Matlab Workshop WS 2013/14

Erik Poller   Cedric Patzer

January 6, 2014

Inhalt

Einführendes

[Matlab] $\Rightarrow$ Matrix Laboratory

- Metasprache zum wissenschaftlichen (numerischen) Rechnen
- System von freien und kommerziellen Toolboxen
- Funktionen zur Visualisierung von Modellen und Daten
- Erstellung von GUIs
- Compiler zum Erstellen von Lauffähigen Programmen

Grundstrukturen

Hilfe

» help	Liste aller Hilfethemen
» help topic	Hilfe zu Themen
» help function	Hilfe zu Funktionen

Kommandozeile

» a = 4*3	Ausgabe des Ergebnisses
» a = 4*3;	Unterdrückung der Ausgabe

Vektoren und Matrizen I

Vektoren

» a=[1 2 3];	Zeilenvektor
» b=[1;2;3];	Spaltenvektor
» b.'	Transponieren
» c = a*b	Skalarprodukt(Zeilen*Spaltenvektor)
» d=b*a	Vektormultiplikation
» e=b.*a	Punktweise Multiplikation

Matrizen

» A=[1 2 3; 4 5 6; 7 8 9];	Matrix
» x=A*b;	Matrix-Vektor-Multiplikation
» A\ b	Direkter Gleichungslöser nach Gauß-Newton-Eliminationsverfahren
» A*A	Matrix Multiplikation

Vektoren und Matrizen II

Indizierung Vektoren

» a=1:10	Erzeugt Zeilenvektor mit Einträgen 1-10
» a(3)	Greift auf 3. Eintrag in a zu
» a(3:7)	Greift auf Elemente 3 bis 7 zu
» a(3:end)	Greift vom 3. bis zum letzten Element zu

Indizierung Matrizen

```
» A=[1 2 3; 4 5 6; 0 2 0]
» B=A+1
» B(2,3)
» B(2:3,1:2)
» B(3,:)
» B(:)
```

Nützliche Befehle

» size(A)	Größe einer Matrix
» length(A)	Länge eines Vektors
» max(A), min(A)	Maxiumum,Minimum
» find(a)	Finden von Elementen
» sort(a)	Sortieren
» diff(a)	Differenz
» zeros(m,n)	erzeugt mxn Matrix aus Nullen
» ones(m,n)	erzeugt mxn Matrix aus Einsen
» eye(m,n)	erzeugt mxn Einheitsmatrix
» linspace(a,b,n)	n äquidistante Werte auf [a,b]
» logspace(a,b,n)	logarithmisch äquidistante W. auf $[10^a, 10^b]$

Graphik Befehle

Darstellung von Kurven

» plot(x,y)	Kurve
» plot(x,y,'r+')	rote Kreuze
» plot(x,y,x2,y2)	zwei Kurven
» xlabel,ylabel	Achsenbeschriftung
» title	Grafiküberschrift
» semilogx, semilogy, loglog	Logarithmische Achsen
» subplot	mehrere Grafiken in einem Figure
» plot3(x,y,z)	Kurve in 3D

Darstellung von Flächen

» imagesc(x,y,z)	x,y..Koordinaten, Z..Matrix
» contour(x,y,z)	Isolinien
» surf	erhöhte Oberfläche Farbcodiert

Plot einer gedämpften Schwingung

```
> x = 0:0.1:4*pi;
> y1 = sin(x);
> y3 = exp(-x/3);
> y4 = y3.*y1;
> figure()
> plot(x,y4);
> xlabel x
> ylabel y
> title('y4 = e^{-x}...
> * sin(x)');
```

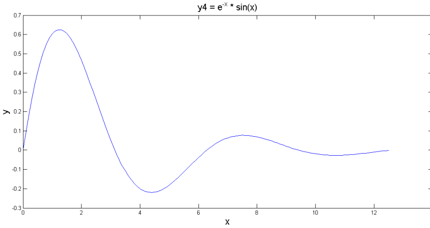


Figure: Entsprechendes Bild

Plot einer Spirale

Gegeben

$0 \leq s \leq 10\pi, \; inkr = 0.1$

$x = \cos(s)$

$y = \sin(s)$

$-5 \leq z \leq 5$

Aufgabe

Stelle die oben definierte Kurve mit dem Befehl >plot3()< dar. Die Kurve soll als rote Linie (Stärke 2pt) mit Dreiecken geplottet werden.

