 = \int_{-1}^{1} \frac{f(x)g(x)}{\pi\sqrt{1-x^2}} dx, \tag{3.1.7}$$

$$\langle f|g\rangle_2 = \int_{-1}^{1} \pi\sqrt{1-x^2} f(x)g(x)dx, \tag{3.1.8}$$

and from there the orthogonality relations read,

$$\langle T_n|T_m\rangle_1 = \frac{1+\delta_{n,0}}{2}\delta_{n,m}, \tag{3.1.9}$$

$$\langle U_n|U_m\rangle_2 = \frac{\pi^2}{2}\delta_{n,m}. \tag{3.1.10}$$

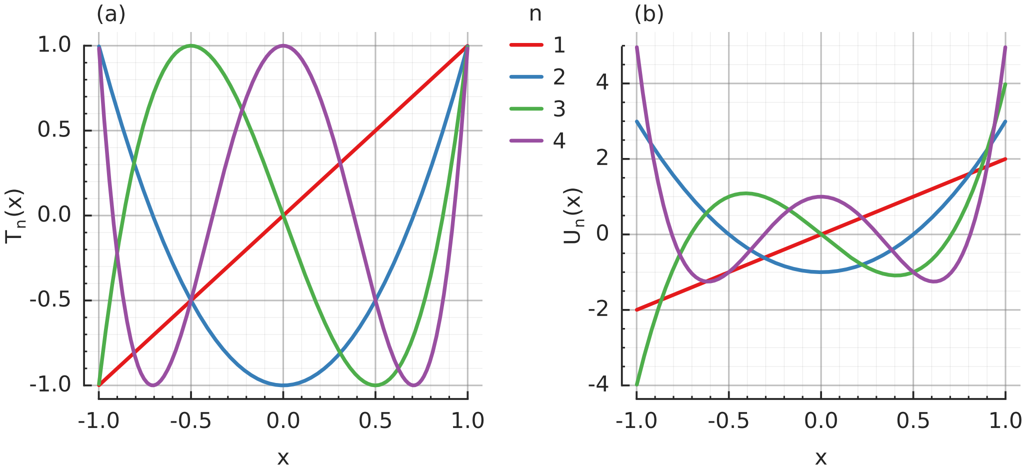


Fig. 3.1.1: Line plot of the Chebyshev polynomials of (a) the first kind $T_n(x)$ and (b) the second kind $U_n(x)$ for the first few values of the integer n. Note that they are only defined for $x \in [-1, 1]$.

The most interesting aspect of these polynomials are the recursive relations, which are the key to an efficient iterative numerical implementation of KPM,

$$\begin{gathered} T_0(x) = 1, \quad T_{-1}(x) = T_1(x) = x, \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \end{gathered} \tag{3.1.11}$$

and

$$\begin{gathered} U_0(x) = 1, \quad U_{-1}(x) = 0, \\ U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x). \end{gathered} \tag{3.1.12}$$

The product relations will also prove to be useful,

$$2T_m(x)T_n(x) = T_{m+n}(x) + T_{m-n}(x), \tag{3.1.13}$$

$$2(x^2 - 1)U_{m-1}(x)U_{n-1}(x) = T_{m+n}(x) - T_{m-n}(x). \tag{3.1.14}$$

The polynomials are related by the Hilbert transformations,

$$\mathcal{P} \int_{-1}^{1} \frac{T_n(y)dy}{(y-x)\sqrt{1-y^2}} = \pi U_{n-1}(x), \tag{3.1.15}$$

$$\mathcal{P} \int_{-1}^{1} \frac{\sqrt{1-y^2}U_{n-1}(y)dy}{y-x} = -\pi T_n(x), \tag{3.1.16}$$

where $\mathcal{P}$ denotes the principal value.

Expansion moments

With knowledge of Chebyshev polynomials, we can proceed to plug T_n into expansion (3.1.1) and determine the coefficients. The expansion reads

$$f(x) = \sum_{n=0}^{\infty} \frac{\langle f|T_n\rangle_1}{\langle T_n|T_n\rangle_1} T_n(x) = \alpha_0 + 2\sum_{n=1}^{\infty} \alpha_n T_n(x), \tag{3.1.17}$$

where the coefficients are

$$\alpha_n = \langle f|T_n\rangle_1 = \int_{-1}^{1} \frac{f(x)T_n(x)}{\pi\sqrt{1-x^2}} dx. \tag{3.1.18}$$

Note that the form of expansion (3.1.17) takes into account the special nature of α_0 according to Eq. (3.1.9).

The problem of reconstructing the function $f(x)$ relies entirely on calculating the moments α_n. It is preferable to avoid the integration in Eq. (3.1.18) and instead take advantage of the recursive relations (3.1.11). To this end, the basic Chebyshev expansion (3.1.17) is rearranged slightly,

$$f(x) = \sum_{n=0}^{\infty} \frac{\langle f|\phi_n\rangle_2}{\langle \phi_n|\phi_n\rangle_2} T_n(x) = \frac{1}{\pi\sqrt{1-x^2}} \left(\mu_0 + 2\sum_{n=1}^{\infty} \mu_n T_n(x) \right) \tag{3.1.19}$$

where the coefficients are now

$$\mu_n = \langle f|\phi_n\rangle_2 = \int_{-1}^{1} f(x)T_n(x) dx, \tag{3.1.20}$$

and the orthogonal functions are

$$\phi_n = \frac{T_n(x)}{\pi\sqrt{1-x^2}}. \tag{3.1.21}$$

Note that this rearrangement uses the second scalar product (3.1.8) instead of the first (3.1.7), with orthogonality relations

$$\langle \phi_n|\phi_m\rangle_2 = \frac{1+\delta_{n,0}}{2} \delta_{n,m}. \tag{3.1.22}$$

The set of modified moments μ_n defined in Eq. (3.1.20) are the core of this expansion method. They are directly tied to Chebyshev polynomials of the first kind which allows the recursive relations (3.1.11) to apply equivalently to μ_n. For the numerical implementation this means that only the first two moments need to be constructed from scratch, while the rest can be computed iteratively.

Rescaling the problem

Chebyshev polynomials are only defined on the interval $[-1, 1]$ and in turn so are the expansion moments (3.1.20). In order to apply the expansion to tight-binding problems, the Hamiltonian matrix and energy will need to be rescaled to the same $[-1, 1]$ interval, according to

$$\tilde{H} = \frac{H - bI}{a}, \qquad \tilde{E} = \frac{E - b}{a}, \tag{3.1.23}$$

where the scaling factors a and b are determined from the extremal eigenvalues of the Hamiltonian,

$$a = \frac{E_{max} - E_{min}}{2 - \epsilon}, \qquad b = \frac{E_{max} + E_{min}}{2}. \tag{3.1.24}$$

The parameter ϵ is a small number used to avoid instabilities which arise if the spectrum exceeds the boundaries $[-1, 1]$. This also means that the minimum and maximum eigenvalues do not need to be determined exactly.

A good way to determine the bounds for a Hermitian matrix is using the Lanczos iterative algorithm [56]. Since accuracy is not essential, the computation requires a small number of iterations. The Lanczos procedure naturally seeks the extremal eigenvalues first and since only two are required, the implementation does not need to take into account loss of orthogonality issues which would usually apply for longer-running procedures of this type. The parameter ϵ can be adjusted according to the accuracy of the eigenvalue computation. Typically, ϵ is on the order of 0.01 for bounds determined using around 100 Lanczos iterations.

Local density of states

An example of the application of the Chebyshev expansion is the calculation of the local density of states. For a Hamiltonian matrix of dimension D and eigenvalues E_k, the local density of states at lattice site i is determined by

$$\rho_i(E) = \frac{1}{D} \sum_{k=0}^{D-1} | \langle i|k \rangle |^2 \delta(E - E_k). \tag{3.1.25}$$

Before expanding this function in terms of Chebyshev polynomials, the problem must first be rescaled using substitutions $H \to \tilde{H}$, $E \to \tilde{E}$, as introduced earlier. We adopt the notation where all rescaled quantities are shown with a tilde. Using the rescaled Eq. (3.1.25) as the function for the Chebyshev moments (3.1.20) gives:

$$\begin{aligned}\mu_n &= \int_{-1}^{1} \tilde{\rho}_i(\tilde{E}) T_n(\tilde{E}) d\tilde{E} \\ &= \frac{1}{D} \sum_{k=0}^{D-1} | \langle i|k\rangle |^2 T_n(\tilde{E}_k) \\ &= \frac{1}{D} \sum_{k=0}^{D-1} \langle i|T_n(\tilde{H})|k\rangle \langle k|i\rangle \\ &= \frac{1}{D} \langle i|T_n(\tilde{H})|i\rangle \,. \end{aligned} \tag{3.1.26}$$

The $\rho_i(E)$ function can be reconstructed by using these moments in Eq. (3.1.19). However, there is still the task of calculating the expectation values $\langle i|T_n(\tilde{H})|i\rangle$.

Moments as expectation values

Moments in the form of expectation values of Chebyshev polynomials of $\tilde{H}$ are quite common for various calculations, not just the local density of states. In the general from, this set of moments can be expressed as

$$\mu_n = \langle \beta|T_n(\tilde{H})|\alpha\rangle \,, \tag{3.1.27}$$

where $|\alpha\rangle$ and $|\beta\rangle$ are certain states of the system. In terms of the implementation for a real-space Hamiltonian matrix, the state $|\alpha\rangle$ is a unit vector with a single non-zero element at lattice site α.

Starting from $|\alpha\rangle$ an iterative process constructs states $|\alpha_n\rangle = T_n(\tilde{H})\,|\alpha\rangle$ using the recursion relations (3.1.11):

$$|\alpha_0\rangle = |\alpha\rangle \,, \tag{3.1.28}$$

$$|\alpha_1\rangle = \tilde{H}\,|\alpha_0\rangle \,, \tag{3.1.29}$$

$$|\alpha_{n+1}\rangle = 2\tilde{H}\,|\alpha_n\rangle - |\alpha_{n-1}\rangle \,. \tag{3.1.30}$$

The final moment is calculated from the scalar product

$$\mu_n = \langle \beta|\alpha_n\rangle \,. \tag{3.1.31}$$

In terms of implementation, a tight-binding Hamiltonian is usually a sparse matrix, while states $|\alpha_n\rangle$ are dense vectors. The repeated application of $\tilde{H}$

to states $|\alpha_n\rangle$ consists of sparse matrix-vector multiplications which scale linearly with problem size.

If the condition $|\beta\rangle = |\alpha\rangle$ is met, as is the case for the local density of states Eq. (3.1.26), the required number of iterations can be cut in half. By taking advantage of the product relations (3.1.13) two moments can be computed based on each new $|\alpha_n\rangle$,

$$\mu_{2n} = 2\langle\alpha_n|\alpha_n\rangle - \mu_0, \tag{3.1.32}$$

$$\mu_{2n+1} = 2\langle\alpha_{n+1}|\alpha_n\rangle - \mu_1. \tag{3.1.33}$$

This simplifies the computation from N to $N/2$ sparse matrix-vector multiplications. However, it does also add $N/2$ dot products for $\langle\alpha_m|\alpha_n\rangle$, which brings the total number of operations back to N. Fortunately, a dot product is computationally cheaper than a matrix-vector multiplication, so even though the computation time is not halved, the improvement is still quite significant.

The iterative procedure described here forms the core of the Chebyshev expansion method and it is the most computationally intensive part. Sparse matrix-vector multiplication can be implemented in many different ways depending on the sparse matrix format and the target computing device (CPU or GPU). Each implementation has different performance implications. In addition, for very large systems, the sparsity of the starting vector α_0 can be exploited for additional performance improvements.

3.2 Kernel polynomials

The expansion Eq. (3.1.19) exactly replicates function $f(x)$ by using an infinite series of polynomials. However, if a finite series is used instead, the expansion is only an approximation of $f(x)$. This is the case for numerical implementations where the order cannot possibly be infinite. For an efficient numerical solution, the key is to find a finite set of moments μ_n which gives the best estimate of the desired function.

The simplest approximation is to just truncate the infinite series, limiting the expansion to the first N moments,

$$f(x) \approx f_{trunc}(x) = \frac{1}{\pi\sqrt{1-x^2}}\left(\mu_0 + 2\sum_{n=1}^{N-1}\mu_n T_n(x)\right). \tag{3.2.1}$$

This approach has poor precision and is prone to fluctuations, known as Gibbs oscillations, especially for discontinuities or singularities of $f(x)$ and near points where the function is not continuously differentiable.

Gibbs oscillations can be mitigated by convolving the approximate function (3.2.1) with kernel polynomials $K_N(x, y)$,

$$f_{KPM}(x) = \langle K_N(x, y) | f_{trunc}(y) \rangle_2 = \int_{-1}^{1} \pi\sqrt{1 - x^2} K_N(x, y) f_{trunc}(y) dy. \tag{3.2.2}$$

This is the key concept behind the kernel polynomial method. The resulting $f_{KPM}(x)$ function is a more accurate approximation of the original function compared to the ordinary truncated expansion $f_{trunc}(x)$. The optimal kernel function $K_N(x, y)$ depends on the desired application, but the general form should match the Chebyshev expansion,

$$K_N(x, y) = g_0\phi_0(x)\phi_0(y) + 2\sum_{n=1}^{N-1} g_n\phi_n(x)\phi_n(y). \tag{3.2.3}$$

Here g_n are damping coefficients which depend on the order N of the approximation. Plugging this form of the kernel function into Eq. (3.2.2) gives a convenient form of $f_{KPM}(x)$ where the transformation $f_{trunc}(x) \to f_{KPM}(x)$ can be performed simply by introducing the substitution $\mu_n \to g_n\mu_n$,

$$f_{KPM}(x) = \frac{1}{\pi\sqrt{1 - x^2}} \left(g_0\mu_0 + 2\sum_{n=1}^{N-1} g_n\mu_n T_n(x) \right). \tag{3.2.4}$$

Thus the entire procedure comes down to picking the right form of the damping coefficients g_n, i.e. the kernel function.

The general condition which all kernel functions must satisfy is that in the limit $N \to \infty$ the absolute difference $||f - f_{KPM}|| \to 0$. A good kernel should ensure this converges as quickly as possible, thus requiring a smaller number of moments N to be computed. For specific applications such as the LDOS or Green's function, additional conditions should be imposed: the kernel should be positive $K_N(x, y) > 0$ and normalized $\int_{-1}^{1} K_N(x, y)dx = \phi_0(y)$. Based on these criteria, the kernel functions may be derived either analytically or empirically. The derivation procedure and a comparison of different kernels can found in Ref. [68]. For most practical applications, the most important ones are the Jackson and Lorentz kernel.

For the Jackson kernel $K_N^J(x,y)$ the damping factors are given as,

$$g_n^J = \frac{(N-n+1)\cos(\pi n/(N+1)) + \sin(\pi n/(N+1))\cot(\pi/(N+1))}{N+1}. \tag{3.2.5}$$

Convolving a kernel with the Chebyshev expansion of the delta function reveals the kernel-imposed broadening. For the Jackson kernel this is a Gaussian function of width $\sigma = \pi/N$. Therefore, the resolution of the reconstructed function will improve directly with the number of moments N.

An alternative is the Lorentz kernel,

$$g_n^L = \frac{\sinh[\lambda(1-n/N)]}{\sinh(\lambda)}, \tag{3.2.6}$$

where λ is a free parameter which can be used to control the resolution of the approximation. The Lorentzian broadening is given by $\epsilon = \lambda/N$. This kernel is most appropriate for the expansion of the Green's function because it most closely mimics the divergences near the true eigenvalues of the Hamiltonian. The λ value is found empirically to be between 3 and 5, and it may be used to fine-tune the smoothness of the convergence. The calculations in this thesis use the Lorentz kernel precisely because of this additional flexibility.

In the case of direct diagonalization, the LDOS would be calculated based on the eigenvalues and eigenvectors according to Eq. (3.1.25). However, an additional broadening is usually applied to account for the discrete nature of the eigenstates. For the Chebyshev expansion of the LDOS, the Gaussian or Lorentzian kernels play a similar role with the notable difference that they are an essential part of the computation procedure: increasing the broadening width will result in lower computation time because of the direct relation with N. This enables additional fine-tuning of the performance based on desired precision. The broadening widths σ or ϵ correspond well to their exactly diagonalized counterparts, but if given in energy units, they must undergo a rescaling as $\tilde{\sigma} = \sigma/a$ and $\tilde{\epsilon} = \epsilon/a$. For the numerical implementation of the Lorentz kernel, it is convenient to set the desired broadening ϵ and then compute the required number of moments $N = a\lambda/\epsilon$.

3.3 Chebyshev particle view

At first glance, it seems like the Chebyshev expansion (3.1.19) introduces a mathematical formalism which is able to solve the given problem, but at the cost of obscuring the physical interpretation. However, further examination reveals an interesting particle-like view with a connection to the real-space crystal lattice.

The key part of the LDOS expansion (3.1.26) is the Chebyshev iteration scheme Eqs. (3.1.28)-(3.1.30) used to compute the expansion moments. The moments have the form of expectation values $\langle\beta|T_n(\tilde{H})|\alpha\rangle$, not of the Hamiltonian $\tilde{H}$, but of the Chebyshev polynomials of $\tilde{H}$. Thus the vectors $|\alpha\rangle$ and $|\beta\rangle$ represent Chebyshev states instead of real Hamiltonian states. Still, the Chebyshev states do have a physical interpretation where the iterative procedure of computing states $|\alpha_n\rangle = T_n(\tilde{H})\,|\alpha\rangle$ may be viewed as the propagation of a particle defined by the starting state $|\alpha\rangle$.

To demonstrate this, we take a simple square quantum dot of graphene and compute the LDOS near its center. In the course of this computation the iterative procedure (3.1.28)-(3.1.30) generates Chebyshev states $|\alpha_n\rangle$, some of which are shown in Fig. 3.3.1. The starting state $|\alpha_0\rangle$ is a unit vector with a single non-zero value which is set to correspond to the lattice site where the LDOS is to be computed. The real-space view of $|\alpha_0\rangle$ is presented in the first panel of Fig. 3.3.1, where the single colored circle is the single non-zero value of the starting vector. The white circles represent lattice sites of the graphene quantum dot where the magnitude is zero. The next step produces the state $|\alpha_1\rangle = \tilde{H}\,|\alpha_0\rangle$. It is clear from matrix-vector multiplication that $|\alpha_1\rangle$ will have more non-zero values than $|\alpha_0\rangle$, which in a real-space view represents the immediate neighbors of the starting site, as seen in the next figure panel. All further iterations are computed from $|\alpha_{n+1}\rangle = 2\tilde{H}\,|\alpha_n\rangle - |\alpha_{n-1}\rangle$, where the matrix-vector multiplication again produces new non-zeros only on the neighbors of the existing non-zero sites. Thus, this iterative procedure is equivalent to the propagation of a Chebyshev particle in real space.

We refer only to zero and non-zero values of $|\alpha_n\rangle$ since this is information which can be obtained even without full knowledge of the Hamiltonian. Calculating the values of the vectors $|\alpha_n\rangle$ requires actually computing the matrix-vector multiplication with full knowledge of the onsite and hopping energy values. However, as demonstrated here, the very simple on-off state at each site can be determined with just structural information. This is because the state propagates strictly to neighboring sites at each iteration step. This is visible in Fig. 3.3.1 for starting states $|\alpha_0\rangle$ to $|\alpha_2\rangle$, as well for later iterations $|\alpha_7\rangle$ to $|\alpha_9\rangle$. Such a predictable pattern which can be determined purely from structural information can be quite useful.

The state $|\alpha_7\rangle$ in Fig. 3.3.1 is spread to most of the system, but it doesn't reach any of the edges. The next one does: state $|\alpha_8\rangle$ touches the edge of

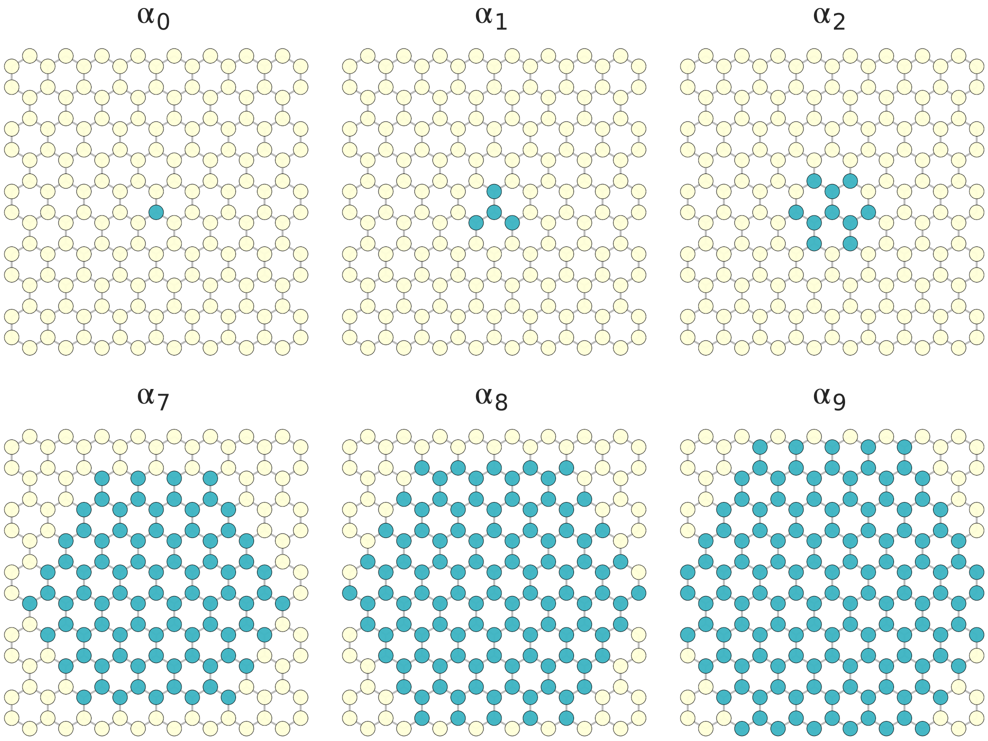


Fig. 3.3.1: Real space representation of the Chebyshev states $|\alpha_n\rangle$ at various steps of the iteration. The example system is a square quantum dot of graphene (side width 2.2 nm). The blue circles represent lattice sites where state $|\alpha_n\rangle$ has non-zero magnitude.

the system on the left and right side. Since we know that changes propagate by only one neighbor per iteration, a reflection from an edge at iteration $n_{edge} = 8$ will take the same number of iterations to travel back, reaching the original site at $N_c = 2n_{edge}$. This information allows us to determine the quality of the LDOS result, based on the total number of iteration steps N used in the calculation. If $N < N_c$, it means the resulting LDOS will not contain any features caused by the reflection from the edges of the system. On the other hand, the LDOS calculated using $N \geq N_c$ moments will surely present edge effects. Which outcome is desirable depends on the goals of the specific study: bulk or finite size properties. Either way, determining the critical number of moments N_c is very useful. Modeling systems with pronounced inhomogeneities (impurities, defects or edges) requires a larger number of moments to be computed in order for those effects to be visible.

The critical N_c can be determined purely from the lattice structure as the

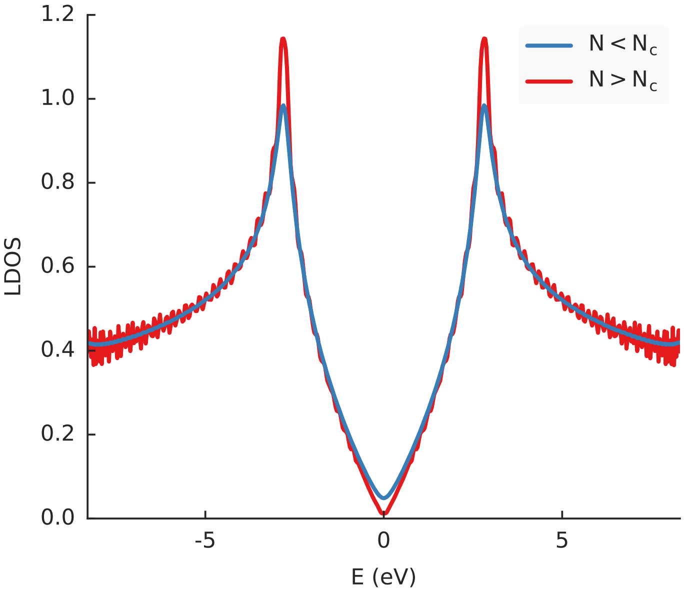


Fig. 3.3.2: KPM computed local density of states at the center of a square quantum dot of graphene (side width 20 nm and $N_c = 256$). The blue line shows the results of the expansion with a number of moments N below the critical value N_c. In that case the reflections from the edges are not captured. The red line shows lots of oscillations due to reflections which feature in the LDOS for $N > N_c$.

shortest hopping distance from the originating site to the closest edge. For a simple system as presented in Fig. 3.3.1 this can be determined trivially as the number of atomic positions at the edge. Note that, if the next-nearest neighbor model of graphene were used, the Chebyshev state propagation would hop two sites for each iteration step. Thus it would reach the edge twice as fast, at $N_c^{nn} = N_c/2$, making the effect of the edge reflection more apparent in the next-nearest neighbor model.

The influence of the number of expansion moments N is shown in Fig. 3.3.2. The example computes the LDOS at the center of a square quantum dot of graphene. For $N < N_c$ the LDOS is smooth since it only captures bulk-like features without any reflections from the edges. When the number of moments is increased to $N > N_c$, the effect of the edges is visible as oscillations in the LDOS. Note that the $N > N_c$ curve does also enhance bulk features: the V-shape of the LDOS at the Dirac point and the Van Hove singularities both appear sharper. As mentioned earlier, Chebyshev expansion is in some ways analogous to the Fourier transform. In this way, the higher-order Chebyshev moments can be seen to contribute as high frequency components which improve sharpness or bring out small fluctuations.

3.4 Green's function

Green's function describes the impulse response at any point in a system due to an excitation at any other. From this description, physical properties may be extracted by applying additional calculations. For example, eigenvalues will appear as poles of the Green's function while the wave-function amplitudes will appear as weights of the poles.

Given an excitation S, the response R is related to it by some differential operator D_{op} as

$$D_{op}R = S. \tag{3.4.1}$$

The corresponding Green's function expresses the response in the form,

$$R = D_{op}^{-1}S = GS, \tag{3.4.2}$$

where $G \equiv D_{op}^{-1}$ is Green's function. For quantum-mechanical problems, the differential operator can be expressed as $D_{op} = E - H$, where E is the energy and H the Hamiltonian, while $R = \Psi$ is the wavefunction,

$$[E - H]\Psi = S. \tag{3.4.3}$$

For a closed system $S = 0$, and the last equation turns into the familiar eigenvalue problem. But for an open system, such as a scattering region connected to leads, S represents the excitation from one of the leads. Owing to $G \equiv D_{op}^{-1}$, Green's function for our problem is found in the form,

$$G = [E - H + i\eta]^{-1}, \tag{3.4.4}$$

where the infinitesimally small term η is added in order to avoid singularities at energies close to the solutions of H. Strictly speaking Eq. (3.4.4) represents the retarded Green's function, but here we shall refer to it simply as "Green's function".

For tight-binding systems, the Hamiltonian is a large sparse matrix, therefore Eq. (3.4.4) represents a sparse matrix inversion. Computationally, this is a very time consuming problem which we would like to avoid by using the kernel polynomial method. Individual elements of the Green's function matrix G can be expressed via the correlation function,

$$\begin{aligned} G_{ij}(E) &= \langle i| \frac{1}{E - H + i\eta} |j\rangle\,, \\ &= \sum_{k=0}^{D-1} \frac{\langle i|k\rangle \langle k|j\rangle}{E - E_k + i\eta}, \end{aligned} \tag{3.4.5}$$

where i and j are site indices, E_k are the eigenstates of the Hamiltonian H in the basis extended by eigenstates $|k\rangle$. Using the Plemelj-Dirac formula,

$$\lim_{\eta\to 0} \frac{1}{x+i\eta} = \mathcal{P}\frac{1}{x} - i\pi\delta(x), \tag{3.4.6}$$

the imaginary part of the Green's function is,

$$\mathrm{Im}G_{ij}(E) = -\pi \sum_{k=0}^{D-1} \langle i|k\rangle \langle k|j\rangle \, \delta(E-E_k). \tag{3.4.7}$$

In order to expand this function in terms of Chebyshev polynomials, the problem must first be rescaled using substitutions $H \to \tilde{H}$, $E \to \tilde{E}$, as introduced earlier. The expansion of the imaginary part of the Green's function is performed according to Eq. (3.1.19) and given by,

$$\mathrm{Im}G_{ij}(\tilde{E}) = -\frac{1}{\sqrt{1-\tilde{E}^2}} \left(\mu_0 + 2\sum_{n=1}^{\infty} \mu_n T_n(\tilde{E}) \right), \tag{3.4.8}$$

where the moments μ_n are computed according to Eq. (3.1.20),

$$\begin{aligned} \mu_n &= \frac{1}{\pi}\int_{-1}^{1} \mathrm{Im}G_{ij}(\tilde{E}) T_n(\tilde{E}) d\tilde{E} \\ &= \sum_{k=0}^{D-1} \langle i|k\rangle \langle k|j\rangle \, T_n(\tilde{E}_k) \\ &= \sum_{k=0}^{D-1} \langle i|T_n(\tilde{H})|k\rangle \langle k|j\rangle \\ &= \langle i|T_n(\tilde{H})|j\rangle \, . \end{aligned} \tag{3.4.9}$$

The real part is fully determined by the imaginary part. From Eq. (3.4.6) and the Hilbert transformation (3.1.15) it follows that

$$\begin{aligned} \mathrm{Re}G_{ij}(\tilde{E}) &= -\frac{1}{\pi}\mathcal{P}\int_{-1}^{1} \frac{\mathrm{Im}G_{ij}(E)}{\tilde{E}-\tilde{E}'} d\tilde{E}' \\ &= -2\sum_{n=1}^{\infty} \mu_n U_{n-1}(\tilde{E}) \end{aligned} \tag{3.4.10}$$

Combining the real and imaginary parts, the full Green's function becomes,

$$G_{ij}(\tilde{E}) = \frac{-i}{\sqrt{1-\tilde{E}^2}} \left(\mu_0 + 2\sum_{n=1}^{\infty} \mu_n \mathrm{e}^{-in\arccos(\tilde{E})} \right). \tag{3.4.11}$$

Note that this represents a single element of the Green's function matrix and that the calculation does not depend on any neighboring matrix elements.

The entire computational effort is tied up in the calculation of the expansion moments μ_n (3.4.9). They appear as expectation values which are computed iteratively according to the scheme outlined in Eqs. (3.1.28)-(3.1.31). The procedure is replicated here in terms of $|i\rangle$ and $|j\rangle$ for convenience:

$$|j_0\rangle = |j\rangle\,, \tag{3.4.12}$$

$$|j_1\rangle = \tilde{H}\,|j_0\rangle\,, \tag{3.4.13}$$

$$|j_{n+1}\rangle = 2\tilde{H}\,|j_n\rangle - |j_{n-1}\rangle\,, \tag{3.4.14}$$

$$\mu_n = \langle i|j_n\rangle\,. \tag{3.4.15}$$

Note that moments of diagonal elements G_{ii} may be computed according to the faster procedure given by Eqs. (3.1.32) and (3.1.33),

$$\mu_{2n} = 2\,\langle j_n|j_n\rangle - \mu_0, \tag{3.4.16}$$

$$\mu_{2n+1} = 2\,\langle j_{n+1}|j_n\rangle - \mu_1. \tag{3.4.17}$$

Since the moments μ_n do not depend on energy, they need to be computed only once and can be reused to determine the Green's function for any energy. The final equation (3.4.11) is computationally cheap compared to the effort needed for the moments. It can even be implemented in terms of the fast Fourier transform (FFT) thereby further improving efficiency [68]. Practically, the KPM-based Green's function is computationally "free" as a function of energy.

In the real-space tight-binding representation, a matrix element is tied to a spatial position. A single computation of G_{ij} will return the entire energy spectrum for a single position, so the effort scales up with the desired spatial area. This is very convenient when studying localized properties: only the very small set of elements needs to be computed. If a larger area is needed, the computation can be parallelized naturally since a single element G_{ij} does not depend on any other. In addition, performance can be improved by computing an entire column of the Green's function matrix at once. Following the iterative scheme Eqs. (3.4.12)-(3.4.14) it is clear that the vector $|j_n\rangle$ only depends on the column index j, while the row index i comes in only in the final stage Eq. (3.4.15). This enables the computation of an entire column of the Green's function matrix at minimal extra cost compared to computing a single element. Parallelization can then be applied over columns instead of individual elements.

After the Green's function is computed, physical properties may be extracted by applying additional calculations, e.g. the eigenvalues can be found as its zeros. The local density of states corresponds to the imaginary part of the local spectral function,

$$\rho_i(E) = -\frac{1}{\pi}\mathrm{Im}G_{ii}(E). \tag{3.4.18}$$

The current density between lattice sites i and j is,

$$J_{ij} = -\frac{1}{\pi}\int \mathrm{Im}[it_{ij}G_{ij}(E) - it^*_{ij}G^*_{ij}(E)]f(E)dE. \tag{3.4.19}$$

CHAPTER 4

Simulating tight-binding systems with computer code

Analytical methods can frequently be used to design a sufficiently simplified model of a physical system while still retaining and predicting real-world behavior. They can easily be applied to infinitely large systems where they can present the whole picture in the form of an elegant analytical expression. However, the conditions imposed to derive such models can fail for particular fields of interest. We encounter several such cases in this thesis. When parameters become too large they invalidate certain approximations (strain up to the full limits of graphene). Continuum models cannot consider features smaller than the wavelength (vacancies and other structural defects). Phenomena which compete on similar energy scales cannot be treated perturbatively (supercritical impurities and magnetic fields). Such cases can only be considered numerically, hence our interest in using and developing new numerical tools.

If the description of only a handful of atoms is required, first principal methods offer very good accuracy, but they also come with an inherent limitation of the size of the system which can be considered. A middle ground is found in the form of the tight-binding method which approximates some features but still gives a very detailed description of the system. It can be used to model systems on the order of millions of atoms.

The tight-binding method also represents a middle ground in terms of how difficult it is to implement using computer code. First principal calculations present a high enough degree of difficulty that few would consider writing their own single-purpose code which would be discarded after one use. Instead, attention is given to building reusable code libraries which greatly simplify usage. A quick online search reveals a large number of ab initio software packages [69], which present good reusable resources and allow more work to be done without having to first "reinvent the wheel". On the other hand, tight-binding is just simple enough that individuals may write semi-complex code which solves a single problem and will later be discarded. A better solution is to write code which can be applied to a whole class of problems. Although it does require more initial effort, development time is actually reduced in the long run thanks to code reusability. This also improves the overall code quality and reduces errors because more time is spend on less code.

While nowhere near as abundant as ab initio software, the domain of tight-binding does feature packages of note: PythTB [70] and Kwant [71]. PythTB is able to construct both finite size and periodic tight-binding systems of arbitrary dimensionality and it specifically includes rich tools for computing Berry phases. It was primarily intended as a teaching tool and is not optimized for large tight-binding problems. Kwant specializes in transport calculations where semi-infinite leads are attached to a finite scattering region. It targets realistically large systems and has a very efficient solver for computing the scattering matrix. It can also be used to construct arbitrary finite-sized systems.

In the course of this thesis, a new tight-binding code package was developed, called *Pybinding*. Although it was initially conceived to aid the numerical work specifically for this thesis, it is sufficiently general purpose that it can be applied to a wide range of tight-binding systems. It is available as an open source project[*] for anyone to use. Appendix A contains a usage tutorial with some concrete code examples. This chapter presents the theoretical framework and ideas behind the code. We examine how the abstractions used in physics need to be modified in order to develop an effective code framework for the tight-binding model. We also present specific tweaks which can be made to improve the performance of the kernel polynomial method both for CPUs and GPUs.

This chapter is presented in general mathematical terms with some language-independent pseudo-code, thus making a concrete implementation possible in any programming language. The actual Pybinding code is written in a combination of Python and C++11. The source code is available online[*].

[*] https://github.com/dean0x7d/pybinding

4.1 Constructing a tight-binding system

An introduction to the tight-binding model was presented in Chapter 2. There, we were mainly concerned with the bulk properties of a material where it is sufficient to describe just the unit cell and apply periodic boundary conditions. In order to consider finite-sized systems and related effects (edges, defects, etc.) we must further develop the real-space representation of the tight-binding model. The size of the Hamiltonian matrix in this representation is equal to the number of atoms (and orbitals/spins) which are contained in the system. Here, we shall discuss ways of constructing such large-scale systems.

For an N-dimensional Bravais lattice, unit cell positions are described by the infinite set,

$$\vec{r} = \sum_{i=1}^{N} n_i \vec{a}_i, \tag{4.1.1}$$

where $\vec{a}_i$ are lattice vectors in real space and n_i are integers. While $\vec{r}$ describes the real space coordinates of a unit cell, we can also define the compound index $n = \{n_1, n_2, n_3, ...\}$ which describes the position of a unit cell in lattice coordinates. A unit cell may contain multiple atoms with distinct positions,

$$\vec{r}_\mu = \vec{r} + \vec{\delta}_\mu, \tag{4.1.2}$$

where $\vec{\delta}_\mu$ is the position of sublattice μ within the unit cell.

In general, the tight-binding Hamiltonian can be written down as,

$$H = \sum_{i,j} H_{ij} \left|i\right\rangle \left\langle j\right|, \tag{4.1.3}$$

where i and j are Hamiltonian indices which refer to the degrees of freedom of the system. For real-space tight-binding models, the integer index i refers to state $|i\rangle = |\vec{r}_\mu \alpha\rangle$, where $\vec{r}_\mu$ is the position and α may label any additional degrees of freedom (e.g. orbitals and spins). Note that the sublattice degree of freedom is coupled directly with the position $\vec{r}_\mu$ because we chose this convention in Eq. (4.1.2).

The Hamiltonian indices i and j are strictly integers which index the scalar matrix elements H_{ij}. Matrix elements where $i \neq j$ contain values of the hopping parameters which connect indices i and j. Diagonal elements H_{ii} are the onsite energy terms. Alternatively, the hoppings can also be defined as matrices,

$$H_{\vec{r}_\mu \vec{r}'_\mu} = \sum_{\alpha,\alpha'} H_{\vec{r}_\mu \alpha \vec{r}'_\mu \alpha'} \left| \vec{r}_\mu \alpha \right\rangle \left\langle \vec{r}'_\mu \alpha' \right| , \tag{4.1.4}$$

where $H_{\vec{r}_\mu \vec{r}'_\mu}$ is a submatrix of the full Hamiltonian. It has the dimensions $N_\alpha \times N_{\alpha'}$ which correspond to the number of degrees of freedom of α and α', respectively. In the simplest case of a single-orbital, spinless model, $H_{\vec{r}_\mu \vec{r}'_\mu}$ would be a scalar value just like H_{ij} and index i would refer to position $\vec{r}_\mu$. When the additional degrees of freedom α are added, $H_{\vec{r}_\mu \vec{r}'_\mu}$ becomes a matrix but it retains the direct correspondence to positions in real space. This is in contrast to H_{ij} where we may find multiple different indices corresponding to the same position (e.g. multiple orbital on the same atom). For a concrete implementation in code, the $H_{\vec{r}_\mu \vec{r}'_\mu}$ approach has the distinct advantage both in terms of algorithmic complexity (it's easier to reason about direct connections in real space) as well as memory requirements (there is a smaller number of unique hoppings to keep track of).

For a pristine material, the hoppings $H_{\vec{r}_\mu \vec{r}'_\mu}$ will be constants, but in general they may be position dependent, e.g. in the presence of an external field. Thus, it is useful to separate the system construction process into two distinct parts. The pristine material is constructed first with predefined onsite and hopping parameters. The energy values can later be modified by introducing appropriate transformations. Similarly, the definition of the geometry of the system can be separated into two parts, where a highly ordered Bravais lattice is considered first and any disorder is added in a later construction step.

The entire construction procedure consists of the following steps:

1. Define the unit cell of the Bravais lattice.
2. Create a finite-sized pristine system by translating the unit cell.
3. Introduce structural defects and disorder.
4. Apply fields or any effects which modify the onsite and hopping terms.

The initial step merely collects information about the pristine material. This includes the lattice vectors $\vec{a}_i$, sublattice offsets $\vec{\delta}_\mu$, and the onsite and hopping terms $H_{\vec{r}_\mu \vec{r}'_\mu}$ which are constants at this point, thus reflecting the pristine and infinitely periodic lattice.

Building a finite-sized pristine system

The geometry of a tight-binding system can be defined by supplying the coordinates of specific cells, i.e. supplying a set of indices n_i as per Eq. (4.1.1).

However, doing so by hand can be very tedious for humans. It is preferable to define the geometry using more natural ways such as simple polygons or freeform shapes which are defined by a mathematical relation. The computer can then translate from that high-level concept into raw lattice indices.

The problem of filling in a shape with discrete points can be solved from two different directions: bottom-up or top-down. The bottom-up approach involves the use of the flood-fill algorithm [72]. It starts from a single point which is known to be inside the desired shape and then recursively visits all neighbors. Points which are found to be inside the shape are kept and the recursive procedure continues until no more points can be added. The downside of this procedure is that it's inherently serial: a new point cannot be added before its previous neighbor is checked. Thus, for performance reasons, we choose to adopt the alternative top-down method using a bounding box and scanning fill.

To demonstrate the bounding box approach, we shall first consider a simple 2D polygon shape defined by a set of vertices $\vec{v}_i$ (see Fig 4.1.1(a)). We need to determine a minimum box which can hold the given shape. This is a box, not in Cartesian coordinates, but in lattice coordinates, i.e. the sides of the box are aligned with the unit vectors of the lattice. For user convenience, polygon vertices $\vec{v}_i$ are given in Cartesian coordinates, so we must first translate then into lattice coordinates.

Let us assume that a polygon vertex $\vec{v}$ can be expressed in terms of the lattice vectors similar to Eq. (4.1.1),

$$\vec{v} = \sum_{i=1}^{N} \eta_i \vec{a}_i, \tag{4.1.5}$$

with the difference that the terms η_i are real numbers, unlike the integers n_i which index unit cells. This change is perfectly natural as the freely positioned vertex $\vec{v}$ does not need to conform perfectly to lattice unit cells. The terms η_i can also be thought of as the components of vector $\vec{\eta}$. We already know the value of $\vec{v}$ as well as all of the lattice vectors $\vec{a}_i$, but we must determine $\vec{\eta}$. Notice that Eq. (4.1.5) defines a system of linear equations. In 3 dimensions the system reads,

$$\begin{pmatrix} a_{1x} & a_{2x} & a_{3x} \\ a_{1y} & a_{2y} & a_{3y} \\ a_{1z} & a_{2z} & a_{3z} \end{pmatrix} \cdot \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}. \tag{4.1.6}$$

For a small number of dimensions, this system is easy to solve by hand. For the general N-dimensional case, it can be easily solved numerically using any

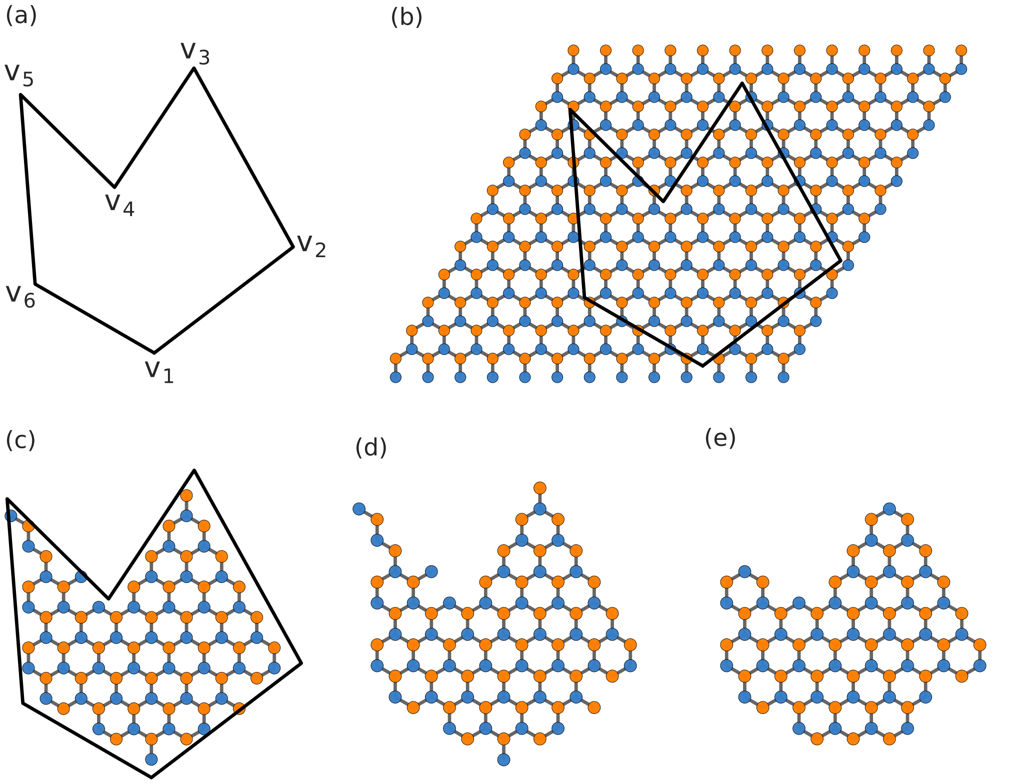


Fig. 4.1.1: Example of the construction procedure for a specific geometry of a graphene system. (a) The desired shape, expressed as a collection of polygon vertices. (b) The lattice-aligned bounding box encompasses the desired shape. (c) All positions are checked and only the ones inside the shape are kept. (d) Final system including dangling bonds. (e) After clearing the dangling bonds.

linear algebra package. After computing the η_i values for each vertex of the polygon, the values are rounded to the nearest integers which aligns them to the unit cells. The extremal values represent the farthest points of the shape which must be contained inside the bounding box. An illustration of this is given in Fig. 4.1.1(b).

The advantage of the lattice-aligned box is that it can be represented as an N-dimensional array in code. All elements of this array can then be processed in parallel to check which points are actually located inside the shape. For shapes given as a list of vertices, this can be resolved quickly using the well known point-in-polygon algorithm [73]. In the most general N-dimensional case of a freeform shape, the mathematical function which describes the shape also defines which points will be kept. A small inconvenience for freeform shapes is that the bounding box cannot be determined automatically, so an extra N-dimensional width parameter needs to be supplied to limit the extent of the shape. Fortunately, this can be a simple Cartesian box and the computer will translate into lattice coordinates as before.

Figures 4.1.1(c) and (d) illustrate the geometry of a graphene system after filling in the desired polygon. Note that there are a number of atoms with only a single neighbor. In the most general case, we can define a dangling atom as one which has less than some minimum number of neighbors. For example, in the nearest neighbor model of graphene, the minimum neighbor count is 2, but for the next-nearest neighbor model all atoms which have less that 4 neighbors are considered dangling. The number itself will depend on the material and specific model. Removing the dangling atoms is a simple case of counting the neighbors of all atoms and eliminating the ones that don't satisfy the minimum.

Adding disorder

After the regular part of the system is constructed, we can add various defects or strain the system. Defects can be expressed as position-dependent functions which can add or remove an atom from the existing system. Similarly bond reconstruction is achieved by designating position pairs where to add or remove a hopping term. Strain is included by defining a position displacement function, as expected.

In general, a tight-binding system does not need to have any kind of regular structure. A collection of randomly positioned sites connected by randomly distributed hoppings would still constitute a tight-binding system. By making the construction of a regular Bravais lattice the first step of our construction procedure, we are biasing our implementation towards more regular structures where disorder is added only in the second stage. This is not the most general purpose approach, however it is a pragmatic choice as most systems in solid state physics are like this. The regular structure also allows for better performance, as we shall see later.

Adding external fields

Up to this point, all onsite and hopping energy terms $H_{\vec{r}_\mu \vec{r}'_\mu}$ have been considered constants, reflecting the pristine material. At this stage we can introduce transformations which apply external fields or any other kinds of

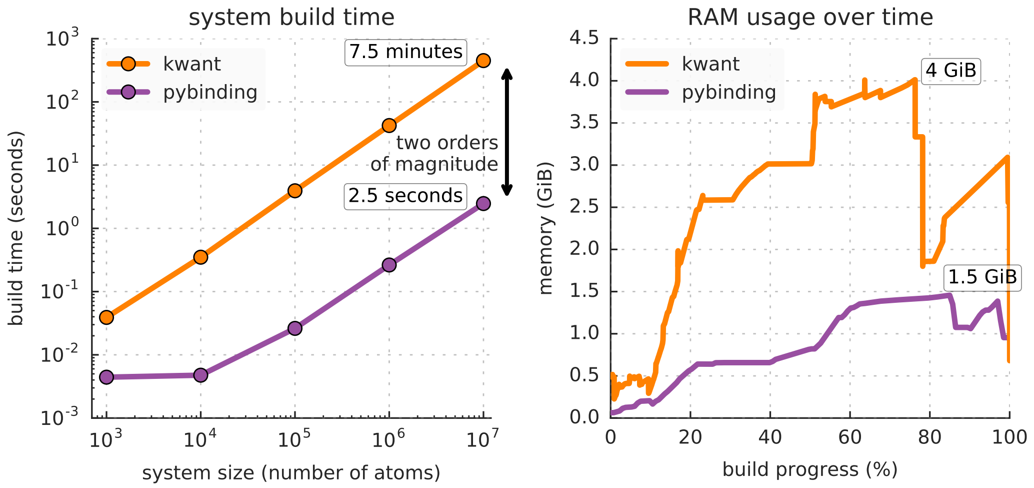


Fig. 4.1.2: Tight-binding Hamiltonian construction performance and memory usage. The results were measured for Pybinding v0.8.0 and Kwant v1.2.2 using: Intel Core i7-4960HQ CPU, 16 GiB RAM, Python 3.5, macOS 10.11. The RAM usage was measured using memory_profiler v0.41.

effects which modify the onsite and hopping terms. For the onsite energy, a function of the form $v(v_0, \vec{r}_\mu)$ takes the existing onsite value v_0 and position $\vec{r}_\mu$ as parameters and returns the new onsite energy values, e.g. representing an electric field. For the hoppings, function $t(t_0, \vec{r}_\mu, \vec{r}\,'_\mu)$ would define the value of the hopping energy between positions $\vec{r}_\mu$ and $\vec{r}\,'_\mu$, where t_0 is the pristine value. For example, a magnetic field can be added using the Peierls substitution in this form.

The Hamiltonian matrix is the final product of the construction procedure. It fully describes the model system, including the structural relationships between atoms and the concrete onsite and hopping energy parameters. Because of the large scale of tight-binding systems, the Hamiltonian is stored as a sparse matrix, specifically, in the CSR format [74]. This is a commonly used format which is compatible with many solver routines. The next step is to apply computations to the matrix to obtain the values of the desired quantum properties. This can be done either using exact diagonalization or approximative methods, like the kernel polynomial method.

System construction performance

The procedure outlined in this chapter is implemented in the Pybinding code package. The benefits of this specific approach are high performance and low memory usage of the tight-binding system builder.

In order to quantify the performance, we compare the built time and memory usage of Pybinding with the Kwant package. Both packages can be used to construct the same tight-binding systems. For this comparison, we construct a circular graphene flake with a pn-junction and a constant magnetic field (for more details, see Appendix A). The system build time is measured from the start of the definition to the point where the Hamiltonian matrix is fully constructed (a sparse matrix is used in both cases).

Pybinding builds the Hamiltonian much faster than Kwant: by two orders of magnitude. The main reason for this is in the way the system shape and fields are implemented. Both Kwant and Pybinding take user-defined functions as parameters for model construction. Kwant calls these functions individually for each atom and hopping which is quite slow. Pybinding stores all atoms and hoppings in contiguous arrays and then calls the user-defined functions just once for the entire dataset. This takes advantage of vectorization and drastically improves performance. Similarly, the lower memory usage is achieved by using arrays and CSR matrices rather than linked lists and trees.

Please note that at the time of writing Pybinding v0.8 does lack certain system construction features compared to Kwant. Specifically, it is currently not possible to build heterostructures in Pybinding, but this will be resolved in future versions. New features will be added while maintaining good performance.

4.2 Optimizing the KPM algorithm

The kernel polynomial method (KPM) was described in detail in Chapter 3. There, we noted that almost all of the computational effort of this method is tied up in the calculation of the Chebyshev expansion moments μ_n. The entire KPM procedure consists of many additional steps which both precede the moments calculation (e.g. finding the energy bounds and rescaling the Hamiltonian) and follow it (e.g. applying the damping kernel and reconstructing the function of interest). Although some of those other steps are conceptually more difficult, computationally by far the most expensive part of the KPM procedure is the calculation of the moments. Here, we shall take a look at the implementation of this algorithm and present specific tweaks which can be made to improve the performance both for CPUs and GPUs.

The expansion moments may have different forms depending on the function which is being approximated, however, it is very common for them to appear as expectation values $\mu_n = \langle\beta|T_n(\tilde{H})|\alpha\rangle$, where $|\alpha\rangle$ and $|\beta\rangle$ are certain states of the system, $\tilde{H}$ is the rescaled Hamiltonian and T_n are Chebyshev polynomials of the first kind (see Chapter 3 for details). For the first optimization step presented here, we shall assume that $|\alpha\rangle$ and $|\beta\rangle$ are unit vectors, however, the later optimizations are equally beneficial for the random starting vector KPM procedure.

The full iterative scheme for computing $\mu_n = \langle\beta|T_n(\tilde{H})|\alpha\rangle$ is outlined in detain in Eqs. (3.1.28)-(3.1.31). A summary of the procedure is replicated here for convenience:

$$|\alpha_0\rangle = |\alpha\rangle \,, \tag{4.2.1}$$

$$|\alpha_1\rangle = \tilde{H}\,|\alpha_0\rangle \,, \tag{4.2.2}$$

$$|\alpha_{n+1}\rangle = 2\tilde{H}\,|\alpha_n\rangle - |\alpha_{n-1}\rangle \,, \tag{4.2.3}$$

$$\mu_n = \langle\beta|\alpha_n\rangle \,. \tag{4.2.4}$$

The corresponding pseudo-code function is presented in Algorithm 1. The function parameters include the sparse matrix $\tilde{H}$, vectors $\vec{\alpha}$ and $\vec{\beta}$ and the total number of moments N which are to be calculated. The result of the function are the expansion moments μ_n.

Algorithm 1 Calculation of expansion moments

```
function CALCULATE_MOMENTS($\tilde{H}, \vec{\alpha}, \vec{\beta}, N$)
    $\vec{\alpha}_0 \leftarrow \vec{\alpha}$
    $\mu_0 \leftarrow \vec{\beta} \cdot \vec{\alpha}_0$
    $\vec{\alpha}_1 \leftarrow \tilde{H}\vec{\alpha}_0$
    $\mu_1 \leftarrow \vec{\beta} \cdot \vec{\alpha}_1$
    $\tilde{H}_2 \leftarrow 2\tilde{H}$
    for $n \leftarrow 2, 3, \dots N-1$ do
        $\vec{\alpha}_0 \leftarrow \tilde{H}_2\vec{\alpha}_1 - \vec{\alpha}_0$          ▷ most compute-intensive line
        $\mu_n \leftarrow \vec{\beta} \cdot \vec{\alpha}_0$
        SWAP($\vec{\alpha}_0, \vec{\alpha}_1$)
    end for
    return $\mu_0$, $\mu_1$, $\mu_2$, ... $\mu_N$
end function
```

At this stage, a few obvious optimizations are already taken into account. Only the two latest vectors (α_n and α_{n-1}) are kept in memory at any given

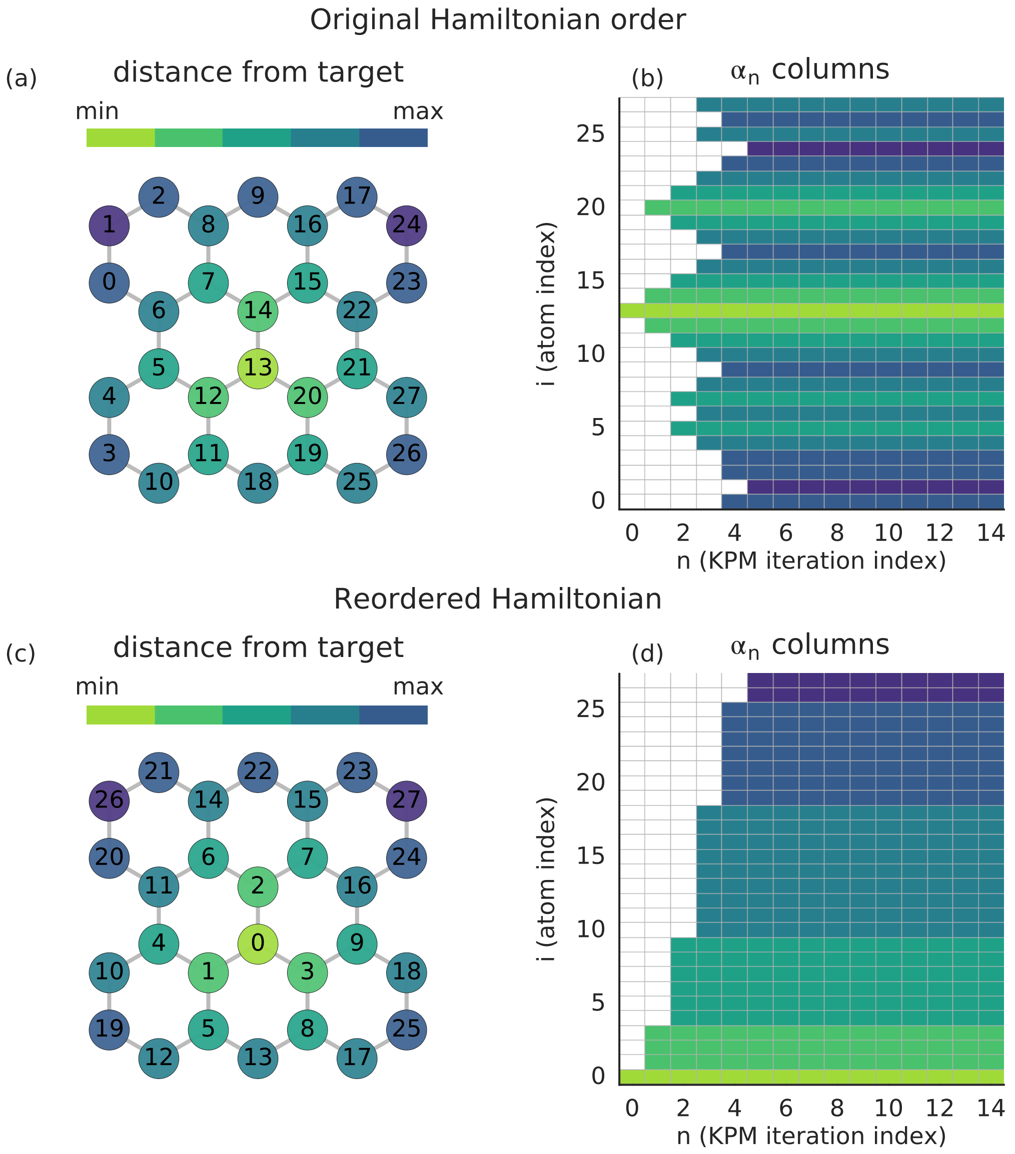


Fig. 4.2.1: Illustration of Hamiltonian reordering and its impact on KPM iteration. (Top) The original Hamiltonian. (Bottom) Reordered Hamiltonian. (Left) Illustration of a small graphene quantum dot. The numbers indicate the Hamiltonian index i of an atom. The bright green atom near the center is the target of the calculation, i.e. the starting unit vector $|\alpha_0\rangle$ is set to 1 at that index. The subsequent colors distinguish atoms by their distance from the target (i.e. nearest, next nearest, etc.). (Right) Each column represent a vector $|\alpha_n\rangle$ at each iteration step n. The white blocks indicate a zero values at a particular atom i. Colored blocks indicate non-zero values, but the color itself corresponds to distance from the target atom.

time. The constant product $2\tilde{H}$ is lifted outside the loop and executed only once. In case $\vec{\beta}$ is a unit vector, the dot product $\vec{\beta} \cdot \vec{\alpha}_0$ simplifies to array indexing. We assume these simple tweaks are applied and we consider this the base "unoptimized" version of the algorithm.

Algorithm 1 is actually extremely simple and there is not much that can be done to further simplify the algorithm itself. Instead, we shall focus on the structure of the data which is used in the computation. Line 8 contains the most expensive part of the procedure: the sparse matrix-vector multiplication (SpMV). The rescaled Hamiltonian $\tilde{H}$ must be a sparse matrix, but the specific sparse format does not matter for the following optimization procedure.

Hamiltonian matrix reordering and slicing

To better understand the data structure of the KPM iteration procedure, we use an example of a small graphene quantum dot, as shown in Fig. 4.2.1(a). Each atom is labeled with its Hamiltonian index. The order in which the atoms are numbered does not matter physically (all permutations are equivalent), so this just represents an implementation detail which can be tweaked as needed.

Figure 4.2.1(b) presents vectors $|\alpha_n\rangle$ as columns and shows the progression with iteration step n. For illustration, we arbitrarily take $N = 15$ as the total number of KPM iterations. Each block i of a vector $|\alpha_n\rangle$ corresponds to an atom. White blocks contain zeros and colored blocks non-zeros. The KPM iteration uses the unit vector $|\alpha_0\rangle$ as the starting point, where only atom $i = 13$ is set to 1 since this is the target of the calculation. At iteration step $n = 1$, vector $|\alpha_1\rangle$ contains 4 non-zero values corresponding to its nearest neighbors. Iteration $n = 2$ contains more non-zeros and so on. This progression is due to SpMV and the non-zeros spread strictly according to their neighbors (this was explained in more detail in Section 3.3). After iteration step $n = 5$ all vectors $|\alpha_n\rangle$ are densely populated.

