ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Integrierte Systeme
Integrated Systems Laboratory

DEPARTMENT OF
INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Spring Semester 2026

Non-Uniform Fourier Transforms in Quantum Transport Simulations

Master Thesis

Yongda Li
yongli@student.ethz.ch

26 June 2026

Advisors: Anders Winka, awinka@iis.ee.ethz.ch
Dr. Alexandros Nikolaos Ziogas, alziogas@iis.ee.ethz.ch

Professor: Prof. Mathieu Luisier, mluisier@iis.ee.ethz.ch

Abstract

The use of a non-uniform adaptive energy grid in quantum transport simulations is investigated. The non-equilibrium Green's function (NEGF) formalism with a self-consistent GW approximation is used. The energy grid is created on-the-fly to target the features of the Green's function G and other variables in the self-consistent Born approximation (SCBA) loop. The goal is to use the same memory footprint as a uniform grid, but to achieve better accuracy and convergence. The difficulty lies in computing the convolution and correlation in the SCBA loop, which requires a Fourier transform. The Fast Fourier Transform (FFT) requires a uniform grid, so a method to perform a Fourier transform on a non-uniform grid is needed. Direct methods to perform the Fourier transform on a non-uniform grid were found to be not accurate enough for Quatrex. Instead, an interpolation method was chosen and implemented in Quatrex. The method takes in a non-uniform grid and interpolates the missing values onto a uniform grid, which can then be transformed using the FFT. The end-to-end results show that the band edges, maximum self-energy update, and currents do not converge when using the adaptive grid with the GW interaction. But by removing the GW interaction and only including phonons, the adaptive grid was able to converge to a lower self-energy update than the uniform grid.

Acknowledgments

I'd like to thank my direct supervisors Anders Winka and Dr. Alexandros Nikolaos Ziogas for the continual support, guidance, and feedback throughout this project. I want to thank the Quatrex team for their their help getting familiar with codebase and their technical help. I want to thank the nano-tcad lunch group for light-hearted discussions at lunch. I thank Prof. Mathieu Luisier for providing the opportunity to work on this project. Lastly, I thank my friends and family for reminding me that there is more to life than work.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. **In consultation with the supervisor**, one of the following two options must be selected:

☐ I hereby declare that I authored the work in question independently, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

☒ I hereby declare that I authored the work in question independently. In doing so I only used the authorised aids, which included suggestions from the supervisor regarding language and content and generative artificial intelligence technologies. The use of the latter and the respective source declarations proceeded in consultation with the supervisor.

Title of paper or thesis:

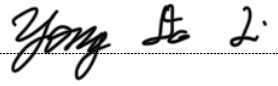
Non-Uniform Fourier Transforms in Quantum Transport Simulations

Authored by:
If the work was compiled in a group, the names of all authors are required.

Last name(s):	**First name(s):**
Li	Yongda

With my signature I confirm the following:
- I have adhered to the rules set out in the Citation Guidelines.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date	**Signature(s)**
Zürich, 26 June 2026	

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

[1] For further information please consult the ETH Zurich websites, e.g. https://ethz.ch/en/the-eth-zurich/education/ai-in-education.html and https://library.ethz.ch/en/researching-and-publishing/scientific-writing-at-eth-zurich.html (subject to change).

Contents

Chapter 1

Introduction

The need for increased compute is almost a certainty in today's world. However, it is harder to achieve compute gains. The paradigm during Moore's Law was to simply wait 2 years for the node process to shrink. And by following appropriate scaling laws, one can pack more transistors in the same footprint and effectively gain free compute.

Moore's Law is ending. The scaling laws for planar transistors have broken down. The gate's control over the channel is greatly diminished due to transistor sizes becoming so small. To keep up with the demand for increasing compute, designers have tried to make improvements across the entire semiconductor stack. At the highest level, these include better algorithms and software. At the middle level, these include better digital architecture and heterogeneous computing. But at the most fundamental level, one needs to improve the transistor.

Research at the device level can be divided up between experimental and non-experimental. Experimental work refers to fabrication and measurement techniques. Non-experimental refers to theoretical work and simulations. They synergistically build on each other and allow designers to make steady improvements. The specific advantages of simulations are two-fold:

1. Simulations allow for rapid parametric optimization of the design without the need for fabrication and testing.

2. Simulations allow any physical variable to be inspected at any instance, freeing designers from the limitations of measurement techniques and noise.

For simulations to be maximally useful, they must be fast and accurate. The previous generation of simulations used a classical or semi-classical theoretical framework that neglected many quantum mechanical effects. These were good enough for planar transistors, but they broke down as transistor sizes shrank. They predicted better device performance than experiments measured. The reason was that they did not account for quantum mechanical effects such as tunneling and many-body interactions. Moving into a quantum mechanical simulation framework can help solve the accuracy issues. But the additional computational complexity and memory requirements cannot be ignored.

This thesis investigates the use of a non-uniform adaptive energy grid to reduce the memory requirement while maintaining accuracy during the Fourier Transform computations of a quantum transport simulation.

1.1. Non-Equilibrium Green's Function (NEGF) with GW Approximation

The Non-Equilibrium Green's Function (NEGF) formalism is a powerful theoretical framework to model current flow. It is a many-body formalism that tracks how a system evolves in time under the influence of external bias. Separate Green's functions (advanced G^A, retarded G^R, greater $G^>$, lesser $G^<$) are used to track different particle states. Self-energies (Σ) are used to to model interactions between particles, such as electron-electron and electron-phonon interactions. They are also used to model the coupling between the device and the contacts. The self-consistent Born approximation (SCBA) is a common approximation used to solve the NEGF equations. It iteratively updates the Green's functions and self-energies until convergence is reached.

The approach preserves the wave nature of the charge carriers. This makes it very accurate for nano-scale devices where tunneling effects are significant.

The GW approximation is a specific approximation used to model electron-electron interactions. The bare Coulomb potential (V) captures the direct interaction between electrons, but it does not account for the screening effect. The surrounding electron density actively weakens and retards electron-electron interactions. Instead, the dynamically screened Coulomb potential (W) is used, which captures the screening effect. It is computed from the bare Coulomb potential (V) and the polarization function (P). The coupled system of equations is shown in Eq. (1.1).

$$\begin{cases} G = G_0 + G_0 \Sigma G \\ \Sigma = \Sigma_{\text{contacts}} + \Sigma_{\text{GW}} \\ \Sigma_{\text{GW}} = iGW \\ W = V + VPW \\ P = -iGG \end{cases} \tag{1.1}$$

Due to the cross-dependence of the Green's function G and screened Coulomb potential W, their governing equations must be solved self-consistently. There is a high computational cost to each iteration. The simulation is commonly stopped after one iteration. This is known as the G_0W_0 approximation and popular in predicting spectral properties of solids. This is not suitable for quantum transport, as the G_0W_0 approximation violates physical conservation laws. Thus a self-consistent solution is necessary.

1.2. Convolution Computations in the Fourier Domain

During each iteration, two convolutions must be computed.

$$P_{ij}^{\lessgtr}(E) = -i \int dE' \left[G_{ij}^{\lessgtr}(E') G_{ji}^{\gtrless}(E' - E) \right] \tag{1.2}$$

$$\Sigma_{\mathrm{GW},ij}^{\lessgtr}(E) = i \int dE' \left[G_{ij}^{\lessgtr}(E') W_{ij}^{\lessgtr}(E - E') \right] \tag{1.3}$$

The convolution can be naively computed in the original real-space domain (energy in this case) in $O(N^2)$ time, where N is the number of energy points. By the convolution theorem, the convolution can be computed in the Fourier domain (pseudo-time) as a point-wise multiplication, which can be computed in $O(N)$ time. The trouble is converting the data back and forth between the real-space and Fourier domains. The standard approach is to use a Fast Fourier Transform (FFT) for the conversion, which can be computed in $O(N \log N)$ time. Thus the overall convolution computation can be reduced from $O(N^2)$ to $O(N \log N)$ time by using the Fourier domain.

The FFT requires the data to be sampled on a uniform grid. However inspecting the data for G reveals that it has many high frequency and high amplitude features in specific regions (see Fig. 1.1). These correspond to electron densities that are localized in energy. A very fine grid is needed to capture these features. The rest of the energy spectrum is essentially flat and does not require a fine grid. However due to the uniform grid requirement of the FFT, a fine grid must be used for the entire energy spectrum. This results in a large number of energy points and a large memory requirement.

The core idea of this thesis is to use a non-uniform energy grid to represent G. The grid is adaptive, meaning it is created on-the-fly based on the features of G. A method to perform a Fourier transform on a non-uniform grid is used to forward and backward transform between the real-space and Fourier domains. The convolution is still performed as a point-wise multiplication in the Fourier domain.

By changing G to be represented on a non-uniform grid, the other variables must also be represented on a non-uniform grid. Due to the nature of the equations in 1.1, the locations of features of P and W are in the same energy range. The locations of features of G are in a different energy range. Thus it was decided to use the same non-uniform grid for P and W, and a different non-uniform grid for G. Since Σ tracks the changes to G, it makes sense to focus on the energy regions where G has features. The large peaks of Σ where there are no features of G do not affect G in the next iteration, so grid points in that region are unnecessary. Thus Σ is represented on the same grid as G.

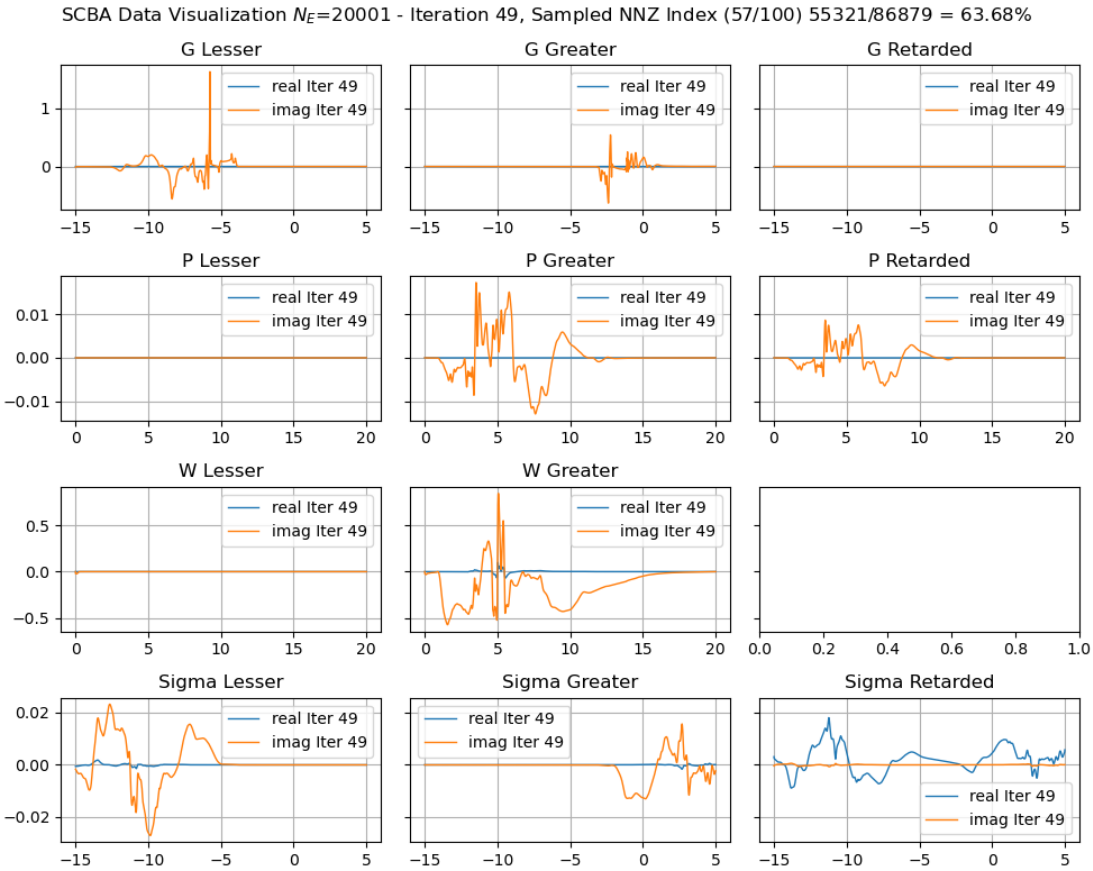


Figure 1.1.: SCBA variables during the SCBA loop.

Chapter 2

Background

2.1. Fourier Transforms

The Fourier Transform is a fundamental tool in the analysis of signals. It decomposes a signal into its constituent frequencies, allowing for the analysis of the frequency content of the signal. It can be accelerated using the Fast Fourier Transform. However the speedup is fragile since it only works if the samples are uniformly spaced. It's quite common in medical imaging or partial differential equations that the samples are non-uniformly spaced, but a Fourier transform is still desired. The non-uniform Discrete Fourier Transform is slow and runs in slow $O(N \cdot M)$ time, where N is number of input samples and M is number of output frequency modes. Instead, various schemes have been developed that allow a fast approximation of the Discrete Fourier Transform over the non-uniform grid.

2.1.1. Discrete Fourier Transform

This is a straightforward one-shot computation that calculates the spectral components of the input data. It simply computes the inner product of the input data with complex exponentials at different frequencies. N is the number of samples in the input data, M is the number of output Fourier modes, and k is the index of the Fourier coefficient being computed.

$$\hat{x}[k] = \sum_{n=0}^{M-1} x[n] e^{-j2\pi n \frac{k}{N}} \tag{2.1}$$

This can be represented in matrix form. The system matrix is size M by N. It is a Vandermonde matrix composed of the exponential phase shifts. Usually $M = N$ is chosen, so the Fourier transform is a square matrix. For simplicity, we introduce $\omega = e^{-j2\pi/N}$ as the primitive n-th root of unity.

$$\begin{bmatrix} \hat{x}[0] \\ \hat{x}[1] \\ \vdots \\ \hat{x}[M-1] \end{bmatrix} = \begin{bmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \cdots & \omega^{0\cdot (N-1)} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \cdots & \omega^{1\cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{(M-1)\cdot 0} & \omega^{(M-1)\cdot 1} & \cdots & \omega^{(M-1)\cdot (N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \tag{2.2}$$

2.1.2. Fast Fourier Transform

The Fast Fourier Transform repeatedly uses the DFT, but it reuses intermediate products so the whole transform is completed with fewer computations. It runs in $O(N \log N)$ time instead of $O(N^2)$ for the DFT. The FFT is only applicable when the samples are uniformly spaced. Due to the uniform spacing, the Fourier transform matrix has a special structure that allows the reuse of intermediate products.

2.1.3. Properties of the Fourier Transform

The Fourier Transform has many properties that relate changes in time domain to changes in frequency domain.

Time Shift

A shift in the time domain corresponds to a phase shift in the frequency domain.

$$\mathcal{F}\{x(t-t_0)\} = e^{-j2\pi f t_0} X(f) \tag{2.3}$$

In discrete time, the shift property uses indices instead of time.

$$\mathcal{F}\{x[n-n_0]\} = e^{-j2\pi \frac{k}{N} n_0} X[k] \tag{2.4}$$

Time Stretch

A stretch in the time domain corresponds to a compression in the frequency domain.

$$\mathcal{F}\{x(at)\} = \frac{1}{|a|} X\left(\frac{f}{a}\right) \tag{2.5}$$

The property in the discrete version is similar, but it uses indices instead of time.

$$\mathcal{F}\{x[an]\} = \frac{1}{|a|} X\left[\frac{k}{a}\right] \tag{2.6}$$

2.1.4. Parseval's Theorem

The theorem states that the total power in the Fourier domain is equal to the total power in the real domain.

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{x}[k]|^2 \tag{2.7}$$

The left-hand side is the power in the real domain, and the right-hand side is the power in the Fourier domain.

2.2. Perfect Reconstruction of Band-Limited Signals

2.2.1. Shannon-Nyquist Sampling Theorem

The Shannon-Nyquist sampling theorem states that a band-limited signal can be perfectly reconstructed from its samples if the sampling rate is greater than twice the highest frequency component in the signal. The Nyquist frequency is the highest frequency that can be accurate represented by a given sampling rate. With a given sampling rate f_s, the Nyquist frequency is $f_N = f_s/2$.

A standard assumption of signal processing is that the signal is band-limited. The frequency spectrum of the signal should go to zero at high frequencies. Otherwise the signal is not band-limited and would require a higher sampling rate to be accurately represented. The band-limited assumption is required to avoid aliasing.

A band-limited signal is a signal that has no frequency components above a certain cutoff frequency f_B. Aliasing occurs when there are frequency components above the Nyquist frequency, which causes them to be misrepresented as lower frequencies in the sampled signal. In the spectrum of the sampled signal, the frequency components above the Nyquist frequency are folded back into the range of frequencies below the Nyquist frequency This causes distortion in the reconstructed signal.

For uniform sampling, the continuous-time signal can be perfectly reconstructed using the Shannon-Whittaker interpolation formula.

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}\left(\frac{t - nT}{T}\right) \tag{2.8}$$

where T is the sampling period, $x[n]$ are the samples of the signal, and $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ is the normalized sinc function.

The idea is to convolve an infinite train of delta functions with a brick wall low-pass filter. The delta functions represent the samples of the signal. The low-pass filter only allows the baseband image through and blocks all higher frequency images. The sinc function is the impulse response of the ideal brick wall low-pass filter.

$$x(t) = \left(\sum_{n=-\infty}^{\infty} T \cdot \underbrace{x(nT)}_{x[n]} \cdot \delta(t - nT) \right) \circledast \left(\frac{1}{T} \text{sinc}\left(\frac{t}{T}\right) \right) \tag{2.9}$$

2.2.2. Fewer Samples than Nyquist Rate

If there are few samples, then the system of equations for reconstruction is underdetermined. The DFT system matrix from Eq. (2.2) has more columns than rows ($N > M$). There are infinitely many solutions to the system of equations, meaning there are infinitely many signals that can produce the same samples. Many of these will have high frequency modes that appear visually incorrect.

The number of samples is usually large (usually 10^3 or greater), so the solution space is very large. Thus an iterative solve method is preferred over a direct solve method. The convergence rate of iterative methods is controlled by the condition number of the system matrix. The condition number is the ratio of the largest singular value to the smallest singular value.

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} \tag{2.10}$$

A large condition number means that the system is ill-conditioned. It will converge slower and be more sensitive to small changes in the input data.

The result from Feichtinger [1] shows that the condition number is dependent on the maximal gap between the samples. A uniform grid has the smallest maximal gap, so it has the best condition number. A non-uniform grid will have a larger maximal gap, so it will have a worse condition number. This is a fundamental limitation of non-uniform sampling. The condition number can be improved by using weights to compensate for the local variables in the sampling density.

2.2.3. Compressed Sensing

One can actually reconstruct a signal with far fewer samples than the Nyquist rate if the signal is sparse in some domain. This is the idea behind compressed sensing. This may seem like a violation of the Nyquist-Shannon sampling theorem, but it is not. The Nyquist-Shannon sampling theorem only states sufficient conditions for perfect reconstruction. It does not state necessary conditions. So using fewer samples is weaker than the sufficient condition, but it may still be strong enough for perfect reconstruction.

The two criteria for compressed sensing are sparsity and incoherence.

- Sparsity means that the signal has only a few non-zero coefficients in some domain (e.g. Fourier domain, wavelet domain, etc).
- Incoherence means that the sampling basis is not aligned with the sparsity basis.

The coherence between two bases is defined as the maximum absolute inner product between any two basis vectors from the two bases.

$$\mu(\Phi, \Psi) = \max_{i,j} |\langle \phi_i, \psi_j \rangle| \tag{2.11}$$

where Φ is the sampling basis and Ψ is the sparsity basis. A low coherence means that the sampling basis is not aligned with the sparsity basis, which allows for better

reconstruction with fewer samples. The standard choice of sampling basis is a uniform random distribution. It is almost guaranteed that a uniform random distribution will be incoherent with any fixed sparsity basis. Note that this is different from the uniform grid used in the FFT, which is a deterministic sampling pattern that is not incoherent with the Fourier basis.

The naive method to solve the under-determined system is to minimize the ℓ_2 norm of the solution. This is the least squares problem.

$$\hat{x} = \arg\min_x \|Ax - b\|_2 \tag{2.12}$$

Where A is the DFT matrix from eqn 2.2, x is the solution, and b is the observed samples.

This will give the solution with the smallest energy, but it may not be the desired solution. Often there are erroneous high frequency modes. These modes can be suppressed by applying a smoothness weighting penalty. The first derivative penalty minimizes the slope.

$$w = \frac{1}{\sqrt{1+k^2}} \tag{2.13}$$

where w is the weight applied to the Fourier coefficients, and k is the index of the Fourier coefficient.

