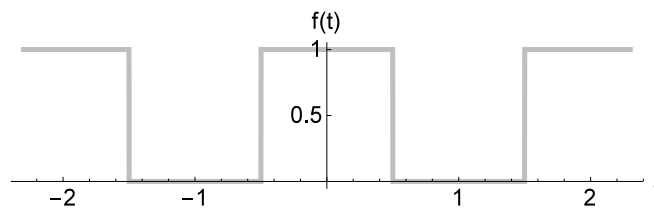


## Ü 10 Fourierreihen, Fourieranalyse, Fast Fourier Transform (FFT)

Es gibt im Hauptteil des Skriptums (noch) kein eigenes Kapitel dazu. Der Stoff wird anhand der Vorlesungsfolien (klick!) und der Übungsaufgaben erklärt und erarbeitet.

### Ü 10.1 Fourierreihen approximieren periodische Funktionen

Gegeben ist ein periodisches Signal  $f(t)$  in Form einer Rechtecks-Kurve mit Periode  $T = 2$ . Für  $-\frac{1}{2} < t < \frac{1}{2}$  ist  $f(t) = 1$ , für  $\frac{1}{2} < t < \frac{3}{2}$  ist  $f(t) = 0$ .



Approximieren Sie diese Funktion durch Fourierreihen. Verwenden Sie zuerst Terme bis zur Ordnung 1, dann 5 und schließlich 55.

Anleitung:

**Fourierreihe** einer periodischen Funktion  $f(t)$  mit Periode  $T$ :

$$f(t) = \frac{a_0}{2} + a_1 \cos\left(\frac{2\pi}{T}t\right) + a_2 \cos\left(\frac{2\pi}{T}2t\right) + \dots + a_n \cos\left(\frac{2\pi n}{T}t\right) + \dots \\ + b_1 \sin\left(\frac{2\pi}{T}t\right) + b_2 \sin\left(\frac{2\pi}{T}2t\right) + \dots + b_n \sin\left(\frac{2\pi n}{T}t\right) + \dots$$

mit

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cdot \cos\left(\frac{2\pi n}{T}t\right) dt \quad b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cdot \sin\left(\frac{2\pi n}{T}t\right) dt$$

Summen-Symbol  $\Sigma$  und Kreisfrequenz  $\omega = 2\pi/T$  vereinfachen die Schreibweise. Außerdem kommt es bei den Integralen nur darauf an, über eine volle Periode zu integrieren; ob von  $-T/2$  bis  $T/2$ , von 0 bis  $T$  oder über irgend einen anderen Bereich, der eine volle Periode abdeckt, ist egal. Daher sind in folgender Formulierung die Grenzen beginnend mit irgendeinem  $t_0$  bis  $t_0 + T$  angegeben.

**Fourierreihe** für  $f(t)$  mit Periode  $T$  und Kreisfrequenz  $\omega = \frac{2\pi}{T}$  :

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t)$$

mit

$$a_n = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cdot \cos(n\omega t) dt \quad b_n = \frac{2}{T} \int_{t_0}^{t_0+T} f(t) \cdot \sin(n\omega t) dt$$

Im Beispiel hier ist  $T = 2$ . Die Funktion  $f$  ist aber nur im Bereich  $-\frac{1}{2} < t < \frac{1}{2}$  ungleich 0 und dort sogar konstant mit Wert  $f(t) = 1$ . Das vereinfacht die Berechnung der Integrale ganz wesentlich:

- Die Iteration läuft nicht von  $-T/2$  bis  $T/2$ , sondern nur dort wo  $f \neq 0$ , also im Intervall  $-\frac{1}{2} < t < \frac{1}{2}$ .
- Beachten Sie: dieses Signal ist symmetrisch bezüglich der  $y$ -Achse! Weil  $\sin(a) = -\sin(-a)$ , heben sich bei der Integration von (Signal mal Sinus-Term) die Beiträge der positiven und der negativen  $t$ -Werte auf, alle  $b_n$ -Koeffizienten sind hier gleich 0.