The optimization opportunity here lies in the starting region $n < 5$ which involves only partially populated vectors (which contain mostly zero elements). Note that this region becomes more significant as the size of the system grows, however, we keep the system small for the example. When the Hamiltonian matrix is multiplied by a partially populated vector, computational resources are wasted with lots of multiplications by zero. The solution is, of course, to skip the useless operations, but this is not as simple as it might seem. The

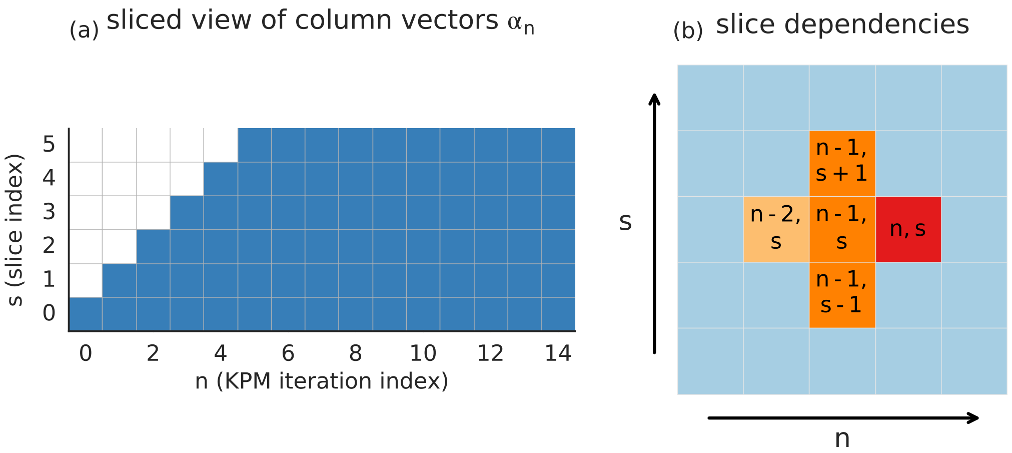


Fig. 4.2.2: Sliced view and dependencies of KPM iteration. (a) Vectors $|\alpha_n\rangle$ where each row represents a region with the same color from Fig. 4.2.1(d), i.e. each row in this figure is a slice of the system containing atoms with the same distance from the target. A colored slice contains non-zero values while the white slices contain only zeros and will not be involved in any calculation. (b) In order to calculate the values in the red slice (n, s), we must first know the values of the orange and yellow slices (and so on as we move backward with the n index).

progression of zero/non-zero elements, as shown in Figs. 4.2.1(a) and (b), is very simple to map out ahead of time using just the structural information and following the neighbors starting from the target. Therefore, a mask could be created for each step n of the KPM iteration which would select just the non-zero elements for multiplication. Unfortunately, such masks would use extra memory and a performance penalty would be incurred for the conditional operation of selecting elements for multiplication (which would need to be checked for each element). The overhead of this selection logic is actually

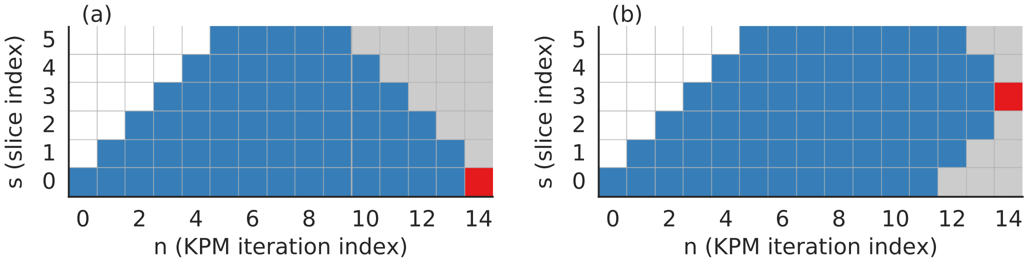


Fig. 4.2.3: Tail end optimization. The gray slices contain non-zero values but they need not be computed. The red block corresponds to the ultimate calculation target defined by the vector $|\beta\rangle$. For the case when (a) $|\beta\rangle = |\alpha\rangle$ and (b) $|\beta\rangle \neq |\alpha\rangle$.

greater than the multiplication by zero which we wish to eliminate, thus this approach is counterproductive.

The main issue is that each vector $|\alpha_n\rangle$ has a non-contiguous distribution of non-zero values. We can resolve this by reordering the Hamiltonian indices of the atoms as shown in Fig. 4.2.1(c). The target atom is repositioned to coincide with $i = 0$, its nearest neighbors take the subsequent numbers, and so on. The vectors of the KPM iteration are now nicely sorted, Fig. 4.2.1(d). The sorting itself does not have any impact on performance, but it does enable the following optimization. Because the non-zero values are all located in a contiguous region at the start, we don't need any masks but just a single number to indicate the size of the non-zero region. This size can be know ahead of time just based on the structural information and a single size value can be assigned to each KPM iteration. We can then do only a partial matrix-vector multiplication, involving only elements up to the given size (no need for any selection logic). Because only a single size number is involved, there is practically no optimization overhead and the final performance gains are directly proportionally to the number of zero-element multiplications which have been eliminated.

Now that the Hamiltonian indices have been sorted, a new way to view KPM iteration is revealed: as slices of the system. We define a slice as containing all the atoms which have the same distance from the target atom, i.e. each color in Fig. 4.2.1(d) represents a different slice. Note that this means that the number of atoms inside a slice is variable. We can now rework the representation of the KPM iteration into its equivalent slice view and present it in Fig. 4.2.2(a). The iterations are indexed with n as before, and s indexes the slices. White slices contain only zero values and will never be part of any calculation as per the previous optimization.

The benefit of the sliced view is that each new KPM iteration will involve exactly one more slice, which makes it easier to reason about. This also makes it easier to notice the dependence relation between slices. As per the KPM iteration procedure (4.2.3) each new vector is produced from the matrix-vector multiplication $|\alpha_n\rangle = 2\tilde{H}\,|\alpha_{n-1}\rangle - |\alpha_{n-2}\rangle$. To calculate only a contiguous slice s of vector $|\alpha_n\rangle$ we need only slices $s-1$ to $s+1$ from vector $|\alpha_{n-1}\rangle$ and slice s from vector $|\alpha_{n-2}\rangle$, as depicted in Fig. 4.2.2(b). We must take 3 slices from $|\alpha_{n-1}\rangle$ because of the matrix-vector multiplication. Only those 3 are required because they are nearest neighbors* and no other slices

*The nearest neighbor relationship of the slices is orthogonal to the nearest neighbor hoppings of the atoms in our example model. This sliced view can be applied completely generally to any model with any level of neighbors taken into account.

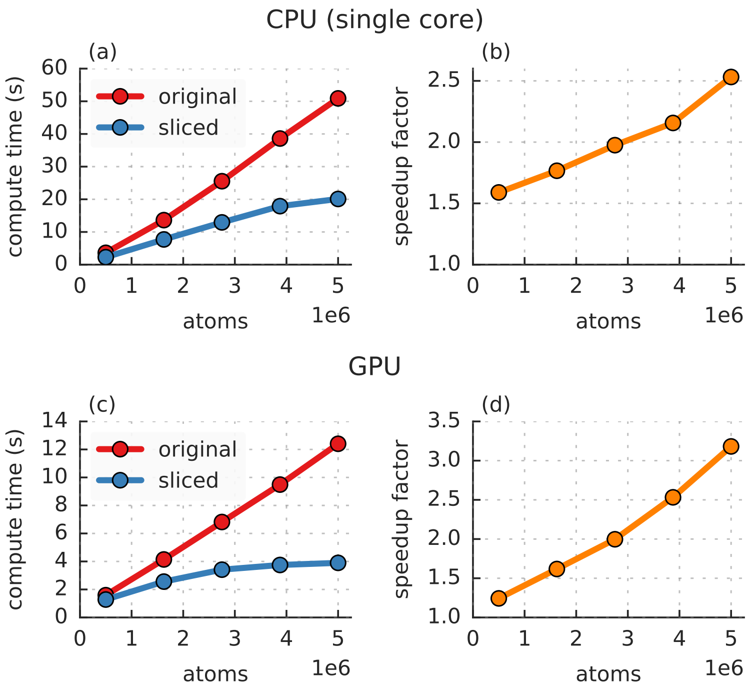


Fig. 4.2.4: Performance benefits of the slicing optimization. (Top) Running on a single CPU core and (bottom) on the GPU. (Left) Computation time of 3.4 thousand KPM moments for a circular graphene flake with a varying number of atoms. $|\beta\rangle = |\alpha\rangle$ and taken in the center. (Right) Speedup factor obtained with the optimization. The results were gathered from Pybinding v0.8.0. The CPU code was execute using Intel Core i7-4960HQ with 16 GiB RAM, and the GPU code using Nvidia GeForce GT 750M with 2 GiB RAM.

will have any influence on the product. Only the single slice s is required from $|\alpha_{n-2}\rangle$ because of the simple subtraction.

With the slice dependence relation worked out as presented in Fig. 4.2.2(b), we can eliminate additional unneeded calculations from the tail of the iteration. Specifically, because the final moment is determined by $\mu_n = \langle\beta|\alpha_n\rangle$, we may not actually need to compute the full vector $|\alpha_n\rangle$, but only the slice corresponding to non-zero values in $|\beta\rangle$. Examples are presented in Fig. 4.2.3. The gray slices of the tail end iterations represent non-zero values, but they need not be computed because their contribution will not affect the final result.

The optimized procedure for KPM iteration involves the same Algorithm 1, where the full SpMV from Line 8 is just replaced with a partial SpMV which takes very simple start and end indices to identify the contiguous region which

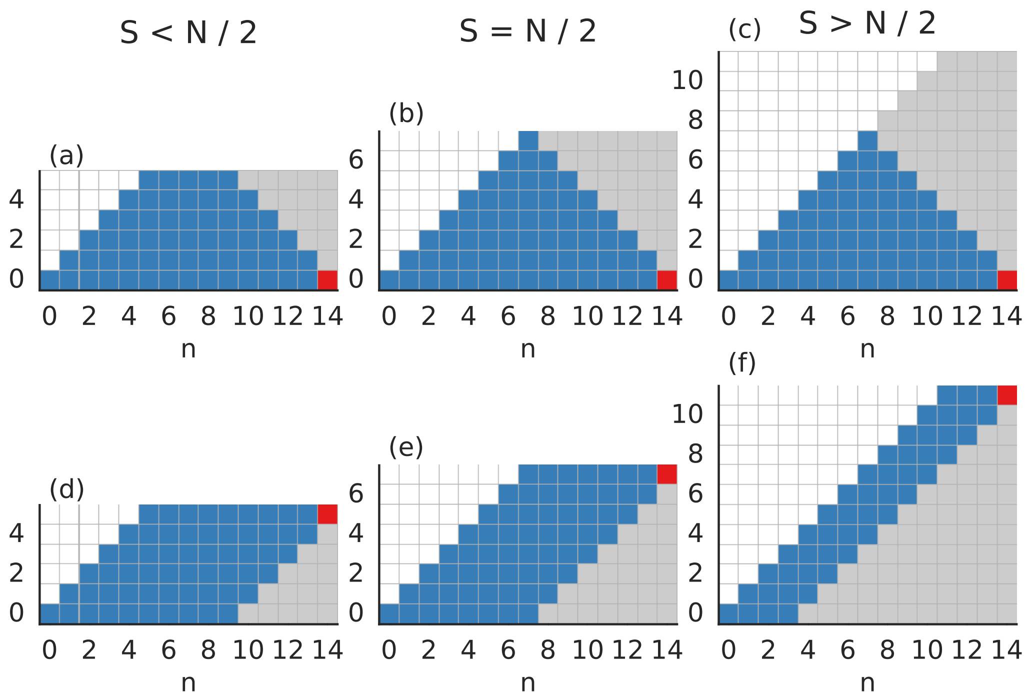


Fig. 4.2.5: KPM iteration sliced view for different system sizes. S is the number of slices which is proportional to system size and N is the total number of KPM iterations. $N = 15$ is constant while (a, d) $S = 6$, (b, e) $S = 8$ and (c, f) $S = 12$. (Top) For $|\beta\rangle = |\alpha\rangle$, e.g. for LDOS calculations. (Bottom) For $|\beta\rangle \neq |\alpha\rangle$ which is taken on the edge of the system, e.g. for Green's function. White slices contain only zeros and are not computed. Gray slices contain non-zeros but will not affect the final result and are not computed. Only blue slices contain useful values and are computed.

is to be computed. The values which are skipped during the SpMV would not have any effect on the calculation anyway, so the results are identical to the unoptimized version.

The performance gains are directly proportional to the amount of removed work which is naturally going to be more significant for larger systems. In order to quantify this, we model a circular graphene flake and calculate 3400 KPM moments while increasing the system size. The results are presented in Fig. 4.2.4 both for code executed on a single CPU core, as well as a GPU. The slicing optimization offers speedup factor which increases with system size. Note that the speedup is more significant on the GPU.

The optimized calculation time actually saturates for large systems. This behavior is explained in Fig. 4.2.5. When the system size is small, we can remove a small amount of work at the head and tail of the iteration procedure. As the size is increased, more useful work can be done (the blue slices). However, when the system becomes very large, the amount of useful work saturates for the $|\beta\rangle = |\alpha\rangle$ case. This is applicable to calculations such as the local density of states (LDOS). On the other hand, when $|\beta\rangle \neq |\alpha\rangle$ there is no saturation, but the optimization scales very well, with a large amount of work being removed at both head and tail. This greatly benefits calculation such as Green's function.

Interleaving moments

So far, we have achieved a significant speedup by identifying and removing unnecessary work. The only modification to Algorithm 1 was the replacement of the full SpMV with a partial one. That optimization benefits only the "local" variant of KPM where $|\alpha\rangle$ and $|\beta\rangle$ are unit vectors, but does not improve the "global" variant of KPM which uses random vectors (which start with all non-zero values). This second optimization benefits both variants by improving the efficiency of the iteration procedure.

Algorithm 2 Interleaved calculation of expansion moments

```
function CALCULATE_MOMENTS($\tilde{H}, \vec{\alpha}, \vec{\beta}, N, S$)
    $\vec{\alpha}_0 \leftarrow \vec{\alpha}$
    $\mu_0 \leftarrow \vec{\beta} \cdot \vec{\alpha}_0$
    $\vec{\alpha}_1 \leftarrow \tilde{H}\vec{\alpha}_0$
    $\mu_1 \leftarrow \vec{\beta} \cdot \vec{\alpha}_1$
    $\tilde{H}_2 \leftarrow 2\tilde{H}$
    for $n \leftarrow 2, 4, 6, \ldots N-1$ do                ▷ $N$ must be a multiple of 2
        for $s \leftarrow 1, 2, 3, \ldots S-1$ do        ▷ calculate vectors slice by slice
            $\vec{\alpha}_0[s+1] \leftarrow$PARTIAL_SPMV($\tilde{H}_2, \vec{\alpha}_1, \vec{\alpha}_0, s+1$)
            $\vec{\alpha}_1[s] \leftarrow$PARTIAL_SPMV($\tilde{H}_2, \vec{\alpha}_0, \vec{\alpha}_1, s$)
        end for
        $\mu_n \leftarrow \vec{\beta} \cdot \vec{\alpha}_0$
        $\mu_{n+1} \leftarrow \vec{\beta} \cdot \vec{\alpha}_1$
    end for
    return $\mu_0, \mu_1, \mu_2, \ldots \mu_N$
end function
```

It is well know that the primary performance limitation for sparse matrix-vector multiplication is the memory bandwidth available to the processor [74,

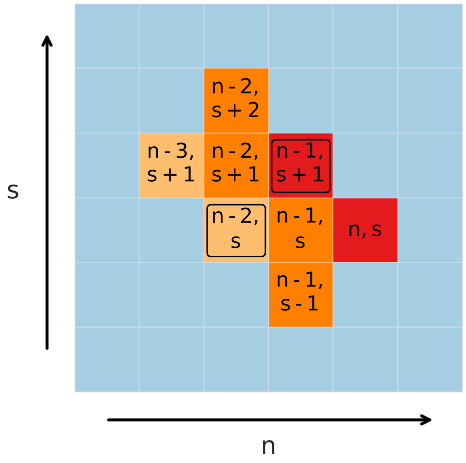


Fig. 4.2.6: Illustration of KPM vector slice dependency. The red slices are the ones being computed and the other colors represent dependent slices. The framed slices are shared between the two calculations.

75]. The computational resources of the processor (CPU or GPU) are wasted while it's waiting for data (matrix and vector elements) to arrive from main memory. This is why processors have a small amount of very fast cache memory built in. Performance is significantly improved if calculations can run directly from cache, thus bypassing main memory bandwidth concerns. In the case of KPM iteration, the $|\alpha_n\rangle$ vectors are very large and do not fit into the cache. Therefore, iteration $n + 1$ cannot benefit from the cache of iteration n and must reload all data from main memory.

Normally, KPM iterations are strictly ordered: iteration n must be fully completed before $n+1$ can begin. However, our analysis of the slice dependence from the previous section presents an opportunity for some improvements. We present a modified KPM iteration in Algorithm 2. The notable difference here is that we calculate two KPM vectors simultaneously. This is possible because one vector runs one slice ahead of the other, therefore providing the dependent data. We refer to this as "interleaving" the iterations. This can be better understood by examining Fig. 4.2.6. Slice $(n-1,\ s+1)$ is calculated first, followed by $(n,\ s)$. This order satisfies all slice dependencies as seen in the figure. A great benefit of this arrangement is that the result of one step, $(n - 1,\ s + 1)$, is immediately used as the input for the next step, therefore bypassing main memory entirely (as long as the slice is small enough to fit into cache). Slice $(n - 2,\ s)$ is similarly reused as input for both steps. As a result the required memory bandwidth is reduced significantly.

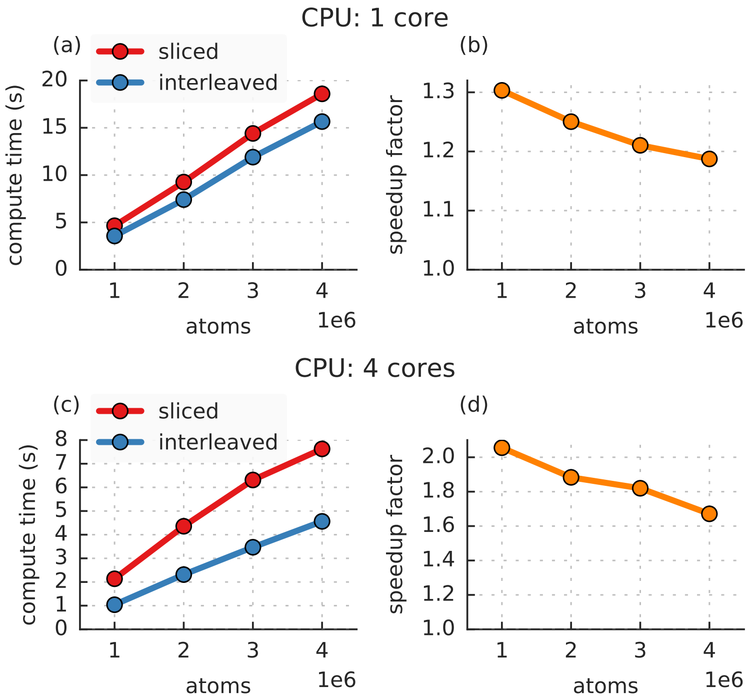


Fig. 4.2.7: Performance benefits of the interleaving optimization. (Top) Running on a single CPU core and (bottom) on multiple CPU cores. (Left) Computation time of 3.4 thousand KPM moments for a circular graphene flake with a varying number of atoms. (Right) Speedup factor obtained with the optimization. The results were gathered from Pybinding v0.8.0. The code was execute using Intel Core i7-4960HQ with 16 GiB RAM.

To measure the performance, we use the same configuration as the previous section. The results are presented in Fig. 4.2.7. This new interleaving optimization works in addition to the previous slicing optimization which we use as the baseline for comparison. When using just 1 CPU core, this shows a nice speedup factor of 1.2 to 1.3, however, a much more significant improvement is achieved when multiple cores are working simultaneously: the performance nearly doubles. The significantly better speedup for multiple cores is easily explained by memory bandwidth. No matter if 1 or 4 cores are working, they all access the same memory. A single core has a plentiful supply of bandwidth so the optimization is not as helpful. Multiple cores must compete for the same memory channel, thus any optimization which relieves memory contention is going to have a much greater impact.

This optimization is also applicable to KPM iterations which start with random vectors. Note that the speedup presented in Fig. 4.2.7 is for the unit vector variant and it is relative to the already optimized “sliced” KPM iteration described in the previous section. Overall, compared to the baseline unoptimized case, the speedup factor is about 3 to 4 times.

One downside of the interleaving optimization is that the speedup is inversely proportional to system size. As the system becomes larger so does the average size of a slice. Larger slices cannot fit into cache memory, thus limiting the performance improvement. Currently, a slice contains all atoms which have the same distance from the target atom. This makes the slice dependencies easy to determine. A possible way to further improve performance is to make even smaller sub-slices, however, this would require a more complicated dependency relation to be worked out. Nevertheless, it would be worth pursuing in the future as it could offer another speedup factor of 2 or more, especially for large systems.

The interleaving optimization as presented here does not benefit GPUs, mainly because GPUs are made to be as parallel as possible, so working only on one small slice at a time is counterproductive. However, in the future it would be possible to adapt the optimization for GPUs by better distributing the work, which would actually require first working out the aforementioned sub-slice division.

Part II

Controlling electronic properties using strain

CHAPTER 5

The strain-induced pseudo-magnetic field in graphene

The effect of mechanical strain in graphene is quite peculiar. Mathematically, in the low-energy effective Hamiltonian, strain can be described by a vector quantity analogous to a magnetic vector potential. The effect is similar to an externally applied magnetic field and thus is often referred to as the *pseudo-magnetic field*. Graphene's high strain tolerance permits the realization of very strong fields, in the range of hundreds of Tesla. With such high values, the linear expansion of the strain field offers only a partial description of the behavior. We investigate several models of the strain-induced field and suggest corrections which should be applied depending on the strain intensity.

5.1 Introduction

Graphene has many superior properties compared to traditional semiconductors, however the lack of a band gap makes electron confinement a challenge. Good conductivity does not matter if the current cannot be turned off as needed. Klein tunneling renders traditional electric barriers ineffective, thus

[*] The results of this chapter were published as:
M. Ramezani Masir, D. Moldovan, and F. M. Peeters, Solid State Commun. 175-176, 76–82 (2013).

an alternative approach is needed. It is well established that magnetic fields can control the motion of electrons in a material, but the realization of strong non-homogeneous magnetic fields can be difficult. An interesting alternative for graphene is found in the application of geometric deformations which cause electrons to behave as if they are in a magnetic field. Thus, strain engineering has become a new way to control the electronic properties of graphene [76, 77].

The two-dimensional nature of graphene enables a large number of ways to apply strain. The thin membrane can be made to conform to a substrate [78–80]. It can be pushed or pulled by atomic probes [23, 37], stretched using micromechanical beams [81], inflated with gas [82] or crumpled [83, 84]. The deformations can be used to tailor the electronic properties, i.e. to create large pseudo-magnetic fields of different shapes. To that end, some strain geometries are more useful than others. It has been predicted that applying strain with triangular symmetry results in a uniform pseudo-magnetic field of the order of 10 T [31]. Bending a graphene nanoribbon would have a similar effect [33]. A method for producing such strain structures has recently been demonstrated by using differential thermal contraction in suspended graphene [85]. The homogeneous pseudo-magnetic field will be accompanied by Landau quantization and a pseudo-quantum Hall effect [20].

Real-space geometric deformation of atoms in graphene changes the hopping amplitudes and shifts the position of the Dirac cones [36]. This shift in k-space is analogous to a magnetic vector potential. The effective vector potential was first derived in Refs. [86, 87] and was based on a tight-binding approach with the important approximation that the local strain does not alter the lattice vectors. The importance of the lattice vector deformation has been a matter of some debate. In Ref. [88], it was shown that including the deformation of the lattice vectors leads to an extra term for the effective magnetic field which is of the same order of magnitude and which differ in the different K points. But later it was shown that this extra term in the effective vector potential does not contribute to the induced pseudo-magnetic field and that subsequently there is no difference in the K-points [89]. Furthermore, in Refs. [90, 91] it was shown that in the presence of strain the Fermi velocity becomes spatial dependent.

In this chapter we examine the pseudo-magnetic field in graphene and present a systematic study of the different corrections to the vector potential and compare them with the numerically obtained full pseudo-magnetic field. We present the effective Hamiltonian that includes different contributions of strain. The previous result for the vector potential and the Fermi velocity is reobtained in our systematic expansion, along with new terms which we introduce here to accurately model higher levels of strain. As an example we present explicit analytical results for strained graphene as induced by uniaxial and triaxial strain. We find the magnetic field induced by the in-plane deformation and compared the different terms for the vector potential [86–88] with the exact numerical results for the pseudo-magnetic field.

5.2 Theoretical model

The tight-binding Hamiltonian up to the first nearest neighbor is given by

$$H = \sum_{i,j} t_0 a_i^\dagger b_j + H.c., \tag{5.2.1}$$

where t_0 is the unstrained hopping parameter and a_i and $a_i^\dagger$ (b_i and $b_i^\dagger$) are the annihilation and creation operators for an electron on sublattice A (B). In the presence of lattice deformation the hopping parameter t changes due to the strained interatomic distance. The modification of the hopping parameter due to strain is given by [36],

$$t_n = t_0 e^{-\beta(d_n/a_{cc}-1)}, \tag{5.2.2}$$

where $\beta = 3.37$ is the strain modulation factor, d_n is the length of the strained nearest neighbor distance and a_{cc} is the unstrained carbon-carbon distance. The index n refers to the 3 nearest neighbors of graphene which can have different strained lengths.

The strain field

Using the Fourier transform of the creation and annihilation operators we obtain the strained Hamiltonian as

$$H = \sum_{n,k} t_n e^{-i\vec{k}\cdot\vec{d}_n} a_k^\dagger b_k + H.c. \tag{5.2.3}$$

where the strained nearest neighbor vectors are given by

$$\vec{d}_n = (\bar{I} + \bar{u})\vec{\delta}_n. \tag{5.2.4}$$

Here $\vec{\delta}_n$ are the unstrained nearest vectors, as illustrated in Fig. 5.2.1(a), $\bar{I}$ is the unity matrix and $\bar{u}$ is the strain tensor. This tensor can be decomposed

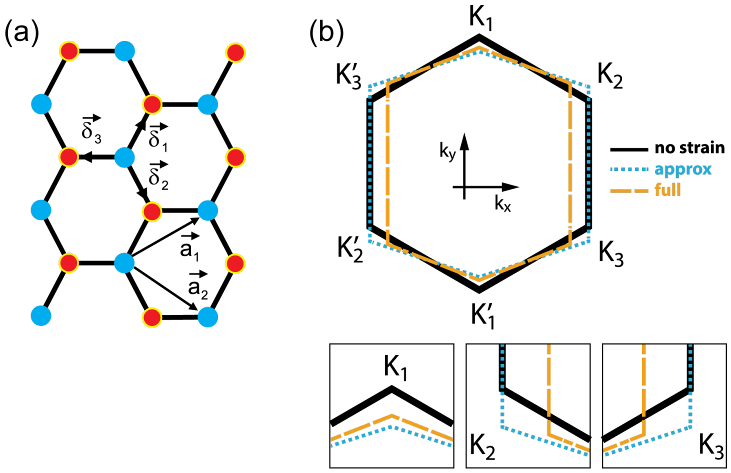


Fig. 5.2.1: (a) A monolayer graphene lattice with unit vectors $\vec{a}_1 = (3, \sqrt{3})a_{cc}/2$ and $\vec{a}_2 = (3, -\sqrt{3})a_{cc}/2$, and nearest neighbor vectors $\vec{\delta}_1 = (1, \sqrt{3})a_{cc}/2$, $\vec{\delta}_2 = (1, -\sqrt{3})a_{cc}/2$ and $\vec{\delta}_3 = (-1, 0)a_{cc}$. (b) The black line indicates the normal unstrained Brillouin zone with $\vec{K}$ points given by $\vec{K}_1 = (0, 1)4\pi/(3\sqrt{3}a_{cc})$, $\vec{K}_2 = (\sqrt{3}, -1)2\pi/(3\sqrt{3}a_{cc})$ and $\vec{K}_3 = (-\sqrt{3}, -1)2\pi/(3\sqrt{3}a_{cc})$. The blue dotted line indicates the deformed Brillouin zone for the first-order in strain approximation, while the yellow dashed line represents a full numerical result.

into two parts, $\bar{u} = \bar{\epsilon} + \bar{\omega}$, which correspond to linear and rotational strain, respectively [89, 91].

We derive the effective Hamiltonian by expanding Eq. (5.2.3) close to the $\vec{K}$ points of graphene, where $\vec{k} = \vec{K} + \vec{q}$ and considering the expansion of t_n Eq. (5.2.2) only up to the first order in strain,

$$
\begin{aligned}
H &= -\sum_{n=1}^{3} t_n \begin{pmatrix} 0 & e^{-i(\vec{K}+\vec{q})\cdot\vec{d}_n} \\ e^{i(\vec{K}+\vec{q})\cdot\vec{d}_n} & 0 \end{pmatrix} \\
&\approx -\sum_{n=1}^{3} t_n \begin{pmatrix} 0 & e^{-i\vec{K}\cdot\vec{a}_n} \\ e^{i\vec{K}\cdot\vec{a}_n} & 0 \end{pmatrix} (1 + i\sigma_z\vec{K}\cdot\bar{u}\vec{a}_n)(1 + i\sigma_z\vec{q}\cdot\vec{d}_n) \qquad (5.2.5) \\
&= -\sum_{n=1}^{3} t_0 \left(1 - \frac{\beta}{a_{cc}^2}\vec{a}_n\cdot\bar{u}\cdot\vec{a}_n\right)\left(\frac{i}{a_{cc}}(\vec{\sigma}\cdot\vec{a}_n)\sigma_z\right)(1 + i\sigma_z\vec{K}\cdot\bar{u}\vec{a}_n) \\
&\qquad \cdot (1 + i\sigma_z\vec{q}\cdot\vec{a}_n + i\sigma_z\vec{q}\cdot\bar{u}\vec{a}_n).
\end{aligned}
$$

The second order terms are included in the subsequent discussion but are not listed in the expansion of the Hamiltonian because those expressions are rather involved. The different terms of the effective Hamiltonian are shown in Table. 1. The first term is the famous Dirac-Weyl equation which governs the behavior of electrons in unstrained graphene,

$$H_0 = -\sum_{n=1}^{3} t_0 \left(\frac{i}{a_{cc}} (\vec{\sigma} \cdot \vec{a}_n) \sigma_z \right) (i\sigma_z \vec{q} \cdot \vec{a}_n) = v_F^0 \vec{\sigma} \cdot \vec{p} \tag{5.2.6}$$

Here $v_F^0 = 3t_0 a_{cc}/(2\hbar)$ is the Fermi velocity in pristine graphene and $\vec{\sigma} = (\sigma_x, \sigma_y)$ are the Pauli matrices. Next, we have three first order terms induced by strain. The first one reads,

$$H_1 = \sum_{n=1}^{3} t_0 \left(\frac{1}{a} (\vec{\sigma} \cdot \vec{a}_n) \sigma_z \right) (\sigma_z \vec{q} \cdot \bar{u} \vec{a}_n) = v_F^0 \vec{\sigma} \cdot \bar{u} \cdot \vec{p} \tag{5.2.7}$$

The second term is β-dependent and given by

$$\begin{aligned} H_2 &= \sum_{n=1}^{3} t_0 \left(\frac{\beta}{a_{cc}^2} \vec{a}_n \cdot \bar{u} \cdot \vec{a}_n \right) \left(\frac{1}{a} (\vec{\sigma} \cdot \vec{a}_n) \sigma_z \right) (\sigma_z \vec{q} \cdot \vec{a}_n) \\ &= \frac{\beta v_F^0}{4} \vec{\sigma} \cdot (2\bar{u} + Tr(u)\bar{I}) \cdot \vec{p} \end{aligned} \tag{5.2.8}$$

This is the same term as introduced in Refs. [90, 91]. The third and last term is β-independent and is given by

$$\begin{aligned} H_3 &= -\sum_{n=1}^{3} t_0 \left(\frac{i}{a} (\vec{\sigma} \cdot \vec{a}_n) \sigma_z \right) \left(i\sigma_z \vec{K} \cdot \bar{u} \vec{a}_n \right) (i\sigma_z \vec{q} \cdot \vec{a}_n) \\ &= i \frac{v_F^0 a}{2} \vec{\sigma} \cdot (\vec{K} \cdot \bar{u} \cdot \vec{\omega}) \cdot \vec{p} \end{aligned} \tag{5.2.9}$$

where $\vec{\omega} = (-\sigma_z, \sigma_x)$. In summary, considering both the β-dependent and β-independent terms, we can write the full effective Hamiltonian up to the first order in strain as

$$H_{eff} = H_0 + H_1 + H_2 + H_3. \tag{5.2.10}$$

With strain, Fermi velocity is replaced by a tensor,

$$\bar{v}_F = v_F^0 \left(\bar{I} + \frac{\beta}{4} [2\bar{u} + Tr(u)\bar{I}] \right), \tag{5.2.11}$$

which is space-dependent. Note that only the β-dependent H_2 term influences the Fermi velocity. The β value directly influences hopping energy, as per Eq. (5.2.2). The β-independent H_1 and H_3 terms only contribute as deformations of the vectors in reciprocal space, but this contribution averages to zero when considering the different $\vec{K}$ points in graphene, as we shall see shortly.

Next we derive the pseudo-magnetic field. The strain-induced vector potential $A_{ps} = A_x + iA_y$ is given by [36],

$$A_{ps} = \frac{1}{ev_F} \sum_n^3 t_n e^{-i\vec{K}\cdot\vec{d}_n}, \tag{5.2.12}$$

where t_n are the strained nearest-neighbor hopping parameters as given by Eq. (5.2.2) for strained nearest vectors $\vec{d}_n$ as given by Eq. (5.2.4). Note that A_{ps} is imaginary because strain breaks inversion symmetry in the nearest neighbor hopping. The effective pseudo-magnetic field induced by strain will shift the $\vec{K}$-points as $\vec{K}_n \rightarrow \vec{K}_n + \vec{A}_n$ (see Fig. 5.2.1(b)). Writing the wave vector $\vec{k}$ with respect to the Dirac cone using $\vec{k} = \vec{K} + \vec{q}$ and expanding the exponent and hopping parameter t_n up to second order we find:

$$\begin{aligned} t_n \approx t_0 &\left(1 + \delta t_n + \frac{1}{2}\delta t_n^2\right)\left(1 - i\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n - \frac{1}{2}(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n)^2\right) \\ &\cdot (1 - i\vec{q}\cdot\vec{\delta}_n - i\vec{q}\cdot\bar{u}\cdot\vec{\delta}_n)e^{-i\vec{K}\cdot\vec{\delta}_n}, \end{aligned} \tag{5.2.13}$$

where $\delta t = -\beta/a_{cc}^2\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n$. The effective vector potential is given by $\vec{q}$ independent terms. Keeping the hopping parameters up to second order and expanding $e^{-i\vec{K}\cdot\vec{d}_n}$ we find

$$\begin{aligned} t_n e^{-i\vec{K}\cdot\vec{d}_n} \approx t_0 e^{-i\vec{K}\cdot\vec{\delta}_n} &\Bigg(1 + \underbrace{\delta t_n}_{1} - \underbrace{i\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n}_{2} \\ &\underbrace{- i\delta t_n\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n - \frac{1}{2}(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n)^2 + \frac{1}{2}\delta t_n^2}_{3}\Bigg). \end{aligned} \tag{5.2.14}$$

The first correction term is the one obtained in Refs. [86, 87] and the second correction term was added by Kitt *et al.* [88]. The third term is the new higher order correction term which we will add.

Considering only the first term, we take constant lattice vectors $\vec{d}_n = \vec{\delta}_n$. Using the three nearest neighbors vectors in real space, as shown in Fig. 5.2.1, we obtain the vector potential in terms of the strain tensor elements,

$$\vec{A}_1 = \frac{\phi_0\beta}{4\pi a_{cc}} \begin{pmatrix} u_{xx} - u_{yy} \\ 2u_{xy} \end{pmatrix}, \tag{5.2.15}$$

N	Expansion terms	
1	$e^{-i\vec{K}\cdot\delta_n}$	no effect
2	$-i\vec{q}\cdot\vec{\delta}_n e^{-i\vec{K}\cdot\delta_n}$	base Dirac equation
3	$-i\vec{q}\cdot\bar{u}\cdot\vec{\delta}_n e^{-i\vec{K}\cdot\vec{\delta}_n}$	modifies v_F as $O(u)$
4	$-\frac{\beta}{a_{cc}^2}\left(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n\right)e^{-i\vec{K}\cdot\vec{\delta}_n}$	β-dependent $\vec{A}$ $O(u)$
5	$\frac{i\beta}{a_{cc}^2}\left(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n\right)\left(\vec{q}\cdot\vec{a}_n\right)e^{-i\vec{K}\cdot\delta_n}$	β-dependent v_F $O(u)$
6	$\frac{i\beta}{a_{cc}^2}\left(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n\right)\left(\vec{q}\cdot\bar{u}\cdot\vec{\delta}_n\right)e^{-i\vec{K}\cdot\vec{\delta}_n}$	β-dependent v_F $O(u^2)$
7	-$i\left(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n\right)e^{-i\vec{K}\cdot\vec{\delta}_n}$	$\vec{A}$ $O(u)$
8	$\frac{i\beta}{a_{cc}^2}\left(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n\right)\left(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n\right)e^{-i\vec{K}\cdot\vec{\delta}_n}$	β-dependent $\vec{A}$ $O(u^2)$
9	$\frac{\beta^2}{2a^4}\left(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n\right)^2 e^{-i\vec{K}\cdot\vec{\delta}_n}$	β-dependent $\vec{A}$ $O(u^2)$
10	$-\frac{1}{2}(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n)^2 e^{-i\vec{K}\cdot\vec{\delta}_n}$	$\vec{A}$ $O(u^2)$
11	$-\left(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n\right)(\vec{q}\cdot\vec{\delta}_n)e^{-i\vec{K}\cdot\vec{\delta}_n}$	v_F $O(u)$
12	$-\left(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n\right)(\vec{q}\cdot\bar{u}\cdot\vec{\delta}_n)e^{-i\vec{K}\cdot\vec{d}_n}$	v_F $O(u^2)$
13	$\frac{\beta}{a_{cc}^2}(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n)\left(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n\right)(\vec{q}\cdot\vec{\delta}_n)e^{-i\vec{K}\cdot\vec{\delta}_n}$	β-dependent v_F $O(u^2)$
14	$-\frac{\beta^2}{2a_{cc}^4}(\vec{\delta}_n\cdot\bar{u}\cdot\vec{\delta}_n)^2(\vec{q}\cdot\vec{\delta}_n)e^{-i\vec{K}\cdot\vec{\delta}_n}$	β-dependent v_F $O(u^2)$

Table 5.1: Different terms induced by strain in the expansion of the vector potential. Right column indicates the order of these terms in the strain, i.e. $O(u^2)$, and their effect on the different properties.

where $\phi_0 = h/e$ is the flux quantum. From here, we can obtain the pseudo-magnetic field as $\vec{B}_{ps} = \vec{\nabla}\times\vec{A}_{ps}$. We have derived the pseudo-magnetic vector potential Eq. (5.2.15) starting from an expansion around the K point. Repeating the same derivation around the K' point yields the same pseudo-magnetic vector potential, just with the opposite sign. Thus, the electrons around in the K' valley of graphene feel the opposite pseudo-magnetic field of the electrons in the K valley. This is the primary difference between the real and strain-induced magnetic fields. The real field acts the same in both valleys and breaks time reversal symmetry. The pseudo-magnetic field preserves global time reversal symmetry by acting with opposite sign in the different valleys.

Next, we consider the second term from Eq. (5.2.14). This correction includes the lattice vector deformation and gives the following extra term of the vector potential,

$$\begin{aligned}
\vec{A}_2^{K_1} &= \frac{\phi_0}{2a_{cc}}\frac{4}{3\sqrt{3}}\begin{pmatrix} u_{yy} \\ u_{xy} \end{pmatrix}, \\
\vec{A}_2^{K_2} &= \frac{\phi_0}{2a_{cc}}\begin{pmatrix} \frac{2}{3}u_{xy} - \frac{2\sqrt{3}}{9}u_{yy} \\ \frac{2}{3}u_{xx} - \frac{2\sqrt{3}}{9}u_{xy} \end{pmatrix}, \\
\vec{A}_2^{K_3} &= \frac{\phi_0}{2a_{cc}}\begin{pmatrix} -\frac{2}{3}u_{xy} - \frac{2\sqrt{3}}{9}u_{yy} \\ -\frac{2}{3}u_{xx} - \frac{2\sqrt{3}}{9}u_{xy} \end{pmatrix}.
\end{aligned} \tag{5.2.16}$$

Unlike $\vec{A}_1$, the $\vec{A}_2$ term has different values in the three K-points, as illustrated in Fig. 5.2.1(b). The three K'-points behave the same, just with the opposite sign of the vector potential. However, this second term of the vector potential does not actually contribute to the pseudo-magnetic field. It is possible to show that this effective vector potential has the form of a gradient a scalar field $\nabla\phi$. We start with

$$\begin{aligned}
A_2 &= -\frac{3t_0a_{cc}}{2}\left(\vec{K}\cdot\bar{u}\cdot\vec{s}\right) \\
&= -\frac{3t_0a_{cc}}{2}\sum_{i,j}K_i\bar{u}_{ij}s_j \\
&= -\frac{3t_0a_{cc}}{2}\sum_{i,j}K_i\left(\frac{\partial u_i}{\partial x_j}\right)s_j \\
&= -\frac{3t_0a_{cc}}{2}\vec{\nabla}(\vec{K}\cdot\vec{u})\cdot\vec{s},
\end{aligned} \tag{5.2.17}$$

and the two components of the vector potential are given by the real and complex part of A_2 as

$$\begin{aligned}
A_x &\propto \partial_x\left(\vec{K}\cdot\vec{u}\right), \\
A_y &\propto \partial_y\left(\vec{K}\cdot\vec{u}\right).
\end{aligned} \tag{5.2.18}$$

Thus, because the curl of the gradient of any scalar field is $\nabla\times(\nabla\phi) = 0$, the $\vec{A}_2$ term will never contribute to the pseudo-magnetic field, $\vec{B}_2 = \vec{\nabla}\times\vec{A}_2 = 0$.

Next, we consider the third term from Eq. (5.2.14). The effective vector potential correction here includes strain up to the second order. We find

$$\vec{A}_3^{\vec{K}} = \underbrace{-i\delta t_n\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n}_{I_1} + \underbrace{\frac{1}{2}\delta t_n^2}_{I_2} - \underbrace{\frac{t_0}{2}(\vec{K}\cdot\bar{u}\cdot\vec{\delta}_n)^2}_{I_3}, \tag{5.2.19}$$

where the vector potential consists of three parts.

The first part of this vector potential term is the different in each $\vec{K}$ point. It is given by

$$
\begin{aligned}
I_1^{K_1} &= \frac{\phi_0}{2a_{cc}}\frac{\beta}{3\sqrt{3}}\begin{pmatrix} -u_{xx}u_{yy} - 3u_{yy}^2 - 2u_{xy}^2 \\ 3(u_{xx}u_{xy} + u_{xy}u_{yy}) \end{pmatrix}, \\
I_1^{K_2} &= \frac{\phi_0}{2a_{cc}}\frac{\beta}{3\sqrt{3}}\begin{pmatrix} -2u_{xy}^2 + 3\sqrt{3}u_{xy}u_{yy} + 3\sqrt{3}u_{xx}u_{xy} - 3u_{yy}^2 - u_{xx}u_{yy} \\ -3\sqrt{3}u_{xx}^2 + 3u_{xx}u_{xy} - \sqrt{3}u_{yy}u_{xx} - 2\sqrt{3}u_{xy}^2 + 3u_{yy}u_{xy} \end{pmatrix}, \\
I_1^{K_3} &= \frac{\phi_0}{2a_{cc}}\frac{\beta}{3\sqrt{3}}\begin{pmatrix} 2u_{xy}^2 + 3\sqrt{3}u_{xy}u_{yy} + 3\sqrt{3}u_{xx}u_{xy} + 3u_{yy}^2 + u_{xx}u_{yy} \\ -3\sqrt{3}u_{xx}^2 - 3u_{xx}u_{xy} - \sqrt{3}u_{yy}u_{xx} - 2\sqrt{3}u_{xy}^2 - 3u_{yy}u_{xy} \end{pmatrix}.
\end{aligned}
\tag{5.2.20}
$$

The correction corresponding to second part is

$$
I_2^{\vec{K}} = \frac{1}{2}\delta t^2 = \frac{\phi_0\beta^2}{2a}\begin{pmatrix} \frac{1}{8\pi}(5u_{xx}^2 - 2u_{xx}u_{yy} - 4u_{xy}^2 - 3u_{yy}^2) \\ -\frac{3}{2\pi}u_{xy}(u_{xx} + 3u_{yy}) \end{pmatrix}, \tag{5.2.21}
$$

and for the third part,

$$
\begin{aligned}
I_3^{K_1} &= \frac{\phi_0}{3a_{cc}}\frac{4\pi}{9}\begin{pmatrix} u_{xy}^2 - u_{yy}^2 \\ -2u_{xy}u_{yy} \end{pmatrix}, \\
I_3^{K_2} &= \frac{\phi_0}{3a_{cc}}\frac{4\pi}{9}\begin{pmatrix} 3u_{xx}^2 - 2u_{xy}^2 - u_{yy}^2 + 2\sqrt{3}u_{xy}(u_{yy} - u_{xx}) \\ -2(u_{xy} - \sqrt{3}u_{xx})(u_{yy} - \sqrt{3}u_{xy}) \end{pmatrix}, \\
I_3^{K_3} &= \frac{\phi_0}{3a_{cc}}\frac{4\pi}{9}\begin{pmatrix} 3u_{xx}^2 - 2u_{xy}^2 - u_{yy}^2 + 2\sqrt{3}u_{xy}(u_{xx} - u_{yy}) \\ -2(u_{xy} + \sqrt{3}u_{xx})(u_{yy} + \sqrt{3}u_{xy}) \end{pmatrix}.
\end{aligned}
\tag{5.2.22}
$$

This $\vec{A}_3$ correction is of second order in the strain and is thus important for large strains and the corresponding effective field is position dependent. The most important term is $I_2 = \frac{1}{2}\delta t^2$ which is $\vec{K}$-independent and it is possible to show that the two other terms I_1 and I_3 have a non-zero contribution to the vector potential but have zero contribution to the pseudo-magnetic field.

Here we have shown that the β-dependent terms $\vec{A}_1$ and $\vec{A}_3$ contribute to the pseudo-magnetic field, while the β-independent $\vec{A}_2$ does not affect the field and can be neglected. Even thought the pseudo-magnetic vector potential can be different for all six K-points of the first Brillouin zone, the resulting pseudo-magnetic field will always have just a single value, with opposite signs in K and K'. The $\vec{A}_3$ term becomes important for the larger strains as we shall show next.

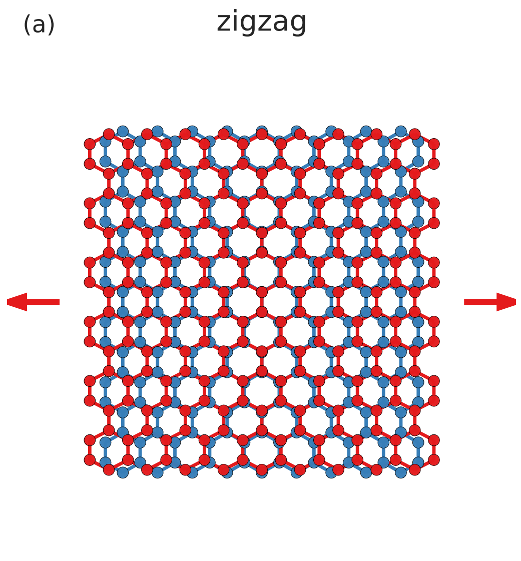


Fig. 5.3.1: Illustration of 10% uniaxial strain applied in the (a) zigzag and (b) armchair direction of graphene. The blue lattice represents unstrained graphene while the red is strained in the directions indicated by the arrows.

5.3 Uniaxial strain and direction-dependent Fermi velocity

A general strain tensor for uniaxial strain in graphene can be expressed as,

$$\bar{u} = u_0 \begin{pmatrix} \cos^2\theta - \nu\sin^2\theta & (1+\nu)\cos\theta\sin\theta \\ (1+\nu)\cos\theta\sin\theta & \sin^2\theta - \nu\cos^2\theta \end{pmatrix}, \tag{5.3.1}$$

where u_0 is the tensile strain, θ is the angle of the strain relative to the zigzag direction and $\nu = 0.165$ is Poisson's ratio [92]. Experimentally, this kind of strain can be applied by mechanically acting on the substrate to which the graphene sheet is adhered [93].

The tight-binding Hamiltonian for an infinite sheet of graphene is given by,

$$H(\vec{k}) = \begin{pmatrix} 0 & f(\vec{k}) \\ f^*(\vec{k}) & 0 \end{pmatrix}, \tag{5.3.2}$$

where,

$$f(\vec{k}) = \sum_{n=1}^{3} t_n e^{i\vec{k}\cdot\vec{d}_n}. \tag{5.3.3}$$

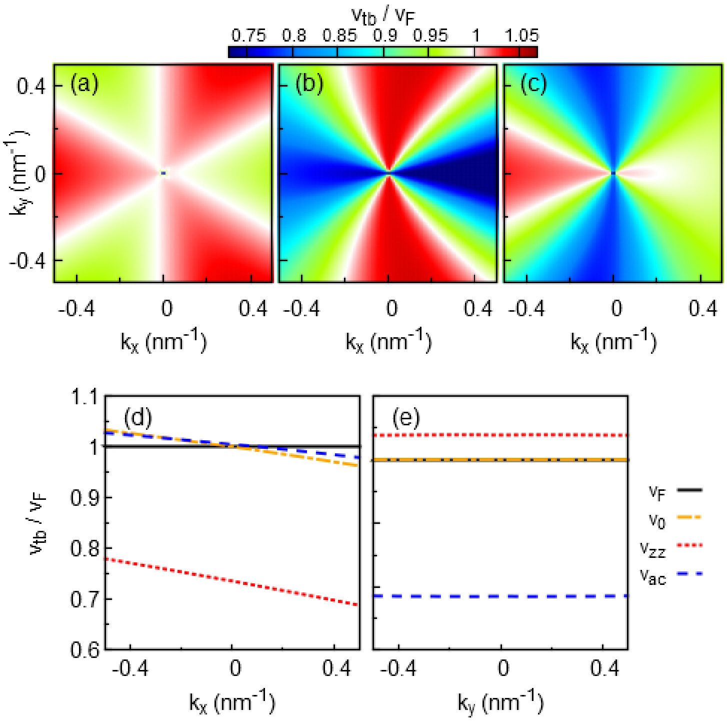


Fig. 5.3.2: Top: Contour plots of the ratio v_{tb}/v_F near the Dirac point, comparing the $\vec{k}$-dependent tight-binding Fermi velocity $v_{tb}(\vec{k})$ with the constant v_F from the continuum limit. For (a) v_0 unstrained graphene, (b) v_{zz} uniaxial zigzag strain and (c) v_{ac} uniaxial armchair strain. The strain intensity is $u_0 = 10\%$. Bottom: Fermi velocity along the cuts where (d) $k_y = 0$ and (e) $k_x = 0$.

Here, t_n is the strained hopping parameter which was previously given in Eq. (5.2.2) and $\vec{d}_n$ is the nearest neighbor vector which is deformed by the strain tensor as per Eq. (5.2.4). We calculate the energy spectrum $E(\vec{k})$ of a graphene sheet using the tight-binding Hamiltonian. The velocity can then be obtained as $\vec{v}_{tb}(\vec{k}) = \vec{\nabla}_k E(\vec{k})$.

We compute the velocity for three cases: 1) unstrained graphene, 2) graphene strained in the zigzag (zz) direction and 3) strained in the armchair (ac) direction. The strain directions are illustrated in Fig. 5.3.1. In terms of the strain tensor (5.3.1), the zigzag direction corresponds to $\theta = 0$ and armchair to $\theta = \pi/2$.

The results are presented in Fig. 5.3.2. Note that we only consider the part of the spectrum that is close to the Dirac point where the continuum limit may be applied (up to 300 meV). The $\vec{k}$-dependent Fermi velocity $v_{tb}(\vec{k})$

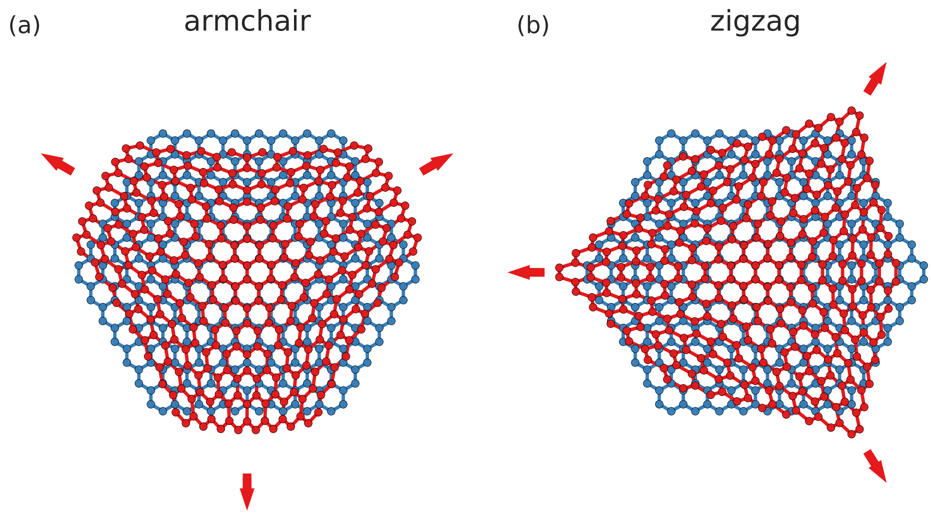


Fig. 5.4.1: Illustration of triaxial strain with $c = 0.1\text{nm}^{-1}$ applied in the (a) zigzag and (b) armchair directions of a hexagonal graphene flake. The blue lattice represents unstrained graphene while the red is strained in the directions indicated by the arrows.

obtained from tight-biding is scaled by the constant Fermi velocity $v_F = 3t_0 a_{cc}/(2\hbar)$ from the continuum limit. In the case of unstrained graphene from Fig. 5.3.2(a), the deviation of v_{tb} from v_F is generally smaller than 3%. Once strain is applied, as in Figs. 5.3.2(b,c), the velocity becomes anisotropic in k-space and deviates from the baseline v_F value by as much as 25%.

5.4 Triaxial strain and the uniform pseudo-magnetic field

Triaxial strain is produced by pulling a graphene flake from three sides, as illustrated in Fig. 5.4.1. This kind of deformation can be realized using differential thermal contraction in suspended graphene [85]. Although both strain in the armchair and zigzag directions is possible, the zigzag case does not induce a field since it preserves sublattice symmetry [35]. Here we shall only consider armchair triaxial strain which breaks sublattice symmetry and produces a uniform pseudo-magnetic field. Note that the strain in Fig. 5.4.1(a) can be rotated by 60° to produce strain in the opposite armchair directions, which results in the reveral of the induced effect, as we shall see.

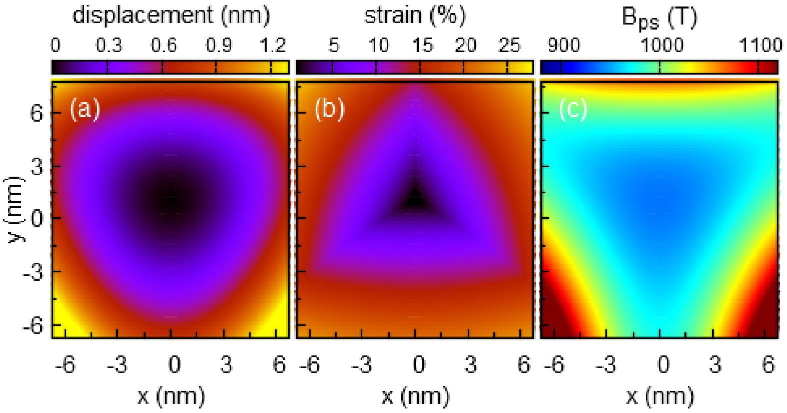


Fig. 5.4.2: Contour plots of: (a) the displacement profile $|\vec{u}(\vec{r})|$ of triaxial strain; (b) strain distribution, (c) pseudo-magnetic field calculated using the full value of the hopping parameter. The constant of the triaxial strain is: $c = 0.015$ nm^{-1}.

In polar coordinates the displacement is given by,

$$\begin{aligned} u_r &= cr^2 \sin 3\theta, \\ u_\theta &= cr^2 \cos 3\theta, \end{aligned} \tag{5.4.1}$$

where c is a constant. Translated into Cartesian coordinates, the displacement of armchair triaxial strain is given by $\vec{u}(\vec{r}) = (u_x, u_y)$,

$$\begin{aligned} u_x &= 2cxy, \\ u_y &= c(x^2 - y^2), \end{aligned} \tag{5.4.2}$$

and the corresponding strain tensor $u_{ij}(\vec{r}) = \partial_j u_i$ is,

$$\bar{u}(\vec{r}) = c \begin{pmatrix} y & x \\ x & -y \end{pmatrix}. \tag{5.4.3}$$

The pseudo-magnetic vector potential induced by strain in graphene is given by Eq. (5.2.14), and the pseudo-magnetic field is then found as $\vec{B}_{ps} = \vec{\nabla} \times \vec{A}_{ps}$. The vector potential depends on the strained hopping parameter Eq. (5.2.2), which can be expanded as,

$$t_n/t_0 = 1 + \delta t_n^{(1)} + \delta t_n^{(2)} + \delta t_n^{(3)} \ldots, \tag{5.4.4}$$

$$t_n = t_0 \left(1 - \beta w_n + \frac{1}{2}\beta^2 w_n^2 - \frac{1}{6}\beta^3 w_n^3 \ldots \right), \tag{5.4.5}$$

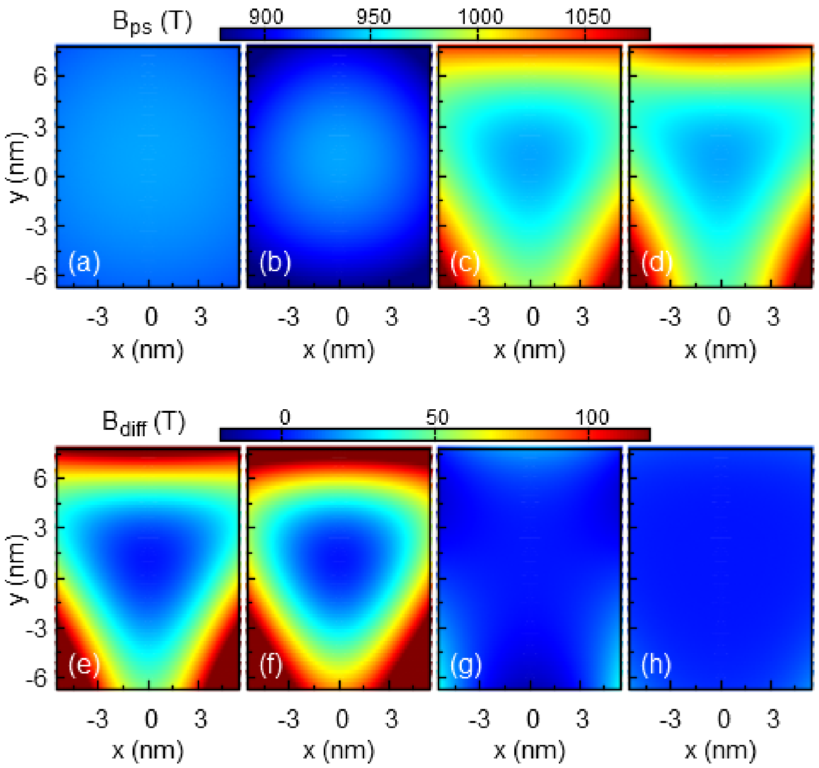


Fig. 5.4.3: Top: Contour plots of the pseudo-magnetic field generated from different approximations of the hopping parameter up to: (a) first, (b) second, (c) third and (d) fourth order. Bottom: (e,f,g,h) Difference plots between the respective approximations (a,b,c,d) and the field calculated using the full value of the hopping parameter, as in Fig. 5.4.2(d). The parameters of the triaxial strain is the same as in Fig. 5.4.2.

where $w_n = d_n/a_{cc} - 1$. Usually, only the first order term $\delta t_n^{(1)}$ is taken, where plugging in the triaxial displacement (5.4.1) and taking $\vec{B}_{ps} = \vec{\nabla} \times \vec{A}_{ps}$ yields a uniform field perpendicular to the graphene plane,

$$B_{ps} = \frac{4\hbar\beta c}{a_{cc} e}. \tag{5.4.6}$$

The direction of the pseudo-magnetic field can be reversed by rotating the angle of the strain by 60°. If we substitute $\theta \to \theta + \pi/3$ in Eq. (5.4.1), the displacement will have the opposite sign and so will the pseudo-magnetic field Eq. (5.4.6). This first order approximation is only valid near the center and only for small strain. Here, we will evaluate the effect of the inclusion of the higher order terms.

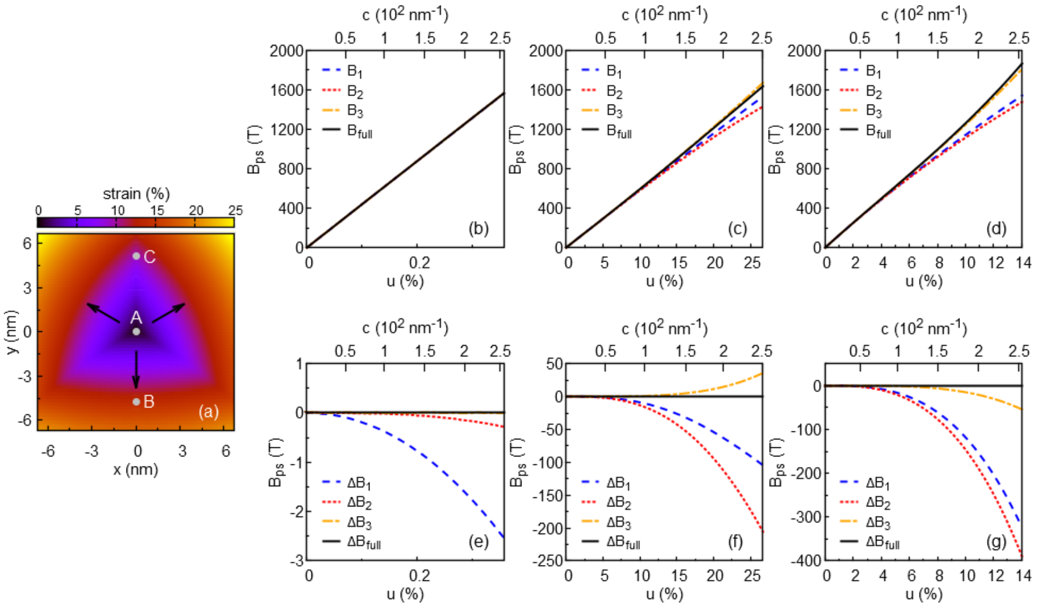


Fig. 5.4.4: (a) Contour plot of the triaxial strain with three test points marked as A, B and C. The arrows indicate the strain directions. The parameters of the triaxial strain are the same as in Fig. 5.4.2. (b,c,d) The pseudo-magnetic field as a function of strain at the three test points: (b) A, (c) B and (d) C. The field is calculated for different approximations of the hopping parameter from first to third order (B_1 to B_3 corresponds to $\delta t^{(1)}$ to $\delta t^{(3)}$), as well as the full solution (B_{full} for $\delta t^{(full)}$). (e,f,g) The differences of the pseudo-magnetic field approximations compared to the full expression ($\Delta B_i = B_i - B_{full}$) at the three test points: (e) A, (f) B and (g) C. In all cases the stain constant c is scaled from 0 to 0.025 nm^{-1}, as shown on the top x-axis. The resulting strain at the test point is shown on the bottom x-axis.