The second derivative penalty minimizes the curvature.

$$w = \frac{1}{1+k^2} \tag{2.14}$$

By assuming sparsity, we want the solution with the fewest non-zero entries. The spectrum is thus mostly flat, with a only a few non-zero frequencies corresponding to the true Fourier modes of the signal. This is the ℓ_0 norm of the solution.

$$\hat{x} = \arg\min_x \|x\|_0 \text{ subject to } Ax = b \tag{2.15}$$

This is a combinatorial optimization problem and is NP-hard to solve.

It was shown by Candes et al. [2] that the ℓ_0 solution can be approximated by the ℓ_1 solution.

$$\hat{x} = \arg\min_x \|x\|_1 \text{ subject to } Ax = b \tag{2.16}$$

The ℓ_1 norm is the sum of the absolute values of the entries in the solution. This is a convex optimization problem and can be solved efficiently with linear programming methods. Unfortunately the signals in this thesis are not sparse in the Fourier domain, so it's not appropriate to use the ℓ_1 or ℓ_0 method. Instead, the smoothness penalty is applied to the ℓ_2 method to suppress the erroneous high frequency modes.

2.2.4. Normalized ℓ_2 Error

The normalized ℓ_2 error is a common metric to evaluate the accuracy of a reconstructed signal. It is computed as:

$$\text{normalized } \ell_2 \text{ error} = \frac{\sum_{i=0}^{N} \|x_{i,\text{reconstructed}} - x_{i,\text{reference}}\|^2}{\sum_{i=0}^{N} \|x_{i,\text{reference}}\|^2} \tag{2.17}$$

The normalization makes the error scale-invariant, meaning it is not affected by the overall magnitude of the signal. In cases where the reference and reconstructed signals are on different grids, they are first interpolated to a common grid before computing the error.

2.3. Non-Uniform Discrete Fourier Transform (NUDFT)

A non-uniform Discrete Fourier transform is a Fourier transform that can be applied to non-uniformly spaced samples. It only exists in the discrete case, since the samples must be discrete to be non-uniformly spaced. The non-uniform Fourier transform is a generalization of the discrete Fourier transform. The type 1 transform takes non-uniformly spaced samples in real space and transforms them to uniformly spaced samples in Fourier space. This is sometimes called the forward transform.

$$\hat{x}[k] = \sum_{n=0}^{M-1} c[n] e^{-j2\pi k \cdot x[n]} \tag{2.18}$$

The $x[n]$ are the locations of the source samples and $c[n]$ are the complex values of the samples. Note that the n/N term in the exponential of the DFT 2.1 is replaced by $x[n]$ in the non-uniform Fourier transform. Care must be taken to ensure that the $x[n]$ are normalized to the range $[0, 1)$ to maintain the correct scaling of the Fourier transform.

The type 2 transform takes uniformly spaced samples in Fourier space and transforms them to non-uniformly spaced samples in real space. This is sometimes called the backward transform. It is used for reconstruction of the signal from the Fourier coefficients.

$$x[n] = \sum_{k=0}^{M-1} \hat{f}[k] e^{j2\pi k x[n]} \tag{2.19}$$

The $\hat{f}[k]$ are the Fourier coefficients, and $x[n]$ are the locations of the non-uniformly spaced target samples in real space. The $x[n]$ are the set of target points where the signal is reconstructed.

For the FFT, the inverse FFT is exactly the inverse operation of the forward FFT. The DFT matrix is square and the operation is unitary. However for the NUFT, the transform matrix is not square since the number of Fourier modes M can be different from the number of non-uniform samples N. So the forward and backward transforms are not exact inverses of each other. The inverse problem for the NUFT is as follows: given a set of uniformly spaced Fourier coefficients and a set of target non-uniform real-space points, find the real-space signal at the target points. If there are more target points (N) than Fourier modes (M), then the system is underdetermined and there are infinitely many solutions. If there are fewer target points (N) than Fourier modes (M), then the system is overdetermined and there is no exact solution. Thus the type 2 transform is not the same as un-doing the type 1 transform. It's more appropriate to think of the type 2 "backward" transform as a pseudo-inverse of the type 1 "forward" transform.

2.3.1. Voronoi Weights

Voronoi weights are a technique to mitigate the power distortion caused by non-uniform sampling. The Voronoi weight of a single point is computed as the average distance to the neighboring points.

$$w_n = \frac{1}{2}(x_{n+1} - x_{n-1}) \tag{2.20}$$

On the boundaries, the weight is simply the distance to the neighboring point.

$$w_0 = x_1 - x_0 \tag{2.21}$$

$$w_{N-1} = x_{N-1} - x_{N-2} \tag{2.22}$$

The Voronoi weights can be thought of as a way to compensate for the local sampling density. In regions where the samples are dense, the Voronoi weights will be small, which reduces the contribution of those samples to the Fourier transform. In regions where the samples are sparse, the Voronoi weights will be large, which increases the contribution of those samples.

The Voronoi weights are applied to the non-uniform discrete Fourier transform (NUDFT) as a weighted sum:

$$\hat{x}[k] = \sum_{n=0}^{N-1} w_n \cdot c[n] e^{-j2\pi k \cdot x[n]} \tag{2.23}$$

2.3.2. Non-Uniform Fast Fourier Transform (NUFFT)

The NUDFT can be computed with a direct method, which is just a matrix-vector multiplication. This runs in $O(NM)$ time, which is very slow for large N and M. The NUFFT is any method to accelerate the computation of the NUDFT, usually to a form resembling $O(N \log N + M)$ time. This gets close to the speed of the FFT, which is $O(N \log N)$ time.

Most methods rely on a spreading function that will spread the samples over a larger domain. The samples can be thought of as a delta function. The spreading function "smears" the delta function into a wider function that can be sampled on a uniform grid. Then, a fine uniform grid can pick up those spread points. A regular FFT is performed over the fine grid. Then finally, a correction step is performed by convolving the FFT result with the Fourier transform of the spreading function to remove spectral distortions.

The spreading function is sometimes called a "window function" or "kernel". In particular, the term "window function" comes from digital signal processing (DSP) and spectral analysis. It was realized by Jackson et al. [3] that the spectral concentration problem was very similar to selecting an ideal spreading function. The spectral concentration problem attempts to find a time window of discrete samples where the Fourier transform is maximally localized over a given frequency interval. In particular according to Barnett et al. [4], good spreading function should be tight in both real space and Fourier space.

The evolution of spreading kernels started with the Gaussian function. Then the B-splines were used, which are piecewise polynomials that are smooth and have compact support. Then the Kaiser-Bessel functions were used, which are a family of functions that are defined in terms of the modified Bessel function of the first kind. They approximate the prolate spheroidal wave functions, which are the optimal spreading functions in terms of spectral concentration. The latest development is the use of the exponential of semicircle, which have an almost identical convergence to the Kaiser-Bessel functions, but they are much faster to evaluate.

A method to accelerate the NUDFT without the use of a spreading kernel is the low-rank approximation method. This is a purely algebraic argument without any interpolation or oversampling. Instead, it concludes that the non-uniform discrete Fourier transform matrix can be well-approximated by a low-rank matrix approximation if the samples are nearly equispaced. It uses a Chebyshev expansion to compute the low-rank approximation.

2.4. Non-Uniform Grid Generation

There are many methods to generate a non-uniform grid. The most typical type is a non-uniform grid that minimizes the integration error of a function. These are referred to as "adaptive quadrature" methods. This is computed with a numerical integration method such as the trapezoidal rule or Simpson's rule. The method computes the area under the curve. More grid points are added until the change in computed area is below a user-specified threshold or until a maximum number of points are used. The grid points are actually a by-product of the numerical integration method. It is often implemented as a recursive function, with multiple calls to evaluate the function to be integrated at various grid points.

Another type of grid generation relies on a so-called "monitor function" that tracks some underlying complexity measure of the function. The goal is to evenly distribute the complexity across the domain. And the grid points are generated by inverting the monitor function.

The literature on non-uniform grid generation goes quite deep [5] [6]. Grids can be generated in 1D, 2D, 3D, and even higher dimensions. There are often additional constraints, such as smoothness of the grid, parallel grid lines, and conservation of mass. These constraints arise in the context of the numerical method, such as fluid flow or heat transfer.

For this thesis, only 1-D grid generation is considered. The complexity monitor chosen is the gradient of the function. The monitor function is inverted by a cumulative sum and 1-D linear interpolation. The first and last grid points are always the endpoints of the domain, which are the same as the uniform grid. The monitor method is guaranteed to produce a user-specified number of points. This is very useful for the FFT computations, since the output of FFTs must be length matched to other results to continue the computation.

The adaptive quadrature method does not have this guarantee, since the user only

specifies a threshold for the integration error, and the number of grid points is a by-product of the method. It was found that very small changes to the error threshold can lead to large changes in the number of grid points, which makes it difficult to use in the FFT computations. A previous project from [7] used the adaptive quadrature method. It often generated too many points. To length match the grids, the points with contributed the least to the integration error were successively removed until the appropriate number of points was reached. This "band-aid" fix is avoided by the monitor method.

Fig 2.1 demonstrates the grid generation with a gradient monitor. It creates a non-uniform grid of 17 target points. The number 17 is small and chosen for visual clarity. It places more grid points in regions of high gradient, and fewer grid points in regions of low gradient.

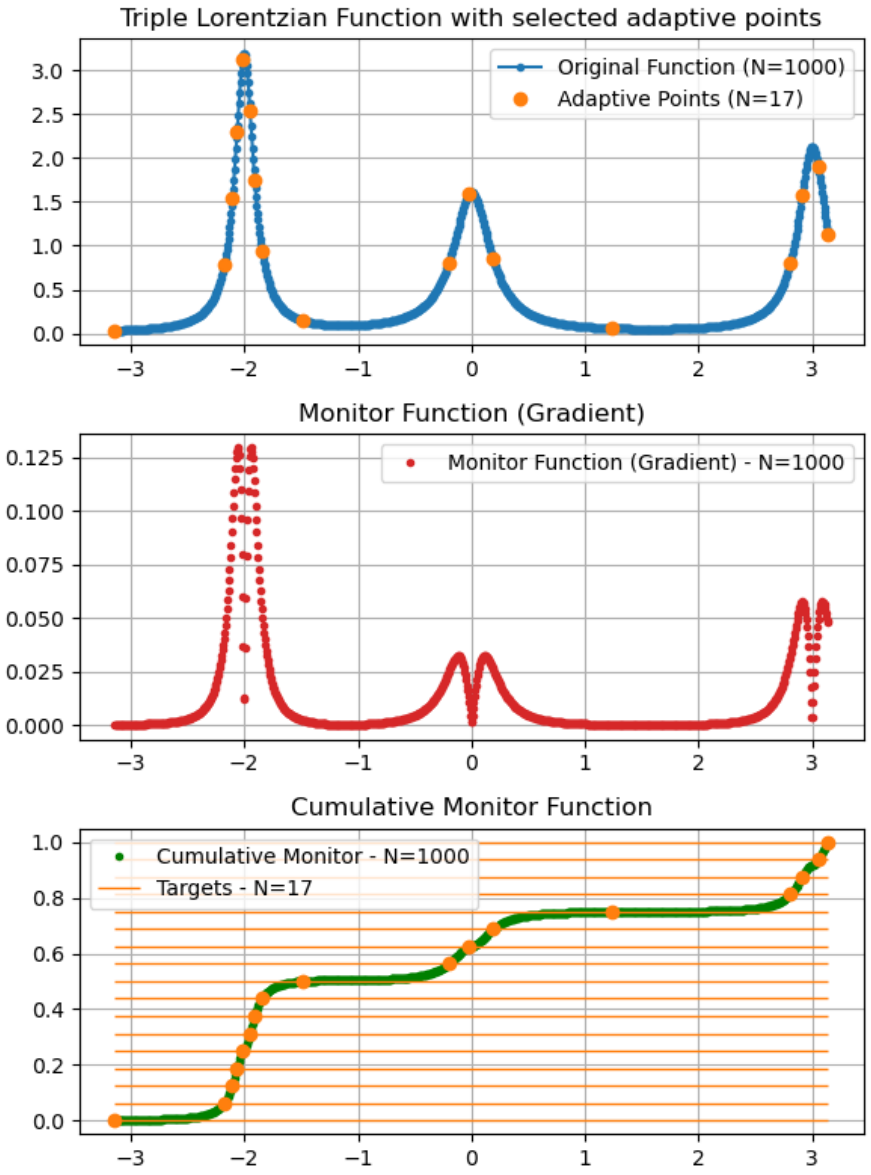


Figure 2.1.: Grid generation with a gradient monitor.

2.5. NEGF with self-consistent GW

The non-equilibrium Green's function (NEGF) is a quantum mechanical framework for describing the transport of charge carriers in a device. It is based on the concept of Green's functions, which are complex-valued functions that describe the response of a system to an external perturbation. The variables to be solved are the energy dependent Green's function. They describe the probability amplitude for an electron to propagate

from one point to another in the device. The lesser Green's function $G^<(E)$ describes the occupation of states in the device (electrons), while the greater Green's function $G^>(E)$ describes the unoccupied states (hole). The retarded Green's function $G^R(E)$ encodes the causal response of the system to a perturbation due to past sources. The advanced Green's function $G^A(E)$ encodes the acausal response of the system to a perturbation due to future sources. For most problems, the physically important quantity is the retarded Green's function. They are conjugates of each other $G^A(E) = (G^R(E))^\dagger$, so only the retarded Green's function needs to be solved. The governing equation is based on Schroedinger's equation, but it is modified to include the effects of interactions and scattering.

$$\left(E - \mathbf{H} - \mathbf{V}_{\text{ext}} - \Sigma_B^R(E) - \Sigma_{GW}^R(E)\right) \mathbf{G}^R(E) = \mathbf{I} \tag{2.24}$$

$$\mathbf{G}^\lessgtr(E) = \mathbf{G}^R(E) \left[\Sigma_B^\lessgtr(E) + \Sigma_{GW}^\lessgtr(E)\right] \left(\mathbf{G}^R(E)\right)^\dagger \tag{2.25}$$

Where E is the energy, H is the Hamiltonian of the system, $\mathbf{V}_{\text{ext}}$ is the external potential, $\Sigma_B^R(E)$ is the retarded self-energy due to the contacts, $\Sigma_{GW}^R(E)$ is the retarded self-energy due to electron-electron interactions, and $\mathbf{I}$ is the identity matrix.

The lesser/greater GW scattering self-energies are computed by a convolution.

$$\Sigma_{GW,ij}^\lessgtr(E) = i \int dE'\, G_{ij}^\lessgtr(E')\, W_{ij}^\lessgtr(E - E') \tag{2.26}$$

where W is the dynamically screened Coulomb interaction, which describes the effective interaction between electrons in the presence of other electrons.

The retarded GW scattering self-energy is composed of a Fock term (sometimes called an exchange term) and a correlation term.

$$\Sigma_{GW}^R(E) = \Sigma_{\text{Fock}}^R(E) + \Sigma_{\text{correlation}}^R(E) \tag{2.27}$$

$$\Sigma_{Fock,ij}^R = i \int dE\, G_{ij}^<(E)\, \tilde{V}_{ij} \tag{2.28}$$

$$\Sigma_{corr,ij}^R(E) = -\frac{i}{2}\Gamma_{ij}(E) + \mathcal{P} \int \frac{dE'}{2\pi} \frac{\Gamma_{ij}(E')}{E - E'} \tag{2.29}$$

The $\mathcal{P}$ symbol denotes the Cauchy principal value of the integral, which is used to handle the singularity at $E = E'$. The $\Gamma_{ij}(E)$ term is the broadening function, which describes the rate at which electrons scatter due to interactions. It is defined as:

$$\Gamma_{ij} = i\left[\Sigma_{GW,ij}^> - \Sigma_{GW,ij}^<\right] \tag{2.30}$$

The dynamically screened Coulomb interaction must be computed in two parts. The retarded component follows the form of a Dyson equation.

$$\left(\mathbf{I} - \tilde{\mathbf{V}} \cdot \mathbf{P}^R(E) - \mathbf{S}_B^R(E)\right) \cdot \mathbf{W}^R(E) = \tilde{\mathbf{V}} \tag{2.31}$$

where $\tilde{\mathbf{V}}$ is the bare Coulomb interaction, $\mathbf{P}^R(E)$ is the retarded polarization, and $\mathbf{S}_B^R(E)$ is the retarded self-energy due to the contacts.

The lesser and greater components are computed with:

$$\mathbf{W}^{\lessgtr}(E) = \mathbf{X}^R(E)\left[\tilde{\mathbf{V}}\cdot\mathbf{P}^{\lessgtr}(E)\cdot\tilde{\mathbf{V}}^{\mathrm{T}} + \mathbf{S}_B^{\lessgtr}(E)\right](\mathbf{X}^R)^{\dagger}(E) \tag{2.32}$$

A new quantity called the polarization P is introduced. It describes the response of the electron density to an external perturbation. The lesser/greater polarization is computed by a convolution of the lesser/greater Green's functions.

$$\mathbf{P}_{ij}^{\lessgtr}(E) = -i\int dE'\, G_{ij}^{\lessgtr}(E')\, G_{ji}^{\gtrless}(E'-E) \tag{2.33}$$

The retarded polarization is approximated as simply the difference between the lesser and greater polarizations.

$$\mathbf{P}^R(E) = \frac{1}{2}\left(\mathbf{P}^{>}(E) - \mathbf{P}^{<}(E)\right) \tag{2.34}$$

Once the Green's functions are solved, they can be used to compute observables such as density of states and currents.

$$n(\mathbf{R}_{\xi}) = -2i\sum_n \int_{CB} \frac{dE}{2\pi}\, G_{nn}^{<}(E) \tag{2.35}$$

$$p(\mathbf{R}_{\xi}) = 2i\sum_n \int_{VB} \frac{dE}{2\pi}\, G_{nn}^{>}(E) \tag{2.36}$$

The sums are over all Wannier orbitals associated with an atom located at $\mathbf{R}_{\xi}$. The integration is performed over the conduction band (CB) for electrons and the valence band (VB) for holes.

The current can be computed with the Meir-Wingreen formula, which is a generalization of the Landauer formula to include interactions.

$$I_n = -\frac{2q}{\hbar}\Re\int_{-\infty}^{\infty}\frac{dE}{2\pi}\,\mathrm{tr}\left\{\tilde{\Sigma}_{B,n}^{>}(E)\,G_{n,n}^{<}(E) - G_{n,n}^{>}(E)\,\tilde{\Sigma}_{B,n}^{<}(E)\right\} \tag{2.37}$$

2.5.1. Convergence

The circular dependence of $G \to P \to W \to \Sigma$ requires a self-consistent solution. The solution scheme is known as the self-consistent Born approximation (SCBA). The solution is iteratively updated until convergence is reached.

Current conservation is a fundamental physical principle that states that the total current entering a system must equal the total current leaving the system. The simulation scheme must respect current conservation to be physically accurate. We use eq. (12.34) in H. Haug and A.-P. Jauho, "Quantum Kinetics in Transport and Optics of Semiconductors"

[8] as one of the convergence metrics for current conservation. This is the Meir-Wingreen current formula, but applied to each transport cell block instead of the contact blocks.

$$\int \frac{d\omega}{2\pi} \mathrm{Tr}\{\Sigma^{<}_{\mathrm{int}}(\omega)\, G^{>}(\omega) - \Sigma^{>}_{\mathrm{int}}(\omega)\, G^{<}(\omega)\} = 0 \tag{2.38}$$

Other convergence metrics used include:

1. stability of band edges
2. maximum self-energy update
3. difference in Meir-Wingreen currents of the contact blocks (i.e. left and right boundaries)

A numerical trick is used to help achieve stable convergence. A mixing factor is applied to the update of the self-energy.

$$\Sigma^{(new)} = \alpha\Sigma^{(current)} + (1-\alpha)\Sigma^{(previous)} \tag{2.39}$$

Where α is the mixing factor, which is a value between 0 and 1. A larger α means that the current self-energy is given more weight. This causes the change between iterations to be smaller. Convergence is slower, but it is usually more stable.

Chapter 3

Related Work

The convolutions in the GW approximation are unavoidable. It is common to use the convolution theorem to speedup the FFT. This technique uses a uniform energy grid and a standard FFT/iFFT to go between the real/Fourier domains. It has been used by Deuschle et al. [9], Vetsch et al. [10], and Thygesen and Rubio [11]. An adaptive non-uniform energy grid has been used by Duflou et al. [12] and Pourfath [**?**] But it was only used to refine the grid near singularities to improve the accuracy of integrals. Both used an area-based quadrature refinement method to create the grid. When a Fourier transform was needed, Duflou et al. interpolated the data onto a fine uniform grid and used a standard FFT.