Es bleibt also zu berechnen:

$$a_n = \int_{-1/2}^{1/2} 1 \cdot \cos(n\pi t) dt$$

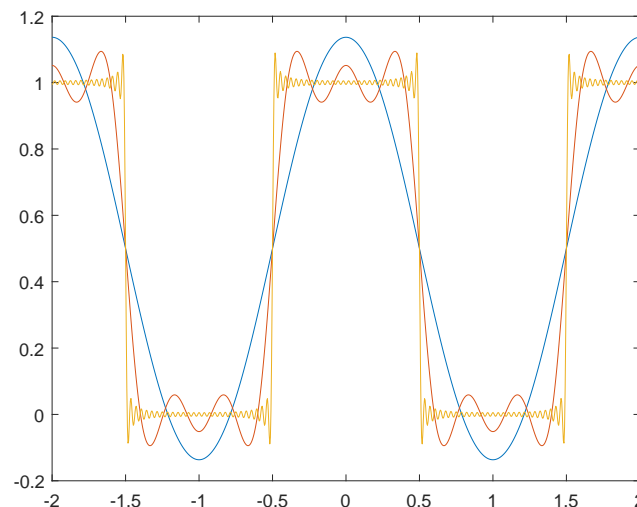
Ergebnis:

$$\begin{aligned} a_0 &= 1 \\ a_n &= (-1)^{(n-1)/2} \cdot \frac{2}{n\pi} && \text{für } n > 0 \text{ ungerade} \\ a_n &= 0 && \text{für } n > 0 \text{ gerade} \end{aligned}$$

Die Fourierreihe lautet somit

$$f(t) = \frac{1}{2} + \frac{2}{\pi} \left( \cos(\pi t) - \frac{1}{3} \cos(3\pi t) + \frac{1}{5} \cos(5\pi t) - \frac{1}{7} \cos(7\pi t) \pm \dots \right)$$

Und so sehen die Approximationen bis  $n = 1, 5, 55$  aus:



Sie können versuchen, noch viel mehr Terme zu summieren, damit die Approximation genauer wird. Aber ganz egal, wie viele Terme Sie summieren:

**Gibbs'sches Phänomen:** Bei Approximation unstetiger Funktionen durch Fourier-Summen treten in der Umgebung von Sprungstellen Überschwüngen auf. Diese überschießenden Zacken werden mit zunehmendem  $n$  schmaler, die maximalen Auslenkung bleibt aber konstant im Bereich  $\approx 10\%$  der Sprunghöhe.

### Aufgabe 85: Rechtecks-Funktion und Gibbs-Phänomen

Gegeben ist ein periodisches Rechtecks-Signal wie oben, nur mit schmälere Rechtecken: Periode  $T = 2$  wie vorher, aber nur für  $-\frac{1}{4} < t < \frac{1}{4}$  ist  $f(t) = 1$ , sonst ist  $f(t) = 0$ .

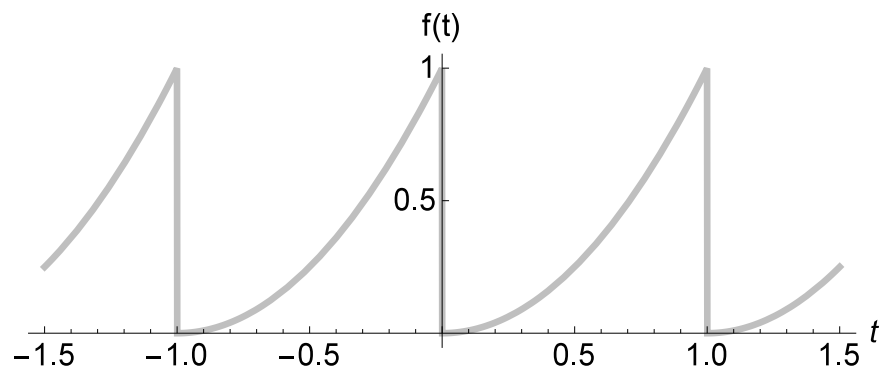
Berechnen Sie die Terme der Fourierreihe (Nehmen Sie MATLABs symbolische Integration zu Hilfe, wenn Sie die Integrale mit Papier, Stift und Hirn alleine nicht auswerten können!)

