Seminar zur Programmierung

Die Fibonacci-Folge und die Vermehrung von Kaninchen

- Im Jahr 1202 stellte Fibonacci ein **idealisiertes Schema** für die Fortpflanzung von Kaninchen auf:
- ***Bedingungen:***
  1.) Start mit jeweils einem neugeborenen weiblichen und männlichen Kaninchen.
  2.) Ein Kaninchen wird nach einem Monat geschlechtsreif.
  3.) Die Tragezeit eines Kaninchens ist ein Monat.
  4.) Nach Erreichen der Geschlechtsreife gebiert ein weibliches Kaninchen jeden Monat.
  5.) Ein weibliches Kaninchen gebiert immer ein weibliches und ein männliches Kaninchen.
  6.) Kaninchen sind unsterblich.
- 1. Monat: Am Anfang gibt es ein neu geborenes Kaninchenpaar ➔ Anzahl Paare: **1**
  2. Monat: Das Paar wird geschlechtsreif, aber noch keinen Nachwuchs ➔ Anzahl Paare: **1**
  3. Monat: Das Paar gebiert ein neues Paar ➔ Anzahl Paare: **2**
  4. Monat: Das 1. Paar gebiert ein weiteres Paar, das 2. wird geschlechtsreif ➔ Anzahl Paare: **3**
  5. Monat: Die ersten beiden Paare gebären, das 3. wird geschlechtsreif ➔ Anzahl Paare: **5**
  6. Monat**: Alle Pärchen, die vor zwei Monaten gelebt haben, gebären** ➔ Anzahl Paare: **8**
  Bezeichnet also x_k die Anzahl der Kaninchenpaare zum Zeitpunkt n, so gewinnen wir folgendes Bildungsgesetz aus der Folge: $x_1 = 1, x_2 = 1, x_3 = 2, x_4 = 3, x_5 = 5, x_6 = 8, \ldots, x_{k+1} = x_k + x_{k-1}$
- Fibonacci stellte die Frage:“Wieviele Kaninchenpaare gibt es nach einem Jahr?”
  http://people.math.uni-bonn.de/woermann/fibonacci.pdf , https://de.wikipedia.org/wiki/Fibonacci-Folge

Realisierung der Fibonacci-Folge

- Rekursive gebildete Folge ➔ ***rekursives Programm***
- Benötigen 2 Startwerte: $x_1 = 1$ und $x_2 = 1$
  Rekursion: $\mathbf{x_{k+1} = x_k + x_{k-1}}$
- **bzw. in Matrizenschreibweise:**

$$\begin{pmatrix} x_{k+1} \\ x_k \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix} = A \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

- oder **direkte Berechnung des k-ten Gliedes** der Folge:

$$\begin{pmatrix} x_{k+1} \\ x_k \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{k-1} \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = A^{k-1} \begin{pmatrix} x_2 \\ x_1 \end{pmatrix}; \quad k \geq 2$$

Verschiedene Programmvarianten:

- ***Fibo.m*** ➔Skriptfile, direkte Ber. sehr einfach, ein Element wird berechnet
- Fibofolge.m ➔Skriptfile, rekursive Ber. ➔ while-Schleife, alle Folgenelemente bis zum k-ten werden berechnet
- ***fibofkt.m*** ➔Funktionsdatei, direkte Berechnung, Startwertübergabe,
- ***FibomitFkt.m*** ➔ Hauptprogramm mit Aufruf von fibofkt.m
- ***Fibogesamtmitfkt.m*** ➔ Hauptprogramm mit while-Schleife, in der fibofkt zur Berechnung der einzelnen Folgenelemente gerufen wird,
- ***Fibogesmitfktuebergabe.m*** ➔ Funktionsdatei, neu: Übergabe des Namens des aufzurufenden Unterprogrammes als Zeichenketten-Parameter: ‘fibofkt‘

Lösung der Gleichung $x^2 - 2px + q = 0$ (1)

1. Gesucht: 2 Lösungen x_1 und x_2 von Gleichung (1)
2. Lösungsformel ist numerisch instabil: $$x_{1/2} = -\frac{(-2p)}{2} \pm \sqrt{\frac{4p^2}{4} - q} = p \pm \sqrt{p^2 - q}$$
3. ***besser:*** x_1 so berechnen, dass ***keine Stellenauslöschung erfolgt***, d.h.

   bei p > 0 zuerst $x_1 = p + \sqrt{p^2 - q}$ berechnen bzw.

   bei p < 0 zuerst $x_1 = p - \sqrt{p^2 - q}$ berechnen.

Danach x_2 aus $q = x_1 * x_2$ bestimmen.

Lösung der Gleichung $x^2 - 2px + q = 0$

3. Gegeben ist die Gleichung ➔ -2p, q ➔ 4. Eingabe von p und q am BS

5. Ausgabe der Lösungen am BS

8. 3 verschiedene Lösungsmöglichkeiten: eine doppelte reelle Lösung, zwei einfache reelle Lösungen, ein Paar konjugiert komplexer Lösungen:

$$(x-1)(x-1) = x^2 - 2x + 1 = 0 \Rightarrow x_1 = 1;\ x_2 = 1; \quad (-2p = -2 \rightarrow p = 1; \quad q = 1)$$

