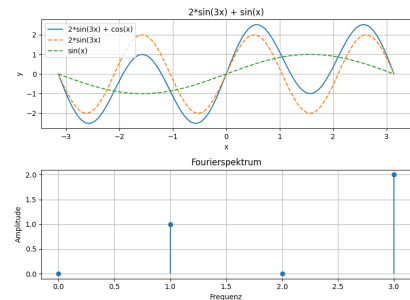
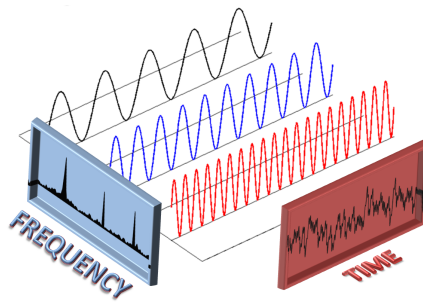


Mini Challenge

Fourier Transformation



Das Ziel dieser Aufgabe besteht darin, dass Sie ein grundlegendes Verständnis der Fouriertransformation zur Näherung von periodischen Funktionen und dessen praktische Anwendung zur Datenanalyse und -bearbeitung erlangen. Hierfür sollen Sie ein Jupyter Notebook erstellen, in dem Sie die unten aufgeführten Aufgaben nacheinander bearbeiten. Zunächst werden Sie die Fouriertransformation in numpy selbst implementieren und damit drei Funktionen fitten. Danach werden Sie nacheinander Sonnendaten, Musik und Bilder mithilfe von Fouriertransformationen analysieren und bearbeiten.

Ziel dieser Aufgabe ist nicht nur, Ihre mathematischen Kenntnisse unter Beweis zu stellen, sondern auch die entsprechende Kommunikation und Präsentation Ihrer Ergebnisse. Ihre Abgaben sollen also nicht nur **mathematisch korrekt**, sondern auch **leicht verständlich** und **reproduzierbar** sein. Genauere Angaben zu den Erwartungen an die Abgabe finden Sie in den Auswertungskriterien. Dokumentieren Sie ihren Arbeitsfortschritt und Erkenntnisgewinn in Form eines Lerntagebuchs, um Lernfortschritte, Schwierigkeiten und Erkenntnisse festzuhalten.

Eigenverantwortung

Diese Aufgabe zählt als unbegleitetes Selbststudium. Es wird also von Ihnen erwartet, sich die nötigen Informationen und Verständnis eigenständig anzueignen, sowie diesen Prozess zu planen. Ich empfehle Ihnen, sich jede Woche zumindest 1-2 Stunden mit dieser Aufgabe zu beschäftigen und die Bearbeitung nicht aufzuschieben. Als Dozent stehe ich Ihnen auf Anfrage (im Unterricht und per Mail) gerne als Ressource zur Verfügung, zB mit individuellem Feedback, Erläuterungen und Diskussion. Kommen Sie jederzeit gerne einfach auf mich zu.

Empfohlenes Vorgehen

Arbeiten Sie zuerst diese Liste nacheinander ab, wo sinnvoll am besten in Gruppenarbeit.

- ☐ Aufgabenstellung sorgfältig durchlesen, Verständnisfragen und unklare Begriffe aufschreiben
- ☐ Mit der Theorie beschäftigen (Hilfreiche Ressourcen) und Fragen beantworten
- ☐ Alle Rechnungen händisch nachvollziehen, um implementation vorzubereiten
- ☐ Händische Rechnungen an Dozenten schicken. (Meilenstein: SW 4)
- ☐ Bewertungskriterien durchlesen, insbesondere Malus beachten

Erst wenn Sie diese Liste abgearbeitet haben sollten Sie mit dem Coden beginnen.

Die Aufgaben bauen aufeinander auf. **Es ist nicht sinnvoll, die Aufgaben unter den Teilnehmern aufzuteilen.** Bearbeiten Sie die Aufgaben stattdessen besser gemeinsam und nacheinander.

Hilfreiche Ressourcen:

Theorie Fourier Transformation

Fourier Transformation - 3Blue1Brown

Youtube Playlist: Fourier Analyse - Daniel Jung

Introduction to Fourier Image Processing

Tutorials: Fourier Transformation in Python

Fourier Transform with scipy.fft

Image manipulation with scipy.fft

Künstliche Intelligenz:

Für diese Aufgaben dürfen KI Tools wie ChatGPT oder Github Copilot als Ressource genutzt werden, um Fragen zu stellen oder bei Problemen Unterstützung zu erhalten. Voraussetzung ist, dass Sie transparent kommunizieren, wo und wie Sie diese Tools eingesetzt haben und welche Verbesserungen nötig waren. Legen Sie Ihrer Abgabe eine PDF mit einem repräsentativen relevanten Chat bei, der Ihren Umgang mit KI zeigt, sowie einer kurze Reflexion über Ihren Umgang mit KI: Wo hat es geholfen? Wo war es hinderlich?