Zeichnen Sie die Approximationen bis  $n = 1, 5, 55$  und vergleichen Sie mit der obigen Abbildung.

Die Approximationen zeigen das Gibbs-Phänomen. Lesen Sie aus der Grafik die maximale Höhe ab. Bessert sich dieser Wert, wenn Sie  $n$  weiter erhöhen? (Wählen Sie  $n$  so hoch, wie sie wollen oder MATLAB vernünftig rechnen kann.)

### Aufgabe 86: Schräge Säge

Das ist ein Beispiel zum allgemeinen Fall, in dem sowohl Sinus- als auch Cosinus-Terme auftreten. Gegeben ist die Funktion  $y = t^2$  für  $0 \leq t \leq 1$ , periodisch fortgesetzt außerhalb dieses  $t$ -Intervalls.



Berechnen Sie bis  $n = 5$  die Koeffizienten der Fourierreihe. Stellen Sie  $y$  und die Fourier-Approximation dar.

Sie können MATLABs symbolische Integration verwenden, zum Beispiel

```
>> int((1-t^2)*cos(3*pi*t),t,0,1)
ans =
2/(9*pi^2)
```

Zeichnen Sie auch Approximationen mit höherem  $n$  und erklären Sie den Begriff „Gibbs'sches Phänomen“ anhand der Darstellungen.

### Aufgabe 87: Leistung, quadratischer Mittelwert

Gegeben ist noch einmal das Rechtecks-Signal  $f(t)$  mit Periode  $T = 2$ . Für  $-\frac{1}{2} < t < \frac{1}{2}$  ist  $f(t) = 1$ , für  $\frac{1}{2} < t < \frac{3}{2}$  ist  $f(t) = 0$ .

In vielen Anwendungen gibt der *quadratische Mittelwert*

$$\frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)^2 dt$$

die Leistung oder Energie an.

Berechnen Sie diesen Wert für das Rechtecks-Signal.

Die Fourierreihe dieser Funktion ist oben angegeben. Summieren Sie die Quadrate aller Fourier-Koeffizienten bis bis  $n = 1, 5, 55$  gemäß folgender Formel (Achtung, der Koeffizient  $a_0$  ist in den Fourier-Formeln immer ein Sonderfall, er stiftet regelmäßig Verwirrung!).

$$P = \frac{a_0^2}{4} + \frac{1}{2} \sum_{n>1} a_n^2 + b_n^2$$

Was fällt Ihnen im Vergleich zum vorher berechneten Integral auf?

In der Physik und den Ingenieurwissenschaften wird dieser Zusammenhang so formuliert: Die Energie eines Signals im Zeitbereich (Integral über  $f^2$ ) ist (bis auf Skalierung) gleich seiner Energie im Frequenzbereich. Jede Frequenz  $> 0$  trägt  $\frac{1}{2}(a_n^2 + b_n^2)$  zur Gesamtenergie bei. Bei komplexer Schreibweise wäre dieser Zusammenhang einfacher, weil dort der Koeffizient zu  $n = 0$  kein Sonderfall ist.

Wie groß ist bei der Rechtecks-Funktion der Beitrag der niedrigsten drei Frequenzen (der ersten drei Terme  $\neq 0$  in der Fourierreihe) im Verhältnis zur Gesamtenergie?