A contour plot of the displacement profile of the triaxial strain is shown in Fig. 5.4.2(a). The pseudo-magnetic field in Fig. 5.4.2(c) is calculated using the full hopping parameter from Eq. (5.2.2). The pseudo-magnetic field is mostly homogeneous in the center. Away from the center, the magnitude of the field follows the triangular shape of the displacement with high magnitudes of the pseudo-magnetic field corresponding to locations of large displacement.

In Fig. 5.4.3 we plot the pseudo-magnetic field for different approximations of the hopping parameter Eq. (5.4.4). The figures are shown in pairs, with the top ones presenting the magnitude of the field and the bottom ones presenting the difference between the approximate and the full pseudo-magnetic field calculated without approximations (as presented in Fig. 5.4.2(d)).

Taking only the first order approximation, Figs. 5.4.3(a) and (e), results in an almost completely homogeneous pseudo-magnetic field. However, this approximation is only valid in the center and it quickly deviates from the full solution as soon as we move away from the center.

With the second order approximation, Figs. 5.4.3(b) and (f), the field is less homogeneous and shows a circular symmetry. While this position-dependent intensity is an improvement, it still strongly deviates from the full field which exhibits triangular symmetry.

Calculating the pseudo-magnetic field using the third order approximation finally shows the same triangular shape as the exact pseudo-magnetic field. Adding the fourth order term further improves the accuracy, but the correct shape is already achieved with the third order approximation.

In Fig. 5.4.4 we compare the pseudo-magnetic field approximations at three points as function of applied strain. We varied the c parameter of the triaxial strain from 0 to 0.025 nm^{-1}. The first point (A) is located in the center where the strain remains very low (below 0.5%) even for high values of c. Because of the low strain, all approximations are able to accurately estimate the field.

Next, we considered point B, where the strain reaches up to 25%. Because of the higher strain, the different approximations start to diverge, although the differences aren't very large. The different approximations diverge above 15% strain but the differences remain small even above 20%.

Finally, point C shows the most significant differences. The approximations diverge already for 6% strain. At high strain, the first order approximation significantly underestimates the field (by as much as 350 T). Adding the second order term actually results in an even larger underestimation of the field. The third order term corrects the field magnitude so that it is in good agreement with the full solution.

From these results we see that a correct estimation of the field in point C is more difficult than in point B even though the maximum strain is actually lower in point C. This is because the pseudo-magnetic field depends not only on the intensity of the strain, but also on the direction. The strain in point B is mostly uniaxial, as it lies exactly along one of the three strain directions (see Fig. 5.4.4(a)). On the other hand, point C feels a strong influence from both of the top strain directions, thus it is strongly non-uniaxial.

5.5 Conclusions

We investigated the pseudo-magnetic field generated by strain using the tight-binding approach. A systematic expansion of the hopping parameter and the deformation of the lattice vectors up to second order in strain was presented. From there we reobtained the previously known $\vec{A}_1$ and $\vec{A}_2$ terms of the pseudo-magnetic vector potential for low energy electrons. Those term only consider strain up to first order. Here we introduced a new $\vec{A}_3$ term which takes strain up to second order and is required for high strains.

We have shown that the β-dependent terms $\vec{A}_1$ and $\vec{A}_3$ contribute to the pseudo-magnetic field, while the β-independent $\vec{A}_2$ does not affect the field and can be neglected. Even thought the pseudo-magnetic vector potential can be different for all six K-points of the first Brillouin zone, the resulting pseudo-magnetic field will always have just a single value, with opposite signs in K and K'.

We have also shown that the Fermi velocity becomes a tensor when strain is introduced. This direction-dependent behavior was demonstrated with a simple uniaxial strain model.

Using a numerical tight-binding model, we have also considered strain terms beyond second order and compared those contributions with the full numerical solution for the pseudo-magnetic field induced by triaxial strain. The first order term was found to be valid only in the center. Otherwise it deviated from the exact solution above 5% strain. In order to go up to graphene's full strain limit of 25%, at least the third order term was required.

CHAPTER 6

Graphene flake strained by a Gaussian bump

The effect of strain in graphene is usually modeled by a pseudo-magnetic vector potential which is, however, derived in the limit of small strain. In realistic cases deviations are expected in view of graphene's very high strain tolerance, which can be up to 25%. Here we investigate the pseudo-magnetic field generated by a Gaussian bump and we show that it exhibits significant differences as compared to numerical tight-binding results. Furthermore, we calculate the electronic states in the strained region for a hexagon shaped flake with armchair edges. We find that the six-fold symmetry of the wave functions inside the Gaussian bump is directly related to the different effects of strain along the fundamental directions of graphene: zigzag versus armchair. Low energy electrons are strongly confined in the armchair directions and are localized on the carbon atoms of a single sublattice.

6.1 Introduction

Deformations due to elastic strain in graphene change the hopping amplitude of the carbon atoms and induce an effective vector potential that shifts the Dirac points [33]. With a proper geometrical deformation it is possible to

[*] The results of this chapter were published as:
D. Moldovan, M. Ramezani Masir, and F. M. Peeters, Phys. Rev. B 88, 035446 (2013).

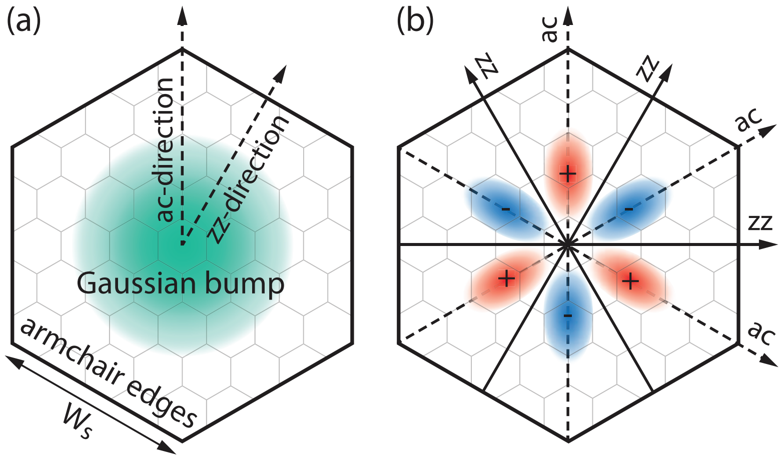


Fig. 6.1.1: (a) Hexagonal graphene flake with armchair edges. Strained is induced by a Gaussian bump in the center. The dashed arrows indicate the armchair and zigzag directions in the radial direction of the bump. (b) The bump-generated pseudo-magnetic field, as calculated from the traditional form of the pseudo-vector potential. The red (blue) color corresponds to positive (negative) magnetic field.

create large pseudo-magnetic fields which can reach up to several hundreds of Tesla [20, 32]. Over the last few years much effort has been devoted to find ways of controlling graphene's electronic properties by strain [79, 80]. Applying in-plane strain with triangular symmetry has been shown theoretically to result in an uniform pseudo-magnetic field [31]. It was reported experimentally that nanobubbles grown on a Pt(111) surface induce pseudo-magnetic fields of more than 300 T [34]. Landau quantization of the electronic spectrum was observed by scanning tunneling microscopy. Thus, with such large strain-induced pseudo-magnetic fields, one can control the electronic properties of graphene through strain engineering [36]. It was shown experimentally that an external nonuniform electric field is able to induce local deformations in graphene with different curved shapes and thus one should be able to induce a pseudo-magnetic field through an electric field [39].

In this chapter we investigate the effect of inhomogeneous, out-of-plane strain in graphene. We consider a hexagonal graphene flake that is strained by a Gaussian bump placed in its center. The effects of strain in graphene can be modeled using a pseudo-magnetic vector potential. In the case of a Gaussian bump, the traditional form of this vector potential [86] results in a three-fold symmetric pseudo-magnetic field, as illustrated in Fig. 6.1.1(b). Recently, it was shown in Ref. [88] that additional lattice corrections are required in order to accurately calculate the pseudo-magnetic vector potential. However, these strain-induced lattice vector corrections do not contribute to the pseudo-magnetic field and may be neglected [89, 91]. Only the strain induced hopping parameter changes will affect the intensity of the pseudo-magnetic field, but this is generally derived only up to first order in strain. Given graphene's excellent mechanical properties, it can sustain strain up to 25%[23]. At that point strain can no longer be considered to be small. For this reason, we investigate additional corrections to the vector potential to higher order in the strain and we compare this pseudo-magnetic field model to results obtained with the full tight-binding result.

Furthermore, we investigate the confinement of electrons inside the strained region. It was shown earlier, using the Dirac equation formalism, that such a Gaussian bump results in low energy localized states [94, 95]. However, those models do not fully explain the origin of the six-fold symmetry of the localized states. Here we investigate the system using the tight-binding model and show that the influence of strain in the zigzag (zz) and armchair (ac) directions of graphene result in different pseudo-magnetic fields and consequently to different localization properties for the electrons. Finally, we examine the energy levels and wave functions in order to show the different confinement regimes.

6.2 Theoretical model

We consider the tight-binding model of graphene with the nearest-neighbor Hamiltonian,

$$H = \sum_{m,n} t_{mn} a_m^\dagger b_n + h.c. \tag{6.2.1}$$

Here t_{mn} is the strained hopping energy between nearest-neighbor atoms at lattice positions m and n, while a_m and b_n are field operators acting respectively on sublattices A and B at their given positions. Previously, it has been shown that the strained hopping parameter is given by [30],

$$t_{mn} = t_0 \mathrm{e}^{-\beta \omega_{mn}}, \tag{6.2.2}$$

where $\omega_{mn} = l_{mn}/a_{cc} - 1$. Here $t_0 = -2.8$ eV is the unstrained hopping parameter, l_{mn} is the strained distance between atoms m and n, $a_{cc} = 0.142$ nm is the unstrained carbon-carbon distance and $\beta = 3.37$ is the strained hopping energy modulation factor. The nearest-neighbor vectors are $\vec{d}_1 =$

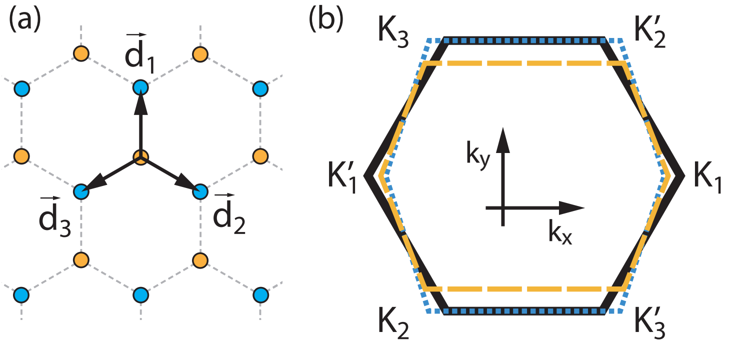


Fig. 6.2.1: (a) The unstrained nearest-neighbor vectors $\vec{d}_{n0}$. (b) The six K points in the unstrained Brillouin zone (black, solid). The zone is also shown for 20% armchair uniaxial strain, as calculated from the first approximation of the pseudo-magnetic vector potential (blue, dotted) and from the full solution of the vector potential (orange, dashed).

$a_{cc}(0,1)$, $\vec{d}_2 = a_{cc}/2(\sqrt{3},-1)$ and $\vec{d}_3 = a_{cc}/2(-\sqrt{3},-1)$ as shown in Fig. 6.2.1(a). The corresponding Brillouin zone and the six K-points are shown in Fig. 6.2.1(b).

In the present paper we consider a finite size system which is taken as a hexagon with armchair edges. There are N_S atoms on the hexagon edge, which corresponds to an edge width of $W_S = a_{cc}(3N_S/2-1)$. The total number of atoms in this hexagonal system is $N = 9N_S\,(N_S/2-1)+6$. We limit ourselves to a hexagonal system that consists only of armchair edges in order to avoid the presence of zigzag edge states which would draw attention away from the bump-induced states. In the following calculations we take an edge width of $W_S = 9$ nm, which corresponds with a flake consisting of 8322 carbon atoms. The x-axis of the system is aligned with the zigzag direction in graphene.

In our model we strain the graphene flake by using a Gaussian bump located at the center of this system as illustrated in Fig. 6.1.1(a). Such a strain profile can be induced with an STM tip [37]. The bump's height profile is given by $h(r) = h_0 \mathrm{e}^{-r^2/b^2}$, where r is the distance from the center of the system, and h_0 and b are parameters that characterize the Gaussian bump. The Gaussian function is defined to infinity ($r \to \infty$), which is inconvenient because increasing the system size would also change the total area of the bump. For that reason we add a cut-off radius R after which the height of the bump will be zero. With this cut-off the bump height profile is expressed as,

$$h(r) = h_0 \mathrm{e}^{-r^2/b^2}\Theta(R-r), \tag{6.2.3}$$

where Θ is the Heaviside step function. It is important to choose the cut-off radius R correctly in relation to the width parameter b so that the most significant part of the bump is included before the cut-off. Taking $R = 3b/\sqrt{2}$ will ensure that 99.7% of the Gaussian is inside the radius R. In the following calculations we take $R = 6.2$ nm as typically realized in experiments [34].

6.3 Corrections to the pseudo-magnetic field

The pseudo-magnetic vector potential in graphene $\vec{A}_{ps} = (\mathrm{Re}A_{ps}, \mathrm{Im}A_{ps})$ is given by [20],

$$A_{ps} = \frac{1}{ev_F}\sum_{n=1}^{3} t_n \mathrm{e}^{-i\vec{K}\cdot\vec{d}_n}, \tag{6.3.1}$$

where $\vec{d}_n$ and t_n are the strained nearest-neighbor vectors and hopping parameters, respectively, and $\vec{K}$ is the location of a K point. The pseudo-magnetic field is found as $\vec{B}_{ps} = \vec{\nabla} \times \vec{A}_{ps}$.

The strained hopping parameter from Eq. (6.2.2) can be expanded to third order as,

$$\begin{aligned} t_n &\approx t_0 + \delta t_n^{(1)} + \delta t_n^{(2)} + \delta t_n^{(3)}, && (6.3.2)\\ t_n &\approx t_0\left(1 - \beta\omega_n + \frac{1}{2}\beta^2\omega_n^2 - \frac{1}{6}\beta^3\omega_n^3\right). && (6.3.3)\end{aligned}$$

The nearest-neighbor vectors $\vec{d}_n$ are also strained, but their total contribution to the pseudo-magnetic field is zero for any strain, so they may be safely neglected [89, 91]. While their inclusion would change the value of the vector potential, the resulting field would not be affected. As we are mainly interested in the pseudo-magnetic field, we will use the unstrained values of the vectors which are constant. Because of the out-of-plane deformation, the hopping will also be affected by curvature (hybridization between π and σ bands), but this contribution may be omitted as it is 100 to 1000 times smaller than the changes induced by the bond length modulation [20].

Plugging the expansion (6.3.2) into Eq. (6.3.1), we can expand the pseudo-magnetic vector potential to third order as,

$$A_{ps} \approx \frac{1}{ev_F}\sum_{n=1}^{3}\left(\underbrace{\delta t_n^{(1)}}_{A_1} + \underbrace{\delta t_n^{(2)}}_{A_2} + \underbrace{\delta t_n^{(3)}}_{A_3}\right)\mathrm{e}^{-i\vec{K}\vec{d}_n}, \tag{6.3.4}$$

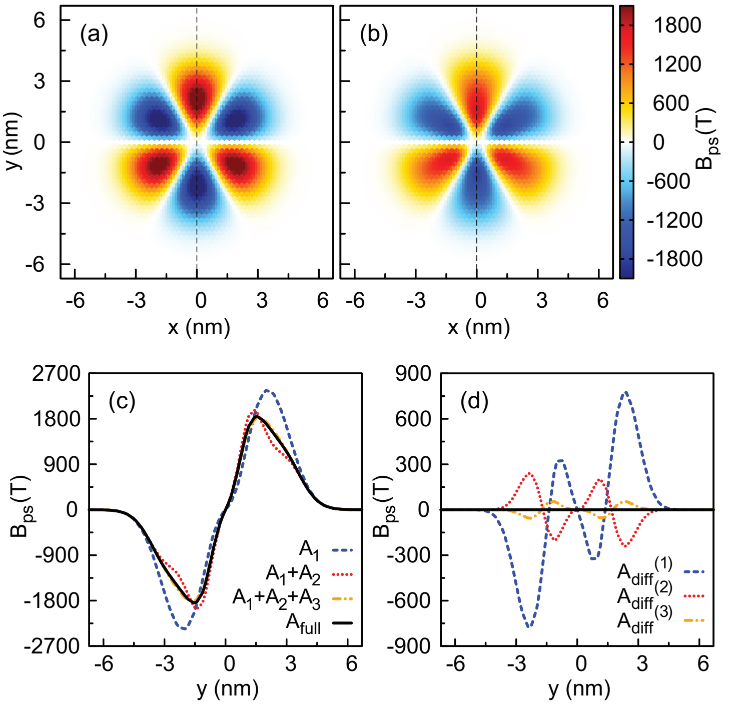


Fig. 6.3.1: Top: Contour plots of the pseudo-magnetic field generated by a Gaussian bump. The field is calculated (a) using the A_1 approximation of Eq. (6.3.4) and (b) using the full form of the vector potential Eq. (6.3.1). The dashed lines show cuts at $x = 0$, along the armchair direction of graphene. Bottom: (c) Plot of the field calculated using successively higher order terms of the vector potential approximation (A_1, A_2 and A_3) as well as the full form A_{full} from Eq. (6.3.1). (d) The difference between the approximations and full solution as $A_{diff}^{(i)} = A_i - A_{full}$. In all cases the height of the bump is $h_0 = 2.2$ nm, which corresponds to a peak strain of 20%.

which we subdivided into three parts A_i. A_1 is a first order term that was originally derived in Ref. [86]. A_2 and A_3 are second and third order terms which turn out to be important for large strain.

Figure 6.3.1(a) shows the pseudo-magnetic field calculated from the first approximation (A_1) of the vector potential. It exhibits three-fold symmetry with positive and negative peaks along the armchair directions of graphene and zero field along the zigzag directions. The pseudo-magnetic field based on the full vector potential, Eq. (6.3.1), without any approximations, is shown in Fig. 6.3.1(b). To better see the difference in field magnitude between the different approximations, we take a cut along the armchair direction of

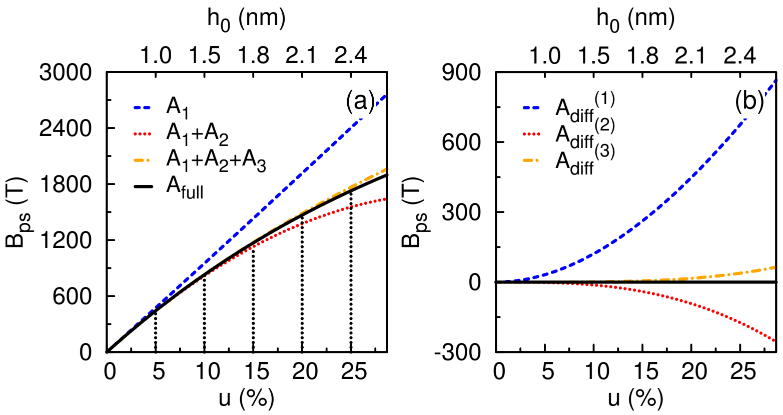


Fig. 6.3.2: (a) Pseudo-magnetic field at the location of maximum strain along the cut ($x = 0$, $y = 2.2$ nm). (b) The difference between the approximations and full solution as $A^{(i)}_{diff} = A_i - A_{full}$. The bump height h_0 is increased from 0 to 2.5 nm, as indicated on the top x-axis, which generates the strain shown on the bottom x-axis.

graphene, as show in Fig. 6.3.1(c). We compare the pseudo-magnetic field resulting from the vector potential approximations with successively higher terms included (A_1, A_2 and A_3) with the full form A_{full} from Eq. (6.3.1). The differences are shown in Fig. 6.3.1(d) as $A^{(i)}_{diff} = A_i - A_{full}$. The first order approximation A_1 overestimates the magnitude by as much as 800 T. Adding the second order corrections (A_2) will give better agreement, but there are still large deviations in the region around $y = 2.2$ nm where the strain is maximum, as well as near the center of the bump. Finally, including the third order term A_3 will result in generally good agreement.

In order to better evaluate the accuracy of the different vector potential approximations as a function of the strain, we plot the field at a fixed point while changing the bump height. As can be seen in Figs. 6.3.2(a) and (b), approximation A_1 diverges from the full solution at values as low as 5% strain. Adding A_2, we find good agreement up to about 15%, after which the field is increasingly underestimated. Finally, adding term A_3 yields good agreement up to 25% strain.

These results bring up two issues. First, even the second order term A_2 is not enough to sufficiently approximate the pseudo-magnetic field for strain above 15%. Expanding the approximation to third order would improve the results, but that would just needlessly complicate matters. Second, even if the second order term were sufficiently accurate, its form is too complicated

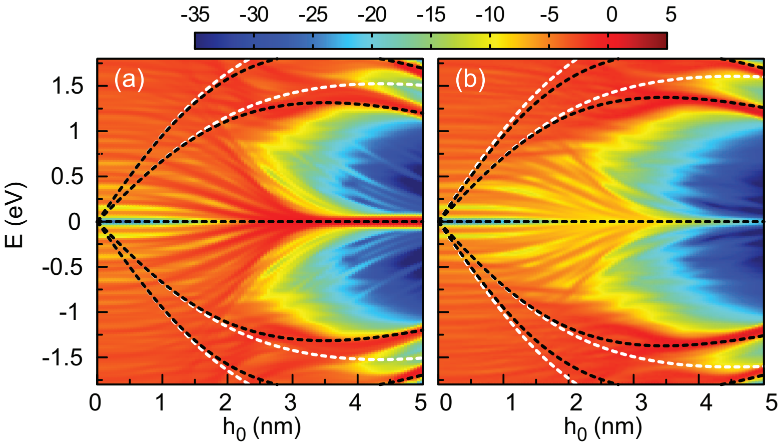


Fig. 6.4.1: Contour plot of the LDOS (in log-scale) for sublattice (a) A and (b) B as a function of bump height and energy, at the location of maximum strain ($x = 0$, $y = 2.2$ nm). The dashed curves are the Landau levels based on the pseudo-magnetic field model, calculated using unstrained (white curves) and strained (black curves) Fermi velocity.

for analytical results. On the other hand, using the numerical approach, there is no need for this, as the full vector potential Eq. (6.3.1) can easily be calculated. Thus, we find that numerical methods are best suited to correctly calculate the pseudo-magnetic field at graphene's high tolerance of strain up to 25%.

6.4 Electronic states

We derive the energy levels E_n and wave functions $\Psi(x, y)$ of the bump strained graphene flake using the tight-binding Hamiltonian Eq. (6.2.1) with the effect of strain included via the modulation of the hopping parameter given by Eq. (6.2.2). We shall compare the results from the tight-binding approach with the pseudo-magnetic field model from the previous section.

The local density of states (LDOS) is given by,

$$\rho(E, x, y) = \sum_n |\Psi(x, y)|^2 \delta(E - E_n). \tag{6.4.1}$$

To calculate the LDOS numerically we introduce a Gaussian broadening,

$$\delta(E - E_n) \to \frac{1}{\sigma\sqrt{\pi}} \exp\left[-\frac{(E - E_n)^2}{\sigma^2}\right]. \tag{6.4.2}$$

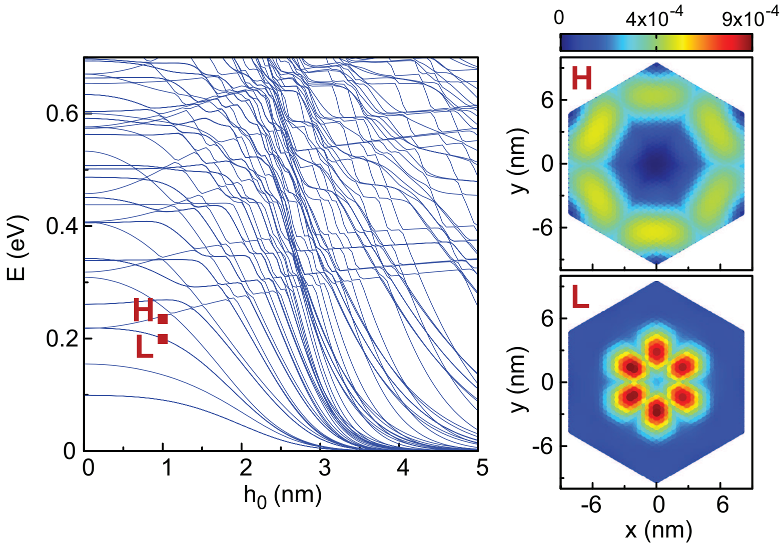


Fig. 6.4.2: Energy spectrum of a hexagonal armchair graphene flake strained by a Gaussian bump. Left: Energy levels as a function of bump height. Right: Spatial probability at the split levels marked with L and H.

As we did previously in Fig. 6.3.2, we select the location of highest strain along the armchair direction and we calculate the LDOS at that point in space as a function of bump height and energy. The results are shown in Fig. 6.4.1. At large bump heights, the LDOS shows Landau levels up to the second Landau level.

For comparison, we calculate the Landau levels using the pseudo-magnetic field model from the previous section and we overlay them on the LDOS as dashed curves in Fig. 6.4.1. In this case the pseudo-magnetic field is calculated according to the full vector potential Eq. (6.3.1). Note that the Landau levels do not follow the usual square root function. This is because the plot is a function of bump height. The Landau levels still behave as a square root of the pseudo-magnetic field.

In the first case (white dashed curves), the Landau levels are plotted for a constant unstrained Fermi velocity $v_F = 3a_{cc}t_0/2\hbar$. This does not give good agreement with the LDOS when the bump height is large. In the second case (black dashed curves), the Landau levels are fitted to the LDOS with an adjusted strained Fermi velocity $v_F^{(s)} = 3(a_{cc}t_0 + \alpha l_{mn}\delta t_{mn})/2\hbar$, where $\delta t_{mn} = t_{mn} - t_0$, and $\alpha = 0.28$ is a fitting constant.

Next, we are interested in finding the spatial localization of the different electron states. We plot the energy levels as a function of bump height in Fig.

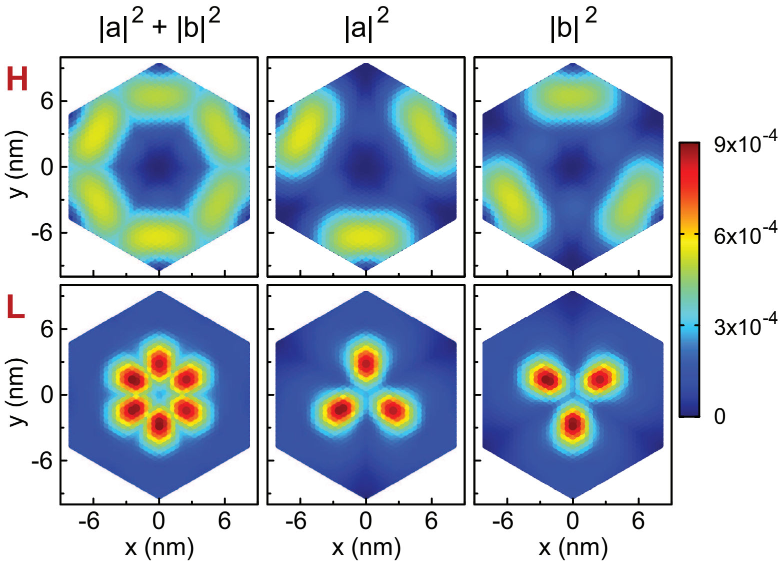


Fig. 6.4.3: Electron probability for different sublattices at the points L (bottom figures) and H (top figures) from Fig. 6.4.2.

6.4.2. The levels split into two groups: some energy levels decrease toward zero as the height of the bump increases, while the other group has the opposite behavior and increases slowly in energy with h_0. To better understand these two types of levels, we examine their wave functions. A point on a rising energy level is marked as H in Fig. 6.4.2. The spatial probability for this state (see the right panel in Fig. 6.4.2) shows an electron state localized away from the center of the system, i.e. it is localized outside the bump. As the height of the bump is increased, the confinement area between the bump and the system edge is reduced. This reduction in confinement area results in an increase of the energy.

On the other hand, we have point L which marks the level that splits downward, away from level H. The spatial probability in point L is plotted in the right bottom panel of Fig. 6.4.2. These decreasing levels are confined inside the bump, in contrast to the previous case. The probability peaks are found in the armchair directions which coincides with the peaks of the pseudo-magnetic field from Fig. 6.3.1. These levels converge toward zero energy, thus forming the zero Landau level. Because the pseudo-magnetic field is nonhomogeneous in this system, higher Landau levels are not clearly visible in the global energy spectrum.

The wave functions on the individual sublattices A and B are shown in Fig. 6.4.3 for both the L and H branches. Each sublattice contributes to three

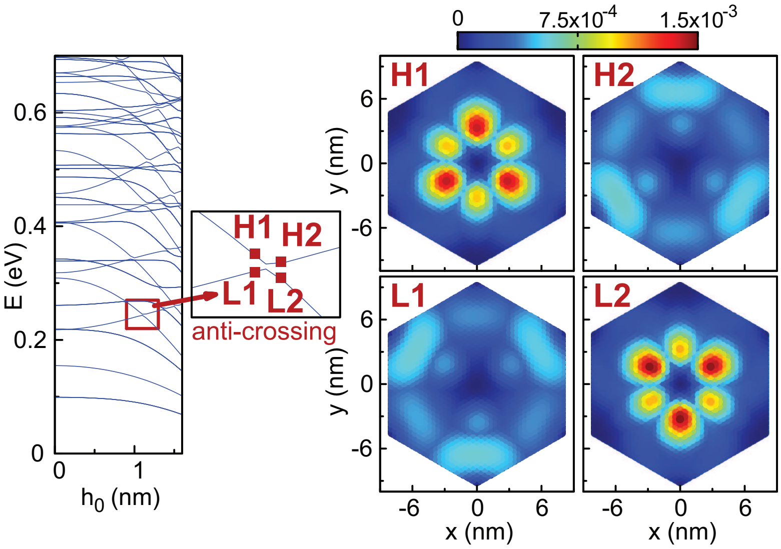


Fig. 6.4.4: Energy levels as a functions of bump height in a hexagonal graphene flake with armchair termination. Left: Anti-crossing point at bump height $h_0 = 1$ nm and energy $E = 0.24$ eV. Right: Spatial probability before and after the anti-crossing point.

of the six probability peaks. The peak heights for each sublattice are the same, but the peaks are positioned opposite to one another. The areas of high probability for sublattice A are positioned at the positive peaks of the pseudo-magnetic field, while those localized on sublattice B coincides with the negative peaks of the pseudo-magnetic field. Note that for the H level, the probability peaks are rotated by 60° in the two sublattices as compared to the L level. This also points to the different origin of these levels.

Another interesting feature of the energy levels are anti-crossing points that switch the two types of energy levels. We examine one of these anti-crossing points in detail in Fig. 6.4.4. We mark points $H1$, $H2$ ($L1$, $L2$) on the higher (lower) level before and after the anti-crossing, respectively. Following the higher level from $H1$ to $H2$, we can see a transition from confinement inside the bump to confinement outside the bump. This is consistent with the previously discussed confinement types for decreasing and increasing energy levels with h_0. Following the lower level from $L1$ to $L2$ reveals the opposite behavior, with the confinement switching from outside to inside. Note also that when we go from $H1$ ($L1$) to $L2$ ($H2$) the position of the peaks are rotated by 60°. The direction of the appearance of the peaks in the lower (higher) branch does not change when passing the anti-crossing point.

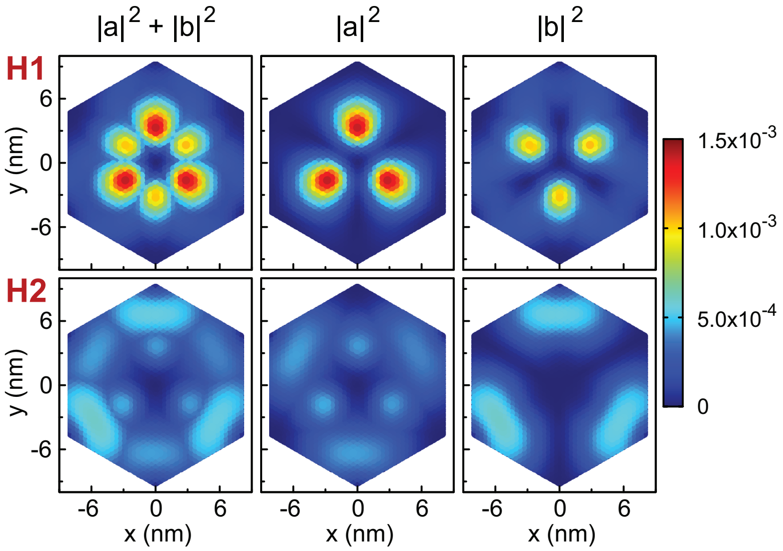


Fig. 6.4.5: Electron probability for different sublattices before and after the anti-crossing point from Fig. 6.4.4.

The probability plots around the anti-crossing point do not show perfect six-fold symmetry as we have seen in previous cases. Instead, we have two sets of three probability peaks with different magnitudes. We examine this asymmetry in Fig. 6.4.5 by plotting the separate probabilities for the two sublattices. At point $H1$, sublattice A has larger probability, but at point $H2$ (after the anti-crossing) this is reversed. Thus, following an energy level through an anti-crossing point from $H1$ to $H2$ (or $L1$ to $L2$) will result in two transitions: both the confinement type (inside or outside the bump) and the sublattice dominance are switched.

For completeness, we plot the probability distribution at the extrema of the

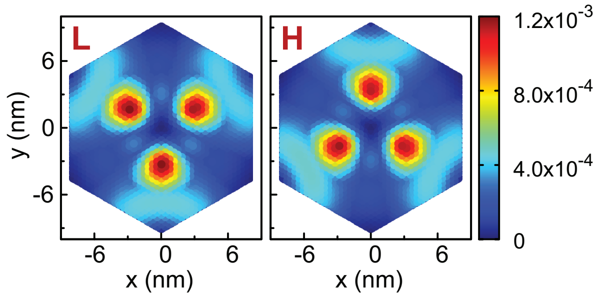


Fig. 6.4.6: Electron probability at the lower (L) and higher (H) extrema of the anti-crossing shown in Fig. 6.4.4.

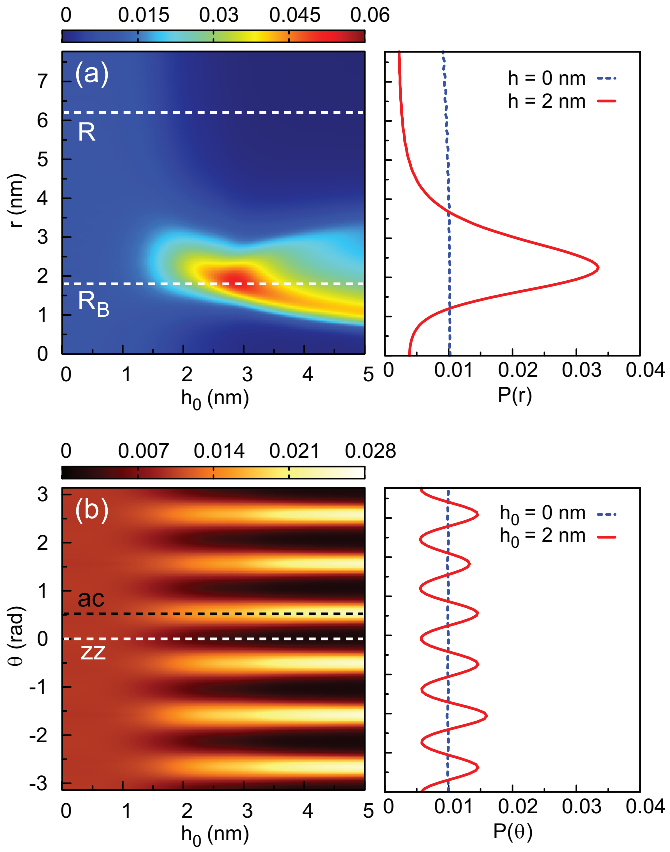


Fig. 6.4.7: Left: (a) Radial and (b) angular electron probability at low energy as a function of the bump height. The dashed line labeled R marks the radius of the bump and R_B is the spatial position of the maximum of the pseudo-magnetic field. In figure (b) the dashed lines indicate the armchair and zigzag directions in graphene. Right: Cuts of the probability at $h_0 = 0$ and $h_0 = 2$ nm. The energy is fixed at $E = 0.05$ eV.

anti-crossing in Fig. 6.4.6. Notice that they exhibit appreciable probability both inside and outside the bump. Both points are three-fold symmetric but rotated by 60° relative to one another.

Since the bump is radial, it is natural to express the electron probability in polar coordinates as $P(r, \theta)$. We are specifically interested in finding electron states that are confined inside the bump and that are not influenced by the finite size of the simulated system. We can take an integral over the angle and only leave the radius dependent part of the probability

$$P(r) = \int_0^{2\pi} P(r, \theta) d\theta. \tag{6.4.3}$$

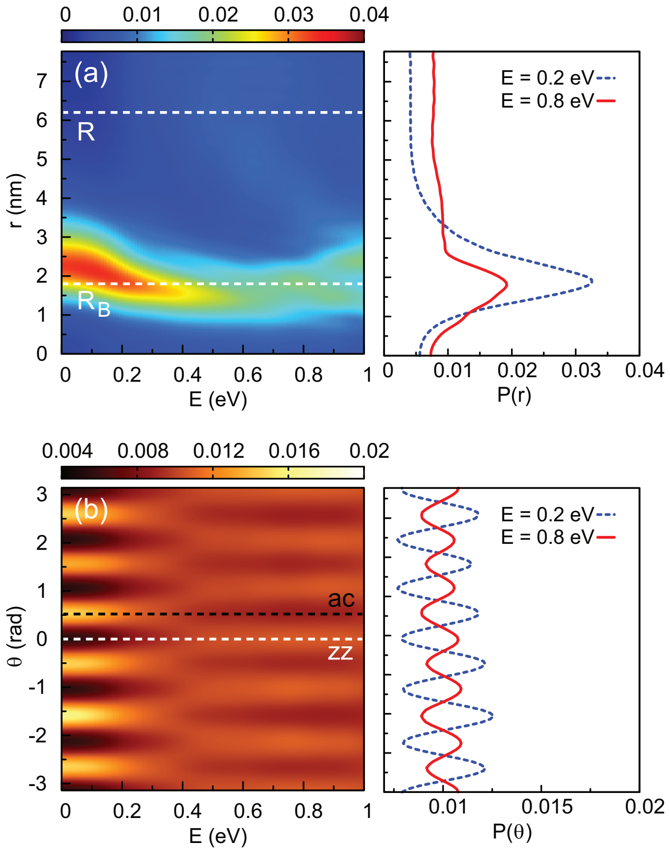


Fig. 6.4.8: Left: (a) Radial and (b) angular electron probability as a function of energy. The marked lines are the same as in Fig. 6.4.7. Right: Cuts of the probability at $E = 0.2$ eV and $E = 0.8$ eV. The bump height is fixed at $h_0 = 2$ nm.

Alternatively, we can do the opposite and take the integral over the radius, which leaves just the dependence on the angle.

The radial probability of low energy electrons as a function of bump height is shown in the left part of Fig. 6.4.7(a). As expected, the probability near zero bump height is practically uniform across the full radius of the system. As the height of the bump increases, we start seeing confinement inside the bump. More specifically, the probability peak is close to the position of the maximum of the pseudo-magnetic field, marked as R_B.

Fig. 6.4.7(b) shows the angular probability. The dashed lines indicate the armchair (black) and zigzag (white) directions of graphene, which alternate every $\pi/6$ radians. For low bump height, the angular distribution is practically uniform. As the height of the Gaussian bump increases, probability

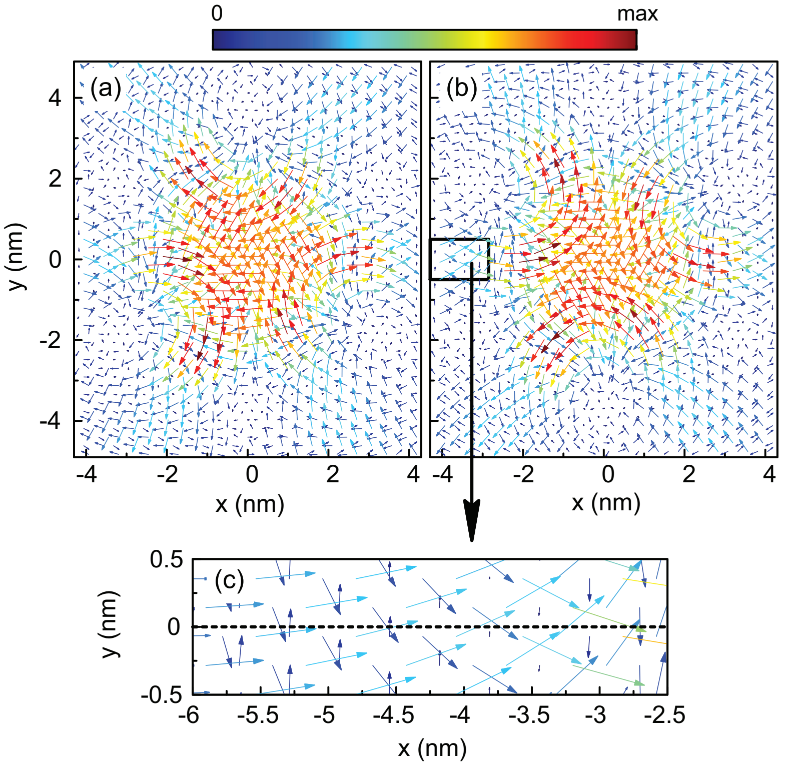


Fig. 6.4.9: Probability current for the L point electron state from Fig. 6.4.2, for sublattices (a) A and (b) B. (c) An enlarged region around snake states. The dashed line indicates the zigzag direction.

maxima start to form in the armchair directions and minima appear in the zigzag directions.

Next, we fix the bump height at $h_0 = 2$ nm and we present the probability as a function of energy in Fig. 6.4.8(a). For low energy, we find that the state is mostly confined inside the bump near the strain maximum. But for energies above 0.3 eV, we find substantial probability outside the bump and thus weaker confinement.

Looking at the angular plot in Fig. 6.4.8(b), we find probability peaks in the armchair directions and minima in the zigzag directions. As the energy increases the peaks disappear around 0.3 eV, which is the same energy where we started seeing substantial probability outside the bump in the radial plot. Once the probability outside the bump becomes substantial (above 0.4 eV), the highest probability shifts to the zigzag directions. As we will see later, the zigzag directions are associated with directions along which the probability current flows, connecting the center and the outside of the bump.

The single valley probability current at carbon atom m is given by,

$$\vec{\jmath}_m = \frac{i}{\hbar}\sum_{n=1}^{3} \Psi_m^* H_{m,m+n} \Psi_{m+n} \vec{d}_n, \tag{6.4.4}$$

where $H_{m,m+n} = -t_{m,m+n}$ is the tight-biding Hamiltonian matrix element. We plot the current inside the hexagonal flake in Fig. 6.4.9 for the same L state as in Fig. 6.4.2. For clarity, the current is plotted separately for sublattices A and B. Circular orbits coincide with the positions of the six probability peaks in the armchair directions. The current is very low at the exact positions of the probability peaks, but there is an appreciable circular current flowing around the peaks. Each sublattice contributes three circular orbits, where the sublattice A orbits have a clockwise direction and sublattice B is counterclockwise. This coincides with the probability peaks of the individual sublattices, as well as with the positive and negative peaks of the pseudo-magnetic field. Lines where the pseudo-magnetic field is zero lie along the zigzag directions. Fig. 6.4.9(c) shows the current along this line, where we find the current flowing successively across the line in both directions. These are snake orbits which are present because of the interface between the positive and negative pseudo-magnetic field.

6.5 Conclusions

We showed that a circular symmetric straining of a hexagonal graphene flake induces a non-circular symmetric pseudo-magnetic field. The average induced pseudo-magnetic field is zero and the field changes sign when we cross a zigzag direction. The pseudo-magnetic field was calculated up to third order in the strain. The first order term was found to be valid only up to 5% strain. The second order term extends the validity of the pseudo-magnetic field expression for a Gaussian bump up to 15%, while the third order is needed to go up to graphene's full strain limit (25%).

Next, we investigated the confinement of the electronic states in the same system. We found that non-uniform strain has a significantly different effect in the two fundamental directions of graphene. The six-fold symmetry of the confinement is directly related to the armchair and zigzag directions. Electrons are well confined in the armchair directions, while the zigzag directions allow the flow of probability current between the inside and outside of the strained region. This mirrors the form of the strain-generated pseudo-magnetic field, which has peaks in the armchair direction and zero magnitude in the zigzag directions.

The energy levels of the Gaussian bump system show splitting and anti-crossing states that correspond to different regions of localization of the electron. The levels that increase in energy with increasing bump height are confined between the bump and the edges of the graphene flake, while the decreasing levels correspond to states confined inside the bump. We also identified several anti-crossing points which switch the confinement type (inside or outside the bump) as well as the sublattice dominance.

We examined the probability of finding the electron at a particular position as a function of the radius and angle. At low energy there is strong confinement inside the bump near the strain maximum. At higher energy, confinement is weaker and more states are found outside the bump. As for directional confinement, we found that low energy states are well confined in the armchair directions, where we see closed circular electron orbits. Higher energy states are more likely to be found in the zigzag directions, where the probability current shows channels (i.e. snake states) to and from the center of the bump.

CHAPTER 7

Strain engineering of bilayer graphene flakes

We study the effect of mechanical deformations on the electronic properties of hexagonal flakes of bilayer graphene. The behavior of electrons induced by triaxial strain can be described by an effective pseudo-magnetic field which is homogeneous in the center of the flake. We find that in-plane strain, applied to both layers equally, can break the layer symmetry leading to different behavior in the top and bottom layers of graphene. At low energy, just one of the layers feels the pseudo-magnetic field: the zero-energy pseudo-Landau level is missing in the second layer, thus creating a gap between the lowest non-zero levels. While the layer asymmetry is most significant at zero energy, interaction with the edges of the flake extends the effect to higher pseudo-Landau levels. The behavior of the top and bottom layers may be reversed by rotating the triaxial strain by 60°.

7.1 Introduction

The possibility of stacking multiple graphene layers has led to new investigations, in particular of bilayer graphene which shares the same excellent electrical properties as found in the monolayer. Although it does not possess

[*] The results of this chapter were published as:
D. Moldovan and F.M. Peeters, Phys. Status Solidi - Rapid Res. Lett. 10, 39 (2015).

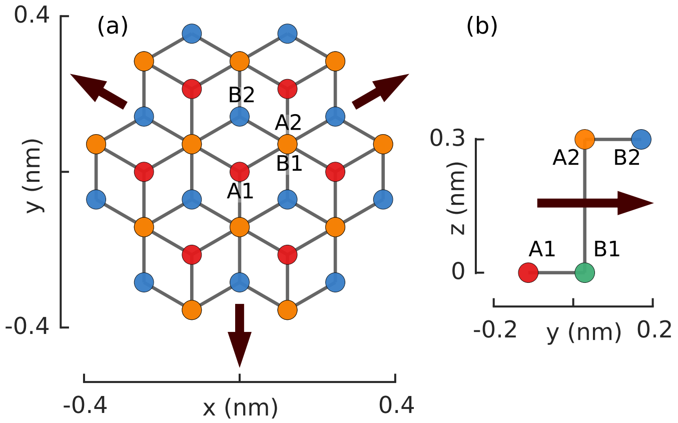


Fig. 7.1.1: Illustration of AB stacked bilayer graphene (a) from top-down (xy plane) and (b) from the side (yz plane). The arrows indicate the direction along which strain is applied. The strain is nonuniform: the intensity is zero in the center and increases in the direction of the arrows.

the monolayer's linear spectrum, pristine bilayer is still gapless with properties which make it unique among 2D electron gas systems. A gap can be opened and even tuned by applying a bias voltage between the two layers [96]. The integer quantum Hall effect in bilayer graphene introduces new physics in the form of massive chiral quasiparticles which makes it distinct from monolayer graphene [97, 98].

Bilayer graphene maintains the excellent mechanical properties of the monolayer and adds additional possibilities for applying strain, such as controlling the interlayer distance and offset to open a band gap [99, 100], changing the topology of the low-energy band structure [101] and the creation of in-plane pseudo-magnetic fields [102].

In this chapter, we study the effect of in-plane triaxial strain on hexagonal flakes of bilayer graphene. In the two-band approximation, the Landau level spectrum of bilayer graphene is expected to behave as a linear function of the magnitude of the field. While this holds for weak magnetic fields, pseudo-magnetic fields of tens or hundreds of Tesla are common which significantly alters this behavior. We show that, as function of the strain, the pseudo-Landau levels quickly deviate from a linear function, trying instead to restore a square root function reminiscent of monolayer graphene. It has been shown previously that applying in-plane strain with triangular symmetry results in a uniform pseudo-magnetic field in monolayer graphene [35]. In addition to this, we find that in bilayer graphene triaxial strain breaks the layer symmetry, leading to different top and bottom layer behavior. One of the layers is missing the zero-energy pseudo-Landau level. The majority of the low-energy states are located only on the second layer, thus polarizing the bilayer flake. The dominant layer may be chosen by changing the angle at which the triaxial strain is applied, with a full reversal every $\pi/3$ radians. We also consider finite size effects and find that the interaction of the zero-energy level with the edges of the flake extends the layer differences to higher pseudo-Landau levels.

7.2 Theoretical Model

The unit cell of bilayer graphene consists of four atoms labeled A1, B1 for the lower layer and A2, B2 for the upper layer (see Fig. 7.1.1). We consider the AB (Bernal) stacking variant of bilayer graphene where atom A2 is positioned directly on top of atom B1. Because of the strong interlayer coupling these sites are referred to as dimer sites. In contrast, A1 and B2 are non-dimer sites which don't have a direct counterpart in the other layer.

The tight-binding Hamiltonian for this structure can be written as

$$\begin{aligned} H = &\sum_m \sum_{\langle i,j \rangle} \left(\gamma_{0,i,j} a^\dagger_{m,i} b_{m,j} + H.c. \right) \\ &+ \gamma_1 \sum_i \left(b^\dagger_{1,i} a_{2,i} + H.c. \right) \\ &+ \sum_{\langle i,j \rangle} \left(\gamma_{3,i,j} a^\dagger_{1,i} b_{2,j} + H.c. \right), \end{aligned} \tag{7.2.1}$$

where $a_{m,i}$ ($b_{m,i}$) annihilates an electron at site $\mathbf{R_i}$ of sublattice A (B), in layer $m = 1, 2$. Here $\gamma_0 = -2.8$ eV is the in-plane hopping parameter, $\gamma_1 = -0.4$ eV is the interlayer hopping between the dimer sites B1 and A2, and $\gamma_3 = -0.3$ eV describes the interlayer coupling between the non-dimer sites A1 and B2. Note that the effect of skew coupling γ_3 is most significant at very low energy for magnetic fields smaller than 1 Tesla [101]. In the context of the current work, the pseudo-magnetic fields are strong enough that the effect of γ_3 may be safely neglected.

Strain is included in the hopping amplitudes according to the relation [30] $\gamma_{0,i,j} = \gamma_0 \mathrm{e}^{-\beta(|\vec{\delta}_{ij}|/a_{cc}-1)}$, where $\beta = 3.37$, $\vec{\delta}_{ij}$ is the vector which connects two neighboring atoms in the strained lattice and $a_{cc} = 0.142$ nm is the unstrained carbon-carbon distance. Since we are considering pure in-plane strain, the interlayer hopping γ_1 will remain constant. Changes of the in-plane hopping amplitudes shift the Dirac points in reciprocal space, thus inducing an effective vector potential [86, 87]

$$A_x + iA_y = \frac{1}{ev_F} \sum_{\vec{\delta}_{ij}} \delta t_{ij} \mathrm{e}^{-i\vec{K}\cdot\vec{\delta}_{ij}}, \tag{7.2.2}$$

where $\delta t_{ij} = \gamma_{0,i,j} - \gamma_0$ is the difference between the strained and unstrained in-plane hopping amplitudes. The position of the $\vec{K}$ valley is taken in the reciprocal space of the original unstrained lattice. The pseudo-magnetic field is found as $\vec{B}_{ps} = \vec{\nabla} \times \vec{A}_{ps}$. Equation (7.2.2) may be evaluated at either the K or K' point which shows that the resulting pseudo-magnetic fields will have opposite signs in different valleys: $B^K_{ps} = -B^{K'}_{ps}$.

Considering states close to the K point, in the two-band approximation of bilayer graphene, the Landau level (LL) spectrum is given by [98]

$$E_n = \pm\hbar\omega_c\sqrt{n(n-1)}, \quad n \geq 2, \tag{7.2.3}$$

where $\omega_c = eB/m$ and $m = |\gamma_1|/2v_F^2 \approx 0.04m_e$. There are two additional states fixed at zero energy: $E_1 = E_0 = 0$. Note that the LL energy is a linear function of the magnetic field B. The levels are approximately equidistant for high values of the index ($n \gg 1$). The wavefunctions in the K valley are,

$$\begin{aligned} \psi_n &= \frac{1}{\sqrt{2}}\begin{pmatrix}\phi_n \\ \pm\phi_{n-2}\end{pmatrix}, \quad n \geq 2, \\ \psi_1 &= \begin{pmatrix}\phi_1 \\ 0\end{pmatrix}, \quad \psi_0 = \begin{pmatrix}\phi_0 \\ 0\end{pmatrix}, \end{aligned} \tag{7.2.4}$$

where the wavefunction components describe the amplitudes on the A1 and B2 sites. Note that the zero-energy states ($n = 0, 1$) have a finite amplitude only on a single sublattice. In the K' valley the role of the two sublattices is reversed,

$$\begin{aligned} \psi_n &= \frac{1}{\sqrt{2}}\begin{pmatrix}\phi_{n-2} \\ \pm\phi_n\end{pmatrix}, \quad n \geq 2, \\ \psi_1 &= \begin{pmatrix}0 \\ \phi_1\end{pmatrix}, \quad \psi_0 = \begin{pmatrix}0 \\ \phi_0\end{pmatrix}. \end{aligned} \tag{7.2.5}$$

We apply in-plane triaxial strain in the armchair directions of a hexagonal bilayer flake as shown in Fig. 7.1.1. The strain is purely in-plane and applied simultaneously to both layers. In polar coordinates the displacement is given by,

$$\begin{aligned} u_r &= cr^2 \sin 3\theta, \\ u_\theta &= cr^2 \cos 3\theta, \end{aligned} \tag{7.2.6}$$

where c is a constant. The induced pseudo-magnetic field may be found by plugging the displacement into Eq. (7.2.2) and expanding up to the first order term, which results in a uniform field perpendicular to the graphene plane,

$$B_{ps} = \frac{4\hbar\beta c}{a_{cc} e}. \tag{7.2.7}$$

The direction of the pseudo-magnetic field can be reversed by rotating the angle of the strain by 60°. If we substitute $\theta \to \theta + \pi/3$ in Eq. (7.2.6), the displacement will have the opposite sign and so will the pseudo-magnetic field Eq. (7.2.7).

The pseudo-magnetic field has the same single valley behavior as a real magnetic field, so Eqs. (7.2.3) and (7.2.4) still hold, but because of the opposite sign of the field in the K' valley, Eq. (7.2.5) is modified such that the sublattice components are reversed,

$$\begin{aligned} \psi_n^{(ps)} &= \frac{1}{\sqrt{2}} \begin{pmatrix} \phi_n \\ \pm\phi_{n-2} \end{pmatrix}, \quad n \geq 2, \\ \psi_1^{(ps)} &= \begin{pmatrix} \phi_1 \\ 0 \end{pmatrix}, \quad \psi_0^{(ps)} = \begin{pmatrix} \phi_0 \\ 0 \end{pmatrix}. \end{aligned} \tag{7.2.8}$$

We consider the tight-binding Hamiltonian (7.2.1) with the strain included directly in the hopping amplitudes according to the displacement created by the triaxial strain. The pseudo-magnetic field relation Eq. (7.2.7) is valid close to the center of the system, where the strain is low. Moving away from the center, higher strain requires higher order terms of the pseudo-vector potential to be taken into account [103], thus deviating from uniform field. Additionally, at very high strain (above 15%) the elasticity theory displacement relation loses accuracy and molecular dynamics simulations are required to obtain relaxed atomic positions [104]. The numerical tight-binding model we implement here takes into account the full non-uniform spatial dependence of the strain, but we shall keep the strain low enough to stay in the validity range of elasticity theory.

We shall compare the numeric solutions of the tight-binding Hamiltonian to the two-band approximation of the Landau level spectrum Eq. (7.2.3). In

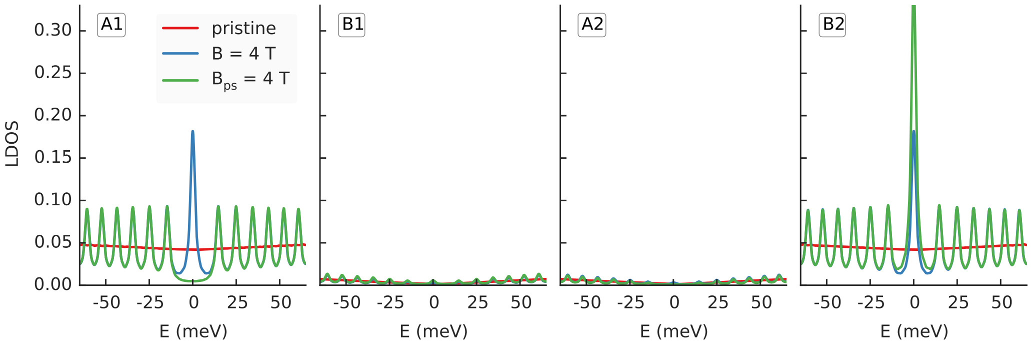


Fig. 7.3.1: Local density of states (LDOS) as a function of energy at the sublattices A1, B1, A2, and B2 of bilayer graphene with real and pseudo-magnetic fields of $B = B_{ps} = 4$ T applied. The LDOS is calculated in the center of the strained bilayer flake. A Lorentzian broadening of 2 meV is applied.

the following calculations we take a hexagonal flake with armchair edges 70 nm in width, which corresponds to a system consisting of about 3.5 million carbon atoms. The two layers consist of an equal number of atoms and have the same edges. The edge type does not influence the results in the center of the system, but we choose a system with only armchair edges in order to avoid low-energy zigzag edge states.

7.3 Numerical results

We compare the LDOS for bilayer graphene for three different situations: (1) pristine bilayer graphene, (2) in the presence of a real external magnetic field, and (3) with a strain-induced magnetic field of the same magnitude. The results are shown in Fig. 7.3.1 individually for each layer and sublattice. The low-energy bands of bilayer graphene arise from the non-dimer sites (A1, B2) which is reflected in the finite zero-energy LDOS on those sites of the pristine bilayer. The dimer sites (B1, A2) mainly contribute to the top band so their low-energy LDOS is about an order of magnitude lower than the non-dimer sites. Within the same layer, A-B sublattice symmetry is broken because of the interlayer hopping on the dimer sites. However, the A-B relationship is reversed in the second layer, thus restoring the overall symmetry.

Turning our attention to the real magnetic field case, the Landau levels appear as sharp peaks in the LDOS. The zero-energy peak on the non-dimer sites has twice the intensity as compared to the rest. This is due to the

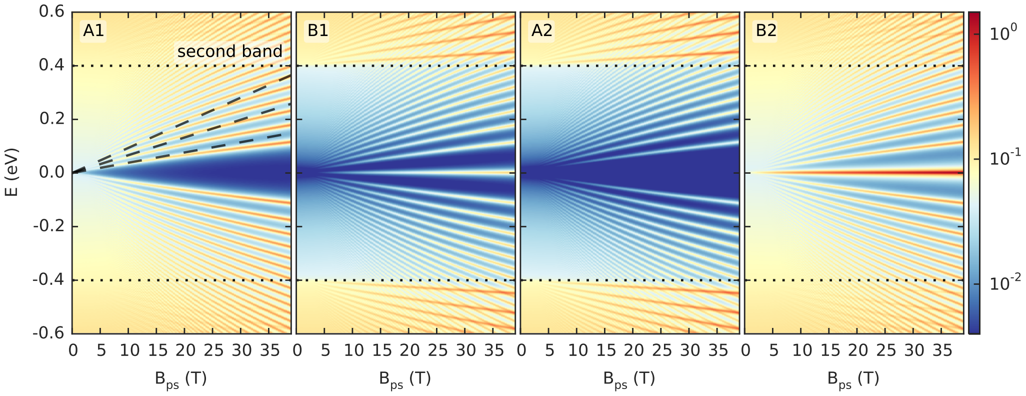


Fig. 7.3.2: Colormap of the LDOS intensity as a function of the pseudo-magnetic field and the energy for the individual sublattices. The pseudo-Landau level spectrum of bilayer graphene is visible. The horizontal dotted lines at ±0.4 eV indicate the bottom of the second band. The dashed lines in panel A1 indicate the expected LL energy based on the two-band Hamiltonian for $n = 2, 3, 4$. The LDOS is calculated in the center of the system, where the pseudo-magnetic field is approximately uniform and may be estimated by Eq. (7.2.7). The LDOS broadening is 4 meV.

twofold orbital degeneracy of the zero-energy Landau level of the bilayer. The layer symmetry is preserved with just an A-B reversal in the second layer, just like for the pristine case.

