

10 Zehnte Übungseinheit

Inhalt der zehnten Übungseinheit:

- Fourier-Analyse
- Fast Fourier Transform (FFT)
- Aliasing
- Musteraufgaben zu Diagrammen und nichtlinearen Gleichungssystemen

Die 10. und 11. Vorlesung konnten ihnen (hoffentlich) die grundlegende Idee erklären: Signale lassen sich als Summe ihrer Frequenz-Komponenten darstellen. Für ein echtes Verständnis, und damit Sie auch praktisch damit umgehen können, müssen Sie sich intensiver mit technischen Details befassen.

Im moodle-Kursbereich zur 11. Vorlesung gibt es MATLAB-Demoprogramme; die Übungen sollen Sie stark motivieren, sich mit diesen zu beschäftigen. Dazu gibt es hier weitere Beispiele zu Fourier-Analyse, FFT und Aliasing.

Neben dem Haupt-Thema „Fourieranalyse“ illustriert dieser Abschnitt auch folgende MATLAB-Befehle von allgemeinem Interesse:

- Einlesen von Audio-Dateien: `audioread('klang1.wav')`
- Abspielen eines Audio-Signals: `sound(y,fs)`
- Einlesen von Daten im Text-Format: `load('Triest2014.dat')`
- Finden und Extrahieren von Komponenten in einem Datenvektor gemäß logischer Ausdrücke: `index=find(f<5000); Y_teil = Y(index);`

Die letzten vier Aufgaben sind als Vorbereitung für den Kenntnissnachweis gedacht.

10.1 Fourieranalyse, Fast Fourier Transform (FFT)

Aufgabe 91: Frequenz-Analyse eines Audio-Signals

Fourier-Analyse zerlegt ein Signal in seine Frequenzanteile. Diese Anleitung zeigt, wie Sie alle Frequenzkomponenten berechnen, aus denen ein Audio-Signal besteht.

Laden Sie dazu die Datei `klang1.wav` von der Übungshomepage und lesen Sie die Daten im MATLAB ein. Es gibt dafür einen speziellen Befehl,

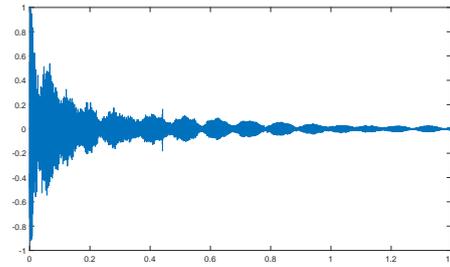
```
[X,fs] = audioread('klang1.wav');
```

Der Vektor `X` enthält die Signal-Messwerte, aufgezeichnet mit der Abtast-Frequenz (*sampling frequency*) `fs` (in Hertz, $1\text{ Hz} = \text{s}^{-1}$). Das heißt, es liegen pro Sekunde `fs` Datenpunkte vor; die gesamte Signaldauer ist also

$$\frac{\text{Anzahl der Datenpunkte} - 1}{\text{Abtastfrequenz}}$$

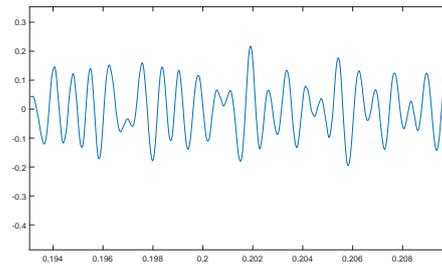
Erzeugen Sie die zu den Datenpunkten gehörige Zeitachse und plotten Sie das Signal.

```
n = length(X);
t = linspace(0, (n-1)/fs, n);
plot(t, X);
```



Sie sehen eine abklingende Schwingung. Wenn Sie neugierig sind und wissen wollen, wie sich das anhört: der Befehl `sound(X, fs)` spielt das Signal ab. Hineinzoomen (Hier abgebildet ist das Zeitintervall $\approx 0.194 \leq t \leq 0.208$) zeigt sinusförmigen Verlauf, überlagert mit anderen Effekten.

