



**HOCHSCHULE
HANNOVER**

*Fakultät I
Elektro- und
Informationstechnik*

**Programmieren in Python:
Dokumentation eines Rechnerprogramms (EDR)**

Prof. Dr. Forgber

Verfasser:

Fedi Boukhris (1535241)

Anis Doudech (1484190)

Rihab Bedhiafi (1353539)

02.01.2022

Inhaltsverzeichnis

| | |
|---|----|
| 1. Aufgabenbeschreibung | 1 |
| 2. Beschreibung des Programmaufbaus | 1 |
| 2.1. GUI_main | 1 |
| 2.2. GUI_support | 1 |
| 2.3. Check_konfiguration | 1 |
| 2.4. Plot_messdaten | 1 |
| 2.5. Error_message | 2 |
| 2.6. Bode_diagramm_Exceptions | 2 |
| 3. Bedienungsanleitung für das Programm | 2 |
| 4. Beschreibung aller Konfigurationsdaten und des Aufbaus der Konfigurationsdatei | 4 |
| 5. Beschreibung eines Testlaufs inkl. Testdaten und Testergebnisse | 8 |
| 5.1. Testlauf eines unsinnigen Parameters in der Konfigurationsdatei | 8 |
| 5.2. Plot von Messdatenreihe | 10 |
| 5.3. Plot von Funktion $f(x)$ | 11 |
| 5.4. Plot von mehreren Funktionen | 11 |
| 5.5. Plot von Parametrischer Funktion | 12 |
| 5.6. Plot von 2-Dimensionalen Funktion $f(x,y)$ in 3D-Graph | 12 |
| 5.7. Plot von Übertragungsfunktion $G(s)$ | 13 |
| 6. Eigenständigkeitserklärung | 13 |
| 7. Abbildungsverzeichnis | 14 |
| 8. Tabellenverzeichnis | 14 |

1. Aufgabenbeschreibung

Es wird ein Programm erstellt, das mit Hilfe des Pakets Matplotlib grafische Darstellungen von Funktionen und Messdaten erzeugen kann. Wobei werden die Funktionen durch eine Konfigurationsdatei eingestellt, die mit den notwendigen Parametern versorgt werden soll.

In diesem Programm werden zudem folgende Punkte betrachtet:

- Das Programm kann mehrere Funktionen in einem Bild darstellen.
- Das Programm kann die Funktionen als Gleichung, aber auch als Messdatentabelle aus einer Textdatei verarbeiten können
- Das Programm kann parametrische Funktionen darstellen
- Das Programm kann Bodediagramm anhand einer Übertragungsfunktion $G(s)$ darstellen
- Das Programm kann auch Funktionen $f(x,y)$ in 3D-Graph darstellen
- Das Programm wird über ein GUI ausgeführt, über das die Konfigurationsdatei geändert und geladen, die Funktion dargestellt und Figuren gespeichert werden können
- Die Software wird objektorientiert erstellt.

2. Beschreibung des Programmaufbaus

Das Programm wurde Objektorientiert erstellt und besteht aus 6 Klassen.

2.1. GUI_main

GUI_main stellt die Main-Klasse dar und erstellt die Programm-Oberfläche basierend auf das Tkinter-Modul. In dieser Klasse wird gezeigt, welche Elemente hat das Programm, wo diese platziert sein sollen und welche Funktion sie besitzen. Da diese etwa schwer für Python Anfänger auffallen, wird ein GUI-Designer für das Platzieren der Elemente als Hilfsmittel verwendet. Diese GUI-builder heißt „PAGE“. Es ist eine frei verfügbare Software. Es ermöglicht dem Designer das Anordnen von Elementen in dem GUI mittels Drag and Drop .

2.2. GUI_support

In diesem Modul werden die Funktionalitäten aller Elementen des GUI erstellt.

2.3. Check_konfiguration

In dieser Klasse werden alle benötigten Konfigurationsdaten aus der Konfigurationsdatei ausgesucht bzw. angepasst und in einem Dictionary fürs Plotten vorbereitet.

2.4. Plot_messdaten

in diesem Modul werden Plot-Klassen entsprechend dem Funktionstyp deklariert und die demensprechend Plot-Methoden erstellt.

2.5. Error_message

Klasse zeigt eine Fehlermeldung Fenster bei falsch eingegebenem Parameter.

2.6. Bode_diagramm_Exceptions

Klasse prüft die Eingabe der Übertragungsfunktion und zeigt ein Tkinter Fehlermeldung bei falscher Eingabe.

3. Bedienungsanleitung für das Programm

Vor dem Starten des Programms soll zunächst die nicht von Python vorinstallierten Bibliotheken installiert werden: PIL, numpy, matplotlib, os, re.

