AUSGEWÄHLTE KAPITEL AUS "NUMERISCHE METHODEN IN DER PHYSIK"

H. Sormann SS 2007

Zweites Thema: Diskrete Fourier-Transformation

Joseph Fourier (Auxerre 1768 - Paris 1830)

In dieser Übung sollen eine Reihe von wichtigen Anwendungen der diskreten Fast-Fourier-Transformation (DFFT) ausprobiert werden: Frequenzanalyse, Datenglättung, Faltung, Korrelation

1. Frequenzanalyse
2. Faltung von Funktionen im Zeit- und Frequenzraum
3. Filterung eines gemessenen Signals mittels DFFT
4. Einfluß statistischer Fehler (noise) auf die Rekonstruktion einer Signalkurve
5. Reduktion des noise mittels der Methode von Dirichlet
6. Korrelation zweier Funktionen
7. Autokorrelation
8. Kreuzkorrelation

1. Frequenzanalyse (Spektralanalyse)

Die Frequenz- oder Spektralanalyse ist die "klassische" Anwendung der diskreten schnellen Fourier-Transformation (*discrete fast Fourier transform* DFFT).

Es geht dabei um die Darstellung eines reellen (periodischen) Signals als Linearkombination von Cosinus- und Sinus-Schwingungen verschiedener Frequenzen. Die theoretischen Grundlagen zu diesem Thema finden Sie in meinem Vorlesungsskriptum "Numerische Methoden in der Physik" und im Übungsskriptum "Computer-Tomographie" zu dieser Lehrveranstaltung.

Als Basis einer Spektralanalyse dienen die Funktionen (9)-(11) aus dem Übungsskriptum "Computer-Tomographie". Ich habe allerdings auf Grund einiger Testrechnungen beschlossen, die Gleichungen (10) und (11), welche die von mir bereits eingehend diskutierte Frequenz-Optimierung beschreiben, geringfügig zu variieren, und zwar auf die folgende Weise:

$$\Re\{I(t)\}_{\text{ideal}} = \sum_{q=0}^{n-1} \left[\Re\left\{\frac{Y_q}{n}\right\} \cos(2\pi\nu_q t) + \Im\left\{\frac{Y_q}{n}\right\} \sin(2\pi\nu_q t) \right] \tag{1}$$

mit

$$\nu_q = \frac{q}{P} \qquad \text{für} \quad q = 0, \ldots, \frac{n}{2} \tag{2}$$

und

$$\nu_q = \frac{q-n}{P} \qquad \text{für} \quad q = \frac{n}{2}+1, \ldots, n-1 \quad . \tag{3}$$

D. h., gegenüber den bisher verwendeten Formeln fällt die Verschiebung nach links um eine Position geringer aus! Wie meine Tests ergeben haben, hat diese Änderung keine negative Auswirkung auf die erhaltenen Interpolationskurven, bringt aber bei der Anwendung der Frequenz-Optimierung einige Vorteile.

Achtung Matlab-User: Um die obigen Gleichungen korrekt verwenden zu können, müssen Sie unbedingt darauf achten, daß die vom FFT-Programm gelieferten Fourierkoeffizienten in der korrekten Reihenfolge ausgegeben werden.

Dies ist bei den von mir zur Verfügung gestellten C- und F90-Programmen der Fall, nicht aber beim internen Matlab-Programm *fft*. Bitte ordnen Sie die Matlab-Ergebnisse in der Weise um, wie ich es bereits im Übungsskriptum "Computer-Tomographie" angegeben habe:

```
yy=fft(werte);
daten=zeros(1,n);
daten(1)=yy(1);
for ind=2:n
  daten(ind)=yy(n+2-ind);
end
```

Man geht nun davon aus, daß die gesamte Information über die reelle, periodische Signalfunktion $s(t)$ innerhalb der entsprechenden Periode P enthalten ist, und daß innerhalb dieser Periode eine Zahl von $n = 2^m$, $m \in N$ äquidistanten Funktionswerten von $s(t)$ gegeben ist[1,2]:

$$t_j = j\,\Delta \qquad s_j = s(t_j) \qquad (j = 0, \ldots, n-1) \tag{4}$$

Der zeitliche Anstand der Stützpunkte Δ hängt mit der Periode P und der Zahl der Stützpunkte n über

$$P = n\,\Delta \tag{5}$$

zusammen, und die zeitliche Dichte der gegebenen Punkte wird als Abtast-Rate (*sampling rate*) bezeichnet:

$$\text{sampling rate } = \frac{n}{P} = \frac{1}{\Delta}\,. \tag{6}$$

Es ist natürlich klar, daß durch das nur diskrete Abtasten des Signals ein Informationsverlust auftritt, der in der Regel umso größer sein wird, je geringer die Abtast-Rate ist.

Eine weitere Konsequenz der diskreten FT besteht darin, daß auch die ermittelten Frequenzen ν_q nicht kontinuierlich verteilt sind, sondern ebenfalls eine äquidistante Punktmenge auf der Frequenz-Achse repräsentieren. Wie aus den Gleichungen (1)-(3) unmittelbar hervorgeht, ergeben sich für n *sampling points* innerhalb der Periode P die Frequenzen

$$-\left(\frac{n}{2P}\right) < \nu \leq \left(\frac{n}{2P}\right) \tag{7}$$

mit dem Abstand zwischen zwei Frequenzpunkten (der Frequenz-Auflösung)

$$\Delta\nu = \frac{1}{P}\,. \tag{8}$$

Wegen der "Frequenz-Optimierung" (2) und (3) sind diese Frequenzen um $\nu = 0$ verteilt über einen Bereich, der nach links und nach rechts durch die sog. Nyquist-Frequenz

$$\nu_{Ny} = \frac{n}{2P} \tag{9}$$

begrenzt ist.

Welche Informationen erhält man aus der Gleichung (1)?

- Der (durch n dividierte) Realteil (Imaginärteil) des q-ten Fourierkoeffizienten ergibt die Amplitude des Cosinus- (Sinus-)Anteils zur Frequenz ν_q der Interpolationsfunktion $\Re\{I(t)\}$.

[1] $n = 2^m$ ist die Bedingung dafür, daß die FFT-Methode anwendbar ist.

[2] Nur bei äquidistanten Punktmengen kann man die Orthogonalitätseigenschaften der Fourier-Basisfunktionen ausnützen.

Aufgabe 1: Einige Tests bzgl. der Spektralanalyse

Im folgenden sollen Sie nun an Hand eines einfachen Testsignals die wichtigsten Aspekte einer Spektralanalyse periodischer Funktionen kennenlernen. Für die ersten Schritte werde ich Sie dabei "begleiten", weitere Tests sollen Sie selbständig durchführen.

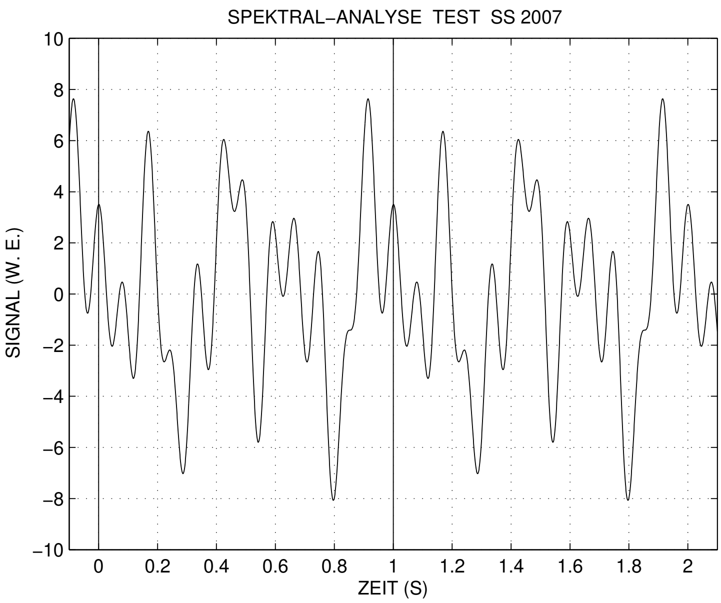
Schreiben Sie ein Programm, mit welchem Sie als erstes in der Lage sind, einen Satz von Testdaten für eine Spektralanalyse aus der folgenden Signalfunktion zu erstellen:

$$s(t) = a_1 \cos(2\pi\nu_1 t) + a_2 \sin(2\pi\nu_2 t) + a_3 \cos(2\pi\nu_3 t) + a_4 \sin(2\pi\nu_4 t) + a_5 \cos(2\pi\nu_5 t)\,. \tag{10}$$

Nehmen Sie für die fünf Amplituden und Frequenzen der Testfunktion die folgenden Werte:

```
 nr           a         frequ(Hz)
--------------------------------
 1    COS    2.0          2.0
 2    SIN   -3.0          4.0
 3    COS   -1.0          4.0
 4    SIN    2.0          7.0
 5    COS    2.5         12.0
--------------------------------
```

Wenn sie diese Signalfunktion für - sagen wir - 5 Sekunden auswerten und grafisch darstellen, erkennen Sie ohne Probleme, daß die Funktion eine Periode von 1 Sekunde hat.



Damit ist ein wesentlicher erster Parameter der nun folgenden Frequenz-Analyse fixiert:

Alle Stützpunkte für die Fourier-Transformation liegen im Zeitintervall zwischen 0 und 1 Sekunde, wobei der erste der äquidistanten Datenpunkte bei $t = 0$ *liegt und der letzte knapp vor* $t = 1$.

Berechnen Sie nun 2^6=64 Stützpunkte, und machen Sie mit diesem Datenmaterial mittels der Gleichungen (1) - (3) eine Frequenz-Analyse:

- Unterwerfen Sie die 64 Datenpunkte einer *fast Fourier transform*;
- Produzieren Sie eine Tabelle der erhaltenen Real- und Imaginärteile der Fourierkoeffizienten (jeweils durch die Stützpunktzahl n dividiert), und weisen Sie jedem Fourierkoeffizienten eine Frequenz gemäß Glg. (2) und (3) zu.
- Präsentieren Sie Ihre Ergebnisse auch in Form eines "Linienspektrums": mit Hilfe von Matlab ist dies besonders einfach möglich:

```
stem(frequenzen,FK-Realteile/n,'ro');
hold on
stem(frequenzen,FK_Imagteile/n,'k*');
xlabel('FREQUENZ (HZ)');
ylabel('AMPLITUDE (W.E.)');
legend('Cosinus-Amplituden','Sinus-Amplituden');
hold off
```

- Berechnen Sie unter Verwendung von Glg. (1) den Realteil der Interpolationsfunktion $I(t)$, und vergleichen Sie diese Funktion mit der Signalfunktion $s(t)$ [Eq. (10)]:
  Wenn Ihre Rechnung richtig ist, muß die Interpolationsfunktion mit der Signalfunktion vollkommen identisch sein!