Sie können die Frequenz f dieser deutlich sichtbaren Frequenzkomponente direkt aus der Grafik abschätzen: Zählen Sie die Wellenberge im Intervall $[0.194; 0.208]$, dividieren Sie durch die Zeitdauer. Damit haben Sie schon eine wesentliche Information aus den Daten herausgelesen: Ein Hauptanteil sind Sinus-Schwingungen mit ≈ 1400 Hz, denen sich weitere Schwingungen überlagern..



Das ist aber nur eine ganz grobe Näherung; das Signal ist wesentlich komplexer.

Der Grundgedanke der Fourieranalyse ist:

Ein Signal, gegeben durch n Datenpunkte, lässt sich als Summe von n Schwingungstermen (Sinus, Cosinus, komplexe E-Funktionen) darstellen

Nun, *eine* Schwingung haben wir gefunden. Es fehlen noch gut 60 000 weitere (so viele Datenpunkte enthält das Signal). Ein direkter Ansatz, diese n Komponenten zu berechnen, würde $O(n^2)$ Rechenoperationen erfordern – viel zu aufwändig bei großen Datenmengen! Im Prinzip hat schon Karl Friedrich Gauß 1805 ein schnelleres Rechenverfahren dazu entworfen. Seit den 1960-er Jahren sind unter dem Begriff “FFT” (*fast Fourier transform*, schnelle Fourier-Transformation) Algorithmen für Computer verfügbar und inzwischen unabdingbar in der Signalverarbeitung. Die FFT reduziert den Rechenaufwand von $O(n^2)$ auf $O(n \log n)$. Nur dadurch ist FFT in Echtzeit-Anwendungen einsetzbar.

In MATLAB lautet der Befehl einfach

```
Y = fft(X);
```

Den n Signal-Messwerten im Datenvektor X entsprechen ebensoviele Fourier-Koeffizienten im der diskreten Fourier-Transformierten Y .

Die Formel, die sich hinter dem `fft`-Befehl verbirgt und die MATLAB sehr trickreich und rechengünstig auswertet, lautet

$$Y_k = \sum_{j=1}^n X_j \exp \left[\frac{-2\pi i}{n} (j-1)(k-1) \right]$$

Ein kleiner Nachteil: dieser FFT-Befehl in Standard-Ausführung liefert nicht direkt Sinus- und Cosinus-Koeffizienten für eine klassische Fourierreihe (Formel mit den a_n - und b_n -Termen in der vorigen Übungseinheit). Diese Koeffizienten lassen sich aber leicht bestimmen.

Die Darstellung des Signalvektors \mathbf{X} als Summe komplexer Exponentialfunktionen lautet

$$X_j = \frac{1}{n} \sum_{k=1}^n Y_k \exp \left[\frac{2\pi i}{n} (j-1)(k-1) \right]$$

Das sieht komplizierter aus, als es ist. Wichtig ist der Aufbau dieser Formel:

Datenvektor X = Skalierungsfaktor mal Summe von ...
 ... (Amplitude Y_k mal Schwingung mit Frequenz ω_k)

FFT:
 Datenvektor $X \mapsto$ Frequenzvektor Y , in MATLAB: $Y = \text{fft}(X)$

Umkehrung, inverse FFT:
 Frequenzvektor $Y \mapsto$ Datenvektor X , in MATLAB: $X = \text{ifft}(Y)$

Noch ein Detail: Die FFT-Formel arbeitet nur mit den Vektoren \mathbf{X} und \mathbf{Y} . Sie indiziert die X -Komponenten mit $j = 1, 2, \dots, n$. Sie weiss nichts von der zugehörigen Zeitachse mit Zeitpunkten t_1, t_2, \dots, t_n und den zu Y_k zugehörigen Kreisfrequenzen ω_k . Die Umrechnung oder Zuordnung von Index j zu Zeitpunkt t_j und Fourier-Komponente k zu Kreisfrequenz ω_k lautet