Plot EM-34-Daten Tharandter Wald 27.04.2012

Gegeben

Gegeben sind die Datein *EM_34_broadside.txt* und *EM_34_inline.txt*. Download

Aufgabe

Lade mit Hilfe von >var = load(pfad)< die .txt-files in Matlab ein. Stelle beide in einem Diagramm mit Legende dar!

Eigene Funktionen

Allgemeines

- Funktionsname = Dateiname
- am Ende steht >return<
- Hilfsvariablen in der Funktion können gleich benannt sein, wie im Hauptscript, dies hat keinen Einfluss
- Alle in der Funktion belegten Variablen werden nach Beendigung der Funktion gelöscht, außer die Variablen out1...outn

Beispiel

```
function [out1, out2, ...] = fcn_name(in1, in2, ...)
% Hier passiert dann das, was die Funktion machen soll
return
```

Mathematische Beispiele

Problemstellung

$$
\begin{aligned}
a_{11} \cdot x + a_{12} \cdot y + a_{13} \cdot z &= b_1 \\
a_{21} \cdot x + a_{22} \cdot y + a_{23} \cdot z &= b_2 \\
a_{31} \cdot x + a_{32} \cdot y + a_{33} \cdot z &= b_3
\end{aligned}
$$

-> in Matrixschreibweise:

$$\mathbf{A}\vec{x} = \vec{b}, \text{ mit}$$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}; \quad \vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}; \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Grundlagen

Wann ist ein LGS lösbar? *Festlegung: Wir betrachten nur quadratische Systeme*

Grundlagen

Wann ist ein LGS lösbar? *Festlegung: Wir betrachten nur quadratische Systeme*

- $\det(\mathbf{A}) \neq 0$

Grundlagen

Wann ist ein LGS lösbar? *Festlegung: Wir betrachten nur quadratische Systeme*

- $\det(\mathbf{A}) \neq 0$
- alternativ: $\text{Rang}(\mathbf{A}) = \text{Anzahl der Zeilen}$

Grundlagen

Wann ist ein LGS lösbar? *Festlegung: Wir betrachten nur quadratische Systeme*

- $\det(\mathbf{A}) \neq 0$
- alternativ: $\text{Rang}(\mathbf{A}) = \text{Anzahl der Zeilen}$
- $\vec{x}$ und $\vec{b}$ sind Spaltenvektoren

Grundlagen

numerische Gleichungslöser(vgl. Numerik I)

- L-U-Zerlegung
- Gauß-Elimination, ohne Pivotisierung
- Gauß-Elimination, mit Pivotisierung

Matlabvariante

$$\vec{x} = \mathbf{A} \backslash \vec{b}$$

("Hexenwerk")

Übungsaufgabe

Aufgabe 1a)

Schreibe eine MATLAB-Funktion mit dem Namen >solve_lgs< zum Lösen eines linearen Gleichungssystems. In die Funktion sollen übergeben werden:

- Koeffizientenmatrix **A**
- Datenvektor $\vec{b}$

Ausgegeben werden soll:

- Lösungsvektor $\vec{x}$

Aufgabe 1b)

Überprüfe die geschriebene Funktion mit der

Koeffizientenmatrix $A = \begin{pmatrix} 5 & -2 & 3 \\ 2 & 2 & -4 \\ -2 & 3 & 1 \end{pmatrix}$ und $b = \begin{pmatrix} 19 \\ -6 \\ -12 \end{pmatrix}$

Übungsaufgabe

Aufgabe 2)

Löse die folgenden Gleichungssysteme

- $A = \begin{pmatrix} 2 & -3 \\ 5 & 2 \end{pmatrix}$ $b = \begin{pmatrix} 12 \\ 11 \end{pmatrix}$
- $A = \begin{pmatrix} -3 & 5 \\ 3 & -9 \end{pmatrix}$ $b = \begin{pmatrix} -16 \\ 24 \end{pmatrix}$
- $A = \begin{pmatrix} 5 & -4 & 6 \\ 2 & 3 & -2 \\ 9 & 2 & 2 \end{pmatrix}$ $b = \begin{pmatrix} 7 \\ 4 \\ 12 \end{pmatrix}$

