

Flexible Positionierung von seismischen Quellen bei der FD-Modellierung

Bachelorarbeit
von

Sophia Götz

am Geophysikalisches Institut (GPI)

Referent: Prof. Dr. Thomas Bohlen
Korreferent: Dr. Thomas Hertweck

Bearbeitungszeit: 01.01.2025 – 31.03.2025

Erklärung zur Selbstständigkeit

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den 31.03.2025,

S. 

Sophia Götz

Abstrakt

In dieser Arbeit wird die 2D-Finite-Differenzen-Modellierung auf einem geschachtelten Gitter und die Möglichkeiten der flexiblen Positionierung von Quellen zwischen Gitterknoten, sowie deren Genauigkeit im Vergleich zu anderen auftretenden Unsicherheiten betrachtet. Die Berechnungen erfolgen anhand eines homogenes und elastischen Modells und werden anschließend mit einer analytisch berechneten Referenzlösung verglichen. Nach einem Konvergenztest, folgt eine Dispersionskorrektur, da dies der größte Verursacher von Abweichungen von der Referenzlösung ist und somit sonst kein Vergleich verschiedener Quellinterpolationsmethoden möglich ist. Der Vergleich der Sinc-Funktion mit der FD-konsistenten Methode ergibt nur geringfügige Unterschiede, die in einem komplizierteren Fall, in dem die Dispersionskorrektur nicht einfach möglich ist, kaum noch relevanten Einfluss auf die Ergebnisse hat.

Abstract

This Bachelor thesis looks at the 2D Finite Difference modelling method on a staggered grid and some methods of implementing a source in between grid nodes. Then their uncertainties get compared to each other and different influencing factors. The calculations are based on a homogenous elastic model and are compared to an analytical reference solution. Following a test of the convergence of the method, a dispersion correction will be applied as numerical dispersion is the largest fault factor and hinders the comparison of the sinc- and FD-consistent source weight functions. The result is only a small difference, which might become irrelevant in a more complicated case where dispersion correction is not as easy as here.

Inhaltsverzeichnis

1. Einleitung	1
2. Theoretische Grundlagen	2
2.1. 2D-Finite-Differenzen-Methode	2
2.2. Quellimplementation	4
2.3. Sinc-Quellgewichtsfunktion	4
2.4. FD-konsistente Quellgewichtsfunktion	5
2.5. Analytische Referenzlösung	5
2.6. L2-Fehler	5
3. Modell und Geometrie	7
4. Konvergenztest	8
5. Dispersionskorrektur	11
6. Flexible Quellenpositionierung	14
7. Schlussfolgerung	17
Anhang	18
A. Zusätzliche Abbildungen	18
A.1. Konvergenztest	18
A.2. Dispersionskorrektur	19
A.3. Flexible Positionierung der Quelle	19
B. Matlab Skripte	26
Literaturverzeichnis	46

Abbildungsverzeichnis

2.1. Definition der elastischen Wellenfeldparameter auf einem verschachtelten Gitter. Schwarze Symbole stehen für Spannungen bei $t = (n + 0,5)\Delta t$ und graue Symbole für Geschwindigkeiten bei $t = n\Delta t$	3
4.1. L2-Fehlers des Vergleiches der im Konvergenztest mit schrittweise verringelter Diskretisierung berechneter Seismogramme der Ordnung $M = 1$ mit den analytisch berechneten Referenzlösungen. Die Seismogramme des Druckes P werden als Punkte, die v_x -Seismogramme als + und die v_z -Seismogramme als x dargestellt. Schwarze Symbole wurden mit der Standarddiskretisierung, blaue mit halbiert und rote mit geviertelter Schrittweite berechnet.	9
4.2. L2-Fehlers des Vergleiches der im Konvergenztest mit schrittweise verringelter Diskretisierung berechneter Seismogramme der Ordnung $M = 2$ mit den analytisch berechneten Referenzlösungen. Die Seismogramme des Druckes P werden als Punkte, die v_x -Seismogramme als + und die v_z -Seismogramme als x dargestellt. Schwarze Symbole wurden mit der Standarddiskretisierung, blaue mit halbiert und rote mit geviertelter Schrittweite berechnet.	9
4.3. Darstellung der für den Konvergenztest der Ordnung $M = 2$ berechneten v_x -Seismogramme im Vergleich mit der schwarz-gestrichelten analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbiert Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	10
5.1. Geschwindigkeitsänderung durch die numerische Dispersion bei $M = 1$ über $k\Delta h$	12
5.2. L2-Fehler der Druck-Seismogramme mit $M = 2, ax = 0, az = 0$ mit(rot) und ohne(schwarz) Dispersionskorrektur	12
5.3. L2-Fehler der Geschwindigkeits-Seismogramme mit $M = 2, ax = 0, az = 0$ mit und ohne Dispersionskorrektur für die S-Welle als + bzw. Punkt. v_z in schwarz, v_x in rot	13
6.1. L2 Fehler der Druck Seismogramme für Quellen auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	15
6.2. L2 Fehler der P-Wellen Seismogramme für Kraftquellen auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	15
6.3. L2 Fehler der S-Wellen Seismogramme für Kraftquellen auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	16
A.1. Für den Konvergenztest der Ordnung $M = 1$ berechneten Druck-Seismogramm des in z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbiert Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	18