$$t_j = \frac{j-1}{f_s}, \quad \omega_k = \frac{2\pi(k-1)}{n} f_s$$

Es gilt bei reellen Datenvektoren mit geradzahlgiger Länge n : Die Real- und Imaginärteile der Fourier-Terme $Y(2)$ bis $Y(n/2)$, mit $2/n$ multipliziert, geben jeweils die Cosinus- und Sinus-Amplituden zu den Frequenzen

$$\frac{f_s}{n}, \quad 2 \frac{f_s}{n}, \quad 3 \frac{f_s}{n}, \quad \dots, \quad \frac{n-1}{2} \frac{f_s}{n}$$

an. (Mit 2π multipliziert sind das die Kreisfrequenzen ω_k .) Ein Sonderfall ist der Term $Y(1)$: er ist die Summe aller Signalwerte; dividiert durch n gibt er den Mittelwert des Signals an. Noch ein Sonderfall ist (bei geradem n) der Term $Y(n/2+1)$: er ist nur mit $1/n$ zu multiplizieren und gibt die Cosinus-Amplitude zur Frequenz $f_s/2$ an. (Bei ungeradem n gibt es keine Information mehr zu $f_s/2$.)

Wichtig ist noch bei reellen Datenvektoren: nur bis zur Mitte des Y -Vektors ist tatsächlich Information enthalten; die zweite Hälfte enthält gespiegelt dieselben Werte, nur komplex-konjugiert.

FFT liefert Frequenz-Information bis zur Hälfte der Abtast-Frequenz f_s .
 Man nennt $f_s/2$ die Nyquist-Frequenz.

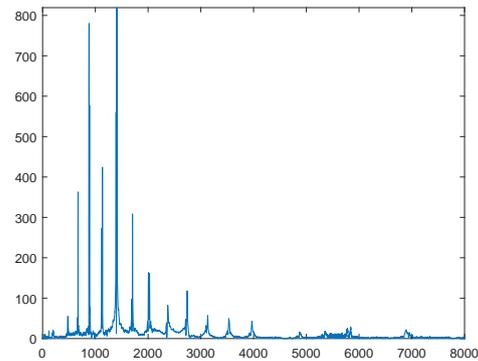
Um Frequenzanteile eines Signals mindestens bis zu einer Frequenz f_{\max} korrekt zu erfassen, muss die Abtastfrequenz f_s größer sein als $2f_{\max}$. Das menschliche Hörvermögen reicht (in

jungen Jahren bestenfalls) bis etwa 22 000 Hz. Deswegen ist auf Audio-CDs (und in unserer Beispieldatei `klang1.wav`) die Abtast-Frequenz $f_s = 44\,100$ Hz.

So stellen Sie das Frequenzspektrum dar:

```
f = linspace(0, fs/2, n/2+1);  
plot(f, abs(Y(1:n/2+1)))
```

Die Grafik zeigt einige deutliche, scharf begrenzte Frequenzspitzen im Bereich bis etwa 4000 Hz. Andererseits gibt es oberhalb von etwa 8000 Hz keine nennenswerten Beiträge. Es bringt nichts, bis zur Nyquist-Frequenz, hier 22 050 Hz, zu zeichnen. Zoomen Sie in den interessantesten Frequenzbereich (interaktiv oder mit dem Befehl: `axis([0 8000 0 max(abs(Y))])`).



Aufgabe 92: Klang einer Gitarren-Saite

Die Datei `klang2.wav` enthält eine kurze Audio-Aufnahme einer gezupften Saite. Das Klangspektrum besteht aus dem Grundton und einer Reihe von Obertönen. Analysieren Sie das Frequenzspektrum, stellen Sie das Frequenzspektrum im Bereich bis 10 000 Hz dar. Üblich ist eine semilogarithmische Darstellung,

```
f = linspace(0, fs/2, n/2+1);  
index=find(f<5000);  
semilogy(f(index), abs(Y(index)))
```

Beantworten Sie folgende Fragen:

- Welche Frequenz liefert den stärksten Beitrag?
- Was ist die Frequenz der Grundschwingung? (Das ist die Tonhöhe, die ein Musiker hört.)
- Bis zu welcher Frequenz lassen sich Oberschwingungen deutlich erkennen?