Nullstellensuche

Die Funktion fzero

- Matlab stellt eine Funktion fzero zur Nullstellen suchen zur Verfügung
- $Nullstelle = fzero('x^2 - 1', 0)$
- **Argument 1:** die Funktion selbst als String
- **Argument 2:** Ein Punkt um den die Nullstele gesucht werden soll. Vorsicht bei der Wahl!!

Interpolation

- Gegeben sind Daten, z.B Zeitreihe zu Diskreten Zeitpunkten.
- Gesucht ist glatte Kurve, zum Berechnen von Werten zwischen den Stützstellen
- Mögliche Anwendung in der magnetischen Variationskorrektur, oder Driftkorrektur in der Gravimetrie

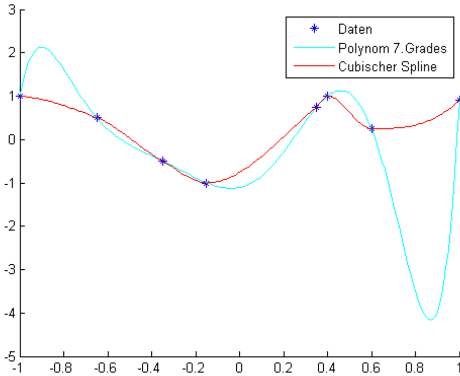


Figure: Interpolation

Grundlagen Lineare Regression

Problemstellung

- Finde eine Gerade, die Punktwolke möglichst exakt beschreibt.
- Daten $\vec{x}$,$\vec{y}$ gegeben
- Gesucht Lineare Funktion: $y = mx + n$, die Daten gut erklärt.
- Problem mehr Gleichungen als Unbekannte -> Überbestimmtes Gleichungssystem
- Exakte Lösung nicht vorhanden

Problemstellung

$$y_1 = m \cdot x_1 + n$$
$$y_2 = m \cdot x_2 + n$$
$$y_3 = m \cdot x_3 + n$$

In Matrixschreibweise $\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} m \\ n \end{pmatrix}}_{\vec{m}}$

Numerische Lösung

Numerische Ausgleichsrechnung vgl. Numerik I

- Normalgleichung
- QR-Zerlegung
- Housholder Transformation
- Givens Rotation

Matlab

$$x = A\backslash b$$

Hexenwerk -> \ Operator löst nicht nur Lineare Gleichungssysteme, sondern auch Ausgleichsprobleme **OHNE** eine Warnung auszugeben!!!!

Übungsaufgabe

Aufgabe 1a)

Bestimme für die Punkte (0,0),(1.2,0.8) und (2,1.8) die Ausgleichsgerade mithilfe des GUI und stelle diese graphisch dar.

Aufgabe 1b)

Datei 'file3_daten.txt' entält eine Punktwolke zu der eine Lineare Regression analog zu Aufgabe 1b) auszuführen ist.

Problemstellung

$$
\begin{aligned}
y_1 &= a_n \cdot x_1^n + a_{n-1} \cdot x_1^{n-1} + \cdots + a_1 \cdot x_1 + a_0 \\
y_2 &= a_n \cdot x_2^n + a_{n-1} \cdot x_2^{n-1} + \cdots + a_1 \cdot x_2 + a_0 \\
y_3 &= a_n \cdot x_3^n + a_{n-1} \cdot x_3^{n-1} + \cdots + a_1 \cdot x_3 + a_0
\end{aligned}
$$

In Matrixschreibweise

$$
\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \cdots & x_2 & 1 \\ x_3^n & x_3^{n-1} & \cdots & x_3 & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_1 \\ a_0 \end{pmatrix}}_{\vec{m}}
$$

Bewerkstelligung mit Matlab

Die Funktion polyfit und polyval

Matlab stellt die Funktion Polyfit zur Verfügung, die eine Polynomapproximation berechnet, sowie die Funktion Polyval, die die Ausgabe von Polyfit weiterverarbeitet.