Wie sollen nun die Ergebnisse im Detail aussehen?

Was die Wertetabelle angeht, sollte Ihr Resultat wie folgt lauten:

```
Frequenz(Hz)  cos-Ampl.  sin-Ampl.

   0.0000      -0.0000    0.0000
   1.0000      -0.0000   -0.0000
   2.0000       1.0000    0.0000
   3.0000      -0.0000    0.0000
   4.0000      -0.5000   -1.5000
   5.0000       0.0000    0.0000
   6.0000       0.0000    0.0000
   7.0000      -0.0000    1.0000
   8.0000      -0.0000   -0.0000
   9.0000      -0.0000    0.0000
  10.0000      -0.0000   -0.0000
  11.0000      -0.0000    0.0000
```

12.0000	1.2500	0.0000
13.0000	0.0000	0.0000
14.0000	0.0000	-0.0000
15.0000	0.0000	-0.0000
16.0000	-0.0000	-0.0000
17.0000	-0.0000	-0.0000
18.0000	-0.0000	0.0000
19.0000	-0.0000	0.0000
20.0000	0.0000	0.0000
21.0000	0.0000	-0.0000
22.0000	0.0000	-0.0000
23.0000	-0.0000	-0.0000
24.0000	0.0000	-0.0000
25.0000	-0.0000	-0.0000
26.0000	-0.0000	0.0000
27.0000	-0.0000	0.0000
28.0000	0.0000	-0.0000
29.0000	0.0000	-0.0000
30.0000	-0.0000	-0.0000
31.0000	-0.0000	0.0000
32.0000	-0.0000	0.0000
-31.0000	-0.0000	-0.0000
-30.0000	-0.0000	0.0000
-29.0000	0.0000	0.0000
-28.0000	0.0000	0.0000
-27.0000	-0.0000	-0.0000
-26.0000	-0.0000	-0.0000
-25.0000	-0.0000	0.0000
-24.0000	0.0000	0.0000
-23.0000	-0.0000	0.0000
-22.0000	0.0000	0.0000
-21.0000	0.0000	0.0000
-20.0000	0.0000	-0.0000
-19.0000	-0.0000	-0.0000
-18.0000	-0.0000	-0.0000
-17.0000	-0.0000	0.0000
-16.0000	-0.0000	0.0000
-15.0000	0.0000	0.0000
-14.0000	0.0000	0.0000
-13.0000	0.0000	-0.0000
-12.0000	1.2500	-0.0000
-11.0000	-0.0000	-0.0000
-10.0000	-0.0000	0.0000
-9.0000	-0.0000	-0.0000
-8.0000	-0.0000	0.0000
-7.0000	-0.0000	-1.0000
-6.0000	0.0000	-0.0000
-5.0000	0.0000	-0.0000

```
  -4.0000       -0.5000   1.5000
  -3.0000       -0.0000  -0.0000
  -2.0000        1.0000  -0.0000
  -1.0000       -0.0000   0.0000
```

Nachdem Sie 64 Stützpunkte verwendet haben, erhalten Sie natürlich auch 64 Fourierkoeffizienten, und zwar - wegen der Frequenz-Optimierung - für Frequenzen von -31 Hz bis +32 Hz mit einer Frequenz-Auflösung gemäß Glg. (8) von 1 Hz. Diese "Grenzfrequenz" paßt natürlich genau zu der in Glg. (9) definierten *Nyquist-Frequenz*

$$\nu_{Ny} = \frac{64}{2 \cdot 1} = 32 \quad \text{Hz.}$$

Es fällt Ihnen sofort auf, daß für die allermeisten Frequenzen sowohl die Cosinus-Amplitude als auch die Sinus-Amplitude Null sind: dies ist keineswegs überraschend, weil ja das zu analysierende Signal nur wenige Cosinus- und Sinus-Terme aufweist.

Um solche langen Tabellen mit vielen "Null-Zeilen" zu vermeiden, sollten Sie in Ihr Programm eine Abfrage einbauen, die den Output solcher "Null-Zeilen" verhindert. Wenn Sie das tun, reduziert sich die obige Tabelle zu

```
Frequenz(Hz)  cos-Ampl.  sin-Ampl.

   2.0000        1.0000   0.0000
   4.0000       -0.5000  -1.5000
   7.0000       -0.0000   1.0000
  12.0000        1.2500   0.0000
 -12.0000        1.2500  -0.0000
  -7.0000       -0.0000  -1.0000
  -4.0000       -0.5000   1.5000
  -2.0000        1.0000  -0.0000
```

Wie Sie sehen - und das ist typisch für die Fourier-Analyse <u>reeller</u> Signaldaten, kommen die aufgefundenen Frequenzen stets "paarweise" vor, und zwar jeweils mit positiven und negativen Vorzeichen, wobei die zugehörigen Cosinus-Amplituden gleich sind und die zugehörigen Sinus-Amplituden verschiedene Vorzeichen haben. Wenn man nun bedenkt, daß die Cosinusfunktion eine <u>gerade</u> Funktion ist und die Sinusfunktion eine <u>ungerade</u>, erkennt man leicht, daß es genügt, *nur die positiven Frequenzen auszugeben und alle zugehörigen Amplituden mit 2 zu multiplizieren.*

<u>Achtung:</u> Wenn Sie die "ausführliche Tabelle" auf den Seiten 5 und 6 genau ansehen, erkennen Sie, *daß die <u>erste</u> und <u>letzte</u> positive Frequenz (entspricht der Frequenz Null und der Nyquist-Frequenz) nur <u>einfach</u> vorkommen. Die Amplituden dieser beiden Zeilen sind daher von der Multiplikation mit 2 ausgenommen !!*

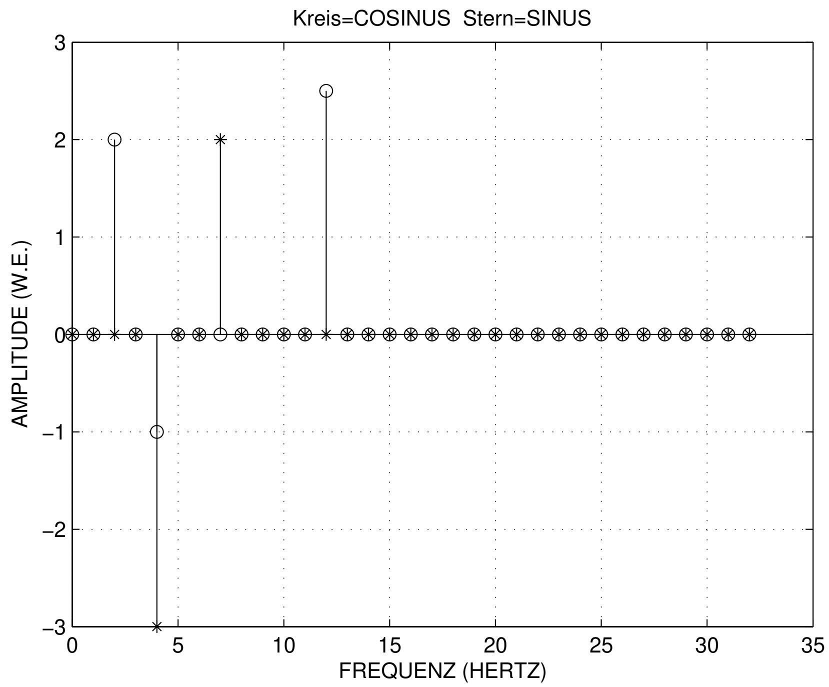
Auf diese Weise reduziert sich die Ausgabe-Tabelle nochmals zu

```
Frequenz(Hz)  cos-Ampl.  sin-Ampl.

   2.0000       2.0000    0.0000
   4.0000      -1.0000   -3.0000
   7.0000      -0.0000    2.0000
  12.0000       2.5000    0.0000
```

Diese Ergebnisse passen exakt zur Testfunktion (10).

Das zugehörige Linienspektrum sieht wie folgt aus:



<u>Weitere Tests im Rahmen der Aufgabe 1:</u>

- Wiederholen Sie nun diese Spektralanalyse mit kleineren Stützpunktzahlen, nämlich mit

  $$n = 32 \qquad \text{bzw.} \qquad n = 16\,.$$

  Diskutieren Sie an Hand der Lösungstabelle, des Linienspektrums und der Interpolationskurve, welche Effekte dabei auftreten.

  Beobachten Sie vor allem, was geschieht, wenn die Nyquist-Frequenz (9) kleiner wird als die größte, in der Signalfunktion enthaltene Frequenz.

  Man nennt den in einem solchen Fall auftretenden Effekt den *aliasing effect*. Ich zitiere hier den entsprechenden Absatz aus meinem Numerik-Skriptum (S. 90f):

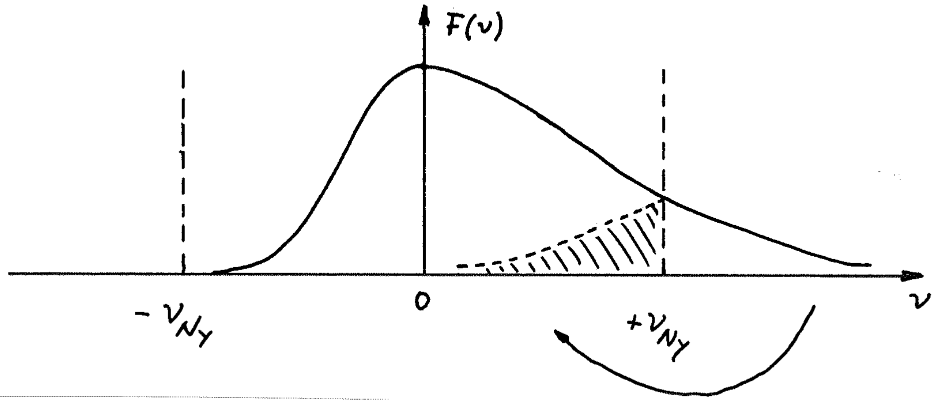


Figure 1: Der *aliasing effect* für Cosinus-Amplituden (schematisch).

Aliasing effect

Wenn das zu analysierende Signal eine Frequenz-Komponente $\nu_0 = q_0/P$ enthält, welche außerhalb des Bereichs (7) liegt, kommt es zu einem störenden Effekt, der darin besteht, daß die entsprechende Amplitude mit dem Index q_0 *in den Nyquist-Bereich hineingespiegelt wird*, d.h. es wird eine Frequenz-Komponente

$$\nu_0' = \frac{n - q_0}{P} \tag{11}$$

angezeigt, die tatsächlich nicht existiert. Dieses "Hineinspiegeln" findet bei Cosinus-Amplituden *unter Beibehaltung des Vorzeichens* statt, bei Sinus-Amplituden wird dabei *das Vorzeichen geändert.* Die Wirkung des *aliasing effect* für Cosinus-Amplituden wird in Abb. 1 schematisch dargestellt.