Non-uniform Fourier transforms have been widely used in medical imaging, especially in the context of 2D (image) or 3D (model) reconstruction. They are mostly concerned with going from non-uniform samples in Fourier domain to a uniform grid in real space. The non-uniform samples in Fourier domain are often obtained from a non-uniform sampling pattern in the acquisition process, such as radial sampling or spiral sampling. The non-uniform Fourier transform is used to reconstruct the image on a uniform grid in real space. Little work exists on transforming from a uniform grid in Fourier space to a target non-uniform grid in real space, which is the case for our application.

To the best of the author's knowledge, there is no literature that uses non-uniform Fast Fourier transforms (NUFFT) on an adaptive non-uniform energy grid in quantum transport simulations.

Chapter 4

Proposed Methods and Method Selection

This chapter describes the methods that were proposed and tested to perform the Fourier transforms on the non-uniform adaptive grid.

The first section describes exploratory work on a compressed sensing and multigrid approach. The compressed sensing approach does not work because the underlying assumption of sparsity in the spectrum does not hold for the signals of interest in Quatrex and the least squares solution is not accurate enough. The multigrid approach does not work because the discrete version of the time shift and stretch properties of the Fourier Transform do not hold when the grids have different grid spacings.

The second section discusses non-uniform Fourier transform (NUFT) methods. We tried directly using a non-uniform Fast Fourier Transform (NUFFT) library, but that caused power distortions. We tried to fix it with a Voronoi weighted non-uniform discrete Fourier transform (NUDFT). Both methods have large reconstruction errors, which are not acceptable for the SCBA iterations in Quatrex.

The last section discusses the interpolation method, which is the method that was ultimately selected for the final implementation in Quatrex. This method interpolates the non-uniform samples onto a uniform grid, performs the convolution using a regular FFT, and then projects the result back to the non-uniform target coordinates.

4.1. Exploratory Methods

4.1.1. Compressed Sensing Approach

By simply inspecting the spectrum of some example signals (ex. Fig 1.1), it was observed that the spectrum is not sparse. Thus a compressed sensing approach was not expected to work, as it relies on the sparsity of the spectrum. However, it was still worth trying to solve the underdetermined system of equations that arises from the non-uniform sampling with the ℓ_2 norm. This is the least squares problem (see subsection 2.2.3).

Naively solving the underdetermined system with a least squares method results in highly oscillatory results (refer to Fig 4.1). The least squares method minimizes the ℓ_2 norm of the error. This is equivalent to minimizing the energy in the signal. In practice, this leads the solution to give non-zero energies to erroneous high frequencies. One can apply a smoothness weighting to penalize high frequencies. Both a first derivative (slope) and second derivative (curvature) smoothness weighting were tried. It can fix the oscillatory behavior.

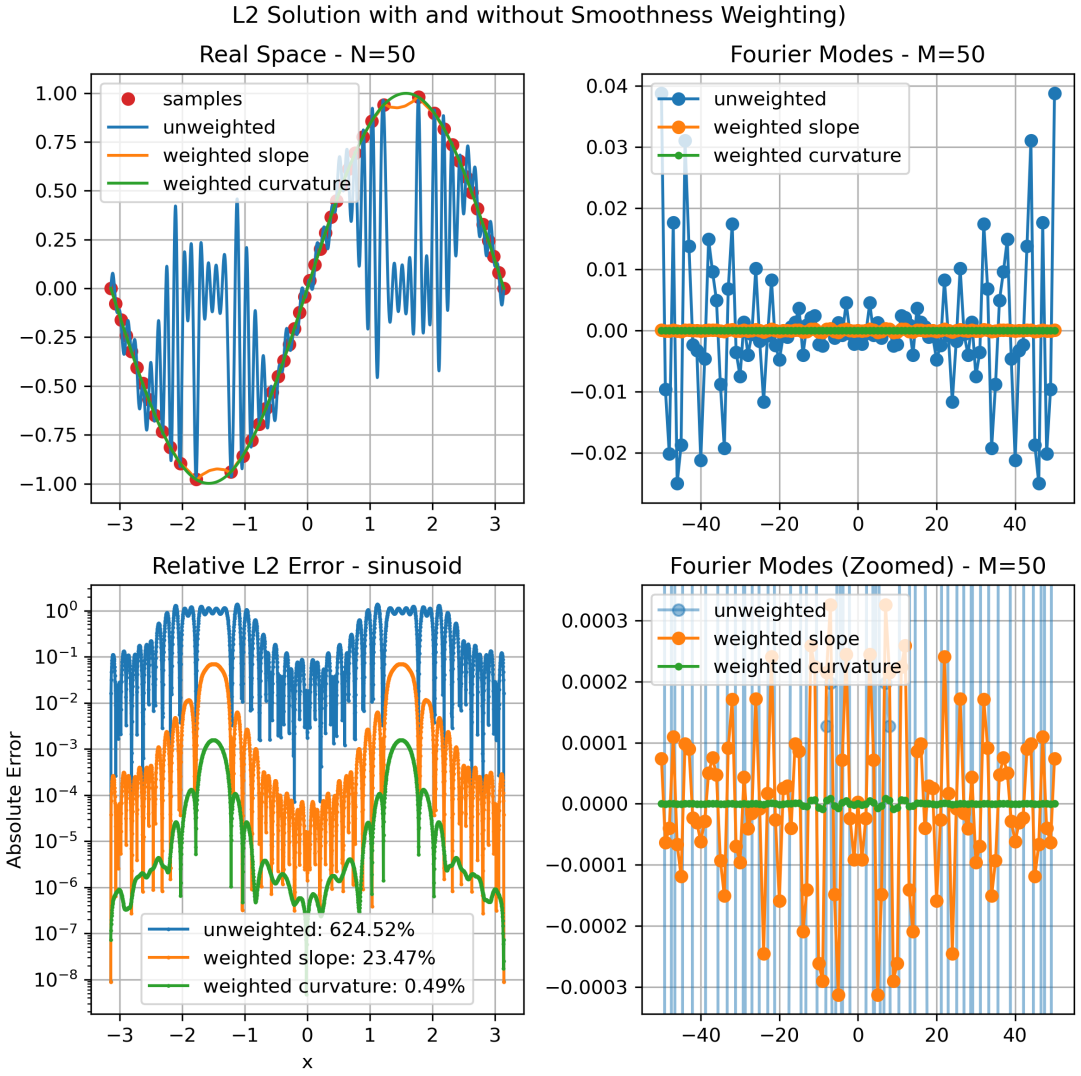


Figure 4.1.: Naively solving the underdetermined system with a least squares method results in highly oscillatory results. But applying a smoothness weighting can fix the oscillatory behavior.

However, there is still a non-trivial error present, especially in the boundary regions.

The normalized ℓ_2 error is shown in the legend on the error plot.

The solutions actually pass through the target non-uniform samples with very small error (10^{-6}). So only computing the error at the target non-uniform coordinates will give a very small error. To punish the high frequency oscillations, the ℓ_2 norm error computation was adjusted to interpolate the reconstructed solution onto the fine uniform grid. This fine grid is the common grid used for the error calculation. The same grid is used for plotting. Thus the oscillations are included in the error and the value is more reflective of the visually observed error.

A test was performed on a Lorentzian signal, which is more similar to the Green's function signals in Quatrex. The results are shown in Fig 4.2. The results are similar to the sinusoid case, but the errors are even larger, especially in the peaks of the Lorentzian.

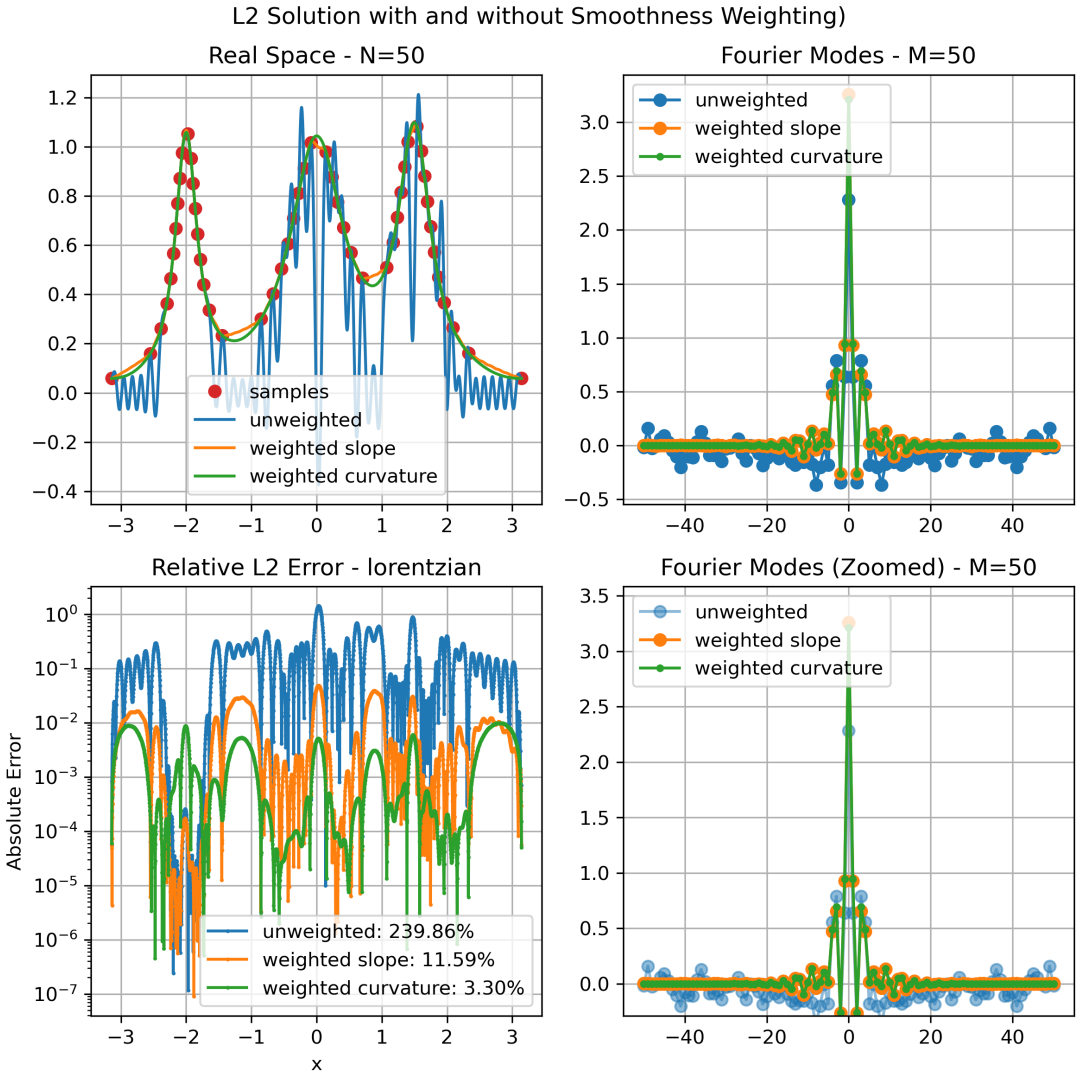


Figure 4.2.: The least squares method also has large errors on a Lorentzian signal, especially in the peaks.

Lastly, a test was performed on intermediate saved Quatrex data from the SCBA loop. The results are shown in Fig 4.3. The errors are unsuitably large for the SCBA iterations in Quatrex, especially in the peaks of the Green's function.

From an accuracy perspective, this method is not usable in Quatrex. From a computational perspective, this method is also not practical since we need to solve a linear system on every convolution. There may be some matrix structure that can be exploited to speed up the computations, but it is still expected to be very computationally expensive. Thus, it was decided not to integrate this method into Quatrex.

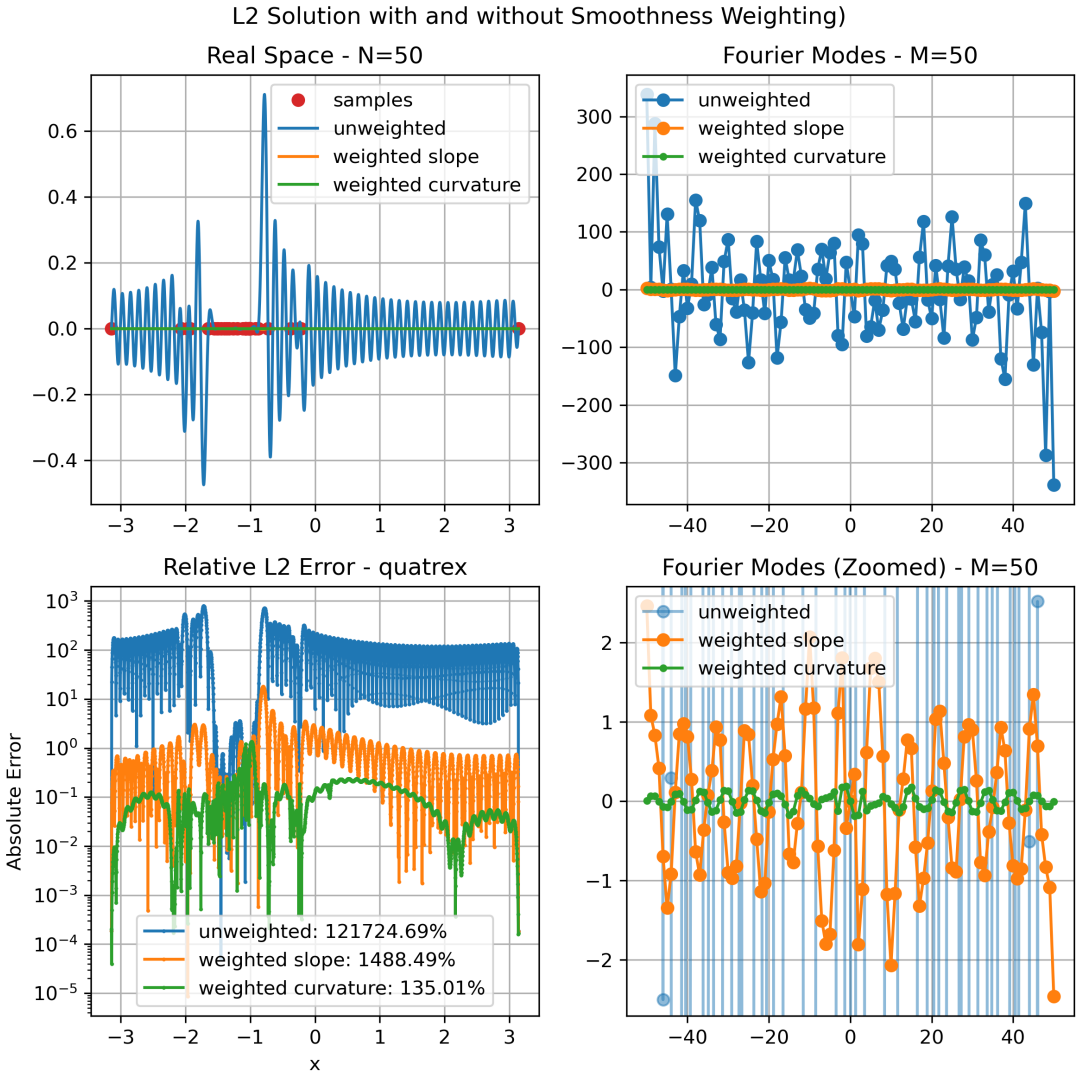


Figure 4.3.: The least squares method also has large errors on intermediate saved Quatrex data from the SCBA loop, especially in the peaks of the Green's function.

4.1.2. Multigrid Piecewise FFT

A method was proposed to use many overlapping uniform grids of different grid spacings to cover the region of interest. A FFT is performed on each uniform grid separately. The results are stitched together using the time shift (subsection 2.1.3) and time stretch (subsection 2.1.3) properties of the Fourier Transform. Thus the Fourier Transform of the entire region can be computed. The proposed advantage of this method is that is falls back on the well-optimized FFT. And as each grid is separate, the FFTs can be performed in parallel. Unfortunately this method does not work.

A proof of concept was performed on a sinusoid signal, using two overlapping grids. But the grids have the same grid spacing. Fig. 4.4 shows the two grids, their separately FFTs, and the combined result. Only the time shift property is used, since the grid spacings are the same. So the grids are just shifted in time to be stitched together. As can be seen in the second last row, the results of the combined transform is the same as the reference in both real and imaginary parts. The last row shows the error, which is on the order of machine precision (10^{-15}).

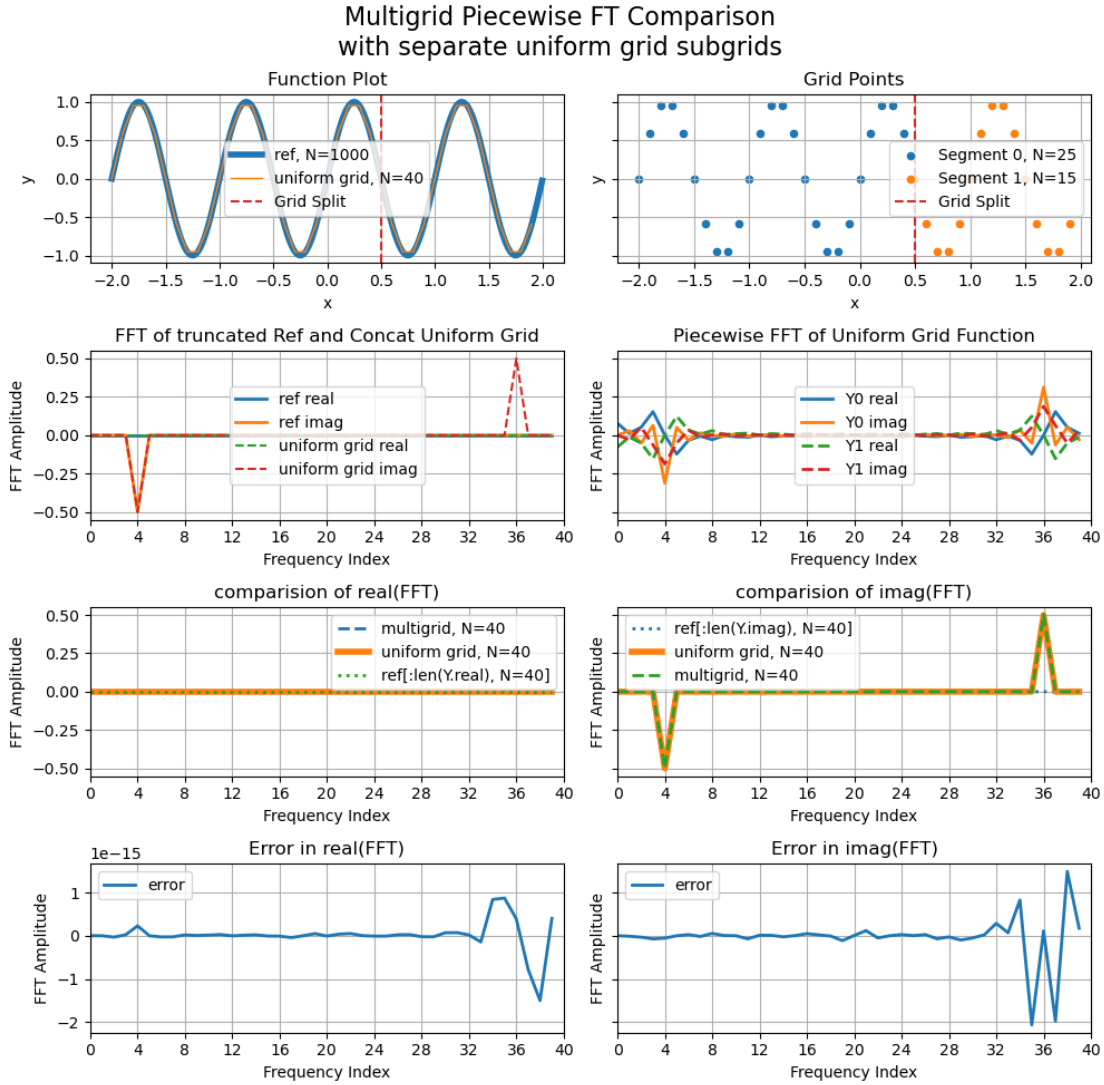


Figure 4.4.: Proof of concept: Multigrid piecewise Fourier Transform on uniform grid.

Fig. 4.5 explicitly shows the segments of the different uniform grids.