Aufgabe 93: Ebbe und Flut

Dieses Beispiel soll zeigen: Fourier-Analyse kann in einem Datensatz periodische Effekte erkennen und aus dem Hintergrund-Rauschen herausfiltern.

Der Meeresspiegel schwankt aufgrund der Gezeiten (Ebbe und Flut). Dabei überlagern sich verschiedene astronomische und geographische Effekte mit unterschiedlichen Frequenzen (oder entsprechenden Perioden). Wind und Wetter sorgen für zusätzliche, unvorhersehbare Schwankungen. Die Theorie sagt (unter anderm) folgende Komponenten voraus:

Bezeichnung	Name	Periode (in Stunden)
M2	halbtägige Hauptmondtide	12.4206012
S2	halbtägige Hauptsonnentide	12
N2	große elliptische Tide 1. Ordnung zu M2	12.65834751
K1	eintägige Hauptdeklinatinstide	23.93447213
O1	eintägige Hauptmondtide	25.81933871
M4	Flachwasser-Oberschwingung zu M2	6.210300601
M6	Flachwasser-Oberschwingung zu M2	4.140200401

Der Datensatz `Triest2014.dat` auf der Übungs-Homepage enthält für das Jahr 2014 stündlich gemessene Werte des Meeresspiegels am Pegel im Hafen von Triest.

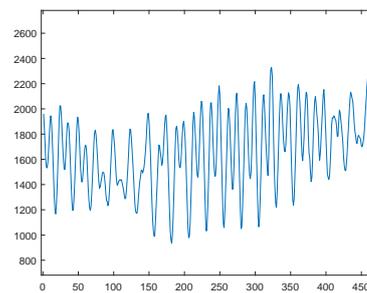
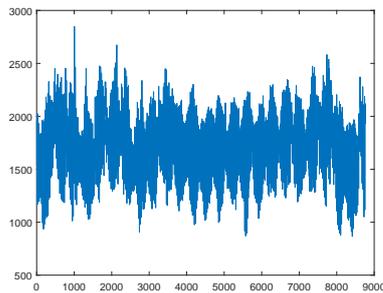
(Quelle: <http://uhs1c.soest.hawaii.edu/data/download/fd#uh829>).

Fourier-Analyse kann die verschiedenen periodischen Anteile herausfiltern.

Lesen Sie die Daten ein.

```
X=load('Triest2014.dat');
n=length(X);           % Anzahl Datenpunkte
fs = 1;                % Abtast-Frequenz 1/Stunde
```

Ein Plot des Datenvektors verschafft Ihnen einen ersten Überblick. Wenn Sie geeignet hineinzoomen, können Sie ungefähr 12- und 24-stündig periodische Schwankungen deutlich erkennen. (Zeiteinheit auf x-Achse ist 1 Stunde!) Sie sehen aber auch, dass sich anscheinend verschiedene kurz- und längerfristige Schwankungen überlagern. Die Fourier-Analyse entdeckt die einzelnen periodischen Schwingungen, deren Überlagerung den Datenvektor erzeugt.



Wichtige Größe ist die Abtastfrequenz f_s (*sampling rate, sampling frequency*, daher das s im Subskript). Bei Zeiteinheit Stunden und einem Messwert pro Stunde ist hier einfach

$$f_s = 1, \quad (\text{Einheit } 1/\text{h})$$

Führen Sie die Fourier-Transformation durch und berechnen Sie die entsprechenden Frequenzen:

```
Y = fft(X);
f = linspace(0, fs/2, n/2+1);
```

Der Term $Y(1)/N$ ist der Mittelwert aller Daten, also der mittlere Pegelstand. Die Terme $\text{abs}(Y(2))$ bis $\text{abs}(Y(N/2+1))$ entsprechen den Amplituden harmonischer Schwingungen mit Frequenzen (von f_s/N bis zur halben Abtastfrequenz $f_s/2$ in Schritten f_s/N).