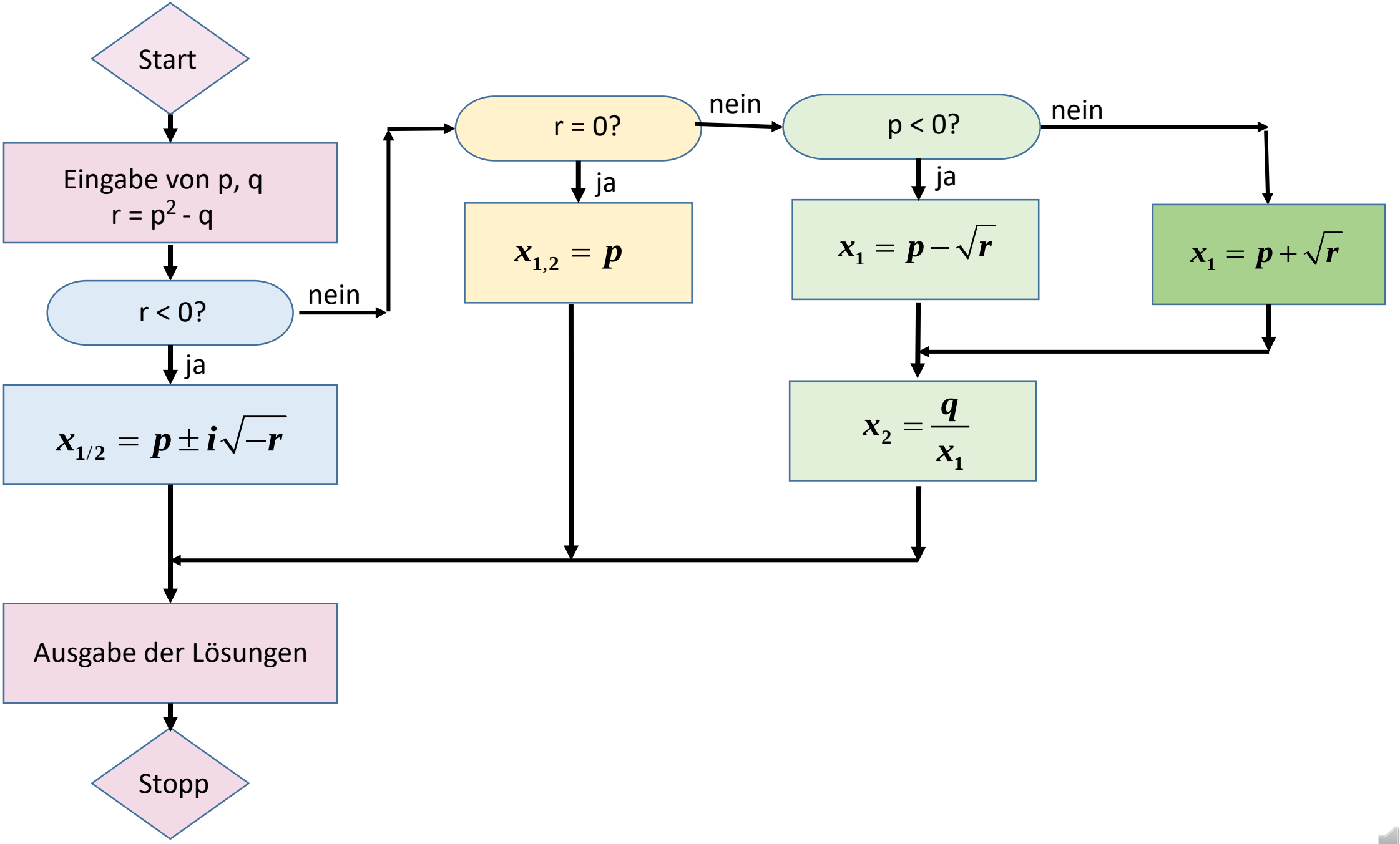
$$(x-1)(x-2) = x^2 - 3x + 2 = 0 \Rightarrow x_1 = 1;\ x_2 = 2; \quad (-2p = -3 \rightarrow p = 1.5; \quad q = 2)$$