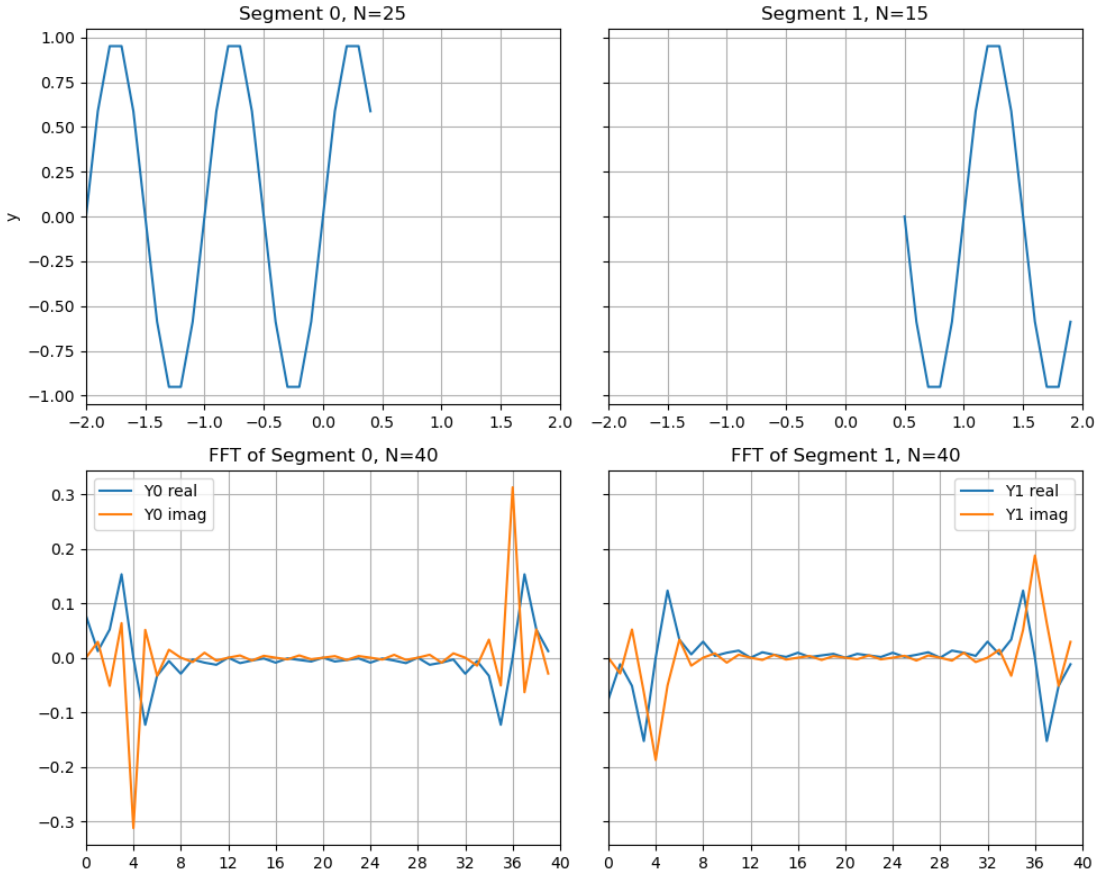


Figure 4.5.: Proof of concept: Multigrid piecewise Fourier Transform on uniform grid, showing the segments.

However, when the grids have different grid spacings, the discrete version of the time shift and stretch properties does not hold. The properties rely on a linear mapping between array index and grid spacing. Due to the overlapping grids of different grid spacings, eventually, a $+1$ step in array index will correspond to a different $+\Delta x$ in grid spacing at the grid boundaries.

Fig. 4.6 shows the method on two uniform grids, but with different grid spacings. The left half is a coarse grid and the right half is a fine grid. The method fails.

Fig. 4.7 shows the segments of the different uniform grids.

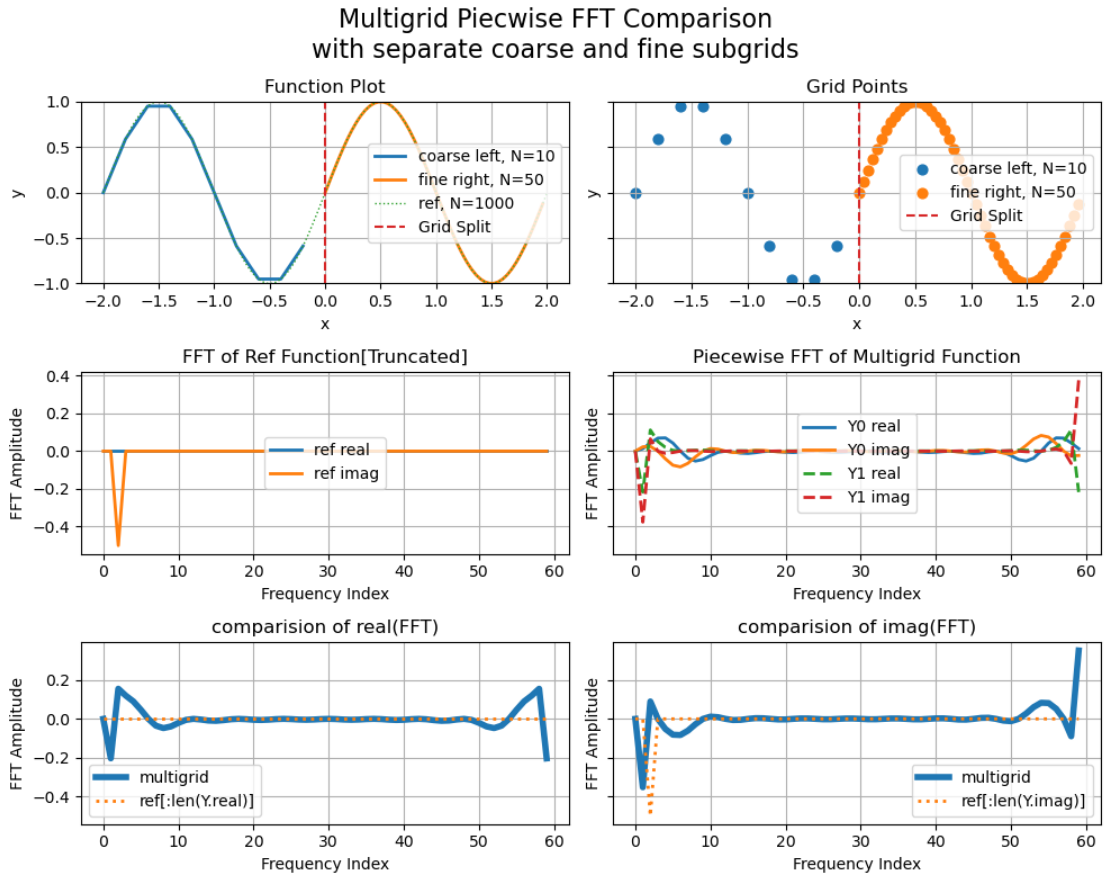


Figure 4.6.: Proof of concept: Multigrid piecewise Fourier Transform on uniform grids with different spacing.

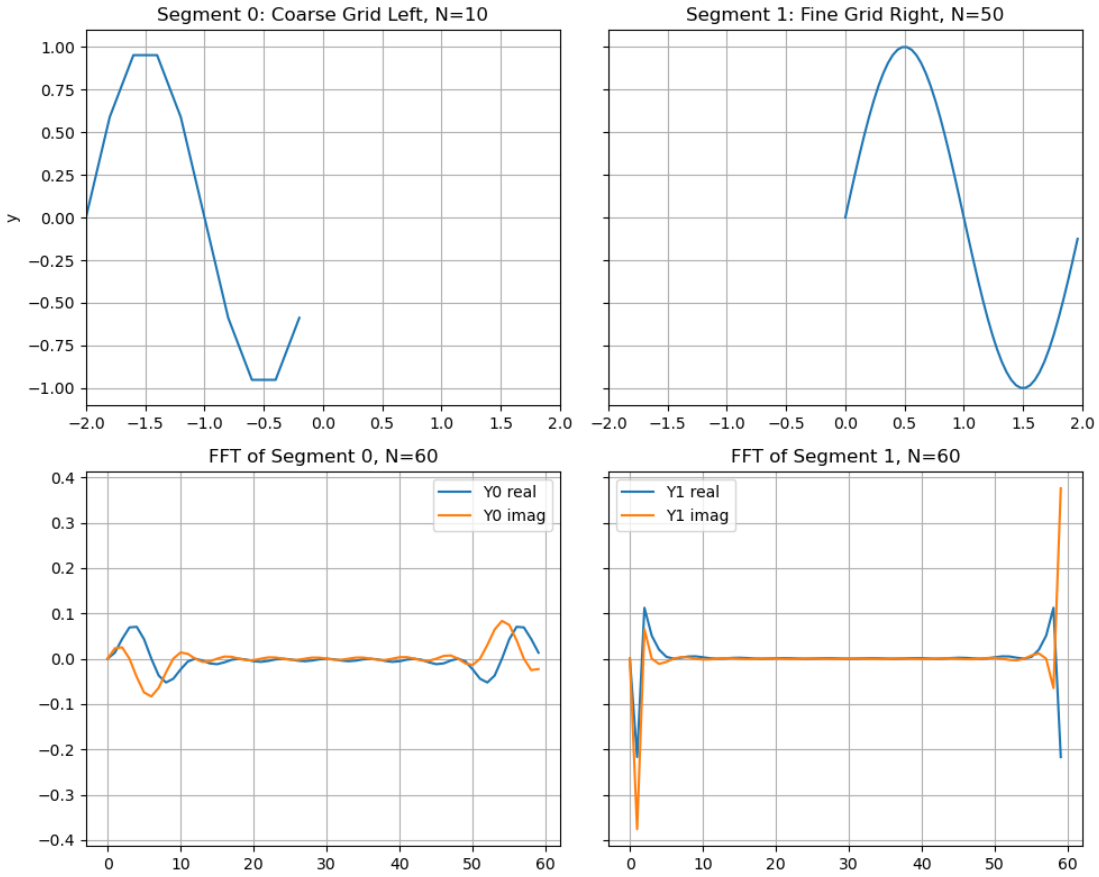


Figure 4.7.: Proof of concept: Multigrid piecewise Fourier Transform on uniform grids with different spacing, showing the segments.

Sampling Theorem Explanation of Method Failure

This proposed method fails when viewed from the perspective of the sampling theorem. The normal uniform grid sampling theorem relies on a very fragile property sometimes known as the "picket fence miracle". The property is that the Fourier Transform of a train of delta functions (i.e., a uniform grid) is another train of delta functions (i.e., a uniform grid in the Fourier domain). The grid spacing is inversely proportional to the grid spacing in the Fourier domain. When the sampling is at different grid spacings, the Fourier transform of the delta functions at the sample points is no longer a clean picket fence. Spectral leakage that occurs due to the non-uniformity.

To convert a continuous signal to discrete samples, the continuous signal is point-wise multiplied in real-space by a train of uniformly spaced delta functions. This is known as an impulse train or a Dirac comb.

The Dirac comb looks like:

$$\mathrm{III}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{4.1}$$

where T is the (uniform) spacing of the Dirac comb.

The discrete samples $x[n]$ are picked out by the integral and delta function. The below equation shows a single discrete point, which comes from a single delta function from the Dirac comb.

$$x[n] = \int_{-\infty}^{\infty} f(t)\delta(t - nT)dt \tag{4.2}$$

By the convolution theorem, a point-wise multiplication in real-space is a convolution in Fourier space. Thus one must convolve the Fourier transform of the signal $f(t)$ with the Fourier transform of the Dirac comb.

The Fourier transform of a single delta function is simple, as it only picks out $t = nT$

$$\mathcal{F}\left[\delta(t - nT)\right] = \int_{-\infty}^{\infty} \delta(t - nT)e^{-j2\pi ft}dt = e^{-j2\pi f \cdot nT} \tag{4.3}$$

By linearity, the Fourier transform of the Dirac comb is a sum of exponentials.

$$\begin{aligned}
\mathcal{F}\left[\mathrm{III}(t)\right](f) &= \mathcal{F}\left[\sum_{n=-\infty}^{\infty} \delta(t - nT)\right](f) \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - nT)e^{-j2\pi ft}dt \\
&= \sum_{n=-\infty}^{\infty} e^{-j2\pi f \cdot nT}
\end{aligned}$$

This is evaluated with a Poisson summation. This technique allows one to relate the sum of real space to the sum over Fourier space (aka dual space). The result is that a

Fourier transform of a Dirac comb with spacing T is another Dirac comb with spacing $1/T$. This is sometimes known as the "picket fence miracle".

$$\mathcal{F}\left[\sum_{n=-\infty}^{\infty}\delta(t-nT)\right](f)=\frac{1}{T}\sum_{k=-\infty}^{\infty}\delta\left(f-\frac{k}{T}\right) \tag{4.4}$$

When the Dirac comb is convolved with the Fourier transform of the signal $\mathcal{F}\left[f(t)\right]$, the Fourier transform will be placed at the impulse locations. The spectrum of $f(t)$ is essentially duplicated at each impulse location. See Fig 4.8 for a visual demonstration. The slight differences in magnitude are a numerical artifact, due to a finite number of frequency modes being used ($N_f = 2000$). If N_f was allowed to go to infinity, the Fourier transform of the Dirac comb would be a perfect picket fence.

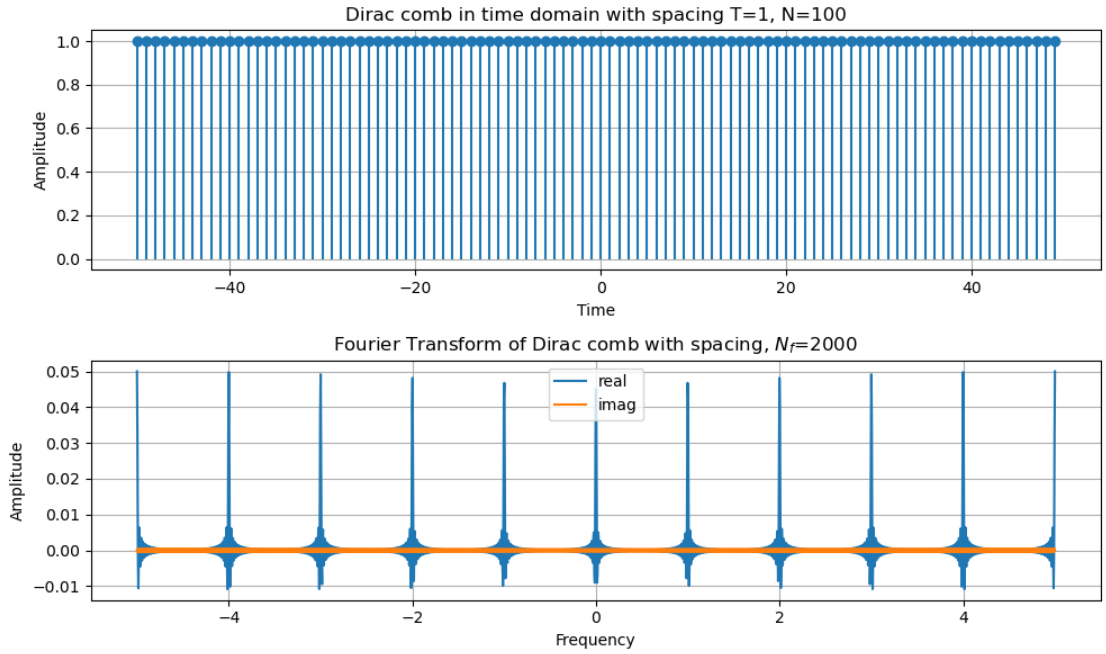


Figure 4.8.: The Fourier transform of a uniform Dirac comb is another uniform Dirac comb, which allows the spectrum to be duplicated at the impulse locations.

However we want to combine uniform grids of different grid spacing. From the sampling theorem perspective, this means the spacing between the delta functions are different. For generality, let's assume each spacing τ_n could be different.

The uneven Dirac comb is

$$\mathrm{III}_u(t)=\sum_{n=-\infty}^{\infty}\delta(t-\tau_n) \tag{4.5}$$

The non-uniform discrete samples are:

$$x[n] = \int_{-\infty}^{\infty} f(t)\delta(t - \tau_n)dt \tag{4.6}$$

The Fourier transform of this uneven Dirac comb is a sum of exponentials.

$$\begin{aligned} \mathcal{F}\left[\mathrm{III}_u(t)\right] &= \mathcal{F}\left[\sum_{n=-\infty}^{\infty} \delta(t - \tau_n)\right](f) \\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - \tau_n) e^{-j2\pi f t} dt \\ &= \sum_{n=-\infty}^{\infty} e^{-j2\pi f \cdot \tau_n} \end{aligned} \tag{4.7}$$

But this time, we don't have a nice symmetry that allows us to relate the real space to the dual-space. So we can't apply the Poisson summation. We're left with this sum of exponentials.

For Fig 4.9, let's assume the left half plane has a coarse spacing T_c and the right half plane has a fine spacing T_f.

$$\mathrm{III}_u(t) = \sum_{n=1}^{\infty} \delta(t + nT_c) + \sum_{m=1}^{\infty} \delta(t - mT_f) \tag{4.8}$$

If a copy of the spectrum of $f(t)$ is placed at each impulse location, the sum of the spectrums will overlap in a non-uniform way which causes spectral leakage. The Fourier Transform from the non-uniform grid will not be an accurate representation of the continuous signal.

This phenomenon can be used to generally explain the difficult of performing a Fourier transform on a non-uniform grid. There will inevitably be some distortion in the spectrum caused by the non-uniform sampling.

The slight differences in magnitude are a numerical artifact, due to a finite number of frequency modes being used ($N_f = 2000$). If N_f was allowed to go to infinity, the magnitude of all the peaks would be the same, except where the two picket fences overlap. Then the magnitude would be two-times the normal magnitude due to the overlapping peaks.

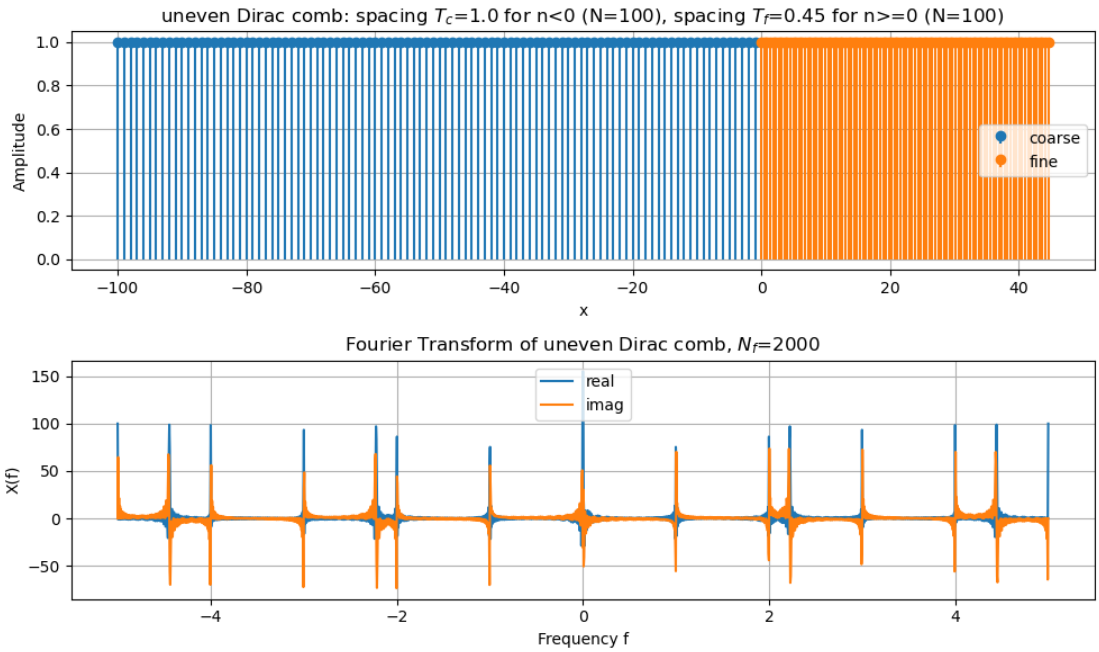


Figure 4.9.: The Fourier transform of an uneven Dirac comb is a sum of exponentials, which does not have a nice symmetry.

4.2. Non-Uniform Fourier Transform (NUFT)

4.2.1. NUFT Workflow

The proposed NUFT workflow is to first use a forward transform to compute the Fourier space spectrum of the non-uniform samples. The sequences are point-wise multiplied in Fourier space. Then a backward transform is used to reconstruct the real-space values at non-uniform target coordinates. This is the same as the standard convolution theorem method to speedup the convolution, except the Fourier transforms are non-uniform. Note that the backward transform is not the exact inverse of the forward transform (see section 2.3 for details).

When the convolution is between two signals on different grids, the original non-uniform samples and final target non-uniform samples may be different. This is the case when transitioning between the G and W to calculate Σ, seen in Eqn 1.3.

4.2.2. Testing Methodology

To evaluate the accuracy of the non-uniform Fourier transform, a simple forward and backward transform was performed. The full convolution workflow was also tested, but the transform results produced too much error to make meaningful conclusions. Thus the simpler test of only forward/backward transforms were used. An adaptive grid is produced on the test data using the gradient monitor. The data is forward transformed

and the Fourier space spectrum is examined. Then the spectrum is backward transformed into real-space to produce the reconstructed signal. The reconstructed signal is compared to the original signal. The metric used to evaluate the error is the normalized ℓ_2 error (see Eqn. 2.17).