Next, we look at the strain-induced pseudo-magnetic field of the same magnitude. On the non-dimer sites, the pseudo-Landau levels coincide very well with the real ones, except for the zero-energy level, which is missing on A1, but double the magnitude on B2. This is a consequence of the pseudo-magnetic field having the opposite sign in the K and K' valleys. Normally, the zero LL only has a finite contribution from the K valley in the top layer: $\psi_K = (\phi, 0)^T$, and in K' only for the bottom layer: $\psi_{K'} = (0, \phi)^T$. However, the pseudo-magnetic field has opposite signs in K and K', thus both valleys will contribute to just a single layer: $\psi_K = (\phi, 0)^T$, $\psi_{K'} = (\phi, 0)^T$. This leads to a broken layer symmetry. Effectively, at low energy only the electrons of a single layer will behave as though they are exposed to the pseudo-magnetic field. The dominant layer may be flipped by changing the sign of the pseudo-magnetic field, i.e. by rotating the triaxial strain by 60°.

The breaking of the layer symmetry may also be seen from the point of view of a single unit cell as illustrated in Fig. 7.1.1(b). The strain is nonuniform and changes in the direction indicated by the arrow. This means that the amplitudes of the in-plane hoppings A1-B1 and A2-B2 will be different, thus

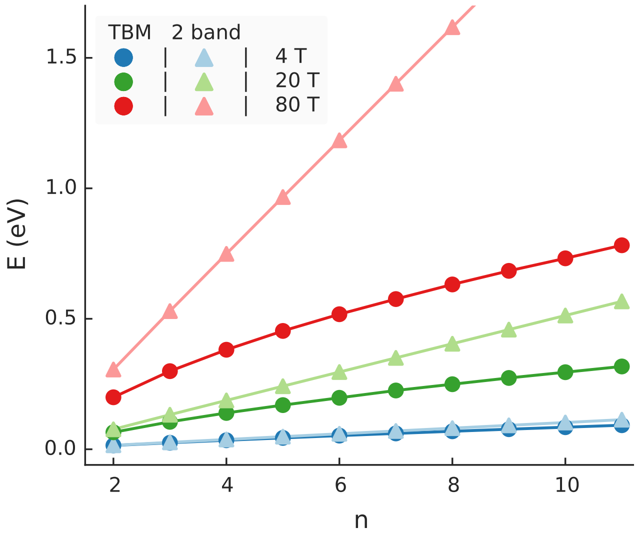


Fig. 7.3.3: Landau level energy as a function of level number n for pseudo-magnetic fields of 4, 20 and 80 T calculated numerically using the tight-binding method (TBM) and using the two-band approximation Eq. (7.2.3). Levels $n = 0$ and 1 are always zero and not shown.

breaking the symmetry between the bottom and top layers. In the case illustrated in Fig. 7.1.1, the top layer will feel higher strain, i.e. have a lower hopping amplitude as compared to the bottom layer. The zero-energy pseudo-Landau level will form on the layer with the lower hopping amplitude. For a single unit cell, the effect is reversed with the direction of the applied nonuniform strain. For the total system this is achieved by rotating the three strain directions by $\pi/3$.

In Fig. 7.3.2 we plot the LDOS as function of the pseudo-magnetic field and energy for the individual sublattices. It's quite clear that the dimer sites (B1, A2) only have a large contribution in the second band which is offset by $\pm|\gamma_1| = \pm 0.4$ eV from the Dirac point. Below that energy, the LDOS is at least an order of magnitude lower than the non-dimer sites. Note that site B2 actually does have a zero-energy Landau level which contributes to the top layer. However, its intensity is almost two orders of magnitude lower than the zero LL at site B2 in the bottom layer, which makes it negligible. The small low-energy contribution of the dimer sites is also clearly visible in Fig. 7.3.1. Effectively, the low energy spectrum of the top layer is well described by considering the non-dimer sites. In the following analysis we shall only consider the A1 and B2 sites as representing the top and bottom layers, respectively.

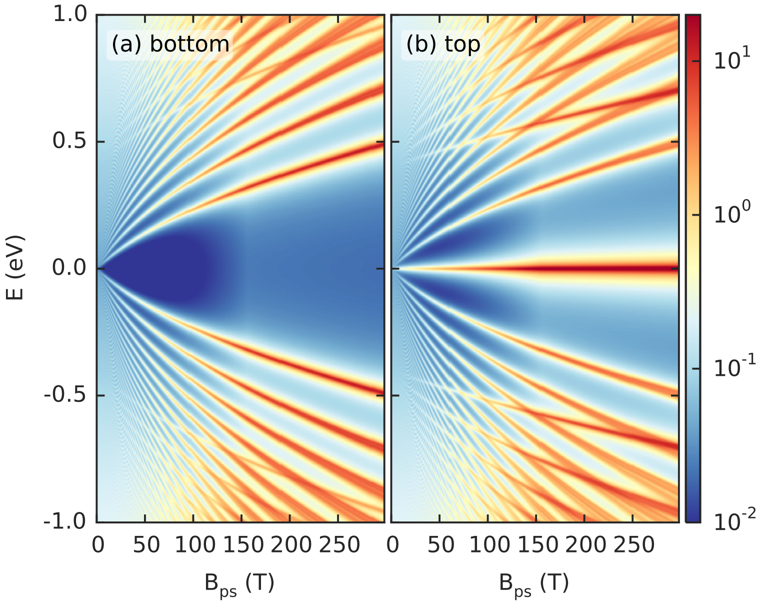


Fig. 7.3.4: Colormap of the LDOS as a function of the pseudo-magnetic field B_{ps} and energy for the (a) bottom and (b) top layer of bilayer graphene. The LDOS broadening is 5 meV.

The Landau level spectrum Eq. (7.2.3) is derived from the two-band Hamiltonian and is only valid under the condition $n\hbar\omega_c \ll \gamma_1$ which means that even the lowest levels are only accurate up to at most 10 Tesla. However, the pseudo-magnetic field may easily reach hundreds of Tesla. The dashed lines in Fig. 7.3.2 A1 show the two-band estimate of the LL energy for $n = 2, 3, 4$ and for high B_{ps} they are clearly very different compared to the LLs determined by the tight-binding calculation. In Fig. 7.3.3 we compare the Landau level energy for the first 10 levels calculated using the tight-binding method and the two-band approximation. For 4 T the two methods agree closely, while at 20 T the results start to diverge. For high values of n, the two-band LL spectrum follows approximately a linear function of n (with equidistant level spacing $\hbar\omega_c$), while the tight-binding spectrum tends to restore to a monolayer-like square root function of n at high energy. The non-linear function of the LL energy with B_{ps} is especially visible in Fig. 7.3.4, where the tight-binding Hamiltonian is used to calculate the LL spectrum for strong pseudo-magnetic fields up to 300 T.

We present a spatial map of the LDOS at zero energy for a hexagonal bilayer flake in Figs. 7.3.5(a) and (b) for the individual layers. A large part of the bottom layer has zero LDOS intensity due to the missing zero-energy

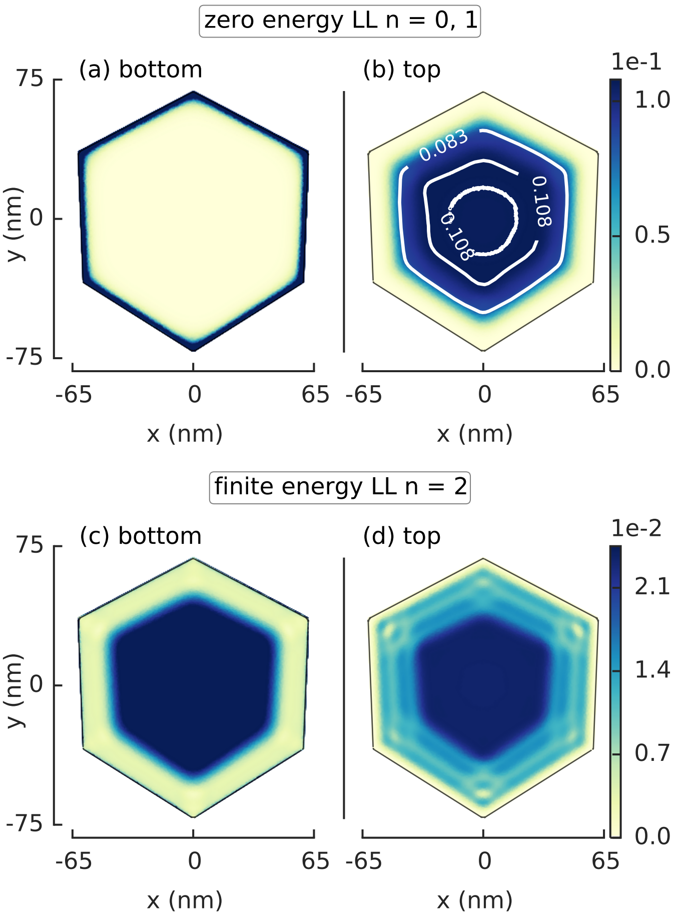


Fig. 7.3.5: Spatial map of the LDOS of a hexagonal bilayer flake with a side width of 70 nm. (a,b) At zero energy (LL $n = 0, 1$) in the bottom and top layer. (c,d) At finite energy (LL $n = 2$) in the bottom and top layer. The flake is strained to achieve $B_{ps} = 20$ T in the center.

LL states. In contrast, the top layer has high intensity in the center of the flake. The pseudo-magnetic field is not completely homogeneous due to the outwardly increasing strain, but there is still a large area (about half the total diameter) where the LDOS is nearly constant, as shown by the values of the two inner most contour lines in Fig. 7.3.5(b). The central contour is circular which is indicative of the pseudo-magnetic field, but moving outward the shape of the next contour line changes to hexagonal and the LDOS loses intensity because of the influence of the flake edges. Even though both layers are terminated with the same armchair edges, only the bottom layer has edge states with high LDOS intensity. Because of the hexagonal shape of the flake, all the edges are positioned at an angle of 60° with respect to the

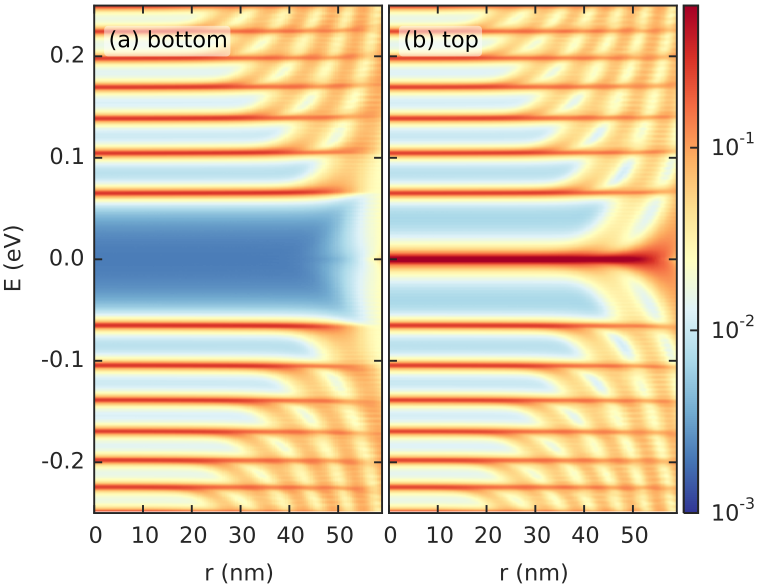


Fig. 7.3.6: Color plot of the LDOS as a function of distance from the center of the bilayer graphene flake, at a fixed angle along the armchair direction of graphene (coinciding with the direction of strain). The pseudo-magnetic field is $B_{ps} = 20$ T. The LDOS broadening is 2 meV.

strain direction and the edge atoms will only feel the angle component of the non-uniform strain. As established earlier, rotating the strain in such a way will flip the sign of the pseudo-magnetic field, thus creating the edge effect only in a single layer opposite the one which is dominant in the center of the flake.

The LDOS map at the first finite energy Landau level ($n = 2$) is shown in Figs. 7.3.5(c) and (d). In this case, Landau level states exist on both layers, but the top has an additional resonance pattern close to the edges. This is caused by the reflection of the zero-energy LL on the edges. This is visible in Fig. 7.3.6 where we plot the LDOS as a function of radial position at a fixed angle along the armchair direction of graphene (coinciding with the direction of strain). The center of the flake supports well formed pseudo-Landau levels, but they break apart at the edges. The higher energy LLs consist of states with larger orbital numbers which makes them more susceptible to finite size effects than the low energy LLs. Note that the interaction of the zero-energy LL with the edges in the top layer produces an additional resonance which is picked up in the higher energy levels. This is responsible for the LDOS pattern along the edges in Fig. 7.3.5(d).

7.4 Conclusions

We studied the effect of triaxial strain on the electronic properties of hexagonal flakes of bilayer graphene. We used the numerical tight-binding approach to obtain the Landau level spectrum of the strain-induced pseudo-magnetic field. Compared to results obtained from an effective two-band Hamiltonian, we found a good match only at relatively low magnetic fields, on the order of 1 Tesla. The strain-induced pseudo-magnetic fields are easily achievable in the range from tens to hundreds of Tesla, which requires our more accurate tight-binding calculations. In strong fields, the pseudo-Landau levels transit from a linear function of the field magnitude to a square root, as in monolayer graphene.

The triaxial strain creates a homogeneous pseudo-magnetic field in the center of the strained bilayer flake, similar to monolayer graphene. However, we found that in bilayer graphene the pseudo-magnetic field breaks the symmetry between the top and bottom layers of graphene. At low energy, the difference between the layers is so large that effectively only one layer feels the effects of the pseudo-magnetic field. The second layer is missing the zero-energy pseudo-Landau level, thus it features a large gap between the first positive and negative levels. In principle, the layer symmetry is restored for the non-zero-energy LLs, however in finite systems like graphene flakes, the interaction of the zero LL with the edges creates additional resonances which extend the layer asymmetry to higher Landau levels. The behavior of the two layers may be flipped by changing the sign of the pseudo-magnetic field, i.e. by rotating the triaxial strain by 60°.

Part III

Atomic collapse in graphene

CHAPTER 8

Atomic collapse with artificial nuclei

The relativistic wave equation for the electron, derived by Paul Dirac in 1928, predicts that heavy atoms will undergo a process called *atomic collapse.* Electronic states sink into the positron continuum triggering the spontaneous creation of electron-positron pairs. The extremely high charge requirements have prevented the observation of this phenomenon. However, thanks to the relativistic nature of the carriers and no energy gap in graphene, the same physics can be observed at a much lower scale. We discuss the atomic collapse analogue in graphene and its realization using artificial nuclei. The required supercritical charge may be created via the deposition of impurities on the surface of graphene or using charged vacancies. These artificial nuclei trap electrons in a sequence of quasi-bound states. Experimentally, these supercritical states can be observed as resonances in the local density of states close to the charge center.

8.1 Introduction

In 1928, Paul Dirac proposed a relativistic formulation of the quantum mechanics of the electron [105], which predicted that relativistic quantum effects can tip the delicate balance of stable atomic orbitals. When the charge of the nucleus exceeds a certain critical value, atomic states are rendered unstable: the energy levels of bound states dive into the lower continuum and the spontaneous generation of positrons is expected [44–46]. The semiclassical trajectories of these unstable states spiral into the nucleus, hence the name *atomic collapse.* This is a long sought-after phenomenon in quantum electrodynamics (QED). The critical charge requires extremely heavy nuclei with $Z > 170$, which has prevented the observation of this phenomenon notwithstanding several attempts using collisions of heavy ions [47, 48].

The topic of Coulomb interactions in graphene has long been of significant interest, especially its impact on the electrical properties in the presence of charged impurities [106–108]. The unusually high carrier mobility in graphene is attributed to the fact that the Dirac quasiparticles cannot be confined by electrostatic potentials or bound by stray impurities [20, 109, 110]. However, when the impurity charge exceeds a critical value it triggers a fundamental change in the electronic properties, manifest by the emergence of quasi-bound localized states in its vicinity [111–113]. This effect is analogous to the atomic collapse as predicted almost a century ago. In graphene, where the fine structure constant is much larger, the critical Coulomb coupling can be reached for relatively modest charge Z, on the order of unity [111–114]. Still, even in graphene this regime has been elusive because screening reduces the effective coupling rendering the charge subcritical [40, 115].

Atomic collapse with heavy nuclei

Classical physics cannot properly explain the structure of an atom. Because the energy is unbounded, an electron in orbit around a nucleus would continually lose energy, assume a spiraling orbit and eventually collide with the nucleus. Quantum mechanics solves this problem and naturally explains stable atomic orbits. Given a nucleus with the atomic number Z, an orbiting electron feels the Coulomb potential $U = -Ze^2/r$. Its kinetic energy is described by $K = p^2/2m$ where we can substitute the uncertainty principle to find $K \approx \hbar^2/mr^2 \propto 1/r^2$, while the potential part is $U \propto -1/r$. Being interested in conditions around the nucleus, we take the limit $r \to 0$ to find that $|K| \gg |U|$ and thus the total energy $E = K + U \to \infty$. An electron would need an infinite amount of energy to find itself at the position of the nucleus. By this fact a collision with the nucleus is made impossible and quantum mechanics leaves only one possible outcome: electron orbitals must be stable.

However, when heavy nuclei (with large Z) are considered, quantum mechanics alone does not give an accurate description and relativity must be taken into account. The relativistic electron will find itself inside the same

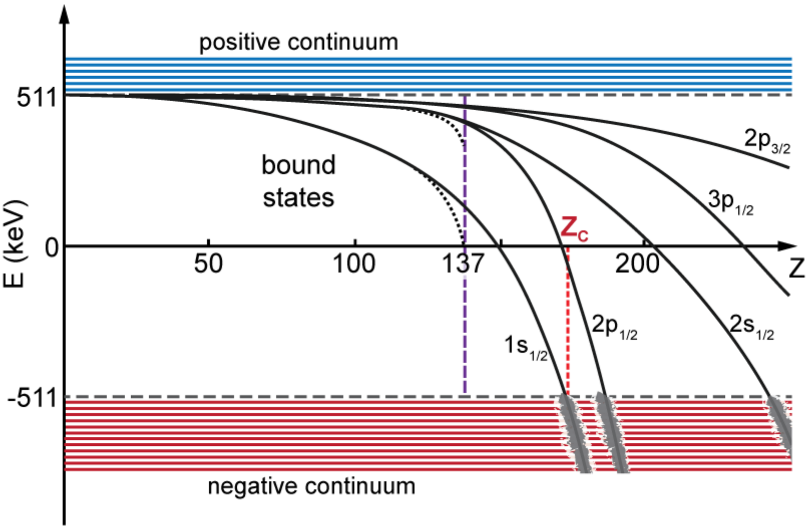


Fig. 8.1.1: Schematic drawing of the atomic bounding energies as function of the atomic number Z and the level diving process in the supercritical regime. The dotted curves indicate the solutions for a point charge while the solid curves take into account the finite size of the nucleus.

Coulomb potential as before, but now its kinetic energy must be described as $K = c\sqrt{p^2 + m_0^2c^2}$, where c is the speed of light and m_0 is rest mass. Considering m_0 to be small and substituting the uncertainty principle, we find that $K \propto 1/r$ which is the same as the potential energy but with the opposite sign. Taking the limit $r \to 0$ will not yield a definitive result this time, because both the potential and kinetic parts are of the same order. This opens the possibility of a finite energy electron state at the nucleus with disruptive results: atomic collapse.

Solving the Dirac equation for an atomic nucleus modeled as a point charge reveals the energy of the 1S atomic bound state, $E_{1S} = m_e\sqrt{1-(Z\alpha)^2}$ [46], where $m_e \approx 511$ keV is the electron rest mass and $\alpha \approx 1/137$ is the fine structure constant. Beyond the value of $Z_c\alpha = 1$, i.e. $Z_c \approx 137$, the energy becomes complex. This "collapses" the wave function and the bound state ceases to exist. However, this outcome is reached by assuming the nucleus is a point charge, which is quite unrealistic especially since it is expected to house more that a hundred protons. Taking into account the finite size of the nucleus will truncate the Coulomb potential thereby removing the divergence. This correction extends the stability beyond $Z \approx 137$ [44, 46]. The 1S state continues to exist up to the new critical value of $Z_c \approx 170$ (see Fig. 8.1.1). This region of stability where $Z < Z_c$ is referred to as the subcritical regime. When Z_c is exceeded (the supercritical regime), the electron state leaves the discrete spectrum and dives into the positron continuum. There, the bound state acquires a finite lifetime, transforming it into a narrow resonance [45, 46]. The electron state is coupled to the positron continuum via tunneling which gives rise to spontaneous positron emission which is the expected measurable signature of atomic collapse.

Because of the extreme requirements, a stable nucleus with supercritical charge cannot be found in nature, but it is possible to create one, at least for a short time. Artificial nuclei with charge values $Z > 170$ can be realized in high-energy collisions of very heavy ions. While such experiments with Uranium atoms did confirm several features of the near-collapse regime, direct proof of supercritical positron emission was not found [47, 48].

Atomic collapse analogue in graphene

The relativistic nature of the charge carriers in graphene allows it to display the same atomic collapse physics of QED but at a much smaller energy scale. The speed of light is substituted with the much smaller Fermi velocity $v_F \approx c/300$, which has a profound effect on the effective fine structure constant:

$$\alpha_G = \frac{c}{v_F}\frac{1}{\kappa}\alpha \approx \frac{2.2}{\kappa} \approx 1. \tag{8.1.1}$$

Here $\kappa \approx 2.5$ is the dielectric constant of graphene. Because the effective fine structure constant α_G is much larger than α, the value of the critical charge Z_C is expected to be much smaller, on the order of unity. This means that modest charged impurities in graphene could play the role of supercritical nuclei.

It is important to keep in mind that the effect in QED concerns real particles while the Dirac-like band structure of graphene refers to quasiparticle excitation in an effective low-energy approximation. Graphene is two-dimensional and massless while QED is three-dimensional and massive. The switch to graphene also changes the energy scale from MeV to sub-eV. Nevertheless, the physical analogue is undeniable and graphene has the key advantage: experimental accessibility. In addition, graphene experiments allow for further study by varying other parameters such as the back-gate voltage, defects or the number of layers. The results of this exploration have an effect on more than just theoretical QED, but also the understanding of confinement and transport in graphene.

8.2 Theoretical model

The tight-binding description for the π_z orbitals of carbon in graphene is given by the Hamiltonian

$$H_0 = t \sum_{\langle i,j \rangle} \left(a_i^\dagger b_j + H.c. \right) + t' \sum_{\langle\langle i,j \rangle\rangle} \left(a_i^\dagger b_j + b_i^\dagger b_j + H.c. \right), \tag{8.2.1}$$

where operators $a_i^\dagger (a_i)$ and $b_i^\dagger (b_i)$ create (annihilate) an electron on site $\vec{R}_i$ of sublattice A and B, respectively. Here $t \approx -2.8$ eV is the nearest neighbor ($\langle i, j \rangle$) hopping energy and $t' \approx 0.1$ eV is the next-nearest neighbor ($\langle\langle i, j \rangle\rangle$) hopping energy. It is sometimes assumed that the contribution of the next-nearest hopping term is negligible, especially in the low energy approximation. However, it plays an important role in graphene since it breaks electron-hole symmetry which is responsible for various effects which can be observed experimentally, especially in the presence of localized defects, as we shall see.

Considering only low-energy states close to the K point and nearest neighbor interactions ($t' = 0$), the tight-binding Hamiltonian Eq. (8.2.1) may be simplified so that the equation $H\Psi = E\Psi$ reduces to the massless Dirac-Weyl equation

$$- \mathrm{i}\hbar v_F \vec{\sigma} \cdot \vec{\nabla} \Psi(\vec{r}) = E\Psi(\vec{r}), \tag{8.2.2}$$

where σ_i are the Pauli matrices, $v_F = 3a_{cc}t/2 \approx 10^6$ m/s is the Fermi velocity, $a_{cc} \approx 0.142$ nm is the carbon-carbon distance. The wave functions $\Psi(\vec{r}) = [A(\vec{r}), B(\vec{r})]^T$ are two-component spinors where the localization of the electron on sublattices A and B plays the role of a pseudospin.

Solving the problem for pristine graphene yields the linear dispersion relation: $E(k) = \pm\hbar v_F |k|$. This corresponds to the dispersion relation of ultrarelativistic particles $E = c|p|$ but with the much smaller Fermi velocity v_F instead of the speed of light c. Furthermore, graphene's band structure describes electron and hole states which are an analogue of the relativistic electrons and positrons (particle and antiparticle states of QED).

Coulomb impurity

In the presence of a charged impurity, a local energy term must be added to the tight-binding Hamiltonian (8.2.1),

$$H_{imp} = \sum_i V_i^A a_i^\dagger a_i + \sum_i V_i^B b_i^\dagger b_i, \tag{8.2.3}$$

where $V_i^{A,B}$ is the impurity potential at site $\vec{R}_i$. A Coulomb center of charge Z generates the potential $V(r) = -\hbar v_F \beta / r$ where $\beta \equiv Ze^2/\kappa\hbar v_F$ is the dimensionless coupling constant.

The total tight-binding Hamiltonian for a Coulomb impurity in graphene is thus,

$$H = H_0 + H_{imp}. \tag{8.2.4}$$

As before, considering the effective low-energy approximation, we find that the system is governed by the wave equation

$$\hbar v_F \left(-\mathrm{i}\vec{\sigma} \cdot \vec{\nabla} - \frac{\beta}{r} \right) \Psi(\vec{r}) = E\Psi(\vec{r}). \tag{8.2.5}$$

The equation describes the behavior of Dirac fermions in a single valley of graphene's Brillouin zone. Because of the long-range nature of the Coulomb field, intervalley processes are not relevant and the problem may be solved independently within each valley.

Due to the axial symmetry of the potential, Eq. (8.2.5) is separable in cylindrical coordinates. The total angular momentum is conserved which allows us to use the eigenstates of the momentum operator $J_z = L_z + \sigma_z/2$, where L_z is the orbital angular momentum [41],

$$\Psi_j(r) = \frac{1}{\sqrt{r}} \begin{pmatrix} \mathrm{e}^{\mathrm{i}(j-1/2)\varphi} a_j(r) \\ 1\mathrm{e}^{\mathrm{i}(j+1/2)\varphi} b_j(r) \end{pmatrix}. \tag{8.2.6}$$

Here $j = \pm 1/2, \pm 3/2, ...$ are the eigenvalues of J_z. Equation (8.2.5) therefore reduces to,

$$\begin{bmatrix} \varepsilon + \beta/r & -(\partial_r + j/r) \\ (\partial_r - j/r) & \varepsilon + \beta/r \end{bmatrix} \begin{bmatrix} a_j(r) \\ b_j(r) \end{bmatrix} = 0, \tag{8.2.7}$$

where $\varepsilon = E/\hbar v_F$. This coupled pair of first order differential equations can be reduced to two decoupled second order equations. In the limit $r \to 0$ the solution behaves as

$$\varphi_j(r) \sim r^\gamma, \tag{8.2.8}$$

where

$$\gamma = \sqrt{j^2 - \beta^2}. \tag{8.2.9}$$

This reveals a problem for the lowest angular momentum ($j = \pm 1/2$), because γ becomes imaginary if $\beta > \beta_c = 1/2$. In this case the solution oscillates endlessly towards the center as $\mathrm{e}^{i \log r}$. From a classical perspective (see Fig. 8.2.1) this can be understood as a critical angular momentum above which the orbits spiral and fall into the potential origin [113]. Thus, a charge in graphene which exceeds the critical value β_c is seen as a supercritical nucleus which triggers the analogue of the atomic collapse phenomena.

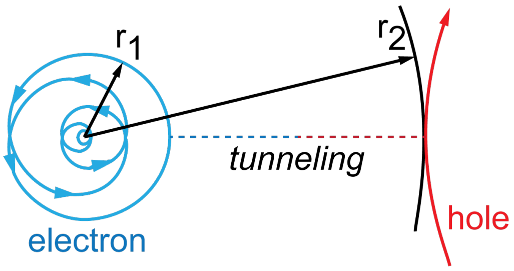


Fig. 8.2.1: Quasiclassical electron trajectory of atomic collapse in graphene. Collapsing electron coupled to escaping hole trajectory.

Supercritical regime

As we have just shown, the quantum-mechanical problem is ill-defined for a point charge in the supercritical regime ($\beta > 1/2$). An additional boundary condition must be introduced to cut off the potential at short distances. This is analogous to the introduction of the finite size of the nucleus in QED [45]. The modified potential reads

$$V(r) = \begin{cases} -\hbar v_F \frac{\beta}{r_0}, & \text{if } r \leq r_0 \\ -\hbar v_F \frac{\beta}{r}, & \text{if } r > r_0 \end{cases}, \tag{8.2.10}$$

where r_0 is the cutoff. For graphene the natural minimal cutoff length originates from the lattice structure, i.e. $r_0 = a_{cc}$, the distance between two nearest carbon atoms. A larger cutoff may be taken (depending on the nature of the charge center) but the lattice cutoff is a natural minimum distance required to regularize the potential and permit the solution to extend into the supercritical regime.

The charge may also be located a significant distance away from the graphene plane in which case the Coulomb potential may be regularized in the alternative form of $V(r) = -\hbar v_F \beta/\sqrt{r^2 + d^2}$, which corresponds to a charge β located a distance d from the graphene plane [116].

With the potential regularized with cutoff r_0, the spectrum of the supercritical states behaves as [113]

$$E_{nm} \approx -\hbar v_F \frac{\beta}{r_0} \mathrm{e}^{-\pi n/\sqrt{\beta^2 - \beta_m^2}}. \tag{8.2.11}$$

The energy scale is set by the regularization cutoff r_0. Because there is no gap in the energy spectrum, true bound states are not possible so these are quasi-bound states (resonances). Note that there is a singularity at the critical

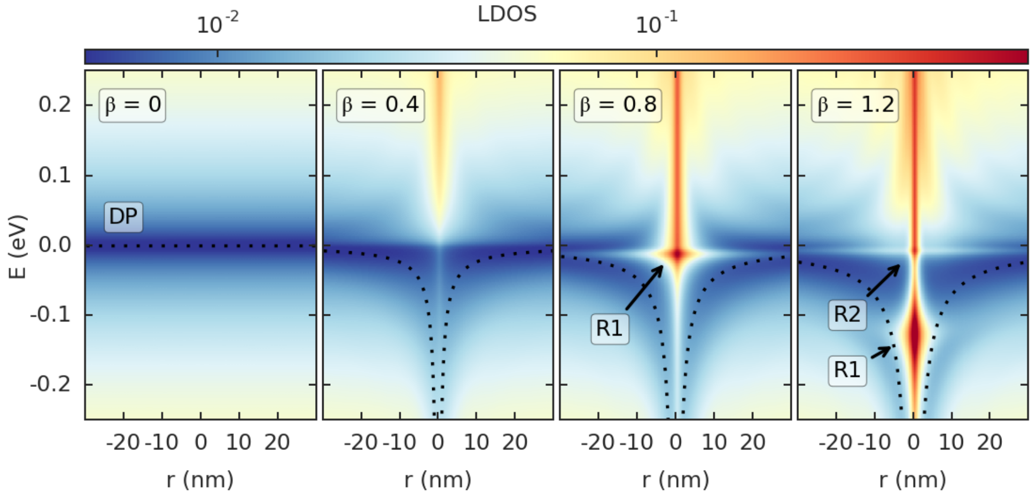


Fig. 8.2.2: Colormap of the LDOS as a function of position and energy. The charge is located in the center ($r = 0$). The coupling β varies from 0 to 1.2 as labeled in the panels. The dotted line in the first panel indicated the Dirac point (DP). In the following panels the dotted line presents the shape of the Coulomb potential. The $R1$ and $R2$ labels indicate the collapse resonances which appear in the supercritical regime $\beta > 0.5$.

value $\beta = \beta_m$ and the states are only defined for $\beta > \beta_m$. The critical value is defined by the zeros of Eq. (8.2.9), i.e. $\beta_m = |j|$ which can be rewritten as $\beta_m = m + 1/2$ where $m = 0, 1, 2...$ is the angular momentum number. Thus, there are multiple sets of collapse resonances with the principal quantum number n and angular number m, akin to regular atomic states. The lowest angular momentum value of $m = 0$ gives us the critical minimum charge $\beta_c = \beta_0 = 1/2$. These energy levels are plotted as black curves in Fig. 8.3.3 where $r_0 = 0.5$ nm. They will be discussed in more detail later in the text.

For a measurable feature of the supercritical states, we turn to the local density of states (LDOS). We calculate the LDOS numerically using the kernel polynomial method (KPM) taking the full form of the tight-binding Hamiltonian (8.2.4). The results are presented in Fig. 8.2.2. Without any charge ($\beta = 0$), the LDOS is position-independent and V-shaped as expected for pristine graphene. With a subcritical charge ($\beta = 0.4$), we see a higher LDOS intensity above the Dirac point (DP) directly at the impurity ($r = 0$). This is associated with resonances above the quantum well which consist purely of electron states. However, there are no states inside the Coulomb potential at this subcritical charge.

Once the charge becomes supercritical ($\beta = 0.8$), a resonance is visible in the LDOS below the Dirac point. This supercritical resonance is labeled $R1$ in Fig. 8.2.2 and it corresponds with the $n = 1$ state of the supercritical spectrum Eq. (8.2.11). The resonance consists of coupled electron-hole states as illustrated with the quasiclassical trajectory in Fig. 8.2.1. The electron state inside the Coulomb potential can tunnel out as a hole, thus it is only quasi-bound.

As the charge β is increased further ($\beta = 1.2$), resonance $R1$ moves down to lower energy and a second resonance emerges just below the DP. It is labeled $R2$ and corresponds with the $n = 2$ state of Eq. (8.2.11). As determined previously in Ref. [113], the resonance broadening in energy is directly proportional to $|E|$ (its energy relative to the Dirac point), while its spatial extent is inversely proportional to the same value.

The $R1$ and $R2$ resonances in the LDOS are features which can be observed experimentally. They correspond to the inner electron part of the collapse wave function. This is in contrast to the QED heavy ion experiments which look for the outer part of the wavefunction: the outgoing positron.

8.3 Supercritically charged vacancy

The resonance peak which appears in the LDOS of a supercritical charge is a clear, experimentally observable signature of atomic collapse in graphene. The LDOS can be measured with scanning tunneling microscopy (STM). An atomically sharp tip scans over a sample of graphene and measures the electric current I which flows between the tip and the surface as a consequence of the tunnelling effect. The current depends on the relative voltage V between the tip and the sample. The derivative of I with respect to V (differential conductance) is proportional to the local density of states,

$$\rho(E, r) \propto \frac{dI}{dV}. \tag{8.3.1}$$

Even though the critical value is relatively modest in graphene, introducing a sufficiently large impurity charge on graphene's surface is still challenging [115]. Because graphene is a good conductor, it is difficult to deposit and maintain a charge on its surface. It is common to find impurities in graphene samples, but the charge of a single impurity atom turned out to be insufficient. Thus it is necessary to create more intense charge centers which can serve the role of artificial nuclei for atomic collapse in graphene.

In order to successfully create a supercritical charge, the experiment conducted by Wang et al. [40] involved depositing multiple Calcium dimers on the surface of a graphene sheet. Individual dimers have subcritical charge, but in the experiment an STM tip is used to move these charge centers and assemble a cluster of supercritical charge. For the theoretical model, the previously introduced potential cutoff r_0 is modified in order to match the size of the cluster ($\approx$ 1 nm). The results of the experiment are shown in Fig. 1.3.1. The insets show the positions of the Calcium dimers as seen by STM. The curves depict the differential conductance dV/dI, i.e. LDOS, as a function of the bias voltage V for different positions of the STM tip. A change of V corresponds to changing the energy and thus allows to scan the band structure. The gap in the LDOS is caused by phonon-assisted inelastic tunneling and is not of interest here. Instead, the main feature is the appearance of a clear narrow maximum in the vicinity of the charge center in the case of 5 dimers. This is the expected evidence of an atomic collapse state in graphene with a supercritical impurity charge. In observing this, the experiment confirmed the longstanding prediction of QED.

In order to produce sufficient charge, a large number of charge centers must be placed in a small area which is difficult due to strong Coulomb repulsion. Alternatively, it has been shown that a vacancy in graphene can stably host a positive charge [117, 118]. The charge can be made supercritical, allowing the vacancy to take on the role of an artificial nucleus for atomic collapse in graphene. Its effectiveness is further enhanced because it is located directly in the plane of graphene instead of on top of the surface.

Vacancies in graphene

A vacancy is the absence of an atom at a given site in graphene. Vacancies can be created by sputtering graphene with He$^+$ ions [119]. The bombardment of ions physically removes carbon atoms from the graphene sheet. This has an effect both on the conduction band electrons as well as on the structural bonds. We shall first consider the effect of removing the π_z orbital at a lattice site, i.e. a charge-neutral vacancy.

Graphene with vacancies can still be described by the tight-binding Hamiltonian (8.2.1) where the hoppings to vacant sites are forbidden. A vacancy may consist of $N_v = N_A + N_B$ missing carbon atoms, where N_A and N_B correspond to the number of atoms removed from sublattices A and B, respectively. In the presence of electron-hole symmetry ($t' = 0$), introducing a vacancy with $N_A \neq N_B$ will break the sublattice symmetry and create a zero

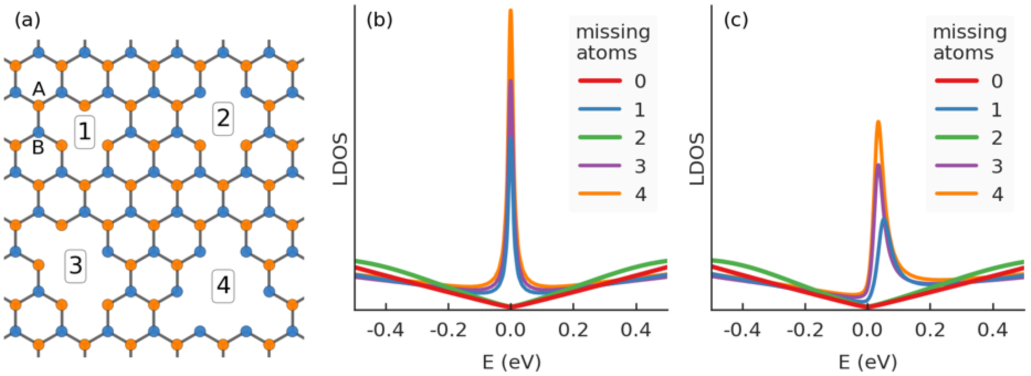


Fig. 8.3.1: (a) Illustration of vacancies with 1 to 4 missing atoms. (b) Simulated LDOS at 1 nm from the center of each type of vacancy including only the first nearest neighbor hopping. (c) The same results including the next-nearest hopping term $t' = 0.1$ eV resulting in broken electron-hole symmetry. The LDOS is computed using the tight-binding model where each case considers a single vacancy on an otherwise pristine sheet of graphene.

energy state which is quasilocalized near the vacancy. In addition, this state exists only on a single sublattice, corresponding to the one with the lowest number of removed atoms [120].

Figure 8.3.1(a) illustrates four types of vacancies labeled 1 to 4 according to the number of missing carbon atoms. The LDOS in Fig. 8.3.1(b) is shown for each vacancy as well as for pristine graphene (0 missing atoms). Vacancies with 1, 3 and 4 missing atoms exhibit a very high intensity peak in the LDOS at the Dirac point which corresponds to the quasilocalized zero energy state. The vacancy with 2 missing atoms retains a V-shaped LDOS with only a slightly increased slope compared to pristine graphene. This is because it preserves the sublattice symmetry with one missing atom from each sublattice. The other vacancies have a sublattice difference $N_B - N_A$ of 1, 1 and -2 for vacancies 1, 3 and 4, respectively.

It is possible to write down an analytical wave function of the zero energy mode introduced by a single-atom vacancy in the continuum limit [121]

$$\Psi(x, y) \simeq \frac{\mathrm{e}^{\mathrm{i}\vec{K}'\cdot\vec{r}}}{x + \mathrm{i}y} + \frac{\mathrm{e}^{\mathrm{i}\vec{K}\cdot\vec{r}}}{x - \mathrm{i}y}. \tag{8.3.2}$$

The state is quasilocalized around the vacancy in the center and its amplitude decays with distance as $1/r$. However, this vacancy state exists in this form only in the presence of electron-hole symmetry. This is also the reason why the vacancy peaks in Fig. 8.3.1(b) are sharp and symmetric. Once

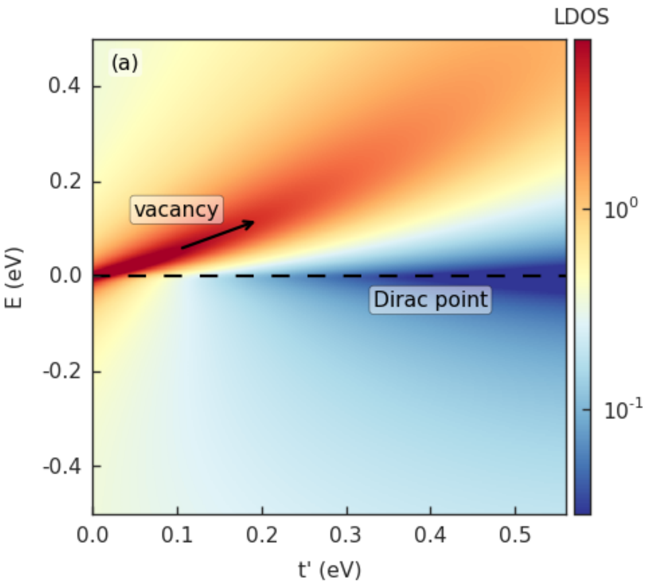


Fig. 8.3.2: Colormap of the LDOS calculated at the edge of a single-atom vacancy as a function of energy and the size of the next-nearest hopping term t'. The vacancy-induced LDOS peak moves up in energy with increasing t', as indicated by the arrow.

electron-hole symmetry is broken, the quasilocalized vacancy state turns into a resonance and the solution can no longer be obtained analytically. Instead, a numerical solution of the tight-binding Hamiltonian is required.

We compute the LDOS of a single-atom vacancy as a function of the next-nearest neighbor hopping energy t' and present the results in Fig. 8.3.2. Including the next-nearest neighbor hopping term ($t' > 0$) breaks electron-hole symmetry. The transformation from a zero energy state to a resonance means that the resulting LDOS peak will broaden and shift in energy, away from the Dirac point. Both the linewidth and the displacement are directly proportional to the degree of induced electron-hole asymmetry, i.e. the value of the next-nearest neighbor parameter. The energy shifts toward the positive energy part of the spectrum in response to the higher density of holes introduced by the t' term.

As show in Fig. 8.3.2, high values of t' can significantly broaden the vacancy peak, thus greatly diminishing its contribution to the LDOS. In the literature, the values of the next-nearest neighbor term vary quite substantially in the range between $t' = 0.02|t|$ to $t' = 0.2|t|$ [20]. Here, we shall take $t' = 0.1$ eV for all further simulations. This value is chosen as a good match with

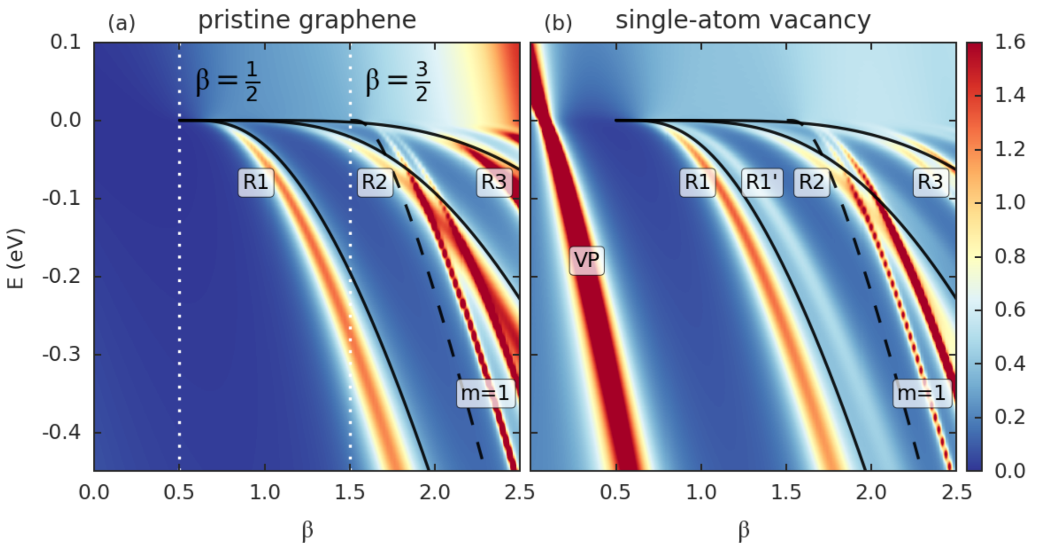


Fig. 8.3.3: Colormap of the LDOS as a function of charge β and energy E for (a) no vacancy and (b) a single-atom vacancy. The LDOS is calculated at the center of the charge. The collapse resonances are labeled R1 to R3 and the vacancy-induced peak is VP. The solid black curves indicate quasiclassical states of the collapse as given by Eq. (8.2.11) for states with angular number $m = 0$, while the dashed black curve indicates the first state with $m = 1$ (note that it originates at $\beta = 3/2$). The vertical dotted lines indicate critical values $\beta = 1/2$ and $3/2$.

experiments [118, 122]. The computed LDOS for several vacancies with this t' value is presented in Fig. 8.3.1(c).

Charging the vacancy

So far, we have only considered a charge-neutral vacancy. Apart from the effect on the conduction band, as described in the previous section, the removal of a carbon atom also affects the structural bonds. One possible outcome is that the lattice will locally undergo a bond reconstruction thereby transforming the usual hexagonal lattice to pentagon/heptagon structures or other similar local reconstructions in the vicinity of the vacancy [123]. On the other had, the disrupted bonds may remain as dangling bonds. The structure around the vacancy retains the general hexagonal shape, but the carbon atoms on the edge of the vacancy are allowed to relax. Subsequently, this lattice relaxation produces a positively charged vacancy with effective charge $Z/\kappa \approx +1|\mathrm{e}|$, where $|e|$ is the elementary charge [117].

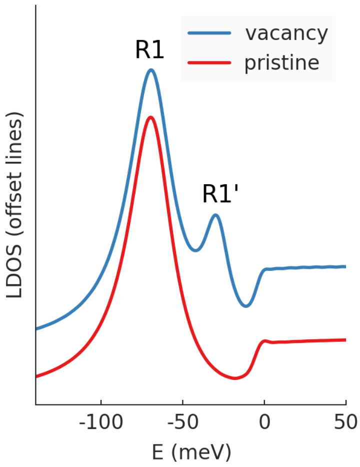


Fig. 8.3.4: LDOS for a charge of $\beta = 1.05$ in pristine graphene and the same charge hosted by a single-atom vacancy. The LDOS lines are offset for clarity.

Experimentally, bombarding a graphene sheet with an ion beam will create mostly charge-neutral vacancies. The charge can then be induced by applying voltage pulses with an STM tip [118]. It is well known that such pulses can functionalize atoms, tailor the local structure or change the charge state [124, 125]. Both positive and negative voltage pulses cause a gradual buildup of a positive charge at the vacancy. The sign of the STM voltage pulses does not matter because the pulses help the vacancy structure to relax, thus indirectly exposing the positive charge which is intrinsic to the dangling bonds of the edge atoms.

In the theoretical model, the charge comes in as a Coulomb potential given by Eq. (8.2.10). We choose the cutoff radius $r_0 = 0.5$ nm based on experimental data. The choice of this parameter will be discussed in more detail in a later section. Here, we use this potential function in combination with several types of vacancies modeled using the tight-binding approach. As a reference point we include the case of pristine graphene with the same kind of potential. This lets us see the effect of the vacancy charge, but without the structural defect itself.

We start with the reference case: pristine graphene with charge β. The LDOS is computed at the center of the potential and presented in Fig. 8.3.3(a). High-intensity peaks are clearly visible in the LDOS. They form lines descending in energy as the charge β is increased. These resonances, labeled $R1$ to $R3$, correspond to the central part of the quasiclassical spiraling orbitals of atomic collapse (see Fig. 8.2.1). Note that all of the collapse resonances only exist in the domain $\beta > \beta_c = 1/2$.

The colormap indicates the LDOS intensity as computed using the tight-

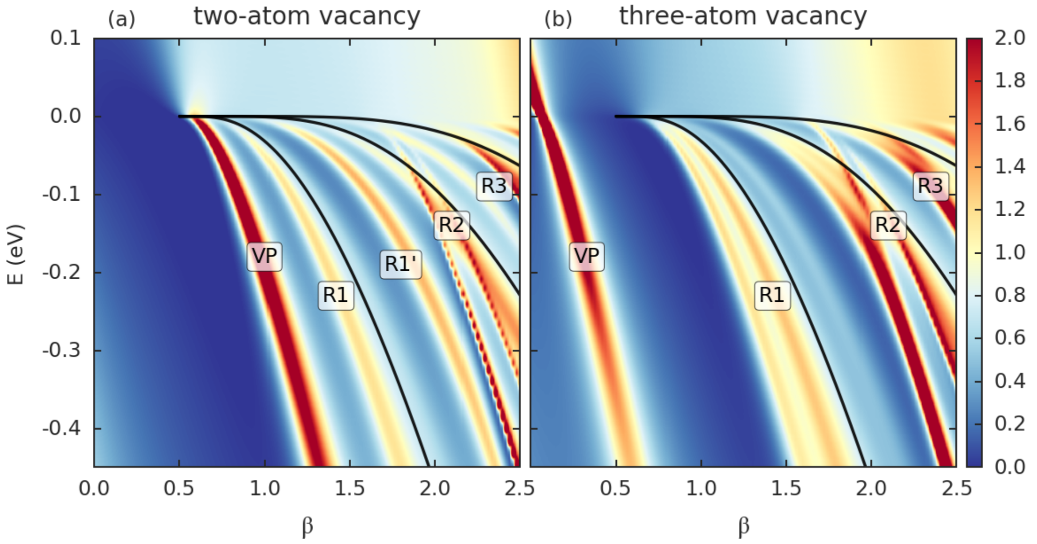


Fig. 8.3.5: Colormap of the LDOS as a function of charge β and energy E for (a) divacancy (two-atom) and (b) a three-atom vacancy. The LDOS is computed at the edge of the vacancy.

binding model. We compare these results to the quasiclassically determined energy E_{nm}, as calculated from Eq. (8.2.11) and give them by the solid and dashed black curves for $m = 0$ and $m = 1$ angular momentum states, respectively. The LDOS resonances $R1$ to $R3$ correspond to $m = 0$ states with $n = 1$ to 3. The quasiclassical calculation slightly underestimates the binding energy of the collapse states, but the general characteristic and appearance of multiple resonances is captured by both models. The differences are due to the continuum limit approximation used in the derivation of the quasiclassical energy. The continuum limit is not well defined far away from the Dirac point, thus the approximation becomes more inaccurate as the energy or charge is increased. Note that the charge does have a significant effect due to the depth of the Coulomb potential which sinks the local Dirac point to very low energy (down to -1 eV). Even though the collapse states form just below zero energy, due to the deep Coulomb well, the electron states will actually originate from the nonlinear part of the spectrum. This is responsible for the disparity between the different models.

One more feature are the thinner lines which can be seen crossing resonance $R2$ in Fig. 8.3.3(a). These narrower resonances are labeled $m = 1$ since they belong to collapse states with higher angular momentum. Resonances $R1$ to $R3$ correspond to the base set of collapse states with $m = 0$ which are analogous to atomic s orbitals. The thinner resonances belong to a new

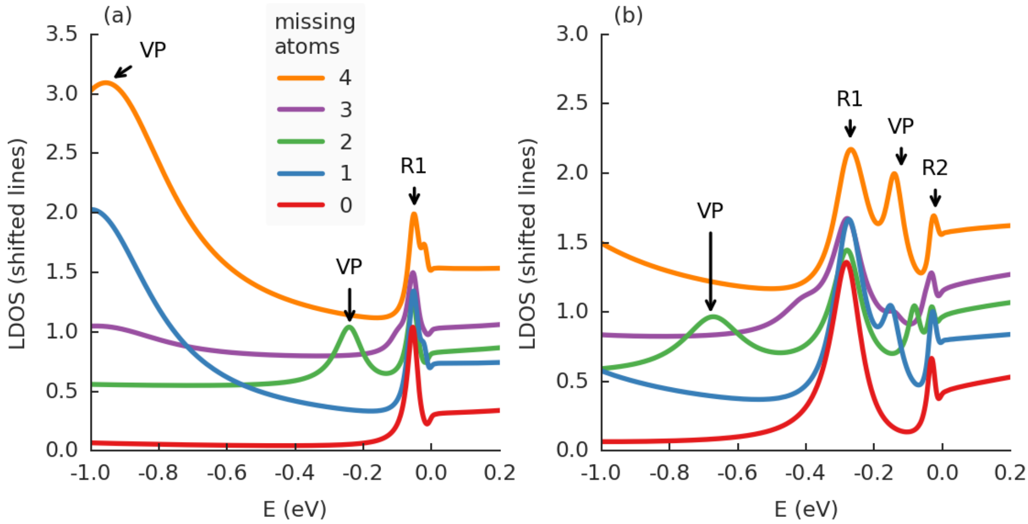


Fig. 8.3.6: LDOS for vacancies with 0 to 4 missing atoms for (a) $\beta = 1$ and (b) $\beta = 1.5$. The LDOS lines are shifted in the y-axis for clarity. Note that the positions of the atomic collapse peaks $R1$ and $R2$ are the same in all cases. The peaks originating from the vacancies are labeled VP.

set with $m = 1$ which are analogous to p orbitals. Note that states with a particular m number appear only for $\beta > \beta_m = m + 1/2$, thus there are two characteristic values $\beta = 1/2$ and $\beta = 3/2$, indicated by the dotted white lines in Fig. 8.3.3(a).

We include a single-atom vacancy and follow the evolution of the LDOS as the charge is increased according to the same potential Eq. (8.2.10) as earlier. The results are shown in Fig. 8.3.3(b). Close to $\beta = 0$ we see the neutral vacancy peak (VP) as seen previously in Fig. 8.3.1(c). Its position decreases rapidly in energy with increasing β and by the time $R1$ appears it is well out of range (for $\beta = 1$ it appears around -1 eV). As a consequence of the broken sublattice symmetry of the single-atom vacancy, collapse resonance $R1$ splits into two peaks which behave similarly as a function of β. The higher-intensity $R1$ peak is a feature of the collapse as shown in Fig. 8.3.4. The lower-intensity $R1'$ peak is unique to the single-atom vacancy. Note that the quasiclassically determined collapse energy cannot account for the presence of the vacancy, thus only a single black curve is present for the first resonance.

Of the different types of vacancies, the single-atom vacancy is of most interest. We shall just quickly mention the properties of the two and three-atom variants. The divacancy (two-atom) changes the picture in the low β region, as shown in Fig. 8.3.5(a). As mentioned previously, the divacancy lacks the LDOS peak near the Dirac point for $\beta = 0$ because it preserves sublattice symmetry. However, as seen in Fig. 8.3.1, it induces a higher LDOS intensity around ±0.4 eV instead. This is responsible for the offset peak labeled VP in Fig. 8.3.5(a). The three-atom vacancy, Fig. 8.3.5(b), restores behavior similar to the single-atom case due to a similar kind of sublattice symmetry breaking.

It's important to point out that the vacancies doesn't interfere with the collapse, i.e. they have no effect on the energy of the collapse peaks. Figure 8.3.6 shows the LDOS for vacancies with 0 to 4 missing atoms. While the various vacancy-induced peaks (labeled VP) change position depending on the number of missing atoms, the atomic collapse peaks $R1$ and $R2$ remain at the same energy in all cases, including pristine graphene (red line).

In the following sections we shall focus on the single-atom vacancy, since it is easy to realize experimentally and can be distinguished from the divacancy based on the presence of the vacancy peak near the Dirac point. The triangular structure of the single-atom variant is very distinct and easy to identify using STM topography.

Spatial extent

From the previous section, it is clear that the vacancy peak will quickly dive to very low energy as the charge is increased. When the first collapse resonance appears ($\beta > 1/2$) and gathers significant LDOS intensity ($\beta \approx 1$), the vacancy peak will be below -1 eV. This large energy distance between the vacancy and collapse peaks make them easy to distinguish. However, for

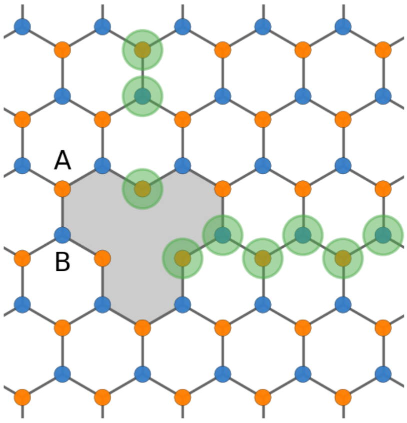


Fig. 8.3.7: Illustration of graphene with a single-atom vacancy indicated by the shaded area. An atom of sublattice B was removed. The atoms highlighted in green form a path along the zigzag (x-axis) and armchair (y-axis) directions moving away from the vacancy. Both paths alternate A and B sublattice atoms.

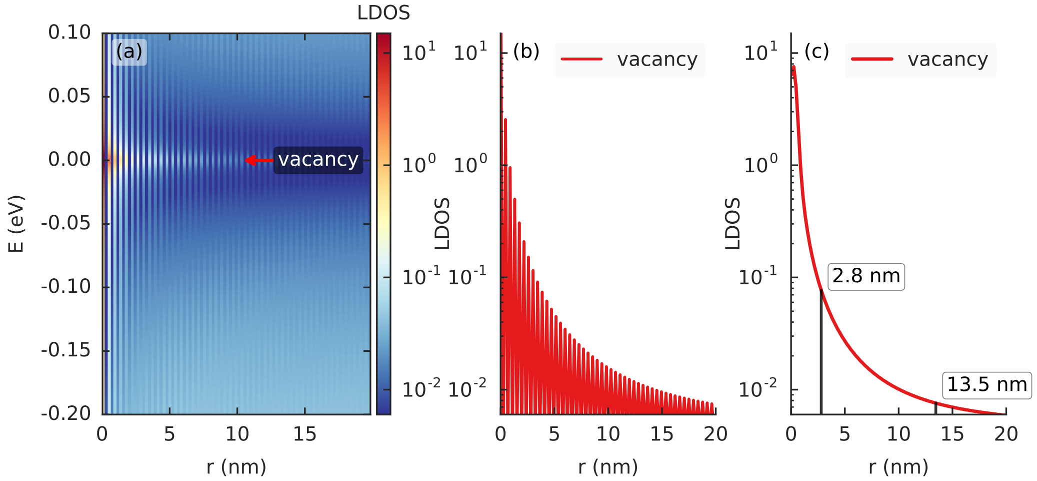


Fig. 8.3.8: (a) The LDOS at a single-atom vacancy as a function of position and energy for zero charge ($\beta = 0$). (b) Cut of the LDOS at the vacancy peak ($E = 0$) as indicated by the arrow in panel (a). (c) Convolution of the LDOS with a Gaussian with $\sigma = 0.25$ nm.

most experimental setups, the distance is actually too large to capture both peaks in the same measurement range. For this reason we must also consider distinguishing the two kinds of peaks based on their spatial properties.

A single-atom vacancy is illustrated in Fig. 8.3.7. Calculations of the LDOS as a function of position conform to the discrete positions of atoms in the tight-binding model. As indicated by the green highlighted atoms, the path of the LDOS calculation (in both zigzag and armchair directions) passes over a pattern of A-B sublattice atoms. This is important to keep in mind as we look at the spatial behavior.

We start by looking at the LDOS as a function of position and energy for zero charge $\beta = 0$ in Fig. 8.3.8(a). The vacancy peak is indicated at zero energy. Note that the LDOS oscillates rapidly as a function of position which gives the appearance of vertical lines in the figure. This is caused by the broken sublattice symmetry of the single-atom vacancy. This is more easily visible on the cut at $E = 0$ in Fig. 8.3.8(b). The vacancy is formed by removing an atom of sublattice B, thus the vacancy state is localized on atoms of sublattice A, i.e. the LDOS is zero at positions corresponding to sublattice B. This creates the high-frequency pattern. The LDOS is computed at atomic positions which follow an A-B sublattice pattern (see Fig. 8.3.7).

In order to smooth out the raw data we do a convolution with a Gaussian

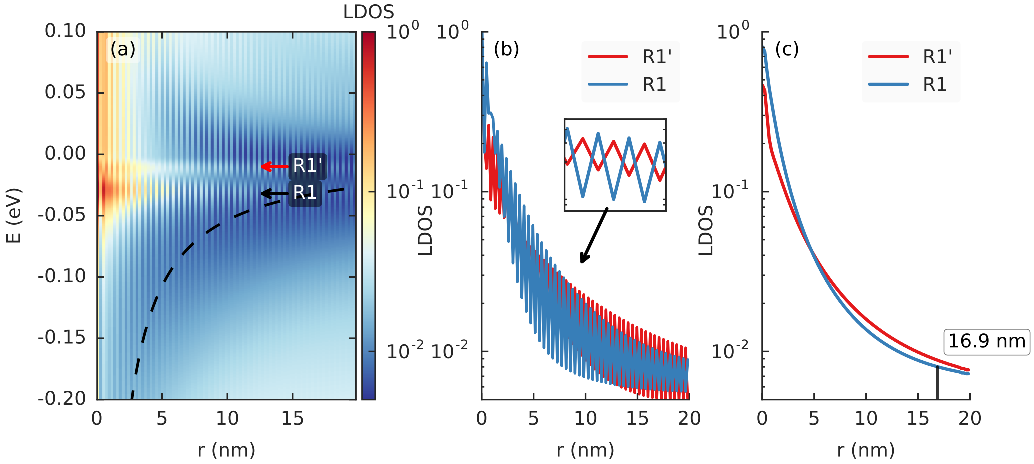


Fig. 8.3.9: Same as Fig. 8.3.8, but including a charge of $\beta = 0.9$. The dashed black line in panel (a) indicates the profile of the Coulomb potential. The inset in panel (b) shows a zoomed in view of the high-frequency pattern.

function with broadening $\sigma = 0.25$ nm in Fig. 8.3.9(c). This smooths out the large A-B oscillations. The LDOS intensity drops very quickly: at just 2.8 nm away from the center, it has lost two orders of magnitude (from 10 to 10^{-1}). Going farther out to 13.5 nm, the intensity is reduced by 3 orders of magnitude compared to the center.