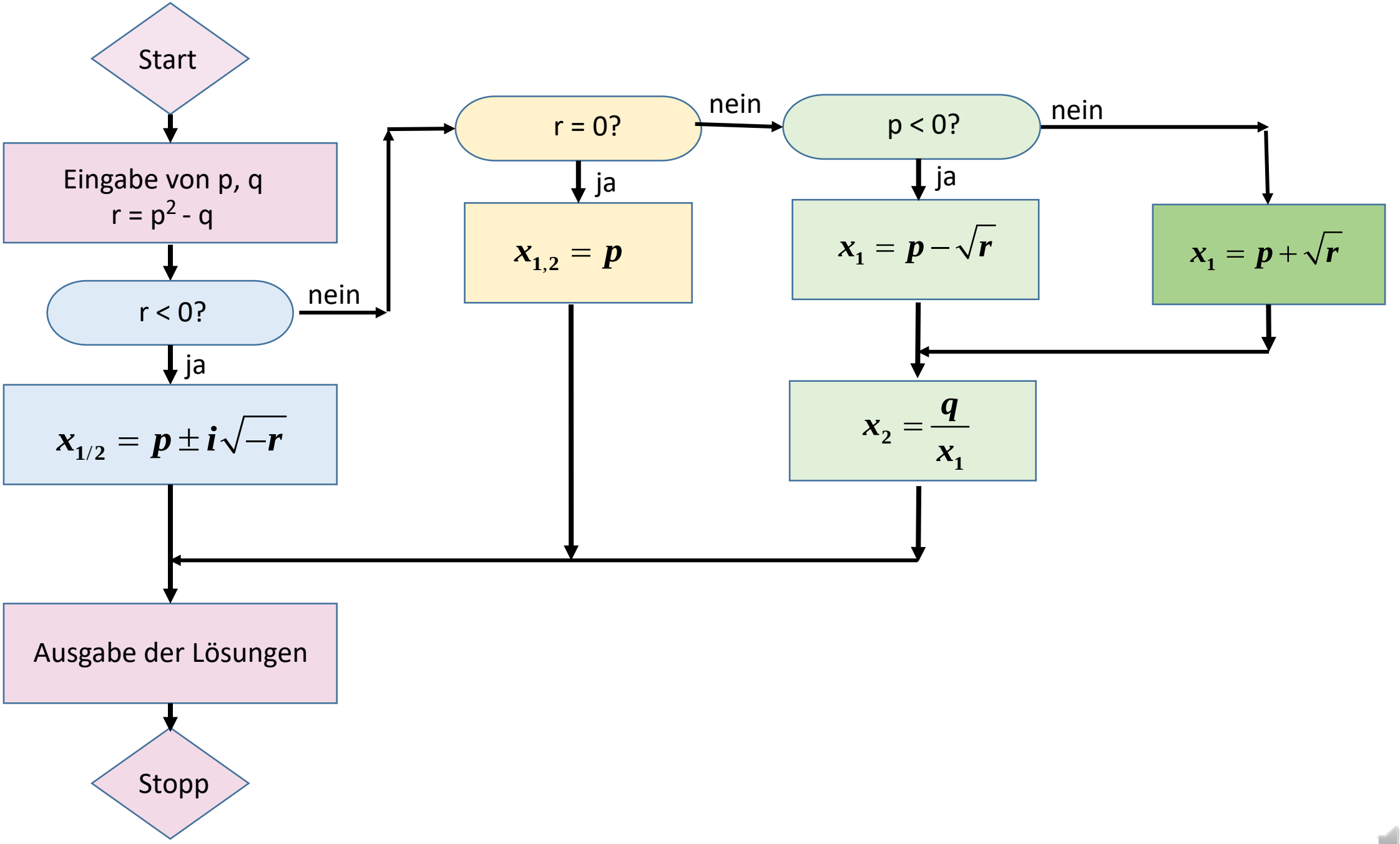
$$x^2 + 1 = 0 \Rightarrow x_1 = i;\ x_2 = -i; \quad (-2p = 0 \rightarrow p = 0; \quad q = 1)$$

9. Sonderfälle im Test: p = 0; q =0; p = 0 und q =0; Nulldivision?

$$(x-1)x = x^2 - x = 0 \Rightarrow x_1 = 1;\ x_2 = 0; \quad (-2p = -1 \rightarrow p = 0.5; \quad q = 0)$$

$$x^2 = 0 \Rightarrow x_1 = 0;\ x_2 = 0; \quad (-2p = 0 \rightarrow p = 0; \quad q = 0)$$


Start
Eingabe von p, q
r = p² - q
r < 0?
nein
ja
r = 0?
nein
ja
p < 0?
nein
ja
$x_{1/2} = p \pm i\sqrt{-r}$
$x_{1,2} = p$
$x_1 = p - \sqrt{r}$
$x_1 = p + \sqrt{r}$
$x_2 = \frac{q}{x_1}$
Ausgabe der Lösungen
Stopp


Start
Eingabe von p, q
r = p² - q
r < 0?
nein
ja
r = 0?
nein
ja
p < 0?
nein
ja
$x_{1/2} = p \pm i\sqrt{-r}$
$x_{1,2} = p$
$x_1 = p - \sqrt{r}$
$x_1 = p + \sqrt{r}$
$x_2 = \frac{q}{x_1}$
Ausgabe der Lösungen
Stopp

Quadratische Gleichung: $x^2 - 2px + q = 0$

```
p=input('p = '); q=input('q= ');
r=p.^2-q
if r<0
   x1=p+i*sqrt(-r);
   x2=p-i*sqrt(-r);
elseif r==0
   x1=p;
   x2=x1;
else
   if p<0
      x1=p-sqrt(r);
   else
      x1=p+sqrt(r);
   end
   x2=q/x1;
end
x1,x2
```

➔ **Einfachstes Programm nach PAP**

➔ **Testen mit den ausgewählten Bsp.**

➔ **„Verschönern“:**

- Kommentare:
  Was macht das Programm?
  Was sind p und q für Größen?
  Warum nicht die übliche Formel?
- Doppelte Berechnungen beseitigen
- Eingabe und Ausgabe informativer machen

Quadratische Gleichung (1)

$$x^2 - 2px + q = 0$$

```
% Programm QUADGLEI: rundungsfehlerstabile
% Berechnung der NST der quadrat. Gleichg. x^2 - 2px + q = 0
char = 'j';
while char == 'j' | char == 'J'
    % Eingabe der Parameter, Festlegung der allgemeingueltigen Groessen
    ……Programmtext 1……..
    % Berechnung der Nullstellen
    ……Programmtext 2……
   disp('  ')
   char = input(' Noch eine quadratische Gleichung? ', 's');
end
```

% Eingabe der Parameter, Festlegung der allgemeingültigen Größen:

Programmtext 1:

```
disp(' Loesung der quadratischen Gleichung x^2 - 2px + q = 0 ')
disp('  ')
p = input(' Eingabe des Parameters p:  ');
disp('  ')
q = input(' Eingabe des Parameters q:  ');
disp('  ')
disp(' Nullstellen x1 und x2 der quadratischen Gleichung: ')
disp('  ')

r = p*p - q;          % Berechnung der Diskriminante
wr = sqrt(abs(r)); % Vermeidung doppelter Berechnungen
```

Programmtext 2

```
% Berechnung der Nullstellen:
    if r < 0                % Komplexwertige NST
        x1 = p + i*wr;
        x2 = p -  i*wr;
    elseif r==0             % doppelte reelle NST
        x1=p;
        x2=x1;
    else                    % zwei einfache reelle NST
        if p<0
            wr=-wr;
        end
        x1=p+wr;
        x2=q/x1;
    end
    x1,x2
```