The datasets used were a sinusoid signal ($N = 200$), a triple Lorentzian signal ($N = 200$), and intermediate saved Quatrex data from the SCBA loop ($N = 2001$, $\Delta E = 10$ meV). Four sets of transform results are shown: FFT, NUFFT via the `finufft` library, Voronoi weighted NUDFT, and interpolation method. All transforms use the same number of Fourier modes as the number of samples ($M = N$). A quick summary is provided in Table 4.1.

4.2.3. FFT Reference results

The FFT results use the same number points as the other method, but on a uniform grid. They also transform to the same number of Fourier modes. The error is very small, on the order of machine precision (10^{-15}), which is expected since the FFT is an exact transform on a uniform grid.

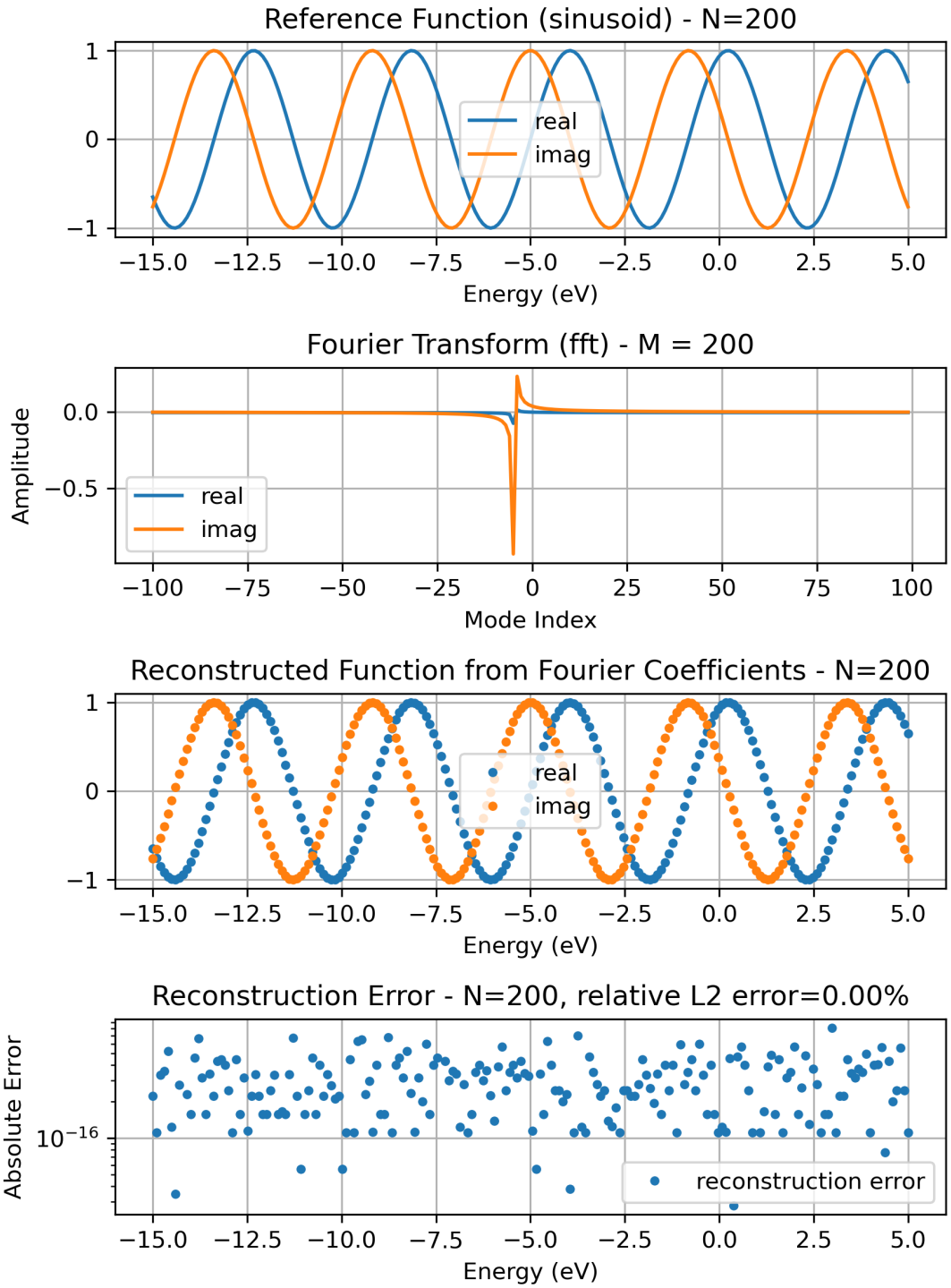


Figure 4.10.: Demonstration of the FFT on a sinusoid signal.

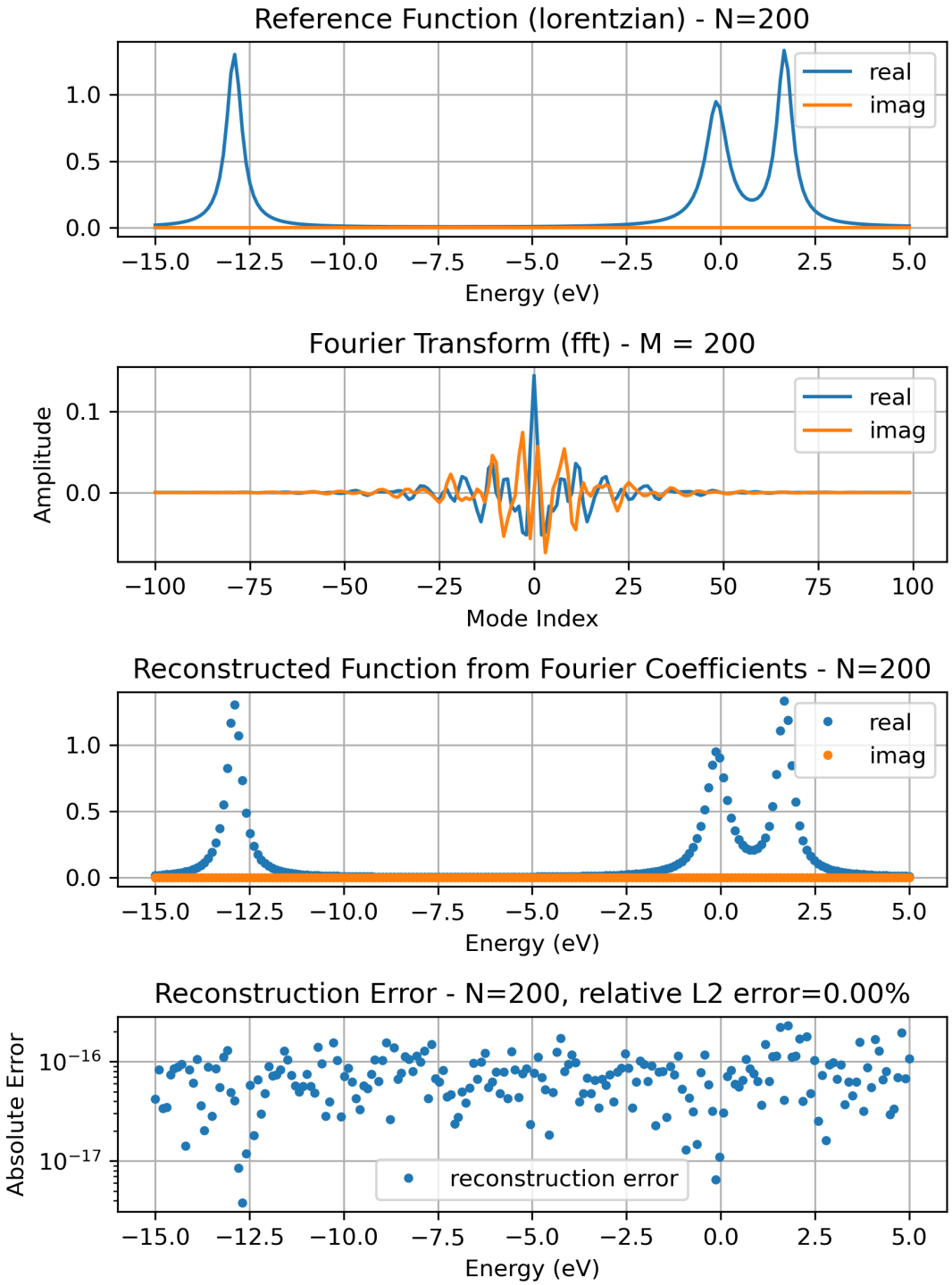


Figure 4.11.: Demonstration of the FFT on a triple Lorentzian signal.

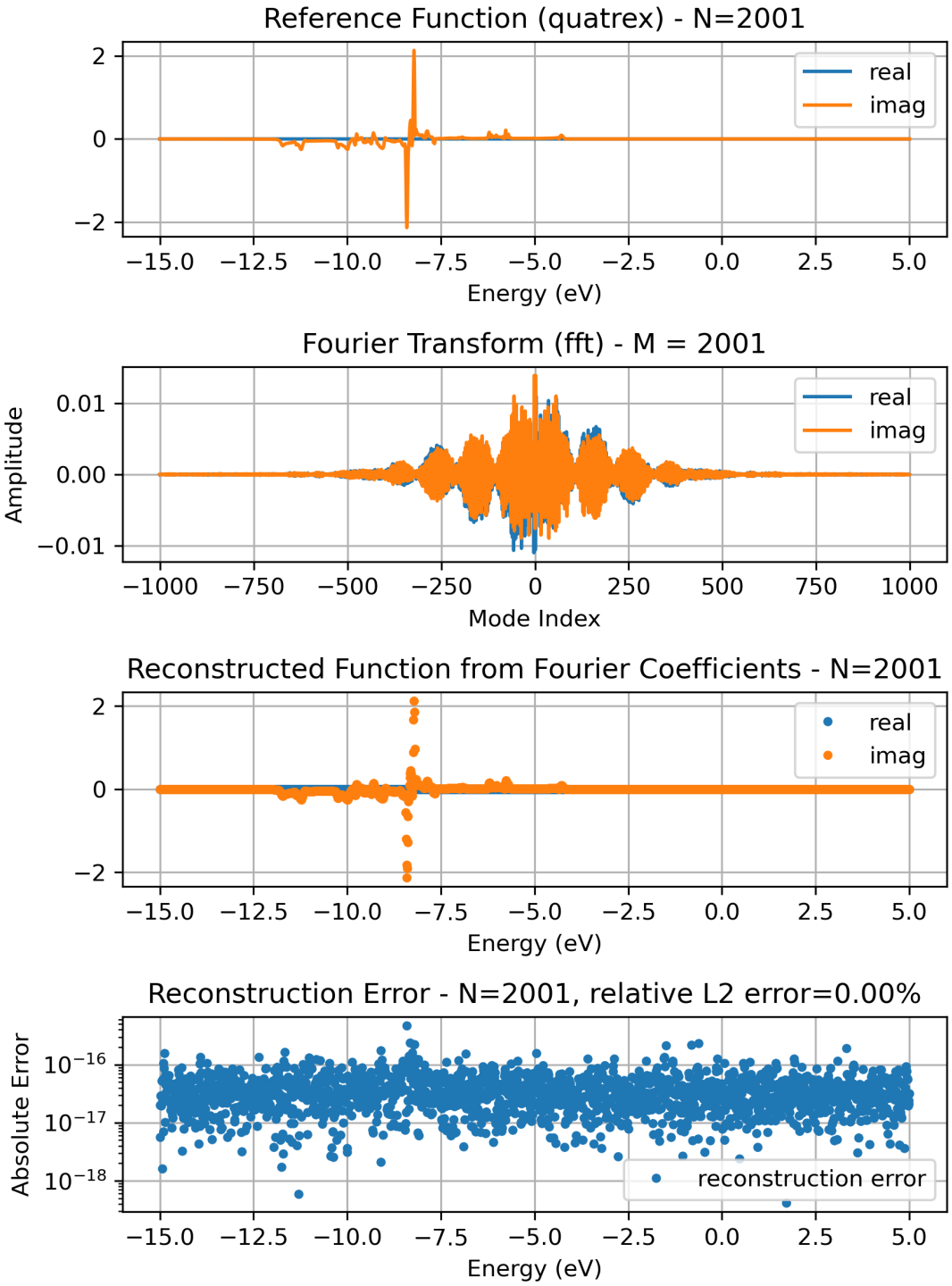


Figure 4.12.: Demonstration of the FFT on intermediate saved Quatrex data from the SCBA loop.

4.2.4. NUFFT Libraries

Several NUFFT libraries were tested to perform the Fourier transforms on the non-uniform grid.

Name	Authors / Institution	Technical Details	Notes
`PyNUFFT` [13]	Jyh-Miin Lin; Univ. of Cambridge; NTU Taiwan	implements min-max interpolator [14]	Fourier-space convolution gave incorrect result
`finufft` [4] [15]	Alex Barnett; Flatiron Institute	Exponential of semi-circle kernel	Great documentation; large team; frequent updates
`NFFT` [16]	Keiner (Lübeck); Kunis (Osnabrück); Potts (Chemnitz)	C with MATLAB interface	Python wrapper is very sparse
Julia Fast Transforms [17]	Ruiz-Antolín (Cantabria); Townsend (Cornell)	C with Julia wrapper; low-rank approx instead of spreading kernel	Same results as `finufft`

The `PyNUFFT` library produced incorrect results when performing the Fourier-space convolution. The documentation did not clearly specify the expected input domain and output range of the functions. The `finufft` library was extremely well-documented and produced correct results. The `NFFT` library was tested in the MATLAB interface, because the Python wrapper was very sparse and seemed to have little support. The Julia Fast Transforms library was tested because it had a different approach to the spreading kernel, using a low-rank approximation instead of an interpolation-based spreading kernel. It produced the same results as the `finufft` library.

In the end, the `finufft` library was selected. It was the most user-friendly, had the best documentation, and had an existing Python wrapper to its C++ implementation with GPU support. This work does not use the GPU interface, but it is a desirable feature for future work.

Fig. 4.13 shows a demonstration of the `finufft` on a sinusoid signal. The locations of the 200 non-uniform sample are computed from the gradient monitor. The points are slightly denser at the peaks of the sinusoid, which is expected since the gradient is larger at the peaks. The Fourier transform correctly captures the frequency peak at $f = 1$, but the magnitude of the peak is slightly smaller than the expected $|c| = 1$.

The reconstructed signal is mostly accurate, except for a large error at both end points. We are not sure why the error is large at the end points, but it may be due to the specific implementation of the `finufft` library. This only occurs for pure sinusoidal signals. Note that care must be taken to ensure the input coordinates are remapped to the range $[-2\pi, 2\pi)$, as the `finufft` library expects the input coordinates to be in that range.

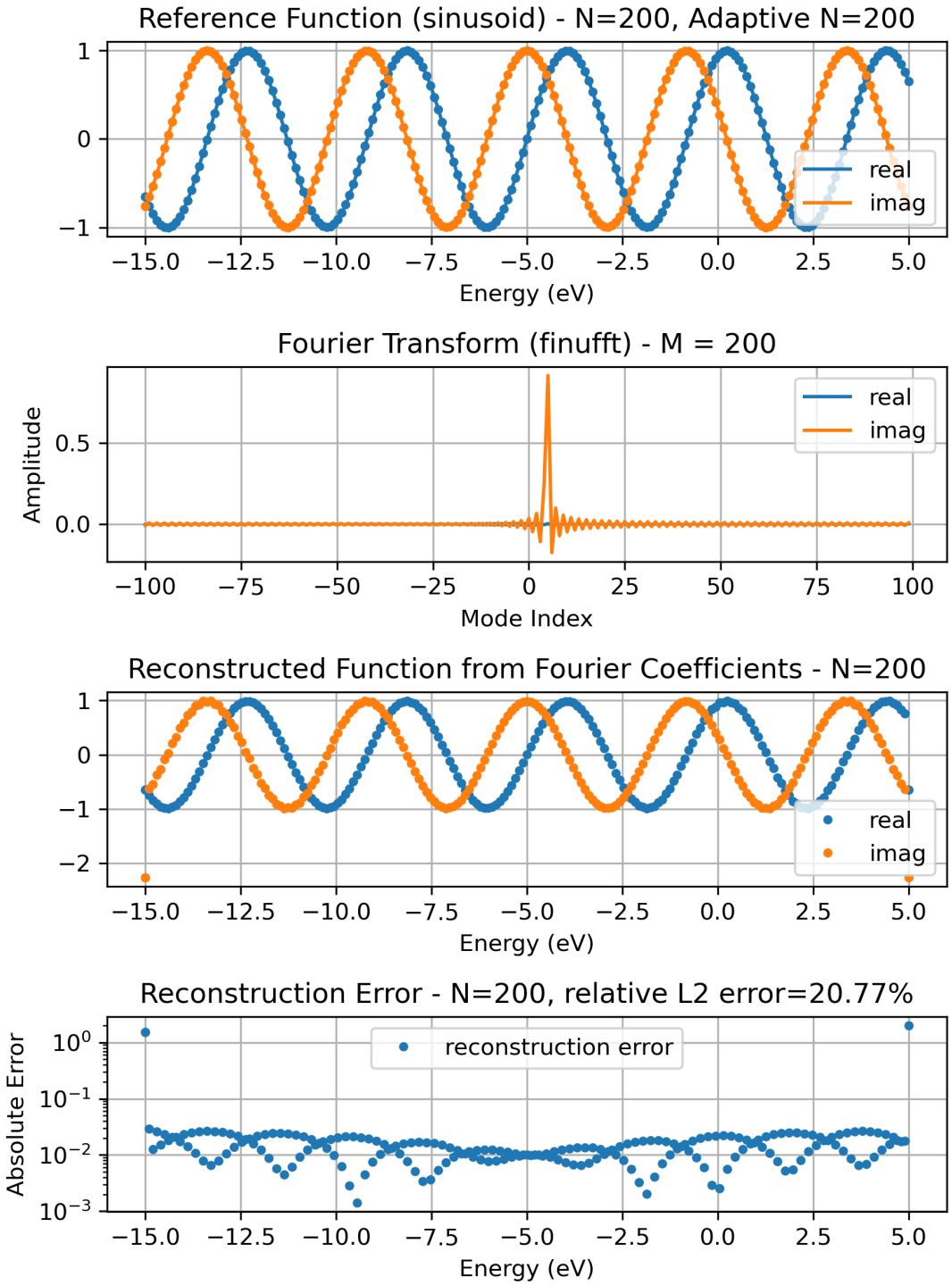


Figure 4.13.: Demonstration of the `finufft` on a sinusoid signal.

The sinusoidal signal is a simple test case, with only one Fourier mode present. It is not representative of the signals of interest in Quatrex, which have more complex spectra. As an intermediate step, the workflow was applied to a triple Lorentzian signal, which is more similar to Green's function signals in Quatrex. The spectrum of a Lorentzian is a decaying exponential, which requires many Fourier modes to accurately capture.

The error is much larger than the sinusoid case, especially at the peaks of the Lorentzian. The locations of the peaks are captured in the reconstruction, but the magnitudes are not accurate. The error can actually be improved by using more Fourier modes in the forward transform. Currently $M = N$, which means the number of Fourier modes is the same as the number of non-uniform samples. Increasing M to be larger than N can improve the accuracy.