Next, we include a charge of $\beta = 0.9$ along with the vacancy in Fig. 8.3.9. The main collapse peak is marked $R1$ and the satellite peak $R1'$. The vacancy peak has dropped to very low energy (≈ -1 eV), far below the limits of the figure. Panel (b) shows cuts of the LDOS at the collapse peaks. Both appear as high-frequency patterns similar to the vacancy peak from Fig. 8.3.8(b). However, in this case both sublattices have a finite contribution to the collapse LDOS, while the vacancy LDOS was zero for sublattice B. The LDOS patterns of $R1$ and $R1'$ are inverted with regard to the sublattices: a maximum on $R1$ is accompanied by a minimum on $R1'$ and vice versa. This can be seen more clearly in the inset of Fig. 8.3.9(b) which shows a zoomed in view.

Once again, a convolution of the LDOS with a Gaussian results in smooth lines presented in Fig. 8.3.9(c). The LDOS of the two collapse states drops by two orders of magnitude at 15.8 nm away from the center. This is in contrast to the vacancy peak where this happens at just 2.8 nm. Note that $R1$ has higher intensity than $R1'$ close to the center, but this is reversed on the outside. State $R1$ has a lower minimum radius and therefore lower

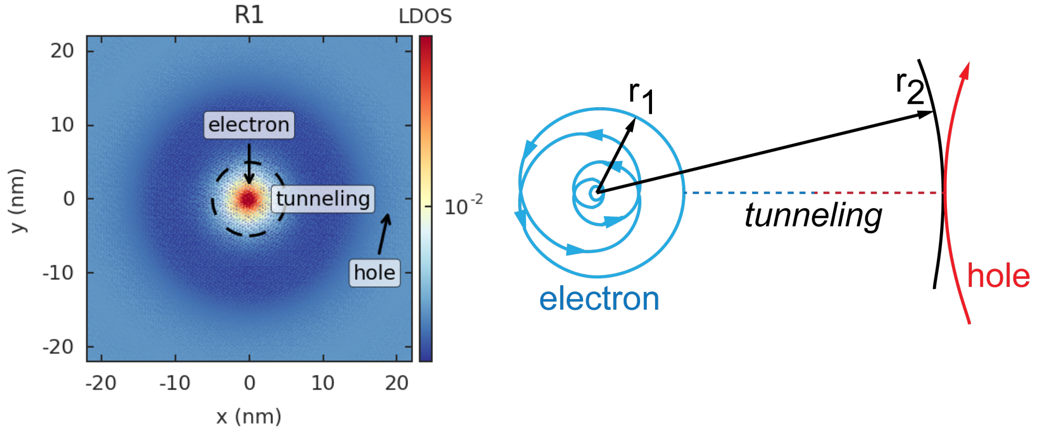


Fig. 8.3.10: Spatial map of the LDOS for collapse resonance $R1$ and an illustration of the corresponding quasiclassical electron and hole trajectories. The LDOS is calculated for $\beta = 1.25$ and $E = -0.15$ eV. The dashed circle indicates the radius of the Coulomb potential at this energy.

energy than $R1'$. This is explored further in the next section.

Figure 8.3.10 shows the spatial colormap of the LDOS for the $R1$ collapse resonance. Good agreement can be seen by comparing the spatial map with the quasiclassical orbits. The high LDOS intensity central region (red) corresponds to quasi-bound electron states inside the Coulomb potential (dashed line). Hole states are found on the outside, but there is a distinct low intensity ring (blue) which is the tunneling barrier, followed by a slightly higher intensity region (light blue) which is the hole continuum. The spatial map

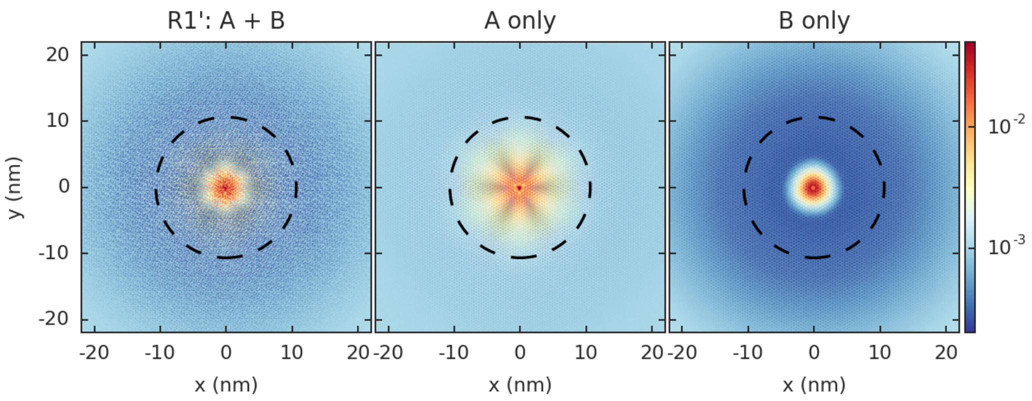


Fig. 8.3.11: Spatial LDOS for resonance $R1'$ ($\beta = 1.25$, $E = -0.07$ eV) and the sublattice components. The dashed circle is a cut of the Coulomb potential.

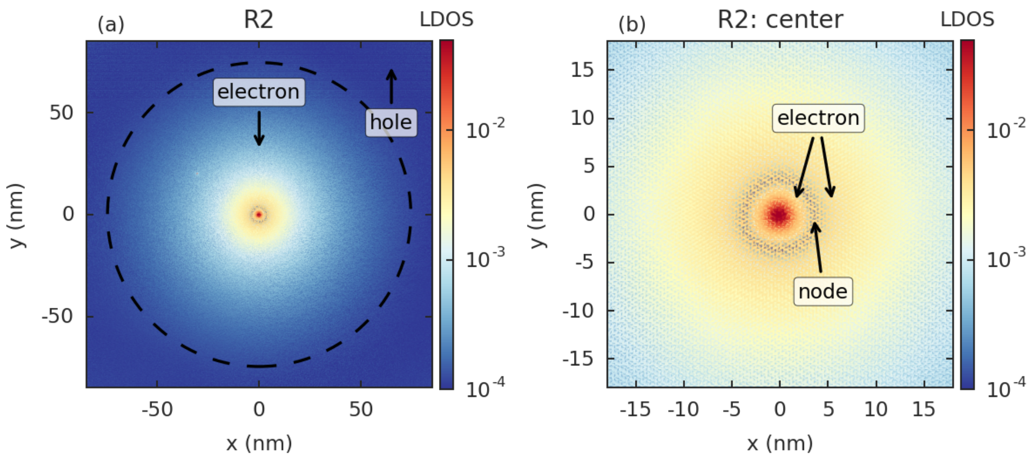


Fig. 8.3.12: Spatial LDOS for resonance $R2$ ($\beta = 1.25$, $E = -0.01$ eV) with (a) a wide view and (b) a zoomed in view of the center. The dashed circle is a cut of the Coulomb potential.

presents with what looks like grain noise. This is a consequence of the broken sublattice symmetry, as shown previously in Figs. 8.3.8 and 8.3.9.

Next, we take a look at the LDOS spatial map of the $R1'$ resonance in Fig. 8.3.11. The situation here is similar to $R1$, but the sublattice-asymmetry-induced grain noise is more apparent. When we separate the individual sublattice components, we see the influence of the vacancy. Since the vacancy is formed by removing a sublattice B atom, electrons are localized on A atoms

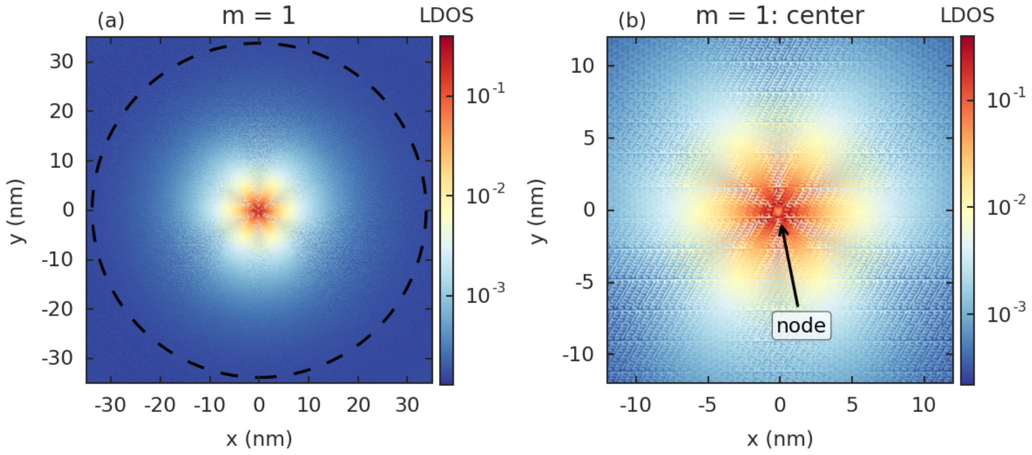


Fig. 8.3.13: The higher angular momentum collapse resonance $m = 1$ ($\beta = 1.7$, $E = -0.03$ eV) with (a) a wide view and (b) a zoomed in view of the center.

around the vacancy and spread out in the zigzag directions of graphene, which accounts for the six-fold symmetry of the A-only LDOS contribution. The atoms on sublattice B are not affected, thus the B-only LDOS contribution is only affected by the circular Coulomb potential and we see separate inner (electron) and outer (hole) regions, just like the $R1$ resonance.

The $R2$ resonance is presented in Fig. 8.3.12. This state is located just below the Dirac point ($E = -0.01$ eV) near the top of the Coulomb potential, thus it has a large radius so the outer hole region is outside the limits of the figure. We zoom in closer to the center of $R2$ in Fig. 8.3.12(b). Note that there is a lower intensity node which splits the electron state into two regions. This is exactly analogous to the atomic $2s$ state which is the stable counterpart of the $R2$ collapse resonance.

All of the collapse resonances that we have looked at so far have been at the lowest angular momentum $m = 0$ which is analogous to atomic s states. In Fig. 8.3.13 we present the spatial map of a higher angular momentum ($m = 1$) collapse resonance. Notice that this state has a low LDOS intensity node at the center and a lobe structure reminiscent of atomic p orbitals which are its stable counterparts. A significant difference compared to real atoms is that all states in graphene are confined to a 2D plane, thus the usual p_x, p_y and p_z orbitals are not directly transferable, but the collapse resonance does exhibit 6-fold symmetry. Unlike the lower angular momentum $R1'$ resonance, the spatial map of the $m = 1$ collapse resonance is not affected by the presence of the vacancy - an identical spatial structure is present in pristine graphene. The only visible effect of the vacancy is the slight sublattice-asymmetry-induced noise seen in Fig. 8.3.13(b).

Influence of the potential cutoff

The form of the truncated potential from Eq. (8.2.10) takes the finite size of the charge into account via the cutoff parameter r_0. In all the previous calculations we set the value to $r_0 = 0.5$ nm. In order to investigate the effect of this parameter we varied it as shown in Fig. 8.3.14(a). Reducing the length of r_0 increases the strength of the potential and in the limit $r_0 \rightarrow 0$ it corresponds to a point charge. As we move closer to this limit, the energy levels of $R1$ and $R1'$ diverge relative to each other.

As mentioned in the previous section, $R1$ and $R1'$ are mostly made up of opposite sublattice states. As shown in Fig. 8.3.14(b), sublattice A is closer to the center with a radius of $a_{cc} = 0.142$ nm (carbon-carbon distance), while

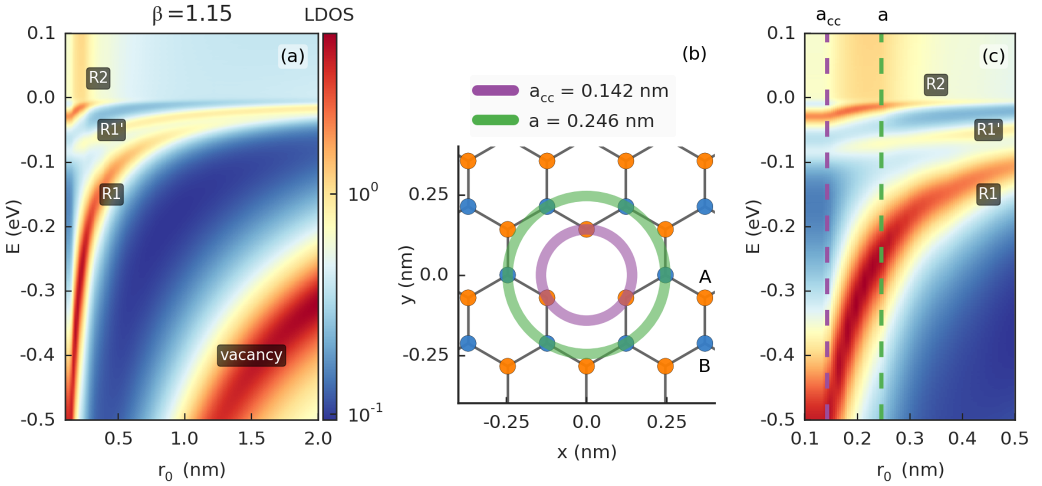


Fig. 8.3.14: (a) LDOS as function of the r_0 parameter for $\beta = 1.15$. (b) Illustration of the model system around the vacancy. The circles indicate minimum radii of sublattices A and B. (c) Zoomed in view of panel (a) close to the center. The vertical lines correspond to the radii from panel (b).

the minimum radius of sublattice B is farther away at $a = 0.246$ nm (lattice constant). This defines the minimum radii of the $R1$ and $R1'$ resonances.

Figure 8.3.14(c) shows a zoomed in view at small r_0 and the vertical lines correspond to the radii from Fig. 8.3.14(b). Resonance $R1'$ loses most of its intensity in the $a_{cc} < r_0 < a$ region, while $R1$ does the opposite and intensifies until the saturation point at a_{cc}. In the region $r_0 < a_{cc}$, the potential radius becomes smaller than the vacancy and is therefore non-physical. Notice that there is also a small buildup of LDOS above the Dirac point in the $a_{cc} < r_0 < a$ region. This is the positive resonance associated with $R2$ (see Fig. 8.2.2).

The r_0 cutoff is mostly a fitting parameter for the simulation which should roughly correspond to the finite size of the charge. In Ref. [40], where the experiment was conducted with a cluster of charges, this parameter was fitted to about 1 nm which was appropriate to the type of charge used there. Additionally, they did not see large variations when r_0 was changed up to ± 0.5 nm. However, in the case of the vacancy, the r_0 parameter becomes much more important because of its influence on the satellite peak $R1'$. The LDOS intensity and energy difference between the $R1$ and $R1'$ peaks is inversely proportional to r_0, as seen in Fig. 8.3.14(c). This difference can be used to estimate the value of r_0.

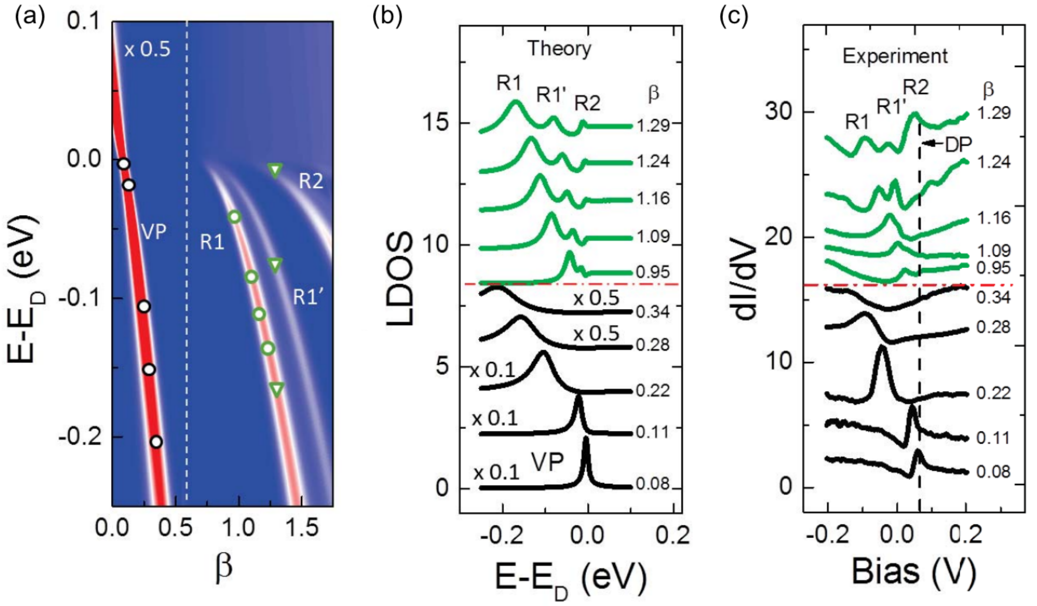


Fig. 8.3.15: (a) Simulated map of the evolution of the spectra with β. The data presented here is the same as Fig. 8.3.3(b). The intensity scale in the vacancy peak (VP) regime is divided by 2 to facilitate the comparison with the atomic collapse (AC) regime. The symbols represent the energies of the VP and AC peaks from panel C. (b) Simulated spectra for the β values in (c). The curves are vertically offset for clarity. (c) Evolution of STS with charge (increasing from bottom to top). Each curve is marked with its corresponding β value. The horizontal dashed line separates between spectra in the subcritical and supercritical regimes. The vertical dashed line represents the bulk DP measured far from any vacancy. From Mao et al. [118]

Comparison with experiment [*]

This work was conducted in collaboration with the experimental group of Eva Andrei from Rutgers University and the results were published in Ref. [118]. Experimentally, the vacancy was created using He-ion bombardment. A single-atom vacancy was identified in the sample based on its distinct triangular symmetry which shows up on topographical images of graphene. In order to create charged artificial nuclei, the vacancy must not be passivated by trapped ions. Strong voltage pulses were applied using the STM tip which caused the vacancy structure to relax, revealing the charge on the dangling

[*] The results of this section were published as:
J. Mao, Y. Jiang, D. Moldovan, G. Li, K. Watanabe, T. Taniguchi, M.R. Masir, F.M. Peeters, and E.Y. Andrei, Nat. Phys. 12, 545 (2016).

σ bonds. Repeating the pulses increased the charge in discrete intervals. Once the vacancy was saturated, the charge remained constant as long as the experiment was kept cold. The gradual charge buildup was monitored with scanning tunneling spectroscopy and Landau level spectroscopy.

The simulated evolution of the LDOS with β is presented in Fig. 8.3.15(a), as calculated from the tight-binding model up to the next-nearest neighbor hopping. The theoretical results were compared to the experimentally observed curves in Fig. 8.3.15(c). The STM tip pulses increase the vacancy charge in discrete intervals, so only certain β values are available for comparison with theory. Cuts of the simulated LDOS are taken and compared with the experiment. The experimental values of the β parameter are determined by matching the energy positions of the vacancy peak (VP) and collapse resonances ($R1$, $R1'$ and $R2$).

The theoretical and experimental data show excellent agreement with regard to the evolution of the LDOS as a function of the charge. At low β the spectrum consists of a single peak which results from the uncharged vacancy. With increasing β the vacancy peak (VP) broadens and its energy becomes more negative, all the while remaining tightly localized on the vacancy site. Upon exceeding the critical value, $\beta > 0.5$, a new branch, labeled $R1$, emerges below the Dirac point. This is the counterpart of the $1s$ collapse state in atoms. The $R1$ collapse resonance is clearly distinguishable from the VP by its significantly larger spatial extent. With increasing β, $R1$ develops a satellite, $R1'$, which tracks its evolution with β. While $R1$ is a universal feature of AC states, $R1'$ is due to the locally broken sublattice symmetry and is peculiar to the supercritically charged single-atom vacancy. As β further increases more branches emerge, starting with $R2$ which is the equivalent of the $2s$ AC state.

8.4 Conclusions

Atomic collapse is a long sought-after phenomenon in quantum electrodynamics which is not observable with real atomic nuclei because of the extremely large charge requirements. Graphene offers the opportunity to observe the same physics at a much smaller charge and energy scale. As the charge is increased, the interaction undergoes a transition into a supercritical regime where electrons are trapped in a sequence of quasi-bound states which resemble an artificial atom. In order to reach the supercritical regime an artificial nucleus with sufficient charge must be constructed. This can be done by assembling a cluster of charged impurities. Alternatively a single-atom vacancy can stably host a local charge that is built up by applying voltage pulses with the tip of a scanning tunneling microscope. The quasi-bound electron states are detected as a strong enhancement of the local density of states within a disc centered on the charge.

We modeled the effect of a supercritically charged vacancy in graphene using the tight-binding model. Special consideration was required to accurately model the behavior of the vacancy by taking into account the second nearest-neighbor hopping term. The results reveal the expected collapse resonances, but also a new branch of collapse states related to the locally broken sublattice symmetry of a single-atom vacancy. The results were compared with experimental observations and good agreement was found.

CHAPTER 9

Supercritical charge controlled by the STM-gating effect

Apart from just measuring sample properties, an STM tip may also be used as a secondary gate which offers an additional knob for experiments. We study this STM-gating effect theoretically for a graphene sample. The tip-induced electric potential forms a circular pn-junction and we calculate its profile for different tip geometries as well as different values of the tip-sample work function difference (which may be changed by coating the tip with various materials). We follow the evolution of the circular junction with back-gate voltage and find that it makes a transition from a wide-area potential, which exhibits electron optics effects such as *whispering-gallery modes*, to a point-charge-like potential which gives rise to the relativistic *atomic collapse* phenomena. Both regimes result in electron quasi-confinement in graphene.

9.1 Introduction

Scanning tunneling microscopy (STM) and spectroscopy (STS) are two important techniques which can be used to probe the topography and electronic properties of materials. They both employ the same hardware: an atomically sharp metallic tip which is brought within a nanometer of the surface of a conducting sample. Due to the close range, the tip can measure tunnelling current induced by a bias difference with the sample. In STM, the tip scans the surface while a fast piezoelectric motor controls the tip height,

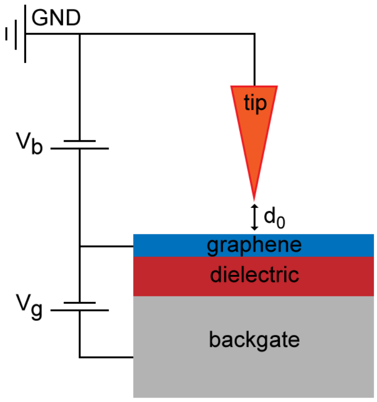


Fig. 9.1.1: Illustration of an STM/STS experimental setup. The graphene sample sits on top of an isolating layer. The voltage V_g is applied between the graphene and the conducting backgate layer. The tip hovers over the sample at minimal distance d_0 and a bias voltage V_b is applied to produce a tunneling current between the tip and graphene.

thus keeping the current constant. The spatially varying height represents a topographic map of the material with possible sub-atomic resolution. On the other hand, in STS, the tip position is kept constant while the bias voltage varies. The result of the measurement is effectively the local density of states (LDOS) as a function of energy, usually with very good resolution ≈ 0.1 meV [126].

An ideal measurement would return information from the system without interfering with it, although practically such effects cannot be avoided. The voltage difference between the tip and sample can slightly disturb the area below the tip. Effectively, a potential well is formed in the sample which leads to a local band bending effect [127]. While not always desirable, the effect can also be quite useful, opening the door for some interesting new experiments based on local gating [128, 129]. In an STM/STS setup (see Fig. 9.1.1), the planar back gate controls the overall bulk carrier density in the sample. The tip influences the carrier density only locally underneath itself and thus behaves like a small movable circular gate. This dual-gate configuration permits the creation of pn-junctions that can be tuned using two independent knobs: the backgate voltage V_g and tip-sample bias V_b.

In this chapter we investigate the effect of STM-tip-induced electric potentials in graphene. We find electronic states quasi-confined in circular pn-junctions. The radius of such a junction varies depending on the geometry of the STM tip (sharp or blunt), the work function difference between the tip and sample, as well as backgate voltage. As the dimensions of the induced quantum well shrink, the nature of this confinement changes [130, 131]. For wide circular pn-junctions, the effects can be understood in terms of electron optics as ray reflections and lensing. When the size is reduced enough to resemble a point-charge, the effect is better described via the atomic collapse phenomena.

9.2 Theoretical model

The tunnelling current between an STM tip and the sample is given by [132],

$$\begin{aligned} I = \frac{4\pi e}{\hbar} \int_{-\infty}^{\infty} \Big(f(E_F - eV_b + E) - f(E_F + E) \Big) |T|^2 \\ \cdot \rho_S(E_F - eV_b + E) \rho_T(E_F + E) dE, \end{aligned} \tag{9.2.1}$$

where $f(x)$ is the Fermi-Dirac distribution, E_F the Fermi energy, V_b the sample bias voltage, ρ_S and ρ_T are the density of states of the sample and tip, respectively. The tunnelling coefficient T depends strongly on the tip-sample distance which is kept constant in STS measurements. In addition, it is usually assumed that the tip density of states ρ_T is constant for a given range of bias voltage V_b. For experiments conducted at sufficiently low temperature, the Fermi-Dirac distribution can be replaced by a step function. Taking all this into consideration, the last equation can be significantly simplified,

$$I(\vec{r}, V) \propto \int_{-\infty}^{eV_b} \rho_S(\vec{r}, E) dE. \tag{9.2.2}$$

Therefore, the local density of states is proportional to the differential conductance measured by the STM tip,

$$\rho_S(\vec{r}, E = eV_b) \propto \frac{dI}{dV_b}(\vec{r}, V_b), \tag{9.2.3}$$

where the bias voltage V_b determines the excitation energy E of the electron in the sample. For positive bias voltages, electrons tunnel from the tip and populate empty states in the sample. For negative voltages, occupied sample states tunnel out to the tip.

The STM tip simultaneously measures the LDOS and influences the carrier density in the sample. Because the tip is radially symmetric, the problem simplifies to one dimension in space determined by position r. Based on the Thomas-Fermi model, the carrier density is [110, 133],

$$n(r) = -\operatorname{sgn}[V(r)] \frac{V(r)^2}{\pi \hbar^2 v_F^2}, \tag{9.2.4}$$

where $V(r)$ is the electric potential. In our case $V(r)$ is induced by the STM tip. The total carrier density $n(r) = n_\infty + \Delta n(r)$ consists of the bulk density n_∞ (as present far away from the STM tip) and the local carrier density $\Delta n(r)$ which is induced by the tip.

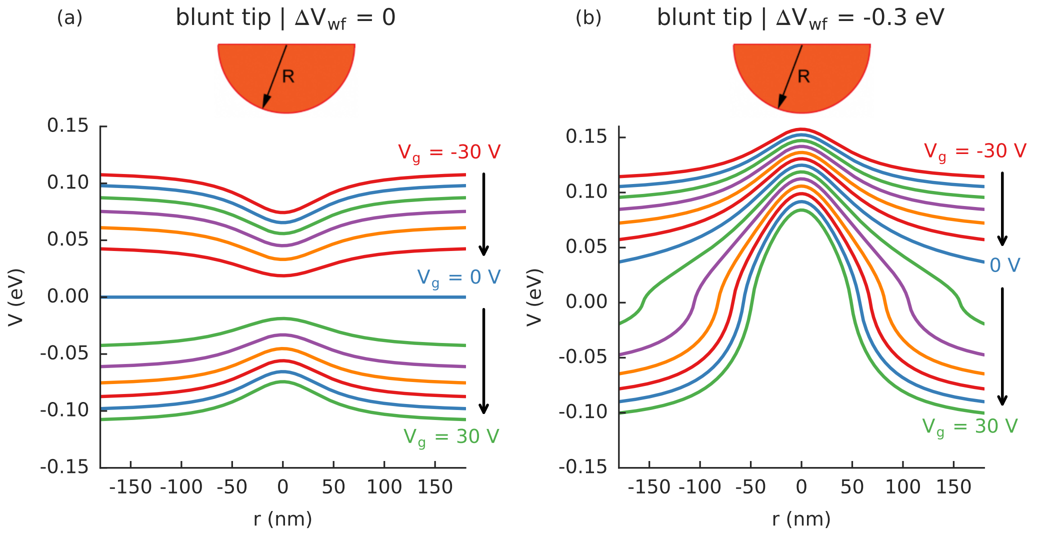


Fig. 9.2.1: The evolution of the tip-induced potential with backgate voltage V_g for a blunt STM tip with $R = 500$ nm. The tip-sample work function difference is (a) $\Delta V_{wf} = 0$ and (b) $\Delta V_{wf} = -0.3$ eV. The tip bias voltage is kept constant $V_b = 0$, while the backgate voltage V_g varies from -30 V to 30 V in 5 V increments.

The bulk carrier density is determined from the parallel capacitor model as $n_\infty = ke(V_g - V_{g0})/(4\pi e^2 d_g)$, where k is the dielectric constant, d_g is the thickness of the substrate and V_{g0} accounts for residual doping in the substrate. In this work we take the value $k/d_g = 5\ \mu\text{m}^{-1}$ and we set $V_{g0} = 0$ so that the charge neutrality point coincides with $V_g = 0$.

The STM tip acts like a secondary gate in the experiment. Because of the shape of the tip, the distance between the sample and a point on the tip is described by a position dependent function $d_T(r)$. We shall consider two geometries. A blunt tip described by $d_T(r) = d_0 + r^2/(2R)$, where d_0 is the minimal distance from the tip and the sample, and R is a large radius of the spherically shaped tip (see the illustration in Fig. 9.2.1). The other geometry describes a sharp STM tip as a cone with $d_T(r) = d_0 + r\cot(\theta/2)$, where θ is the angle of the cone (see the illustration in Fig. 9.2.2). Based on the capacitor model, a small segment of the tip at position r will induce a small change directly below it,

$$\Delta n(r) = -\frac{eV_b - \Delta V_{WF} - V(r)}{4\pi e^2 d_T(r)}, \qquad (9.2.5)$$

where V_b is the tip bias voltage and ΔV_{WF} is the work function difference between the tip and graphene.

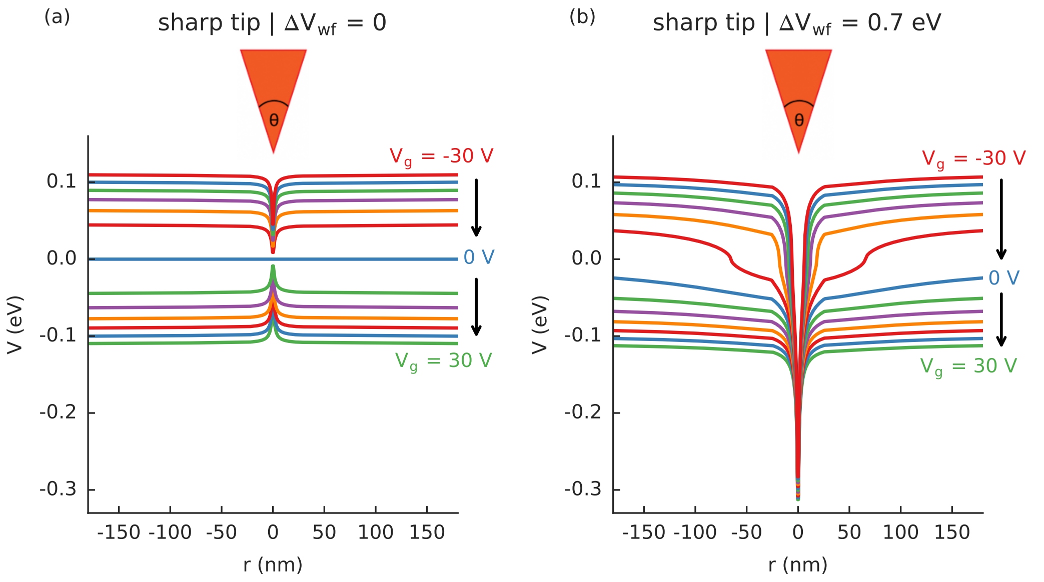


Fig. 9.2.2: Like Fig. 9.2.1, but for a sharp STM tip with $\theta = 80°$. The tip-sample work function difference is (a) $\Delta V_{wf} = 0$ and (b) $\Delta V_{wf} = 0.7$ eV. The tip bias voltage is kept constant $V_b = 0$, while the backgate voltage V_g varies from -30 V to 30 V in 5 V increments.

We can now match the tip-induced carrier density $\Delta n(r)$ as given by Eqs. (9.2.4) and (9.2.5),

$$-\operatorname{sgn}[V(r)]\frac{V(r)^2}{\pi\hbar^2 v_F^2} - n_\infty = -\frac{eV_b - \Delta V_{WF} - V(r)}{4\pi e^2 d_T(r)}. \qquad (9.2.6)$$

From here we obtain the effective electric potential induced by the STM tip for given values of V_b and V_g,

$$V(r, V_g, V_b) = \frac{-a(r) + \sqrt{a(r)^2 + 4|b(r, V_g, V_b)|}}{2\operatorname{sgn}[b(r, V_g, V_b)]}, \qquad (9.2.7)$$

where,

$$a(r) = \frac{\hbar^2 v_F^2}{4e^2 d_T(r)}, \qquad (9.2.8)$$

$$b(r, V_g, V_b) = ea(r)\left((V_b - \Delta V_{wf}) - \kappa\frac{d_T(r)}{d_g}(V_g - V_{g0})\right). \qquad (9.2.9)$$

A sample of the resulting STM potential is presented in Fig. 9.2.1 for the blunt tip shape. When the tip-sample work function difference ΔV_{wf} is equal

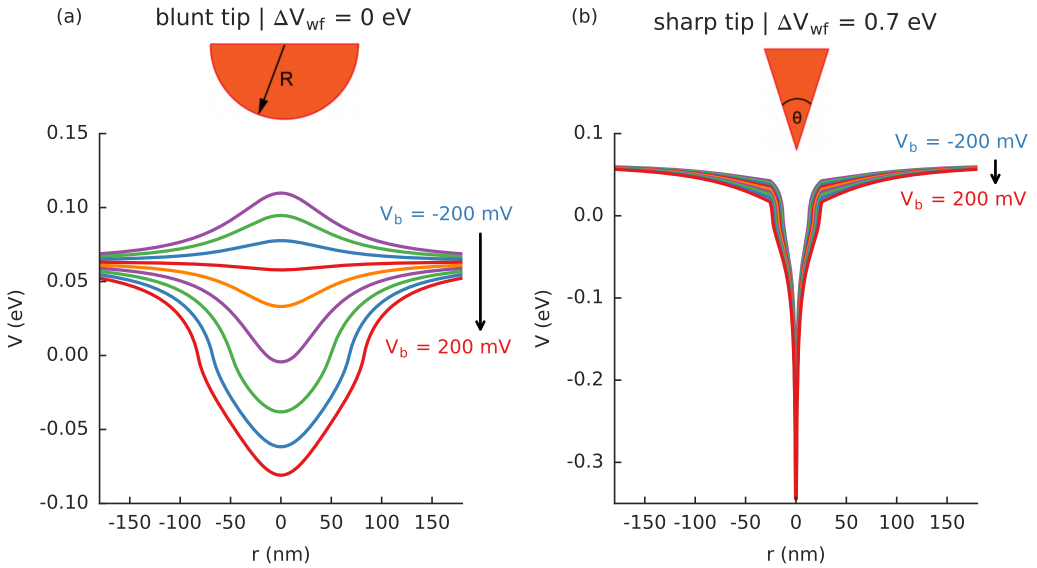


Fig. 9.2.3: The evolution of the tip-induced potential with bias voltage V_b for (a) a blunt tip with $R = 500$ nm and $\Delta V_{wf} = 0$, and (b) a sharp tip with $\theta = 80°$ and $\Delta V_{wf} = 0.7$ eV. The backgate voltage is kept constant $V_g = -10$ V, while the tip bias voltage V_b varies from -200 mV to 200 mV in 50 mV increments.

to zero, the profile of the tip-induced potential is symmetric around the charge neutrality point ($V_g = 0$) with a potential well forming in the p-doped regime ($V_g < 0$) and a barrier in the n-doped regime ($V_g > 0$). The work function can be changed by exposing the tip to other materials, e.g. in Ref. [134] deuterium absorption was used to realize $\Delta V_{wf} = -0.3$. As seen in Fig. 9.2.1(b), with this ΔV_{wf}, the tip generates a strong potential barrier on both sides of the charge neutrality point.

Next, we take a look at the potential generate by a sharp tip, as presented in Fig. 9.2.2. For $\Delta V_{wf} = 0$, the evolution of the potential with V_g is symmetric with regard to the charge neutrality point. In this regard, the blunt and sharp tips are very similar (compare Figs. 9.2.1(a) and 9.2.2(a)). The only significant difference is the area of effect: the sharp tip generates a narrow potential profile, as expected. This time we set a positive tip-sample work function difference, $\Delta V_{wf} = 0.7$ eV, which can be achieved by coating the tip with gold [135]. As seen in Fig. 9.2.2(b), this creates a point-charge-like potential which is deep and narrow. Notice that the transition from the p- to n-doped regime is accompanied by a sudden broadening of the top part of the potential well – see the central red line in Fig. 9.2.2(b), just above the blue $V_g = 0$ line. This transition takes place near the charge neutrality point and it is of particular interest (to be discussed in more detail in a later section).

So far we have looked at the evolution of the potential with backgate voltage V_g while keeping the tip bias V_b constant. The bias voltage determines the excitation energy of the electrons for the measurement, but it also affects the induced potential. As shown in Fig. 9.2.3(a) for a neutral blunt tip, varying V_b can completely change the potential profile. This effect is significantly diminished in the case of a large work function and sharp tip, as see in Fig. 9.2.3(b). However, for good model accuracy, the influence of V_b must be taken into account.

We model the effect of the STM tip on the sample by including the previously derived potential function $V(r, V_g, V_b)$ from Eq. (9.2.7) in the tight-binding Hamiltonian of graphene $H(V_g, V_b)$ within the first-nearest neighbor approximation,

$$H_0 = t \sum_{\langle i,j \rangle} \left(a_i^\dagger b_j + H.c. \right), \tag{9.2.10}$$

$$H_{STM}(V_g, V_b) = \sum_i V(R_i^A, V_g, V_b) a_i^\dagger a_i + \sum_i V(R_i^B, V_g, V_b) b_i^\dagger b_i, \tag{9.2.11}$$

$$H(V_g, V_b) = H_0 + H_{STM}(V_g, V_b), \tag{9.2.12}$$

where operators $a_i^\dagger(a_i)$ and $b_i^\dagger(b_i)$ create (annihilate) an electron on site R_i of sublattice A and B, respectively. Here $t = -2.8$ eV is the nearest neighbor hopping energy. We compute the LDOS numerically using the kernel polynomial method. Note that the local density of states $\rho(r, E, V_g, V_b)$ becomes a function of 4 parameters. Since we wish to compute the same LDOS as would be measured by an STM tip, we match its position at the center of our radially symmetric system, i.e. $r = 0$. As per Eq. (9.2.3), the measured energy matches the bias voltage $E = eV_b$. Thus, the simulated LDOS $\rho(V_g, V_b)$ is directly related only to the experimental knobs V_b and V_b. We may still free the spatial position r to compute a spatial map of the LDOS, although that would not be possible in experiments with a single STM tip.

9.3 Whispering gallery modes

The linear spectrum of electrons in graphene resembles that of light. Adding to the similarities, it has been shown that the ballistic regime persists even at room temperature over distances on the order of a micrometer [21, 22].

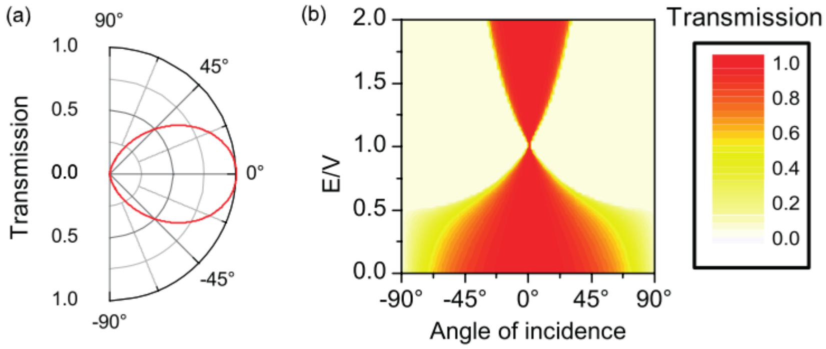


Fig. 9.3.1: (a) Transmission probability through a monolayer graphene pn-junction with potential V for electrons with energy $E = V/2$. (b) Contour plot of the angle and energy dependence of the transmission. Adapted from [146].

This makes ray optics a particularly interesting and accurate description of electrons in graphene [136–138]. It gives rise to effects such as electron focusing [139, 140] and waveguides [141]. It also enables the creation of electronic Fabry-Perot interferometers [55, 142]. Whispering-gallery mode (WGM) resonators represent another class of very high quality optical devices [143]. The name originates from the acoustic resonances found in large structures like cathedrals. It was shown that these kinds of resonators can be realized in graphene using circular pn-junctions [134, 144, 145]. Moreover, realizations in the form of STM-tip-induced pn-junctions enable the cavity radius of these WGM resonators to be tuned using the backgate voltage.

A WGM resonator cavity in graphene is defined by a circular pn-junction where the reflection of electrons is governed by Klein scattering. At first glance, it may seem strange that Klein scattering is the main mechanism behind this phenomena. After all, it's best known for perfect transmission in graphene. However, this only applies to incidence normal to a barrier. As seen in Fig. 9.3.1, at oblique angles, Klein scattering at a pn-junction actually gives rise to perfect reflection and this is exactly the mechanism of confinement of whispering-gallery modes: the ray skips around the perimeter of the circular pn-junction being reflected at oblique angles. When the length of this closed loop is equal to an integer multiple of the wavelength, a WGM resonance is formed which partially confines the electron. Because of the radial symmetry, the wavefunction can be decomposed in polar coordinates as $\Psi_{n,m}(r, \phi) = \psi_n(r)\mathrm{e}^{im\phi}$, where n and m are integers which denote the principal and angular quantum number, respectively. Higher angular momentum modes m are reflected at more oblique angles and therefore feature better confinement compared to lower m modes.

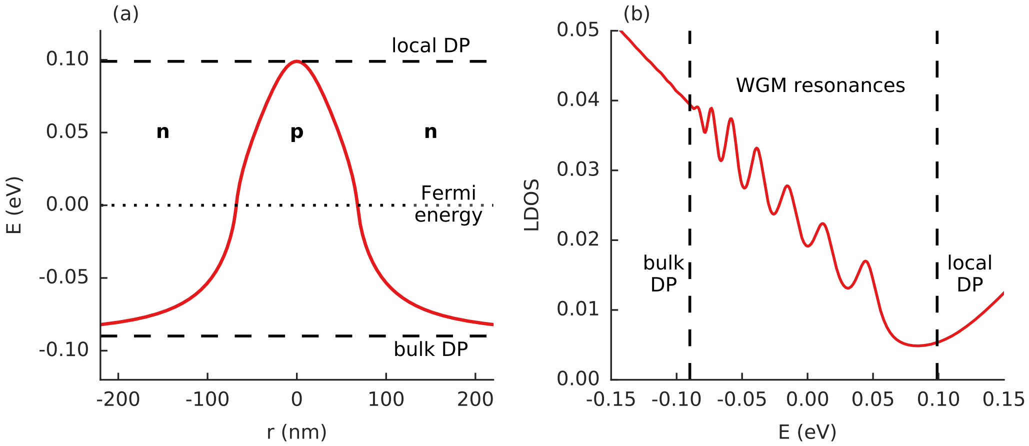


Fig. 9.3.2: (a) A circular pn-junction potential induced by a blunt STM tip with the same parameters as presented in Fig. 9.2.1(b) but with the backgate voltage fixed at $V_g = 20$ V. The energy positions of the local and bulk Dirac points (DP) and the Fermi energy are indicated. (b) The corresponding LDOS with WGM resonances.

Whispering-gallery modes require cavity sizes much larger than the wavelength in order for the electrons to behave like classical light rays. For this reason, we model a blunt tip with a large radius $R = 500$ nm. The work function difference is set to $\Delta V_{wf} = -0.3$ eV as is the case for some experiments [134]. The resulting potential at a fixed backgate voltage $V_g = 20$ V is presented in Fig. 9.3.2(a). The corresponding LDOS is computed in Fig. 9.3.2(b). The peaks in the LDOS can be attributed to WGM resonances as we will see shortly. The local and bulk Dirac points (DP) are indicated in the figures. The local DP is defined as the potential energy at the position of the STM tip ($r = 0$), while the bulk DP reflects the behavior far away from the tip, where the material is not affected by tip-gating. The bulk DP depends only on the backgate voltage V_g.

Figure 9.3.3 shows the spatial map of the wavefunction $|\Psi|$ for a strongly confined WGM resonance. The given state is calculated for $E = -50$ meV, which corresponds to a circular pn-junction with radius $r = 90$ nm. Because of the large angular momentum number $m = 15$ presented here, the wavefunction contains a large number of resonances distributed around the circumference of the pn-junction. This is more clearly visible for the individual sublattice components $|\Psi_A|$ and $|\Psi_B|$ which more closely resemble the traditional picture of WGM. These resonances form due to electrons reflecting at the circular pn-junction. Higher angular momentum modes are

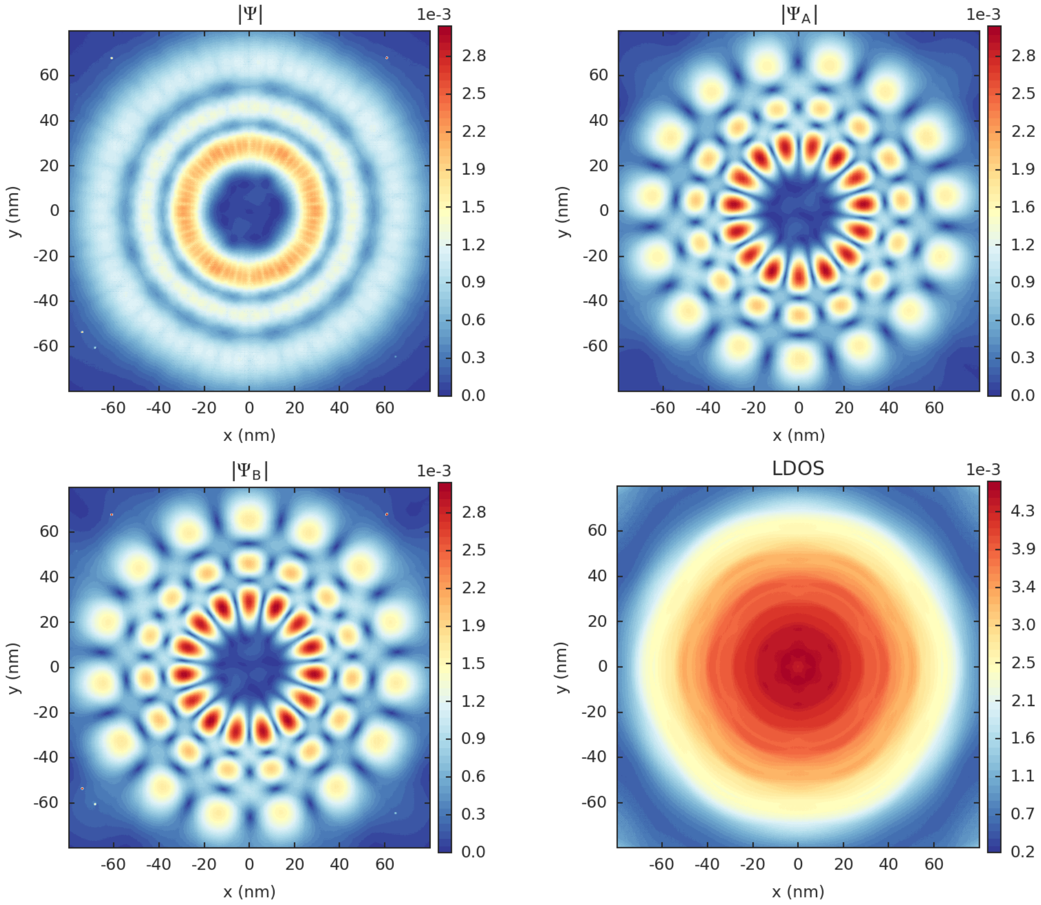


Fig. 9.3.3: The wavefunction $|\Psi|$ for a strongly confined WGM resonance for $E = -50$ meV (with large angular momentum $m = 15$). The individual sublattice components $|\Psi_A|$ and $|\Psi_B|$ each contain half the resonances shifted radially with regard to one another. The last panel presents the LDOS calculated with 4 meV broadening around the same energy as the presented state $|\Psi|$.

reflected at more oblique angles thus forming a higher number of resonances with stronger confinement.

The last panel of Fig. 9.3.3 shows the spatial LDOS for the same energy as the wavefunction in the previous panels. Note that the LDOS is calculated with a small broadening of 4 meV, thus it captures multiple WGM modes with different confinements strengths while the wavefunction figure represents just a single WGM mode. Higher angular momentum modes will be confined closer to the circumference, while lower m modes are closer to the center. The LDOS captures multiple modes in a small energy range which gives the multiple concentric ring structure.

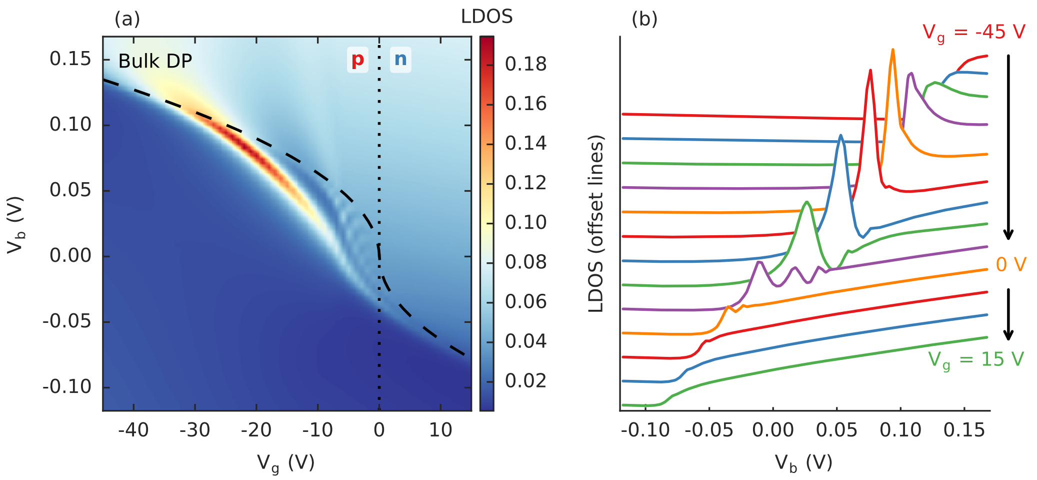


Fig. 9.4.1: (a) Colormap of the LDOS as a function of backgate (V_g) and tip bias voltage (V_b). The dashed curve indicates the bulk Dirac point (DP) and the vertical dotted line at $V_g = 0$ indicates the charge neutrality point which separates the p- and n-doped regimes. (b) Line cuts of the LDOS colormap taken in 5 V increments of V_g (lines are offset for clarity).

9.4 Tip-induced collapse

We turn our attention to the sharp STM tip model. The effective tip-induced potential was already presented in Fig. 9.2.2(b) for a cone-like tip shape with $\Delta V_{wf} = 0.7$ eV. Note that the potential is very narrow, resembling the Coulomb potential function. We compute the LDOS for a full range of gate voltages V_g and V_b. The results are presented as a colormap in Fig. 9.4.1(a). A single high-intensity resonance is clearly visible in the LDOS. It corresponds to the atomic collapse (AC) state which forms in the point-charge-like potential of the sharp STM tip.

In the p-doped regime, at about $V_g = -30$ V, the resonance forms below the bulk Dirac point and moves down as V_g is increased. The resonance reaches its maximum distance from the DP at the charge neutrality point ($V_g = 0$) but then disappears abruptly after crossing it. This sudden disappearance matches the behavior of AC resonances at the transition between the p- and n-doped regimes which was previously observed experimentally in Refs. [40, 118]. Comparing the LDOS in the p-doped and n-doped regimes directly, as shown in Fig. 9.4.2, it is clear that the AC state is only present in the p-doped region and disappears completely in the n-doped regime. Our model does not define any special conditions for the charge neutrality point and the

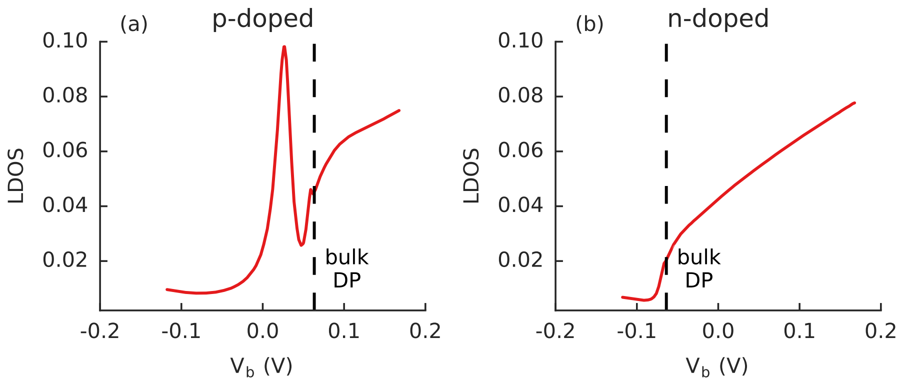


Fig. 9.4.2: Cuts of the LDOS from Fig. 9.4.1 for (a) the p-doped ($V_g = -10$ V) and (b) n-doped regime ($V_g = 10$ V).

simulation evolves smoothly as function of V_g, thus the abrupt disappearance is purely due to the change of the tip-induced potential profile.

The behavior around the charge neutrality point and the transition that happens here deserves a closer look. We examine it in Fig. 9.4.3. The potential in the p-doped regime ($V_g = -10$ V) is sharp and Coulomb-like. The point-charge-like behavior produces a single peak in the LDOS corresponding to the AC state. As V_g is increased to the near-neutral range (-5 to -2 V), the

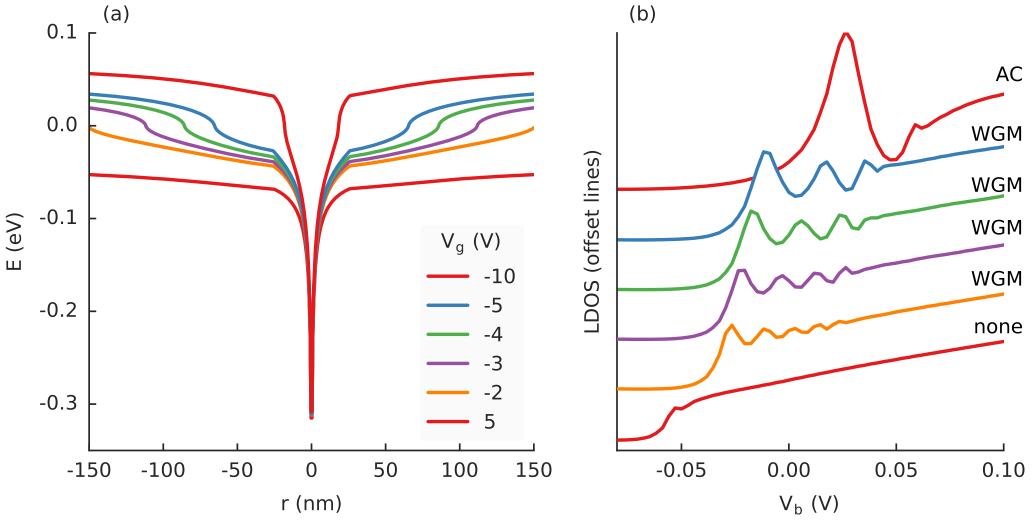


Fig. 9.4.3: Closer look at V_g values at the transition from the p-doped ($V_g < 0$) to the n-doped ($V_g > 0$) regimes. (a) Tip-induced potential. (b) The corresponding LDOS.

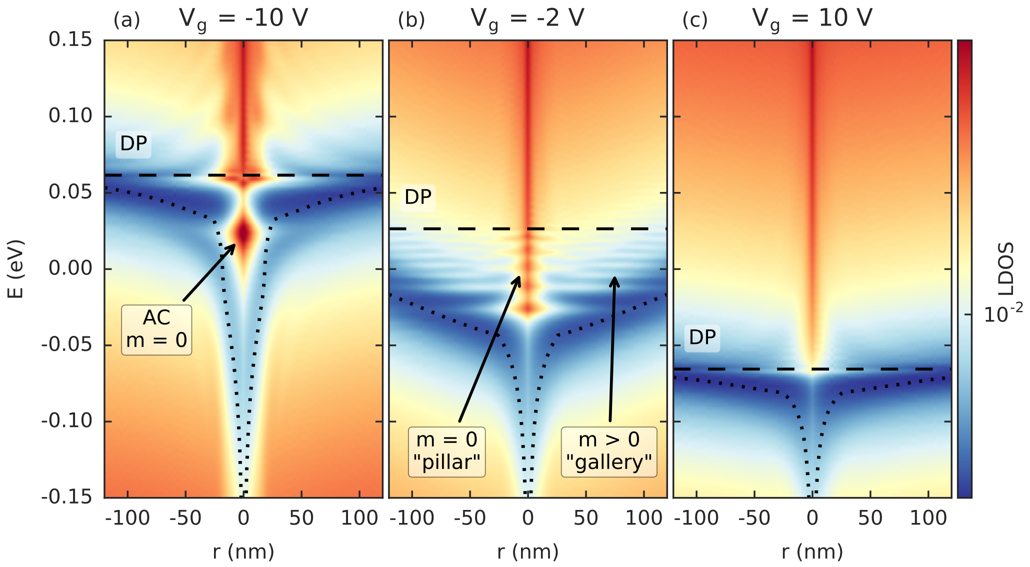


Fig. 9.4.4: Colormap of the LDOS as a function of position for three regimes: (a) AC regime ($V_g = -10$ V), (b) WGM regime ($V_g = -2$ V) and (c) empty well ($V_g = 10$ V). The dashed horizontal line indicates the build Dirac point. The dotted curves indicate the spatial profile of the tip-induced potential.

top of the potential opens up which produces additional smaller LDOS peaks. These are WGM resonances which form in the newly widened cavity. After crossing the charge neutrality point ($V_g = 5$ V), the top part of the potential completely flattens out. The WGM states cannot be maintained without the resonator cavity, thus the LDOS also flattens out. At this point the AC peak is also gone. Notice that the bottom of the potential in Fig. 9.4.3(a) is constant with V_g, but the top descends steadily. This lowers the strength of the potential back towards to the subcritical regime and thus the collapse cannot be sustained.

We have now identified three distinct regimes which arise as the backgate voltage changes the sample doping. To better characterize these regimes, we compute the LDOS as a function of position in Fig. 9.4.4.

While the sample is heavily p-doped, Fig. 9.4.4(a), a high intensity resonance, associated with atomic collapse, is present in the narrow part of the potential. Figure 9.4.5(a) presents a wavefunction computed at this energy. Because of the quasi-bound nature of AC states, multiple wavefunctions are caught in the narrow energy range of this resonance. This is only one of several very similar wavefunctions present there, which all feature central confinement, indicative of zero angular momentum. The narrow potential well does not

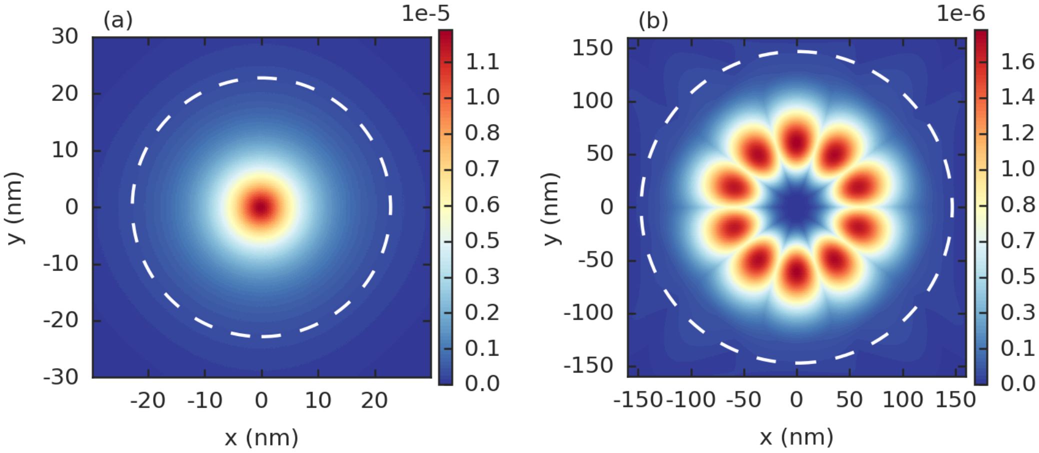


Fig. 9.4.5: Wavefunction $|\Psi|^2$ for (a) an $m = 0$ state in the AC regime and (b) an $m > 0$ state in the WGM regime, as indicated by the arrows in Figs. 9.4.4(a) and (b), respectively. The white dashed circle indicates the circumference of the pn-juncion at the energy of each state (≈ 27 meV and ≈ -5 meV, respectively). Note that the radius is much larger for the (b) state, as expected by the wide potential seen in Fig. 9.4.4.

support any higher angular momentum modes.

When the backgate voltage brings the sample close to the charge neutrality point, Fig. 9.4.4(b), the WGM regime takes over: all the states are located in the large resonator cavity. Two distinct features are visible: the “pillar” and the “whispering gallery”. The high-intensity LDOS region in the center resembles a pillar and consists of $m = 0$ states. These states have a maximum in the center and they look identical to the states found in the AC regime, as shown in 9.4.5(a). On the other hand, $m > 0$ states have a node in the center, as seen in 9.4.5(b). Another example of an even higher angular momentum WGM state is presented in Fig. 9.3.3. These high angular momentum states feature confinement away from the center, thus they populate the “whispering gallery”.

Comparing the AC and WGM regimes, the main difference is due to the size of the resonator cavity. Unlike the AC regime, the wide potential profile of the WGM regime allows it to supports both low and high angular momentum modes. The states with high angular momentum are easy to distinguish as they are confined closer to the circumference of the pn-junction.

When the sample becomes n-doped, Fig. 9.4.4(c), the WGM resonator cavity is gone and the lower narrow potential is also too weak to support any states.