Übungen zur Fehleranalyse

1. Werten Sie folgende Identität einzeln numerisch mit Hilfe eines Matlab-Unterprogrammes aus und berechnen Sie die absoluten und relativen Fehler. Der korrekte Wert lautet $A = 0.005051$. Interpretieren Sie die Resultate!:

$$A = \left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)^3 = \left(\sqrt{2}-1\right)^6 = \left(3-2\sqrt{2}\right)^3 = 99-70\sqrt{2}$$

a) mit $\sqrt{2} \approx \frac{7}{5} = 1.4$ b) mit $\sqrt{2} \approx \frac{17}{12} = 1.41\bar{6}$

c) Formen Sie den Ausdruck $99-70\sqrt{2}$ in eine numerisch günstigere Form um und berechnen Sie dann ebenfalls den absoluten und relativen Fehler. Hinweis: Erweitern Sie mit $99+70\sqrt{2}$ (3. binomische Formel!)

2. Die folgenden Ausdrücke sollen so umgeformt werden, dass ihre Auswertung numerisch "gutartig" ist:

			Hinweis
a)	$\frac{1}{1+2x} - \frac{1-x}{1+x}$	für $\mid x \mid \ll 1$	Hauptnenner
b)	$\sqrt{1+\frac{1}{x}} - \sqrt{1-\frac{1}{x}}$	für $x \gg 1$	analog zu 1c)
c)	$\frac{1-\cos x}{x}$	für $x \neq 0, \quad \mid x \mid \ll 1$	ersetze (1- cos x) durch Halbwinkelformel
d)	$\log x - \log y$	für $0 < x \approx y$	Logarithmengesetze

3. Berechnen Sie mit Hilfe der beiden angegebenen Rekursionsformeln die Folge der Integrale I_n.

$$I_n = \int_0^1 \frac{x^n}{x+5} dx;\ n = 0, 1, ..., 30.$$

$$a) \quad I_0 = \ln(\frac{6}{5}); \qquad I_n = \frac{1}{n} - 5I_{n-1}$$

$$b) \quad I_{30} = 0; \qquad I_{n-1} = \frac{1}{5}\left(\frac{1}{n} - I_n\right)$$

Was beobachten Sie? Untersuchen Sie die numerische Stabilität der beiden Verfahren entsprechend dem Beispiel aus der Vorlesung.

Übung Fouriertransformation

A) Aufgaben zur Vorbereitung

Wenn Sie unsicher sind beim Lösen der folgenden Aufgaben, schauen Sie bitte in die Lösungen. Dort gibt es erst einmal Tipps, wie man die Lösung finden kann.

1. Benutzen Sie die Funktionen $f_1(t)$ und $f_3(t)$ aus der Tabelle der bekannten Fouriertransformierten.

   a) Zeichnen Sie die Funktionen
   $g_1(t) = f_1(t-T),\ g_2(t) = f_1(3t),\ g_3(t) = f_1(\frac{t}{4}),\ g_4(t) = -f_1(t)$, geben Sie eine formelmäßige Beschreibung der Funktionen analog zur Tabelle an und lesen Sie „Breite", „Höhe" und Verschiebung der Funktion im Vergleich zur Tabellenfunktion an, also bei $g_1(t)$ wäre das: Breite $= T$, Höhe $= U_0$ und Verschiebung $b = -T$.

   b) Führen Sie das Entsprechende für die Funktion $f_3(t)$ durch:
   $h_1(t) = f_3(t-T),\ h_2(t) = f_3(3t),\ h_3(t) = f_3(\frac{t}{4}),\ h_4(t) = -f_3(t)$.

2. Zeichnen Sie folgende Funktionen und vergleichen Sie sie mit den Funktionen $f_1(t)$ und $f_3(t)$ aus der Tabelle der bekannten Fouriertransformierten. Geben Sie jeweils „Breite" und „Höhe" sowie die Verschiebung der Funktionen an.

   a) $f(t) = \begin{cases} U_0 & 0 \le t \le T \\ 0 & sonst \end{cases}$ (verschobener Rechteckimpuls)

   b) $f(t) = \begin{cases} U_0 & -T \le t \le T \\ 0 & sonst \end{cases}$

   c) $f(t) = \begin{cases} -U_0 & -\frac{T}{4} \le t \le \frac{T}{4} \\ 0 & sonst \end{cases}$

   d) $f(t) = \begin{cases} \frac{U_0}{T}t & 0 \le t \le T \\ -\frac{U_0}{T}t + 2U_0 & T \le t \le 2T \\ 0 & sonst \end{cases}$ (verschobener Dreiecksimpuls)

   e) $f(t) = \begin{cases} \frac{U_0}{2T}t + U_0 & -2T \le t \le 0 \\ -\frac{U_0}{2T}t + U_0 & 0 \le t \le 2T \\ 0 & sonst \end{cases}$ (gestreckter Dreiecksimpuls)

B) Übungsaufgaben

1. Zeichnen Sie die folgenden Funktionen und berechnen Sie die Fouriertransformierten von $f(t)$ unter Anwendung bekannter Transformationen aus der Tabelle sowie unter der Verwendung von Eigenschaften und Rechenregeln.