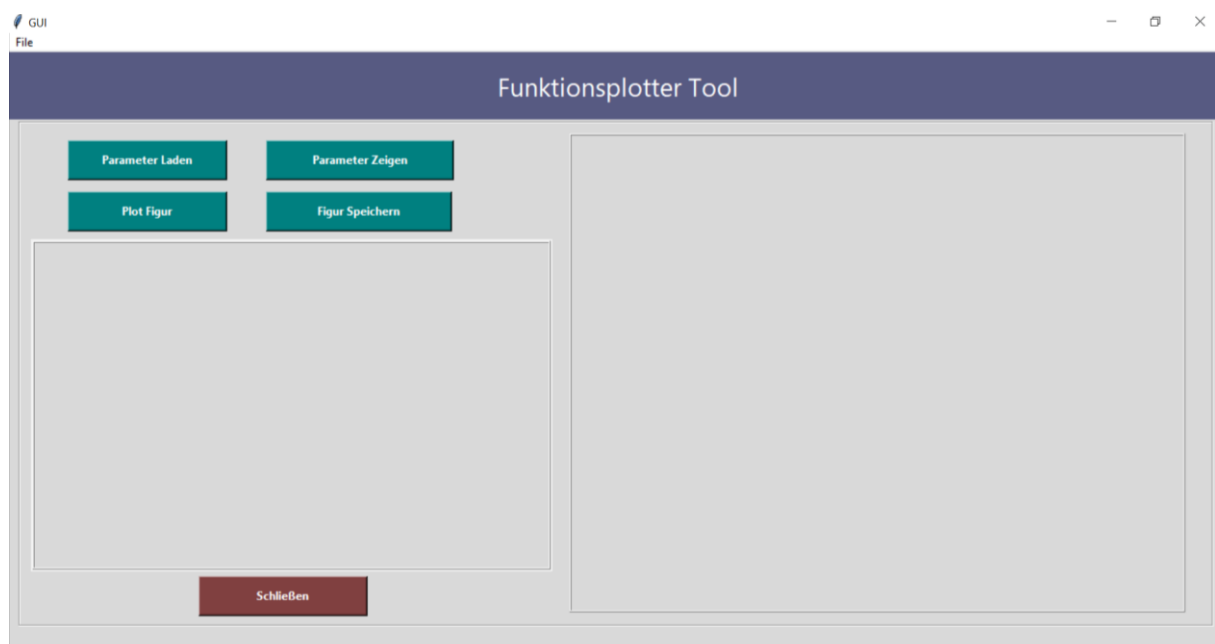


Abbildung 1: Programm GUI

Nachdem das Programm über die Main-Klasse gestartet wird, wird die Programm-Oberfläche angezeigt (vgl. Abbildung 1). Nach dem Einstellen von der Konfigurationsdatei (File -> Konfigurationsdatei Einstellen), soll die Datei über das Button „Parameter Laden“ geladen werden, und dann kann die Funktion über das Button „Plot Figur“ geplottet werden. Sollte ein unsinniger Parameter in der Konfigurationsdatei antreten, wird eine Fehlermeldung angezeigt. Die Richtigkeit der Eingabe von Konfigurationsdaten werden im Absatz 4 gut beschrieben. Die Konfigurationsdatei kann jederzeit wieder geändert und geladen werden, sobald das GUI noch nicht geschlossen wird. Man kann verschiedene weitere Funktionalitäten nebenbei ausführen. Diese lassen sich gut im folgenden Ablaufdiagramm erkennen (vgl. Abbildung 2).