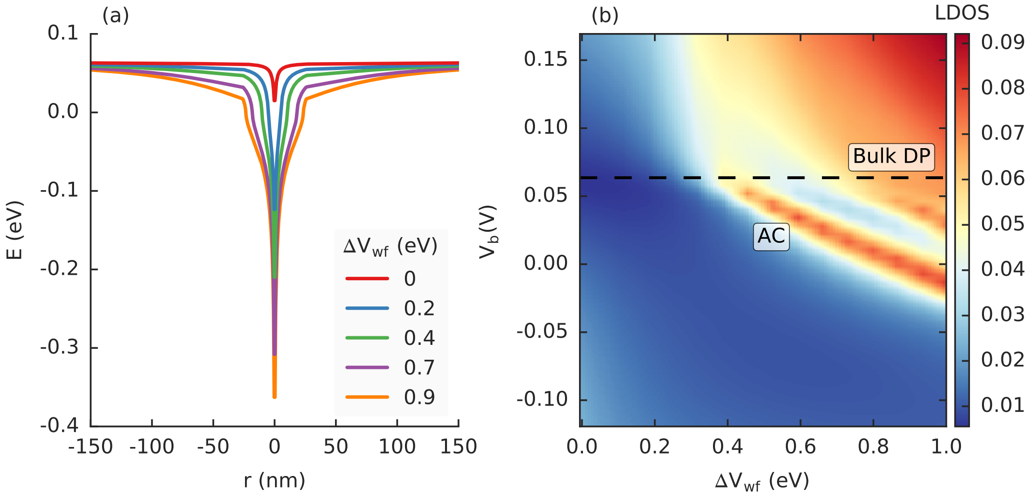


Fig. 9.4.6: Influence of the work function difference between the STM tip and graphene sample for $V_g = -10$ V and $V_b = 0$. (a) The tip-induced potential. (b) The simulated LDOS. The label indicates the atomic collapse peak (AC) which appears just below the bulk Dirac point (DP) after ΔV_{wf} crosses into the supercritical regime.

Influence of the work function difference

The work function difference between the STM tip and graphene is very important for the formation of the AC state. As can be seen in Fig. 9.4.6, only high values of $\Delta V_{wf} > 0.4$ eV support the formation of resonances in the LDOS. Smaller values present a featureless LDOS and the collapse resonance appears abruptly below the Dirac point, just like for external impurities with supercritical charge. Although the tip-induced effect resembles a small supercritical charge, the potential is not exactly the Coulomb function. Thus, quantitative differences are expected when comparing the evolution of the tip-induced AC with ΔV_{wf} to the evolution of the impurity-induced AC with β (see the supercritical vacancy for example).

It has been shown experimentally that ΔV_{wf} can be changed by functionalizing the tip with different materials [134]. The commonly used Pt-Ir tip has a work function very close to that of graphene and thus presents a non-invasive probe which has been used in many experiments [126]. On the other hand, covering the tip with Au coating would produce a very high $\Delta V_{wf} = 0.7$ eV [135]. The induced charge could then tap into the supercritical regime. This enables the experimental observation of the first atomic collapse resonance, as shown in Fig. 9.4.6(b).

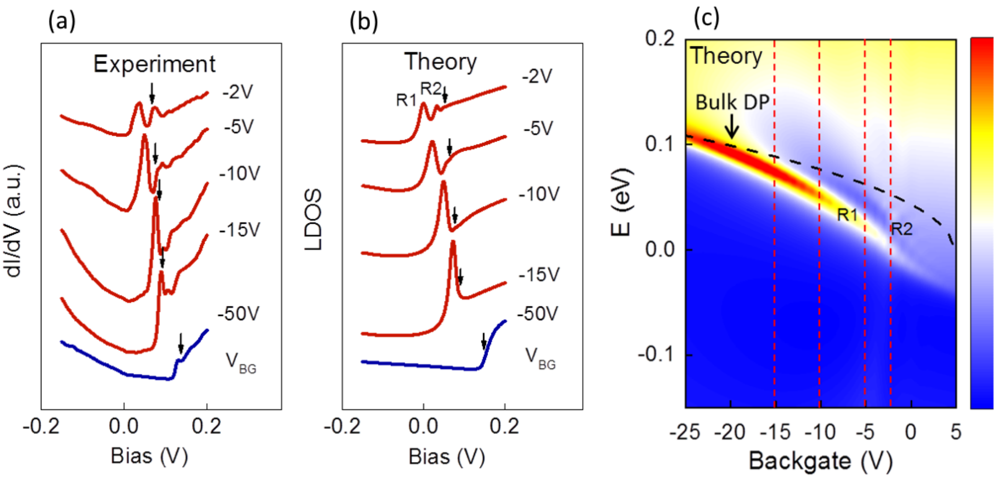


Fig. 9.4.7: (a) Backgate dependence of the dI/dV spectrum as measured by the STM tip (V_b = -50meV, I = 0.1nA). (b) Simulated LDOS. The black arrows indicate the position of the Dirac point (DP) for each backgate voltage. R1 and R2 label the resonance peaks of the atomic collapse states. (c) Simulated evolution of the LDOS with the backgate voltage. The black dashed curve labels the Dirac point far away from the pn-junction. The vertical lines indicate the position of the cuts plotted in panel (b). The experimental results are taken from [147].

Comparison with experiment

This work was conducted in collaboration with the experimental group of Eva Andrei from Rutgers University. The experiment consists of a dual-gate setup. The graphene sheet sits on top of a 300 nm SiO_2 dielectric layer and a highly n-doped silicon substrate which acts as the backgate electrode (see Fig. 9.1.1 for an illustration). The STM tip acts as the top gate which controls local sample doping and allows for the creation of pn-junctions, as described earlier. A non-invasive Pt-Ir is first used to characterize the neutral gate dependence of the graphene sample. The tip is then gently pushed into a gold electrode. The Au coating which remains on the tip changes the work function difference to 0.7 eV. The dI/dV spectra is measured at different values of the backgate voltage. The experimental results are presented in Fig. 9.4.7(a).

For the simulation, we use the theoretical model as outlined in the previous section. The model has several parameters which can be tuned to more closely match the experimental setup. These include the shape of the STM tip $d_T(r)$, the tip-sample work function difference ΔV_{wf} and the ratio of the dielectric constant and thickness of the gate insulator k/d_g. The values of

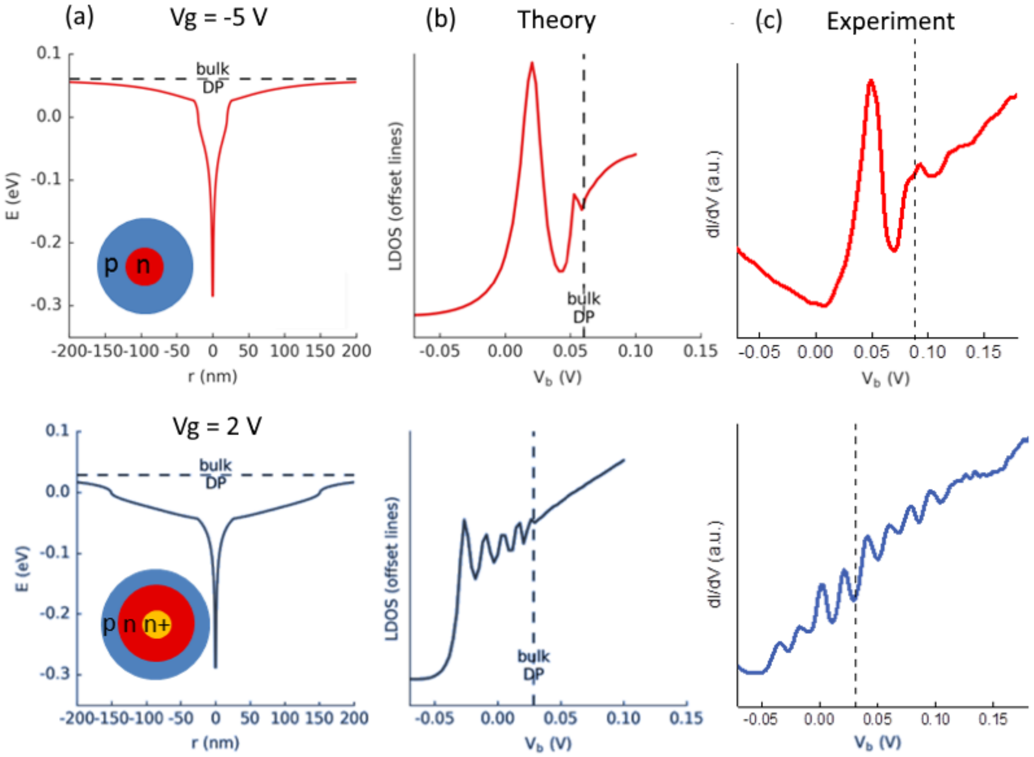


Fig. 9.4.8: Comparison of the AC (top) and WGM (bottom) regimes. (a) Profile of the tip-induced potential. Inset shows a schematic drawing of the junction. (b) Simulated LDOS for the potential profile on the left. (c) Experimental dI/dV curves for the same doping level. The experimental results are taken from [147].

these parameters, as specified in the theoretical model section, have already been adjusted according to the experiment. The only additional change we shall make here is to set $V_{g0} = 4$ V instead of 0. This parameter accounts for the residual doping of the substrate, i.e. it indicates the backgate voltage of the charge neutrality point, which is 4 V for this experimental setup. This change does not make any qualitative difference for the results but merely acts as an offset value (e.g. Fig. 9.4.1 would be identical, just shifted by +4 V in V_g).

With this experimental fit taken into account, the simulated LDOS is presented in Figs. 9.4.7(b) and (c). We see good agreement with the experimental data. The highly p-doped regime ($V_g = -50$ V) is rather featureless, but as V_g is increased a sharp peak appears below the Dirac point. This feature is consistent with the atomic collapse phenomena.

Increasing the backgate voltage up to the charge neutrality point results in a transition to the WGM regime, as discussed previously. Figure 9.4.8 compares the theoretical and experimental results. The agreement is good, as both theory and experiment show a clear distinction between the AC and WGM regimes.

9.5 Conclusions

We studied the effect of STM-tip-induced circular pn-junctions in graphene. We computed the effective potential profile induced by the STM tip based on the shape, bias voltage and work function of the STM tip. We used the kernel polynomial method to compute the LDOS as measured by the tip. It was shown that a sharp tip can trigger the atomic collapse phenomena. In contrast to previously considered deposited impurities or charged vacancies, the point-like charge is produced solely by STM gating which allows for a high degree of control. In order to reach the supercritical charge regime, the tip needs to be coated in gold which creates a large work function difference relative to graphene.

As graphene becomes increasingly p-doped (higher backgate voltage) the profile of the induced potential widens. The increased radius of the pn-junction means that it can also support very high angular momentum states: whispering-gallery modes. Thanks to the large resonator cavity, electrons can reflect from the interface at oblique angles. Klein scattering ensures that such reflections are near perfect thus resonances form at energies where the wavelength matches an integer multiple of the circumference.

By tuning the backgate and tip bias voltages the system was continuously transformed from the quantum-relativistic phenomena of atomic collapse to the electron optics effect of whispering-gallery modes. Both phenomena confine electrons, but the underlying mechanism is quite different. The supercritical charge traps low-angular momentum electrons in spiraling orbits where the probability of finding the electron increases sharply toward the center. On the other hand, WGM confinement is ring-like and presents a high-angular momentum electron skipping along the circumference.

CHAPTER 10

Atomic collapse in the presence of a magnetic field

10.1 Introduction

It was shown that graphene exhibits the analogue of atomic collapse [112, 113], a fundamental phenomena in quantum electrodynamics (QED). Thanks to graphene's very large fine structure constant, the critical charge required to trigger collapse is much smaller than for real atomic nuclei. This has made it possible to realize this phenomena experimentally, with observations closely matching the predictions of QED [40, 118].

Another longstanding prediction from QED is that a magnetic field should be able to enhance the effect [148]. A magnetic field confines the motion of the electron, therefore bringing it closer to the nucleus. As a result, the required value of the critical charge decreases as a function of the field strength. However, this is where the graphene analogue may diverge from the original. QED considers (3 + 1) dimensions where the magnetic field acts on the electron in a plane, but not on the other degree of freedom. Due to its flat nature, the electrons in graphene are confined to (2 + 1) dimensions, which results in a different problem.

It was shown experimentally that a charged impurity lifts the orbital degeneracy of Landau levels, thus splitting them into discrete states [116]. However, the experiment only considered a charge in the subcritical regime. Previous theoretical studies have had conflicting conclusions about the influence of the magnetic field on the value of the critical charge. In Ref. [149], it was predicted that the critical charge vanishes for massless carriers at any finite magnetic field. On the other hand, Refs. [150–152] showed that the critical charge will not change. The problem itself cannot be solved exactly in analytic form. Some approaches for solving the problem, such as perturbation theory, can have great difficulty due to the strong effect of both electric and magnetic fields. In this chapter, we examine the problem using the numerical tight-binding approach in an effort to gain more accurate results and to solve the above issue.

We investigate the problem of a supercritically charged impurity in graphene in the presence of a magnetic field. The Landau levels split into individual orbital states in the presence of the impurity, where the lowest energy states are of special interest. They are closely related to the atomic collapse resonances which appear without a magnetic field.

10.2 Theoretical model

The tight-binding Hamiltonian for graphene in the presence of a charged impurity is given by

$$H = \sum_{\langle i,j \rangle} \left(t_{ij} a_i^\dagger b_j + H.c. \right) + \sum_i V(\vec{r}_i{}^A) a_i^\dagger a_i + \sum_i V(\vec{r}_i{}^B) b_i^\dagger b_i, \tag{10.2.1}$$

where $t_{ij} = -2.8$ eV is the hopping energy, operators $a_i(a_i^\dagger)$ and $b_i(b_i^\dagger)$ create (annihilate) an electron at site $\vec{r}_i$ of sublattice A and B, respectively, and $\vec{r}_i^{A,B}$ is the distance between the carbon atoms and the impurity. In the presence of a uniform magnetic field, of strength B, perpendicular to the graphene plane, the hopping parameters are replaced by the Peierls substitution, $t_{ij} \to t_{ij} e^{i2\pi\Phi_{ij}}$, where $\Phi_{ij} = (1/\Phi_0) \int_{\vec{r}_i}^{\vec{R}_j} \vec{A} \cdot \vec{dl}$ is the Peierls phase, with $\Phi_0 = h/e$ the magnetic quantum and $\vec{A} = B/2(y, -x)$ the vector potential in the symmetric gauge.

The electric potential of the impurity $V(r)$ has the form of the Coulomb potential with cutoff radius r_0 which accounts for the finite size of the charge,

$$V(r) = \begin{cases} -\hbar v_F \frac{\beta}{r_0}, & \text{if } r \leq r_0 \\ -\hbar v_F \frac{\beta}{r}, & \text{if } r > r_0 \end{cases}, \tag{10.2.2}$$

where we take $r_0 = 0.5$ nm as in the previous chapter. Here we only consider a pure charge impurity, without a vacancy. The dimensionless coupling constant $\beta \equiv Ze^2/\kappa\hbar v_F$ gives the raw impurity charge Z scaled by the relative permittivity κ and the Fermi velocity v_F. In the theoretical model we change the charge coupling through the parameter β. In experiments, a constant charge Z may be present, while the relevant coupling β may be tuned by applying a gate voltage which controls the relative permittivity κ via Landau level occupancy [116] (i.e. screening of the charge Ze).

Considering only states close to the K point, the Hamiltonian Eq. (10.2.1) reduces to $H = v_F \vec{\sigma} \cdot \vec{\Pi} + V(\vec{r})$, where σ_i are the Pauli matrices and $\vec{\Pi} = -i\hbar\vec{\nabla} + e\vec{A}$ is the canonical momentum operator. Solving the eigenvalue problem in the absence of the impurity and neglecting the spin yields the unperturbed Landau level (LL) sequence $E_N = \pm\hbar v_F/l_B\sqrt{2|N|}$, where $l_B = \sqrt{\hbar/(eB)}$ is the magnetic length, $N = 0, \pm1, \pm2, ...$ the level index, and +(-) refers to electron (hole) states. The wavefunctions $\Psi_{Nm}(\vec{r})$ depend on the level index N and the orbital number $m \geq -|N|$ [153]. Without a charge, the problem is translationally invariant so E_N is independent of m and the LLs are infinity degenerate. The impurity lifts the degeneracy, splitting the energy into sublevels. The lowest orbital states are centered around the impurity, while higher order states form concentric orbits around it.

Here, we solve the tight-binding Hamiltonian (10.2.1) numerically for a finite size system in the shape of a hexagonal flake with armchair edges. The impurity is positioned in the center of the flake. In the following calculations we take an edge width of 200 nm, which corresponds with a flake consisting of about four million carbon atoms. The flake is taken sufficiently large such that its finite size does not influence the physics we are interested in.

10.3 Numerical results

Before turning on the magnetic field, we shall briefly review the real-space picture of the atomic collapse resonances in the local density of states (LDOS) in graphene. Figure 10.3.1 presents the space-energy map of the LDOS in the subcritical and supercritical regimes. A subcritical charge ($\beta = 0.4$) enhances

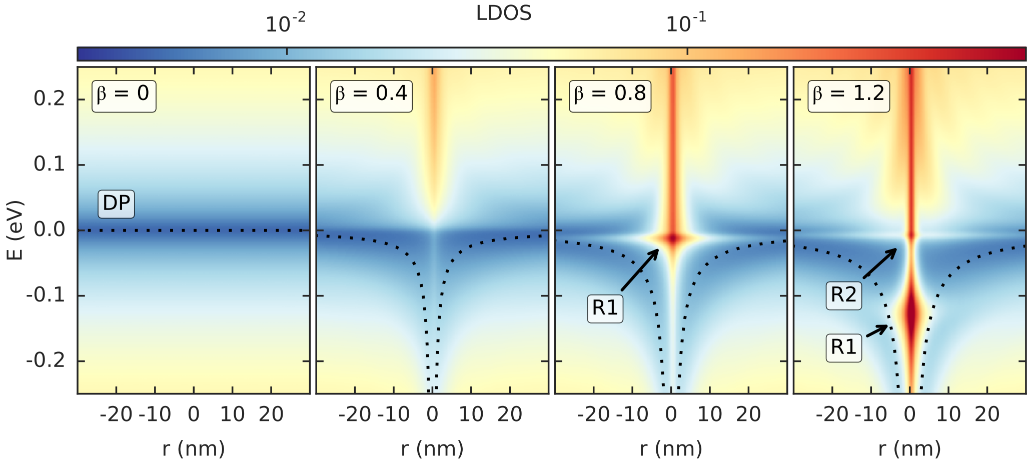


Fig. 10.3.1: Colormap of the LDOS as a function of position and energy in the absence of a magnetic field. The impurity is located in the center ($r = 0$). The charge β varies from 0 to 1.2 as indicated. The dotted line shows the position of the Dirac point (DP) in the first panel and the spatial profile of the Coulomb potential in the following panels. The collapse resonances are labeled $R1$ and $R2$.

the LDOS in the positive part of the spectrum in proximity of the impurity ($r = 0$). Note that it does not cross below the Dirac point. Once the charge becomes supercritical ($\beta = 0.8 > \beta_c$) the high intensity LDOS region crosses below zero energy. The atomic collapse states can only be found at negative energy since they represent coupled states, where an electron from the center can tunnel out and escape as a hole. The collapse resonance in the LDOS is labeled $R1$ as the first of such states to appear with increasing charge β. The LDOS intensity is highest at the center, but disappears quickly at about 10 nm away from the impurity.

Level splitting

The result of the LDOS computation for a magnetic field of $B = 12$ T are presented in Fig. 10.3.2. Without the impurity ($\beta = 0$), the Landau levels appear constant in space, as expected. When a small charge is introduced ($\beta = 0.4$) the Landau levels start to bend and split into individual orbital states near the impurity ($r = 0$). When the charge is increased ($\beta = 0.8$), multiple split levels are clearly visible. States with smaller orbital numbers have lower energy and sink down with the Coulomb potential. At LL $N = 0$, the orbital state $m = 0$ is clearly separated. Similarly, at LL $N = 1$ states $m = -1$ and $m = 0$ have moved lower and separate from the rest of the LL.

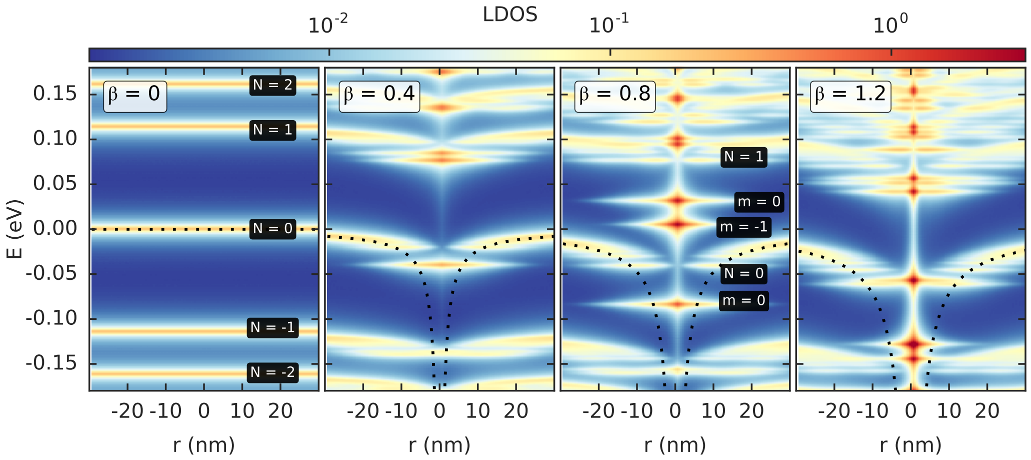


Fig. 10.3.2: Colormap of the LDOS as a function of position and energy with a magnetic field of $B = 12$ T. The Landau levels are labeled as $N = 0, \pm1, \pm2$, while m labels indicate orbital states of their corresponding levels.

Notice that the LDOS at the impurity is an order of magnitude higher than the surrounding area, similar to the case without a magnetic field. After further increasing the charge to $\beta = 1.2$, it becomes difficult to keep track of the split states since they start mixing as we shall see next.

Anticrossing series

The resonances associated with atomic collapse in graphene have high intensity only in close proximity to the charge. They disappear quickly, only a few nanometers away from the impurity. For this reason, we will be focusing mainly on the properties at the point of the impurity ($r = 0$). We calculated the LDOS as a function of energy E and the charge of the impurity β in Fig. 10.3.3 for various magnetic fields B.

Without a magnetic field, the signature of collapse is easy to spot as high intensity resonances at negative energy (labeled $R1$-$R3$ in Fig. 10.3.3). As β is increased the $R1$ resonance moves down and broadens, while a second ($R2$) resonance appears just below the Dirac point. Both resonances are clearly set apart from the rest of the (mostly homogeneous) local density of states.

When a magnetic field is applied ($B = 2$ and 12 T in Fig. 10.3.3) we can see a mix of Landau levels and collapse resonances. Landau levels are clearly formed at low β. As the charge β is increased, we can see Landau level

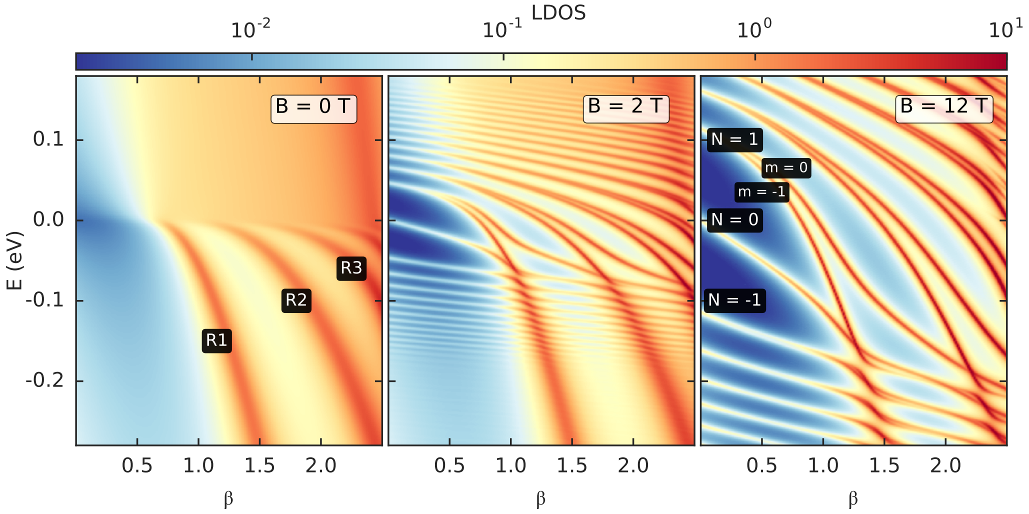


Fig. 10.3.3: Colormap of the LDOS at the impurity ($r = 0$) as a function of charge β and energy E. The magnetic field B varies as indicated in the panels. The labels $R1$ to $R3$ indicate the collapse resonances in the order in which they appear. The $N = 0, \pm1$ labels indicate Landau levels.

$N = 1$ splitting into individual orbital states $m = -1$ and $m = 0$. At higher β, Landau level $N = 2$ splits, then $N = 3$ and so on.

The lower split state ($m = -1$) of each positive LL is of special interest. These lines correspond exactly with the lines of the collapse resonance at $B = 0$. Once the lower orbital state of LL $N = 1$ intersects with $N = 0$, a series of anticrossings is formed which continues to follow the same line as the $R1$ collapse resonance at $B = 0$. This line of anticrossings also retains a very high LDOS intensity which is at least an order of magnitude higher than the regular Landau levels.

The inter-level spacing of the LLs is generally preserved while β changes. It is only disturbed while crossing a collapse resonance. After crossing the $R1$ resonance, the LLs with $N < 1$ shift down by one, e.g. $N = 0$ moves lower to take the place of $N = -1$ while $N = -1$ shifts to $N = -2$, etc. The positive LLs behave differently. As described earlier, the lower energy state LL ($N = 1$, $m = -1$) follows the collapse resonance until it crosses $N = 0$. The higher orbital state ($N = 1$, $m = 0$) continues to the second resonance. Only levels up to $N = 1$ are affected by the $R1$ resonance. After crossing the $R2$ resonance, $N = 2$ is included and the pattern repeats: the $N < 2$ Landau levels shift down by one, while $N = 2$ splits so that the lower state follows resonance $R2$, while the higher states continues to $R3$.

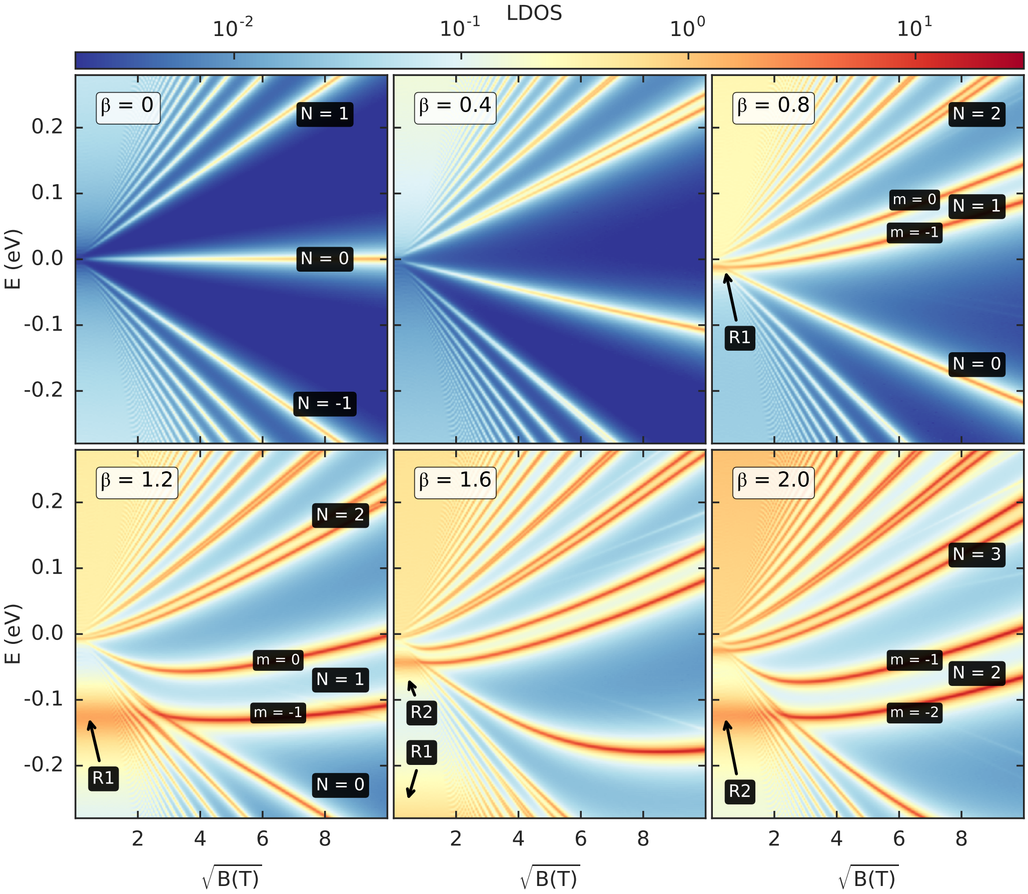


Fig. 10.3.4: Colormap of the LDOS at the impurity ($r = 0$) as a function of energy E and the square root of the magnetic field $\sqrt{B}$. The charge β varies as labeled in the panels.

Landau level bending

The Landau levels in graphene feature a linear dependence on the square root of the magnetic field $\sqrt{B}$. Computing the LDOS as a function of $\sqrt{B}$ without any charge ($\beta = 0$) reveals the expected linear LL lines of high LDOS intensity in Fig. 10.3.4. When a small charge is added ($\beta = 0.4$), the LLs remain generally linear, but they are slightly tilted downwards. At this point LL $N = 1$ is split into individual orbital states ($m = -1$ and $m = 0$). As the charge is increased into the supercritical regime ($\beta = 0.8$) the levels become non-linear. The collapse resonance $R1$ is visible near $\sqrt{B} = 0$.

The curvature of the Landau levels becomes especially visible for $\beta = 1.2$. The $R1$ resonance is quite apparent at low values of $\sqrt{B}$. Without a magnetic field, this peak appears as a broad LDOS resonance which is an indicator of the supercritical regime. Note that the $R1$ resonance (low $\sqrt{B}$) is located at the same energy as the $m = -1$ state of Landau level $N = 1$ (high $\sqrt{B}$). At $\beta = 1.6$, the second resonance starts to appear just below zero energy, as indicated by label $R2$ in Fig. 10.3.4. Further increasing the charge ($\beta = 2$) moves the $R2$ resonance lower and with it the $N = 2$, $m = -2$ state.

Note that the Landau levels in the energy range of a resonance ($R1$ or $R2$) have a larger LDOS intensity than their neighboring levels, which further supports the connection of the collapse resonance and the lowest orbital states. The LLs also appear to bend slightly in this region, while the levels that are not influenced by the resonance remain linear as function of $\sqrt{B}$. The collapse resonance looks to be directly connected to non-linear scaling of Landau levels.

Scaling anomaly

The energy of the Landau levels can be written as,

$$E_N(B) = v_F\sqrt{2|N|\hbar}\sqrt{B} = \nu_N\sqrt{B}, \tag{10.3.1}$$

where ν_N is the level scaling factor. When there is no impurity in the system, this factor is constant and independent of the magnetic field, i.e.

$$\frac{\partial \nu_N}{\partial B} = 0. \tag{10.3.2}$$

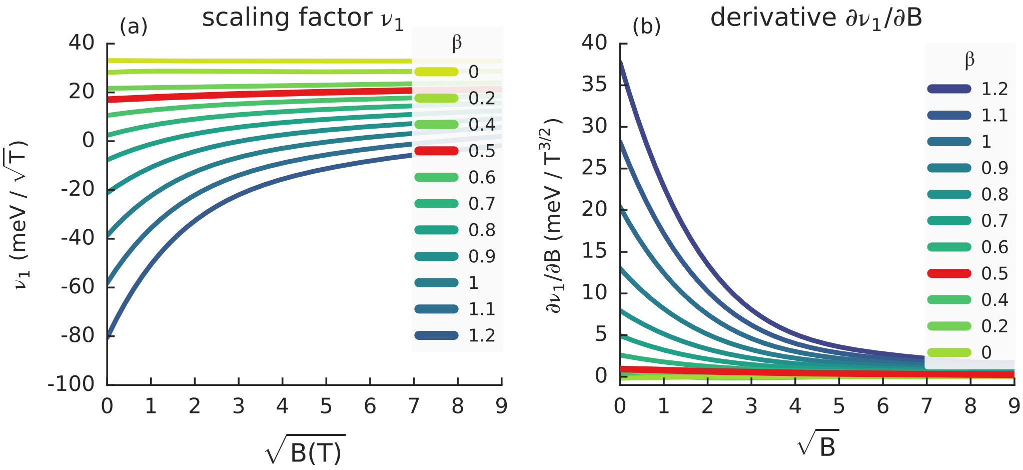


Fig. 10.3.5: (a) Scaling factor ν_1 for different values of the charge β. (b) The derivative of the scaling factor for Landau level $N = 1$.

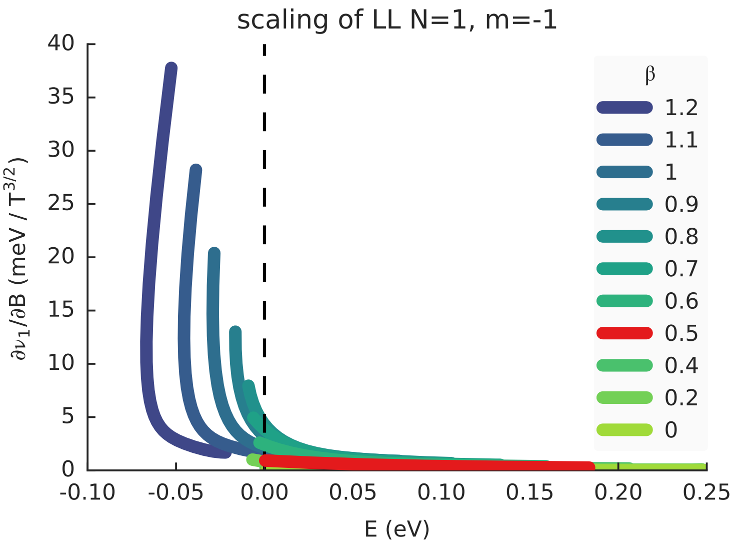


Fig. 10.3.6: The derivative of the scaling factor for Landau level $N = 1$. This is the same as Fig. 10.3.5, but this time as a function of energy instead of the magnetic field.

On the other hand, when Eq. (10.3.2) is non-zero, it means that the level has a scaling anomaly, which should be the case for a supercritical charge.

We used the LDOS data to calculate the scaling factor ν_N for LL $N = 1$. The results are shown in Fig. 10.3.5(a). However, the more interesting results are for the derivative $\partial\nu_1/\partial B$, presented in Fig. 10.3.5(b). The derivative is close to zero for β up to $\beta_c = 0.5$, independent of the magnetic field. For $\beta > 0.5$ there is a clear non-zero derivative at small values of the magnetic field. As the magnetic field is increased, the derivative approaches zero asymptotically. Note that this does not indicate that the scaling anomaly disappears at very high values of the magnetic field. The scaling anomaly is mainly a function of energy.

To show the strong dependence of the derivative to energy, we replotted Fig. 10.3.5(a) as function of energy in Fig. 10.3.6. Notice that the β values up to 0.5 (that have zero derivative, i.e. normal scaling) are located in the region of positive energy. Once the LL starts crossing into negative energy ($\beta = 0.6$ and higher), the derivative becomes finite indicating anomalous scaling. This mirrors the appearance of the collapse resonance below the Dirac point without a magnetic field. As β is increased the LL moves lower in energy (just like the resonance) and the derivative increases indicating stronger anomalous scaling as a function of increasing β.

10.4 Conclusions

We used the tight-binding method to model a charged impurity in graphene in the presence of a magnetic field. Without the field, atomic collapse in graphene appears as a resonance that forms just below zero energy. When the magnetic field is activated, the resonances are replaced by a Landau level anticrossing series at the same energy.

A Coulomb-like charge causes Landau levels to split into individual orbital states. The lowest orbitals of the positive Landau levels have special behavior. They are directly related to the collapse resonance. When expressed as a function of the charge, the lowest orbital states follow the exact line of the resonance until they cross LL $N = 0$. At that point a series of anticrossings is formed which continues along the line of the collapse resonance.

Landau levels that are caught in the collapse resonance exhibit anomalous scaling as a function of the magnetic field. The scaling is mainly a function of the energy and it closely mirrors the collapse resonance: anomalous scaling appears just as a Landau level crosses below the Dirac point.

We find that the critical charge for atomic collapse does not change in the presence of a magnetic field. The two main indicators of supercritical charge that have been shown, the anticrossing series and anomalous scaling, both appear in a broad energy range and closely mirror the zero field collapse resonance.

CHAPTER 11

Molecular collapse

11.1 Introduction

Supercritical charge centers in graphene exhibit the atomic collapse phenomena, i.e. they represent the analogue of the collapsing artificial atoms. The collapse resonances R1 and R2 are the analogues of the 1S and 2S atomic states. With real atoms, molecules form thanks to the overlap of atomic orbitals. The overlap can be modeled using the linear combination of atomic orbitals method. As the two atomic nuclei are brought closer to each other, a single atomic states splits into a lower and higher energy branch: the bonding and anti-bonding states. The bonding states is characterized by a finite probability of finding the electron at the midpoint between two atoms. The anti-bonding states has higher energy and zero-probability at the midpoint, thus it works against the molecular bond. Because of the atomic collapse analogue in graphene, it is expected that two supercritical charges should exhibit the same kind of states.

11.2 Results

We model two identical Coulomb charges with $\beta = 0.95$ located at a distance d from each other. For simplicity we consider a standalone charge in pristine graphene (no vacancy). The LDOS is calculated as a function of the distance between the artificial nuclei and shown in Fig. 11.2.1(a). When the charges

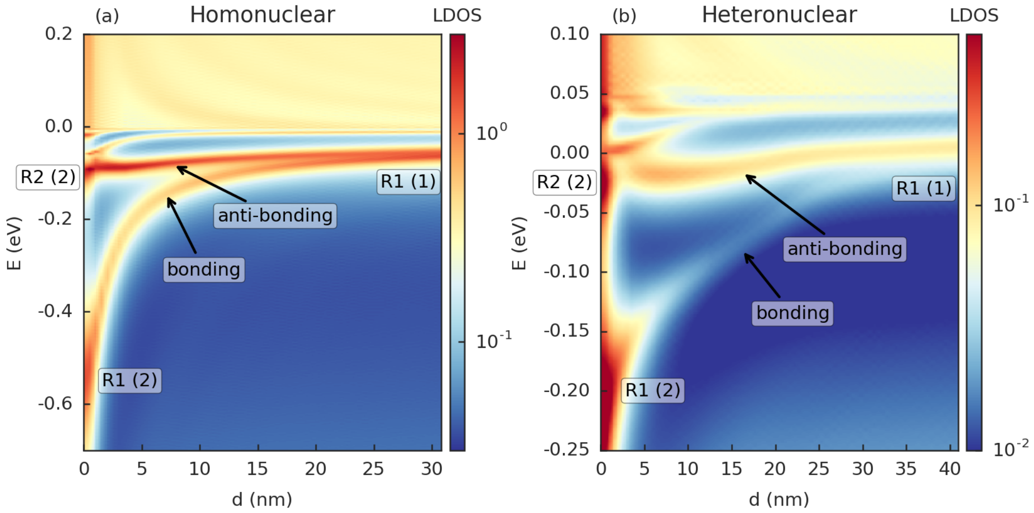


Fig. 11.2.1: Colormap of the LDOS as a function of energy and the distance between charge centers. (a) For two identical charged vacancies with $\beta = 0.95$. The LDOS is computed at the position of one of the charges. (b) Two different charges. One is a charged vacancy with $\beta = 0.95$ and the other is the STM tip with $V_g = -5$ V. The LDOS is computed at the position of the STM tip.

are far apart they behave as two independent supercritical nuclei. This is visible in Fig. 11.2.1(a) at $d \approx 30$ nm where only a single collapse resonance $R1$ is present. As the distance is reduced, the resonance splits into a lower-energy bonding and higher-energy anti-bonding state. This is analogous to molecule consisting of two identical atoms (homonuclear). When the distance goes to zero, the two nuclei have completely merged and behave like one nucleus with double the charge. The bonding state transits into the $R1$ collapse state of the new nucleus, while the anti-bonding state transits into $R2$.

The LDOS spatial maps at $d = 8$ nm, Fig. 11.2.2, reveals the opposite nature of the split states. As its name implies, the lower energy bonding state has significant LDOS intensity in the bonding region between the nuclei. On the other hand, the anti-bonding states has low LDOS intensity at the midpoint.

Next, we model a heteronuclear molecule consisting of two different artificial nuclei. Experimentally, this can be realized as a supercritically charged vacancy with $\beta = 0.95$ and collapse-inducing STM. The STM tip-induced charge is significantly weaker compared to the charged vacancy: it would be close to $\beta = 0.6$, although we cannot speak in terms of β for the tip-induced charge since it does not have a Coulomb barrier shape. Still, the charge

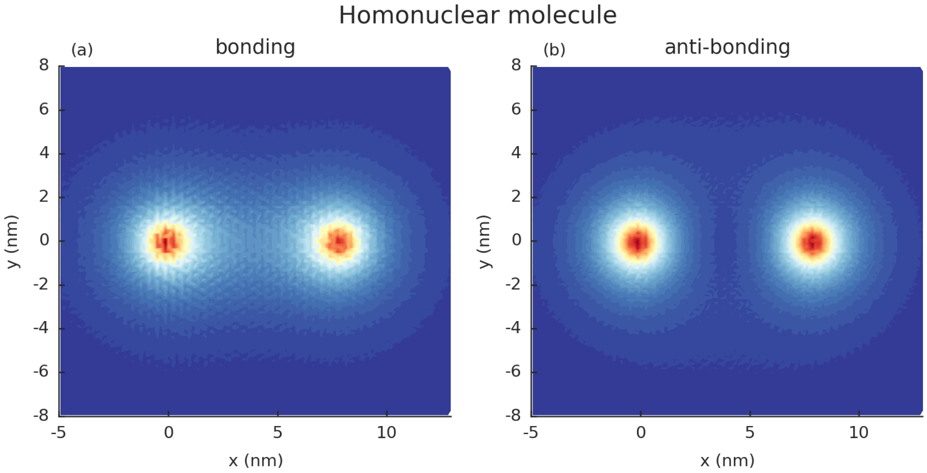


Fig. 11.2.2: Spatial LDOS colormap for two identical charges $\beta = 0.95$ at distance $d = 8$ nm. (a) The bonding state with $E = -135$ meV. (a) The anti-bonding state with $E = -77$ meV. The LDOS is calculated with a broadening of 8 meV.

mismatch between the nuclei is significant which makes this case analogous to a heteronuclear molecule (e.g. HCl as opposed to mononuclear H_2). The LDOS as a function of distance is presented in Fig. 11.2.1(b) where the LDOS is computed following the position of the STM tip. As the distance shrinks, the collapse resonance splits as before but in this case the bonding

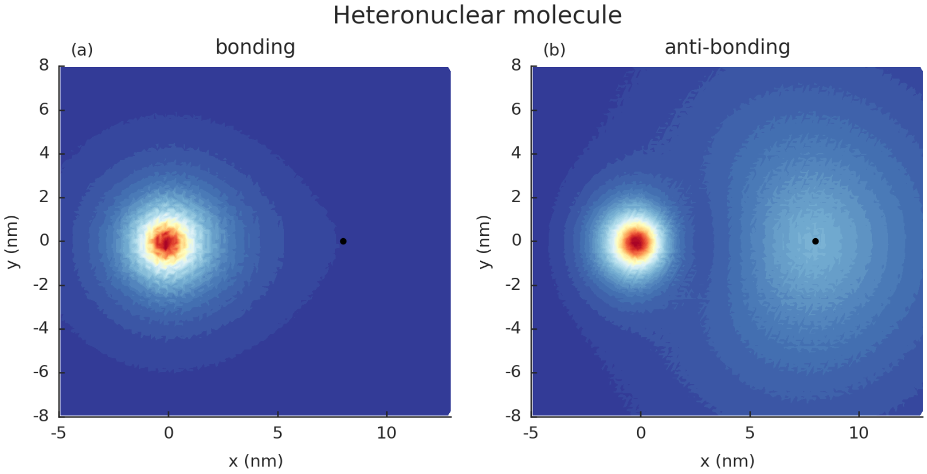


Fig. 11.2.3: Spatial LDOS colormap for a charge $\beta = 0.95$ and STM tip $V_g = -5$ V at distance $d = 8$ nm. (a) The bonding state with $E = -115$ meV. (a) The anti-bonding state with $E = -20$ meV. The black dot indicates the position of the STM tip. The LDOS is calculated with a broadening of 8 meV.

state is much weaker than the anti-bonding state. Looking at the spatial maps in Fig. 11.2.3, we can see a large imbalance between the nuclei due to the charge mismatch. The stronger vacancy charge has a much greater LDOS intensity, however signs of the bonding state are still there as a region of high LDOS is extended from left to right across the nuclei. The anti-bonding state shows a similarly pronounced asymmetry but retains a low LDOS intensity in the bonding region between nuclei. The kind of asymmetry exhibited here is analogous to real heteronuclear molecules. The bonding state is weak because of the large mismatch between the energy levels of the individual nuclei.

11.3 Conclusions

We considered the case of two supercritical charges placed close to each other. It was shown that such artificial nuclei form molecular bonding and anti-bonding states analogous to real molecules. The only difference is that regular atomic orbitals are replaced with collapse resonances, i.e. spiraling orbitals as seen from the quasiclassically point of view.

CHAPTER 12

Summary

In this thesis, we theoretically investigated ways of controlling the electronic properties of graphene using mechanical strain and supercritical electric fields. Klein tunneling renders traditional electric barriers ineffective in graphene which is why these alternative approaches are needed. Strain has long been used to enhance the electrical properties of semiconductors, but the very high strain tolerance of graphene makes it especially well-suited for strain engineering. Supercritical charge centers are a long-standing prediction of quantum electrodynamics. Thanks to the relativistic behavior of low-energy electrons in graphene, the same effect can be achieved at modest charge which presents a new way of producing quasi-bound states.

These phenomena were investigated mainly using the numerical tight-binding approach. The tight-binding model of graphene was described in Chapter 2 along with some of graphene's important electronic features. In order to investigate the physical properties of the outlined phenomena, the eigenvalue problem was solved either using exact diagonalization of the Hamiltonian matrix or using approximative methods. The kernel polynomial method, as described in Chapter 3, was used extensively in this thesis. It is an extremely fast method which can be used to compute the (local) density of states or the Green's function, approximately, with controllable precision. In order to aid the numerical work, a code framework was developed which is available as an open source project. The theoretical framework and ideas behind the code are presented in Chapter 4 while a user guide and concrete code examples are given in Appendix A.

The results part began in Chapter 5 with the investigation of the strain-induced pseudo-magnetic field in graphene. This field mimics many of the properties of a real magnetic field, including the ability to confine electrons in graphene. One key difference is that the pseudo-magnetic field does not break time-reversal symmetry, since it has the opposite sign in the K and K' valleys of graphene. We investigated the model of the strain-induced field derived with increasingly higher order terms in strain. The first order strain term was found to be valid only for very low strains, up to 5%. In order to go up to graphene's full strain limit of 25%, at least the third order term was required.

In Chapter 6 we studied the specific case of out-of-plain strain due to a Gaussian-shaped bump in a graphene flake. Interestingly, even though the bump has perfect circular symmetry, it generates a very distinct non-circular pseudo-magnetic field with six-fold symmetry. Electrons are well-confined in the armchair directions, while the zigzag directions carry valley-polarized current between the inside and outside of the strained region.

Chapter 7 examined the effect of in-plane triaxial strain in bilayer graphene. The generated pseudo-magnetic field was found to be uniform with a peculiar side-effect: broken layer symmetry. At low energy, the difference between the layers is so large that effectively only one layer feels the effects of the pseudo-magnetic field. The second layer is missing the zero-energy pseudo-Landau level, thus it features a large gap between the first positive and negative levels. Effectively, this means that low-energy electrons are localizes to a single layer. The affected layer can be switched by rotating the strain direction by 60°.

Chapter 8 introduced the phenomena of atomic collapse in graphene. The effect was studied theoretically in the presence of supercritically charged impurities and vacancies. The charged single-atom vacancy is of particular interest due to experimental accessibility. A satellite $R1'$ peak emerged as a new branch of collapse due to the locally broken sublattice symmetry of the vacancy. The results were compared with experimental observations and good agreement was found.

Chapter 9 studied the atomic-collapse effect induced solely by the sharp tip of a scanning tunneling microscope (STM). By adjusting the backgate voltage, the induced charge can be continuously tuned from the subcritical to the supercritical regime. The transition is accompanied by whispering-gallery modes, a phenomena originating in optics, which presents an alternate confinement mechanism compared to collapse. This model was derived based on an experimental STM setup. The simulated results showed good agreement with the experiment.

In Chapter 10 we investigated supercritical charge centers in the presence of a magnetic field. We show that the atomic collapse effect persists even after the field is activated. The previously observed collapse resonances are replaced with a series of Landau level anti-crossings as the new signature of collapse. The Landau levels also display anomalous scaling with regard to magnetic field strength in the region of the collapse.

Chapter 11 considered the creation of artificial "collapsing" molecules made up of supercritical nuclei. We found that placing two supercritical charges in close proximity will split the atomic collapse $R1$ state into a lower-energy bonding and higher-energy anti-bonding state.

12.1 Outlook

Although the term "atomic collapse" comes from the analogue to the original phenomena in quantum electrodynamics, these are actually persistent resonances in graphene as seen in experiments. Given the correspondence of collapse states to regular atomic states, supercritically charged impurities can be viewed as artificial atoms in graphene. Therefore, a bonding of two supercritical charges can be viewed as an artificial supercritical molecule as found in Chapter 11. Expanding further on this, a superlattice of charged impurities could produce a supercritical band structure. Studying such a structure could yield new ways of controlling the electronic properties of graphene. As we have seen in Chapter 9, a sharp gate can also act like a supercritical charge. Thus, assembling a comb-like structure of such gates could be used to realized the superlattice structure.

The tight-binding code used for the numerical calculations in this thesis is available as an open source project, called Pybinding. While the code started out as a specialized tool for graphene it has grown into a generalized framework for constructing arbitrary tight-binding models in 1 to 3 dimensions. The package has builtin solvers for finite size and periodic systems, but a recent addition is the ability to interface with the Kwant package and also make use of its solvers which are tailored for transport problems. In that same vein, Pybinding could also be made interoperable with ab initio packages. The ab initio results can be used as input parameters for tight-binding models. Compared to inputing parameters by hand, the ab initio interface would allow for more complex models to be considered. On the other hand, compared to ab initio simulations, the tight-binding approach would use simplified parameters, thus enabling the construction of much larger systems.

CHAPTER 13

Samenvatting

In deze thesis, maakten we een theoretische studie van de manier waarop de elektronische eigenschappen van grafeen beheerst kunnen worden door middel van mechanische rek en superkritische elektrische velden. Klein tunneling maakt traditionele elektrische barrières nutteloos in grafeen, hierdoor zijn deze alternatieve methodes noodzakelijk. Rek wordt sinds lang gebruikt om de elektronische eigenschappen van halfgeleiders te verbeteren, maar de erg hoge rek tolerantie van grafeen, maakt dit materiaal uitermate geschikt voor deze manier van het beheersen van materiaaleigenschappen. Superkritische ladingscenters zijn een reeds lang voorspeld fenomeen in kwantum elektrodynamica. Dankzij het relativistische karakter van de lage energie elektronen in grafeen, kan hetzelfde effect gerealiseerd worden bij eerder bescheiden ladingen, wat leidt tot een nieuwe manier om quasi-gebonden toestanden te realiseren.

Deze fenomenen werden voornamelijk onderzocht met de numerieke methode van de sterke binding (tight-binding). In Hoofdstuk 2 werd het tight-binding model van grafeen beschreven, net zoals enkele van de meest belangrijke elektronische eigenschappen van grafeen. Er werd in deze thesis uitgebreid gebruik gemaakt van de kernel polynomiale methode, die beschreven wordt in Hoofdstuk 3. Dit is een extreem snelle methode die gebruikt kan worden om de (lokale) toestandsdichtheid of de Greense functie benaderend te berekenen met een controleerbare precisie. Om het numerieke werk te vereenvoudigen, werd een codebasis ontwikkeld die ook beschikbaar is als open source project. Het theoretische kader en de ideeën achter de code worden voorgesteld in Hoofdstuk 4, terwijl een handleiding en concrete voorbeelden

voor het gebruik van de code worden getoond in Appendix A.

Het deel met resultaten begint in Hoofdstuk 5 met de studie van het door middel van rek geïnduceerde pseudomagnetische veld, welks theoretische werd afgeleid met steeds hogere orde termen in de rek. De eerste orde term bleek enkel geldig voor erg lage waarden van de rek tot 5%. Om tot de volledige limiet van rek in grafeen te gaan, is minstens de derde orde term nodig.

In Hoofdstuk 6, bestudeerden we het specifieke scenario van een rek loodrecht op het vlak veroorzaakt door een Gaussisch gevormde bult. Opmerkelijk is dat alhoewel de bult een perfect cirkelvormige symmetrie heeft, het resulterende pseudomagnetische veld zeer niet-cirkelvormig is en een rotatiesymmetrie van orde 6 heeft. Elektronen zijn sterk opgesloten in de badkuip richtingen, terwijl de zigzag richtingen een vallei-gepolariseerde stroom vertonen tussen de binnenkant en buitenkant van de uitgerekte regio.

In Hoofdstuk 7 wordt het effect van een triaxiale rek in het vlak bij bilaag grafeen onderzocht. Het gegenereerde pseudomagnetische veld bleek uniform, maar met een opmerkelijk neveneffect: gebroken laag symmetrie. Bij lage energieën, is het verschil tussen de lagen zo groot dat slechts één laag het effect van het pseudomagnetische veld voelt.

Hoofdstuk 8 introduceert het fenomeen van atomaire instorting in grafeen. Dit effect werd theoretisch bestudeerd in aanwezigheid van superkritisch geladen onzuiverheden en vacatures. Het geval van een geladen vacature van één atoom is bijzonder interessant vanwege de experimentele realiseerbaarheid. Ten gevolge van de door de vacature lokaal verstoorde subrooster symmetrie verschijnt er een $R1'$ nevenpiek, tekenend voor een nieuwe instortingstak. Het resultaat werd vergeleken met experimentele waarnemingen en bleek hiermee goed overeen te stemmen.

In Hoofdstuk 9 bestuderen we het effect van atomaire instorting geïnduceerd door de scherpe tip van een scanning-tunnelingmicroscoop (STM). Door het aanpassen van de achterdeur spanning, kan de geïnduceerde lading continu geregeld worden van het subkritische naar het superkritische regime. De overgang wordt gekenmerkt door fluisterende-galerij modes, een fenomeen dat gekend is vanuit optica, dat een alternatief mechanisme voor opsluiting voorstelt, vergelijkbaar met instorting.

In Hoofdstuk 10 onderzoeken we superkritische ladingscenters in de aanwezigheid van een magnetisch veld. We toonden aan dat het effect van atomaire instorting aanhoudt als het magnetisch veld wordt aangezet. De Landau niveaus vertonen abnormale schaling met de sterkte van het magnetisch veld in de regio van instorting.

In Hoofdstuk 11 beschouwen we de creatie van een artificieel “instortend” molecuul opgebouwd uit superkritische kernen. We vonden dat het dicht bij elkaar plaatsen van twee superkritische ladingen de atomaire instorting $R1$ toestand doet opsplitsen in een lagere energie bindings- en hogere energie anti-bindingstoestand.

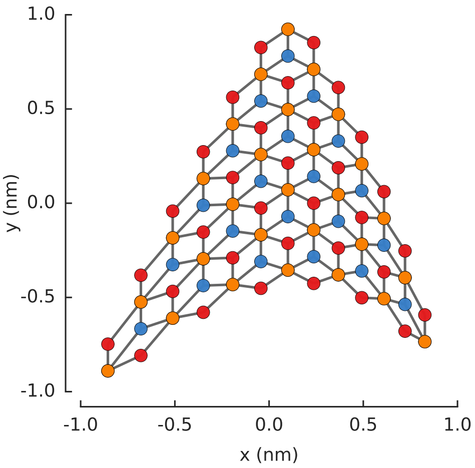
APPENDIX A

Code package: Pybinding

Pybinding is a Python package for tight-binding calculations which was developed along with this thesis to aid the numerical work. It represent a general modeling approach which can be used to for arbitrary tight-binding systems. This appendix contains a brief overview of the features of the code framework. More details are available online: http://docs.pybinding.site

As a very quick sample, the following code creates a triangular quantum dot of bilayer graphene and then applies a custom asymmetric strain function:

```
import pybinding as pb
from pybinding.repository import graphene

def asymmetric_strain(c):
    @pb.site_position_modifier
    def displacement(x, y, z):
        ux = -c/2 * x**2 + c/3 * x + 0.1
        uy = -c*2 * x**2 + c/4 * x
        return x + ux, y + uy, z
    return displacement

model = pb.Model(
    graphene.bilayer(),
    pb.regular_polygon(num_sides=3, radius=1.1),
    asymmetric_strain(c=0.42)
)
model.plot()
```



Within the Pybinding framework, tight-binding models are assembled from logical parts which can be mixed and matched in various ways. The package comes with a few predefined components: crystal lattices, shapes, symmetries, defects, fields and more (like the `graphene.bilayer()` lattice and the `regular_polygon()` shape shown above). Users can also define new components (just like the asymmetric strain above). This modular approach enables the construction of arbitrary tight-binding models with clear, easy-to-use code. Various solvers, computation routines and visualization tools are also part of the package. See the *Basic Tutorial* for a walkthrough of the features.

The full user guide including installation instruction can be found online[*]. The source code repository is located on GitHub[†].

A.1 Features and language

Pybinding is a Python package for numerical tight-binding calculations in solid state physics. The main features include:

- **Declarative model construction** - The user just needs to describe *what* the model should be, but not *how* to build it. Pybinding will take care of the numerical details of building the Hamiltonian matrix so users can concentrate on the physics, i.e. the quantum properties of the model.
- **Fast compute** - Pybinding's implementation of the kernel polynomial method allows for very fast calculation of the Green's function of the Hamiltonian. Exact diagonalization is also available through the use of scipy's eigensolvers. The framework is very flexible and allows the addition of user-defined computation routines.
- **Result analysis and visualization** - The package contains utility functions for post-processing the raw result data. The included plotting functions are tailored for tight-binding problems to help visualize the model structure and to make sense of the results.

The main interface is written in Python with the aim to be as user-friendly and flexible as possible. Python is a programming language which is easy to learn and a joy to use. It has deep roots in the scientific community as evidenced by the rich scientific Python library collection: SciPy. As such, Python is the ideal choice as the main interface for Pybinding. Under the hood, C++11 is used to accelerate demanding tasks to deliver high performance with low memory usage. This is done silently in the background.

A.2 General workflow

The general workflow starts with model definition. Three main parts are required to describe a tight-binding model:

- **The crystal lattice** - This step includes the specification of the primitive lattice vectors and the configuration of the unit cell (atoms, orbitals and spins). This

[*] http://docs.pybinding.site

[†] https://github.com/dean0x7d/pybinding

can be user-defined, but the package also contains a repository of the pre-made specifications for several materials.

- **System geometry** - The model system can be infinite through the use of translational symmetry or it can be finite by specifying a shape. The two approaches can also be composed to create periodic systems with intricate structural patterns. The structure can be controlled up to fine details, e.g. to form specific edge types as well as various defects.

- **Fields** - Functions can be applied to the onsite and hopping energies of the model system to simulate external fields or various effects. These functions are be defined independently of any lattice or specific structure which makes them easily reusable and mutually composable.

Once the model description is complete, Pybinding will build the tight-binding Hamiltonian matrix. The next step is to apply computations to the matrix to obtain the values of the desired quantum properties. To that end, there are the following possibilities:

- **Green's function** - Pybinding implements a fast Chebyshev polynomial routine for calculating the Green's function of the Hamiltonian. This can be applied to quickly compute the local density of states or the transport characteristics of the system.

- **Exact diagonalization** - Eigensolvers may be used to calculate the eigenvalues and eigenvectors of the model system. Common dense and sparse matrix eigensolvers are available via SciPy.

- **User-defined compute** - Pybinding constructs the Hamiltonian in the standard sparse matrix CSR format which can be plugged into custom compute routines.

After the main computation is complete, various utility functions are available for post-processing the raw result data. The included plotting functions are tailored for tight-binding problems to help visualize the model structure and to make sense of the results.

A.3 Performance

One of the main features of Pybinding is an easy-to-use and fast model builder: it constructs the tight-binding Hamiltonian matrix. This can be a demanding task for large or complicated systems (with many parameters). Great care was taken to make this process fast.

We compare the performance of Pybinding with the Kwant package [71]. Both code packages are based on the numerical tight-binding method and can build identical Hamiltonian matrices. For calculations involving these matrices, the packages specialize in different ways: Kwant is intended for transport calculations with scattering systems while Pybinding targets large finite-sized and periodic systems in 1 to 3 dimensions. Pybinding can also be used to construct scattering systems, however it does not have a builtin solver for transport problems. This is where the *Kwant compatibility* layer comes in: it's possible to build a system in Pybinding and use Kwant's solvers for transport calculations. This combination takes advantage of Pybinding's much faster model builder.

The code used to obtain these results is available in the online documentation. You can download it and try it on your own computer. Usage instructions are located at the top of the script file.