- Wiederholen Sie die Spektralanalyse der Signalfunktion (10) mit 64 Stützpunkten, aber mit der (geringfügig) falschen Periode

  $$P = 1.01 \quad s\,.$$

  Diskutieren Sie an Hand der Lösungstabelle, des Linienspektrums und der Interpolationskurve, welche Effekte dabei auftreten.

- Bei der nun zu untersuchendenden Sägezahnfunktion (s. Abb. 2) hat die Signalfunktion grundsätzlch andere Eigenschaften als Funktionen vom Typ (10): während diese aus einer endlichen Zahl von Cosinus-Sinus-Termen bestehen, braucht man zur perfekten Beschreibung der Sägezahnfunktion (im Prinzip) unendlich viele Cosinus- und Sinus-Beiträge. Das bedeutet natürlich auch, daß man keine Chance hat, dem *aliasis effect* zu entkommen; man kann nur die Nyquist-Frequenz so weit erhöhen, daß die danach kommenden Fourierkoeffizienten schon so weit abgeklungen sind, daß sie das numerische Ergebnis nur mehr gering beeinflussen.

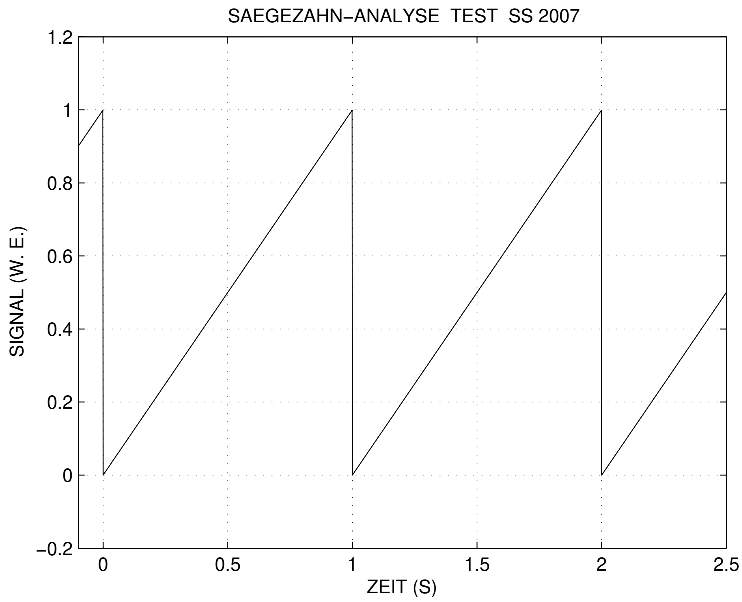


Figure 2: Die Sägezahnfunktion mit der Periode P=1 s.

Führen Sie bitte die folgenden Tests durch (jeweils mit Wertetabelle, Linienspektrum und Darstellung der Interpolationsfunktion):

– 64 Stützpunkte in der Periode P=1 s.

– 128 Stützpunkte über 2 Zähne, d.h. P=2 s.

Es ist natürlich zu erwarten, daß diese beiden Spektralanalysen dasselbe Ergebnis liefern, denn die Stützpunktdichte (*sampling rate*) ist in beiden Fällen gleich, und der zweite Sägezahn enthält genau dieselbe Information wie der erste. Allerdings ist durch die Verdoppelung der Periode die Frequenzauflösung ($\Delta\nu$=0.5 Hz) wesentlich besser als beim ersten Test ($\Delta\nu$= 1 Hz). Diskutieren Sie die Zusammenhänge.

- Erhöhen Sie nun (wieder mit der Periode P=1 s) die Stützpunktzahl von n= 64 auf n= 128 und 256, und zeigen Sie die Sinus-Amplituden für alle drei Werte von n als Funktion der Frequenz in <u>einem</u> Diagramm.

  Diskutieren Sie, wie sich der *aliasing effect* in diesem Diagramm bemerkbar macht.

2. Faltung von Funktionen im Zeit- und Frequenzraum

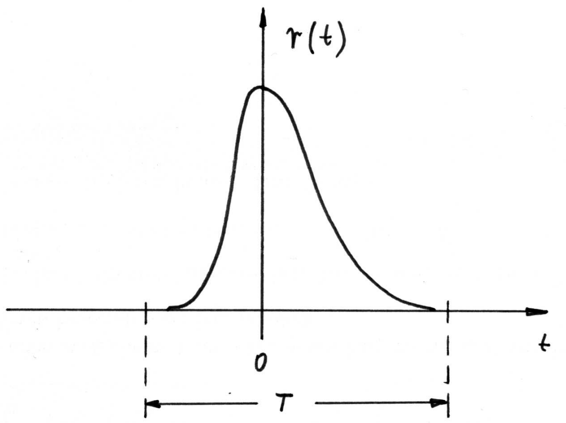
Wir wollen uns im folgenden etwas grundsätzlicher mit dem Thema *Faltung zweier Funktionen* beschäftigen als dies in der Numerik-Grundvorlesung möglich war, und wir beginnen mit der mathematischen Definition der Faltung (der Faltungs*integrals*) der Funktionen $r(t)$ und $s(t)$:

$$m(t) = s * r \equiv \int_{-\infty}^{+\infty} dt'\, s(t')\, r(t-t')\,. \tag{12}$$

Für die nun folgenden Ausführungen werden einige Annahmen bzgl. der beteiligten Funktionen getroffen:

1. die *Signalfunktion* $s(t)$ sei eine periodische Funktion mit der Periode T, d.h.
$$s(t + jT) = s(t) \qquad \text{für} \quad j = -\infty, \ldots, \infty\,. \tag{13}$$

2. die Funktion $r(t)$ (*resolution function, Apparatefunktion, Filterfunktion*) soll innerhalb der Periode von $s(t)$ nach beiden Seiten auf Null abfallen:



Wertet man nun das Faltungsintegral (12) unter den gegebenen Annahmen aus, so ergibt sich

$$m(t) = \sum_{\nu=-\infty}^{+\infty} \int_{\nu T}^{(\nu+1)T} dt'\, s(t')\, r(t-t') = \sum_{\nu=-\infty}^{+\infty} \int_{0}^{T} d\tau\, s(\tau)\, r(t-\nu T-\tau)\,, \tag{14}$$

wobei der letzte Rechenschritt die Periodizität von $s(t)$ ausnutzt.

In der folgenden **Aufgabe** soll nun ein solches Faltungsintegral auf drei verschiedene Arten berechnet werden, nämlich

- mittels einer "direkten" analytischen Auswertung der Formel (14),
- mittels einer numerischen Integration der Formel (14),
- mittels einer diskreten *fast*-Fourier-Transformation (DFFT).

Diese Aufgabe soll für die folgenden Angaben ausgeführt werden:

$$\text{Periode} \quad T = 1.0 \quad s$$

$$s(t) = \frac{t}{2}\,. \tag{15}$$

Periodisch nach links und rechts fortgesetzt, stellt die obige Funktion eine einfache "Sägezahnfunktion" dar. Die Funktion $r(t)$ ist eine auf Eins flächennormierte Gaussfunktion mit der Standardabweichung σ, wobei dieses σ eine variable Inputgröße Ihres Programms sein soll:

$$r(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right)\,. \tag{16}$$

Direkte analytische Auswertung des Faltungsintegrals:

Setzt man die konkreten Funktionen (15) und (16) in das Faltungsintegral (14) ein, so ergibt sich nach einiger Rechenarbeit[3]:

$$\begin{aligned} m(t) = \frac{1}{2}\sum_{\nu=-\infty}^{+\infty} \Bigg\{ & \frac{t-\nu T}{2}\left[\mathrm{erf}\left(\frac{t-\nu T}{\sigma\sqrt{2}}\right) - \mathrm{erf}\left(\frac{t-(\nu+1)T}{\sigma\sqrt{2}}\right)\right] + \\ & + \frac{\sigma}{\sqrt{2\pi}}\left[\exp\left(-\left(\frac{t-\nu T}{\sigma\sqrt{2}}\right)^2\right) - \exp\left(-\left(\frac{t-(\nu+1)T}{\sigma\sqrt{2}}\right)^2\right)\right]\Bigg\} \end{aligned} \tag{17}$$

Anmerkung: diese Lehrveranstaltung ist nicht dazu da, Ihre Fertigkeiten zur Berechnung bestimmter Integrale zu testen. Im Falle, daß Sie dennoch den Ehrgeiz haben, dieses Integral selbständig zu rechnen, sollten Sie die Definition der Gauss'schen Fehlerfunktion kennen, nämlich

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z dt\, \mathrm{e}^{-t^2}\,.$$

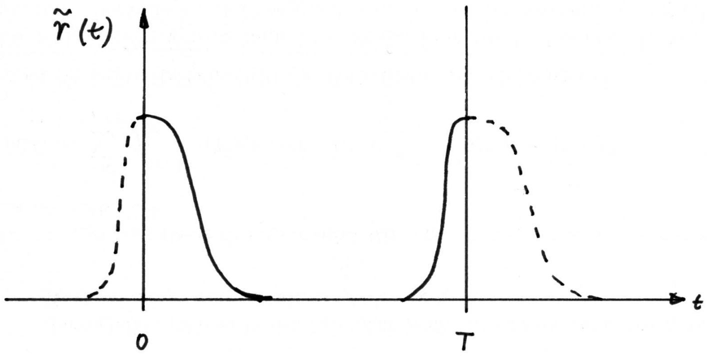
Numerische Auswertung des Faltungsintegrals mittels der Trapezformel:

Wie bereits im Numerik-Vorlesungsskriptum, Abschnitt 3.4.1, erläutert, kann man das Faltungsintegral (14) auch durch Anwendung der Trapezformel näherungsweise lösen. Begnügt man sich bei der Auswertung von $m(t)$ mit dem Zeitbereich $0 \le t \le T$ (was wegen der Periodizität von $m(t)$ natürlich völlig ausreicht), so benötigt man nur die Terme $\nu = 0$ und $\nu = 1$ von Glg. (14), nämlich

$$m(t) = \int_0^T d\tau\, s(\tau)\,\left[r(t-\tau) + r(t-\tau-T)\right] = \int_0^T d\tau\, s(\tau)\,\tilde{r}(t-\tau)\,. \tag{18}$$

[3] Die in dieser Gleichung vorkommende Gauss'sche Fehlerfunktion $erf(z)$ kann im Fall von C und Matlab einfach unter dem Funktionsnamen "erf" aufgerufen werden; im Fall von Fortran finden Sie auf der Internet-Seite dieser LV. den Programmfile erfdp.f90.