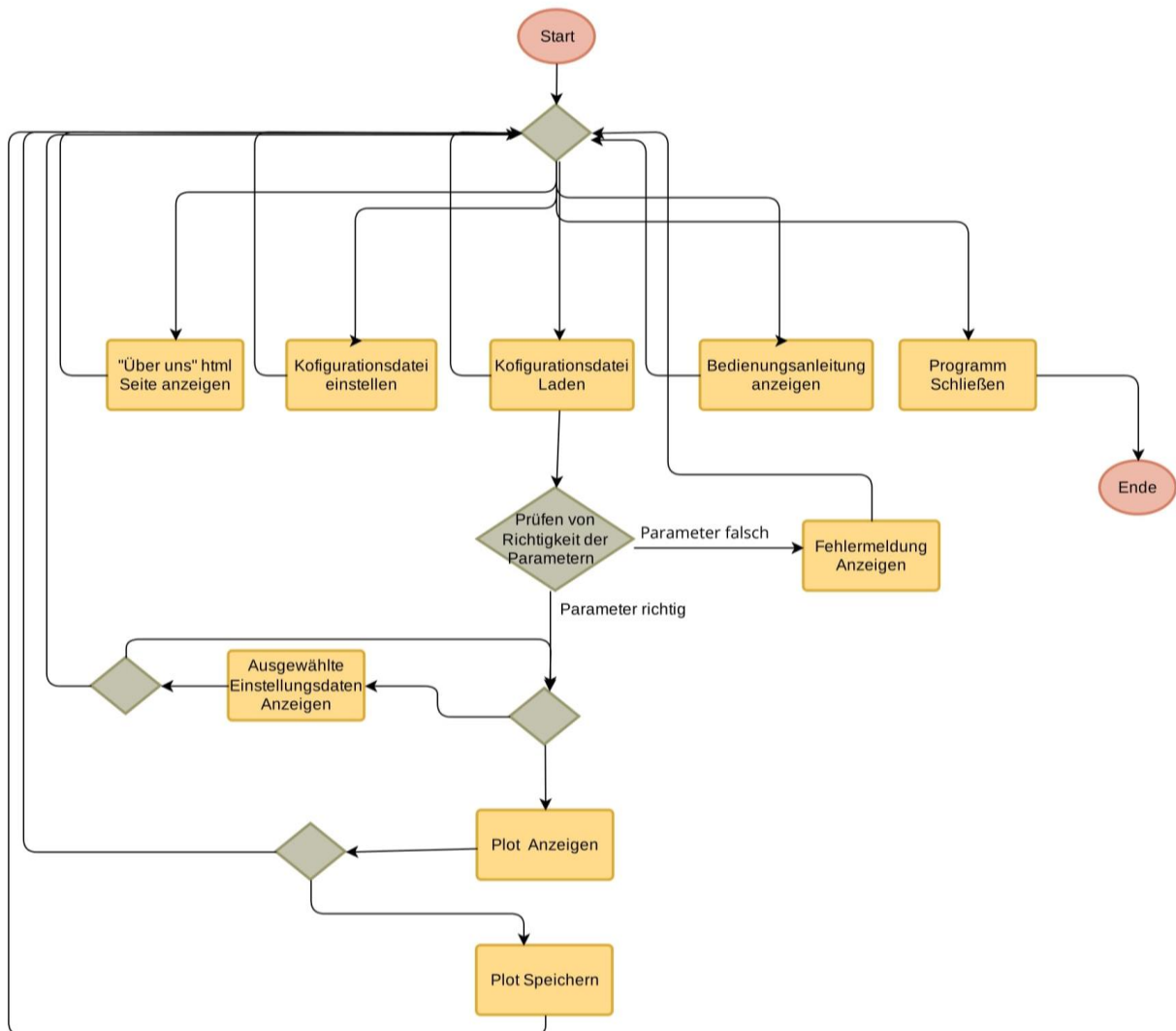


Abbildung 2: Ablaufdiagramm vom Programm

4. Beschreibung aller Konfigurationsdaten und des Aufbaus der Konfigurationsdatei

Um die Bedienung des Programms zu gewährleisten, ist es notwendig, ein geeignetes Textdokument einzuführen, da das Hauptprogramm die für den Plot notwendigen Konfigurationen aus einer Textdatei entnimmt.

Die Textdatei besteht aus insgesamt vier verschiedenen Funktionen mit eigenen Parametern, nämlich Funktionsgleichung, parametrische Funktion, mehrdimensionale Funktion und Übertragungsfunktion, zusätzlich kann der Nutzer auch eine Reihe von Messungen eingeben.

Je nach Bedarf des Nutzers ist es möglich, die Art der Funktion zu wählen, dazu ist einfach den gewünschten Funktionstyp neben dem Schlüsselwort "Check" ganz oben in der Textdatei anzugeben.

Es gibt für jede Art von Plot ein Schlüsselwort, dass man angeben muss, diese lauten wie folgt:

Tabelle 1: Erlaubte Funktionsarten

| Schlüsselwort | Funktionsart |
|----------------------------------|---|
| Messdaten | Eingabe von Messdaten |
| Funktionsgleichung | Eingabe von maximal 3 Funktionen $f(x)$ |
| Parametrische Funktion | Eingabe von einer $f(x,a,b)$ Funktion |
| Mehredimensionalefunktion | Eingabe von einer mehrdimensionalen Funktion |
| Uebertragungsfunktion | Eingabe von einer Übertragungsfunktion $g(s)$ |

Zum Beispiel kann man auf Wunsch eine Übertragungsfunktion plotten, während die Parameter der anderen Funktionstypen in der Textdatei verbleiben, dazu muss der Benutzer nur 'Uebertragungsfunktion' neben dem Schlüsselwort 'Check' ganz oben im Inhalt der Textdatei angeben (siehe folgende Abbildung).

```
#####
# Konfigurationsdaten Funktionsplotter

Check: Uebertragungsfunktion
-----
Messdaten
-----
x_Reihe : 1,3,4,5,,6,6,6,6,5,4,43
y_Reihe : 1,3,5,6,9,0,4,2,3,2,1
grid (Ja/Nein): ja
color: Red
linewidth: 2
linestyle: None
title:
x_Label: this is x
y_Label: this is y
-----
```

Abbildung 3: Wählen von „Übertragungsfunktion“

Jeder Funktionstyp hat eigene Parameter, die die Eigenschaften des jeweiligen Funktionstyps berücksichtigen. So ist es zum Beispiel bei einer Übertragungsfunktion sinnvoller, keine Schritte einzuführen, da der Plot eines Bodediagramms alle möglichen Frequenzen zwischen der Start- und der Endfrequenz benötigt.

Die Parameter der einzelnen Funktionstypen sind in den folgenden Tabellen dargestellt:

Tabelle 2: Parameter der einzelnen Funktionstypen

| Funktionstyp: Messdaten | | | |
|----------------------------------|---------------------|---|--|
| Schlüssel | Beispiel von Inhalt | Beschreibung | Bemerkung |
| x_Reihe | 1,3,4,5,6 | Werte an der X-Axe | Die Elemente müssen unbedingt mit einem Komma getrennt sein. |
| y_Reihe | 2,4,6,10,12 | y Reihe | Ganzzahlserie |
| color | blue | Farbe | Matplotlib.colors |
| Grid | Nein | Gitter | Ja wenn gewünscht |
| linewidth | 2 | Linienbreite | Ganzzahl |
| linestyle | -. | Linienstil | '--', '-.', '-', ':' |
| title | Any | Title vom Plot | Alles ist möglich |
| x_Label | Any | x Axe Label | Alles ist möglich |
| y_Label | Any | y Axe Label | Alles ist möglich |
| Funktionstyp: Funktionsgleichung | | | |
| Schlüssel | Beispiel von Inhalt | Beschreibung | Bemerkung |
| f_1(x) | x**2 | eine x abhängige Funktion | kann je nach Wahl zwischen 1 und 3 Funktionen eingegeben |
| f_2(x) | | | |
| f_3(x) | | | |
| x_start | 1 | x Anfang | nur Ganzzahlen |
| x_end | 24 | y Ende | |
| x_step | 2 | Schritt | |
| title | Any | Title vom Plot | Alles ist möglich |
| x_Label | Any | x Axe Label | Alles ist möglich |
| y_Label | Any | y Axe label | Alles ist möglich |
| grid | Ja oder Nein | Gitter | Ja wenn gewünscht |
| color_1 | blue | Farbe für jede einzelne Funktion | Matplotlib.colors |
| color_2 | | | |
| color_3 | | | |
| linewidth_1 | 2 | Linienbreite für jede einzelne Funktion | Ganzzahl |
| linewidth_2 | | | |
| linewidth_3 | | | |
| linestyle_1 | -- | Linienstil für jede einzelne Funktion | '--', '-.', '-', ':' |
| linestyle_2 | | | |
| linestyle_3 | | | |
| title | Any | Title vom Plot | Alles ist möglich |