Übungsaufgabe

Die Datei 'Daten.mat' enthält Daten mit quadratischen Zusammengang, und einem statistischen Fehler. Finde ein quadratisches Polynom, welches die Daten möglichst gut erklärt. Stelle sowohl die Punkte, als auch die Ausgleichskurve in einem Plot graphisch dar
Hinweis: Die Daten liegen in einem eigenen Matlab Datenformat '.mat' vor. Zum einlesen 'load Daten.mat'

Übungsaufgabe

Aufgabe2

Finde ein Polynom, dass die Daten aus der Datei 'Aufgabe2.mat' möglichst gut beschreibt. Stelle wieder sowohl die Punkte als auch das Ausgleichspolynom graphisch in einem Plot dar. Was fällt dir auf?

Geophysikalische Beispiele

Vorbereitung

Was muss ich bei der Auswertung von (Magnetik-)Daten beachten?

- Was für Daten habe ich vorliegen (GPS, Totalintensität, Variationskurve,...)

Vorbereitung

Was muss ich bei der Auswertung von (Magnetik-)Daten beachten?

- Was für Daten habe ich vorliegen (GPS, Totalintensität, Variationskurve,...)
- In welchem Format liegen diese vor? (gibt es Textzeilen, kann ich die Datei einfach mit load einlesen?)

Vorbereitung

Was muss ich bei der Auswertung von (Magnetik-)Daten beachten?

- Was für Daten habe ich vorliegen (GPS, Totalintensität, Variationskurve,...)
- In welchem Format liegen diese vor? (gibt es Textzeilen, kann ich die Datei einfach mit load einlesen?)
- Welche Bearbeitungsschritte muss ich machen um die Daten richtig und interpetierbar darzustellen?

Zu beachten!

GPS-Daten

- Kontrolle ob alle Daten im geichen Format?
- (ggf. Umrechnung der Daten)
- Alle Koordinaten auf einen Punkt beziehen, damit die Werte kleiner werden

Zu beachten!

Magnetik-Daten

- Variationskorrektur
  1. Korrektur Darstellen
  2. Durch Polynom approximieren
  3. Werte korrigieren
- Regionalfeldeinflüsse?
- Polreduktion
- Bezug auf ein T_0

Das haben wir

Magnetik.zip

- Zu finden im Downloadbereich
  - GPS-Daten
  - KPM-Daten
  - Variometer-Daten

Das haben wir

Magnetik.zip

- Zu finden im Downloadbereich
  - GPS-Daten
  - KPM-Daten
  - Variometer-Daten
- Datei *VarST_25_04.txt* beinhaltet Variometerdaten

Das haben wir

Magnetik.zip

- Zu finden im Downloadbereich
  - GPS-Daten
  - KPM-Daten
  - Variometer-Daten
- Datei *VarST_25_04.txt* beinhaltet Variometerdaten
- *2504_KPMmitGPS.txt* beinhaltet die KPM Daten inklusive GPS Werten

Generelle Notwendige Arbeitsschritte

1. Zuordnung GPS-Werte → Magnetikwerte
2. Entfernen von Fehlwerten
3. Daten so formatieren, dass diese gut eingelesen werden können
4. Korrekturen

Für Schritt 1+2 sind Tabellenkalkulationsprogramme am besten geeignet.

Darstellung erster Werte

Aufgabe 1: Variationskurve

Lade die Datei *VarST_25_04.txt* in Matlab ein und stelle die Variationskurve grafisch dar.

Aufgabe2: Variationskorrektur

Lade die Datei *2504_KPMmitGPS.txt* ein und führe eine Variationskorrektur mithilfe der Variometer Daten aus Aufgabe 1 durch. Dazu ist die Funktion interp1 geeignet.

Aufgabe 3: Lage der KPM Messpunkte

Stelle die Hoch- und Rechtswerte der KPM Messpunkte als blaue Kreuze in einem Plot dar.