Die oben definierte Funktion $\tilde{r}(t)$ wird in der Literatur oft als *function* $r(t)$ *in wrap-around order* bezeichnet: wie sie sehen, enthält diese Darstellung die gesamte Information über $r(t)$ innerhalb des Grundintervalls $[0, T]$:



Definiert man jetzt in $[0, T]$ die n äquidistanten Stützpunkte

$$t_j = j\Delta \quad \text{mit} \quad \Delta = \frac{T}{n} \quad \text{und} \quad j = 0, \ldots, n-1$$

so ergibt sich

$$m(t) \approx \Delta \sum_{j=0}^{n-1} s(t_j)\, \tilde{r}(t - t_j)\,.$$

Wertet man die gefaltete Funktion nur an diesen Stützpunkten aus (d.h. für die Zeiten $t_\ell = \ell\Delta$), ergibt sich

$$m(t_\ell) = \Delta \sum_{j=0}^{n-1} s(t_j)\, \tilde{r}(t_\ell - t_j)$$

bzw., wenn man $m(t_\ell) \equiv m_\ell$ usw. schreibt:

$$m_\ell \approx \Delta \sum_{j=0}^{n-1} s_j\, \tilde{r}_{\ell-j}\,. \tag{19}$$

Die in dieser Summe auftretenden negativen Indizes können durch die folgende Periodizitätsbedingung vermieden werden:

$$\tilde{r}_{-j} = \tilde{r}_{n-j}\,. \tag{20}$$

Numerische Auswertung des Faltungsintegrals mittels der DFFT:

Wenn Sie die Faltung der Funktionen s und r im Zeitraum vornehmen wollen, können Sie dies also durch analytische Auswertung des Faltungsintegrals [Glg. (17)] oder durch die Anwendung der Summenformel [Glg. (19)] tun.

Eine weitere, sehr effiziente Möglichkeit bietet, wie Sie ebenfalls bereits aus der Grundvorlesung wissen, die Transformation von Glg. (19) aus dem Zeitraum in den Frequenzraum: setzt man nämlich für die drei in Glg. (19)

vorkommenden Größen m, r und s die entsprechenden Fourierreihen gemäß Numerik-Skriptum (3.16) ein, so ergibt sich

$$
\begin{aligned}
\frac{1}{n}\sum_{k=0}^{n-1} M_k\,\mathrm{e}^{-i2\pi k\ell/n} &\approx \Delta \sum_{j=0}^{n-1} \frac{1}{n}\sum_{k=0}^{n-1} R_k\,\mathrm{e}^{-i2\pi k(\ell-j)/n}\,\frac{1}{n}\sum_{k'=0}^{n-1} S_{k'}\,\mathrm{e}^{-i2\pi k' j/n} \\
&= \frac{\Delta}{n^2}\sum_{k=0}^{n-1}\sum_{k'=0}^{n-1} R_k\,S_{k'}\,\mathrm{e}^{-i2\pi k\ell/n}\underbrace{\sum_{j=0}^{n-1}\mathrm{e}^{i2\pi(k-k')/n}}_{n\,\delta_{k,k'}} \\
&= \frac{\Delta}{n}\sum_{k=0}^{n-1} R_k\,S_k\,\mathrm{e}^{-i2\pi k\ell/n}\,.
\end{aligned}
$$

Daraus folgt sofort die sehr wichtige Beziehung

$$M_k = \Delta\, S_k\, R_k\,. \tag{21}$$

Die Rücktransformation der so berechneten Fourierkoeffizienten M_k gibt Ihnen sofort die gefaltete Funktion m im Zeitraum.

Aufgabenstellung

- Entwickeln Sie ein Programm, mit dem man die Faltungskurve der Funktionen $s(t)$ gemäß Glg. (15) und $r(t)$ gemäß Glg. (16) berechnen kann, und zwar unter Verwendung der drei oben beschriebenen mathematischen Methoden.
- Verwenden Sie für die Standardabweichung der Gaussfunktion $r(t)$ den Wert $\sigma = 0.15$ s. Dies ist ungefähr der größte σ-Wert, bei dem die Gaussfunktion innerhalb der Periode P=1 s noch rasch genug abfällt.
- Präsentieren Sie Ihre Ergebnisse in einem Diagramm, welches die folgenden Kurven enthält:

  1. die "ungestörte" Signalfunktion $s(t)$,
  2. die analytisch erhaltene Faltungskurve gemäß Glg. (17),
  3. die numerisch erhaltene Faltungskurve nach der Trapezformel (19),
  4. die numerisch erhaltene Faltungskurve mittels der DFFT gemäß Glg. (21).

- Für die Auswertung mit Trapezformel oder DFFT müssen Sie die Funktionen $s(t)$ und $\tilde{r}(t)$ innerhalb der Periode T für n äquidistante Stützpunkte auswerten, wobei Sie wegen der Verwendung der *fast Fourier method* an Punktzahlen gebunden sind, die der Formel

  $$n = 2^m \qquad \text{mit} \qquad m = 2, 3, 4, ...$$

  genügen.
  Wie sehen Ihre Ergebnisse aus, wenn Sie Ihr Programm mehrmals

laufen lassen, und zwar mit steigender Anzahl der Stützpunkte, wie z.B. mit

$$n = 16, 32, 64, 128 \qquad ?$$

Diskutieren Sie diese Ergebnisse.

- Zeigen Sie auch, wie die gefaltete Funktion $m(t)$ der Signalfunktion $s(t)$ immer ähnlicher wird, je kleiner die Standardabweichung σ der *resolution function* $r(t)$ ist.
  Verringern Sie für diesen Test σ auf 0.10 und 0.05 ...

Zero Padding

In meinem Vorlesungsskriptum steht auf der Seite 73 die Fußnote: *Es spielt dabei* (d.h. für die Anwendung der DFFT) *keine Rolle, ob die zu interpolierenden Funktionswerte tatsächlich eine periodische Funktion beschreiben.* Diese Aussage ist beim gegenwärtig behandelten Problem, nämlich der Faltung von Signalfunktionen, nicht mehr gültig!
Die Faltung bewirkt ja offenbar, daß die Signalfunktion $s(t)$ für jedes Argument t auch Beiträge rechts und links von diesem Zeitpunkt aufsummiert. Wenn also die Signalfunktion realiter periodisch ist, werden am Beginn und Ende jeder Periode auch Einflüsse der vorhergehenden bzw. der nachfolgenden Periode wirksam. Dieser Effekt fällt natürlich bei einer nichtperiodischen Signalfunktion weg!

Am einfachsten können Sie sich dieses Problem klar machen, wenn Sie die analytische Lösung der Faltungsfunktion gemäß Glg. (17) einmal unter der Voraussetzung auswerten, daß die "Sägezahnfunktion" $s(t)$ periodisch mit T ist, und ein zweites Mal unter der Voraussetzung, daß $s(t)$ nur durch einen Sägezahn im Intervall $[0, T]$ repräsentiert ist: zu diesem Zweck brauchen Sie nur die Summe über j zu trivialisieren:

$$\sum_{j=-\infty}^{+\infty} \quad \text{(periodisch)} \quad \rightarrow \quad \sum_{j=0}^{0} \quad \text{(nicht periodisch.)}$$

Das Ergebnis eines solchen Tests sieht folgendermaßen aus:

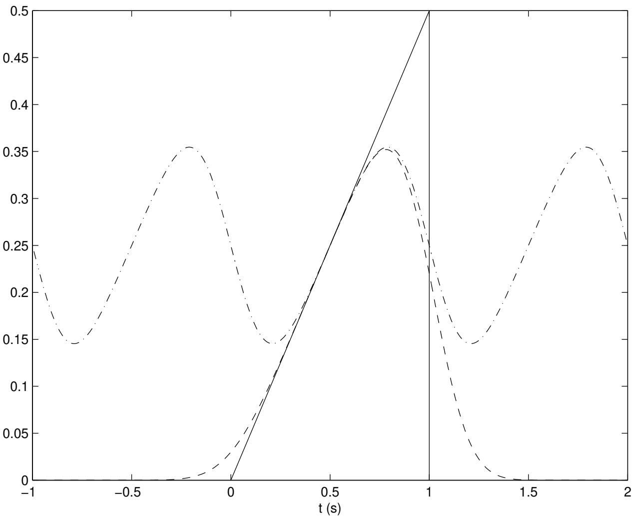


Abb.: Faltung einer nicht-periodischen Signalfunktion (am Beispiel eines

"Sägezahns"). Ausgezogene Kurve: Signalfunktion, strichlierte Kurve: gefaltete Funktion. Zum Vergleich ist auch die gefaltete Funktion einer periodischen Sägezahnkurve dargestellt (strich-punktierte Kurve).

Was kann man nun tun, um auch im nicht-periodischen Fall mittels der Trapezformel bzw. der DFFT die richtige Faltungskurve zu erhalten?

Eine Möglichkeit stellt die in der Literatur beschriebene Methode des *zero padding* dar (Engl. *to pad* = auffüllen). Dabei wird die Periode T auf der rechten Seite erweitert, und die äquidistanten Stützpunkte in diesem Bereich werden mit Nullen aufgefüllt. Die Breite ℓ dieser *padding region* muß mindestens so groß sein, daß die *resolution function* $r(t)$ in diesem Bereich bequem auf Null abklingen kann:

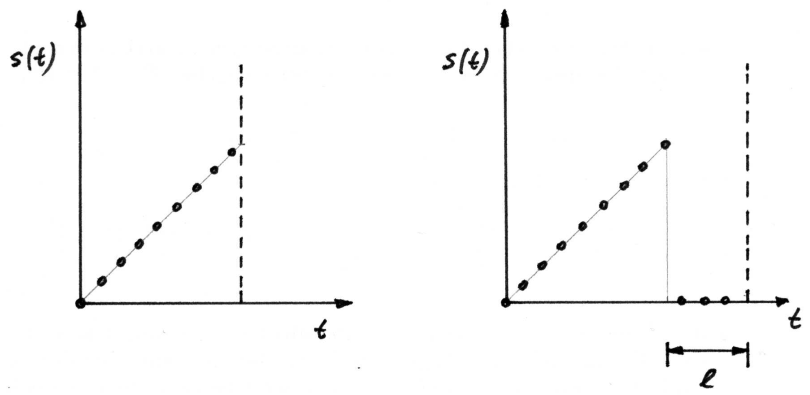


Abb.: Am Beispiel der Sägezahnkurve (schematisch):
Links: ohne *zero padding*, rechts: mit *zero padding* der Breite ℓ.
Die Periode T der Signalfunktion wird durch die strichlierte Linie markiert.