| | | | |
|---|----------------------------|------------------------------------|--|
| x_Label | Any | x Axe Label | Alles ist möglich |
| y_Label | Any | y Axe label | Alles ist möglich |
| Legend | Nein | Legende | Ja wenn gewünscht ist |
| Funktionstyp: Parametrische Funktion | | | |
| Schlüssel | Beispiel von Inhalt | Beschreibung | Bemerkung |
| f_parametriert (x,a,b) | (x+2+a)+b | eine x, a und b abhängige Funktion | a und b müssen angegeben werden |
| a | 2.2 | a Parameter | Ganzzahl |
| b | 2.4 | b Parameter | Ganzzahl |
| color | blue | Farbe | Matplotlib.colors |
| x_start | 1 | x Anfangswert | nur Ganzzahlen |
| x_end | 24 | y Endwert | |
| x_step | 2 | Schritt | |
| Grid | Nein | Gitter | Ja wenn gewünscht |
| linewidth | 2 | Linienbreite | Ganzzahl |
| linestyle | - | Linienstil | '--', '-.', ':', oder ':' |
| title | Any | Titel vom Plot | Alles ist möglich |
| x_Label | Any | x Axe Label | Alles ist möglich |
| y_Label | Any | y Axe Label | Alles ist möglich |
| Funktionstyp: Mehrdimensionalefunktion | | | |
| Schlüssel | Beispiel von Inhalt | Beschreibung | Bemerkung |
| f_3D(x,y) | x**2+y**2 | eine x und y abhängige Funktion | Funktion muss gültig sein |
| color | blue | Farbe | Matplotlib.colors |
| start | 1 | x und y Anfangswert | nur Ganzzahlen |
| end | 24 | x und y Endwert | |
| step | 2 | Schritt | |
| Grid | Nein | Gitter | Ja wenn gewünscht |
| linewidth | 2 | Linienbreite | Ganzzahl |
| linestyle | - | Linienstil | Soll in diesem Bereich sein ['--', '-.', ':', ':'] |
| title | Any | Title vom Plot | Alles ist möglich |
| x_Label | Any | x Axe Label | Alles ist möglich |
| y_Label | Any | y Axe Label | Alles ist möglich |
| z_Label | Any | z Axe Label | Alles ist möglich |

| Funktionstyp: Uebertragungsfunktion | | | |
|-------------------------------------|---------------------|---------------------------|--|
| Schlüssel | Beispiel von Inhalt | Beschreibung | Bemerkung |
| G(s) | $(s+1)/(s+1)$ | eine s abhängige Funktion | muss eine gültige G(s) eingegeben werden |
| color | blue | Farbe | Matplotlib.colors |
| f_start | 0.00001 | Anfangsfrequenz | nur Ganzzahlen müssen sinnvolle Frequenzen eingegeben werden |
| f_end | 100000 | End frequenz | |
| Grid | Nein | Gitter | Ja wenn gewünscht |
| linewidth | 2 | Linienbreite | Ganzzahl |
| linestyle | - | Linienstil | '--', '-.', ':', ':' |
| title | Any | Title vom Plot | Alles ist möglich |

Ausnahmeregelung:

Es ist wichtig, dass der Benutzer sicherstellt, dass er die richtigen Werte für jeden Parameter angibt. Es ist jedoch möglich, dass der Benutzer vergisst, einen Parameter auszufüllen oder eine falsche Eingabe macht z. B. einen Buchstaben anstelle einer ganzen Zahl für den Parameter linewidth, step, oder x_Reihe angibt (siehe folgende Abbildung). In diesem Fall wird das Programm den Benutzer mit Hilfe einer Tkinter-Meldebox über seinen Fehler informieren, dies gilt für absolut alle Parameter, die eine ganze Zahl erhalten sollen.

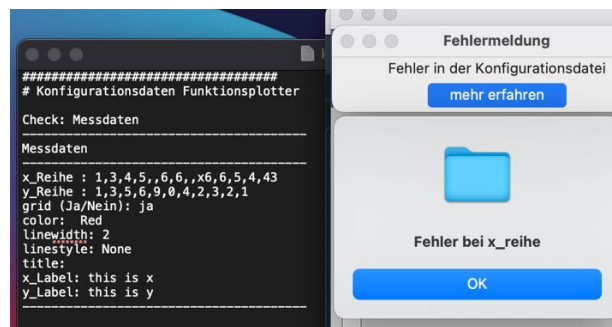


Abbildung 4: Fehler beim Parameter „x_Reihe“

Bei den Parametern für Linienstil, Linienfarbe oder Grid ist eine falsche Eingabe in keinem Fall fatal für das Funktionieren des Programms, da in diesem Fall ein Standardwert zugewiesen wird.

Ein weiterer häufiger Fehler ist es, den Parameter G(s) für Übertragungsfunktion falsch einzugeben, da der Benutzer dies mit einer einfachen Funktion verwechseln kann. In diesem Fall erscheint ein Tkinter-

Fenster, in dem der Benutzer die korrekte Übertragungsfunktion eingeben kann (vgl. Abbildung 5). Nach der Korrektur soll das Button „Plot Figur“ nochmal geklickt.