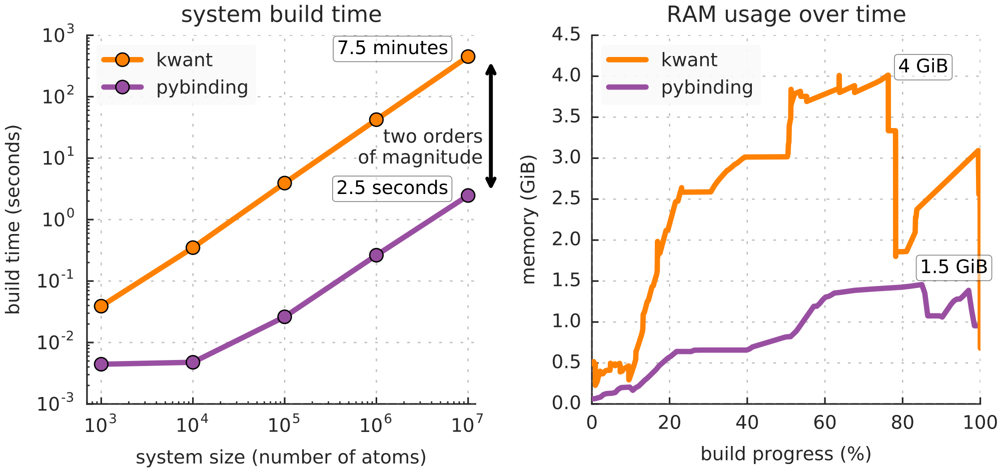


Fig. A.3.1: The results were measured for pybinding v0.8.0 and kwant v1.2.2 using: Intel Core i7-4960HQ CPU, 16 GiB RAM, Python 3.5, macOS 10.11. The RAM usage was measured using memory_profiler v0.41.

The benchmark constructs a circular graphene flake with a pn-junction and a constant magnetic field. The system build time is measured from the start of the definition to the point where the Hamiltonian matrix is fully constructed (a sparse matrix is used in both cases).

Pybinding builds the Hamiltonian much faster than Kwant: by two orders of magnitude. The main reason for this is in the way the system shape and fields are implemented. Both Kwant and Pybinding take user-defined functions as parameters for model construction. Kwant calls these functions individually for each atom and hopping which is quite slow. Pybinding stores all atoms and hoppings in contiguous arrays and then calls the user-defined functions just once for the entire dataset. This takes advantage of vectorization and drastically improves performance. Similarly, the lower memory usage is achieved by using arrays and CSR matrices rather than linked lists and trees.

Please note that at the time of writing Pybinding v0.8 does lack certain system construction features compared to Kwant. Specifically, it is currently not possible to build heterostructures in Pybinding, but this will be resolved in the near future. New features will be added while maintaining good performance.

At first glance it may seem like system build time is not really relevant because it is only done once and then multiple calculations can be applied to the constructed system. However, every time a parameter is changed (like some field strength) the Hamiltonian matrix will need to be rebuilt. Even though Kwant does take this into account and only does a partial rebuild, Pybinding is still much faster and this is very apparent in transport calculations which sweep over some model parameter. Performance improvements of 3 to 4 times have been measured for such cases – more information and a direct comparison are available in the *Kwant compatibility* section of the online documentation.

A.4 Tutorial

This section will present the essential features of Pybinding with example code to get you started quickly. The tutorial assumes that you already have a basic understanding of tight-binding theory, the Python 3 programming language and at least part of the scientific Python stack (SciPy). But don't worry: while this tutorial will not specifically explain basic language and scientific Python concepts, they are presented in a straightforward way and will be easy to pick up on the fly.

Imports

This tutorial includes code snippets directly within the text to illustrate features. In order to save space and not distract from the main point, some common and repetitive code will be omitted (like import statements). It is assumed that the following lines precede any other code:

```
import pybinding as pb
import numpy as np
import matplotlib.pyplot as plt

pb.pltutils.use_style()
```

The `pb` alias is always used for importing Pybinding. This is similar to the common scientific package aliases: `np` and `plt`. These import conventions are used consistently in the tutorial.

The function `pb.pltutils.use_style()` applies Pybinding's default style settings for matplotlib. This is completely optional and only affects the esthetics of the generated figures.

Lattice

A `Lattice` object describes the unit cell of a crystal lattice. This includes the primitive vectors, positions of sublattice sites and hopping parameters which connect those sites. All of this structural information is used to build up a larger system by translation.

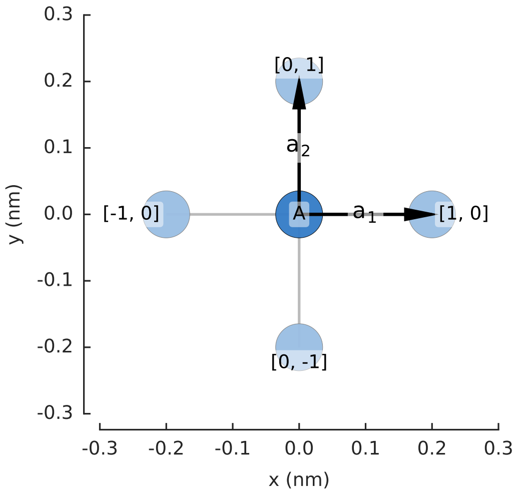
Square lattice

Starting from the basics, we'll create a simple square lattice.

```
import pybinding as pb

d = 0.2  # [nm] unit cell length
t = 1    # [eV] hopping energy

# create a simple 2D lattice with vectors a1 and a2
lattice = pb.Lattice(a1=[d, 0], a2=[0, d])
lattice.add_sublattices(
    ('A', [0, 0])  # add an atom called 'A' at position [0, 0]
)
lattice.add_hoppings(
    # (relative_index, from_sublattice, to_sublattice, energy)
    ([0, 1], 'A', 'A', t),
    ([1, 0], 'A', 'A', t)
)
```

It may not be immediately obvious what this code does. Fortunately, `Lattice` objects have a convenient `Lattice.plot()` method to easily visualize the constructed lattice.

```
lattice.plot()  # plot the lattice that was just constructed
plt.show()      # standard matplotlib show() function
```



In the figure we see lattice vectors a_1 and a_2 which were used to initialize `Lattice`. These vectors describe a Bravais lattice with an infinite set of positions,

$$\vec{R} = n_1\vec{a}_1 + n_2\vec{a}_2,$$

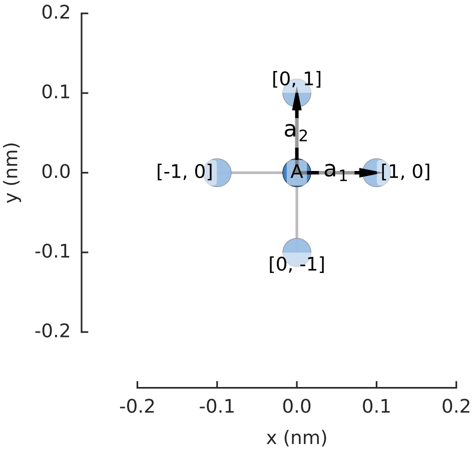
where n_1 and n_2 are integers. The blue circle labeled A represents the atom which was created with the `Lattice.add_sublattices()` method. The slightly faded out circles represent translations of the lattice in the primitive vector directions, i.e. using the integer index $[n_1, n_2]$.

The hoppings are specified using the `Lattice.add_hoppings()` method and each one consists of `(relative_index, from_sublattice, to_sublattice, energy)`:

- The main cell always has the index $[n_1, n_2] = [0, 0]$. The `relative_index` represents the number of integer steps needed to reach another cell starting from the main one. Each cell is labeled with its `relative_index`, as seen in the figure.
- A hopping is created between the main cell and a neighboring cell specified by `relative_index`. Two hoppings are added in the definition: [0, 1] and [1, 0]. The opposite hoppings [0, -1] and [-1, 0] are added automatically to maintain hermiticity.
- This lattice consists of only one sublattice so the `from` and `to` sublattice fields are trivial. Generally, `from_sublattice` indicates the sublattice in the [0, 0] cell and `to_sublattice` in the neighboring cell. This will be explained further in the next example.
- The last parameter is simply the value of the hopping energy.

It's good practice to build the lattice inside a function to make it easily reusable. Here we define the same lattice as before, but note that the unit cell length and hopping energy are function arguments, which makes the lattice easily configurable.

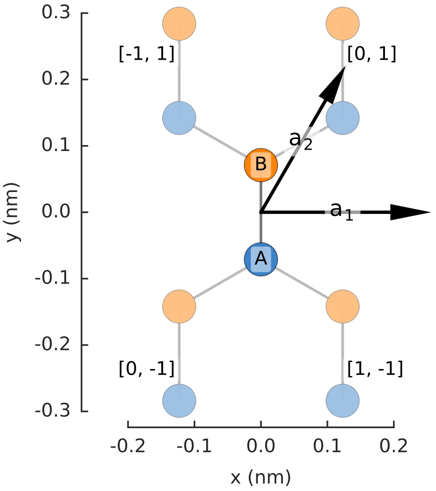
```
def square_lattice(d, t):
    lat = pb.Lattice(a1=[d, 0], a2=[0, d])
    lat.add_sublattices(('A', [0, 0]))
    lat.add_hoppings(([0, 1], 'A', 'A', t),
                     ([1, 0], 'A', 'A', t))
    return lat

# quickly set a shorter unit length `d`
lattice = square_lattice(d=0.1, t=1)
lattice.plot()
plt.show()
```



Graphene

The next example shows a slightly more complicated two-atom lattice of graphene.

```
def monolayer_graphene():
    a = 0.24595   # [nm] unit cell length
    a_cc = 0.142  # [nm] carbon-carbon distance
    t = -2.8      # [eV] nearest neighbour hopping

    lat = pb.Lattice(a1=[a, 0],
                     a2=[a/2, a/2 * sqrt(3)])
    lat.add_sublattices(('A', [0, -a_cc/2]),
                        ('B', [0,  a_cc/2]))
    lat.add_hoppings(
        # inside the main cell
        ([0,  0], 'A', 'B', t),
        # between neighboring cells
        ([1, -1], 'A', 'B', t),
        ([0, -1], 'A', 'B', t)
    )
    return lat

lattice = monolayer_graphene()
lattice.plot()
plt.show()
```



The `Lattice.add_sublattices()` method creates atoms A and B (blue and orange) at different offsets: $[0, -a_{cc}/2]$ and $[0, a_{cc}/2]$. Once again, the translated cells are given at positions $\vec{R} = n_1\vec{a}_1 + n_2\vec{a}_2$, however, this time the lattice vectors are not perpendicular which makes the integer indices $[n_1, n_2]$ slightly more complicate (see the labels in the figure).

The hoppings are defined as follows:

- `([0, 0], 'A', 'B', t)` specifies the hopping inside the main cell, from atom A to B. The main [0,0] cell is never labeled in the figure, but it is always the central cell where the lattice vectors originate.

- `([1, -1], 'A', 'B', t)` specifies the hopping between [0, 0] and [1, -1], from A to B. The opposite hopping is added automatically: [-1, 1], from B to A. In the tight-binding matrix representation, the opposite hopping is the Hermitian conjugate of the first one. The lattice specification always requires explicitly mentioning only one half of the hoppings while the other half is automatically added to guarantee hermiticity.

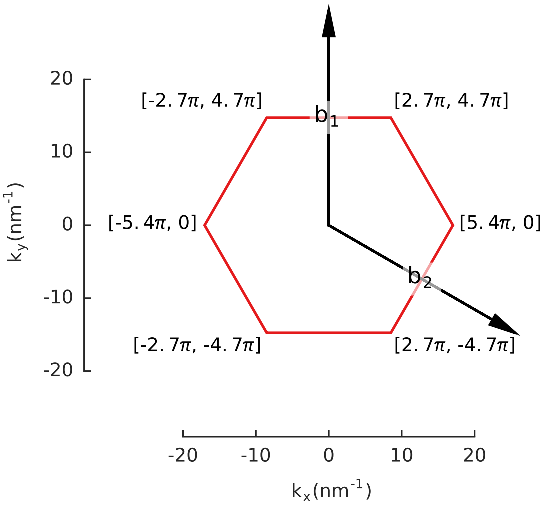
- `([0, -1], 'A', 'B', t)` is handled in the very same way.

The `Lattice.plot()` method will always faithfully draw any lattice that has been specified. It serves as a handy visual inspection tool.

Brillouin zone

The method `Lattice.plot_brillouin_zone()` is another handy tool that does just as its name implies.

```
lattice = monolayer_graphene()
lattice.plot_brillouin_zone()
```

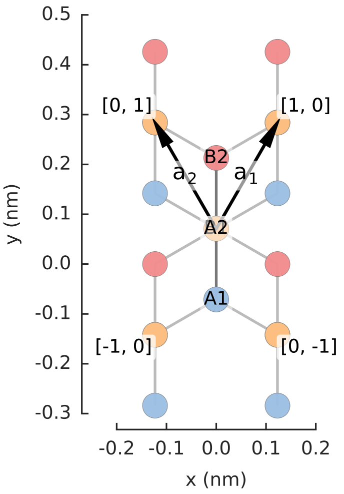


The reciprocal lattice vectors b_1 and b_2 are calculated automatically based on the real space vectors. There is no need to specify them manually. The first Brillouin zone is determined as the Wigner–Seitz cell in reciprocal space. By default, the plot method labels the vertices of the Brillouin zone.

Material repository

A few common lattices are included in Pybinding's *Material Repository*. More information is avalable in the online documentation. You can get started quickly by importing one of them. For example:

```
from pybinding.repository import graphene
lattice = graphene.bilayer()
lattice.plot()
```



Further reading

Additional features of the `Lattice` class are explained in the *Advanced Topics* section which is available online. It explains how to define supercells, add intrinsic onsite energy, remove dangling bonds, etc. Lattice specification examples for various other materials are also available in the online documentation.

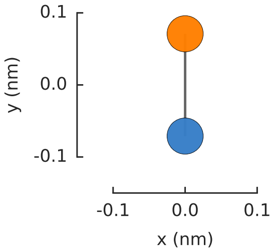
Band structure

In order to calculate the band structure of a crystal lattice, this section introduces the concepts of a `Model` and a `Solver`.

Model

A `Model` contains the full tight-binding description of the physical system that we wish to solve. We'll start by assigning a lattice to the model, and we'll use a pre-made one from the material repository.

```
from pybinding.repository import graphene

model = pb.Model(graphene.monolayer())
model.plot()
```



The result is not very exciting: just a single graphene unit cell, with 2 atoms and a single hopping between them. The model does not assume translational symmetry or any other physical property. Given a lattice it will just create a single unit cell. The model has a `System` attribute which keeps track of structural properties like the positions of lattice sites and the way they are connected, as seen in the figure above. The raw data can be accessed directly:

```
>>> model.system.x
[0, 0]
>>> model.system.y
[-0.071  0.071]
>>> model.system.sublattices
[0 1]
```

Each attribute is a 1D array where the number of elements is equal to the total number of lattice sites in the system. The model also has a `hamiltonian` attribute:

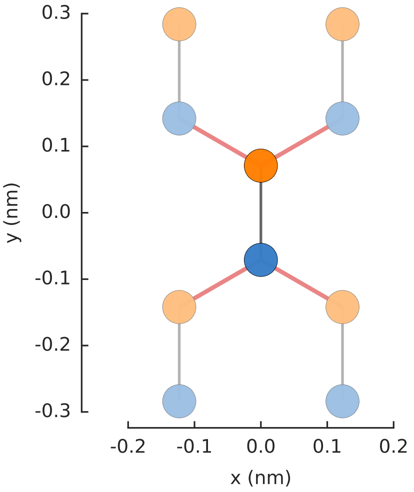
```
>>> model.hamiltonian
(0, 1)   -2.8
(1, 0)   -2.8
```

It's a sparse matrix (see `scipy.sparse.csr_matrix`) which corresponds to the tight-binding Hamiltonian of our model. The output above shows the default sparse representation of the data where each line corresponds to `(row, col) value`. Alternatively, we can see the dense matrix output:

```
>>> model.hamiltonian.todense()
[[ 0.0 -2.8]
 [-2.8  0.0]]
```

Next, we include `translational_symmetry()` to create an infinite graphene sheet.

```
model = pb.Model(
    graphene.monolayer(),
    pb.translational_symmetry()
)
model.plot()
```



The red lines indicate hoppings on periodic boundaries. The lighter colored circles represent the translations of the unit cell. The number of translations is infinite, but the plot only presents the first one in each lattice vector direction.

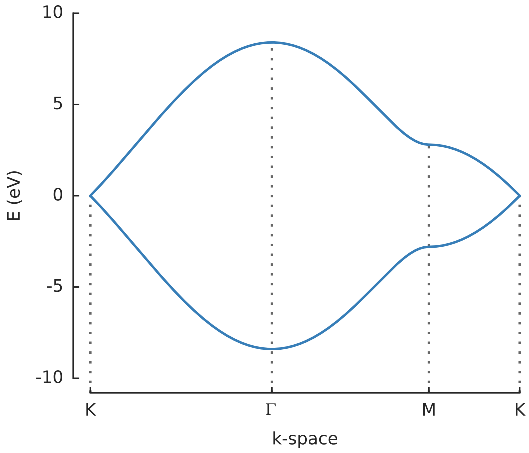
Solver

A `Solver` can exactly calculate the eigenvalues and eigenvectors of a Hamiltonian matrix. We'll take a look at various *Eigensolvers* and their capabilities in a later section, but right now we'll just grab the `lapack()` solver which is the simplest and most appropriate for small systems.

```
>>> model = pb.Model(graphene.monolayer())
>>> solver = pb.solver.lapack(model)
>>> solver.eigenvalues
[-2.8 2.8]
>>> solver.eigenvectors
[[-0.707 -0.707]
 [-0.707  0.707]]
```

Beyond just the `eigenvalues` and `eigenvectors` properties, `Solver` has a convenient `calc_bands()` method which can be used to calculate the band structure of our model.

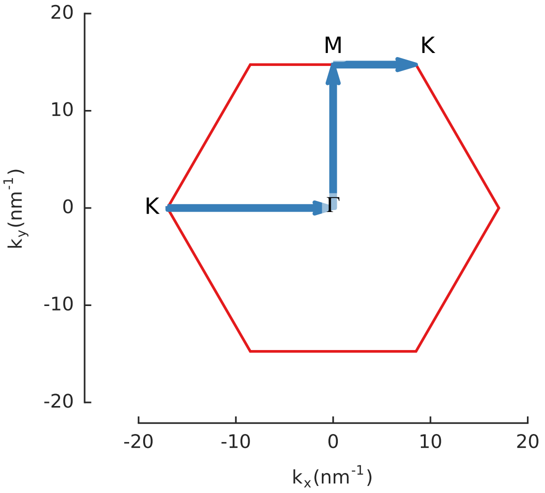
```
from math import sqrt, pi

model = pb.Model(
    graphene.monolayer(),
    pb.translational_symmetry()
)
solver = pb.solver.lapack(model)

a_cc = graphene.a_cc
Gamma = [0, 0]
K1 = [-4*pi / (3*sqrt(3)*a_cc), 0]
M = [0, 2*pi / (3*a_cc)]
K2 = [2*pi / (3*sqrt(3)*a_cc),
      2*pi / (3*a_cc)]

bands = solver.calc_bands(K1, Gamma, M, K2)
bands.plot(point_labels=['K', r'$\Gamma$',
                         'M', 'K'])
```



The points Γ, K and M are used to draw a path in the reciprocal space of graphene's Brillouin zone and `Solver.calc_bands()` calculates the band energy along that path. The return value of the method is a `Bands` result object.

All result objects have built-in plotting methods. Aside from the basic `plot()` seen above, `Bands` also has `plot_kpath()` which presents the path in reciprocal space. Plots can easily be composed, so to see the path in the context of the Brillouin zone, we can simply plot both:

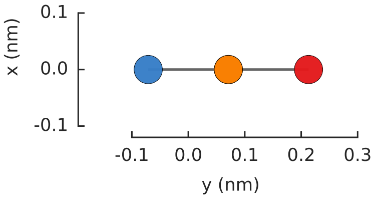
```
model.lattice.plot_brillouin_zone()
bands.plot_kpath(
    point_labels=['K', r'$\Gamma$',
                  'M', 'K']
)
```



Switching lattices

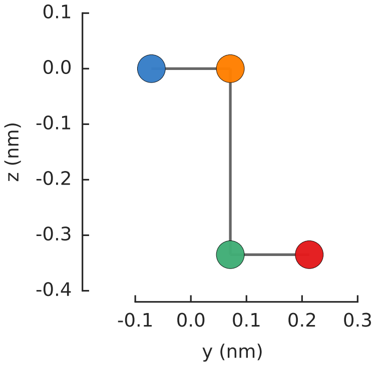
We can easily switch to a different material, just by passing a different lattice to the model. For this example, we'll use our pre-made `graphene.bilayer()` from the *Material Repository*. But you can create any lattice as described in the previously in the *Lattice* section.

```
model = pb.Model(graphene.bilayer())
model.plot(axes='yx')
```



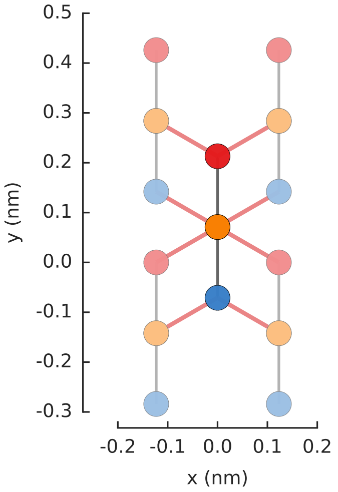
Without translational_symmetry(), the model is just a single unit cell with 4 atoms. Our bilayer lattice uses AB-stacking where a pair of atoms are positioned one on top of the another. By default, the Model.plot() method shows the xy-plane, so one of the bottom atoms isn't visible. We can pass an additional plot argument to see the yz-plane:

```
model = pb.Model(graphene.bilayer())
model.plot(axes='yz')
```



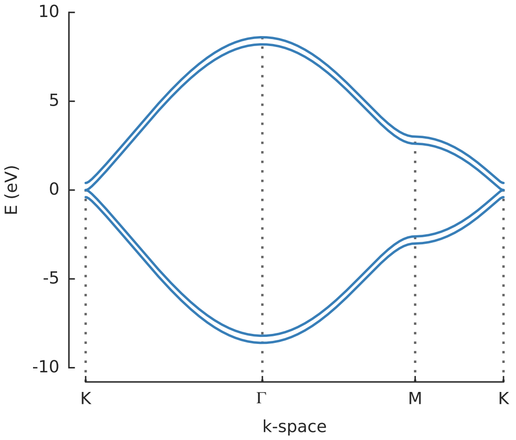
To compute the band structure, we'll need to include translational_symmetry().

```
model = pb.Model(
    graphene.bilayer(),
    pb.translational_symmetry()
)
model.plot()
```



As before, the red hoppings indicate periodic boundaries and the lighter colored circles represent the first of an infinite number of translation units. We'll compute the band structure for the same Γ, K and M points as monolayer graphene:

```
solver = pb.solver.lapack(model)
bands = solver.calc_bands(K1, Gamma, M, K2)
bands.plot(point_labels=['K', r'$\Gamma$',
                         'M', 'K'])
```

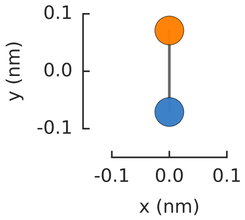


More band structure calculation examples are available online. Eigensolvers will be covered in more detail at a later point in the tutorial, but this is enough information to get started. The next few sections are going to be dedicated to model building.

Finite size

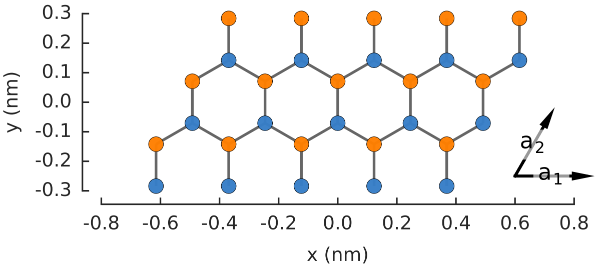
This section introduces the concept of shapes with classes `Polygon` and `FreeformShape` which are used to model systems of finite size. The simplest finite-sized system is just the unit cell of the crystal lattice.

```
from pybinding.repository import graphene

model = pb.Model(graphene.monolayer())
model.plot()
```



The unit cell can also be replicated a number of times to create a bigger system.

```
model = pb.Model(
    graphene.monolayer(),
    pb.primitive(a1=5, a2=3)
)
model.plot()
model.lattice.plot_vectors()
```

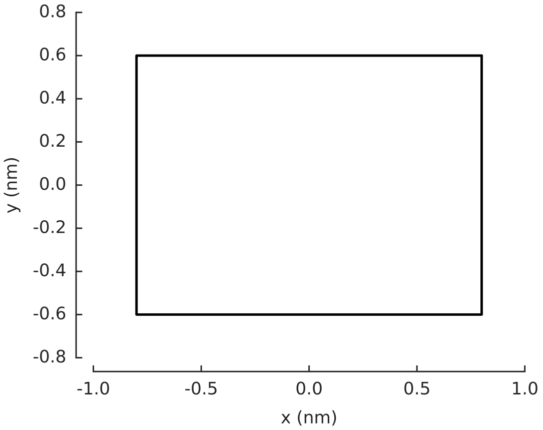


The `primitive()` parameter tells the model to replicate the unit cell 5 times in the a_1 vector direction and 3 times in the a_2 direction. However, to model realistic systems we need proper shapes.

Polygon

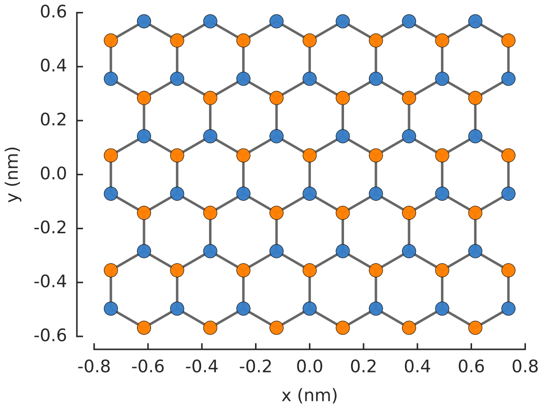
The easiest way to create a 2D shape is with the `Polygon` class. For example, a simple rectangle:

```
def rectangle(width, height):
    x0 = width / 2
    y0 = height / 2
    return pb.Polygon([[ x0,  y0],
                       [ x0, -y0],
                       [-x0, -y0],
                       [-x0,  y0]])

shape = rectangle(1.6, 1.2)
shape.plot()
```



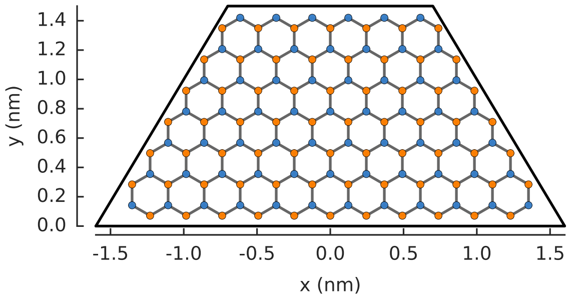
A `Polygon` is initialized with a list of vertices which should be given in clockwise or counterclockwise order. When added to a `Model` the lattice will expand to fill the shape.

```
model = pb.Model(
    graphene.monolayer(),
    rectangle(width=1.6, height=1.2)
)
model.plot()
```



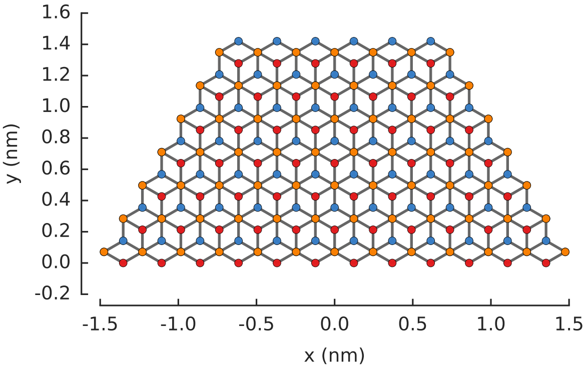
To help visualize the shape and the expanded lattice, the polygon outline can be plotted on top of the system by calling both plot methods one after another.

```
def trapezoid(a, b, h):
    return pb.Polygon([[-a/2, 0],
                       [-b/2, h],
                       [ b/2, h],
                       [ a/2, 0]])

model = pb.Model(
    graphene.monolayer(),
    trapezoid(a=3.2, b=1.4, h=1.5)
)
model.plot()
model.shape.plot()
```



In general, a shape does not depend on a specific material, so it can be easily reused. Here, we shall switch to a `graphene.bilayer()` lattice, but we'll keep the same `trapezoid` shape as defined earlier:

```
model = pb.Model(
    graphene.bilayer(),
    trapezoid(a=3.2, b=1.4, h=1.5)
)
model.plot()
```

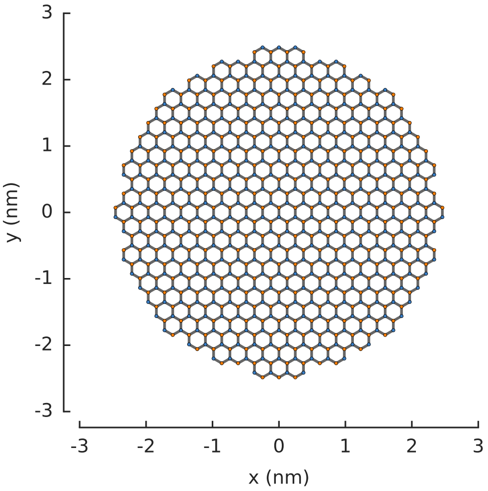


Freeform shape

Unlike a `Polygon` which is defined by a list of vertices, a `FreeformShape` is defined by a `contains` function which determines if a lattice site is inside the desired shape.

```
def circle(radius):
    def contains(x, y, z):
        return np.sqrt(x**2 + y**2) < radius
    return pb.FreeformShape(contains, width=[2*radius, 2*radius])
```

```
model = pb.Model(
    graphene.monolayer(),
    circle(radius=2.5)
)
model.plot()
```

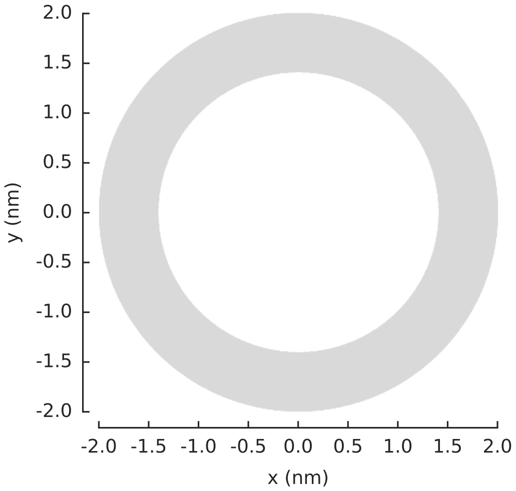


The `width` parameter of `FreeformShape` specifies the bounding box width. Only sites inside the bounding box will be considered for the shape. It's like carving a sculpture from a block of stone. The bounding box can be thought of as the stone block, while the `contains` function is the carving tool that can give the fine detail of the shape.

As with `Polygon`, we can visualize the shape with the `FreeformShape.plot()` method.

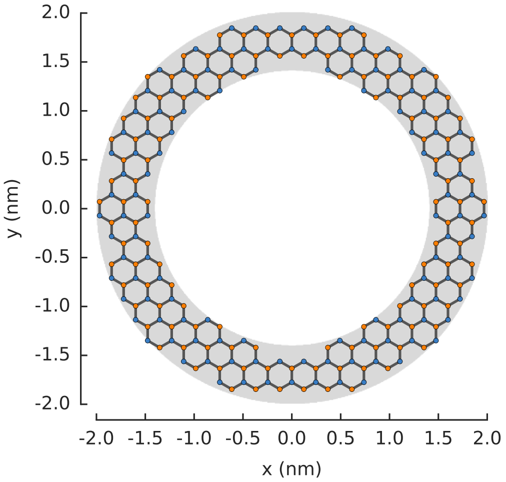
```
def ring(inner_radius, outer_radius):
    def contains(x, y, z):
        r = np.sqrt(x**2 + y**2)
        return np.logical_and(inner_radius < r, r < outer_radius)
    return pb.FreeformShape(contains, width=[2*outer_radius, 2*outer_radius])
```

```
shape = ring(inner_radius=1.4,
             outer_radius=2)
shape.plot()
```



The shaded area indicates the shape as determined by the `contains` function. Creating a model will cause the lattice to fill in the shape.

```
model = pb.Model(
    graphene.monolayer(),
    ring(inner_radius=1.4, outer_radius=2)
)
model.plot()
model.shape.plot()
```

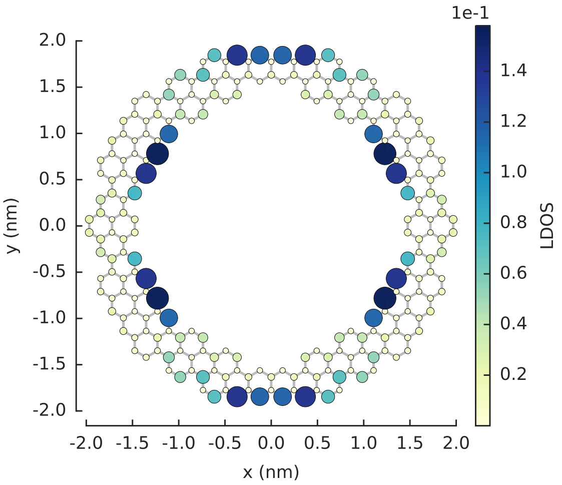


Spatial LDOS

Now that we have a ring structure, we can exactly diagonalize its `model.hamiltonian` using a `Solver`. We previously used the `lapack()` solver to find all the eigenvalues and eigenvectors, but this is not efficient for larger systems. The sparse `arpack()` solver can calculate a targeted subset of the eigenvalues, which is usually desired and much faster.

In this case, we are interested only in the 20 lowest energy states.

```
model = pb.Model(
    graphene.monolayer(),
    ring(inner_radius=1.4, outer_radius=2)
)
# only the 20 lowest eigenstates
solver = pb.solver.arpack(model, k=20)

ldos = solver.calc_spatial_ldos(energy=0,␣
↪broadening=0.05)  # eV
ldos.plot(site_radius=(0.03, 0.12))
pb.pltutils.colorbar(label="LDOS")
```



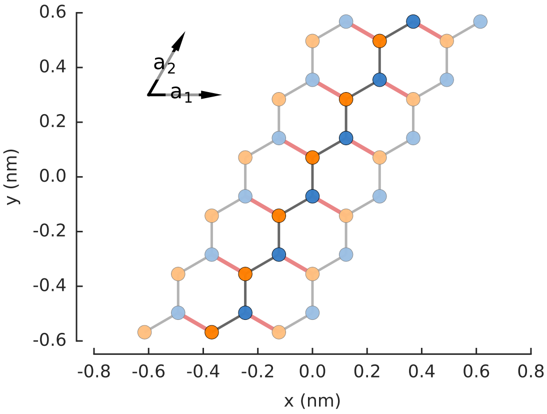
The convenient `Solver.calc_spatial_ldos()` method calculates the local density of states (LDOS) at every site for the given energy with a Gaussian broadening. The returned object is a `StructureMap` which holds the LDOS data. The `StructureMap.plot()` method will produce a figure similar to `Model.plot()`, but with a colormap indicating the LDOS value at each lattice site. In addition, the `site_radius` argument specifies a range of sizes which will cause the low intensity sites to appear as small circles while high intensity ones become large. The states with a high LDOS are clearly visible on the outer and inner edges of the graphene ring structure.

Shape and symmetry

The last two sections showed how to model shape and symmetry individually, but we can be more creative and combine the two.

To create a graphene nanoribbon, we'll need a shape to give the finite width of the ribbon while the infinite length is achieved by imposing translational symmetry.

```
from pybinding.repository import graphene

model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(1.2),  # nm
    pb.translational_symmetry(a1=True,␣
↪a2=False)
)
model.plot()
model.lattice.plot_vectors(
    position=[-0.6, 0.3]  # nm
)
```

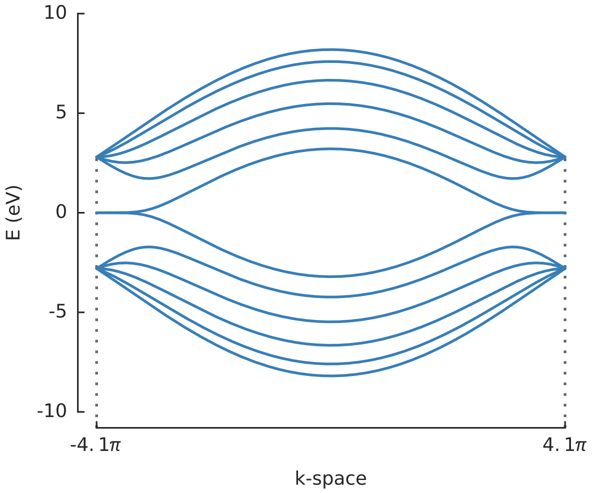


As before, the central darker circles represent the main cell of the nanoribbon, the lighter colored circles are the translations due to symmetry and the red lines are boundary hoppings. The two arrows in the upper left corner show the primitive lattice vectors of graphene.

The `translational_symmetry()` is applied only in the a_1 lattice vector direction which gives the ribbon its infinite length, but the symmetry is disabled in the a_2 direction so that the finite size of the shape is preserved. The builtin `rectangle()` shape gives the nanoribbon its 1.2 nm width.

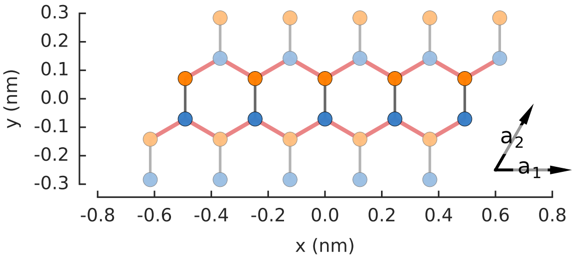
The band structure calculations work just as before.

```
from math import pi, sqrt

solver = pb.solver.lapack(model)
a = graphene.a_cc * sqrt(3)
bands = solver.calc_bands(-pi/a, pi/a)
bands.plot()
```



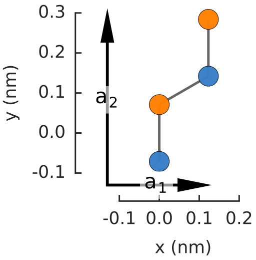
This is the characteristic band structure for zigzag nanoribbons with zero-energy edge states. If we change the direction of the translational symmetry to a_2, the orientation will change, but we will still have a zigzag nanoribbon.

```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(1.2),  # nm
    pb.translational_symmetry(a1=False,␣
↪a2=True)
)
model.plot()
model.lattice.plot_vectors(
    position=[0.6, -0.25]  # nm
)
```



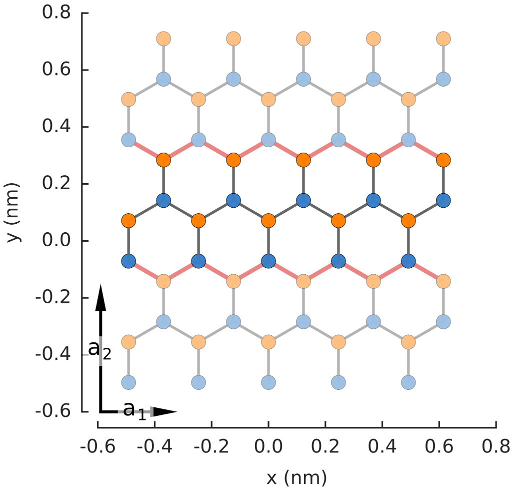
Because of the nature of graphene's 2-atom unit cell and lattice vector, only zigzag edges can be created. In order to create armchair edges, we must introduce a different unit cell with 4 atoms.

```
model = pb.Model(graphene.monolayer_4atom())
model.plot()
model.lattice.plot_vectors(position=[-0.13, -0.13])
```



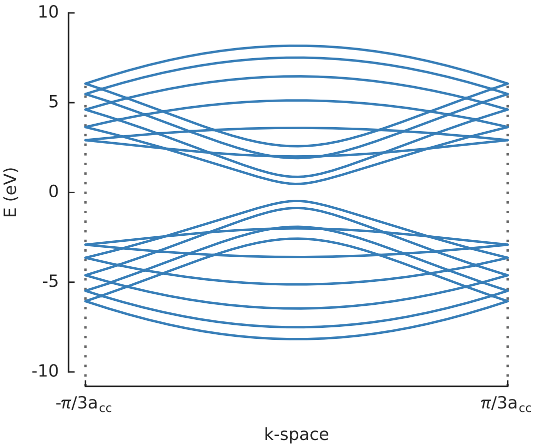
Note that the lattice vectors a_1 and a_2 are at a right angle, unlike the sharp angle of the base 2-atom cell. The lattice properties are identical for the 2 and 4 atom cells, but the new geometry helps to create armchair edges.

```
model = pb.Model(
    graphene.monolayer_4atom(),
    pb.primitive(a1=5),
    pb.translational_symmetry(a1=False,␣
↪a2=True)
)
model.plot()
model.lattice.plot_vectors(
    position=[-0.59, -0.6]
)
```



To calculate the band structure we must enter at least two points in k-space between which the energy will be calculated. Note that because the periodicity is in the direction of the second lattice vector a_2, the points in k-space are given as `[0, pi/d]` instead of just `pi/d` (which would be equivalent to `[pi/d, 0]`).

```
solver = pb.solver.lapack(model)
d = 3 * graphene.a_cc
bands = solver.calc_bands([0, -pi/d],
                          [0,  pi/d])
bands.plot(point_labels=['$-\pi/3 a_{cc}$',
                         '$\ pi/3 a_{cc}$'])
```



Fields and effects

This section will introduce `@onsite_energy_modifier` and `@hopping_energy_modifier` which can be used to add various fields to the model. These functions can apply user-defined modifications to the Hamiltonian matrix which is why we shall refer to them as *modifier* functions.

Electric potential

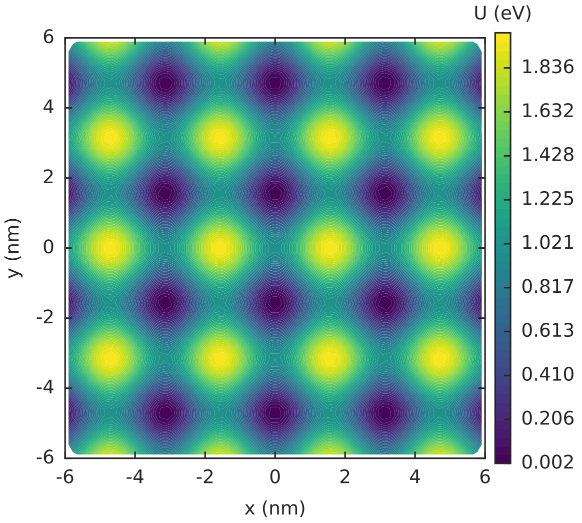
We can define a simple potential function like the following:

```python
@pb.onsite_energy_modifier
def potential(x, y):
    return np.sin(x)**2 + np.cos(y)**2
```

Here `potential` is just a regular Python function, but we attached a pretty `@` decorator to it. The `@onsite_energy_modifier` decorator gives an ordinary function a few extra properties which we'll talk about later. For now, just keep in mind that this is required to mark a function as a *modifier* for use with Pybinding models. The `x` and `y` arguments are lattice site positions and the return value is the desired potential. Note the use of `np.sin` instead of `math.sin`. The `x` and `y` coordinates are `numpy` arrays, not individual numbers. This is true for all modifier arguments in Pybinding. When you write modifier functions, make sure to always use `numpy` operations which work with arrays, unlike regular `math`.

To use the potential function, just place it in a `Model` parameter list. To visualize the potential, there's the handy `Model.onsite_map` property which is a `StructureMap` of the onsite energy of the Hamiltonian matrix.

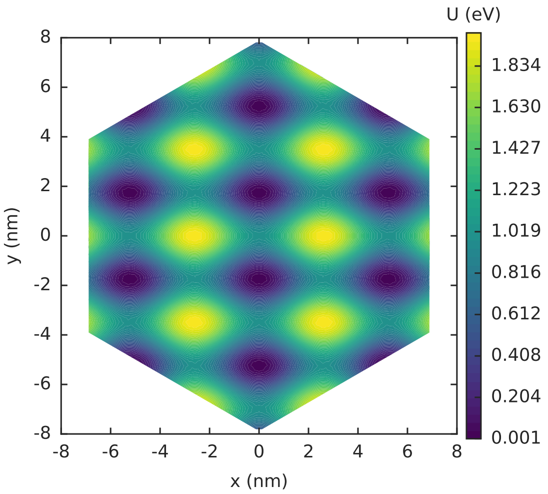
```python
from pybinding.repository import graphene

model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(12),
    potential
)

model.onsite_map.plot_contourf()
pb.pltutils.colorbar(label="U (eV)")
```



The figure shows a 2D colormap representation of our wavy potential in a square system. The `StructureMap.plot_contourf()` method we just called is implemented in terms of matplotlib's `contourf` function with some slight adjustments for convenience.

To make the potential more flexible, it's a good idea to enclose it in an outer function, just like this:

```
def wavy(a, b):
    @pb.onsite_energy_modifier
    def potential(x, y):
        return np.sin(a * x)**2
            + np.cos(b * y)**2
    return potential

model = pb.Model(
    graphene.monolayer(),
    pb.regular_polygon(num_sides=6,
                       radius=8),
    wavy(a=0.6, b=0.9)
)
model.onsite_map.plot_contourf()
pb.pltutils.colorbar(label="U (eV)")
```



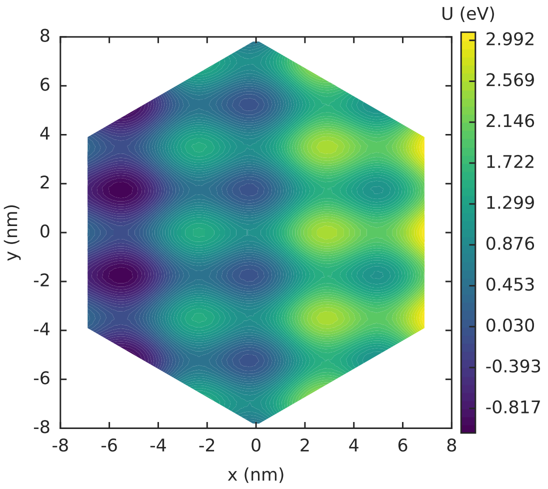
Note that we are using a system with hexagonal shape this time (via regular_polygon()). The potential is only plotted inside the area of the actual system.

We can make one more improvement to our wavy function. We'll add an energy argument:

```
def wavy2(a, b):
    @pb.onsite_energy_modifier
    def potential(energy, x, y):
        v = np.sin(a * x)**2 + np.cos(b * y)**2
        return energy + v
    return potential
```

The energy argument contains the existing onsite energy in the system before the new potential function is applied. By adding to the existing energy, instead of just setting it, we can compose multiple functions. For example, let's combine the improved wavy2 with a linear potential.

```
def linear(k):
    @pb.onsite_energy_modifier
    def potential(energy, x):
        return energy + k*x
    return potential

model = pb.Model(
    graphene.monolayer(),
    pb.regular_polygon(num_sides=6,
                       radius=8),
    wavy2(a=0.6, b=0.9),
    linear(k=0.2)
)
model.onsite_map.plot_contourf()
pb.pltutils.colorbar(label="U (eV)")
```



We see a similar wavy pattern as before, but the magnitude increases linearly along the x-axis because of the contribution of the `linear` potential.

About the decorator

Now that you have a general idea of how to add and compose electric potentials in a model, we should talk about the role of the `@onsite_energy_modifier`. The full signature of a potential function looks like this:

```
@pb.onsite_energy_modifier
def potential(energy, x, y, z, sub_id):
    return ... # some function of the arguments
```

This function uses all of the possible arguments of an onsite energy modifier: `energy`, `x`, `y`, `z` and `sub_id`. We have already explained the first three. The `z` argument is, obviously, the z-axis coordinate of the lattice sites. The `sub_id` argument tells us which sublattice a site belongs to. Its usage will be explained below.

As we have seen before, we don't actually need to define a function to take all the arguments. They are optional. The `@` decorator will recognize a function which takes any of these arguments and it will adapt it for use in a Pybinding model. Previously, the `linear` function accepted only the `energy` and `x` arguments, but `wavy` also included the `y` argument. The order of arguments is not important, only their names are. Therefore, this is also a valid modifier:

```
@pb.onsite_energy_modifier
def potential(x, y, energy, sub_id):
    return ... # some function
```

But the argument names must be exact: a typo or an extra unknown argument will result in an error. The decorator checks this at definition time and decides if the given function is a valid modifier or not, so any errors will be caught early.

Opening a band gap

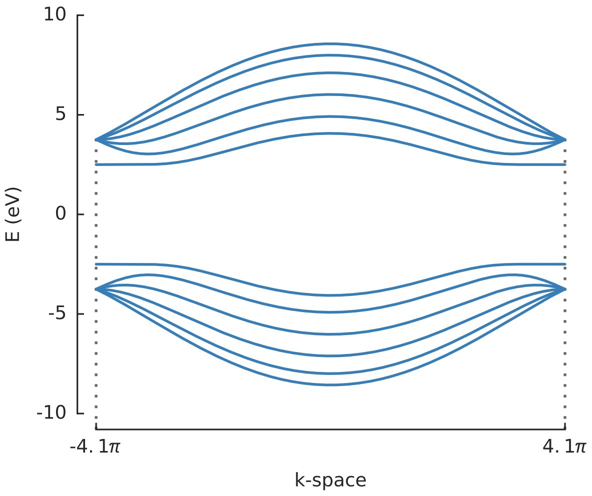
The last thing to explain about `@onsite_energy_modifier` is the use of the `sub_id` argument. It tells us which sublattice a site belongs to. If you remember from early on in the tutorial, *in the process of specifying a lattice*, we gave each sublattice a unique name. This name can be used to filter out sites of a specific sublattice. For example, let's add mass to electrons in graphene:

```
def mass_term(delta):
    """Break sublattice symmetry with opposite A and B onsite energy"""
    @pb.onsite_energy_modifier
    def potential(energy, sub_id):
        energy[sub_id == 'A'] += delta
        energy[sub_id == 'B'] -= delta
        return energy
    return potential
```

Note that we don't need `x`, `y` or `z` arguments because this will be applied everywhere evenly. The `mass_term` function will add an energy `delta` to all sites on sublattice `A` and subtract `delta` from all `B` sites. Note that we are indexing the `energy` array with a condition on the `sub_id` array of the same length. This is a standard `numpy` indexing technique which you should be familiar with.

A simple way to demonstrate our new `mass_term` is with a graphene nanoribbon.

```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(1.2),
    pb.translational_symmetry(a1=True, a2=False),
    mass_term(delta=2.5)  # eV
)
solver = pb.solver.lapack(model)
bands = solver.calc_bands(-pi/a, pi/a)
bands.plot()
```



We set a very high `delta` value of 2.5 eV for illustration purposes. Indeed, a band gap of 5 eV (`delta * 2`) is quite clearly visible in the band structure.

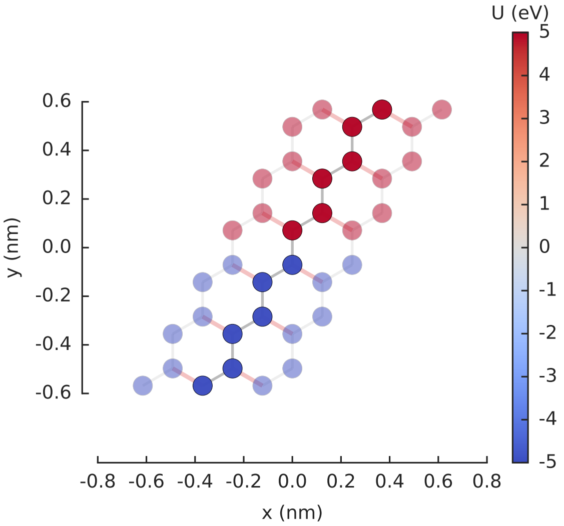
PN junction

While we're working with a nanoribbon, let's add a PN junction along its main axis.

```
def pn_junction(y0, v1, v2):
    @pb.onsite_energy_modifier
    def potential(energy, y):
        energy[y < y0] += v1
        energy[y >= y0] += v2
        return energy
    return potential
```

The `y0` argument is the position of the junction, while `v1` and `v2` are the values of the potential (in eV) before and after the junction. Let's add it to the nanoribbon:

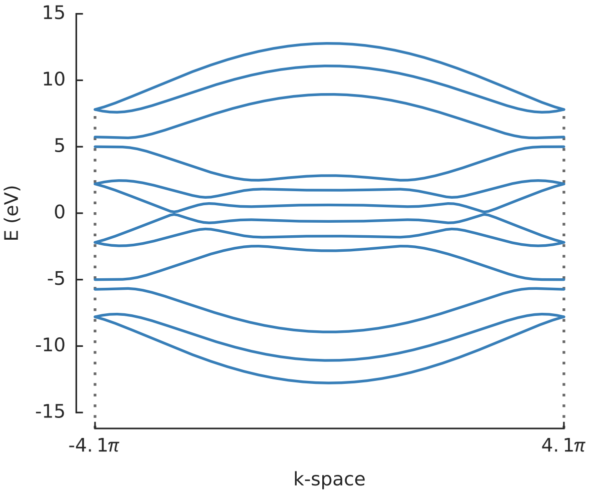
```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(1.2),
    pb.translational_symmetry(a1=True,␣
↪a2=False),
    pn_junction(y0=0, v1=-5, v2=5)
)
model.onsite_map.plot(cmap="coolwarm",
                      site_radius=0.04)
pb.pltutils.colorbar(label="U (eV)")
```



Remember that the `Model.onsite_map` property is a `StructureMap`, which has several plotting methods. A contour plot would not look at all good for such a small nanoribbon, but the method `StructureMap.plot()` is perfect. As before, the ribbon has infinite length along the x-axis and the transparent sites represent the periodic boundaries. The PN junction splits the ribbon in half along its main axis.

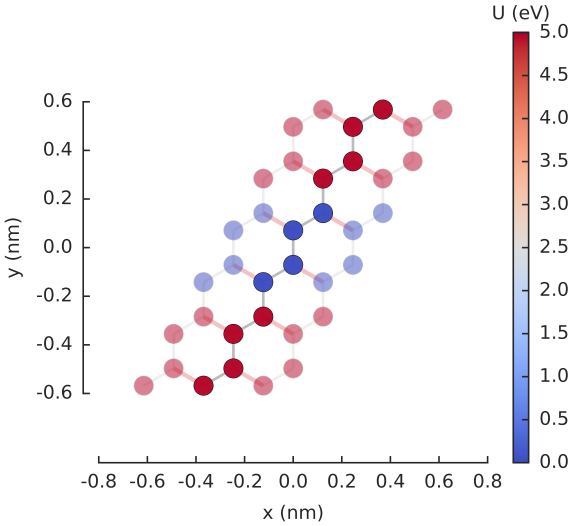
We can compute and plot the band structure:

```
solver = pb.solver.lapack(model)
bands = solver.calc_bands(-pi/a, pi/a)
bands.plot()
```



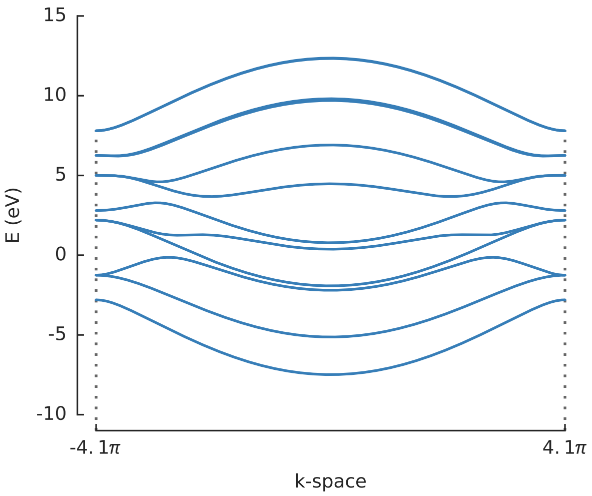
Next, let's create a square potential well. We could define a new modifier function, as before. But lets take a different approach and create the well by composing two PN junctions.

```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(1.2),
    pb.translational_symmetry(a1=True,␣
↪a2=False),
    pn_junction(y0=-0.2, v1=5, v2=0),
    pn_junction(y0=0.2, v1=0, v2=5)
)
model.onsite_map.plot(cmap="coolwarm",
                      site_radius=0.04)
pb.pltutils.colorbar(label="U (eV)")
```



It works as expected. This can sometimes be a nice and quick way to extend a model. The square well affects the band structure by breaking electron-hole symmetry:

```
solver = pb.solver.lapack(model)
bands = solver.calc_bands(-pi/a, pi/a)
bands.plot()
```



Magnetic field

To model a magnetic field, we need to apply the Peierls substitution:

$$t_{nm} \rightarrow t_{nm} \mathrm{e}^{i \frac{2\pi}{\Phi_0} \int_n^m \vec{A}_{nm} \cdot d\vec{l}}$$

Here t_{nm} is the hopping energy between two sites, $\Phi_0 = h/e$ is the magnetic quantum, h is the Planck constant and $\vec{A}_{nm}$ is the magnetic vector potential along the path between sites n and m. We want the magnetic field to be perpendicular to the graphene plane, so we can take the gauge $\vec{A}(x, y, z) = (By, 0, 0)$.

This can all be expressed with a `@hopping_energy_modifier`:

```python
from pybinding.constants import phi0

def constant_magnetic_field(B):
    @pb.hopping_energy_modifier
    def function(energy, x1, y1, x2, y2):
        # the midpoint between two sites
        y = 0.5 * (y1 + y2)
        # scale from nanometers to meters
        y *= 1e-9

        # vector potential along the x-axis
        A_x = B * y

        # integral of (A * dl) from position 1 to position 2
        peierls = A_x * (x1 - x2)
        # scale from nanometers to meters (because of x1 and x2)
        peierls *= 1e-9

        # the Peierls substitution
        return energy * np.exp(1j * 2*pi/phi0 * peierls)
    return function
```

The `energy` argument is the existing hopping energy between two sites at coordinates (`x1`, `y1`) and (`x2`, `y2`). The function computes and returns the Peierls substitution as given by the equation above.

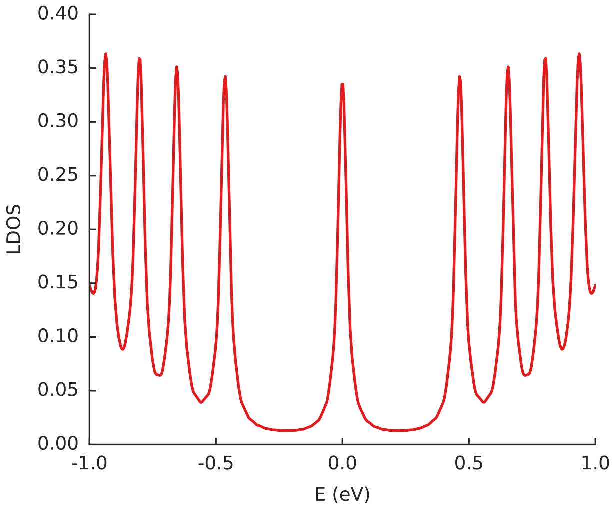
The full signature of a `@hopping_energy_modifier` is actually:

```python
@pb.hopping_energy_modifier
def function(energy, x1, y1, z1, x2, y2, z2, hop_id):
    return ... # some function of the arguments
```

The `hop_id` argument tells us which type of hopping it is. Hopping types can be specifically named during the creation of a lattice. This can be used to apply functions only to specific hoppings. However, as with all the modifier arguments, it's optional, so we only take what we need.

To test out our `constant_magnetic_field`, we'll calculate the local density of states (LDOS), where we expect to see peaks corresponding to Landau levels. The computation method used here is explained in detail in the *Green's function* section of the tutorial.

```python
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(20),
    constant_magnetic_field(B=200)  # Tesla
)
greens = pb.greens.kpm(model)

ldos = greens.calc_ldos(energy=np.linspace(-1, 1, 500), broadening=0.02, position=[0, 0])
ldos.plot()
plt.show()
```



The values of the magnetic field is exaggerated here (200 Tesla), but that is done to keep the computation time low for the tutorial (under 1 second for this LDOS calculation).

Defects and strain

This section will introduce `@site_state_modifier` and `@site_position_modifier` which can be used to add defects and strain to the model. These modifiers are applied to the structure of the system before the Hamiltonian matrix is created.

Vacancies

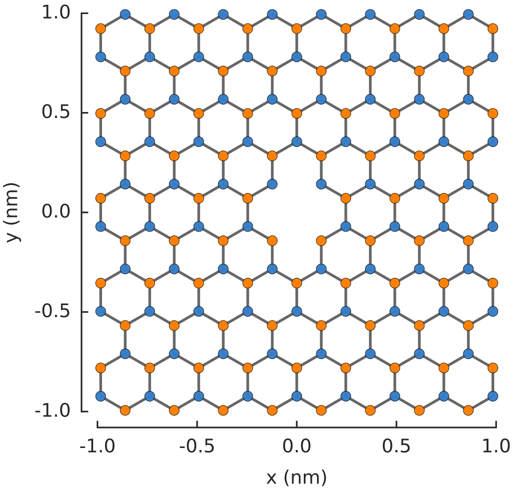
A `@site_state_modifier` can be used to create vacancies in a crystal lattice. The definition is very similar to the onsite and hopping modifiers explained in the previous section.

```
def vacancy(position, radius):
    @pb.site_state_modifier
    def modifier(state, x, y):
        x0, y0 = position
        state[(x-x0)**2 + (y-y0)**2 < radius**2] = False
        return state
    return modifier
```

The `state` argument indicates the current boolean state of a lattice site. Only valid sites (`True` state) will be included in the final Hamiltonian matrix. Therefore, setting the state of sites within a small radius to `False` will exclude them from the final system. The `x` and `y` arguments are lattice site positions. As with the other modifiers, the arguments are optional (`z` is not needed for this example) but the full signature of the site state modifier can be found on its `API reference page`.

This is actually very similar to the way a `FreeformShape` works. In fact, it is possible to create defects by defining them directly in the shape. However, such an approach would not be very flexible since we would need to create an entire new shape in order to change either the vacancy type or the shape itself. By defining the vacancy as a modifier, we can simply compose it with any existing shapes:

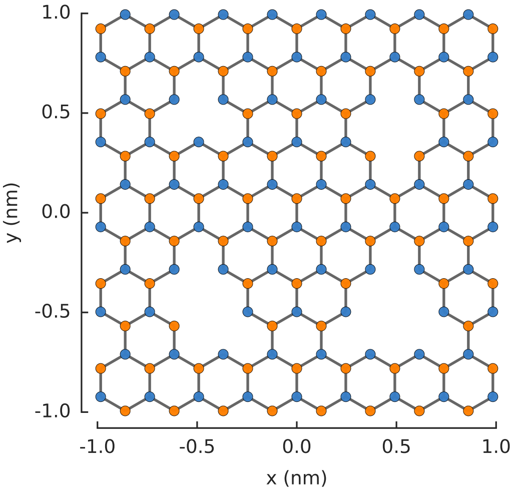
```
from pybinding.repository import graphene

model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(2),
    vacancy(position=[0, 0], radius=0.1)
)
model.plot()
```



The resulting 2-atom vacancy is visible in the center of the system. The two vacant sites are completely removed from the final Hamiltonian matrix. If we were to inspect the number of rows and columns by looking up `model.hamiltonian.shape`, we would see that the size of the matrix is reduced by 2.