Bei diesen Datensatz ist nicht die Frequenz, sondern die Periode aussagekräftiger. Es sind Schwingungen mit Perioden bis etwa 30 Stunden von Interesse. Längperiodische Vorgänge beschreiben Schwankungen über Tage oder Monate; die wollen wir hier nicht untersuchen.

Berechnen Sie zum Frequenz-Vektor f den entsprechenden Perioden-Vektor T . Filtern Sie aus dem T -Vektor den relevanten Bereich und zeichnen Sie die zugehörigen Amplituden.

```
T = 1./f;
index = find(T<30);
plot(T(index), abs(Y(index)))
```

Aus dem Diagramm können Sie zum Beispiel ablesen:

- Die Schwingungen mit welcher Periode liefert den stärksten, zweit- und drittstärksten Beitrag?
- In welchem Verhältnis stehen die Amplituden der Hauptmondtide M2 und der Hauptsonnentide S2?
- Lassen sich die Oberschwingungen M4 und M6 in den Daten nachweisen?

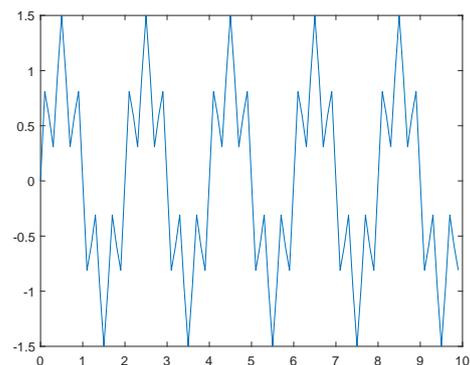
10.2 Aliasing

Die folgenden Befehle erzeugen einen Datenvektor, der das Signal

$$X(t) = \sin(\pi t) + \frac{1}{2} \sin(5\pi t)$$

mit $n = 100$ Datenpunkten und Abtastfrequenz $f_s = 10$ Hz speichert.

```
fs=10;
n = 100;
t = linspace(0, (n-1)/fs, n);
X = sin(pi*t) + 1/2*sin(5*pi*t);
```

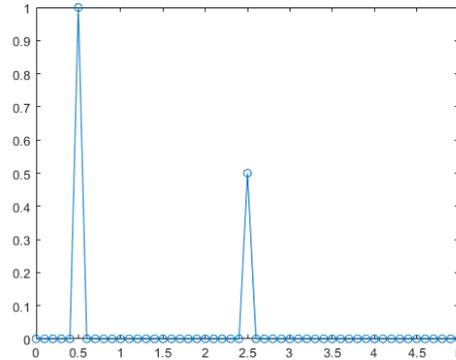


Fourier-Analyse zeigt korrekt die beiden Frequenzen 0.5 Hz und 2.5 Hz an. Auch die entsprechenden Amplituden stimmen, weil im Plot mit $2/n$ skaliert wurde).

```

%% Fourier-Transformation
Y = fft(X);
%% Frequenz-Achse, Frequenzspektrum zeig
f = linspace(0, fs/2, n/2+1);
index=find(f<10000);
plot(f(index), abs(Y(index))*2/n, '-o')

```



Aufgabe 94: Und wo ist das Problem?

Erzeugen Sie Datenvektoren, die höhere Signalfrequenzen enthalten, zum Beispiel

$$X(t) = \sin(2\pi t) + \frac{1}{2} \sin(12\pi t)$$

oder

$$X(t) = \cos(4\pi t) + \cos(9\pi t)$$

oder

$$X(t) = \cos(11\pi t) + \sin(15\pi t)$$

Welche Signalfrequenzen liefert die Fourieranalyse? Wo stimmen die Ergebnisse mit den Signalfrequenzen überein, wo nicht?

Aliasing ist der Begriff für Fehler, die auftreten, wenn das Signal $X(t)$ Frequenzanteile grösser oder gleich der Nyquist-Frequenz, das ist die halbe Abtastfrequenz, enthält.

$$\text{Nyquist Frequenz } f_{Ny} = \frac{f_s}{2}$$

Höhere Frequenzanteile interpretiert die Fourier-Analyse fälschlicher Weise als Frequenzen unterhalb der Nyquist-Frequenz. (Diese hohen Frequenzen treten also unter „Alias-Namen“ als niedrige Frequenzen auf.)