   a) $f(t) = \begin{cases} 3 & 7 \le t \le 11 \\ 0 & sonst \end{cases}$

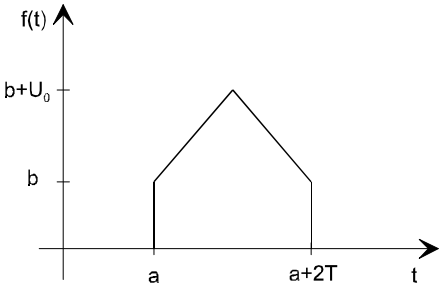
   b) $f(t) = \begin{cases} 2t-2 & 0 \le t \le 1 \\ -2t-2 & -1 \le t \le 0 \\ 0 & sonst \end{cases}$

   c) $f(t) = \begin{cases} 4(t-3.5)+2 & 3 \le t \le 3.5 \\ -4(t-3.5)+2 & 3.5 \le t \le 4 \\ 0 & sonst \end{cases}$

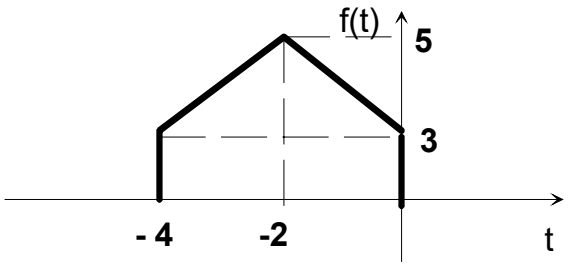
2. Berechnen Sie die Fouriertransformierten $F(i\omega)$ der Funktionen aus der Aufgabe A) 2. unter Anwendung bekannter Transformationen sowie unter der Verwendung von Eigenschaften und Rechenregeln.

3. Berechnen Sie die Fouriertransformierten $F(i\omega)$ der Funktion aus der Zeichnung, indem Sie diese zuerst zeichnerisch in die Summe von Funktionen aus der Tabelle zerlegen und dann den Additionssatz anwenden.

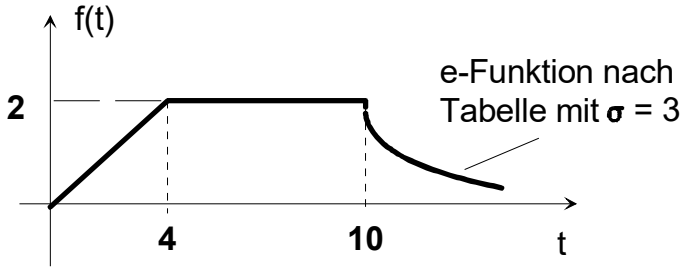
   a)



b)



c)



Zeichnen Sie nun die Funktionen aus den restlichen Aufgaben. Danach wollen wir die Fouriertransformierten dazu berechnen, wobei die Aufgaben 4 und 5 durch additive Zerlegung in 2 Funktionen vom Typ $f_2(t)$ aus der Tabelle behandelt werden und Aufgabe 10 über die Definition berechnet werden muss. Die restlichen Aufgaben 6 bis 9 sind sehr praxisrelevant, aber nicht prüfungsrelevant, weil dabei Distributionen benötigt werden. Diese Lösungen werden in einer Übungsvorlesung erklärt. Deshalb gibt es für diese Aufgaben 6 bis 9 keine Lösungen in diesem Übungsblatt.

4. $f(t) = e^{-a|t|}, \qquad a > 0,\ t \in \mathbb{R}$ (symmetrisch abfallender Impuls)

5. $f(t) = e^{-a|t|} sgn(t), \qquad a > 0,\ t \in \mathbb{R}$ (antisymmetrisch abfallender Impuls)

6. $f(t) = sgn(t) = \begin{cases} 1 & t > 0 \\ -1 & t < 0 \\ 0 & t = 0 \end{cases}$ (Signumfunktion)

7. $s(t) = \int_{-\infty}^{t} \delta(\tau) d\tau = \begin{cases} 1 & 0 \le t \le \infty \\ 0 & sonst \end{cases}$ (Sprungfunktion)

8. Einheitsstoß oder Dirac - Impuls $\delta(t)$

9. $f(t) = 1 \qquad für\ t \in \mathbb{R}$

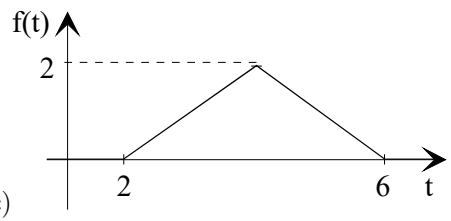
10. $f(t) = \begin{cases} 1 - t & 0 < t \le 1 \\ 0 & sonst \end{cases}$

C) Hausaufgabe

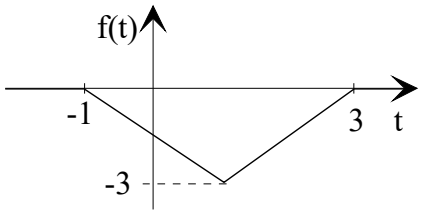
1. Bestimmen Sie die Fouriertransformierte von

a) $f(t) = \begin{cases} -2 & -5 \le t \le 5 \\ 0 & sonst \end{cases}$

b) $f(t) = \begin{cases} U_0 & -1 \le t \le 4 \\ 0 & sonst \end{cases}$

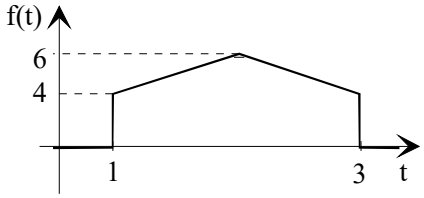


c)

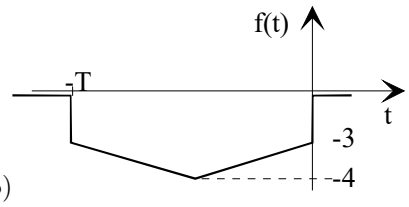


d)

2. Bestimmen Sie die Fouriertransformierte von



a)



b)

Lösungen zum Teil A)