Darstellung der Magnetikdaten

Notwendige Arbeitsschritte

1. Erzeugen eines Gitters, auf dem die Messwerte interpoliert werden
2. Erstellen eines Interpolanten
3. Interpolation der Messwerte auf das in 1 erzeugte Gitter
4. Darstellung der Werte

1) Erzeugen eines Gitters, auf dem die Messwerte interpoliert werden

Notwendige Funktion: **meshgrid**:
$[XI, YI] = \text{meshgrid}(xi, yi)$; mit:
$nx = \text{length}(xi)$; $ny = \text{length}(yi)$;
$XI,\ YI = nx \times ny$

$$XI = \begin{pmatrix} x_1 & x_2 & \dots & x_{nx} \\ \vdots & & & \vdots \\ x_1 & x_2 & \dots & x_{nx} \end{pmatrix} \rightarrow ny \text{ Kopien von Vektor } xi$$

$$YI = \begin{pmatrix} y_1 & \dots & y_1 \\ \vdots & \ddots & \vdots \\ y_{ny} & \dots & y_{ny} \end{pmatrix} \rightarrow nx \text{ Kopien von Vektor } yi$$

$\rightarrow$ Ermöglicht 2D-Funktion darzustellen.
xi, yi können beliebige Vektoren in der Größenordnung der Datenvektoren sein! $\text{linspace}(\min(x), \max(x))$ bietet sich an.

2) Erstellen eines Interpolanten

Ausgangspunkt

- Punkte x,y mit zugehörigen variationskorrigierten Messwerten
- zugehöriges Gitter für x,y

Ziel

- Reguläres Gitter XI,YI mit Messwerten auf diesem Gitter

Lösung: Funktion **TriScatteredInterp**

2) Erstellen eines Interpolanten

$F = \text{TriScatteredInterp}(x, y, z);$

- Erzeugt aus den Daten $(x, y, z) = (RW, HW, dT)$ einen Interpolant F zur räumlichen beschreibung der Daten.
- F kann wie eine Funktion behandelt werden. $dT' = F(x', y')$, mit beliebigen Punkt (x', y') aus $\min(x) \leq x' \leq \max(x)$, $\min(y) \leq y' \leq \max(y)$
- Funktioniert auch mit Matrizen...

3) Interpolation der Messwerte auf das in 1 erzeugte Gitter

bis jetzt geschafft:

- Erzeugen eines Gitters auf Basis von RW, HW
- Erzeugen eines Interpolants auf Basis von RW, HW, dT

Interpolation der Werte dT auf das erstellte Gitter

- $DT = F(XI, YI);$
- Jetzt 3 Matrizen: $XI, YI, DT \rightarrow$ Darstellen der Werte jetzt möglich!

Flächenhafte Darstellung

» surf(XI,YI,DT)	3D-Darstellung mit gefüllten Flächen
» contour(XI,YI,DT)	erzeugt Isoliniendarstellung
» contourf(XI,YI,DT)	erzeugt Isoliniendarstellung, Flächen dazwischen gefüllt

Für Magnetikdaten bietet sich die Darstellung mit contourf an.

Darstellung der Magnetikwerte

Aufgabe6

Stelle die Magnetikwerte aus *2504_KPMmitGPS.txt* dar! Nutze dafür ein Gitter mit einer geeigneten Anzahl (x, y)-Werten in konstantem Abstand.Welche Farbe entspricht welchen Werten? Erstelle eine Darstellung mit und eine ohne den Messpunkten in der Magnetikdarstellung.

Zusatz

In der Magnetik werden normalerweise positive Anomalien rot, negative Anomalien blau und ungestörte Gebiete weiß dargestellt. Nutze dazu das M-file *bluewhitered.m*.Positioniere die Farbskala unterhalb des Bildes und beschrifte diese lesbar!

Darstellung der AZM Daten

Aufgabe 7

Nutze das entstande Skript um auch die AZM Daten aus der Datei *Treppenhauer_AZM_alles.txt* darzustellen.

Sondierungskurve

Was haben wir?