## Ü 10.2 Fourieranalyse findet Frequenzen, FFT findet' flotter.

### Aufgabe 88: Frequenz-Analyse eines Audio-Signals

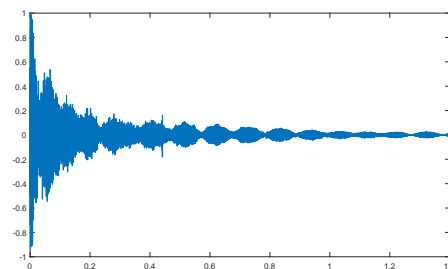
Fourier-Analyse zerlegt ein Signal in seine Frequenzanteile. Diese Anleitung zeigt, wie Sie alle Frequenzkomponenten berechnen, aus denen ein Audio-Signal besteht.

Laden Sie dazu die Datei `klang1.wav` von der Übungshomepage (klick!) 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

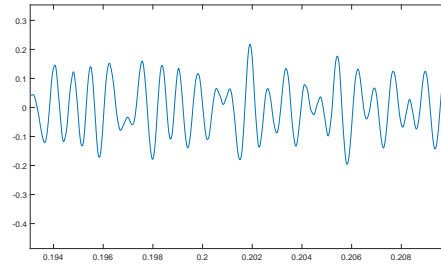
$$\frac{\text{Anzahl der Datenpunkte} - 1}{\text{Abtastfrequenz}}$$



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 einen annähernd 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  $\approx 1400$  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  $\mathbf{X}$  entsprechen ebensoviele Fourier-Koeffizienten in der diskreten Fourier-Transformierten  $\mathbf{Y}$ .

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 (die Formel mit den  $a_n$ - und  $b_n$ -Termen). 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  $\omega_k$ )

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  $X$  und  $Y$ . 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  $\omega_k$ . Die Umrechnung oder Zuordnung von Index  $j$  zu Zeitpunkt  $t_j$  und Fourier-Komponente  $k$  zu Kreisfrequenz  $\omega_k$  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  $2\pi$  multipliziert sind das die Kreisfrequenzen  $\omega_k$ .) 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.

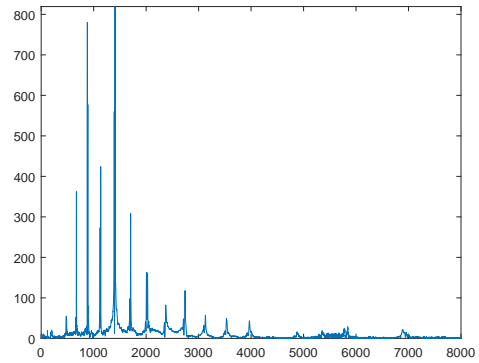
So stellen Sie das Frequenzspektrum dar:

```
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 interessanten Frequenzbereich (interaktiv oder mit dem Befehl: `axis([0 8000 0 max(abs(Y))])`).

## Aufgabe 89: Klang einer Gitarren-Saite

Die Datei `klang2.wav` (siehe Übungshomepage (klick!)) 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 5000 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 90: 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 anderem) 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 Hauptdeklinatationstide	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.

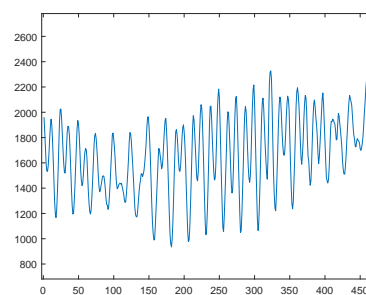
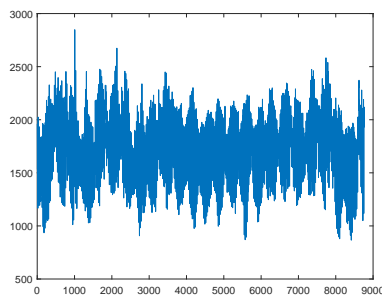
(Quelle: <http://uhs1c.soest.hawaii.edu/data/download/fd#uh829>).

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?

### Aufgabe 91: Pendel mit großer Amplitude

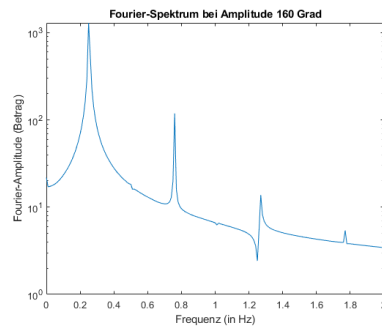
Die Differentialgleichung des mathematischen Pendels lautet

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

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

Wählen Sie Werte für  $\ell$  und  $g$  und berechnen Sie die Lösung  $\phi(t)$  an 1000 äquidistanten Datenpunkten im Bereich  $0 < t \leq 100$  für Amplituden  $45^\circ$ ,  $90^\circ$ ,  $170^\circ$ . 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  $170^\circ$  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?

(Diese Resultate sind unabhängig von Ihrer konkreten Wahl für  $\ell$  und  $g$ .)

### Ü 10.3 Aliasing und Leakage

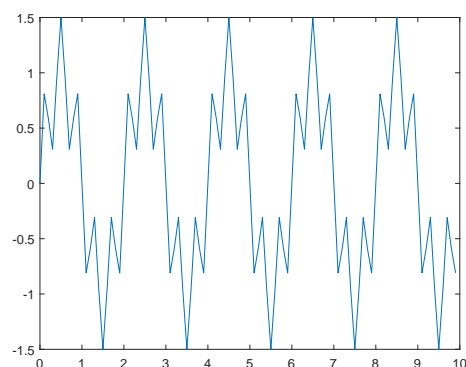
Die Fourieranalyse eines Signals, für das Daten nur zu endlich vielen diskreten Zeitpunkten vorliegen, kann nicht alle im kontinuierlichen, unbegrenzt andauernden Signal vorhandenen Frequenzen exakt erkennen. Zwei wesentliche Einschränkungen sind Aliasing und Leakage. Dazu stellt dieser Abschnitt Beispiele vor.

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);
```