Wenn Sie die Berechnung einer Faltung mittels der Trapezformel (19) bzw. mittels der DFFT ausgehend von den Stützpunkten der Signalfunktion inklusive der *padding points* durchführen, erhalten Sie im Prinzip das korrekte Ergebnis: die berechnete Faltungskurve ist zwar wieder periodisch, durch das *padding* liegen aber zwischen den "Faltungs-Peaks" Bereiche, in denen sie auf Null zurückgeht.

Aufgabenstellung

- Entwickeln Sie eine Version Ihres Faltungsprogramms, mit der Sie unter Verwendung des *zero paddings* die Faltung nicht-periodischer Signalfunktionen durchführen können.
- Nehmen Sie als konkrete Funktionen für $s(t)$ und für $r(t)$ wieder die Sägezahnfunktion (15) und die Gaussfunktion (16) mit der Standardabweichung $\sigma = 0.15$.
- Die Berechnung der Faltungsfunktion soll wieder (1) analytisch, (2) mittels Trapezformel, (3) mittels DFFT durchgeführt werden.
- Stellen Sie die drei Ergebnisfunktionen in einem Diagramm zusammen.

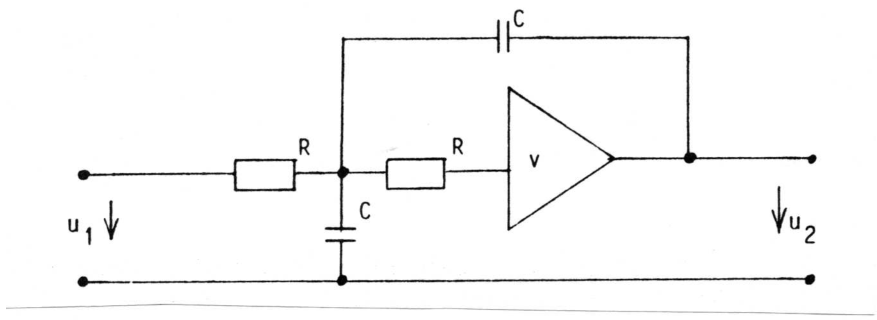
- Was das *zero padding* betrifft: experimentieren Sie etwas, bis Sie die richtige Breite ℓ der *padding region* (s. obige Abbildung) gefunden haben.

- Wie Sie an den Trapezregel- und DFFT-Ergebnissen sehen werden, erhalten Sie die an sich richtige Faltungskurve in einem falschen Zeitbereich, d.h. der Bereich vor $t = 0$ fehlt und taucht auf der rechten Seite wieder auf! Da Sie wissen, daß die numerischen Ergebnisse periodisch sind, sollte es Ihnen nicht schwerfallen, unter Ausnutzung dieser Tatsache die "richtige" Faltungskurve aus den gegebenen Datenpunkten zusammenzustellen.

3. Filterung eines gemessenen Signals mittels DFFT

Quelle: P. Jacobs und S. Jancar, *BASIC: Interpolationen–Approximationen–Splines-FFT*, ed. H. Kohler, Vieweg-Verlag.

Ausgangspunkt des Übungsbeispiels sind Messungen an einem sogenannten "aktiven Tiefpaß:



Eine derartige Schaltung verändert das Eingangssignal $u_1(t)$ in das Ausgangssignal $u_2(t)$, wobei die Beziehung zwischen u_1 und u_2 durch das folgende Faltungsintegral gegeben ist:

$$u_2(t) = \int_{-\infty}^{+\infty} dt'\, f(t-t')\, u_1(t') \tag{22}$$

Die Funktion $f(t)$ stellt dabei die im Numerik-Skriptum als "Apparatefunktion" bezeichnete Größe dar. Im konkreten Fall zieht man für $f(t)$ die Bezeichnungen *Übertragungsfunktion* bzw. *Filterfunktion* vor.

Das Problem der folgenden Aufgabe besteht nun darin, aus der *gemessenen* Ausgangsfunktion $u_2(t)$ auf das unbekannte *Eingangssignal* $u_1(t)$ zurückzuschließen. Dies soll mit Hilfe der *diskreten Fast Fourier Transform* (DFFT) geschehen.

Dazu bedarf es natürlich der Kenntnis der Filterfunktion $f(t)$; eine Messung dieser Funktion kann z. B. so erfolgen, daß man dem System als Eingangssignal eine möglichst "delta-förmige" Funktion anbietet. Es folgt nämlich aus (1)

$$u_1(t) = \delta(t) \quad \rightarrow \quad u_2(t) = \int_{t'=-\infty}^{+\infty} dt'\, f(t-t')\, \delta(t') = f(t)\,.$$

Konkrete Filterfunktionen $f(t)$ stehen in tabellarischer Form auf den Files

`filter_a.dat` und `filter_c.dat`

zum Downladen bereit.

Wenn Sie die obigen Daten Fourier-transformieren, sehen Sie sofort, daß diese Filter vor allem die höheren Frequenzen unterdrücken; deshalb der in der Elektrotechnik bekannte Name "aktiver Tiefpaß".

Natürlich wird es auch hier das beste sein, das Faltungsintegral (22) aus dem Zeitraum in den Frequenzraum überzuführen. Nennt man, um den Anschluß an bereits diskutierte Formeln zu finden, die Fourierkoeffizienten der Funktionen $u_2(t)$, $f(t)$ und $u_1(t)$ M_k, R_k und S_k, so ergibt sich wieder die Glg. (21), deren Bedeutung darin liegt, daß man sie ohne Probleme nach der "ungestörten" Signalfunktion auflösen kann:

$$S_k = \frac{1}{\Delta} \frac{M_k}{R_k}\,. \tag{23}$$

Kennt man also die gemessene Funktion $u_1(t)$ und die Apparatefunktion $f(t)$ im realen Zeitraum, so kann man wie folgt vorgehen:

1. FT von $u_2(t) \quad \rightarrow \quad M_k$
2. FT von $f(t) \quad \rightarrow \quad R_k$
3. Berechnung von S_k aus Glg. (23)
4. Inverse FT von $S_k \quad \rightarrow \quad u_1(t)$

Aufgabenstellung

Es ist ein Programm zu entwickeln, mit Hilfe dessen man aus tabellarisch gegebenen Ausgangs- und Filterfunktionen [$u_2(t)$ und $f(t)$] die Eingangsfunktion $u_1(t)$ rekonstruieren kann. Es ist dafür die Methode der *Fast Fourier Transform* anzuwenden.

Matlab-User können die *matlab*-Funktionen *fft* und *ifft* verwenden.

Für C-User steht die Routine *four1.c* zum Downladen bereit. Eine Beschreibung dieser Routine finden Sie im Vorlesungs-Skriptum, Abschnitt 3.3.2.

Allgemeine Parameter: Anzahl n der Stützpunkte $= 64$
zeitliche Periode $P = 0.00419$ s

Die folgende Tabelle gibt Ihnen die Kombinationen von Meßdaten und Filtern, aus denen Sie jeweils die entsprechenden Eingangssignale ermitteln sollen (alle genannten Files stehen zum Downladen bereit):

```
Ausgangssignal u2(t)                  Filter f(t)
===============================================================

  messung4.dat                        filter_a.dat

  messung5.dat                        filter_c.dat

  messung6.dat                        filter_a.dat

  messung1.dat                        filter_a.dat

  messung3.dat                        filter_a.dat

===============================================================
```

Bitte stellen Sie alle gemessenen Funktionen und die zugehörigen rekonstruierten Eingangssignal-Funktionen im Zeitraum grafisch dar.

4. Einfluß statistischer Fehler (noise) auf die Rekonstruktion einer Signalfunktion

In den bisherigen Aufgaben zu dieser Übung haben wir uns ausschließlich mit Meßkurven $m(t)$ beschäftigt, welche keinerlei statistische Fehler enthielten. Solche Meßdaten ohne "Verrauschung" (*noise*) kommen natürlich in der experimentellen Praxis nicht vor. D. h., die Signal-Daten, mit denen man es i.a. zu tun hat, sind *in zweifacher Hinsicht* beeinträchtigt, nämlich

1. durch die Filterwirkung der Meßapparatur, und
2. durch die beim Experiment unvermeidlich auftretenden statistischen Fehler.

In dieser und den folgenden Aufgaben geht es nun darum, den Einfluß der statistischen Fehler (der *noise*) auf die Rekonstruktion der Signalfunktion zu studieren und gegebenenfalls durch geeignete Methoden zu eliminieren.

Bei einem qualitätsvollen Experiment kann man davon ausgehen, daß die Beeinflussung der Meßwerte durch statistische Schwankungen relativ klein sein wird.

In solchen Fällen könnte man versucht sein, die (geringe) Verrauschung der Meßwerte bei der Rekonstruktion der Signalkurve einfach zu ignorieren.

Im Rahmen der folgenden Aufgabe sollen Sie untersuchen, ob eine solche Vorgangsweise vernünftig ist oder nicht!

Zu diesem Zweck soll eine tabellarisch gegebene, glatte Meßfunktion $m(t)$ von Ihnen künstlich verrauscht werden, und zwar durch Addition einer *noise*-Funktion $n(t)$:

$$\tilde{m}(t_j) = m(t_j) + n(t_j)\,, \tag{24}$$

wobei $n_j \equiv n(t_j)$ *normalverteilte Abweichungen mit einer vorgegebenen Standardabweichung* σ darstellen, d.h., die Werte n_j gehorchen der Verteilungsfunktion

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}}\,\exp\left(-\frac{n^2}{2\sigma^2}\right)\,. \tag{25}$$

Wie kann nun eine normalverteilte Verrauschung programmiert werden?

Für Matlab gibt es dazu die Funktion *randn*, die eine solche Simulation äußerst einfach ermöglicht. Die Syntax lautet:

```
y_plus_noise = y + sigma*randn(ndata,1)

% Dabei bedeuten:

y               Vektor des nicht-verrauschten Signals
sigma           Standard-Abweichung der Normalverteilung
```

```
ndata          Laenge des Vektors  y

y_plus_noise   Vektor mit dem verrauschten Signal

Anmerkung: der zweite Parameter in  randn, "1", bedeutet, dass es sich
           bei  y  bzw.  y_plus_noise  um EINdimensionale Matrizen,
   d.h. um Vektoren handelt.
```

Weitere Informationen über *randn* (und verwandte Funktionen) können Sie bei Matlab unter "help randn" abrufen.

Für C- und Fortran-UserInnen finden sich in verschiedenen Programmbibliotheken sicher Routinen, die ähnliches leisten wie das oben vorgestellte Matlab-Programm. Ich möchte hier aber die Gelegenheit nutzen, Ihnen einen einfachen, "selbstgestrickten" Algorithmus vorzustellen, mit dem man eine gegebene Punktmenge y_i mit einer simulierten Verrauschung n_i (Abweichungen normalverteilt mit Standardabweichung σ) versehen kann.