- Zugrundeliegende Messanordnung: Schlumberger
- Abstände der Potentialelektroden A,B
- ρ_a scheinbare spezifische Widerstände des korrespondierenden Halbraumes

Darstellung

In einer Sondierungskurve wird ρ_a über $\bar{AB}/2$ auf doppelt logarithmischer Skala dargestellt. Dazu ist die Funktion *loglog* hilfreich.

dcfwdf

Die Berechnung des scheinbaren spezifischen Widerstandes erfolgt mithilfe der Funktion *dcfwdf*. (Zu finden im Downloadbereich)

- rhoa = dcfwdf(rho,d,AB)
- rhoa - scheinbare spezifische Widerstand
- rho - Schichtwiderstände
- d - Schichtmächtigkeiten
- AB - Elektrodenabstände

Aufgaben

Aufgabe 1

Lade die Datei *DreiSchicht.dat* ein und stelle die Daten in einer Sondierungskurve dar.

Aufgabe 2

Nimm ein geeignetes Startmodell und berechne synthetische Daten mithilfe der Funktion *dcfwdf*. Stelle die Ergebnisse aus Aufgabe 1 und 2 in einem gemeinsamen plot dar.

Aufgaben Fortsetzung

Aufgabe 3

Verändere das Startmodell und versuche die Sondierungskurve möglichst exakt anzupassen.

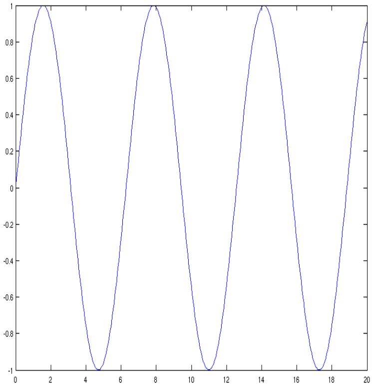
Aufgabe 4

Die Datei *Hundewiese.dat* enthält reale Messdaten aus Freiberg. Wiederhole Aufgabe 1 - 3 mit dem neuen Datensatz.

Superpositionsprinzip

Superposition von 2 Schwingungen $f1$ und $f2$

- $f_1(t) = A_1 \sin \omega t$

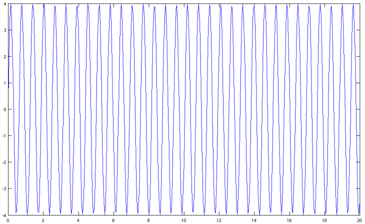


Typischerweise stammen Frequenzen von 50 bzw 16 2/3 Hz von externen unerwünschten Quellen. **Wie kann man diese in einem Signal erkennen und entfernen?**

Superpositionsprinzip

Superposition von 2 Schwingungen $f1$ und $f2$

- $f_1(t) = A_1 \sin \omega t$
- $f_2(t) = A_2 \sin \omega_2 t$



Typischerweise stammen Frequenzen von 50 bzw 16 2/3 Hz von externen unerwünschten Quellen.
Wie kann man diese in einem Signal erkennen und entfernen?

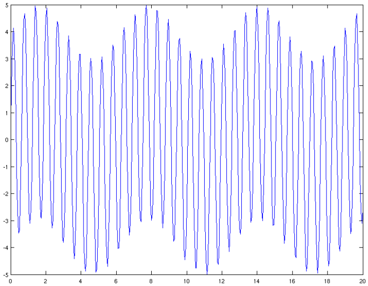
Superpositionsprinzip

Superposition von 2 Schwingungen $f1$ und $f2$

- $f_1(t) = A_1 \sin \omega t$
- $f_2(t) = A_2 \sin \omega_2 t$
- $f_3(t) = f_1 + f_2$

Typischerweise stammen Frequenzen von 50 bzw 16 2/3 Hz von externen unerwünschten Quellen.

Wie kann man diese in einem Signal erkennen und entfernen?



Diskrete Signale

» Kontinuierliche Signale $f(t)$ an diskreten Zeitpunkten t_i mit einer Samplingrate dt abtasten.
» Je kleiner dt desto genauere Auflösung, bei zunehmenden Speicherverbrauch
» Nyquistfrequenz gibt Auskunft über die kleinst möglich wählbare Abtastrate

$$f_{ny} = 2 \cdot f_{max} \leq \frac{1}{dt}$$

Übergang vom Zeit in den Frequenzbereich

Fouriertransformation

Transformation von Zeit in den Frequenzbereich

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt$$

Rücktransformation von Frequenz in Zeitbereich

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} F(\omega) e^{i\omega t} d\omega$$

Wird in Matlab mithilfe von *fft* und *ifft* bewerkstelligt.