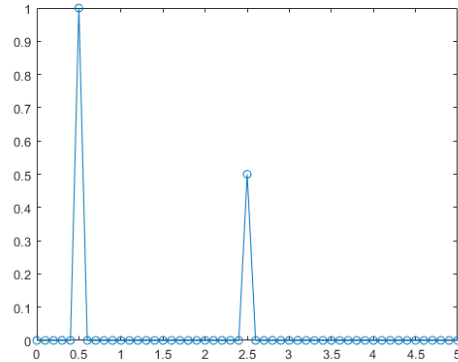


Fourier-Analyse zeigt korrekt die beiden Frequenzen 0,5 Hz und 2,5 Hz an. Auch die entsprechenden Amplituden stimmen, weil im Plot mit  $2/n$  skaliert wurde).

```

%% 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')

```



Diese Darstellung (scharfe Frequenzpeaks an den korrekten Stellen) ist aber nicht der Normalfall. Die nächste Aufgabe zeigt: Frequenzen können an der falschen Stelle angezeigt (Aliasing) oder unscharf verschmiert (Leakage) sein.

### Aufgabe 92: Verschwundene Frequenzen, fälschlich versteckt oder verschmiert

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!

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?

**Leakage-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.

Leakage kann auch als eine Form von Aliasing gesehen werden: Die korrekte Frequenz tritt im Spektrogramm zum Teil als benachbarte Frequenzen auf.

Die Abbildung zu Aufgabe 91 lässt auch ein Leakage-Phänomen erkennen: der erste Peak ist nicht so nadelscharf, wie er theoretisch sein sollte. Beim dritten Peak zeigt sich ein Ausreißer nach unten. Diese Störeffekte gehen hauptsächlich auf die endliche Signaldauer und Abtastfrequenz zurück.