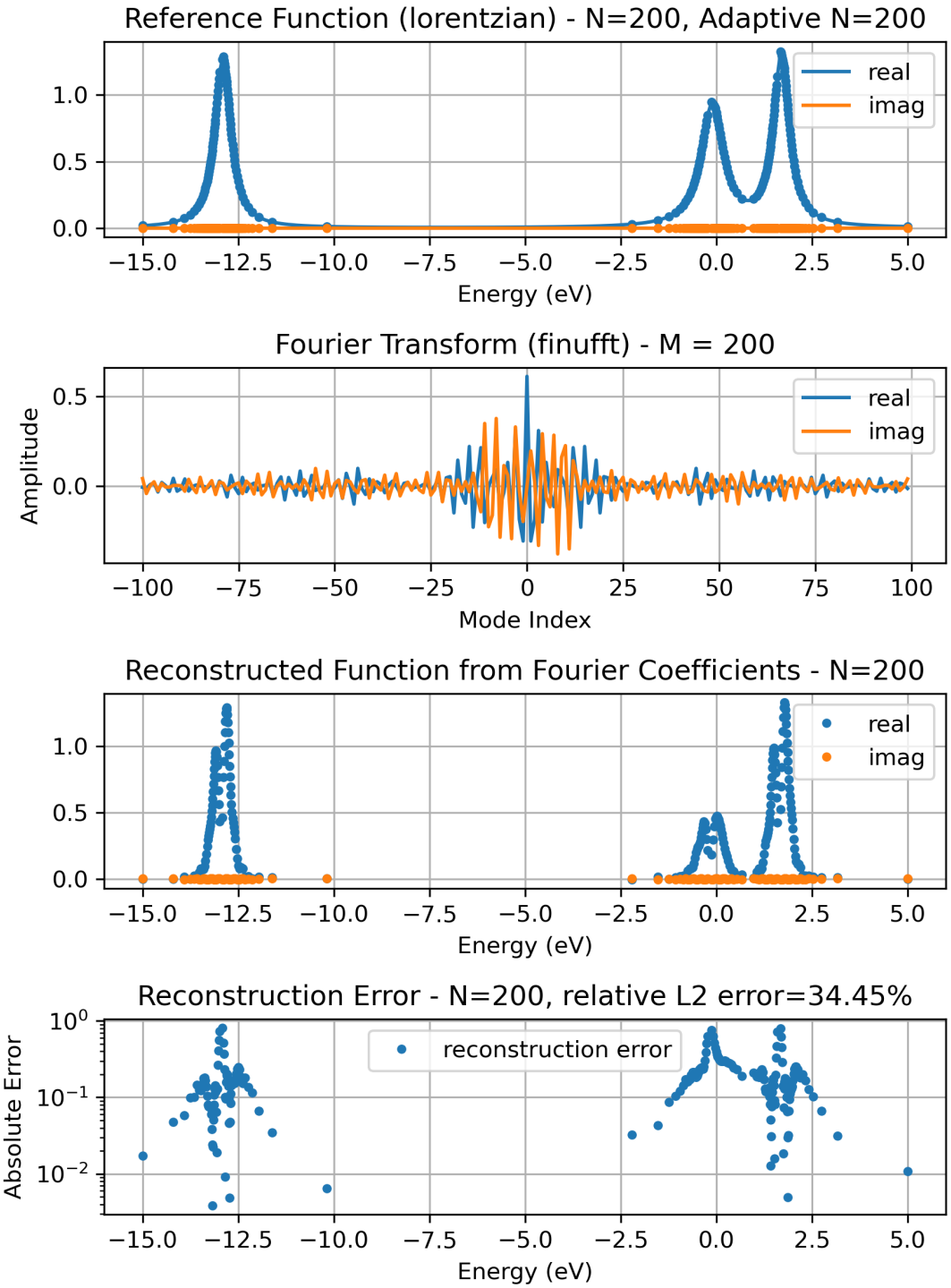


Figure 4.14.: Demonstration of the `finufft` on a triple Lorentzian signal.

For example, Fig. 4.15 shows $M = 10N$. The improvement is modest, going from 34.45% to 19.86% with a 10x more points. An error of less than 1% is desired. However using $M = 100N$, the error is still 3.44%. Going up to $M = 1000N$ reduces the error to 0.89%. To achieve the desired error, a 1000x increase in the number of Fourier modes is needed. This directly corresponds to 1000x increase in computational cost, which is not practical.

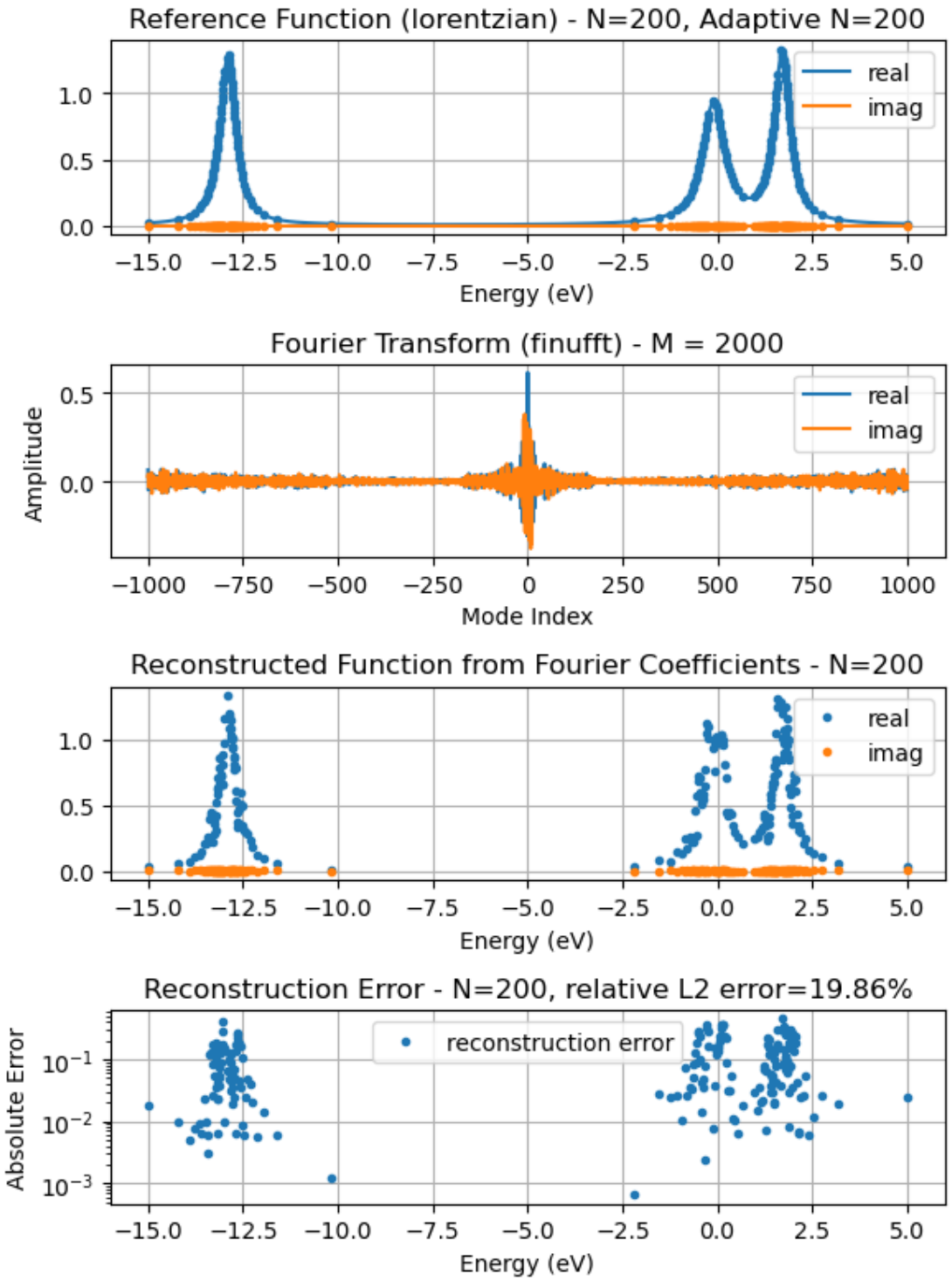


Figure 4.15.: Demonstration of the `finufft` on a triple Lorentzian signal, but $M = 10N$. The error is improved by using more Fourier modes in the forward transform.

4.2.5. Voronoi weights

It was observed that the non-uniform sampling causes spectral power distortions in the Fourier domain. This is inherently due to the non-uniform sampling. Some regions are sampled more densely than others, which causes the Fourier transform to have more power in those regions. Voronoi weights were applied to the NUDFT to partially fix the power distortion (see 2.3.1). Fig. 4.16 shows the effect of the Voronoi weights on the spectrum of a double Lorentzian signal.

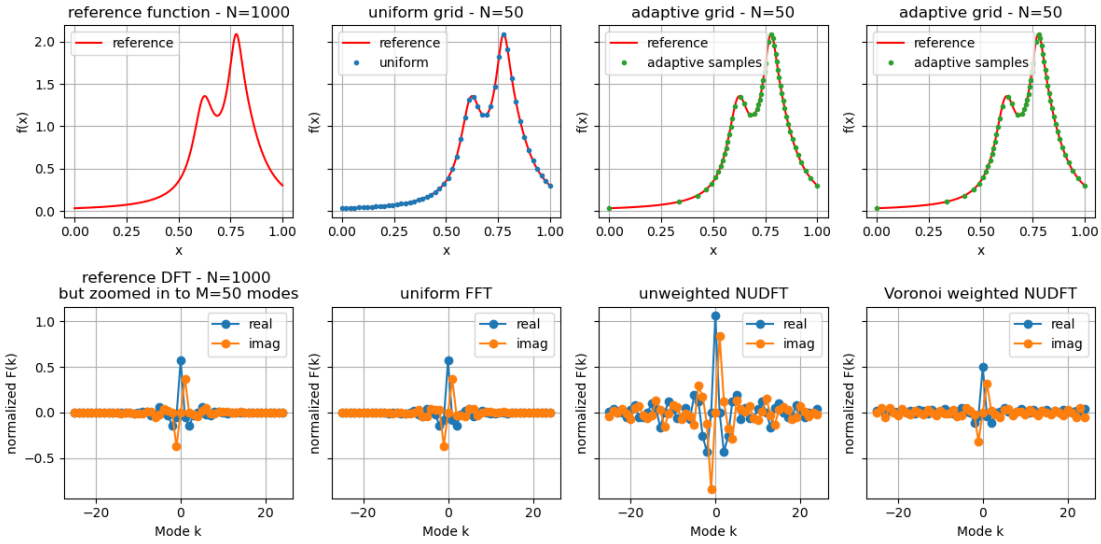


Figure 4.16.: Voronoi weights improve the power spectrum.

The effect of fixing the power distortion can be inspected using Parseval's theorem (see 2.1.4). Fig. 4.17 shows the effect of the Voronoi weights on the energy. We see that the unweighted NUDFT tends to overestimate the power in the Fourier domain. The Voronoi weights tend to lower the power, which is more accurate to the real energy. However, adding more points causes the Voronoi weighted power spectrum to drift upward. Note that it is coincidence that using 200-500 adaptive points is closest to the reference energy.

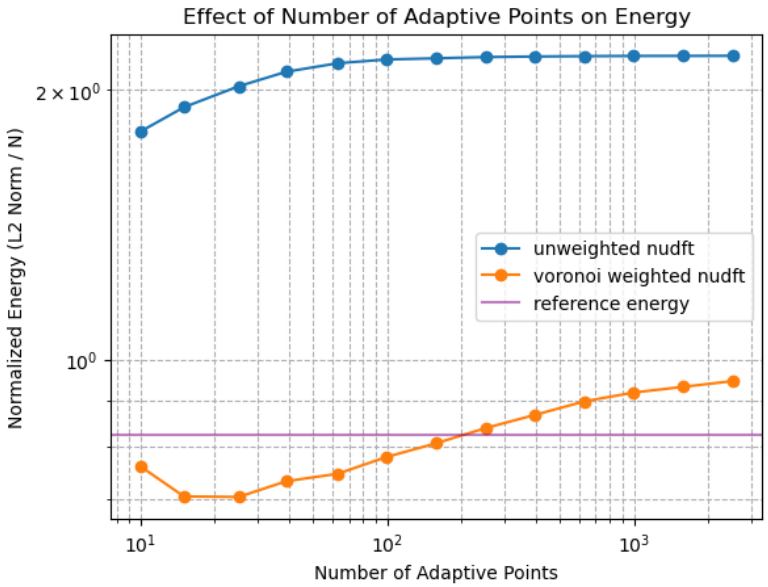


Figure 4.17.: Voronoi weights improve the power spectrum, which can be inspected using Parseval's theorem. The reference function used is the same as Fig. 4.16.

The Voronoi weights were only applied to the forward transform. The backward transform is unweighted, as the spectrum is assumed to be corrected already. The backwward transform was also computed in the discrete form (see Eqn. 2.19), instead of relying on the `finufft` library. The results are better than the unweighted NUFFT, but the error is still quite large. Note that the NUDFT takes much longer to compute, since it is a direct summation (two nested for loops) and cannot be accelerated with the existing NUFFT libraries.

The results for a sinusoid are shown in Fig. 4.18.

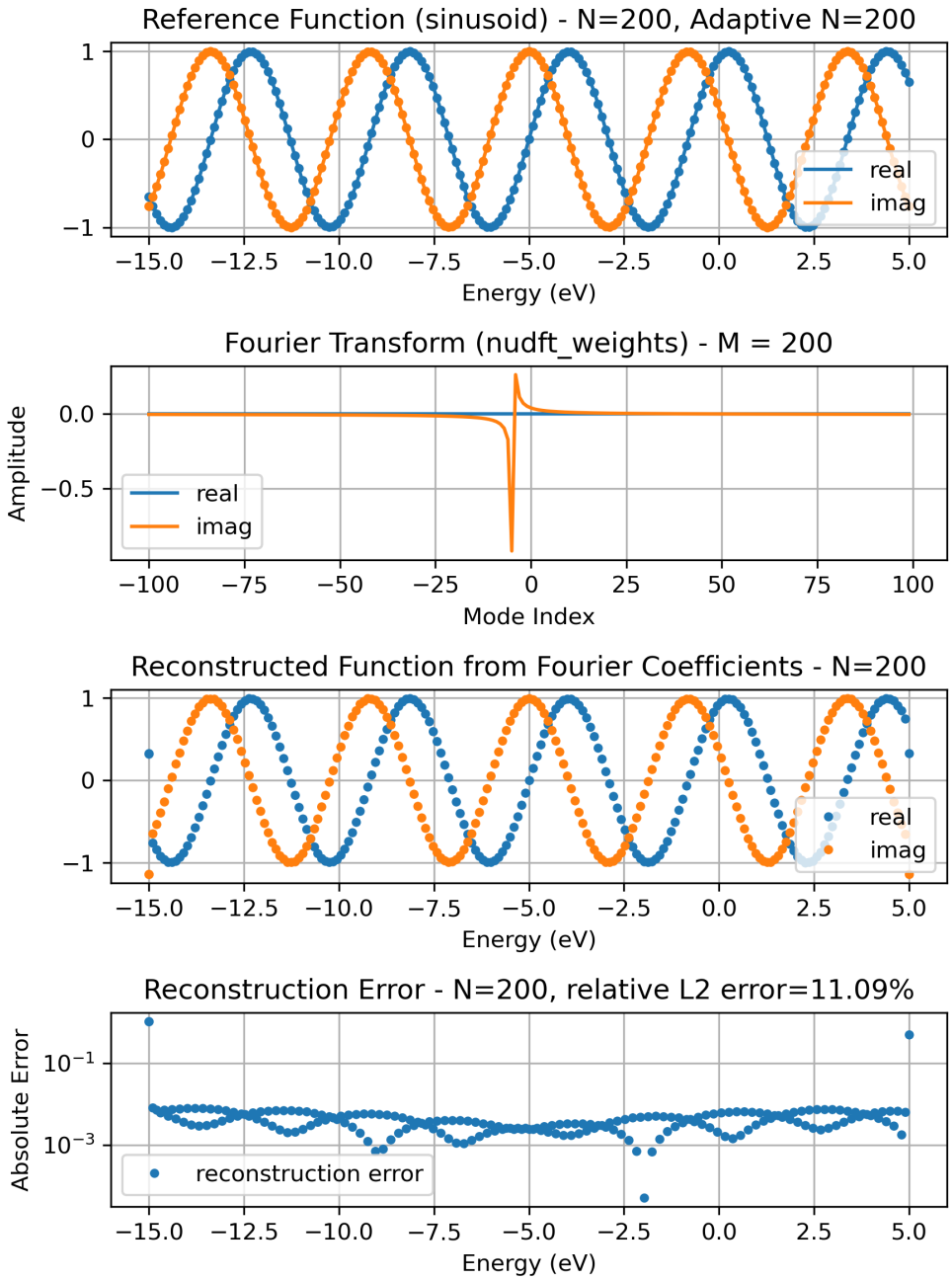


Figure 4.18.: Demonstration of the Voronoi weighted NUDFT on a sinusoid signal.

The results for a triple Lorentzian are shown in Fig. 4.19.

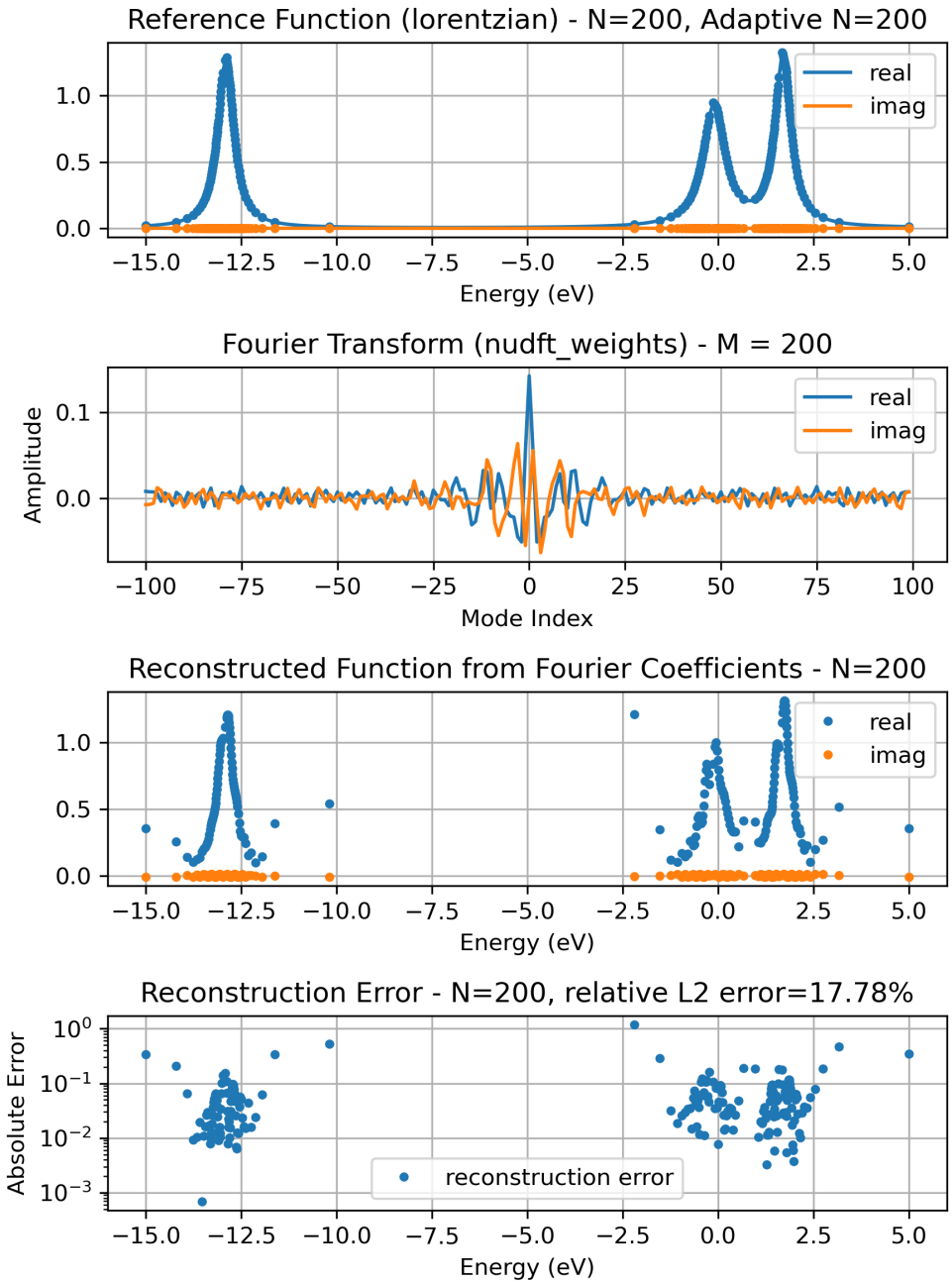


Figure 4.19.: Demonstration of the Voronoi weighted NUDFT on a triple Lorentzian signal.

The results for intermediate saved Quatrex data from the SCBA loop are shown in Fig. 4.20.

