

# NUMERISCHE METHODEN IN DER PHYSIK

## Zweite Übung WS 2012/2013 [MATLAB]

### Schnelle diskrete Fourier-Transformation (Fast Fourier Transform FFT)

Darstellung der Methode: Skriptum Kap. 3.3 und 3.4.

Die Berechnungen für diese Übung sollen alle unter Verwendung der von mir variierten MATLAB-Routinen

`fft_sor.m` und `ifft_sor.m`

durchgeführt werden. Alles, was Sie über diese Routinen wissen sollten, steht im VL-Skriptum, Abschnitt 3.3.2.

#### Liste der Aufgaben im Rahmen dieser Übung:

1. Austesten der FFT-Routinen an Hand der Testdaten im VL-Skriptum S. 84.  
Dokumentation der Wirksamkeit der sog. "Frequenz-Optimierung" (VL-Skriptum, Abschnitt 3.4.3).
2. Weitere Tests zu den Themen "Frequenzanalyse", "Frequenzauflösung" und "Aliasing-Effekt" (s. VL-Skriptum, Abschnitt 3.3.4).
3. Entfaltung von glatten und verrauschten Meßwerten; Dirichlet-Glättung (s. VL-Skriptum, Abschnitte 3.4.1 und 3.4.2).

## 1. Aufgabe

### Austesten von FFT an Hand der Testdaten Skriptum S. 84.

- Wenden Sie das Matlab-Programm *fft\_sor* auf die folgenden 8 komplexen Werte an ( $n = 8, \Delta = 1.0$ ):

j	RT{y(j)}	IT{y(j)}
0	0.7013	0.0437
1	-0.0724	0.5133
2	0.0988	-0.2688
3	0.0715	-0.1162
4	0.4013	0.1188
5	-0.0901	-0.1408
6	-0.1263	-0.0688
7	0.2660	-0.3813

Die gegebenen Stützstellen repräsentieren somit eine periodische Funktion mit der Periode  $P = 8$ , und das Programm *fft\_sor* sollte die folgenden Fourierkoeffizienten liefern:

k	RT{Y(k)}	IT{Y(k)}
0	1.2501	-0.3001
1	0.0001	0.3000
2	0.2601	0.0001
3	-0.7000	-0.7003
4	0.9001	-0.0501
5	0.9999	0.0000
6	2.0001	1.0001
7	0.9000	0.0999

- Wenden Sie anschließend die Routine *ifft\_sor.m* auf die erhaltenen Fourierkoeffizienten an und demonstrieren Sie, daß sich wieder die ursprünglichen Daten ergeben.
- Berechnen Sie unter Verwendung dieser Fourierkoeffizienten die Interpolationsfunktion  $I(x)$  gemäß Skriptum Glg. (3.15) für  $0 \leq x \leq 7$  mit  $\Delta x = 0.05$ .  
Erstellen Sie zwei Diagramme (getrennt für Real- und Imaginärteil), welche die gegebenen Datenwerte plus die berechnete Interpolationsfunktion zeigen.

- Wie im Abschnitt 3.4.3 des VL-Skriptums ("Frequenz-Analyse") erläutert wird, ist die oben berechnete Interpolationsfunktion nicht die günstigste (glatteste) Verbindung der gegebenen Punkte. Variieren Sie Ihr Programm unter Verwendung der Gleichungen Skriptum (3.26–3.28), und erstellen Sie **zwei weitere Diagramme**, welche die gegebenen Daten und die "glattest-mögliche" Interpolationskurve darstellen (bitte wieder getrennte Diagramme für die Real- und Imaginärteile).

## 2. Aufgabe

Einige Tests zum Thema "Frequenzanalyse" (s. Skriptum, Abschnitt 3.4.3).

- Schreiben Sie ein Programm, mit Hilfe dessen Sie eine Frequenz-Analyse von  $n$  reellen Daten durchführen können, wobei Sie die Testdaten aus der folgenden Funktion berechnen:

$$y(t) = 2*\cos[2 \text{ pi } f_1 t] - 3*\sin[2 \text{ pi } f_2 t] - 1*\cos[2 \text{ pi } f_3 t] + 2*\sin[2 \text{ pi } f_4 t] + 2.5*\cos[2 \text{ pi } f_5 t]$$

$$\text{fuer } t = t_{\{j\}} = j*\delta \quad \text{mit } \delta = P/n \quad (j=0, \dots, n-1).$$

$$n = \text{Anzahl der Stuetzpunkte} \quad P = \text{Periode in Sekunden}$$

Die  $f_1 \dots f_5$  sind die Frequenzen der Testfunktion in Hertz.