Ausgangspunkt ist die Gauss'sche Normalverteilung als *differentielle Verteilungsfunktion*:

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{n^2}{2\sigma^2}\right) . \tag{26}$$

Die daraus resultierende *integrale Verteilungsfunktion* lautet

$$P(n) \equiv \int_{-\infty}^{n} dx\, p(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{n}{\sigma\sqrt{2}}\right)\right] , \tag{27}$$

wobei die *error function* wie auf Seite 12 definiert ist.

Für die Berechnung der normalverteilten Abweichungen kann nun die sog. *Inversionsmethode* verwendet werden (s. jedes Statistik-Lehrbuch oder z.B. mein Skriptum *Computer-Simulation*), welche auf der Auswertung der Beziehung

$$P(n) = \kappa \tag{28}$$

beruht, wobei κ eine in $(0,1)$ gleichverteilte Zufallszahl ist. Berechnet man die Lösung von Glg. (28) für die Abweichung n_i des i-ten y-Wertes mittels einer Newton-Iteration, so ergibt sich der folgende Algorithmus:

$$n_i^{(1)} = 0$$

$$n_i^{(j+1)} = n_i^{(j)} - \frac{a_j}{b_j} \qquad (j = 1, 2, \ldots)$$

mit

$$a_j = 1 + \mathrm{erf}\left(\frac{n_i^{(j)}}{\sigma\sqrt{2}}\right) - 2\kappa$$

und

$$b_j = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{n_i^{(j)2}}{2\sigma^2}\right) .$$

Nach Ausführung dieser Iteration bis auf die gewünschte Genauigkeit ergibt sich der verrauschte y-Wert zu

$$\tilde{y}_i = y_i + n_i \,. \tag{29}$$

Ein C-Programm randn.c bzw. ein entsprechendes Fortran-Programm randn.f90, welches auf diesen Formeln beruht, finden Sie auf der Website dieser Lehrveranstaltung.

Aufgabenstellung

Rekonstruieren Sie die Signalfunktion $s(t)$ unter Verwendung der als bekannt vorausgesetzten Filterfunktion $r(t)$ durch Entfaltung sowohl der *unverrauschten* Meßfunktion $m(t)$ als auch der *verrauschten* Meßfunktion $\tilde{m}(t)$.

- Verwenden Sie für die ersten Tests die Datenfiles

```
    messung6.dat      % Faltung einer Saegezahnfunktion mit
                      % einem Chebyshev-Filter

    filter_a.dat      % Chebyshev-Filter

Fuer beide Datenfiles gilt:  64 Stuetzpunkte, Periode P = 0.00419 s
```

  Für die numerische Simulation der normalverteilten Abweichungen verwenden Sie nacheinander die folgenden Werte:

```
sigma =  0.00001 / 0.0001 / 0.0005 / 0.001 / 0.005
```

- Stellen Sie nun unter Verwendung der folgenden, zum *downladen* bereitstehenden Daten fest, ob sich die erhaltenen Rekonstruktionsergebnisse bei einer Erhöhung der Stützpunktzahl verbessern oder verschlechtern:

```
    messung6_1.dat      % Faltung einer Saegezahnfunktion mit
                        % einem Chebyshev-Filter

    filter_a_1.dat      % Chebyshev-Filter

Fuer beide Datenfiles gilt:  256 Stuetzpunkte, Periode P = 0.00419 s
```

- **Stellen Sie die Unterschiede zwischen $m(t)$ und $\tilde{m}(t)$ sowie die entsprechenden rekonstruierten Signalfunktionen $s(t)$ und $\tilde{s}(t)$ in einer Reihe von Grafiken dar.**

5. Reduktion des noise mittels der Methode von Dirichlet

Die letzte Aufgabe hat Ihnen deutlich gezeigt, daß ein simples Ignorieren von statistischen Fehlern in den Meßdaten völlig unmöglich ist, da bereits sehr kleine Fehler, die im Zeitraum kaum ins Gewicht fallen, die Rekonstruktion der Signalfunktion aus dem Frequenzraum heraus komplett verderben.

Es muss daher bereits im Frequenzraum versucht werden, den in der Meßfunktion enthaltenen *noise* effizient zu unterdrücken, d.h. eine *Glättung* der Meßdaten vorzunehmen. Einige sehr einfache Methoden dazu finden Sie im Vorlesungsskriptum "Numerische Methoden ...", darunter die *Methode von Dirichlet* (s. Abschnitt 3.4.2).

Die Dirichlet-Methode geht von der Idee aus, daß die statistischen Schwankungen in den Werten von $m(t)$ im Fourierraum, also in der Funktion $M(k)$, sich hauptsächlich im mittleren k-Bereich bemerkbar machen, also in den Fourierkoeffienten mit den Indizes

```
istart bis n-istart,
```

wohingegen die $M(k)$ für

```
0 bis istart-1    sowie  n-istart+1 bis n-1
```

im wesentlichen die glatte Messfunktion beschreiben. Eine wirksame Art und Weise, den *noise* zu unterdrücken, besteht also darin,

```
M(j) = 0   fuer  j=istart, ..., n-istart
```

zu setzen.

Im folgenden Test sollen Sie sich nun davon überzeugen, ob und in welchem Ausmaß diese Dirichlet'sche Glättung funktioniert.

Aufgabenstellung

- Verwenden Sie noch einmal das Datenmaterial

```
messung6_1.dat      % Faltung einer Saegezahnfunktion mit
                    % einem Chebyshev-Filter

filter_a_1.dat      % Chebyshev-Filter

Fuer beide Datenfiles gilt:  256 Stuetzpunkte, Periode P = 0.00419 s
```

- Unterwerfen Sie die gegebenen Meßwerte einer künstlichen Verrauschung in Form von normalverteilten Abweichungen mit $\sigma = 0.0005$.
- Machen Sie eine diskrete FT der verrauschten Meßdaten und behandeln Sie die erhaltenen Fourierkoeffizienten nach der Methode von Dirichlet.
- Verwenden Sie diese *Dirichlet-bereinigten* Fourierkoeffizienten für eine Rekonstruktion der Signalfunktion $s(t)$.

- Stellen Sie die folgenden Ergebnisse grafisch dar:

  (1) Rekonstruktion der glatten, d.h. von statistischen Fehlern freien Signalfunktion,
  (2) Rekonstruktion der fehlerbehafteten Signalfunktion ohne Glättung,
  (3) Rekonstruktion der fehlerbehafteten Signalfunktion nach erfolgter Dirichlet-Glättung.

- Ein wichtiger Punkt bei der Anwendung der Dirichlet-Methode ist natürlich die richtige Wahl des Index *istart*; experimentieren Sie etwas mit diesem Parameter:
  Wie verändert sich die Qualität der Rekonstruktion, wenn Sie *istart zu groß oder zu klein wählen?*
  Dokumentieren Sie diese Effekte durch Diagramme.

Zum letzten Punkt noch eine Anmerkung: in der Literatur wird oft die Qualität der Rekonstruktion durch die folgende, einfache *Least-Squares-Formel* quantifiziert:

$$MSE = \frac{1}{n} \sum_{j=0}^{n-1} \left[s_{rekonst}(t_j) - s(t_j) \right]^2 \tag{30}$$

- Versuchen Sie nun unter Verwendung der Dirichlet-Glättung jenes *optimale (?) istart* zu finden, welches die Fehlergröße MSE zu einem Minimum macht.

- Ist die entsprechende rekonstruierte Signalfunktion wirklich auch *optisch* die beste Annäherung an die wirkliche Signalfunktion?

Noise-Reduktion von nicht-periodischen Daten

Bereits beim Thema "Zero Padding" (S. 15f) habe ich von den Problemen gesprochen, die bei der Fourier-Analyse von Datensätzen auftreten, welche physikalisch <u>nicht</u> Teil einer periodischen Punktmenge sind.

Auch bei der *noise*-Reduktion z.B. mittels der Dirichlet-Methode können solche Datenpunkte Probleme bereiten, wie ich im folgenden Beispiel zeigen werde:

Betrachten Sie die per Experiment erhaltenen (verrauschten) Daten in der Abb. 3 a: diese Daten sind keinesfalls ein Teil einer periodischen Gesamtheit; dennoch werden Sie bei der Fourier-Transformation so behandelt (s. Abb. 3 b), was zu numerisch unangenehmen Sprungstellen an den Periodengrenzen führt.

Welche Konsequenzen ergeben sich aus dieser Tatsache, wenn man mittels der Dirichlet-Methode eine Datenglättung dieser Meßpunkte durchführt? Dies zu untersuchen, ist Ihre nächste Aufgabe.

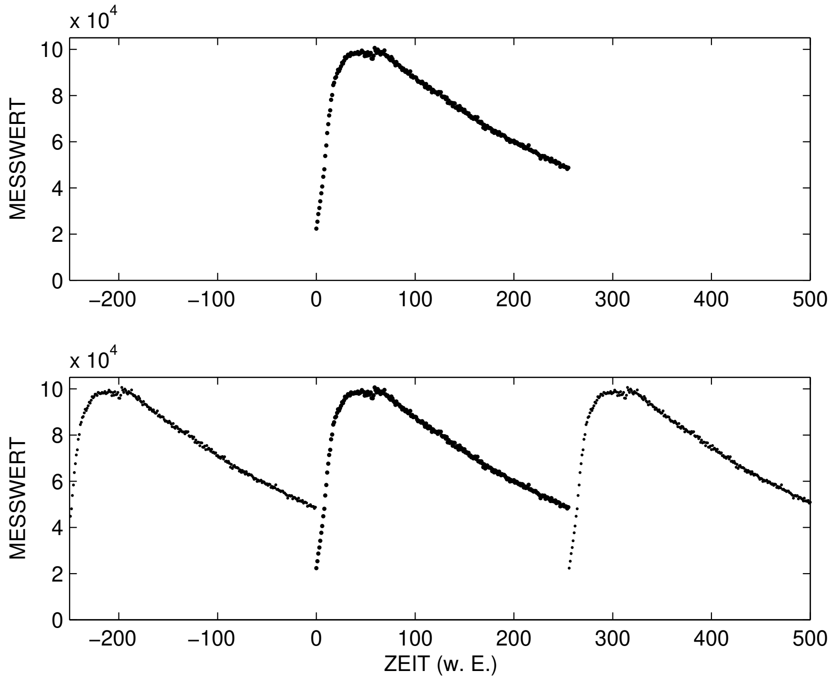


Figure 3: Die tatsächlich gemessenen Punkte (oben) werden bei der Fourier-Transformation als Teil einer periodischen Gesamtheit angesehen (unten).