Filter

- $H = 0$ im Sperrbereich
- $H = 1$ im Durchlassbereich
- $F_{filtered} = F_3(\omega) \cdot H(\omega)$
- Rücktransformation in den Zeitbreich ergibt Ausgangsignal ohne Störfrequenzen

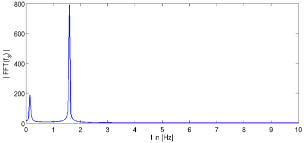


Figure: FFT von f_3

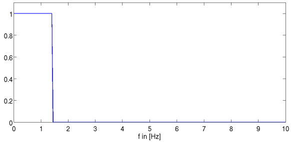


Figure: Filter zum herausschneiden von Frequenzen > 1.5 Hz

Aufgaben

Aufgabe 1

Lade die Datei *Aufgabe1.mat* ein. Stelle die darin enthaltene Zeitreihe dar. Berechne danach die Fouriertransformation und stelle beide Kurven in einem gemeinsamen subplot dar. Was ist bei der numerischen FFT zu beachten?

Aufgabe 2

Auf dem gemessen Signal liegt eine Stör Frequenz 50 Hz. Baue einen Filter der in der Lage ist diese zu unterdrÃ¼cken. Und stelle das Ausgangsignal und das gefilterte Signal zusammen in Subplots dar.

Aufgabe 3

Fahre analog zu den Aufgaben 1 und 2 mit den Daten aus der Datei *Aufgabe3.mat* vor.

Strahlenseismik

Snelliussches Brechungsgesetz

Der Brechungswinkel einer ebenen Welle berechnet sich nach:

$$\sin(\beta) = \frac{c_2}{c_1} \sin(\alpha) \tag{1}$$

mit Brechungswinkel β, Einfallswinkel α, Geschwindigkeit der oberen Schicht c_1, Geschwindigkeit der unteren Schicht c_2

Grenzwinkel der Totalrefelxion

Bei Einfallswinkeln größer als:

$$\alpha > \arcsin(\frac{c_1}{c_2}) \tag{2}$$

tritt keine Brechung mehr auf, stattdessen wird die gesammte Wellenenergie reflektiert

Aufgaben

Aufgabe1

Eine Ebene Welle trifft auf einen ebenen Reflektor unter einem Winkel von 45°. Die Geschwindigkeiten betragen 3000 und 3500 m/s. Die Schichtmächtigkeit der ersten Schicht beträgt 200m. Berechne die Koordinaten x eines Strahles in den Tiefen 200 und 700m.

Aufgabe2

Fahre Analog zu Aufgabe 1 mit drei Schichtgeschwindigkeiten von 6000, 7000 und 8000 m/s sowie Mächtigkeiten der ersten beiden Schichten von jeweils 100m vor. Stelle den Strahlverlauf grafisch dar.

Aufgaben Teil 2

Aufgabe3

Fahre analog zu Aufgabe 1 und 2 für ein kontinuierliches Geschwindigkeitsmodell der Form

$$v(z) = 1000\mathrm{m/s} + 1,45\mathrm{s}^{-2} \cdot z \qquad (3)$$

vor. Approximiere dazu das Modell durch eine beliebige Anzahl an Schichten konstanter Geschwindigkeit. Breche die Berechnung ab, sobald Totalreflexion einsetzt.

Aufgabe4

Erweitere den Code aus Aufgabe 3 in dem die Lösung an der grenze zur Totalreflexion gespiegelt wird

Nützliche Mathematische Funktionen

» sin/sind	Sinus Grad/Bogenmaß
» cos/cosd	Cosinus Grad/Bogenmaß
» eig	Eigenwerte und Vektoren
» det	Determinante
» lu	LU Zerlegung
» cross	Kreuzprodukt
» svd	Singulärwertzerlegung
» inv	Inverse einer Matrix
» rank	Rang einer Matrix
» trace	Spur
» fft	Fouriertransformation
» conv	Faltung