Hier gibt es nur Tipps zur Lösung. Die komplette Lösung finden Sie eingescannt in einer separaten Datei.

zu 1. allgemein:

- Überlegen Sie , was die Veränderungen des Argumentes bewirken und zeichnen sie danach entsprechend die Funktion:
  1.a) Verschiebung von $f_1(t)$ um T nach rechts
  1.b) Stauchung von $f_1(t)$ auf $\frac{1}{3}$ der Rechteckbreite
  1.c) Streckung von $f_1(t)$ auf das Vierfache der Rechteckbreite
  1.d) Spiegelung an der $x-$Achse

- Lesen Sie Breite, Höhe und Verschiebung aus der Zeichnung ab. Die Verschiebung erhalten Sie, wenn Sie den Abstand von 2 charakteristischen Punkte der Funktion in Mittelpunktslage (wie in der Tabelle) und der aktuellen Funktion ablesen. Geeignet sind z.B. jeweils der Anfangs-, Mittel- oder Endpunktvom nichtverschwindenenTeil der Funktion. Verschiebung um 5 nach rechts heißt. $b = -5$.
  Achtung aufpassen:
  Die Breite der Funktion $f_1(t)$ aus der Tabelle beträgt T,
  die Breite der Funktion $f_3(t)$ aus der Tabelle beträgt $2T$.

- Schreiben Sie nun die Formeln für die Funktionen auf:
  bei a) so wie Sie es sehen, weil es konstante Funktionen sind
  bei b) setzen Sie das veränderte Argument in die Formel der ursprünglichen Funktionen ein.

zu 2. allgemein:

- Zeichnen Sie zuerst ein Koordinatensystem und markieren Sie dort die Punkte auf der $t-$Achse, wo das Intervall beginnt, in dem die Funktion von Null verschieden ist.

- Danach setzen Sie den linken Randpunkt, den Mittelpunkt und den rechten Randpunkt des Intervall in die Funktionsbeschreibung ein und zeichnen diese Werte in das Koordinatensystem ein.

- Nun verbinden Sie je 2 aufeinander folgende dieser Punkte mit einer Geraden. Das ist möglich, da es sich entweder um konstante oder um lineare Funktionen handelt.
  Lesen Sie zuletzt Breite, Höhe und Verschiebung aus der Zeichnung ab.

(Nur zur Kontrolle: Es handelt sich in 2. bei den Funktionen um:
2.a) $f(t) = f_1(t - \frac{T}{2})$ 2.b) $f(t) = f_1(\frac{t}{2})$ 2.c) $f(t) = -f_1(2t)$ 2.d) $f(t) = f_3(t - T)$ 2.e) $f(t) = f_3(\frac{t}{2})$)

Lösungen zum Teil B)

1a) $F(i\omega) = 12e^{-9i\omega}si(2\omega)$ 1.b) $F(i\omega) = -\frac{4}{\omega^2}(1 - \cos(\omega))$

1c) $F(i\omega) = e^{-3.5i\omega}\frac{8}{\omega^2}(1 - \cos(\frac{\omega}{2}))$

2a) $F(i\omega) = e^{-\frac{T}{2}i\omega}U_0 Tsi(\frac{\omega T}{2})$ 2.b) $F(i\omega) = 2U_0 Tsi(\omega T)$

2c) $F(i\omega) = -\frac{1}{2}U_0 Tsi(\frac{\omega T}{4})$ 2.d) $F(i\omega) = e^{-Ti\omega}\frac{2U_0}{T\omega^2}(1 - \cos(\omega T))$

2.e) $F(i\omega) = \frac{U_0}{T\omega^2}(1 - \cos(2\omega T))$

3a) $F(i\omega) = 2e^{-(a+T)i\omega}\left[bTsi(\omega T) + \frac{U_0}{T\omega^2}(1 - \cos(\omega T))\right]$

3b) $F(i\omega) = e^{2i\omega}(12si(2\omega) + \frac{2}{\omega^2}(1 - \cos(2\omega)))$

3c) $F(i\omega) = -\frac{1}{2\omega^2}(e^{-4i\omega}(-4i\omega - 1) + 1) + 12e^{-7i\omega}si(3\omega) + \frac{2}{3+i\omega}e^{-10i\omega}$

4) $F(i\omega) = \frac{1}{a-i\omega} + \frac{1}{a+i\omega} = \frac{2a}{a^2+\omega^2}$

5) $F(i\omega) = -\frac{1}{a-i\omega} + \frac{1}{a+i\omega} = \frac{-2i\omega}{a^2+\omega^2}$

10) $F(i\omega) = \frac{1}{\omega^2}\left(1 - e^{-i\omega}\right) - \frac{i}{\omega}$

Lösungen zum Teil C)

1.a) $F(i\omega) = -20si(5\omega)$; 1.b) $F(i\omega) = 5U_0 e^{-1.5i\omega}si(2.5\omega)$

1.c) $F(i\omega) = \frac{2}{\omega^2}e^{-4i\omega}(1 - \cos(2\omega))$; 1.d) $F(i\omega) = -\frac{3}{\omega^2}e^{-i\omega}(1 - \cos(2\omega))$

2.a) $F(i\omega) = 8e^{-2i\omega}si(\omega) + \frac{4}{\omega^2}e^{-2i\omega}(1 - \cos\omega)$;

2.b) $F(i\omega) = -3Te^{i\omega T/2}si(\frac{\omega T}{2}) - \frac{4}{T\omega^2}e^{i\omega T/2}(1 - \cos(\frac{\omega T}{2}))$

Praktikum zur FFT

In Matlab enthält die Rücktransformation den Vorfaktor 1/N, wobei N die Anzahl der Datenpunkte ist, die transformiert werden sollen. Das Unterprogramm fft führt die Hintransformation aus, das UP ifft die Rücktransformation.