Schlagen Sie in Wikipedia unter „Alias-Effekt“ nach, dort finden Sie sehr anschauliche Darstellungen!

Leck-Effekt Die Datenpunkte im Spektrogramm einer diskreten Fouriertransformation haben Abstand f_s (Abtastfrequenz) auf der Frequenzachse. In unserem Beispiel: 0.1 Hz. Wenn die Signalfrequenz nicht exakt mit so einem diskreten Frequenzwert übereinstimmt, zeigt das Spektrogramm keine scharfen Spitzen. Die Signalfrequenz verteilt sich über mehrere benachbarte Datenpunkte. Sie „sickert“ oder „leckt“ zu benachbarten Datenpunkten.

Ein Beispiel ist ein Signal mit Frequenz $f = \frac{8}{3}$, zum Beispiel

$$X(t) = \cos\left(\frac{16\pi}{3}t\right);$$

Stellen Sie wie vorhin das Frequenzspektrum dar. Welcher Frequenz entspricht die Spitze im Diagramm? Welche Frequenzen treten als die beiden nächststärksten Beiträge auf?

Leck-Effekt: Fourieranalyse eines diskreten Datenvektors liefert nur in Spezialfällen – Abtastfrequenz ein ganzzahliges Vielfaches der Signalfrequenz – scharfe Frequenz-Datenpunkte. Ein sinusförmiges Signal verteilt sich im Allgemeinen auf mehrere benachbarte Frequenz-Datenpunkte.

Auch der Leck-Effekt kann als eine Form von Aliasing gesehen werden: Die korrekte Frequenz tritt im Spektrogramm zum Teil auch als benachbarte Frequenzen auf.

Aufgabe 95: Über den Wolken

Wenn Sie ein Flugticket mit Fensterplatz billig buchen, kann es sein, dass der Blick nach draußen so aussieht. In der Kabine ist es dann auch entsprechend laut. Die kurze Videosequenz `propeller.mp4` (Download von der Übungs-Homepage) vermittelt einen Eindruck davon.

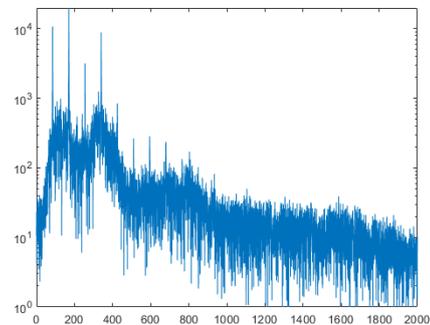


Auf dem Video dreht sich der Propeller scheinbar recht langsam. Wenn Sie genau hinsehen und auf den Anstellwinkel der Propellerblätter achten, dreht er sich sogar rückwärts! Keine Panik, das sind Aliasing-Effekte. Die tatsächliche Propellerdrehzahl können Sie aus dem Betriebsgeräusch durch Fourieranalyse ermitteln. Der Befehl

```
[y,fs] = audioread('propeller.mp4')
```

liest das Audiosignal ein.

Stellen Sie das Frequenzspektrum im Bereich 0–2000 Hz dar. In semilogarithmischer Darstellung erkennen Sie einige scharfe Peaks über einem breitbandigem Hintergrundrauschen.



Die Peaks entsprechen den Fourier-Komponenten des Signals der rotierenden Propellerblätter: Grundfrequenz und Vielfache davon. Das Triebwerk ist ein Pratt & Whitney Canada Corp PW 150 mit maximaler Propellerdrehzahl 1200 rpm. Der Propeller ist sechsstrahlig, die Grundfrequenz des Signals ist daher die sechsfache Drehzahl.

Welche Drehzahl (in rpm, Umdrehungen pro Minute) können Sie aus dem Signal bestimmen?