Any number of modifiers can be included in the model and they will compose as expected. We can take advantage of this and create four different vacancies, with 1 to 4 missing atoms:

```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(2),
    vacancy([-0.50,  0.50], radius=0.1),
    vacancy([ 0.50,  0.45], radius=0.15),
    vacancy([-0.45, -0.45], radius=0.15),
    vacancy([ 0.50, -0.50], radius=0.2),
)
model.plot()
```

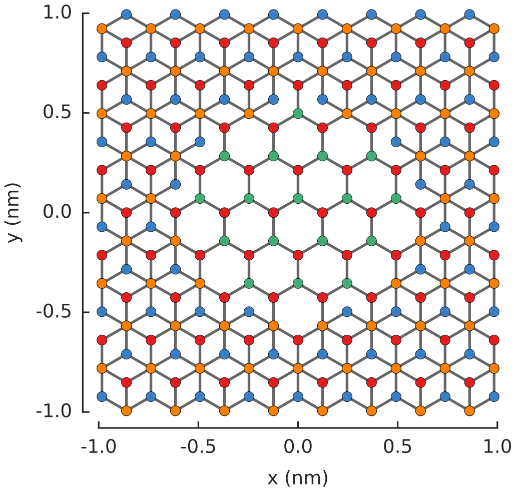


Layer defect

The site state modifier also has access to sublattice information. This can be used, for example, with bilayer graphene to remove a single layer in a specific area. We'll use the bilayer lattice that's included in the `graphene.bilayer()` lattice is laid out so that sublattices A1 and B1 belong to the top layer, while A2 and B2 are on the bottom.

```
def scrape_top_layer(position, radius):
    """Remove the top layer of graphene in the area specified by position and radius"""
    @pb.site_state_modifier
    def modifier(state, x, y, sub_id):
        x0, y0 = position
        is_within_radius = (x-x0)**2 + (y-y0)**2 < radius**2
        is_top_layer = np.logical_or(sub_id == 'A1', sub_id == 'B1')
        final_condition = np.logical_and(is_within_radius, is_top_layer)
        state[final_condition] = False
        return state
    return modifier
```

```
model = pb.Model(
    graphene.bilayer(),
    pb.rectangle(2),
    scrape_top_layer(position=[0, 0],
                     radius=0.5)
)
model.plot()
```



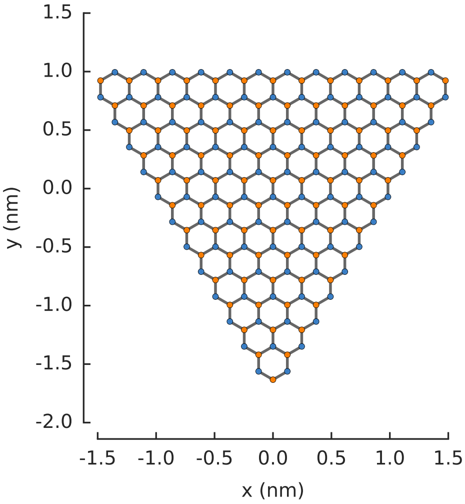
The central monolayer area is nicely visible in the figure. We can actually create the same structure in a different way: by considering the `z` position of the lattice site to distinguish the layers. An alternative modifier definition is given below. It would generate the same figure. Which method is more convenient is up to the user.

```
def scrape_top_layer_alt(position, radius):
    """Alternative definition of `scrape_top_layer`"""
    @pb.site_state_modifier
    def modifier(state, x, y, z):
        x0, y0 = position
        is_within_radius = (x-x0)**2 + (y-y0)**2 < radius**2
        is_top_layer = (z == 0)
        final_condition = np.logical_and(is_within_radius, is_top_layer)
        state[final_condition] = False
        return state
    return modifier
```

Strain

A `@site_position_modifier` can be used to model the lattice site displacement caused by strain. Let's start with a simple triangular system:

```
from math import pi

model = pb.Model(
    graphene.monolayer(),
    pb.regular_polygon(num_sides=3,␣
↪radius=2, angle=pi),
)
model.plot()
```

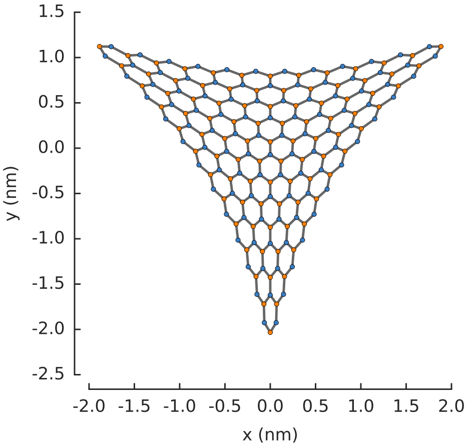


We're going to apply strain in three directions, as if we are pulling outward on the vertices of the triangle. The displacement function for this kind of strain is given below. The `c` parameter lets us control the intensity of the strain.

```
def triaxial_displacement(c):
    @pb.site_position_modifier
    def displacement(x, y, z):
        ux = 2*c * x*y
        uy = c * (x**2 - y**2)
        return x + ux, y + uy, z
    return displacement
```

The modifier function takes the `x`, `y`, `z` coordinates as arguments. The displacement `ux`, `uy` is computed and the modified coordinates are returned.

```
model = pb.Model(
    graphene.monolayer(),
    pb.regular_polygon(num_sides=3,␣
↪radius=2, angle=pi),
    triaxial_displacement(c=0.15)
)
model.plot()
```



As seen in the figure, the displacement has been applied to the lattice sites and the new position data is saved in the system. However, the hopping energies have not been modified yet. Every hopping element of the Hamiltonian matrix is equal to the hopping energy of pristine graphene:

```
>>> np.all(model.hamiltonian.data == -2.8)
True
```

We now need to use the new position data to modify the hopping energy according to the relation $t = t_0 e^{-\beta(\frac{d}{a_{cc}}-1)}$, where t_0 is the original unstrained hopping energy, β controls the strength of the strain-induced hopping modulation, d is the strained distance between two atoms and a_{cc} is the unstrained carbon-carbon distance. This can be implemented using a `@hopping_energy_modifier`:

```
@pb.hopping_energy_modifier
def strained_hopping(energy, x1, y1, z1, x2, y2, z2):
    d = np.sqrt((x1-x2)**2 + (y1-y2)**2 + (z1-z2)**2)
    beta = 3.37
    w = d / graphene.a_cc - 1
    return energy * np.exp(-beta*w)
```

The structural modifiers (site state and position) are always automatically applied to the model before energy modifiers (onsite and hopping). Thus, our `strain_hopping` modifier will get the new displaced coordinates as its arguments, from which it will calculate the strained hopping energy.

```
model = pb.Model(
    graphene.monolayer(),
    pb.regular_polygon(num_sides=3, radius=2, angle=pi),
    triaxial_displacement(c=0.15),
    strained_hopping
)
```

Including the hopping modifier along with the displacement will yield position dependent hopping energy, thus the elements of the Hamiltonian will no longer be all equal:

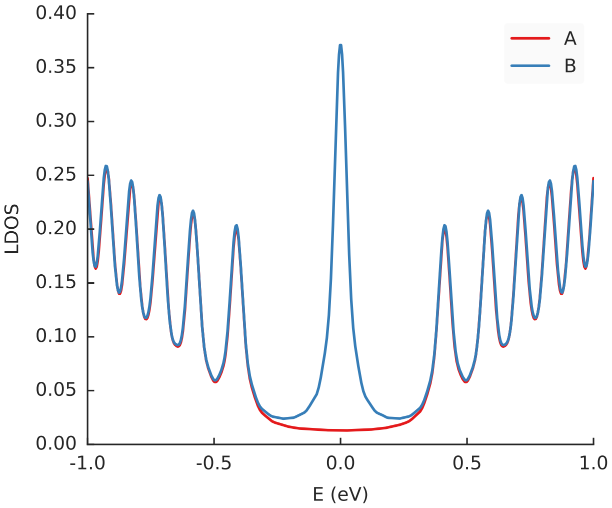
```
>>> np.all(model.hamiltonian.data == -2.8)
False
```

However, it isn't convenient to keep track of the displacement and strained hoppings separately. Instead, we can package them together in one function which is going to return both modifiers:

```
def triaxial_strain(c, beta=3.37):
    """Produce both the displacement and hopping energy modifier"""
    @pb.site_position_modifier
    def displacement(x, y, z):
        ux = 2*c * x*y
        uy = c * (x**2 - y**2)
        return x + ux, y + uy, z

    @pb.hopping_energy_modifier
    def strained_hopping(energy, x1, y1, z1, x2, y2, z2):
        l = np.sqrt((x1-x2)**2 + (y1-y2)**2 + (z1-z2)**2)
        w = l / graphene.a_cc - 1
        return energy * np.exp(-beta*w)

    return displacement, strained_hopping
```

The `triaxial_strain` function now has everything we need. We'll apply it to a slightly larger system so that we can clearly calculate the local density of states (LDOS). For more information about this computation method see the *Green's function* section. Right now, it's enough to know that we will calculate the LDOS at the center of the strained system, separately for sublattices A and B.

```
model = pb.Model(
    graphene.monolayer(),
    pb.regular_polygon(num_sides=3, radius=40, angle=pi),
    triaxial_strain(c=0.0025)
)
greens = pb.greens.kpm(model)

for sub_name in ['A', 'B']:
    ldos = greens.calc_ldos(energy=np.linspace(-1, 1, 500), broadening=0.03,
                            position=[0, 0], sublattice=sub_name)
    ldos.plot(label=sub_name)
pb.pltutils.legend()
```



Strain in graphene has an effect similar to a magnetic field. That's why we see Landau-level-like features in the LDOS. Note that the zero-energy peak has double intensity on one sublattice but zero on the other: this is a unique feature of the strain-induced pseudo-magnetic field.

Eigensolvers

Solvers were first introduced in the *Band structure* section and then used throughout the tutorial to present the results of the various models we constructed. This section will take a more detailed look at the concrete `lapack()` and `arpack()` eigenvalue solvers and their common `Solver` interface.

LAPACK

The `Solver` class establishes the interface of a solver within Pybinding, but it does not contain a concrete diagonalization routine. For this reason we never instantiate the plain solver, only its implementations such as `solver.lapack()`.

The LAPACK implementation works on dense matrices which makes it well suited only for small systems. However, a great advantage of this solver is that it always solves for all eigenvalues and eigenvectors of a Hamiltonian matrix. This makes it perfect for calculating the entire band structure of the bulk or nanoribbons, as has been shown several times in this tutorial.

ARPACK

The `solver.arpack()` implementation works on sparse matrices which makes it suitable for large systems. However, only a small subset of the total eigenvalues and eigenvectors can be calculated. This tutorial already contains a few examples where the ARPACK solver was used, and one more is presented below.

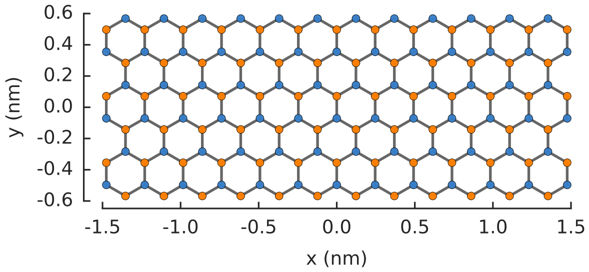
Solver interface

No matter which concrete solver is used, they all share a common `Solver` interface. The two primary properties are `eigenvalues` and `eigenvectors`. These are the raw results of the exact diagonalization of the Hamiltonian matrix.

```
>>> from pybinding.repository import graphene
>>> model = pb.Model(graphene.monolayer())
>>> model.hamiltonian.todense()
[[ 0.0 -2.8]
 [-2.8  0.0]]
>>> solver = pb.solver.lapack(model)
>>> solver.eigenvalues
[-2.8 2.8]
>>> solver.eigenvectors
[[-0.707 -0.707]
 [-0.707  0.707]]
```

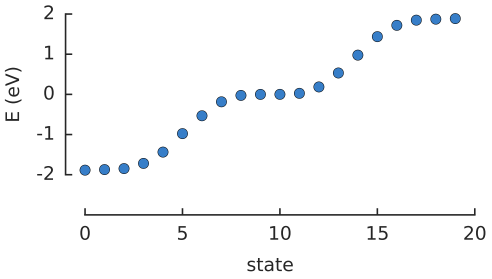
The properties contain just the raw data. However, `Solver` also offers a few convenient calculation methods. We'll demonstrate these on a simple rectangular graphene system.

```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(x=3, y=1.2)
)
model.plot()
```



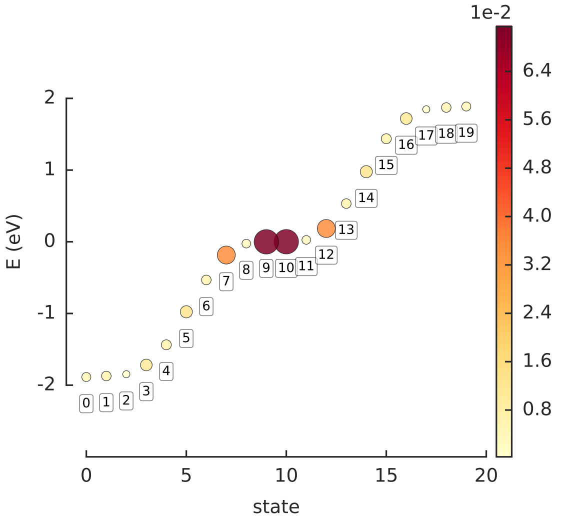
First, we'll take a look at the calc_eigenvalues() method. While its job is essentially the same as the eigenvalues property, there is one key difference: the property returns a raw array, while the method returns an Eigenvalues result object. These objects have convenient functions built in and they know how to plot their data:

```
# for the 20 lowest energy eigenvalues
solver = pb.solver.arpack(model, k=20)
eigenvalues = solver.calc_eigenvalues()
eigenvalues.plot()
```



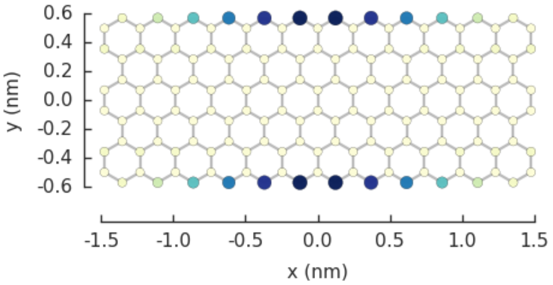
The basic plot just shows the state number and energy of each eigenstate, but we can also do something more interesting. If we pass a position argument to calc_eigenvalues() it will calculate the probability density $|\Psi(\vec{r})|^2$ at that position for each eigenstate and we can view the result using Eigenvalues.plot_heatmap():

```
eigenvalues = solver.calc_eigenvalues(
    # position in [nm]
    map_probability_at=[0.1, 0.6]
)
eigenvalues.plot_heatmap(
    show_indices=True
)
pb.pltutils.colorbar()
```



In this case we are interested in the probability density at [x, y] = [0.1, 0.6], i.e. a lattice site at the top zigzag edge of our system. Note that the given position does not need to be precise: the probability will be computed for the site closest to the given coordinates. From the figure we can see that the probability at the edge is highest for the two zero-energy states: numbers 9 and 10. We can take a look at the spatial map of state 9 using the `calc_probability()` method:

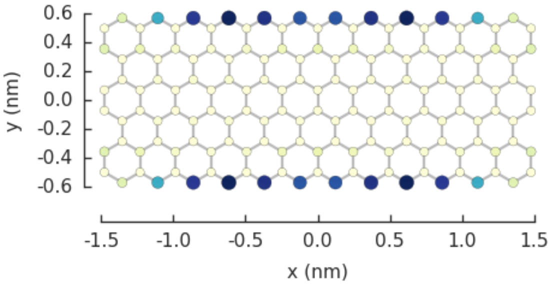
```
prob = solver.calc_probability(9)
prob.plot()
```



The result object in this case is a `StructureMap` with the probability density $|\Psi(\vec{r})|^2$ as its data attribute. As expected, the most prominent states are at the zigzag edges of the system.

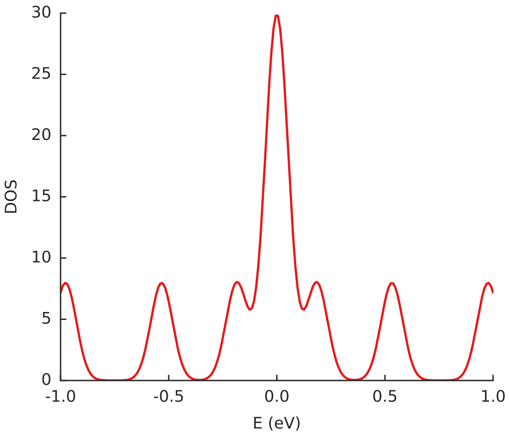
An alternative way to get a spatial map of the system is via the local density of states (LDOS). The `calc_spatial_ldos()` method makes this easy. The LDOS map is requested for a specific energy value instead of a state number and it considers multiple states within a Gaussian function with the specified broadening:

```
ldos_map = solver.calc_spatial_ldos(
    energy=0, broadening=0.05  # [eV]
)
ldos_map.plot()
```



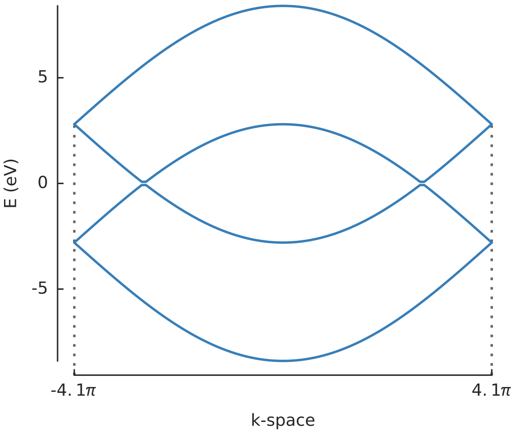
The total density of states can be calculated with `calc_dos()`:

```
dos = solver.calc_dos(
    energies=np.linspace(-1, 1, 200),
    broadening=0.05  # [eV]
)
dos.plot()
```



Our example system is quite small so the DOS does not resemble bulk graphene. The zero-energy peak stands out as the signature of the zigzag edge states. For periodic systems, the wave vector can be controlled using `Solver.set_wave_vector()`. This allows us to compute the eigenvalues at various points in k-space. For example:

```
from math import pi

model = pb.Model(
    graphene.monolayer(),
    pb.translational_symmetry()
)
solver = pb.solver.lapack(model)

kx_lim = pi / graphene.a
kx_path = np.linspace(-kx_lim, kx_lim, 100)
ky_outer = 0
ky_inner = 2*pi / (3*graphene.a_cc)

outer_bands = []
for kx in kx_path:
    solver.set_wave_vector([kx, ky_outer])
    outer_bands.append(solver.eigenvalues)

inner_bands = []
for kx in kx_path:
    solver.set_wave_vector([kx, ky_inner])
    inner_bands.append(solver.eigenvalues)

for bands in [outer_bands, inner_bands]:
    result = pb.results.Bands(kx_path,␣
↪bands)
    result.plot()
```



This example shows the basic principle of iterating over a path in k-space in order to calculate the band structure. However, this is made much easier with the `Solver.calc_bands()` method. This was already covered in the *Band structure* section and will not be repeated here. But keep in mind that this calculation does not need to be done manually, `Solver.calc_bands()` is the preferred way.

Green's function

Green's function methods were used briefly in the *Fields and effects* and *Defects and strain* sections. As with the eigensolvers, there is one common `Greens` interface while the underlying algorithm may be implemented in various ways. At this time, `kpm()` is the only one that comes with the package.

KPM

The `kpm()` implementation is a very efficient way of calculating Green's function, especially for large sparse Hamiltonian matrices. Based on the kernel polynomial method, the approach approximates the Green's function by expanding it into a series of Chebyshev polynomials.

A great advantage of this method is that memory usage and computation time scale linearly with problem dimension. In addition, computation time can be tuned based on the required accuracy which is conveniently expressed as a Lorentzian broadening width. Finally, each spatial site can be computed separately which means local properties can be calculated efficiently at a fraction of the time required for the entire system.

Greens interface

The interface is quite simple. A `Greens` function is created with the desired implementation:

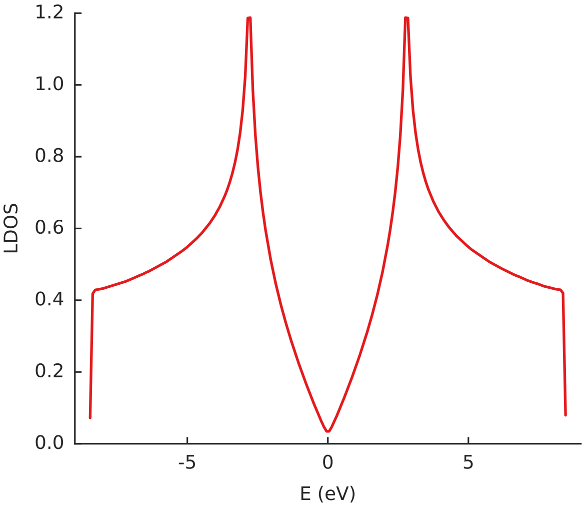
```
model = pb.Model(graphene.monolayer())
greens = pb.greens.kpm(model)
```

It can then be used to calculate the Green's function corresponding to Hamiltonian matrix element `i,j` for the desired energy range and broadening:

```
g_ij = greens(i, j, energy=np.linspace(-9, 9, 100), broadening=0.1)
```

The result is the raw Green's function data for the given matrix element. However, there is also a convenient `Greens.calc_ldos()` method which makes it very easy to calculate the local density of states (LDOS). In the next example we'll use a large square sheet of pristine graphene:

```
model = pb.Model(
    graphene.monolayer(),
    pb.rectangle(60)
)
greens = pb.greens.kpm(model)
ldos = greens.calc_ldos(
    energy=np.linspace(-9, 9, 200),
    broadening=0.05,
    position=[0, 0]
)
ldos.plot()
```



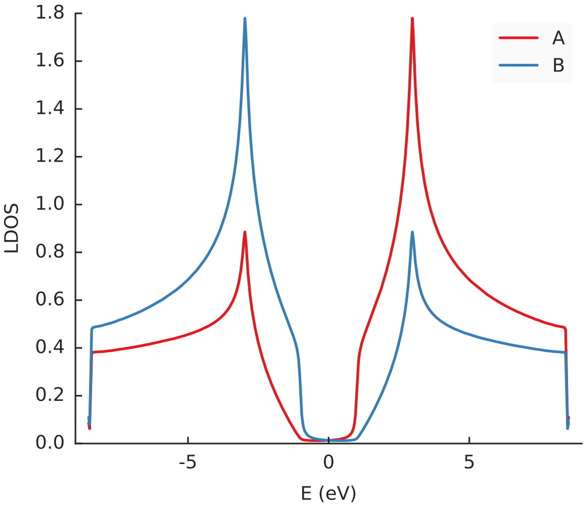
The LDOS is calculated for energies between -9 and 9 eV with a Lorentzian broadening of 50 meV. Since this is the *local* density of states, position is also a required argument. We target the center of our square system where we expect to see the well-known LDOS shape of pristine graphene. Indeed, that is what the resulting `LDOS` object shows after invoking its `plot()` method.

Tight-binding systems have lattice sites at discrete positions, which in principle means that we cannot freely choose just any position for LDOS calculations. However, as a convenience the `Greens.calc_ldos()` method will automatically find a valid site closest to the given target position. We can optionally also choose a specific sublattice:

```
ldos = greens.calc_ldos(energy=np.linspace(-9, 9, 200), broadening=0.05,
                        position=[0, 0], sublattice='B')
```

In this case we would calculate the LDOS at a site of sublattice B closest to the center of the system. We can try that on a graphene system with a mass term:

```
model = pb.Model(
    graphene.monolayer(),
    graphene.mass_term(1),
    pb.rectangle(60)
)
greens = pb.greens.kpm(model)

for sub_name in ['A', 'B']:
    ldos = greens.calc_ldos(
        energy=np.linspace(-9, 9, 500),
        broadening=0.05,
        position=[0, 0],
        sublattice=sub_name
    )
    ldos.plot(label=sub_name)
pb.pltutils.legend()
```



Multiple plots compose nicely here. A large band gap is visible at zero energy due to the inclusion of `graphene.mass_term()`. It places an onsite potential with the opposite sign in each sublattice. This is also why the LDOS lines for A and B sublattices are antisymmetric around zero energy with respect to one another.

Bibliography

[1] G. E. Moore. Cramming More Components Onto Integrated Circuits. *Electronics*, 38(8):114–117, (1965). doi:10.1109/JPROC.1998.658762.

[2] G. E. Moore. Progress in digital integrated electronics. *Int. Electron Devices Meet.*, 21:11–13, (1975). doi:10.1109/N-SSC.2006.4804410.

[3] J. Hruska. This is what the death of Moore's law looks like, *ExtremeTech*, (2014). http://goo.gl/4AXYdE.

[4] Intel. Tick-Tock Model, (2006). http://www.intel.com/content/www/us/en/silicon-innovations/intel-tick-tock-model-general.html.

[5] Intel. Introduction of the Process-Architecture-Optimization model, *Annu. Rep.*, (2016). http://files.shareholder.com/downloads/INTC/867590276x0xS50863-16-105/50863/filing.pdf.

[6] D. Misra, H. Iwai, and H. Wong. High-k Gate Dielectrics. *Interface*, 14(2):30–34, (2005).

[7] A. K. Geim and K. S. Novoselov. The rise of graphene. *Nat. Mater.*, pages 183–191, (2007). doi:10.1038/nmat1849.

[8] R. Peierls. Quelques propriétés typiques des corps solides. *Ann. l'I.H.P.*, 5(3):177–222, (1935).

[9] L. D. Landau. Zur Theorie der phasenumwandlungen II. *Phys. Z. Sowjetunion*, 11:26–35, (1937).

[10] N. D. Mermin. Crystalline order in two dimensions. *Phys. Rev.*, 176 (1):250–254, (1968). doi:10.1103/PhysRev.176.250.

[11] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov. Electric field effect in atomically thin carbon films. *Science*, 306(5696):666–9, (2004). doi:10.1126/science.1102896.

[12] J. C. Meyer, A. K. Geim, M. I. Katsnelson, K. S. Novoselov, T. J. Booth, and S. Roth. The structure of suspended graphene sheets. *Nature*, 446(7131):60–63, (2007). doi:10.1038/nature05545.

[13] A. Fasolino, J. H. Los, and M. I. Katsnelson. Intrinsic ripples in graphene. *Nat. Mater.*, 6(11):858–861, (2007). doi:10.1038/nmat2011.

[14] H. P. Boehm, R. Setton, and E. Stumpp. Nomenclature and terminology of graphite intercalation compounds (IUPAC Recommendations 1994). *Pure Appl. Chem.*, 66(9), (1994). doi:10.1351/pac199466091893.

[15] P. R. Wallace. The band theory of graphite. *Phys. Rev.*, 71(9):622–634, (1947). doi:10.1103/PhysRev.71.622.

[16] J. W. McClure. Diamagnetism of Graphite. *Phys. Rev.*, 104(3):666–671, (1956). doi:10.1103/PhysRev.104.666.

[17] G. W. Semenoff. Condensed-Matter simulation of a three-Dimensional anomaly. *Phys. Rev. Lett.*, 53(26):2449–2452, (1984). doi:10.1103/PhysRevLett.53.2449.

[18] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, M. I. Katsnelson, I. V. Grigorieva, S. V. Dubonos, and A. A. Firsov. Two-dimensional gas of massless Dirac fermions in graphene. *Nature*, 438 (7065):197–200, (2005). doi:10.1038/nature04233.

[19] Y. Zhang, Y.-W. Tan, H. L. Stormer, and P. Kim. Experimental observation of the quantum Hall effect and Berry's phase in graphene. *Nature*, 438(7065):201–204, (2005). doi:10.1038/nature04235.

[20] A. H. Castro Neto, N. M. R. Peres, K. S. Novoselov, and A. K. Geim. The electronic properties of graphene. *Rev. Mod. Phys.*, 81(1):109–162, (2009). doi:10.1103/RevModPhys.81.109.

[21] D. Gunlycke, H. Lawler, and C. White. Room-temperature ballistic transport in narrow graphene strips. *Phys. Rev. B*, 75(8):085418, (2007). doi:10.1103/PhysRevB.75.085418.

[22] A. S. Mayorov, R. V. Gorbachev, S. V. Morozov, L. Britnell, R. Jalil, L. A. Ponomarenko, P. Blake, K. S. Novoselov, K. Watanabe, T. Taniguchi, and A. K. Geim. Micrometer-Scale Ballistic Transport in Encapsulated Graphene at Room Temperature. *Nano Lett.*, 11(6): 2396–2399, (2011). doi:10.1021/nl200758b.

[23] C. Lee, X. Wei, J. W. Kysar, and J. Hone. Measurement of the elastic properties and intrinsic strength of monolayer graphene. *Science*, 321 (5887):385–8, (2008). doi:10.1126/science.1157996.

[24] N. Savage. Materials science: Super carbon. *Nature*, 483(7389):S30–S31, (2012). doi:10.1038/483S30a.

[25] O. Klein. Die Reflexion von Elektronen an einem Potentialsprung nach der relativistischen Dynamik von Dirac. *Zeitschrift fur Phys.*, 53(3-4): 157–165, (1929). doi:10.1007/BF01339716.

[26] M. I. Katsnelson, K. S. Novoselov, and A. K. Geim. Chiral tunnelling and the Klein paradox in graphene. *Nat. Phys.*, 2(9):620–625, (2006). doi:10.1038/nphys384.

[27] L. Geppert. The amazing vanishing transistor act. *IEEE Spectr.*, 39 (10), (2002).

[28] S. W. Bedell, A. Khakifirooz, and D. K. Sadana. Strain scaling for CMOS. *MRS Bull.*, 39(February):131–137, (2014). doi:10.1557/mrs.2014.5.

[29] S. Mirsky. Balancing Act. *Sci. Am.*, 305(5):96–96, (2011). doi:10.1038/scientificamerican1111-96.

[30] V. M. Pereira, A. H. Castro Neto, and N. M. R. Peres. Tight-binding approach to uniaxial strain in graphene. *Phys. Rev. B*, 80(4):045401, (2009). doi:10.1103/PhysRevB.80.045401.

[31] F. Guinea, M. I. Katsnelson, and A. K. Geim. Energy gaps and a zero-field quantum Hall effect in graphene by strain engineering. *Nat Phys*, 6(1):30–33, (2010).

[32] F. Guinea, M. I. Katsnelson, and M. A. H. Vozmediano. Midgap states and charge inhomogeneities in corrugated graphene. *Phys. Rev. B*, 77 (7):075422, (2008). doi:10.1103/PhysRevB.77.075422.

[33] F. Guinea, A. K. Geim, M. I. Katsnelson, and K. S. Novoselov. Generating quantizing pseudomagnetic fields by bending graphene ribbons. *Phys. Rev. B*, 81(3):035408, (2010). doi:10.1103/PhysRevB.81.035408.

[34] N. Levy, S. A. Burke, K. L. Meaker, M. Panlasigui, A. Zettl, F. Guinea, A. H. C. Neto, and M. F. Crommie. Strain-Induced Pseudo-Magnetic Fields Greater Than 300 Tesla in Graphene Nanobubbles. *Science*, 329 (5991):544–547, (2010). doi:10.1126/science.1191700.

[35] F. Guinea, M. I. Katsnelson, and A. K. Geim. Energy gaps and a zero-field quantum Hall effect in graphene by strain engineering. *Nat. Phys.*, 6(1):30–33, (2009). doi:10.1038/nphys1420.

[36] V. M. Pereira and A. H. Castro Neto. Strain Engineering of Graphene's Electronic Structure. *Phys. Rev. Lett.*, 103(4):046801,

(2009). doi:10.1103/PhysRevLett.103.046801.

[37] N. N. Klimov, S. Jung, S. Zhu, T. Li, C. A. Wright, S. D. Solares, D. B. Newell, N. B. Zhitenev, and J. A. Stroscio. Electromechanical properties of graphene drumheads. *Science*, 336(6088):1557–61, (2012). doi:10.1126/science.1220335.

[38] J. Bunch, S. Verbridge, and J. Alden. Impermeable atomic membranes from graphene sheets. *nano Lett.*, pages 3–7, (2008).

[39] T. Georgiou, L. Britnell, P. Blake, R. V. Gorbachev, A. Gholinia, A. K. Geim, C. Casiraghi, and K. S. Novoselov. Graphene bubbles with controllable curvature. *Appl. Phys. Lett.*, 99(9):093103, (2011). doi:10.1063/1.3631632.

[40] Y. Wang, D. Wong, A. V. Shytov, V. W. Brar, S. Choi, Q. Wu, H.-Z. Tsai, W. Regan, A. Zettl, R. K. Kawakami, S. G. Louie, L. S. Levitov, and M. F. Crommie. Observing Atomic Collapse Resonances in Artificial Nuclei on Graphene. *Science*, 340(6133):734–737, (2013). doi:10.1126/science.1234320.

[41] D. P. DiVincenzo and E. J. Mele. Self-consistent effective-mass theory for intralayer screening in graphite intercalation compounds. *Phys. Rev. B*, 29(4):1685–1694, (1984). doi:10.1103/PhysRevB.29.1685.

[42] F. D. M. Haldane. Model for a quantum hall effect without landau levels: Condensed-matter realization of the "parity anomaly". *Phys. Rev. Lett.*, 61(18):2015–2018, (1988). doi:10.1103/PhysRevLett.61.2015.

[43] C. Itzykson and J.-B. Zuber. *Quantum field theory.* Courier Corporation, (2006).

[44] I. Pomeranchuk and Y. Smorodinsky. No Title. *J. Phys.*, 9:97, (1945).

[45] Y. Zeldovich and V. S. Popov. No Title. *Sov. Phys. Usp.*, 14:673, (1972).

[46] W. Greiner, B. Müller, and J. Rafelski. *Quantum Electrodynamics of Strong Fields.* Springer Berlin Heidelberg, Berlin, Heidelberg, (1985). doi:10.1007/978-3-642-82272-8.

[47] J. Schweppe, A. Gruppe, K. Bethge, H. Bokemeyer, T. Cowan, H. Folger, J. Greenberg, H. Grein, S. Ito, R. Schule, D. Schwalm, K. Stiebing, N. Trautmann, P. Vincent, and M. Waldschmidt. Observation of a Peak Structure in Positron Spectra from U+Cm Collisions. *Phys. Rev. Lett.*, 51(25):2261–2264, (1983). doi:10.1103/PhysRevLett.51.2261.

[48] T. Cowan, H. Backe, M. Begemann, K. Bethge, H. Boke-

meyer, H. Folger, J. Greenberg, H. Grein, A. Gruppe, Y. Kido, M. Klüver, D. Schwalm, J. Schweppe, K. Stiebing, N. Trautmann, and P. Vincent. Anomalous Positron Peaks from Supercritical Collision Systems. *Phys. Rev. Lett.*, 54(16):1761–1764, (1985). doi:10.1103/PhysRevLett.54.1761.

[49] J. C. Slater and G. F. Koster. Simplified LCAO method for the periodic potential problem. *Phys. Rev.*, 94(6):1498–1524, (1954). doi:10.1103/PhysRev.94.1498.

[50] W. A. Harrison. *Electronic structure and the properties of solids.* Dover Publications, (1989).

[51] A. P. Sutton. *Electronic Structure of Materials.* (1993). doi:0198517548.

[52] M. Finnis. *Interatomic forces in condensed matter*, volume 001. (2003). doi:10.1093/acprof:oso/9780198509776.001.0001.

[53] A. Altland and B. Simons. *Condensed Matter Field Theory.* (2010). doi:10.1017/CBO9781107415324.004.

[54] A. C. Hewson. *The Kondo Problem to Heavy Fermions.* (1993). doi:10.1017/CBO9780511470752.

[55] A. F. Young and P. Kim. Quantum interference and Klein tunneling in graphene heterojunctions. *Nat. Phys.*, 5(3):222–226, (2008). doi:10.1038/nphys1198.

[56] C. Lanczos. An iteration method for the solution of the eigenvalue prob- lem of linear differential and integral operators. *Natl. Bur. Standards*, (1950).

[57] D. Calvetti, L. Reichel, and D. C. Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electron. Trans. Numer. Anal.*, 2(March):1–21, (1994).

[58] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide.* Society for Industrial and Applied Mathematics, (1998). doi:10.1137/1.9780898719628.

[59] E. Polizzi. Density-matrix-based algorithm for solving eigenvalue problems. *Phys. Rev. B*, 79(11):115112, (2009). doi:10.1103/PhysRevB.79.115112.

[60] A. Weisse. Chebyshev expansion approach to the AC conductivity of the Anderson model. *Eur. Phys. J. B*, 40(2):125–128, (2004). doi:10.1140/epjb/e2004-00250-6.

[61] L.-W. Wang and A. Zunger. Solving Schrodinger's equation around a desired energy: Application to silicon quantum dots. *J. Chem. Phys.*, 100(3):2394, (1994). doi:10.1063/1.466486.

[62] L.-W. Wang. Calculating the density of states and optical-absorption spectra of large quantum systems by the plane-wave moments method. *Phys. Rev. B*, 49(15):10154–10158, (1994). doi:10.1103/PhysRevB.49.10154.

[63] H. Röder and R. N. Silver. *Densities of States and Thermodynamics of Mega-Dimensional Sparse Matrices*, chapter Densities, pages 301–308. Springer US, Boston, MA, (1995). ISBN 978-1-4615-1937-9. doi:10.1007/978-1-4615-1937-9_28. http://dx.doi.org/10.1007/978-1-4615-1937-9{_}28http://link.springer.com/10.1007/978-1-4615-1937-9{_}28.

[64] H. Röder, R. N. Silver, D. a. Drabold, and J. J. Dong. Kernel polynomial method for a nonorthogonal electronic-structure calculation of amorphous diamond. *Phys. Rev. B*, 55(23):15382–15385, (1997). doi:10.1103/PhysRevB.55.15382.

[65] L. Covaci, F. M. Peeters, and M. Berciu. Efficient Numerical Approach to Inhomogeneous Superconductivity: The Chebyshev-Bogoliubov–de Gennes Method. *Phys. Rev. Lett.*, 105(16):167006, (2010). doi:10.1103/PhysRevLett.105.167006.

[66] J. H. Garcia, L. Covaci, and T. G. Rappoport. Real-Space Calculation of the Conductivity Tensor for Disordered Topological Matter. *Phys. Rev. Lett.*, 114(11):116602, (2015). doi:10.1103/PhysRevLett.114.116602.

[67] A. Alvermann and H. Fehske. Chebyshev approach to quantum systems coupled to a bath. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 77 (4):1–17, (2008). doi:10.1103/PhysRevB.77.045125.

[68] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske. The kernel polynomial method. *Rev. Mod. Phys.*, 78(1):275–306, (2006). doi:10.1103/RevModPhys.78.275.

[69] List of ab initio software. https://en.wikipedia.org/wiki/List{_}of{_}quantum{_}chemistry{_}and{_}solid-state{_}physics{_}software.

[70] T. Yusufaly, D. Vanderbilt, and S. Coh. PythTB. http://physics.rutgers.edu/pythtb/.

[71] C. W. Groth, M. Wimmer, A. R. Akhmerov, and X. Waintal. Kwant:

A software package for quantum transport. *New J. Phys.*, 16, (2014). doi:10.1088/1367-2630/16/6/063065.

[72] J. R. Shaw. QuickFill: An efficient flood fill algorithm. *Code Proj.*, (2004).

[73] K. Hormann and A. Agathos. The point in polygon problem for arbitrary polygons. *Comput. Geom.*, 20(3):131–144, (2001). doi:10.1016/S0925-7721(01)00012-8.

[74] A. Buluç, J. T. Fineman, M. Frigo, J. R. Gilbert, and C. E. Leiserson. Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks. *Proc. twenty-first Annu. Symp. Parallelism algorithms Archit. - SPAA '09*, i:233, (2009). doi:10.1145/1583991.1584053.

[75] F. Vázquez, J. J. Fernández, and E. M. Garzón. A new approach for sparse matrix vector product on NVIDIA GPUs. *Concurr. Comput. Pract. Exp.*, 23(8):815–826, (2011). doi:10.1002/cpe.1658.

[76] F. Guinea. Strain engineering in graphene. *Solid State Commun.*, 152 (15):1437–1441, (2012). doi:10.1016/j.ssc.2012.04.019.

[77] G. W. Jones and V. M. Pereira. Designing electronic properties of two-dimensional crystals through optimization of deformations. *New J. Phys.*, 16, (2014). doi:10.1088/1367-2630/16/9/093044.

[78] S. Scharfenberg, D. Z. Rocklin, C. Chialvo, R. L. Weaver, P. M. Goldbart, and N. Mason. Probing the mechanical properties of graphene using a corrugated elastic substrate. *Appl. Phys. Lett.*, 98(9):091908, (2011). doi:10.1063/1.3553228.

[79] M. Neek-Amal and F. Peeters. Strain-engineered graphene through a nanostructured substrate. II. Pseudomagnetic fields. *Phys. Rev. B*, 85 (19):195446, (2012). doi:10.1103/PhysRevB.85.195446.

[80] M. Neek-Amal, L. Covaci, and F. M. Peeters. Nanoengineered nonuniform strain in graphene using nanopillars. *Phys. Rev. B*, 86(4):041405, (2012). doi:10.1103/PhysRevB.86.041405.

[81] C. Androulidakis, E. N. Koukaras, J. Parthenios, G. Kalosakas, K. Papagelis, and C. Galiotis. Graphene flakes under controlled biaxial deformation. *Sci. Rep.*, 5:18219, (2015). doi:10.1038/srep18219.

[82] S. P. Koenig, N. G. Boddeti, M. L. Dunn, and J. S. Bunch. Ultrastrong adhesion of graphene membranes. *Nat. Nanotechnol.*, 6(9):543–6, (2011). doi:10.1038/nnano.2011.123.

[83] W. Bao, F. Miao, Z. Chen, H. Zhang, W. Jang, C. Dames, and C. N. Lau. Controlled ripple texturing of suspended graphene and ultrathin graphite membranes. *Nat. Nanotechnol.*, 4(9):562–566, (2009). doi:10.1038/nnano.2009.191.

[84] J. Zang, S. Ryu, N. Pugno, Q. Wang, Q. Tu, M. J. Buehler, and X. Zhao. Multifunctionality and control of the crumpling and unfolding of large-area graphene. *Nat. Mater.*, 12(4):321–325, (2013). doi:10.1038/nmat3542.

[85] C. S. C. Downs, A. Usher, and J. Martin. Towards observation of pseudo-magnetic fields in suspended graphene devices. *J. Appl. Phys.*, 119(19):194305, (2016). doi:10.1063/1.4950879.

[86] H. Suzuura and T. Ando. Phonons and electron-phonon scattering in carbon nanotubes. *Phys. Rev. B*, 65(23):235412, (2002). doi:10.1103/PhysRevB.65.235412.

[87] J. L. Mañes. Symmetry-based approach to electron-phonon interactions in graphene. *Phys. Rev. B*, 76(4):045430, (2007). doi:10.1103/PhysRevB.76.045430.

[88] A. L. Kitt, V. M. Pereira, A. K. Swan, and B. B. Goldberg. Lattice-corrected strain-induced vector potentials in graphene. *Phys. Rev. B*, 85(11):115432, (2012). doi:10.1103/PhysRevB.85.115432.

[89] A. L. Kitt, V. M. Pereira, A. K. Swan, and B. B. Goldberg. Erratum: Lattice-corrected strain-induced vector potentials in graphene [Phys. Rev. B 85 , 115432 (2012)]. *Phys. Rev. B*, 87(15):159909, (2013). doi:10.1103/PhysRevB.87.159909.

[90] F. de Juan, M. Sturla, and M. a. H. Vozmediano. Space Dependent Fermi Velocity in Strained Graphene. *Phys. Rev. Lett.*, 108(22):227205, (2012). doi:10.1103/PhysRevLett.108.227205.

[91] F. de Juan, J. L. Manes, and M. a. H. Vozmediano. Gauge fields from strain in graphene. *Phys. Rev. B*, 87(16):165131, (2013). doi:10.1103/PhysRevB.87.165131.

[92] O. L. Blakslee. Elastic Constants of Compression-Annealed Pyrolytic Graphite. *J. Appl. Phys.*, 41(8):3373, (1970). doi:10.1063/1.1659428.

[93] Z. H. Ni, T. Yu, Y. H. Lu, Y. Y. Wang, Y. P. Feng, and Z. X. Shen. Uniaxial Strain on Graphene: Raman Spectroscopy Study and Band-Gap Opening. *ACS Nano*, 2(11):2301–2305, (2008). doi:10.1021/nn800459e.

[94] K.-J. Kim, Y. M. Blanter, and K.-H. Ahn. Interplay between real and

pseudomagnetic field in graphene with strain. *Phys. Rev. B*, 84(8): 081401, (2011). doi:10.1103/PhysRevB.84.081401.

[95] G. M. M. Wakker, R. P. Tiwari, and M. Blaauboer. Localization and circulating currents in curved graphene devices. *Phys. Rev. B*, 84(19): 195427, (2011). doi:10.1103/PhysRevB.84.195427.

[96] E. V. Castro, K. S. Novoselov, S. V. Morozov, N. M. R. Peres, J. M. B. L. Dos Santos, J. Nilsson, F. Guinea, A. K. Geim, and A. H. C. Neto. Biased bilayer graphene: Semiconductor with a gap tunable by the electric field effect. *Phys. Rev. Lett.*, 99(21):8–11, (2007). doi:10.1103/PhysRevLett.99.216802.

[97] K. S. Novoselov, E. McCann, S. V. Morozov, V. I. Fal'ko, M. I. Katsnelson, U. Zeitler, D. Jiang, F. Schedin, and A. K. Geim. Unconventional quantum Hall effect and Berry/'s phase of 2[pi] in bilayer graphene. *Nat Phys*, 2(3):177–180, (2006).

[98] E. McCann and V. I. Fal'ko. Landau-Level Degeneracy and Quantum Hall Effect in a Graphite Bilayer. *Phys. Rev. Lett.*, 96(8):086805, (2006). doi:10.1103/PhysRevLett.96.086805.

[99] S.-M. Choi, S.-H. Jhi, and Y.-W. Son. Controlling Energy Gap of Bilayer Graphene by Strain. *Nano Lett.*, 10(9):3486–3489, (2010). doi:10.1021/nl101617x.

[100] B. Verberck, B. Partoens, F. M. Peeters, and B. Trauzettel. Strain-induced band gaps in bilayer graphene. *Phys. Rev. B*, 85(12):125403, (2012). doi:10.1103/PhysRevB.85.125403.

[101] M. Mucha-Kruczyński, I. L. Aleiner, and V. I. Fal'ko. Strained bilayer graphene: Band structure topology and Landau level spectrum. *Phys. Rev. B*, 84(4):041404, (2011). doi:10.1103/PhysRevB.84.041404.

[102] W.-Y. He, Y. Su, M. Yang, and L. He. Creating in-plane pseudomagnetic fields in excess of 1000 T by misoriented stacking in a graphene bilayer. *Phys. Rev. B*, 89(12):125418, (2014). doi:10.1103/PhysRevB.89.125418.

[103] M. Ramezani Masir, D. Moldovan, and F. Peeters. Pseudo magnetic field in strained graphene: Revisited. *Solid State Commun.*, 175-176: 76–82, (2013). doi:10.1016/j.ssc.2013.04.001.

[104] M. Neek-Amal, L. Covaci, K. Shakouri, and F. M. Peeters. Electronic structure of a hexagonal graphene flake subjected to triaxial stress. *Phys. Rev. B*, 88(11):115428, (2013).

doi:10.1103/PhysRevB.88.115428.

[105] P. A. M. Dirac. The Quantum Theory of the Electron. *Proc. R. Soc. A Math. Phys. Eng. Sci.*, 117(778):610–624, (1928). doi:10.1098/rspa.1928.0023.

[106] N. M. R. Peres, F. Guinea, and A. H. Castro Neto. Electronic properties of disordered two-dimensional carbon. *Phys. Rev. B*, 73(12):125411, (2006). doi:10.1103/PhysRevB.73.125411.

[107] T. Ando. Screening Effect and Impurity Scattering in Monolayer Graphene. *J. Phys. Soc. Japan*, 75(7):074716, (2006). doi:10.1143/JPSJ.75.074716.

[108] E. Hwang, S. Adam, and S. Sarma. Carrier Transport in Two-Dimensional Graphene Layers. *Phys. Rev. Lett.*, 98(18):186806, (2007). doi:10.1103/PhysRevLett.98.186806.

[109] V. N. Kotov, B. Uchoa, V. M. Pereira, F. Guinea, and A. H. Castro Neto. Electron-Electron Interactions in Graphene: Current Status and Perspectives. *Rev. Mod. Phys.*, 84(3):1067–1125, (2012). doi:10.1103/RevModPhys.84.1067.

[110] S. Das Sarma, S. Adam, E. H. Hwang, and E. Rossi. No Title. *Rev. Mod. Phys.*, 83:407, (2011).

[111] V. R. Khalilov and C.-L. Ho. Dirac Electron in a Coulomb Field in (2+1) Dimensions. *Mod. Phys. Lett. A*, 13(08):615–622, (1998). doi:10.1142/S0217732398000668.

[112] V. M. Pereira, J. Nilsson, and A. H. Castro Neto. Coulomb Impurity Problem in Graphene. *Phys. Rev. Lett.*, 99(16):166802, (2007). doi:10.1103/PhysRevLett.99.166802.

[113] A. V. Shytov, M. I. Katsnelson, and L. S. Levitov. Atomic Collapse and Quasi–Rydberg States in Graphene. *Phys. Rev. Lett.*, 99(24):246802, (2007). doi:10.1103/PhysRevLett.99.246802.

[114] A. V. Shytov, M. I. Katsnelson, and L. S. Levitov. Vacuum Polarization and Screening of Supercritical Impurities in Graphene. *Phys. Rev. Lett.*, 99(23):236801, (2007). doi:10.1103/PhysRevLett.99.236801.

[115] Y. Wang, V. W. Brar, A. V. Shytov, Q. Wu, W. Regan, H.-Z. Tsai, A. Zettl, L. S. Levitov, and M. F. Crommie. Mapping Dirac quasiparticles near a single Coulomb impurity on graphene. *Nat. Phys.*, 8(9): 653–657, (2012). doi:10.1038/nphys2379.

[116] A. Luican-Mayer, M. Kharitonov, G. Li, C.-p. Lu, I. Skachko, A.-M. B.

Gonçalves, K. Watanabe, T. Taniguchi, and E. Y. Andrei. Screening Charged Impurities and Lifting the Orbital Degeneracy in Graphene by Populating Landau Levels. *Phys. Rev. Lett.*, 112(3):036804, (2014). doi:10.1103/PhysRevLett.112.036804.

[117] Y. Liu, M. Weinert, and L. Li. Determining charge state of graphene vacancy by noncontact atomic force microscopy and first-principles calculations. *Nanotechnology*, 26(3):035702, (2015). doi:10.1088/0957-4484/26/3/035702.

[118] J. Mao, Y. Jiang, D. Moldovan, G. Li, K. Watanabe, T. Taniguchi, M. R. Masir, F. M. Peeters, and E. Y. Andrei. Realization of a tunable artificial atom at a supercritically charged vacancy in graphene. *Nat. Phys.*, 12(6):545–549, (2016). doi:10.1038/nphys3665.

[119] O. Lehtinen, J. Kotakoski, A. V. Krasheninnikov, A. Tolvanen, K. Nordlund, and J. Keinonen. Effects of ion bombardment on a two-dimensional target: Atomistic simulations of graphene irradiation. *Phys. Rev. B*, 81(15):153401, (2010). doi:10.1103/PhysRevB.81.153401.

[120] V. M. Pereira, J. M. B. Lopes dos Santos, and A. H. Castro Neto. Modeling disorder in graphene. *Phys. Rev. B*, 77(11):115109, (2008). doi:10.1103/PhysRevB.77.115109.

[121] V. M. Pereira, F. Guinea, J. M. B. Lopes dos Santos, N. M. R. Peres, and A. H. Castro Neto. Disorder Induced Localized States in Graphene. *Phys. Rev. Lett.*, 96(3):036801, (2006). doi:10.1103/PhysRevLett.96.036801.

[122] R. S. Deacon, K. C. Chuang, R. J. Nicholas, K. S. Novoselov, and A. K. Geim. Cyclotron resonance study of the electron and hole velocity in graphene monolayers. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 76(8):2–5, (2007). doi:10.1103/PhysRevB.76.081406.

[123] F. Ding. Theoretical study of the stability of defects in single-walled carbon nanotubes as a function of their distance from the nanotube end. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 72(24):1–7, (2005). doi:10.1103/PhysRevB.72.245409.

[124] J. Repp. Controlling the Charge State of Individual Gold Adatoms. *Science*, 305(5683):493–495, (2004). doi:10.1126/science.1099557.

[125] A. D. Zhao, Q. X. Li, L. Chen, H. J. Xiang, W. H. Wang, S. Pan, B. Wang, X. D. Xiao, J. L. Yang, J. G. Hou, and Q. S. Zhu. Controlling the Kondo effect of an adsorbed magnetic ion through its chemical

bonding. *Science*, 309:1542–1544, (2005). doi:10.1126/science.1113449.

[126] E. Y. Andrei, G. Li, and X. Du. Electronic properties of graphene: a perspective from scanning tunneling microscopy and magnetotransport. *Reports Prog. Phys.*, 75(5):056501, (2012). doi:10.1088/0034-4885/75/5/056501.

[127] R. Dombrowski, C. Steinebach, C. Wittneven, M. Morgenstern, and R. Wiesendanger. Tip-induced band bending by scanning tunneling spectroscopy of the states of the tip-induced quantum dot on InAs(110). *Phys. Rev. B*, 59(12):8043–8048, (1999). doi:10.1103/PhysRevB.59.8043.

[128] V. W. Brar, R. Decker, H.-M. Solowan, Y. Wang, L. Maserati, K. T. Chan, H. Lee, C. O. Girit, A. Zettl, S. G. Louie, M. L. Cohen, and M. F. Crommie. Gate-controlled ionization and screening of cobalt adatoms on a graphene surface. *Nat. Phys.*, 7(October 2010):43–47, (2011). doi:10.1038/nphys1807.

[129] D. Wong, J. Velasco, L. Ju, J. Lee, S. Kahn, H.-z. Tsai, C. Germany, T. Taniguchi, K. Watanabe, A. Zettl, F. Wang, and M. F. Crommie. Characterization and manipulation of individual defects in insulating hexagonal boron nitride using scanning tunnelling microscopy. *Nat. Nanotechnol.*, 10(11):949–953, (2015). doi:10.1038/nnano.2015.188.

[130] R. L. Heinisch, F. X. Bronold, and H. Fehske. Mie scattering analog in graphene: Lensing, particle confinement, and depletion of Klein tunneling. *Phys. Rev. B*, 87(15):155409, (2013). doi:10.1103/PhysRevB.87.155409.

[131] J.-S. Wu and M. M. Fogler. Scattering of two-dimensional massless Dirac electrons by a circular potential barrier. *Phys. Rev. B*, 90(23): 235402, (2014). doi:10.1103/PhysRevB.90.235402.

[132] J. Bardeen. Tunnelling from a Many-Particle Point of View. *Phys. Rev. Lett.*, 6(2):57–59, (1961). doi:10.1103/PhysRevLett.6.57.

[133] L. Brey and H. A. Fertig. Linear response and the Thomas-Fermi approximation in undoped graphene. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 80(3):1–6, (2009). doi:10.1103/PhysRevB.80.035406.

[134] Y. Zhao, J. Wyrick, F. D. Natterer, J. F. Rodriguez-Nieva, C. Lewandowski, K. Watanabe, T. Taniguchi, L. S. Levitov, N. B. Zhitenev, and J. A. Stroscio. Creating and probing electron whispering-gallery modes in graphene. *Science*, 348(6235):672–675, (2015). doi:10.1126/science.aaa7469.

[135] N. T. Khoa, S. W. Kim, D.-H. Yoo, E. J. Kim, and S. H. Hahn. Size-dependent work function and catalytic performance of gold nanoparticles decorated graphene oxide sheets. *Appl. Catal. A Gen.*, 469:159–164, (2014). doi:10.1016/j.apcata.2013.08.046.

[136] V. V. Cheianov, V. Fal'ko, and B. L. Altshuler. The Focusing of Electron Flow and a Veselago Lens in Graphene p-n Junctions. *Science*, 315(5816):1252–1255, (2007). doi:10.1126/science.1138020.

[137] J. Cserti, A. Pályi, and C. Péterfalvi. Caustics due to a negative refractive index in circular graphene p-n junctions. *Phys. Rev. Lett.*, 99(24):1–4, (2007). doi:10.1103/PhysRevLett.99.246801.

[138] A. Matulis, M. R. Masir, and F. M. Peeters. Application of optical beams to electrons in graphene. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 83(11):1–7, (2011). doi:10.1103/PhysRevB.83.115458.

[139] A. V. Shytov, M. S. Rudner, and L. S. Levitov. Klein backscattering and Fabry-P??rot interference in graphene heterojunctions. *Phys. Rev. Lett.*, 101(15):10–13, (2008). doi:10.1103/PhysRevLett.101.156804.

[140] T. Taychatanapat, K. Watanabe, T. Taniguchi, and P. Jarillo-Herrero. Electrically tunable transverse magnetic focusing in graphene. *Nat Phys*, 9(4):225–229, (2013).

[141] F.-M. Zhang, Y. He, and X. Chen. Guided modes in graphene waveguides. *Appl. Phys. Lett.*, 94(21):212105, (2009). doi:10.1063/1.3143614.

[142] M. Ramezani Masir, P. Vasilopoulos, and F. Peeters. Fabry-Pérot resonances in graphene microstructures: Influence of a magnetic field. *Phys. Rev. B*, 82(11):1–12, (2010). doi:10.1103/PhysRevB.82.115417.

[143] M. Pöllinger, D. O'Shea, F. Warken, and A. Rauschenbeutel. Ultrahigh-Q Tunable Whispering-Gallery-Mode Microresonator. *Phys. Rev. Lett.*, 103(5):1–4, (2009). doi:10.1103/PhysRevLett.103.053901.

[144] J. Lee, D. Wong, J. Velasco Jr, J. F. Rodriguez-Nieva, S. Kahn, H.-z. Tsai, T. Taniguchi, K. Watanabe, A. Zettl, F. Wang, L. S. Levitov, and M. F. Crommie. Imaging electrostatically confined Dirac fermions in graphene quantum dots. *Nat. Phys.*, (June):1–6, (2016). doi:10.1038/nphys3805.

[145] C. Gutiérrez, L. Brown, C.-J. Kim, J. Park, and A. N. Pasupathy. Klein tunnelling and electron trapping in nanometre-scale graphene quantum dots. *Nat. Phys.*, (June):1–8, (2016). doi:10.1038/nphys3806.

[146] B. Van Duppen and F. M. Peeters. Klein paradox for a pn junction

in multilayer graphene. *EPL (Europhysics Lett.*, 102(2):27001, (2013). doi:10.1209/0295-5075/102/27001.

[147] Y. Jiang, J. Mao, and E. Andrei. Unpublished experimental work regarding STM-tip-induced pn-junctions in graphene.

[148] V. N. Oraevskii, A. I. Rex, and V. B. Semikoz. Spontaneous production of positrons by a Coulomb center in a homogeneous magnetic field. *Sov. Phys. JETP*, 45(3):428–435, (1977).

[149] O. V. Gamayun, E. V. Gorbar, and V. P. Gusynin. Magnetic field driven instability of charged center in graphene. *Phys. Rev. B*, 83(23): 235104, (2011). doi:10.1103/PhysRevB.83.235104.

[150] Y. Zhang, Y. Barlas, and K. Yang. Coulomb impurity under magnetic field in graphene: A semiclassical approach. *Phys. Rev. B*, 85(16): 165423, (2012). doi:10.1103/PhysRevB.85.165423.

[151] T. Maier and H. Siedentop. Stability of impurities with Coulomb potential in graphene with homogeneous magnetic field. *J. Math. Phys.*, 53(9):095207, (2012). doi:10.1063/1.4728982.

[152] S. Kim and S. Yang. Coulomb impurity problem of graphene in magnetic fields. *arXiv Prepr. arXiv1404.0468*, (2):1–13, (2014).

[153] M. Goerbig. Electronic properties of graphene in a strong magnetic field. *Rev. Mod. Phys.*, 83(4):1193–1243, (2011). doi:10.1103/RevModPhys.83.1193.

List of publications

- *Resonant valley filtering of massive Dirac electrons*, D. Moldovan, M. Ramezani Masir, L. Covaci, and F. M. Peeters, Phys. Rev. B **86**, 115431 (2012). doi:10.1103/PhysRevB.86.115431
- *Pseudo magnetic field in strained graphene: Revisited*, M. Ramezani Masir, D. Moldovan, and F. M. Peeters, Solid State Commun. **175-176**, 76–82 (2013). doi:10.1016/j.ssc.2013.04.001
- *Electronic states in a graphene flake strained by a Gaussian bump*, D. Moldovan, M. Ramezani Masir, and F. M. Peeters, Phys. Rev. B **88**, 035446 (2013). doi:10.1103/PhysRevB.88.035446
- *Strain engineering of the electronic properties of bilayer graphene quantum dots*, D. Moldovan and F. M. Peeters, Phys. Status Solidi - Rapid Res. Lett. **10**, 39 (2015). doi:10.1002/pssr.201510228
- *Veselago lensing in graphene with a p-n junction: Classical versus quantum effects*, S. P. Milovanovic, D. Moldovan, and F. M. Peeters, J. Appl. Phys. **118**, 154308 (2015). doi:10.1063/1.4933395
- *Realization of a Tunable Artificial Atom at a Charged Vacancy in Graphene*, J. Mao, Y. Jiang, D. Moldovan, G. Li, K. Watanabe, T. Taniguchi, M.R. Masir, F.M. Peeters, and E.Y. Andrei, Nat. Phys. **12**, 545 (2016). doi:10.1038/nphys3665