Beispiel: x=[1,1,1,1,0,0,0,0]; c=fft(x,8),f=ifft(c,8), abs(x-f)

Syntax: fft(vector,n), ifft(vector,n), n: Anzahl der in Betracht gezogenen Punkte, mehrere Parameter sind möglich

Kleine Details im Zeitbereich, wie z.B. Sprünge, Extrema, ... werden im Frequenzbereich über große Intervalle verschmiert. D.h. solche Details brauchen zur Rekonstruktion sehr viele verschiedene Frequenzen.

Beispiel: x=[1 zeros(1,11)];fftgui(x);
Ändern Sie zwei Datenpunkte im Zeitbereich und verfolgen Sie, welche Änderungen im Frequenzbereich auftreten:
x=[0 1 zeros(1,10)];fftgui(x);
t=linspace(0,1,50); x=sin(2*pi*t); fftgui(x)

Die beiden Peaks im letzten Diagramm rühren daher, dass die Fouriertransformierte zwar bei der Nummerierung "0" beginnt, aber eigentlich um die Nyquistfrequenz in der Darstellung verschoben ist, so dass die beiden Peaks zur selben Frequenz, der Abschneidefrequenz, gehören, aber in der Summation der diskreten FT bei $k = \pm N/2$ auftreten. Das wird mit einem zentrierten Periodogramm behoben.

Beispiel:
fs=100; %fs=1/Δt: Abtastfrequenz zum Abtastintervall Δt
t=0:1/fs:10-1/fs; % Zeitdiskretisierung in einem 10s-Intervall
x=1.3*sin(2*pi*15*t)+1.7*sin(2*pi*40*(t-2))+2.5*randn(size(t));
% Addition zweier Schwingungen mit 15 bzw. 40 Hz plus Rauschen

Es ist sinnvoll, als Anzahl der Datenpunkte eine Zweierpotenz zu benutzen, um die hohe Geschwindigkeit der FFT zu nutzen. Handelt es sich bei der zu transformierenden Funktion um eine periodische Funktion, ist bei der Aufnahme der Datenpunkte darauf zu achten. Es wird dann nur in einer Periode gerechnet, da der weitere Verlauf der Kurve durch die Periodizität bestimmt ist. Das Abtastintervall Δt bestimmt die Abschneide- oder Nyquistfrequenz und umgekehrt: $\Delta t = \frac{\pi}{b}$ Es müssen 2 Abtastungen pro Periode T erfolgen, damit die Rekonstruktion entsprechend Abtasttheorem von Shannon möglich ist:

$$f(t) = \sum_{n=-\infty}^{\infty} f(n\Delta t) \cdot si(b(t - n\Delta t))$$

Bei nichtperiodischen Funktionen ist eine Bandbegrenzung b des Signals wünschenswert und in der Praxis im Allg. vorhanden. Der zu diskretisierende Bereich des Signals sollte so groß gewählt werden, dass das Signal darin möglichst abgeklungen ist. Um zu einer Zweierpotenz von Datenpunkten zu kommen, kann dann mit Nullen bis dorthin aufgefüllt werden. Das macht Matlab automatisch, wenn die Anzahl der Transformationspunkte größer ist als die der eingegebenen Datenpunkte. Ist die Funktion jedoch noch nicht abgeklungen, führt diese Auffüllung mit Nullen zu einem Sprung in der Funktion, der real nicht vorhanden ist, aber mittransformiert wird und das Ergebnis verfälscht.

Allgemein sollte das Sampling- oder Abtastintervall Δt so klein gewählt werden, dass die Nyquistfrequenz deutlich höher ist als die zu erwartende maximale Frequenz b. In unserem Fall beträgt die höchste zu detektierende Frequenz 40 Hz. Da zwei Abtastungen pro Periode benötigt werden, führt die Abtastung mit Δt=0.01 zu einer maximal zu detektierenden Frequenz von 50 Hz, und der interessante Bereich wird umfasst: $\Delta t = \frac{\pi}{2\pi 40} s = \frac{1}{80} s = 0.0125 > 0.01$

Weiter im Beispiel:

```
m = length(x);          % Anzahl der Datenpunkte
n = pow2(nextpow2(m));  % Länge der FFT, Anzahl der zu transfor-
                        % mierenden Punkte, Nullenauffüllung
y = fft(x,n);
f = (0:n-1)*(fs/n);     % Frequenzbereich
power = y.*conj(y)/n;   % Energieäquivalent (= abs(y).^2) bei den
                        % entsprechenden Frequenzen

% nextpow2 findet den nächsten Exponenten der Zweierpotenz, die
% größer oder gleich m ist
% und pow2 berechnet diese Zweierpotenz

% Um das Ergebnis zu veranschaulichen, sind Plots von abs(y),
% abs(y).^2, oder log(abs(y)) allgemein üblich. Ein Plot von
% Energie über der Frequenz heißt Periodogramm:

plot(f,power)
xlabel('Frequenz (Hz)')
ylabel('Energie')
title('{\bf Periodogramm}')
```

```
%Die Daten sind spiegelsymmetrisch zu 50 Hz und damit ist die
%erste Hälfte ausreichend zur Rekonstruktion. Es ist üblich, das
Periodogramm bei einer Frequenz "0" zu zentrieren, was durch
fftshift realisiert werden kann:


y0 = fftshift(y);           % Umordnen der Werte
f0 = (-n/2:n/2-1)*(fs/n);  % zentrierter Frequenzbereich
power0 = y0.*conj(y0)/n;   % zentrierte Energie
figure
plot(f0,power0)
xlabel('Frequenz (Hz)')
ylabel('Energie')
title('{\bf Zentriertes Periodogramm}')

%Analog kann man ein Phasendiagramm erstellen:

phase = unwrap(angle(y0));
figure
plot(f0,phase*180/pi)
xlabel('Frequenz (Hz)')
ylabel('Phasenwinkel (Grad)')
grid on

% Der Trend in der Zeichnung ist der unwrap-Funktion geschuldet,
% die verschieden oft 2*pi addiert bzw. subtrahiert.
```

unwrap(p) korrigiert den Phasenwinkel um $\pm 2\pi = \pm 360^o$, wenn die Winkel von 2 aufeinanderfolgenden Komponenten des Vektors p größer oder gleich der Standardtoleranz von 180 Grad sind.