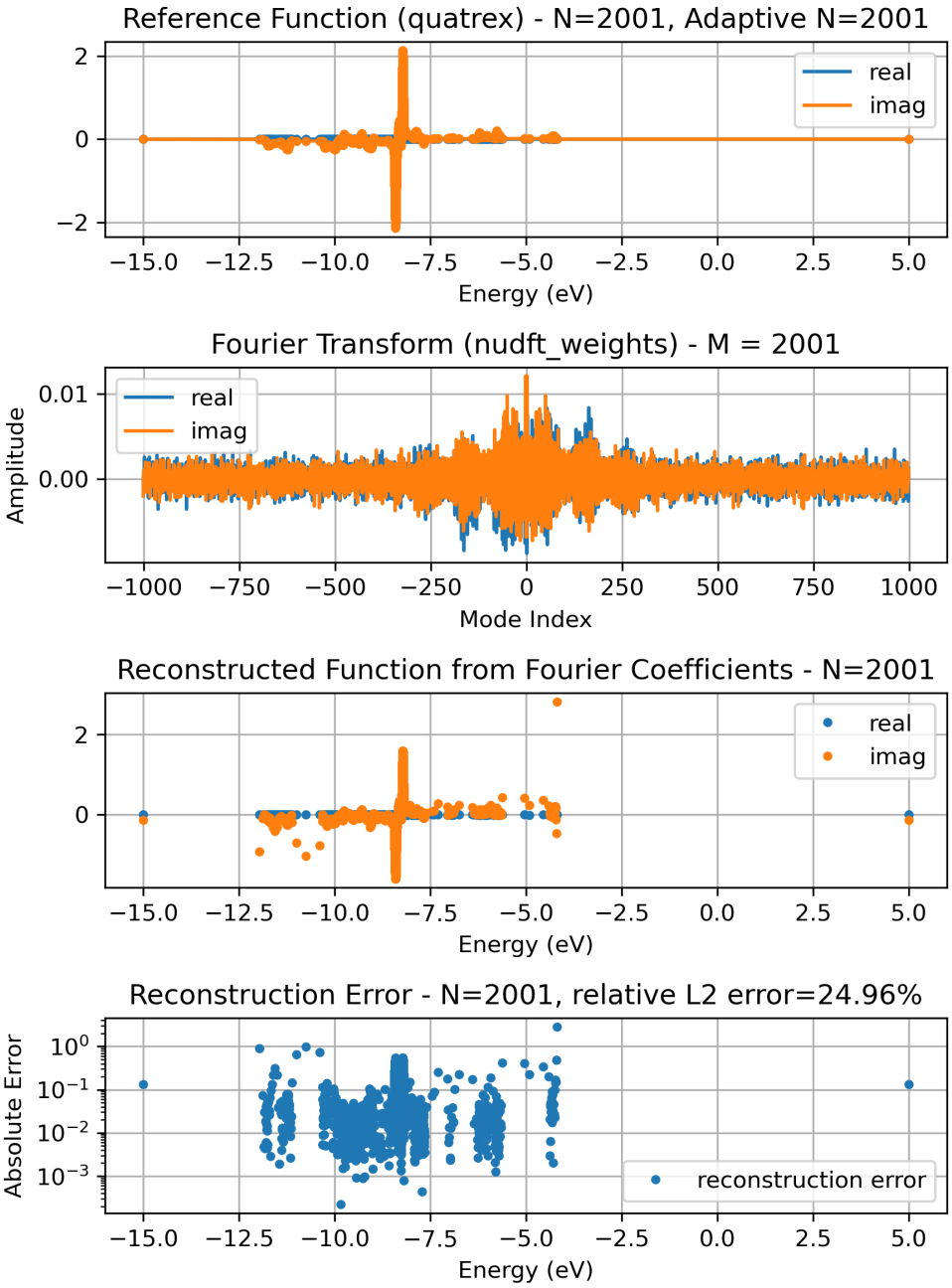


Figure 4.20.: Demonstration of the Voronoi weighted NUDFT on intermediate saved Quatrex data from the SCBA loop.

From an accuracy perspective, the Voronoi weights will not work in Quatrex. A normalized ℓ_2 error of $\approx 1\%$ is desired, but the error is much larger than that.

There is also no easy way to accelerate the weighted NUDFT. The existing NUFFT libraries do not support weighted transforms. The addition of the weights completely changes the formalism of the NUFFT spreading kernel and cannot be easily adapted to the existing algorithms. A custom implementation of the weighted NUDFT would be required to accelerate the computations. Due to the poor results, this was not pursued further

Thus, the Voronoi weights were not used in the final implementation.

4.3. Interpolation

The interpolation method was ultimately selected for implementation in Quatrex. It does not perform a non-uniform Fourier transform. Instead, it interpolates the non-uniform samples onto a fine uniform grid. Then the normal convolution theorem method is used to perform the convolution (FFT, point-wise multiplication, inverse FFT). The last step is to map the fine grid result on the target non-uniform grid of the resulting variable. This is also done by interpolation.

For example, when calculating Σ from G and W in Eqn. 1.3, the original non-uniform samples are from G and W, and the target non-uniform samples are for Σ. The samples from G and W are interpolated onto a common fine uniform grid, the convolution is performed with the FFT, and then the result is projected back to the non-uniform target coordinates for Σ.

The interpolation step is a "fill-in" step, where the non-uniform samples are filled in to a uniform grid. The ratio of the number of points in the uniform grid to the number of non-uniform samples is a hyperparameter that can be tuned to improve the accuracy. We call this the "oversampling ratio" r. Thus, the number of points in the interpolated uniform grid is $N_{uniform} = r \cdot N_{adaptive}$.

By default, the fill-in uses linear interpolation. Higher order interpolation (ex. cubic) can also be used. But it was found that higher order interpolation usually increases error. When there are large flat gaps between two points, the higher order inteprolation connects the two points with a curve. But the true signal is flat in the gap, so the curve is an error. The linear interpolation just connects the two points with a straight line, which is more accurate.

Fig 4.21 shows a demonstration of the interpolation method on a triple Lorentzian signal. The oversampling ratio is set to $r = 5$. The sampling density of the peaks is decreased. But the points between the flat regions are filled in.

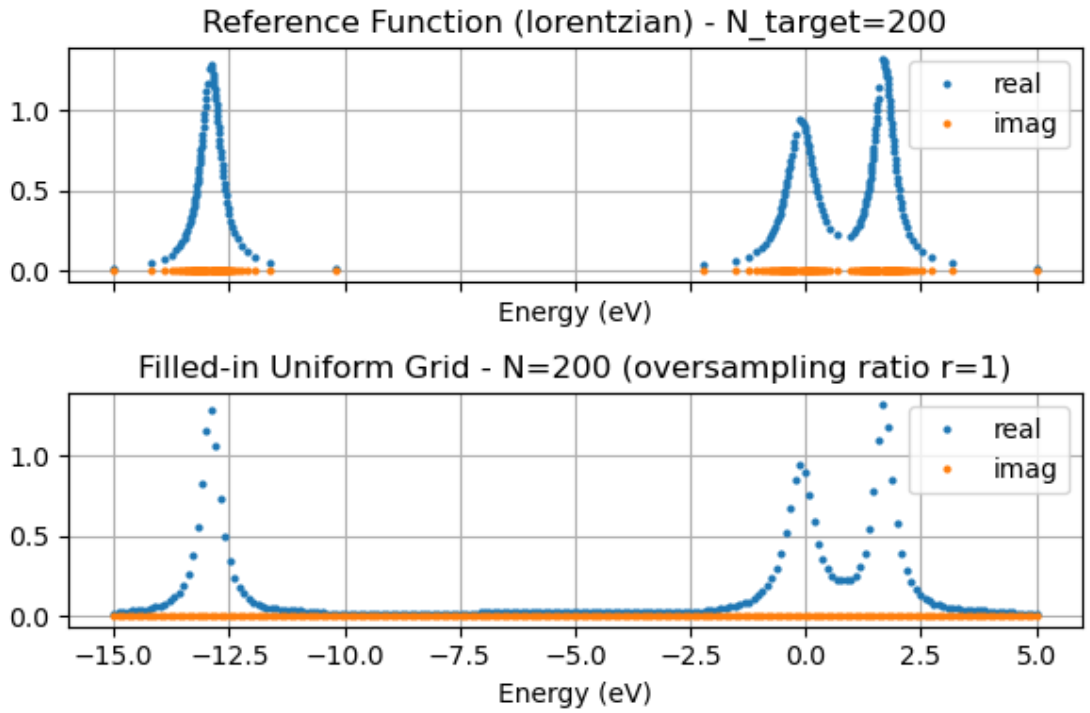


Figure 4.21.: Demonstration of the fill-in step of the interpolation method on Lorentzian data. For visualization purposes, the oversampling ratio is set to $r = 5$.

It was found that setting the oversampling ratio $r = 1$ is a good tradeoff between accuracy and computational time. As can be seen in Fig. 4.22, the error goes down as r increases. For Quatrex data specifically, the error is already quite small at $r = 1$. Further improvements to the error requires setting $r = 10$. A few experimental runs with $r = 5$ and $r = 10$ in the implemented Quatrex code showed that setting a higher r does not significantly change the results of the SCBA iterations or improve convergence. But it does noticeably increase the computational time, since the FFT is performed on a much larger grid. Thus $r = 1$ was used by default in the implementation.

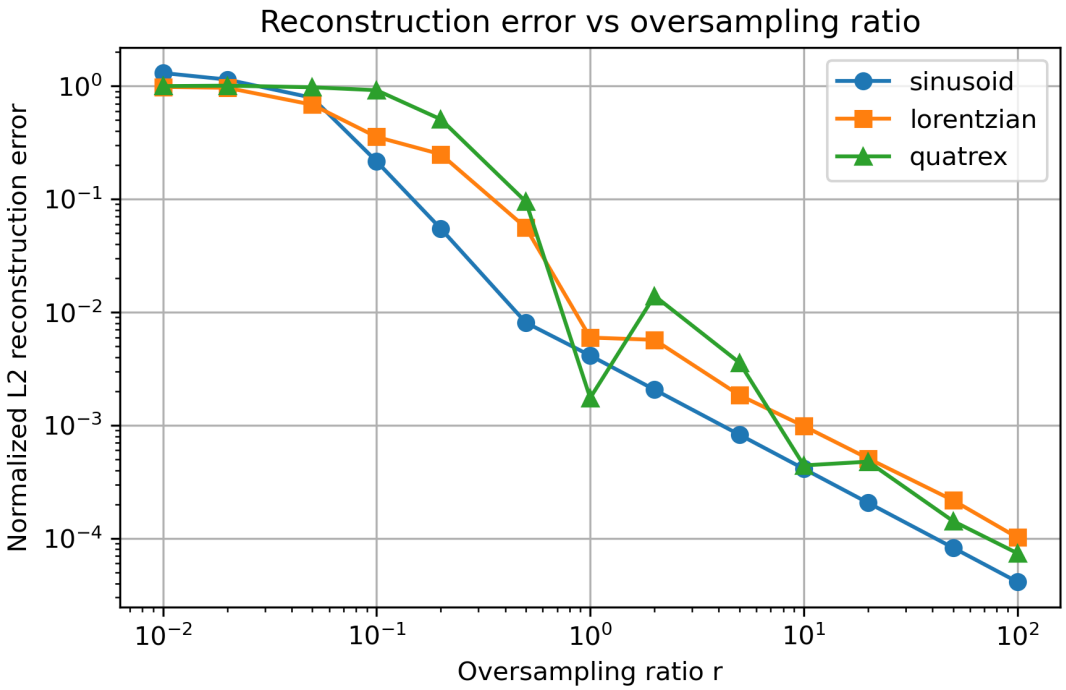


Figure 4.22.: The reconstruction error goes down as the oversampling ratio r increases.

For Quatrex data, the Fig. 4.23 shows the fill-in step of the interpolation method with $r = 1$. There are much fewer points in the peak regions, but there is still sufficient detail to capture the peak location and peak amplitude.

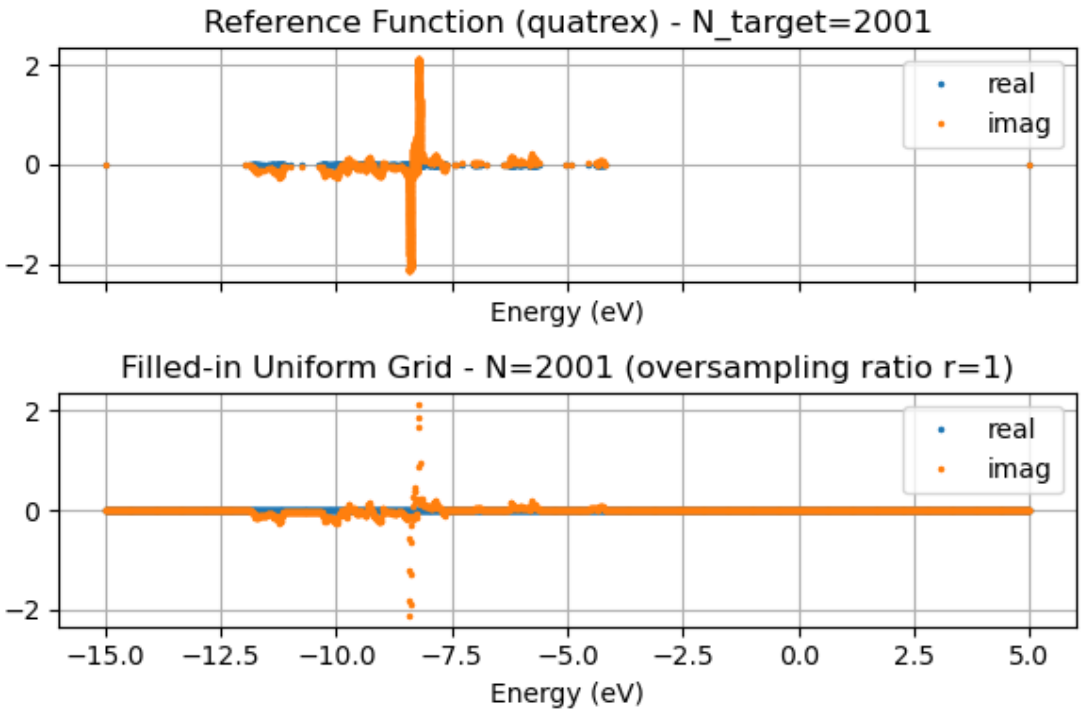


Figure 4.23.: Demonstration of the fill-in step of the interpolation method on Quatrex data. The oversampling ratio is set to $r = 1$.

4.4. Summary

The compressed sensing approach and the multigrid piecewise FFT approach were only exploratory approaches and not used. The NUFFT, weighted NUDFT, and interpolation approaches were compared in a microbenchmark against the standard FFT approach. The metric used was the ℓ_2 norm of the error. The results are summarized in the table 4.1 below. For NUFT methods, the Quatrex data is most difficult to accurately reconstruct. The Voronoi weights improve the error of the non-uniform Fourier transform for all datasets, but the error is still too large for use in Quatrex.

Method / Data Set	**Sinusoid**	**Lorentzian**	**Quatrex**
Uniform FFT	0.00%	0.00%	0.00%
Finufft	20.77%	34.45%	35.27%
Weighted NUDFT	11.09%	17.78%	24.96%
Interpolation	0.41%	0.60%	0.17%

Table 4.1.: Relative L2 reconstruction errors for different transforms and datasets.

Chapter 5

Implementation

The adaptive energy grid was implemented in Quatrex using an interpolation workflow. The simulation starts with a uniform grid. Then, after a user-specified amount of iterations, it switches to the adaptive grid. The adaptive grid is created at the start of the first adaptive iteration. The adaptive grid is computed using the absolute sum of all orbital indices as the input to the complexity monitor. This is either `np.sum(np.abs(G))` for the G/Σ grid or `np.sum(np.abs(P))` for the P/W grid. The original function values from the uniform grid are projected to the adaptive grid with B-spline interpolation. By default, the interpolation is of degree 1, which corresponds to linear interpolation.

When an integral is needed, a user-specified area rule is used to compute the integral over the adaptive grid. By default, the area rule is the trapezoidal rule, which computes the area of the trapezoids formed by adjacent points on the grid. An alternative option is Simpson's rule, which fits a parabola to every three adjacent points on the grid and computes the area under the parabola.

Figure 5.1 shows the interpolation workflow.

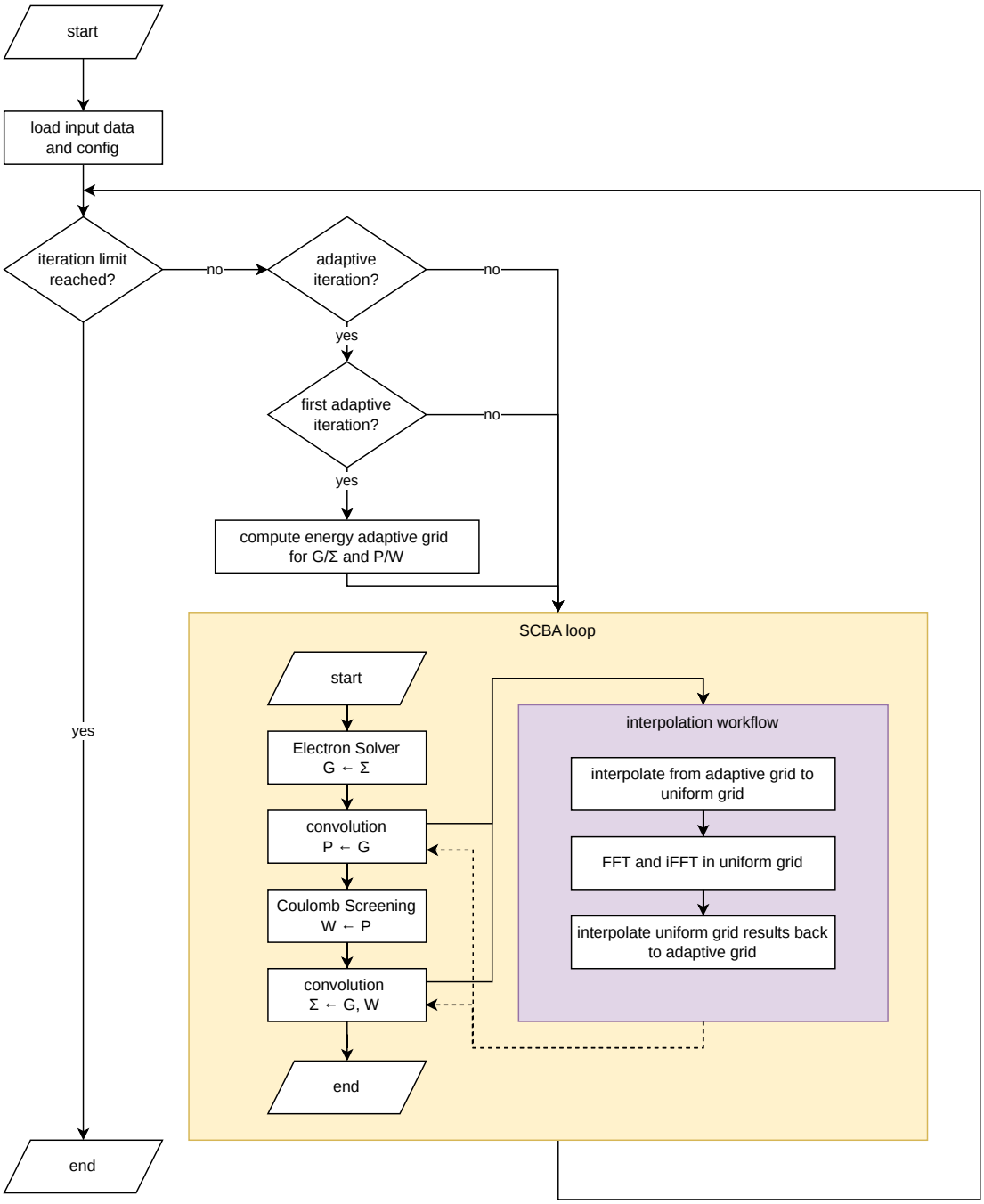


Figure 5.1.: Interpolation workflow.

Unless otherwise specified, the computations were run with the parameters in Table 5.1.

Parameter	Value
structure	carbon nanotube
potential difference between contacts	0.2 V
number of orbitals	768
energy window	-15 to 5 eV
number of energy points	2001 (10 meV energy resolution)
complexity monitor	absolute sum of all orbital indices
number of adaptive grid points	same as uniform grid
interactions	electron-electron (GW) and electron-phonon
interpolation degree	1 (linear interpolation)
oversampling ratio	1
area rule	trapezoidal rule
mixing factor	0.5

Table 5.1.: Default parameters for the simulations.

In testing, it was determined that a energy grid resolution of 10 meV was sufficient to capture the features of the signal and ensure that the signal is effectively band-limited (see Section 2.2.1).

The `scipy.interpolate.make_interp_spline function` [1] was use to perform the interpolation. The function creates a B-Spline object which can be evaluated at any point. The function allows batched input and output to parallelize the interpolation over multiple orbital indices. A gradient monitor was used to create the adaptive grid.

5.1. Separate grids for G/Σ and P/W

Each SCBA variable (G, P,W, Σ,) has slightly different features. In a perfect world, each variable would have its own grid optimized for its features. However, each time there is a grid crossing, we introduce interpolation error. Thus the number of grids should be minimized.

The Green's function G must have a grid adapted for its features, since it is the main variable that is being solved for. The polarization P is a convolution of G with itself, so the features and locations of features are different. The screened Coulomb potential W has features similar to P. P and W were deemed similar enough to share the same grid, although the grid is optimized and based on the features of P.