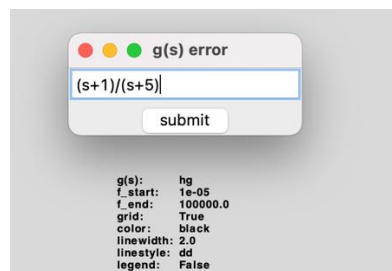


Abbildung 5: $G(s)$ korrigieren

5. Beschreibung eines Testlaufs inkl. Testdaten und Testergebnisse

5.1. Testlauf eines unsinnigen Parameters in der Konfigurationsdatei

Zunächst wird die Konfigurationsdatei über das GUI geöffnet (vgl. Abbildung 6 und Abbildung 7), und dann werden die Konfigurationsdaten eingegeben (vgl. Abbildung 8). Nach dem Ändern und Speichern der Konfigurationsdatei, soll sie über die GUI geladen (Button „Parameter Laden“ drücken). Nachdem das Button „Plot Funktion“ gedrückt wird, wird eine Fehlermeldung in einer MessageBox angezeigt (vgl. Abbildung 9).

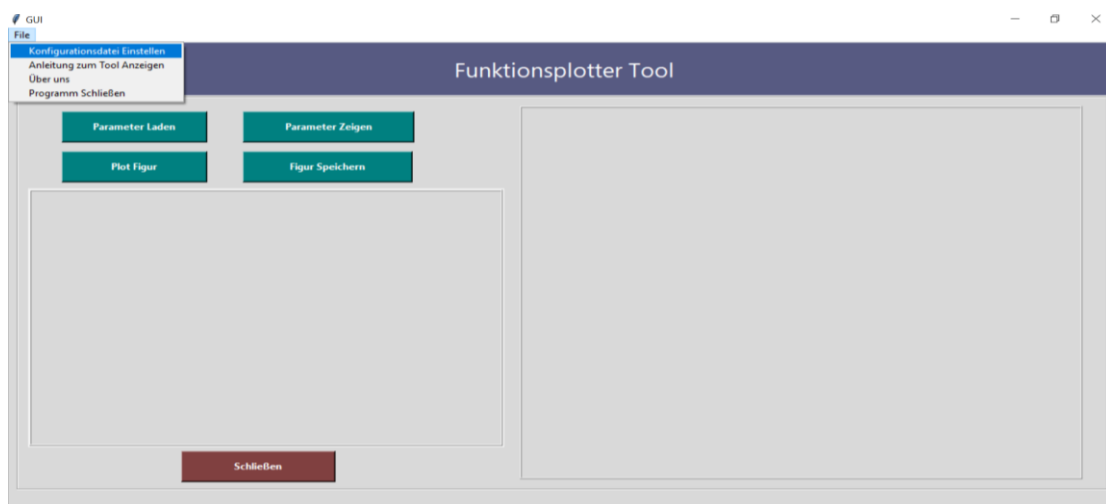


Abbildung 6: Konfig-Datei ändern

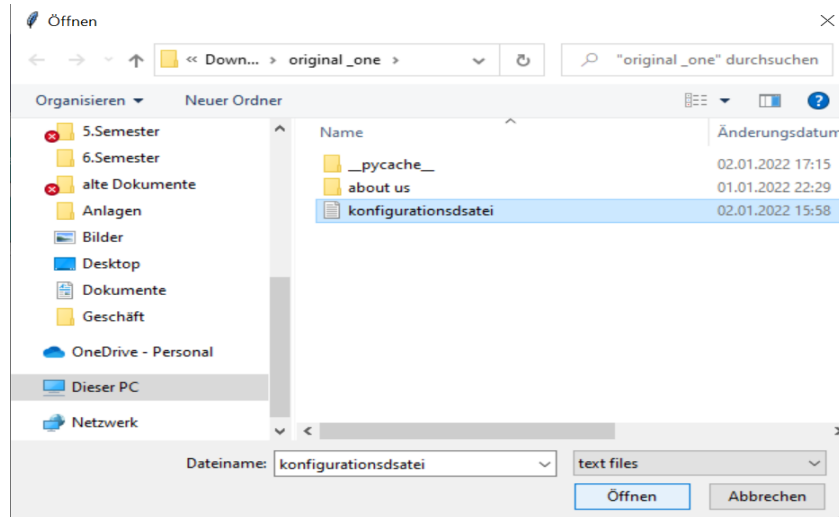


Abbildung 7: Konfig-Datei öffnen

```
-----
Funktionsgleichung
-----
f_1(x): x**2
f_2(x): x+1
f_3(x): x*2

x_start: -10
x_end: 10
x_step: 0.1
grid (Ja/Nein): Ja
color_1: red
color_2: green
color_3: yellow
linewidth_1: xxxxx
linewidth_2: 5
linewidth_3: 3
linestyle_1: -
linestyle_2: -.
linestyle_3: --
title: Beispiel
Legend (Ja/Nein): Ja
x_Label: X_axe
y_Label: Y_axe
-----
```

Abbildung 8: Konfigurationsdaten ändern

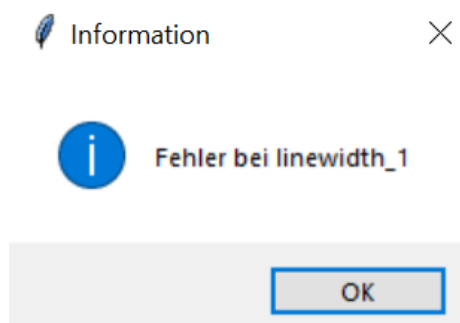


Abbildung 9: Anzeige von Fehlermeldung

5.2 Plot von Messdatenreihe

Wichtig ist hier, neben dem Schlüssel „Check“ in der Konfigurationsdatei den richtigen Funktionstyp einzugeben („Messdatenreihe“).