Das Video zeigt 29.9 Einzelbilder pro Sekunde. Um welchen Winkel dreht sich ein Propellerblatt von einem Bild zum nächsten? Um welchen Winkel dreht sich der sechsstrahlige Propeller *scheinbar* aufgrund des Alias-Effektes?

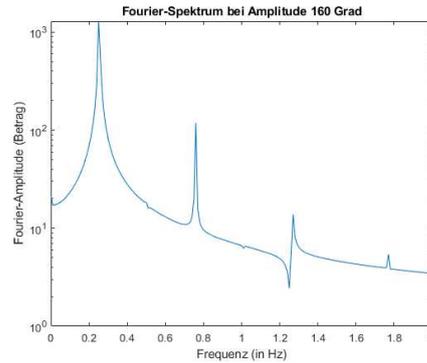
Aufgabe 96: Pendel mit großer Amplitude

Die Differentialgleichung des mathematischen Pendels lautet

$$\ddot{\phi} + \frac{g}{\ell} \sin \phi = 0$$

mit $\phi : t \mapsto \phi(t)$ die Winkelauslenkung (in Radiant) als Funktion der Zeit t (in Sekunden); Fallbeschleunigung g in m/s^2 ; Pendellänge ℓ in m.

Berechnen Sie die Lösung $\phi(t)$ an 1000 äquidistanten Datenpunkten im Bereich $0 < t \leq 100$ für Amplituden 45° , 90° , 170° . Zeichnen Sie jeweils ein Frequenzspektrum der Lösung. Ihre Zeichnungen könnten etwa so aussehen, wie hier rechts gezeigt.



Sie sollten aus den Graphiken ablesen können:

- Die erste Frequenzspitze (Grundfrequenz) liegt in der Nähe von

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{g}{\ell}}$$

(Das ist die klassische Formel für die Schwingungsfrequenz des Pendels bei kleinen Amplituden.)

- bei größerer Anfangsauslenkung verringert sich die Schwingungsfrequenz.
- Je größer die Anfangsauslenkung, desto mehr treten auch Frequenz-Spitzen bei höheren Frequenzen auf (Oberschwingungen).
- Die Frequenzen der Oberschwingungen sind ganzzahlige Vielfache der Grundfrequenz.
- Im Idealfall wären die Frequenz-Spitzen scharfe Zacken an jeweils einem Frequenzwert. Tatsächlich verteilen sich die Frequenzen im Signal über einige benachbarte Datenpunkte im Fourier-Spektrum (Leck-Effekt).

Geben Sie für Amplitude 170° an:

1. Um wieviel Prozent höher ist die Schwingungsdauer, verglichen mit der linearen Näherung für kleine Amplituden?
2. In welchem Verhältnis steht die Amplitude der ersten Oberschwingung zur Amplitude der Grundfrequenz?

10.3 Nichtlineare Gleichungssysteme: Beispiele zur Wiederholung

Die erste Teilaufgabe für den Kenntnissnachweis wird zu folgenden Themenkreisen gestellt:

- Gleichungssysteme: Newton-Raphson-Verfahren, `fsolve` oder einfache Fixpunkt-Iteration.
- Zeichnen von Kurven, graphische Lösung von Gleichungen.
- Darstellen von Funktionen $z : (x,y) \mapsto z(x,y)$ als Isolinien oder Flächen, graphisches Finden von Minima/Maxima.

Vergleichen Sie dazu auch die Aufgaben 29 und 30 der dritten Übungseinheit.