Starten Sie die Testrechnungen mit den folgenden Angaben:

$$n = 64 \quad P = 1 \text{ s}$$

$$\text{Frequenzen: } f_1 = 2.0 \quad f_2 = 4.0 \quad f_3 = 4.0 \quad f_4 = 7.0 \quad f_5 = 12.0 \text{ Hz}$$

Verifizieren Sie, daß die mit Ihrem Programm durchgeführte Frequenzanalyse genau die Frequenzen und Intensitäten der Testfunktion ermittelt. Geben Sie die Ergebnisse auf dem Bildschirm in Form einer Tabelle (wie im Skriptum, S. 93) aus.

Anmerkung: In Ihrer Tabelle gibt es viele Frequenzen mit "Null-Intensitäten". Unterdrücken Sie durch eine geeignete Abfrage die Ausgabe solcher Zeilen.

- In diesem Test sollen Sie untersuchen, wie Ihr Programm reagiert, wenn Sie einen der oben angegebenen Input-Parameter ändern, nämlich:

$$f_4 = 7.5 \text{ Hz}$$

- Diskutieren Sie die Ergebnisse.
- Welchen weiteren Input-Parameter müssen Sie (im Sinne von Skriptum, Kap. 3.4.3) ändern, um korrekte Ergebnisse zu erhalten?

- Für diesen Test arbeiten Sie mit den folgenden Parametern:

$$n = 64 \quad P = 4 \text{ s}$$

$$\text{Frequenzen: } f_1 = 2.75 \quad f_2 = 4.0 \quad f_3 = 4.0 \quad f_4 = 7.0 \quad f_5 = 13.0 \text{ Hz}$$

- Liefert Ihre Frequenzanalyse die richtigen Ergebnisse?  
Wenn Nein, woran kann das liegen?
  - Was können Sie tun, um auch in diesem Fall ein korrektes Ergebnis zu erhalten?
- Alle bisherigen Testbeispiele dieser Aufgabe bestanden aus einer endlichen Zahl von Cosinus- bzw. Sinus-Termen. Eine Fourieranalyse ist in solchen Fällen kein Problem. Schwieriger wird die Sache, wenn die zu analysierende Testfunktion *Unstetigkeiten* im Funktionswert bzw. in einer Ableitung besitzt. Dieser Fall soll nun untersucht werden, und zwar an Hand der Testfunktion

$$\begin{array}{llll}
 P = 4 \text{ s} & y(t) = 1 & \text{fuer } 0 \leq t < 0.5 & , \\
 & y(t) = 1 + 2(t-0.5) & \text{fuer } 0.5 \leq t < 1.5 & , \\
 & y(t) = 147/40 - 3/10 t^2 & \text{fuer } 1.5 \leq t < 3.5 & , \\
 & y(t) = 9 - 2 t & \text{fuer } 3.5 \leq t < 4 & .
 \end{array}$$

Führen Sie die Frequenzanalyse für die folgenden Stützpunktzahlen durch:  $n = 256, 512$  und  $1024$ . Dokumentieren Sie Ihre Ergebnisse in **einem** Diagramm der folgenden Art:

Stellen Sie die *Absolutquadrate der Fourierkoeffizienten, dividiert durch die jeweiligen Werte von  $n$* , in einem bzgl. der  $y$ -Achse halblogarithmischen Koordinatensystem dar. In Matlab steht Ihnen dafür die folgende Routine zur Verfügung:

```
semilogy(x-Vektor,y-Vektor,'...')
```

Diskutieren Sie an Hand dieses Diagramms die Wirkung des *aliasing*-Effekts.