Eine Hilfestellung zur sinnvollen Nutzung von KI als Lernassistent, insbesondere für Mathematik, habe ich in diesen Videos vorbereitet:

Lernen mit Chatbots

Mathe Lernen mit Chatbots

Die folgenden Aufgabenstellungen präzisieren die einzelnen Bearbeitungsschritte und geben die Struktur des Notebooks vor.

Aufgabe 1.

In dieser Aufgabe implementieren Sie die Berechnung der Fourier-Approximation in numpy (kein fft)

- Implementieren Sie mit numpy eine Methode zur Berechnung der k -ten Fourier-Komponente einer Funktion gegeben durch arrays x und y mithilfe der Trapezregel (`np.trapz`).
- Implementieren Sie mit numpy eine Methode zur Berechnung der Fourier-Approximation f_{app} mit einer gegebenen Anzahl (m) von Frequenzkomponenten.
- Nähern Sie die Funktion $f_1(x) = 1/(e^x + e^{-x})$ auf dem Intervall $[-3, 3]$ (Auflösung: $N = 600$ Schritte) mit verschiedenen Anzahlen von Komponenten ($2m + 1$). Plotten Sie den Vergleich zur Originalfunktion f .
- Bestimmen Sie die Anzahl der Fourier-Komponenten, die benötigt werden, um eine Approximation mit höchstens 0,1% relativen Fehler zu erhalten.
- Versuchen Sie die Funktionen $f_2(x) = e^x$ und $f_3(x) = e^{|x|}$ auf dem Intervall $[-3, 3]$ (Auflösung: $N = 600$ Schritte) mit einem Approximationsfehler unter 0.1 % anzunähern. Beschreiben und erklären Sie die Unterschiede.

Aufgabe 2.

In dieser Aufgabe wenden Sie die Fourier-Analyse auf Sonnenfleckendaten an (`sunspots.txt`). Sie dürfen dafür die `fft` routine von numpy oder scipy verwenden.

- Plotten Sie das Power-Spektrum und finden Sie die dominante Periodizität der Schwingung. Damit erhalten Sie eine Abschätzung auf die Dauer des Sonnenzyklus.
- Können Sie weitere Perioden in den Sonnenfleck-Daten mithilfe des Fourier-Spektrums finden?
- Entfernen Sie einige Komponenten im Fourier-Spektrum und führen dann eine Rücktransformation durch:
 - grössten Komponenten entfernen ($k > 20$)
 - kleinsten Komponenten entfernen ($k < 5$)

Beschreiben und erklären Sie die Ergebnisse.

- Vergleichen Sie die Ergebnisse mit denen aus Aufgabe 1. Beschreiben und erklären Sie Unterschiede in den Fourier-Koeffizienten.

Aufgabe 3.

In dieser Aufgabe analysieren und bearbeiten Sie ein Musikstück (`music.wav`). Sie dürfen dafür zB die `fft` routine von numpy oder scipy sowie das `librosa` und `soundfile` paket verwenden.

- Plotten Sie die Wellenform des Klangs (`waveplot`) - einmal vollständig und einmal für einen kleinen Zeitbereich, sodass die Wellenform deutlich wird - sowie das Amplitudenspektrum. Wieso ist das Amplitudenspektrum symmetrisch bezüglich des Vorzeichens?

- (b) Trennen Sie den Sound-Signal in einen Teil oberhalb und unterhalb von 2 kHz in Frequenzen. Identifizieren Sie jeweils die Top 5 Frequenzen in diesen Frequenzbereichen.
- (c) Berechnen Sie aus diesen beiden Teilen jeweils eine Rekonstruktion. Berechnen Sie den Fehler der Rekonstruktion wie in Aufgabe 1 für die gefilterten Signale und schreiben Sie die gefilterten rekonstruierten Daten in eine wav-Datei. Beschreiben und erklären Sie das Ergebnis.

Aufgabe 4.

In dieser Aufgabe analysieren und bearbeiten Sie eine Bilddatei (img.jpg). Sie dürfen dafür die `fft` routine von `numpy` oder `scipy` verwenden.

- (a) Plotten Sie die 2D-Verteilung der absoluten Werte der Fourier-Komponenten, sodass die kleinsten k -Werte in der Mitte liegen. Erklären Sie die Natur der Verteilung.
- (b) Entfernen Sie je die Fourier-Komponenten mit absoluter Amplitude unter dem Schwellenwert 0.001%, 0.01%, 0.1%, 1% und 10% der maximalen Amplitude. Führen Sie eine inverse Fourier-Transformation durch und stellen Sie die gefilterten Fourierkomponenten sowie die resultierenden Bilder dar. Beschreiben und erklären Sie die Unterschiede.
- (c) Berechnen Sie den Fehler (Aufgabe 1) zwischen rekonstruiertem und Originalbild. Wie weit können die Daten komprimiert werden, ohne einen Fehler von höchstens 1% zu überschreiten?