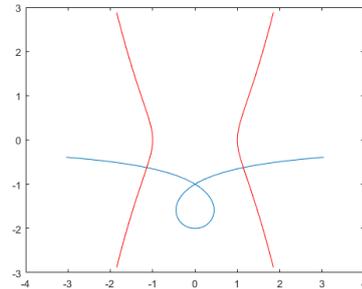
Aufgabe 97: Kurven in Parameterform

Zwei schon in der Antike untersuchte Kurven in Parameterdarstellung sind gegeben, eine *Konchoide von Nikomedes* und eine *Kampyle von Eudoxus*:

$$\begin{aligned} x &= s - \frac{2s}{\sqrt{1+s^2}} & x &= \pm \frac{1}{\cos(t)} \\ y &= -\frac{2}{\sqrt{1+s^2}} & y &= \frac{\sin(t)}{\cos^2(t)} \end{aligned}$$

Stellen Sie die beiden Kurven für $s \in [-5, 5]$ und $t \in [-1, 1]$ in der xy -Ebene dar und bestimmen Sie graphisch die Lage der Schnittpunkte. Hinweis: Die Kampyle besteht aus zwei Teilkurven, deswegen das \pm im x -Term. Zeichnen sie beide Teilkurven, einmal mit dem positiven, dann mit negativen Vorzeichen im x -Term.

Geben Sie die Lage der Schnittpunkte (auf 1 Nachkommastelle genau) an.



Aufgabe 98: Schnittpunkte rechnerisch bestimmt

Die Schnittpunkte aus der vorigen Aufgabe sind auch als Lösung des folgenden Gleichungssystems bestimmt.

$$\begin{aligned} x^4 &= x^2 + y^2 \\ 8y + 4 &= (x^2 - 3)y^2 + y^4 + 2y^3 \end{aligned}$$

(a) Finden Sie Lösungen des nichtlinearen Gleichungssystems im gegebenen Bereich mit der MATLAB-Funktion `fsolve`. Zwei passende Startvektoren können Sie aus der Grafik ablesen. Es gibt für dieses Gleichungssystem auch eine Lösung im Bereich $1 < x, y < 2$. Probieren Sie Startwerte aus diesem Bereich und finden Sie auch diese Lösung.

(b) Finden Sie die Lösungen des nichtlinearen Gleichungssystems mit dem Newton-Verfahren.

Aufgabe 99: Schraubenlinie

Die in Parameterdarstellung gegebene Schraubenlinie

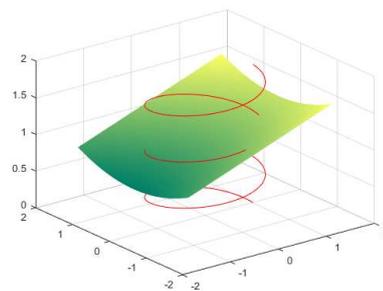
$$\begin{aligned} x &= \cos(s) \\ y &= \sin(s) & \text{für } s \in [0, 6\pi] \\ z &= \frac{s}{10} \end{aligned}$$

durchstößt die durch

$$z = \frac{x}{4} + \frac{y^2}{10} + 1$$

definierte Fläche in drei Punkten. Stellen Sie die Schraubenlinie für $0 \leq s \leq 6\pi$ mit `plot3` und die Fläche für $-1.5 \leq x, y \leq 1/5$ mit `surf` in einer 3D-Graphik dar. Lesen Sie aus der Darstellung die Lage aller Schnittpunkte ab.

Tipp: wählen Sie für Kurve und Fläche Farben mit hohem Kontrast. Schalten Sie das Gitternetz auf der Fläche mit `surf(X,Y,Z,'EdgeColor','none')` ab. Rotieren Sie die 3D-Graphik so, dass die Durchstoßpunkte gut zu sehen sind.



Geben Sie die Lage der Schnittpunkte $\mathbf{x} = [x; y; z]$ (auf 1 Nachkommastelle genau) an.

Aufgabe 100:

Die Schnittpunkte aus der vorigen Aufgabe sind auch als Lösung des folgenden Gleichungssystems bestimmt.

$$\begin{aligned}\frac{x}{4} + \frac{y^2}{10} + 1 &= z \\ x^2 + y^2 &= 1 \\ x \tan(10z) &= y\end{aligned}$$

Finden Sie die drei Lösungen des nichtlinearen Gleichungssystems mit dem Newton-Verfahren. Passende Startwerte können Sie aus der grafischen Darstellung entnehmen.

Geben Sie die Lösungen des Systems (mit 8 Nachkommastellen) an.