Aufgabenstellung

- Führen Sie an den Meßdaten

  `nichtperiodisch.dat`

  eine *noise*-Reduktion mittels der Methode von Dirichlet aus.

- Machen Sie dabei eine *istart*-Analyse, indem Sie mit *istart*=80 beginnen und in Schritten von 5 bis zu *istart*=5 zurückgehen. Beobachten Sie dabei, ob und wie die Datenglättung funktioniert (grafische Dokumentation).

- Wie sie feststellen werden, treten Probleme auf, die offensichtlich von den Sprungstellen der Daten an den Periodengrenzen herrühren.

  Diese unerwünschten Effekte können mittels einer Methode wenigstens teilweise unterdrückt werden, die in der Literatur als *remove of linear trend* bezeichnet wird. Diese ganz simple Methode werde ich im Rahmen meiner Übungsstunden erklären.

  Wiederholen Sie Ihre *istart*-Analyse und studieren Sie die Wirkung dieser *remove of linear trend*-Methode auf die Glättungskurven.

Die Problemstellung *noise*-Reduktion betrifft ein großes und hochentwickeltes Teilgebiet der "Diskreten Fourier-Transformation", welches ich in dieser LV. nur in den Grundzügen behandeln kann. Viele für die Praxis wichtige Methoden wie z.B. die "Wiener-Filterung" usw. müssen hier aus Zeitgründen unbesprochen bleiben.

Ich möchte aber hier wenigstens eine sehr einfache und sehr leistungsfähige "Konkurrenz-Methode" zum *Dirichlet*-Verfahren vorstellen, nämlich die *Datenglättung nach Tikhonov.*

In diesem Fall erfolgt die *noise reduction* dadurch, daß die Fourierkoeffizienten der gegebenen Daten in der Form

$$\tilde{Y}_q = \frac{Y_q}{1 + \alpha q^2} \tag{31}$$

verändert werden. Wie Sie sehen, handelt es sich dabei um eine typische *low-pass*-Filterung, d.h., die Fourierkoeffizienten für $|q|$ nahe Null (also für betragsmäßig kleine Frequenzen) bleiben (fast) unverändert, während die F_q für (betragsmäßig) größere Frequenzen - je nach Größe des Parameters α - mehr oder minder stark reduziert werden. Der Unterschied zur Dirichlet-Methode ist evident: während bei dieser die Fourierkoeffizienten "schlagartig" ab einem gewissen Index *istart* Null gesetzt werden, erfolgt diese Koeffizienten-Reduktion nach der Tikhonov-Methode kontinuierlich.

Wie Sie aus der Formel (31) sofort ersehen können, wird hier angenommen, daß die q-Werte *symmetrisch um q=0 angeordnet sind*; man muß also, um die Tikhonov-Formel anwenden zu können, unbedingt die Optimierung (2, 3) anwenden!

Aufgabenstellung

- Verwenden Sie die Tikhonov-Methode für die *noise reduction* der Daten auf dem File

  `nichtperiodisch.dat`

- Verwenden Sie dabei auch die *remove of linear trend*-Methode.

- Experimentieren Sie dabei mit dem Parameter α.

- Vergleichen Sie die "best-geglättete" Kurve nach Tikhonov mit der "best-geglätteten" Kurve nach Dirichlet, und stellen Sie diesen Vergleich grafisch dar.

6. Korrelation zweier Funktionen

Ebenso wie die Faltung ist auch die Korrelation zweier Funktionen durch ein Integral definiert, und zwar in der Form

$$m(t) = p \odot q \equiv \int_{-\infty}^{+\infty} dt' \, p(t') \, q(t + t') \,, \tag{32}$$

wobei angenommen werden soll, daß beide Funktionen p und q reellwertig sind.

Wie im Falle der Faltung (s. S. 5), kann auch hier nach wenigen Rechenschritten gezeigt werden, daß die diskrete Fourier-Transformation der Gleichung (32) zum einfachen Ausdruck

$$M_k = \Delta \, P_k^* \, Q_k \tag{33}$$

führt[4]. Wenn Sie dieses Ergebnis mit dem "Faltungs-Ergebnis" (21) vergleichen, so erkennen Sie als einzigen Unterschied, daß für die erste Funktion der *konjugiert-komplexe* Fourierkoeffizient zu nehmen ist.

Beginnen wir ohne weitere Vorrede sogleich mit der ersten

Aufgabenstellung

zum Thema Korrelation:

- Schreiben Sie ein Programm, mit welchem Sie die Korrelationsfunktion der beiden Funktionen

  $$p(t) = \begin{cases} 0, & t < 0, \\ 1, & 0 < t < 1, \\ 0, & t > 1, \end{cases} \qquad \text{und} \qquad q(t) = \begin{cases} 0, & t < 0, \\ 1 - t, & 0 < t < 1, \\ 0, & t > 1 \end{cases} \tag{34}$$

  sowohl mittels der folgenden Gleichungen (35) bzw. (37) als auch mit Hilfe der DFFT berechnen können.

  Es geht also um die Korrelation der "Quadratfunktion" $p(t)$ und der "Dreiecksfunktion" $q(t)$. Sie können sich durch eine kurze Rechnung davon überzeugen, daß das analytische Ergebnis wie folgt lautet:

  $$m(t) = p \odot q = \begin{cases} 0, & t < -1, \\ \frac{1-t^2}{2}, & -1 \leq t \leq 0, \\ \frac{(1-t)^2}{2}, & 0 < t \leq 1, \\ 0, & t > 1. \end{cases} \tag{35}$$

  An diesem Beispiel kann man auch sofort zeigen, daß - im Unterschied zur Faltung - die Korrelation *von der Reihenfolge der Funktionen* p *und* q ist, daß also gilt:

  $$p \odot q \neq q \odot p \,. \tag{36}$$

[4] Δ bedeutet wieder den Abstand zweier Stützpunkte im t-Raum.

In der Tat ergibt auch die analytische Integration für das konkrete Beispiel das Resultat

$$m(t) = q \odot p = \begin{cases} 0, & t < -1, \\ \frac{(1+t)^2}{2}, & -1 \le t \le 0, \\ \frac{1-t^2}{2}, & 0 < t \le 1, \\ 0, & t > 1. \end{cases} \tag{37}$$

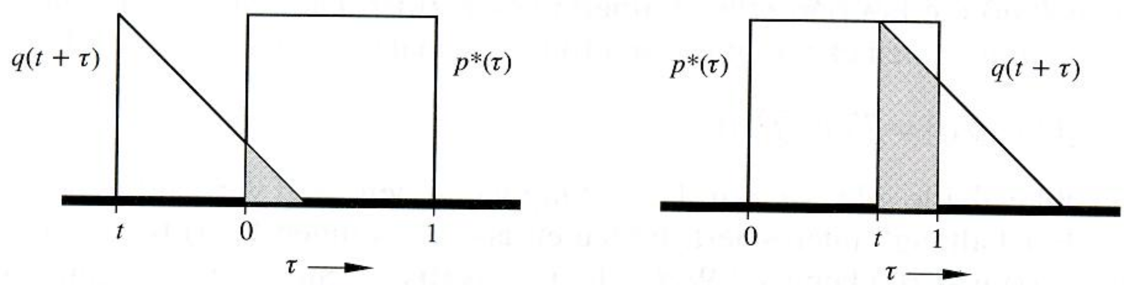


Abb.: Graphische Darstellung einer Korrelation: die "Dreiecksfunktion" q wird um eine Zeit t relativ zur "Quadratfunktion" p verschoben, und das Produkt der beiden Funktionen wird über alle τ integriert, um die Korrelation zum Zeitpunkt t zu erhalten.

Die obige Zeichnung zeigt Ihnen, was Korrelation im Prinzip bedeutet, nämlich die Flächenüberlappung der beiden Funktionen: für $t < -1$ und für $t > 1$ überlappen Dreieck und Quadrat überhaupt nicht, und die Korrelation ist Null; für $t = 0$ liegt das Dreieck komplett innerhalb des Quadrates, und die Korrelation hat einen Maximalwert.

- Vergleichen Sie in Diagrammen die Ergebnisse von $p \odot q$ und $q \odot p$, welche Sie aus den analytischen Formeln und mittels der DFFT erhalten.

Dazu noch einige Hinweise:

1. Für die DFFT müssen Sie natürlich eine Periode T angeben, in welche beide Funktionen gut hineinpassen. Innerhalb dieser Periode müssen Sie die Funktionswerte $p(t_j)$ und $q(t_j)$ für 2^m äquidistante Zeitpunkte t_j berechnen.
2. Experimentieren Sie mit der Periode T: ist ein *zero-padding* erforderlich?
3. Experimentieren Sie mit der Anzahl der Stützpunkte. Wie im Fall der numerischen Berechnung der Faltungsintegrale werden Sie auch hier die Erfahrung machen, daß eine zu geringe Stützpunktzahl unterschiedliche Ergebnisse zwischen dem FFT-Resultat (33) und den analytischen Resultaten (35) und (37) verursacht.

7. Autokorrelation

Wie der Name bereits vermuten läßt, versteht man unter Autokorrelation die Korrelation einer Funktion $p(t)$ "mit sich selbst", also

$$p \odot p \equiv \int_{-\infty}^{+\infty} dt'\, p(t')\, p(t+t')\,, \tag{38}$$

Von besonderer Bedeutung ist die Autokorrelation bei periodischen Funktionen (Periode T), also mit der Eigenschaft

$$p(t+T) = p(t)\,. \tag{39}$$

Hier verschwindet der Integrand in Glg. (38) nicht für $t \to \pm\infty$, und das Korrelations-Integral braucht nicht zu existieren. In einem solchen Fall definiert man die *mittlere Korrelationsfunktion*

$$p \odot p \equiv \frac{1}{T} \int_0^T dt'\, p(t')\, p(t+t')\,. \tag{40}$$

Die folgenden Ausführungen sollen Ihnen die praktische Verwendbarkeit solcher Autokorrelationen demonstrieren, und zwar basierend auf dem einfachen Beispiel

$$p(t) = \sin(\omega t) \qquad \text{mit der Periode} \quad T = 2\pi\,. \tag{41}$$

Die im Argument des Sinus vorkommende Größe ω bedeutet die Zahl der vollständigen Sinus-Schwingungen im Zeitraum $0 \le t \le T$.

Wieder ist es kein Problem, $p \odot p$ analytisch zu berechnen. Man erhält mit geringem Aufwand das Ergebnis

$$p \odot p = \frac{1}{2\pi} \int_0^{2\pi} dt'\; \sin(\omega t')\; \sin(\omega\,[t+t']) = \frac{1}{2} \cos(\omega t)\,. \tag{42}$$

Dieses Resultat ist typisch für die Autokorrelation periodischer Funktionen, die jeweils ein Maximum ausmacht, wenn die Funktion um eine ganze Anzahl von Perioden verschoben wird.