### Aufgabe 93: 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.

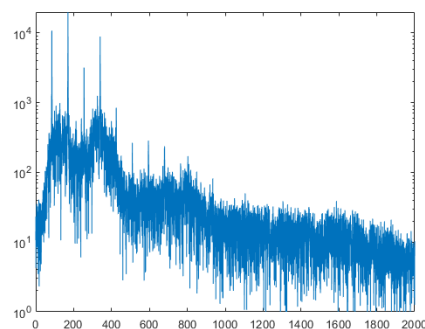


Auf dem Video dreht sich der Propeller scheinbar recht langsam. Wenn Sie genau hinsehen und auf den Anstellwinkel der Propellerblätter achten, dreht er sich sogar rückwärts! Keine Panik, das sind Aliasing-Effekte. Die tatsächliche Propellerdrehzahl können Sie aus dem Betriebsgeräusch durch Fourieranalyse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

## Ü 10.4 Filtern von Signalen im Fourier-Raum

Beispiele zum Filtern von Datenvektoren und Datenfeldern.

### Aufgabe 94: Bearbeiten von Audio-Signalen im Fourier-Raum

Lesen Sie die Datei `klang2.wav` ein und lassen Sie sie abspielen:

```
[X,fs] = audioread('klang2.wav');  
n = length(X);  
sound(X,fs)
```

Wenden Sie FFT auf das Signal `X` an.

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

Sie sollen nun das Signal im Frequenzraum bearbeiten. Das bearbeitete Signal können Sie rücktransformieren und das Ergebnis abhören.

Zuordnung Index  $k$  im  $Y$ -Vektor  $\mapsto$  Frequenz  $f_k$  im Frequenzraum:

$$f_k = \frac{k-1}{n} f_s \quad \text{für } k \leq \frac{n}{2}$$

Sie können Komponenten in  $Y$  verstärken oder schwächen und damit die entsprechenden Frequenzen im Audiosignal beeinflussen.

Ein Filter-Vektor  $F$  multipliziert ein Signal  $Y$  im Fourierraum:

$$Y_{\text{gefiltert}} = Y \cdot F$$

Für einen reellen Signalvektor  $X$  und dessen Fouriertransformierte  $Y$  müssen die Komponenten  $F(k)$  des Filtervektors  $F$  für  $k > 1$  symmetrisch bezüglich Index  $n/2 + 1$  liegen. In MATLABs Schreibweise:

$$F(n/2+2:n) = F(n/2:-1:2);$$

Definieren Sie dazu verschiedene Filter-Funktionen.

- Tiefpass-Filter: lässt nur Frequenzen bis zu einer bestimmten Grenzfrequenz  $f_{\text{cut}}$  durch; höhere Frequenzen werden unterdrückt. Hier zum Beispiel ein Tiefpass-Filtervektor mit Grenzfrequenz 200 Hz:

```
Filt = ones(size(Y));  
fcut=200;  
ncut=round(fcut/fs*n)+1; % Index im Y-Vektor  
Filt(ncut:n/2) = 0;  
% Filter muss symmetrisch um n/2+1 liegen  
Filt(n/2+2:n) = Filt(n/2:-1:2);
```

Wenden Sie diesen Filter an, transformieren Sie das Signal zurück vom Frequenz- in den Zeitbereich und hören Sie sich den Unterschied an:

```
Y1 = Y.*Filt;  
X1 = ifft(Y1);  
sound(X1,fs)
```

Das Original-Signal war der Klang einer tiefen A-Saite auf der Gitarre, Grundton 110 Hz. Ein Tiefpass-Filter mit 200 Hz schneidet sämtliche Obertöne ab. Das gefilterte Signal klingt deswegen etwas dumpf und leer.

Probieren Sie aus: wenn Sie Grenzfrequenz 100 Hz wählen, hören Sie gar keinen Ton mehr, weil nun auch die Grundfrequenz weggefiltert wird. Wenn Sie höhere Grenzfrequenzen wählen, klingt das gefilterte Signal immer ähnlicher dem Ausgangssignal.

- Hochpass-Filter: lässt nur Frequenzen ab einer bestimmten Grenzfrequenz  $f_{\text{cut}}$  durch; tiefere Frequenzen werden unterdrückt.

Probieren Sie aus: Je höher Sie die Grenzfrequenzen wählen, desto schärfer klingt das gefilterte Signal.