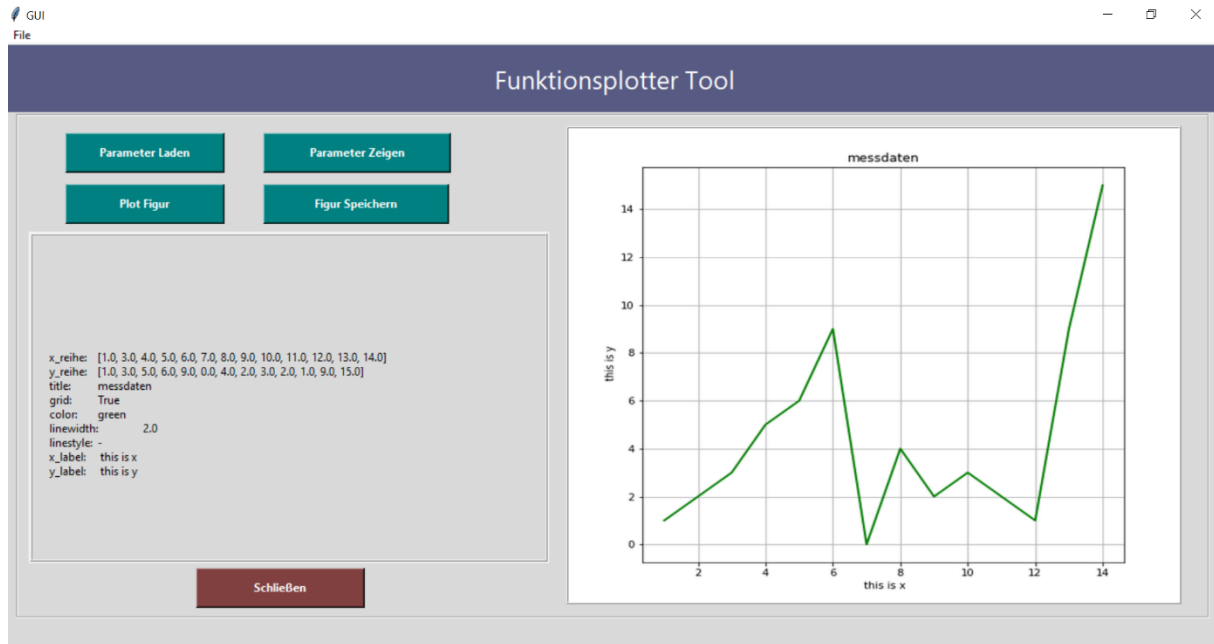


Abbildung 10: Messdatenreihe

5.3 Plot von Funktion $f(x)$

Wichtig ist hier, neben dem Schlüssel „Check“ in der Konfigurationsdatei den richtigen Funktionstyp einzugeben („Funktionsgleichung“).

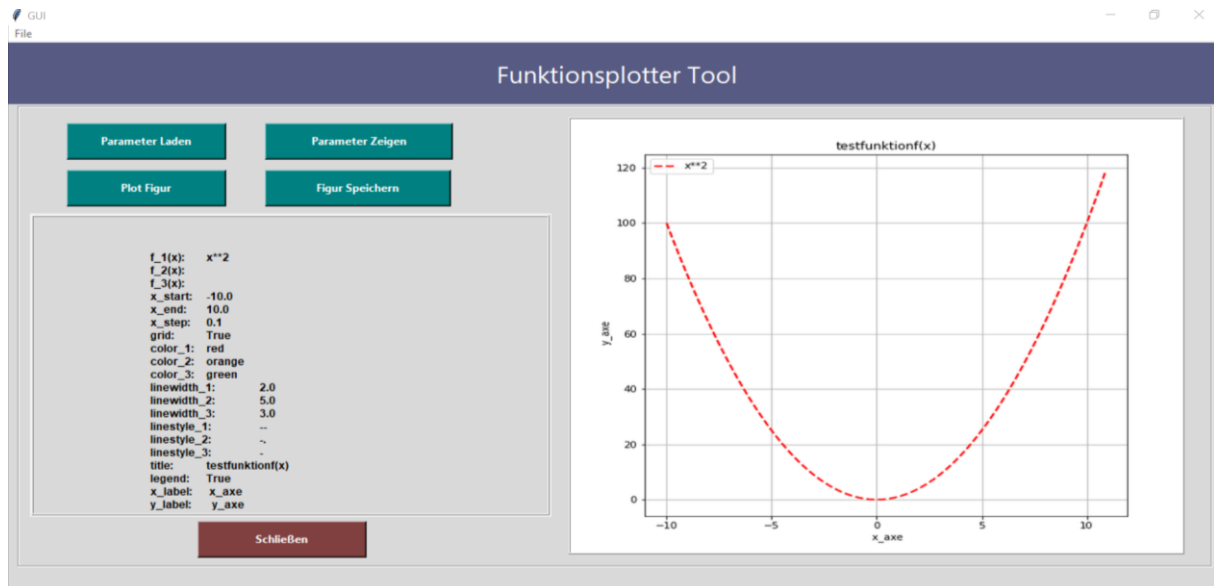


Abbildung 11: Funktion $f(x)$

5.4 Plot von mehreren Funktionen

Wichtig ist hier, neben dem Schlüssel „Check“ in der Konfigurationsdatei den richtigen Funktionstyp einzugeben („Funktionsgleichung“). Dann können die Schlüssel $f_1(x)$, $f_2(x)$ und $f_3(x)$ mit den zu plottenden Funktionen ausgefüllt werden.