Although the features of the self-energy Σ are very different compared to G, it is used to update G. If there is a large Σ peak where there is no G feature, there are no states in that region to scatter. It does not affect G in the next iteration, so grid points in that region are unnecessary. Thus Σ is represented on the same grid as G.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.make_interp_spline.html

Note that the same adaptive grid is shared for all orbital-orbital interactions. In general, each orbital-orbital interaction has different features and would require a different grid. However this requirement is imposed due to the `ElectronSolver` interface of Quatrex, which requires all orbital-orbital interactions to share the same energy grid.

Chapter 6

Results

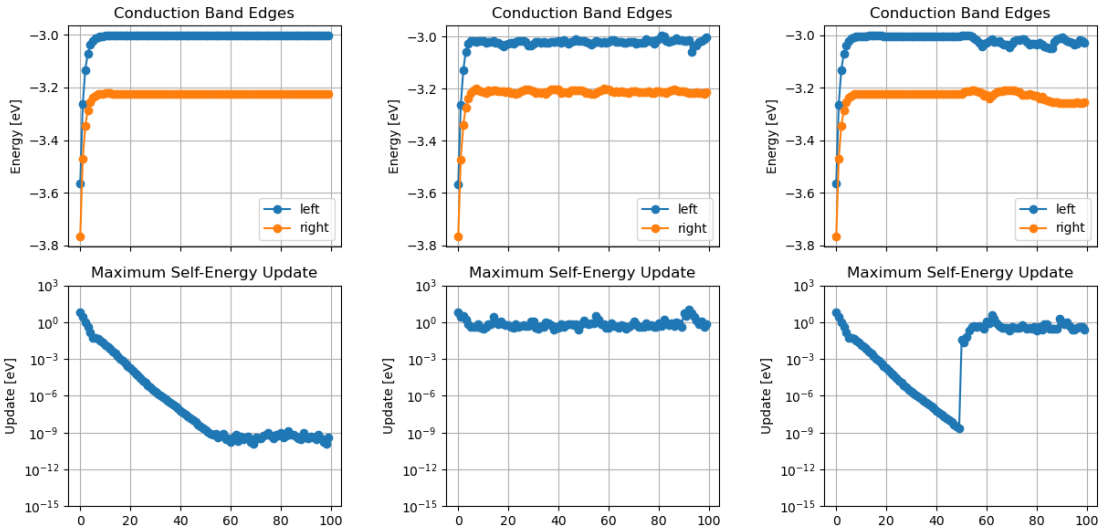
The results of an adaptive energy grid using the interpolation are shown, compared next to the reference uniform grid. Fig 6.1 shows stability of conduction band edges and the maximum self-energy update across iterations. They are used as convergence metrics (see subsection 2.5.1). The band edges of the uniform grid convergence to a stable value within 15 iterations. The maximum self-energy update continues to fall until reaching a low plateau of approximately 10^{-9} after 50 iterations.

Fig 6.2 shows the current convergence. The left and right currents of the uniform grid converges to a stable value within 15 iterations. The difference between the left and right currents is less than 10^{-9} after 50 iterations. Thus full convergence is reached after 50 iterations for the uniform grid.

The grid was switched to an adaptive grid after the first iteration. The adaptive grid does not converge. The conduction band edges continuously wobble around the reference solution. The maximum self-energy stays at approximately 1, which is a very high value. The left and right currents oscillate slightly above the reference value. The current difference between the left and right currents also never converges to a stable value.

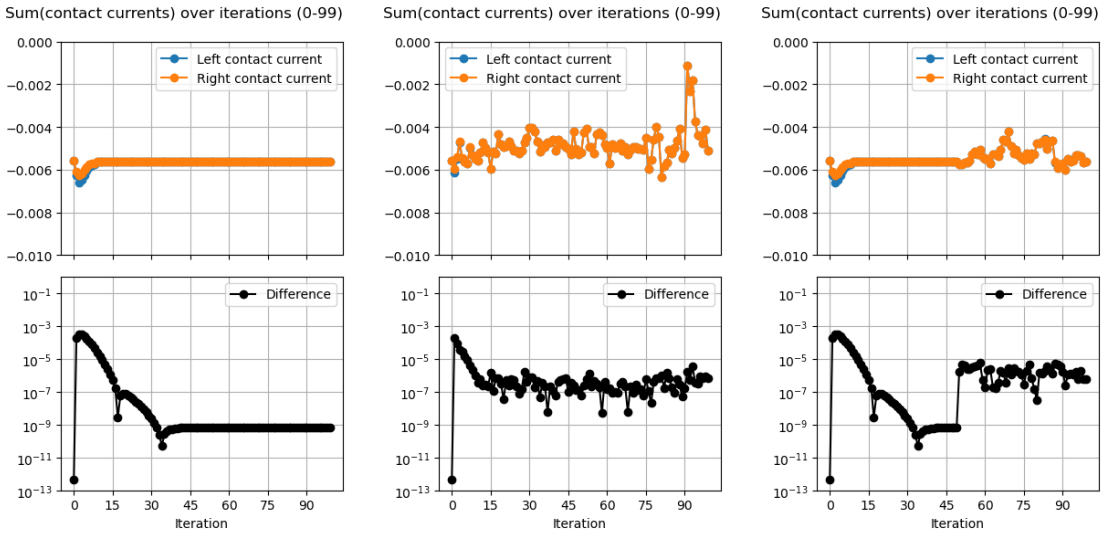
Perhaps switching to an adaptive grid after the first iteration is too early. The adaptive grid is created based on the features of G and P after the first iteration. The features of G after the first iteration are roughly stable, but perhaps not similar enough to that of G after convergence. Thus the switch from uniform grid to adaptive grid was tried after convergence was reached, which was 50 iterations on the uniform grid.

The results are shown in the column (c) of Fig 6.1 and Fig 6.2. The adaptive grid still does not converge. On iteration 51, the conduction band edges start wobbling like the previous case. The maximum self-energy update, although already low, jumps up to approximately 1 and stays there. Likewise the currents and current differences show the same oscillatory behavior as the previous case.



(a) uniform grid (b) adaptive start iteration 1 (c) adaptive start iteration 50

Figure 6.1.: Convergence metrics for SCBA. The adaptive versions do not converge. The conduction band edges wobble around the reference solution. The maximum self-energy update stays at approximately 1.



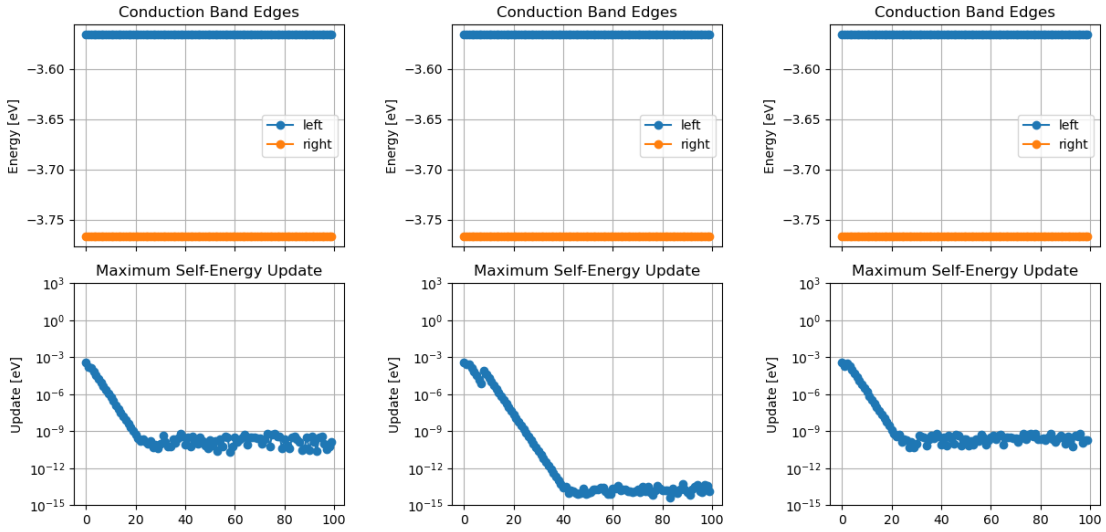
(a) uniform grid (b) adaptive start iteration 1 (c) adaptive start iteration 50

Figure 6.2.: Current convergence. The adaptive versions do not converge. The current oscillates around the reference solution.

As a sanity check and demonstration of the method, the GW interaction was removed from the SCBA loop. Only the phonon interaction was used. The phonon interaction

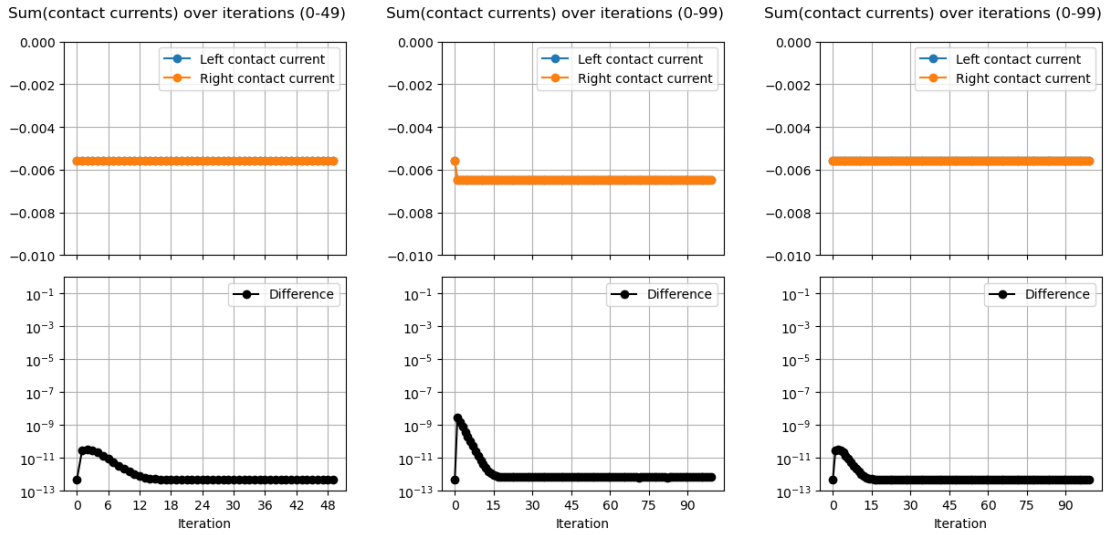
is actually a pseudo-phonon interaction since true phonons are not implemented in Quatrex as of this thesis writing. The results are shown in Fig 6.3 and Fig 6.4. Since only phonons are used, the conduction band edges converge after one iteration. This is the case for all three versions.

The maximum self-energy update converges with approximately the same rate of convergence (approximately 10^{-3} every 10 iterations) between the different versions. The adaptive grid versions converge further than the uniform grid version, reaching a lower plateau of approximately 10^{-14} after 40 iterations. This is a good demonstration that suggests the adaptive grid is able to capture the features of the self-energy better than the uniform grid.



(a) uniform grid (b) adaptive start iteration 1 (c) adaptive start iteration 50

Figure 6.3.: Convergence metrics for only phonons (no GW). The adaptive version converges further as the maximum self-energy update reaches a lower plateau.



(a) uniform grid (10 meV) converged: -5.60×10^{-3}

(b) adaptive start iteration 1 converged: -6.47×10^{-3}

(c) uniform grid (1 meV) converged: -5.68×10^{-3}

Figure 6.4.: Current convergence. All versions immediately converge, but to slightly different values. There is a known bug with the phonon self-energy in the adaptive version, which may be the cause of the different convergence values.

6.1. Current limitations

There is a known bug in the phonon self-energy calculation. When the energy is shifted by $\pm\hbar\omega$, the resulting energy may not be on the energy grid (regardless of whether it is uniform or adaptive). However the code implements the shift as simply an plus or minus index shift. A patch fix [1] was applied, but it is not a complete fix. The correct version should use interpolation to compute the self-energy at the shifted energy.

The adaptive grid is shared across all orbital indices. This is a limitation because the features of the SCBA variables are different across orbital indices. Thus the grid cannot be perfectly adapted for any single orbital index, but rather is a compromise across all orbital indices.

[1] https://github.com/quatrex/quatrex/pull/285

Chapter 7

Conclusion and Future Work

7.1. Conclusion

The use of an adaptive energy grid in quantum transport simulations has been investigated in this thesis. The adaptive grid was created using a gradient monitor, instead of the more common quadrature method. The difficulty of using an adaptive grid lies in performing Fourier transforms on a non-uniform grid. Various methods were investigated which included a direct non-uniform Fast Fourier Transform (NUFFT), Voronoi weighted non-uniform discrete Fourier transform (NUDF), and interpolation. The non-uniform transform methods were found to be not accurate enough for Quatrex. Thus the interpolation method was implemented and tested. The end-to-end results show that the band edges, maximum self-energy update, and currents do not converge when using the adaptive grid with the GW interaction. The results wobble around the reference solution, which is computed using a uniform grid. The individual SCBA variables (G, P, W, Σ) also exhibit a wobbling around reference values. By removing the GW interaction, the adaptive grid was able to converge to a lower self-energy update than the uniform grid.

7.2. Future Work

The source of the error that causes the SCBA variables to wobble around the reference solution is not known. Perhaps it is due to crossing over between the G/Σ grid and the P/W grid, as they are separate grids. A way to isolate the error contribution from each step of the SCBA loop should be investigated, so the source of the error can be identified. The known bug in the phonon self-energy calculation should be fixed. The adaptive energy grid can be applied to the quantum transmitting boundary method (QTBM). It is expected to perform better because there are no convolutions in QTBM. One can simply select the energy points that are needed to compute the transmission function, and then perform the calculations on those energy points.

Appendix A

Device Transport Models

This section presents a brief summary of semiconductor device transport models and their limitations. They are presented from least to most complex. It's important to keep in mind that "all models are wrong, but some are useful" (commonly attributed to George Box).

A.1. Drift-Diffusion (DD)

This is the simplest model of carrier transport in semiconductors. It is a classical model that describes charge carriers as simple particles that are under the influence of 2 forces.

1. Drift: Force from an external electric field $\vec{E}$.
2. Diffusion: Force for a particle to move down from a concentration gradient ∇n (electrons) or ∇p (holes).

The conduction current is the sum of the electron and hole currents:

$$\vec{J_n} = q\mu_n n\vec{E} + qD_n\nabla n \tag{A.1}$$
$$\vec{J_p} = q\mu_p p\vec{E} - qD_p\nabla p \tag{A.2}$$
$$\vec{J} = \vec{J_n} + \vec{J_p} \tag{A.3}$$

where:

- q is the elementary charge
- μ_n and μ_p are the electron and hole mobilities respectively
- D_n and D_p are the electron and hole diffusion coefficients, respectively
- n and p are the electron and hole concentrations, respectively

The DD model is simple and fast. It is often the first model taught in undergraduate semiconductor physics courses. For large planar bulk transistors that can be modeled as a 1D channel, the DD model is sufficient. It can capture bulk effects. However, for any modern transistor, the DD model is too simple to capture any nuanced effects. It is actually a special case of the more general Boltzmann Transport Equation (BTE) based methods.

A.2. Boltzmann Transport Equation (BTE) based methods

The Boltzmann Transport Equation (BTE) is a single-particle transport model that describes the time evolution of the Boltzmann distribution function $f(\vec{r},\vec{k},t)$. It describes the probability of finding a particle at position $\vec{r}$ with momentum $\vec{k}$ at time t.

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla_{\vec{r}} f + \vec{F} \cdot \nabla_{\vec{k}} f = \left(\frac{\partial f}{\partial t}\right)_{\text{collision}} \tag{A.4}$$

It is a semi-classical model that accounts for various scattering mechanisms such as phonon scattering, impurity scattering, and surface roughness scattering. The most common method to solve the BTE is the Monte Carlo method. It simulates a large ensemble of particles and their trajectories, with a random chance of a particle experiencing a scattering event at each time step. The particles are assumed to be non-interacting, follow ballistic trajectories, and the scattering events are treated as instantaneous.

The Method of Moments technique can be applied to the BTE to simplify the model. It tracks the aggregate properties of the particle ensemble. The first moment is the charge density, the second moment is the current density, and the third moment is the energy density, etc. The DD model is a special case of the BTE where only the first two moments are tracked and the energy density is neglected. By including higher-order terms, the BTE produces models such as the hydrodynamic model and the energy transport (or sometimes called energy balance) model. These are commonly found in commercial device simulators such as Sentaurus TCAD and Silvaco Atlas.

The limitation of the BTE is that it is a single-particle description. It does not account for interactions between particles. At low electron densities, this is a good approximation. However, at high electron densities, the interactions between particles become significant and the BTE breaks down.

Another limitation of the BTE is that it is a semi-classical model. It treats the electron wave packet as a classical particle, but the collisions are treated quantum mechanically. This is valid when the electron's De Broglie wavelength is much smaller than the distance traveled between scattering events. As transistor sizes shrink, the De Broglie wavelength becomes comparable to the device dimensions and quantum mechanical effects such as tunneling and quantum confinement become significant. The BTE cannot capture these effects and thus becomes inaccurate for modern transistors.

List of Figures

List of Tables

Bibliography

[1] H. G. Feichtinger, K. Gröchenig, and T. Strohmer, "Efficient numerical methods in non-uniformsampling theory," vol. 69, no. 4, pp. 423–440. [Online]. Available: https://doi.org/10.1007/s002110050101

[2] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" vol. 52, no. 12, pp. 5406–5425. [Online]. Available: https://ieeexplore.ieee.org/document/4016283

[3] J. Jackson, C. Meyer, D. Nishimura, and A. Macovski, "Selection of a convolution function for fourier inversion using gridding (computerised tomography application)," vol. 10, no. 3, pp. 473–478. [Online]. Available: https://ieeexplore.ieee.org/document/97598

[4] A. H. Barnett, J. F. Magland, and L. a. Klinteberg, "A parallel non-uniform fast fourier transform library based on an "exponential of semicircle" kernel." [Online]. Available: http://arxiv.org/abs/1808.06736

[5] W. Huang, Y. Ren, and R. D. Russell, "Moving mesh partial differential equations (mmpdes) based on the equidistribution principle," *SIAM Journal on Numerical Analysis*, vol. 31, no. 3, pp. 709–730, 1994. [Online]. Available: https://doi.org/10.1137/0731038

[6] W. Huang, "Variational mesh adaptation: Isotropy and equidistribution," vol. 174, no. 2, pp. 903–924. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999101969451

[7] H. Yan, "Adaptive energy grid algorithm for a next-generation quantum transport simulator."

[8] H. Haug and A.-P. Jauho, *Quantum Kinetics in Transport and Optics of Semiconductors*, 2nd ed., ser. Springer Series in Solid-State Sciences. Springer, 2008, vol. 123.

[9] L. Deuschle, J. Cao, A. N. Ziogas, A. Winka, A. Maeder, N. Vetsch, and M. Luisier, "Electron-electron interactions in device simulation via nonequilibrium green's functions and the GW approximation," vol. 111, no. 19, p. 195421. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.111.195421

[10] N. Vetsch, A. Maeder, V. Maillou, A. Winka, J. Cao, G. Kwasniewski, L. Deuschle, T. Hoefler, A. N. Ziogas, and M. Luisier, "Ab-initio quantum transport with the GW approximation, 42,240 atoms, and sustained exascale performance." [Online]. Available: http://arxiv.org/abs/2508.19138

[11] K. S. Thygesen and A. Rubio, "Conserving GW scheme for nonequilibrium quantum transport in molecular contacts," vol. 77, no. 11, p. 115333. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.77.115333

[12] R. Duflou, G. Gaddemane, M. Houssa, and A. Afzalian, "Fully coupled electron-phonon transport in two-dimensional-material-based devices using efficient FFT-based self-energy calculations," vol. 307, p. 109430. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465524003539

[13] J.-M. Lin, "Python non-uniform fast fourier transform (PyNUFFT): An accelerated non-cartesian MRI package on a heterogeneous platform (CPU/GPU)," vol. 4, no. 3, p. 51. [Online]. Available: https://www.mdpi.com/2313-433X/4/3/51

[14] J. Fessler and B. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," vol. 51, no. 2, pp. 560–574. [Online]. Available: https://ieeexplore.ieee.org/document/1166689

[15] A. Barnett, "Building a better non-uniform fast fourier transform."

[16] J. Keiner, S. Kunis, and D. Potts, "Using NFFT 3—a software library for various nonequispaced fast fourier transforms," vol. 36, no. 4, pp. 19:1–19:30. [Online]. Available: https://dl.acm.org/doi/10.1145/1555386.1555388

[17] D. Ruiz-Antolin and A. Townsend, "A nonuniform fast fourier transform based on low rank approximation." [Online]. Available: http://arxiv.org/abs/1701.04492