(Fourier0.m)

Fourier2.m

%Filtern gegen Rauschen

Beispiel: Signalsynthese und –analyse

1. Kreation eines Beispielsignals:

```
t = 0:.001:.255;
x = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
plot(x(1:256))
y = x + 2*randn(size(t));
plot(y(1:256))
title('Verrauschtes Ausgangssignal')
```

2. Signalanalyse

```
Y = fft(y,256);
Pyy = Y.*conj(Y);  % Quadrat der Beträge der  Koeffizienten
                   % Energieäqivalent
f = 1000/256*(0:127);
plot(f,Pyy(1:128))
title('Spektraldichte')
xlabel('Frequenz (Hz)')
```

3. Zoom

```
plot(f(1:50),Pyy(1:50))
title('Spektraldichte')
xlabel('Frequenz (Hz)')
```

4. Synthese

```
z=ifft(Y);
t1=0:255;plot(t1,y,'r',t1,z,'b')
subplot(1,3,1),plot(t1,y),subplot(1,3,2),plot(t1,z),subplot(1,3,3),
plot(t1,y-z)
```

5. Denoising

```
for k=1:256; if abs(Y(k))<80; Y(k)=0;end;Y(k) ;end;
z1=ifft(Y);
plot(t1,y,'r',t1,z,'b',t1,z1, 'g');

figure

subplot(2,1,1),plot(t,x); subplot(2,1,2),plot(t1,z1, 'g');
```

Fourier3.m

```
% Blauwal
whaleFile=[matlabroot '/help/matlab/math/examples/bluewhale.au'];
[x,fs]=audioread(whaleFile);
plot(x)
xlabel('Sample Number')
ylabel('Amplitude')
title('{\bf Blue Whale Call}')
sound(x,fs)

% The B call is simpler and easier to analyze.
% Use the previous plot to determine approximate indices
% for the beginning and end of the first B call.
%Correct the time base for the factor of 10 speed-up in the data
bCall = x(2.45e4:3.10e4);
tb = 10*(0:1/fs:(length(bCall)-1)/fs); % Time base

plot(tb,bCall)
xlim([0 tb(end)])
xlabel('Time (seconds)')
ylabel('Amplitude')
title('{\bf Blue Whale B Call}')

% Use fft to compute the DFT of the signal.
% Correct the frequency range for the factor of 10 speed-up in
the data
m = length(bCall);      % Window length
n = pow2(nextpow2(m));  % Transform length
y = fft(bCall,n);       % DFT of signal
f = (0:n-1)*(fs/n)/10;  % Frequency range
p = y.*conj(y)/n;       % Power of the DFT

%Plot the first half of the periodogram, up to the Nyquist
frequency:

plot(f(1:floor(n/2)),p(1:floor(n/2)))
xlabel('Frequency (Hz)')
ylabel('Power')
set(gca,'XTick',[0 50 100 150 200]);
title('{\bf Component Frequencies of a Blue Whale B Call}')
% The B call is composed of a fundamental frequency
% around 17Hz and a sequence of harmonics mit der 2.
dominierenden Harmonischen.
```

Fourier1.m

```
% Abtastung der Funktion entsprechend Bandbegrenzung
% Zeichnung der Funktion, der abgetasteten Werte
% und der rücktransformierten Werte
% Ausgabe von abs(ck)
% Ausgabe der Differenzen von Hin und Rück

t=-10:0.01:10;
y=exp(-0.1.*t.^2).*(2*sin(t)+6*cos(3*t));%
plot (t,y,'b')
hold on

b=2*pi; %Bandbegrenzung?
dt=pi/b;
ta=-10:dt:10;
ya=exp(-0.1.*ta.^2).*(2*sin(ta)+6*cos(3*ta));%
L=length(ya)
Nfft=2^nextpow2(L)
c=fft(ya,Nfft);
yn=ifft(c,Nfft);
yp=interp1(ta,yn(1:L),t,'spline');
plot(ta,ya,'ro',t,yp,'k')

ck=c/Nfft;
figure
kf=linspace(0,1,Nfft/2+1)*b;
plot(kf,2*abs(ck(1:Nfft/2+1)).^2)

figure
plot(1:L,abs(yn(1:L)-ya))
```

Fourier2.m

```
Fs=1000;
T=1/Fs;
L=1000;
t=(0:L-1)*T;
x=0.7*sin(2*pi*50*t)+sin(2*pi*120*t);
y=x+2*randn(size(t));
plot(Fs*t(1:100),y(1:100))
title('Verrauschtes Signal');
xlabel('Zeit in ms');
Nfft=2^nextpow2(L);
c=fft(y,Nfft)/L;
f=Fs/2*linspace(0,1,Nfft/2+1);
figure
plot(f,2*abs(c(1:Nfft/2+1)))
title('Einseitiges Amplitudenspektrum von y(t)')
xlabel('Frequenz in Hz')
ylabel('|c(f)|')
```