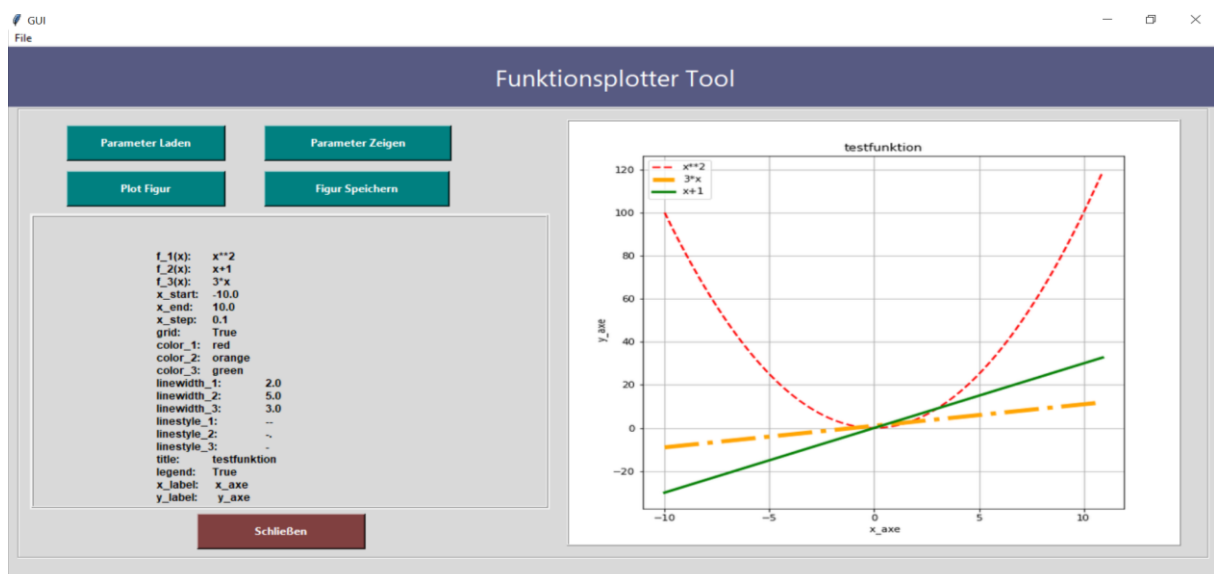


Abbildung 12: mehrere Funktionen plotten

5.5 Plot von Parametrischer Funktion

Wichtig ist hier, neben dem Schlüssel „Check“ in der Konfigurationsdatei den richtigen Funktionstyp einzugeben („Parametrische Funktion“).

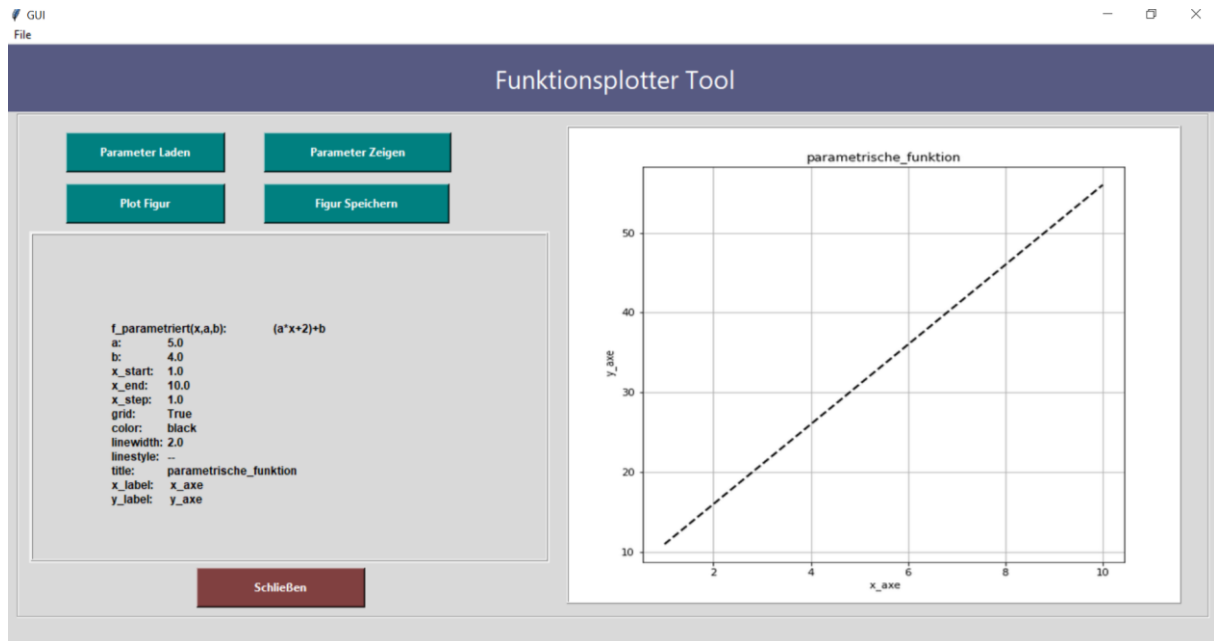


Abbildung 13: Parametrische Funktion plotten

5.6. Plot von 2-Dimensionalen Funktion $f(x,y)$ in 3D-Graph

Wichtig ist hier, neben dem Schlüssel „Check“ in der Konfigurationsdatei den richtigen Funktionstyp einzugeben („Mehrdimensionalefunktion“).