Führt man aber die Berechnung der Autokorrelation im Fourier-Raum durch, so ergibt sich aus Glg. (33) der Ausdruck

$$M_k = \frac{1}{T} \Delta P_k^* P_k = \frac{1}{n} |P_k|^2\,. \tag{43}$$

In der Literatur nennt man ein solches Spektrum von Absolut-Quadraten von Fourier-Koeffizienten ein *power spectrum* (Leistungsspektrum)[5].

[5] In der Literatur zur Fourier-Transformation gibt es den *Wiener-Chintschin'schen Satz*: "Die Fourier-Transformierte der Autokorrelationsfunktion ist das Leistungsspektrum."

Aufgabenstellung

- Schreiben Sie ein Programm, mit welchem man die Autokorrelationsfunktion (AK) einer gegebenen (periodischen) Funktion $p(t) = \sin(\omega t)$ mittels DFFT berechnen kann.
- Nehmen Sie für ω den Wert 10, d.h. genau 10 Schwingungen im Zeitraum von 0 bis $T = 2\pi$.
- Verifizieren Sie in einem Diagramm, daß die AK-Funktion gemäß Glg. (46) eine Cosinusfunktion darstellt.

Von besonderer Bedeutung sind Autokorrelationen im Fall von verrauschten Signalen, also von Signalfunktionen, welche von einem normalverteilten *noise* korrumpiert sind (s. S. 12-14):

$$p(t) = s(t) + n_\sigma(t)\,. \tag{44}$$

In diesem Fall sieht die Autokorrelation wie folgt aus:

$$p \odot p = (s + n_\sigma) \odot (s + n_\sigma) = s \odot s + s \odot n_\sigma + n_\sigma \odot s + n_\sigma \odot n_\sigma\,. \tag{45}$$

Man kann sich nun leicht überlegen, daß die AK einer glatten Funktion $s(t)$ und einer statistischen Abweichungsfunktion $n_\sigma(t)$ zu sehr kleinen Integrationswerten von

$$\frac{1}{T}\int_0^T dt'\, s(t')\, n_\sigma(t+t')$$

führen wird, weil die Funktion n_σ stets zwischen positiven und negativen Werten wechselt. Diese Aussage, die zumindest für den Grenzfall $T \to \infty$ exakt gilt, ist unabhängig von der Reihenfolge der Funktionen:

$$\lim_{P\to\infty} s \odot n_\sigma = \lim_{P\to\infty} n_\sigma \odot s = \lim_{P\to\infty} n_\sigma \odot n_\sigma = 0\,. \tag{46}$$

Aus den Gleichungen (45) und (46) folgt für endliche T

$$p \odot p \approx s \odot s\,. \tag{47}$$

Dieses Ergebnis ist von weitreichender Bedeutung! Es bedeutet nämlich, daß die Autokorrelation eines statistisch verrauschten Signals $p(t)$ *die Periodizität des ungestörten Signals $s(t)$ deutlich wiedergibt, auch wenn das Signal-Noise-Verhältnis ungünstig ist.*

Aufgabenstellung

Gehen Sie nun wie folgt vor: Belegen Sie die Signalfunktion (45) mit einem normalverteilten *noise* mit der Standardabweichung (a) $\sigma = 0.5$ und
(b) $\sigma = 1.5$.

- Grafische Darstellung der korrumpierten Signalfunktionen,
- grafische Darstellung der entsprechenden AK,
- Diskussion.

8. Kreuzkorrelation

Der letzte Abschnitt soll Ihnen noch einmal zeigen, wie bedeutend Korrelationseffekte für die Datenauswertung in Physik und Technik sein können. Als Beispiel soll nun die Echolot-Entfernungsmessung diskutiert werden.

Das Prinzip, das Ihnen sicherlich bekannt ist, ist in der folgenden Zeichnung etwas naiv dargestellt: ein Schiff sendet von der Meeresoberfläche eine Schallwelle $s(t)$ in Richtung Meeresgrund. Ein Teil des Signals wird dort reflektiert und kehrt zur Meeresoberfläche zurück, wo es mit einer Zeitverschiebung Δt registriert wird. Kennt man die Ausbreitungsgeschwindigkeit c der Schallwelle im Wasser, so kann die Meerestiefe A mittels der Gleichung

$$2A = \frac{\Delta t}{c} \tag{48}$$

bestimmt werden.

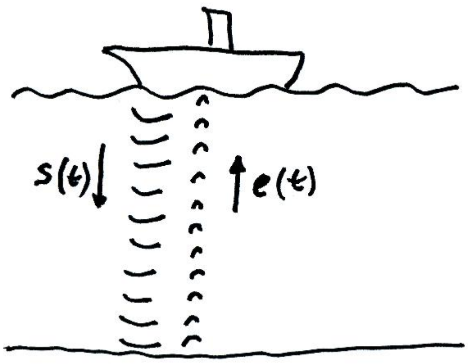


Abb.: Das Echolot-Prinzip

Obwohl die folgenden numerischen Tests gegenüber der Realität sehr vereinfacht sind - insbesondere was die verwendeten Einheiten betrifft -, wird das Prinzip der Vorgangsweise korrekt wiedergegeben!

- Erste Komplikation:

  Auf dem Weg durch das Wasser wird das ausgesendete Signal $s(t)$ sehr stark amplituden-gedämpft, und außerdem wird nur ein Teil des Signals reflektiert. Das registrierte Echo-Signal $e(t)$ hat die Form

  $$e(t) = \alpha\, s(t - \Delta t)\,. \tag{49}$$

  α beschreibt die Amplituden-Reduktion, und Δt ist die zeitliche Verzögerung des Echos.

Aufgabenstellung

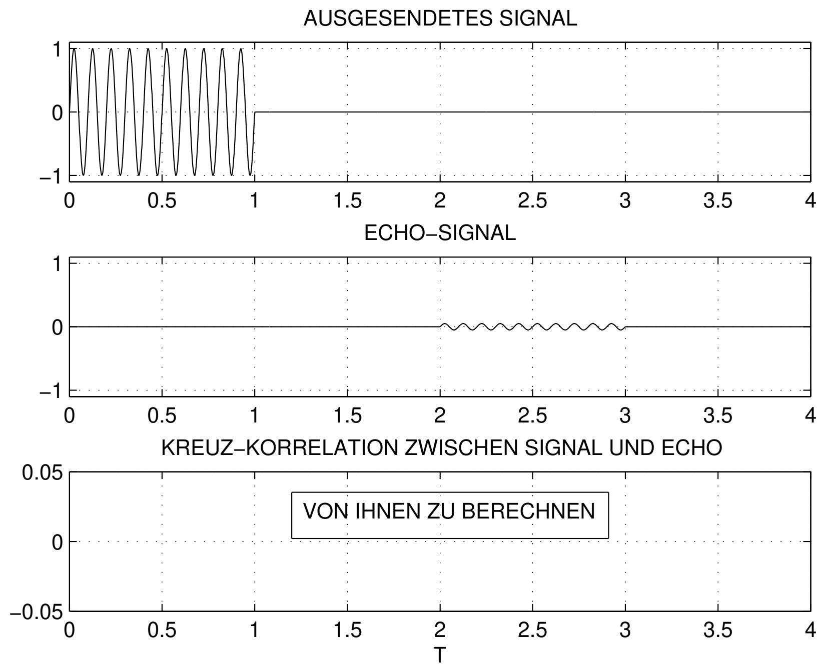
Alle 4 Zeiteinheiten wird ein Signal ausgesendet (Periode $T = 4$), welches 1 Zeiteinheit dauert und aus 10 Sinusschwingungen besteht:

$$s(t) = \begin{cases} \sin(20\pi\, t) & 0 \le t \le 1\,, \\ 0 & 1 < t \le 4\,. \end{cases} \tag{50}$$

Die Laufzeit des Echosignals sei 2 Zeiteinheiten, und die gedämpfte Amplitude sei $\alpha = 0.05$:

$$e(t) = \alpha\, \sin\left[20\pi(t-2)\right] \qquad \text{für} \quad 2 \le t \le 3\,, \text{sonst} \quad 0\,. \tag{51}$$

Diese Ausgangssituation ist im folgenden Diagramm dargestellt:



Die zeitliche Differenz zwischen dem Beginn des Signals (t=0) und dem Einsetzen des Echos (t=2) ermöglicht nach Glg. (48) eine Aussage über die Meerestiefe.

Nun kommt Ihre Arbeit:

Bestimmen Sie die sogenannte <u>Kreuzkorrelation</u> zwischen dem Signal und dem Echo, d.h. berechnen Sie mittels DFFT die Funktion

$$s \odot e \tag{52}$$

Wie sieht die Kreuzkorrelations-Funktion aus (Diagramm!)?

Diskutieren Sie die Bedeutung dieser Funktion für das Echolot-Verfahren.

- Zweite Komplikation:

  Es ist klar, daß das empfangene Echo bei seiner "Reise" durch das Wasser durch die verschiedensten Einflüsse stark korrumpiert wird. Es soll hier angenommen werden, daß diese Effekte durch eine normalverteilte Verrauschung mit einer Standardabweichung σ beschrieben werden kann:

  $$e_\sigma(t) = \alpha\, s(t - \Delta t) + n_\sigma(t)\,. \tag{53}$$

  Solange die Echo- und die *noise*-Amplitude vergleichbar groß sind, wird das Eintreffen des Echo-Signals trotz der Verrauschung gut beobachtet werden können. Wird die *noise*-Amplitude aber zu groß, so "ertrinkt" das Echo-Signal im statistischen Rauschen, und seine Identifizierung wird schwierig bis unmöglich.

  Nun kommt Ihre Arbeit:

  Zeigen Sie, wie die Numerik bzw. die DFFT auch hier helfen kann, indem man wieder die Kreuzkorrelations-Funktion zwischen dem korrumpierten Echo-Signal $e_\sigma(t)$ und dem gesendeten Signal $s(t)$ berechnet.

  Produzieren Sie 2 Diagramme "im Stil" des letzten hier gezeigten Diagramms für die folgenden *noise*-Standardabweichungen:

  (a) σ=0.05 (b) σ=0.075 (c) σ=0.100

Literatur:

H. Sormann, Vorlesungsskriptum *Numerische Methoden in der Physik*, Abschnitt 3.4.1

P. L. DeVries, *Computerphysik*, Spektrum Akademischer Verlag, Heidelberg 1995, Kap. 6

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press 1994, Kap. 12 und 13.

Ch. Überhuber, *Computerphysik 2*, Springer-Verlag Berlin, 1995, Kap. 11.