- Bandpass-Filter: lässt nur Frequenzen innerhalb eines Bereiches  $f_{\text{min}} < f < f_{\text{max}}$  durch.
- Sie können auch wildere Filter-Funktionen wählen. Achten Sie aber darauf, dass der Filter-Vektor symmetrisch um  $n/2+1$  liegen muss: `Filt(n/2+2:n) = Filt(n/2:-1:2);`. (Andernfalls enthält das rücktransformierte Signal imaginäre Anteile!) Auch müssen Filter- und Datenvektor die gleiche Gestalt (beide jeweils Zeilen- oder Spalten-Vektoren) sein; andernfalls führt die Anwendung der Filterfunktion `Y.*Filt` zu Fehlermeldungen.
- Wenn Ihnen das Erstellen eines symmetrische Filter-Vektors zu kompliziert ist, können Sie der Einfachheit halber beim rücktransformierten Signal den Imaginärteil abschneiden. Sie schreiben dann statt `X1 = ifft(Y1)`; eben `X1 = real(ifft(Y1))`; Das ist zwar eine grobe Vorgangsweise, aber sie funktioniert.

Das Signal von Datei `klang2.wav` ist allerdings nicht sehr komplex. Besser hören Sie den Einfluss verschiedener Filter zum Beispiel, wenn Sie die Aufnahme einer menschlichen Stimme bearbeiten.

### Aufgabe 95: Bearbeiten von Bild-Dateien im Fourier-Raum

Die folgenden Befehle lesen eine Bilddatei ein und zeigen eine Schwarzweiß-Version davon an:

```
% Einlesen einer Bilddatei
photoarray = imread('katzekatze.jpg');

% Eine Standard-Formel rechnet RGB-Werte auf Grautöne um
redpix = single(photoarray(:,:,1))/255;
greenpix = single(photoarray(:,:,2))/255;
bluepix = single(photoarray(:,:,3))/255;
graypix = 0.299*redpix + 0.587*greenpix + 0.114*bluepix;
graypix = rot90(graypix,2); % so wird Foto seitenrichtig angezeigt...

figure(1)
imagesc(graypix)
colormap(gray), axis off image

% Fouriertransformierte
Y = fft2(graypix);
```

Der Befehl `fft2` führt eine 2-dimensionale Fouriertransformation des Datenfeldes durch. Sie können nun im Fourier-Raum gezielt Frequenzen verstärken oder abschwächen. Wenn Sie nur die niedrigsten Frequenz-Anteile behalten (Tiefpass-Filter), wird das Bild unscharf und weichgezeichnet:

```
% Filter Tiefpass
% Anzahl Fouriermoden, die bleiben
% n=8; % fast nicht erkennbar
% n=16; % erkennbar, sehr unscharf
% n=30; % unscharf
% n=50; % unscharf
% n=100 % maessig gut
% n=200 % ganz gut!

Y(n:end-n,:) = 0;
Y(:,n:end-n) = 0;

y = real(iff2(Y));
figure(3)
imagesc(y)
colormap(gray), axis off image
```

Wenn Sie Fourier-Komponenten ändern, enthält das rücktransformierte Signal imaginäre Anteile. Nur wenn die Filter-Funktion gewisse Symmetrie-Eigenschaften hat, ist auch das rücktransformierte Signal reell. Bei der eindimensionalen FFT des Audio-Signals haben wir das berücksichtigt. Im zweidimensionalen Fall ist die Symmetrie-Bedingung komplizierter. Wir schneiden deswegen der Einfachheit halber beim rücktransformierten Signal den Imaginärteil ab. (Das ist eine *Hau-Ruck*-Methode; sie funktioniert, aber die Signal-Energie insgesamt wird um den Anteil der abgeschnittenen Imaginärteile verringert.)

Sie können auch wildere Filter-Funktionen ausprobieren und sehen, wie sich Änderungen im Fourier-Raum auf das Bild auswirken.