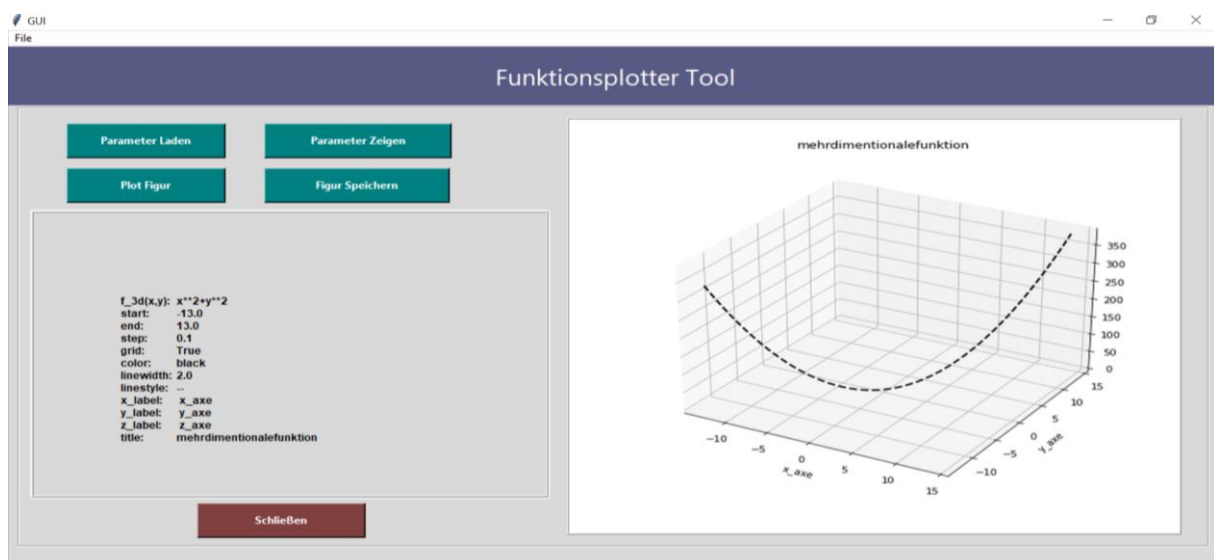


Abbildung 14: 3D-Plot

5.7 Plot von Übertragungsfunktion $G(s)$

Wichtig ist hier, neben dem Schlüssel „Check“ in der Konfigurationsdatei den richtigen Funktionstyp einzugeben („Übertragungsfunktion“).

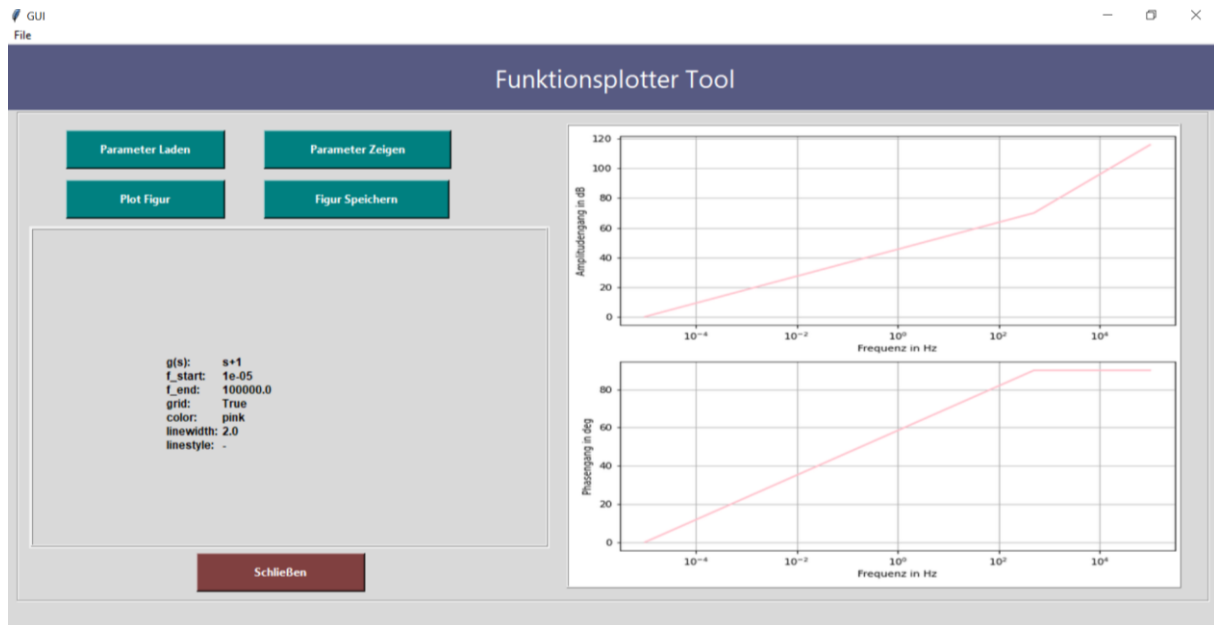


Abbildung 15: Bodediagramm von Übertragungsfunktion plotten

6. Eigenständigkeitserklärung

Hiermit erklären wird, dass die Dokumentation, wenn Quellen es nicht anders belegen, von uns selbst verfasst wurde.

Anis DH
Hannover, den 02.01.2022

B. Fiedler

Hannover, den 02.01.2022

7. Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1: Programm GUI | 2 |
| Abbildung 2: Ablaufdiagramm vom Programm..... | 3 |
| Abbildung 3: Wählen von „Übertragungsfunktion“ | 4 |
| Abbildung 4: Fehler beim Parameter „x_Reihe“ | 7 |
| Abbildung 5: G(s) korrigieren | 8 |
| Abbildung 6: Konfig-Datei ändern | 8 |
| Abbildung 7: Konfig-Datei öffnen..... | 9 |
| Abbildung 8: Konfigurationsdaten ändern | 9 |
| Abbildung 9: Anzeige von Fehlermeldung..... | 9 |
| Abbildung 10: Messdatenreihe | 10 |
| Abbildung 11: Funktion f(x) | 11 |
| Abbildung 12: mehrere Funktionen plotten | 11 |
| Abbildung 13: Parametrische Funktion plotten | 12 |
| Abbildung 14: 3D-Plot | 12 |
| Abbildung 15: Bodediagramm von Übertragungsfunktion plotten | 13 |

8. Tabellenverzeichnis

| | |
|---|---|
| Tabelle 1: Erlaubte Funktionsarten | 4 |
| Tabelle 2: Parameter der einzelnen Funktionstypen | 5 |