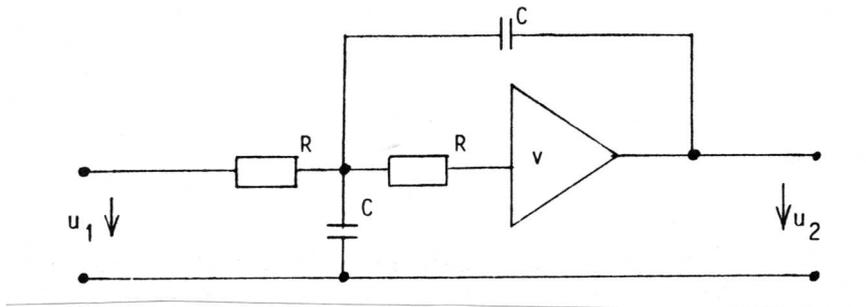
### 3. Aufgabe

#### Entfaltung von glatten und verrauschten Meßwerten. Dirichlet-Glättung

In dieser Aufgabe soll ein Problem untersucht werden, das bei der Auswertung experimenteller Daten eine Hauptrolle spielt. Es geht um die Filterung gemessener Werte von Einflüssen von zweierlei Art: (1) Einflüsse der Meßapparatur, (2) Einflüsse statistischer Meßfehler.

Beide Effekte werden im VL-Skriptum in Kürze beschrieben, und zwar in den Abschnitten 3.4.1 und 3.4.2.

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



Quelle: P. Jacobs und S. Jancar, *BASIC: Interpolationen-Approximationen-Splines-FFT*, ed. H. Kohler, Vieweg-Verlag.  $u_1 = s(t)$   $u_2 = m(t)$

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

$$m(t) = \int_{-\infty}^{+\infty} dt' r(t-t') s(t') \quad (1)$$

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

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

Dazu bedarf es natürlich der Kenntnis der Filterfunktion  $r(t)$ ; 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)

$$s(t) = \delta(t) \quad \rightarrow \quad m(t) = \int_{t'=-\infty}^{+\infty} dt' r(t-t') \delta(t') = r(t).$$

Eine konkrete Filterfunktion  $r(t)$  und ein konkretes Ausgangssignal ("Meßfunktion")  $m(t)$  stehen in tabellarischer Form auf den Files



Bisher haben wir uns in der 3. Aufgabe dieser Übung ausschließlich mit einer Meßkurve  $m(t)$  beschäftigt, welche keinerlei statistische Fehler enthielt. 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.

Im Folgenden geht es nun darum, den Einfluß der statistischen Fehler (des *noise*) auf die Rekonstruktion der Signalfunktion zu studieren und gegebenenfalls durch geeignete Methoden zu eliminieren.

Bei einem qualitätvollen 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 die tabellarisch gegebene, glatte Meßfunktion  $m(t)$  auf dem File *messung6.txt* künstlich verrauscht werden, und zwar durch Addition einer *noise*-Funktion  $n(t)$ :

$$\tilde{m}(t_j) = m(t_j) + n(t_j), \quad (3)$$

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

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{n^2}{2\sigma^2}\right). \quad (4)$$

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.

### 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 diese Tests wieder die Datenfiles

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

```
filter_a.txt      % 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
```

- **Produzieren Sie für jedes sigma die beiden folgenden Diagramme:**
  - Diagramm 1: Vergleichen Sie die Messfunktionen  $m(t)$  und  $\tilde{m}(t)$ .
  - Diagramm 2: Vergleichen Sie die entsprechenden Signalfunktionen  $s(t)$  und  $\tilde{s}(t)$ .

## Dirichlet-Dämpfung:

Die letzten Rechnungen haben Ihnen deutlich gezeigt, daß ein simples Ignorieren von statistischen Fehlern in den Meßdaten völlig unmöglich ist, da bereits sehr kleine Verrauschungen, 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. Eine sehr einfache Methode dazu finden Sie im Vorlesungsskriptum, Abschnitt 3.4.2, unter dem Namen "Dirichlet-Dämpfung".

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 Fourierkoeffizienten 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 dazu das Datenmaterial

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

```
filter_a_1.txt     % 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  $s(t)$ <sup>1</sup>.
- 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.

---

<sup>1</sup>Achtung: wenn die Dirichlet-Dämpfung korrekt ist, müssen alle rekonstruierten Werte der Signalfunktion reell sein. Das Auftreten von Imaginärteilen weist auf einen Fehler in Ihrem Programm hin!