A.2. Für den Konvergenztest der Ordnung $M = 1$ berechneten Druck-Seismogramm eines zwischen x- und z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	18
A.3. Darstellung der für den Konvergenztest der Ordnung $M = 1$ berechneten v_z -Seismogramme als Differenz mit der analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	19
A.4. Darstellung der für den Konvergenztest der Ordnung $M = 2$ berechneten Druck-Seismogramme im Vergleich mit der analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	20
A.5. Darstellung der für den Konvergenztest der Ordnung $M = 2$ berechneten v_x -Seismogramme im Vergleich mit der schwarz gestrichelten analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	20
A.6. Für den Konvergenztest der Ordnung $M = 2$ berechneten v_z -Seismogramm des in z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	21
A.7. Für den Konvergenztest der Ordnung $M = 2$ berechneten v_z -Seismogramm eines zwischen x- und z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.	21
A.8. L2-Fehler der Geschwindigkeits-Seismogramme mit $M = 2, ax = 0, az = 0$ mit und ohne Dispersionskorrektur für die S-Welle als + bzw. Punkt. v_z in schwarz, v_x in rot	22
A.9. L2 Fehler der Druck-Seismogramme mit $M = 1$ für verschiedene versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	22
A.10.L2 Fehler der P-Wellen-Seismogramme einer Kraftquelle mit $M = 1$ für verschiedene versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	22
A.11.L2 Fehler der S-Wellen-Seismogramme mit $M = 1$ für verschiedene versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	23
A.12.L2 Fehler der Druck-Seismogramme auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion bei $M = 2$	23
A.13.L2 Fehler der P-Wellen-Seismogramme einer Kraftquelle auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion bei $M = 2$	23
A.14.L2 Fehler der S-Wellen-Seismogramme auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion bei $M = 2$	24
A.15.L2-Fehler der Druck-Seismogramme mit $M = 2$ für verschiedene versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	24
A.16.L2-Fehler der P-Wellen-Seismogramme mit $M = 2$ für verschiedene versetzte Kraftquellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	24

A.17.L2 Fehler der S-Wellen-Seismogramme mit $M = 2$ für verschiedene versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion	25
--	----

Tabellenverzeichnis

2.1. Taylor-Koeffizienten für die FD-Modellierung verschiedene Ordnungen M berechnet nach Gleichung 2.7	3
---	---

1. Einleitung

Die Modellierung Seismischer Wellen ist ein wichtiger Teil der Geophysik und wird in vielen Aufgaben angewendet. Die Finite-Differenzen(FD)-Methode ist eine häufig angewendete Methode, um Wellenfelder und Seismogramme zu berechnen. Auch diese wird stetig weiterentwickelt und optimiert, um immer genauere Ergebnisse zu erlangen.

Ein Ansatz zur Verbesserung ist die flexible Platzierung von Quellen außerhalb des Modellgitters, mit dem Anspruch trotzdem ähnlich gute Ergebnisse zu ermöglichen. Es gibt viele Veröffentlichungen, die verschiedene Methoden und Vorgehensweisen vorstellen. Hier wird nur die relativ einfache Möglichkeit der Interpolation mithilfe einer sinc-Funktion nach Hicks (2002) und die etwas komplexere FD-konsistente-Methode von Koene et al. (2020) betrachtet.

Um allerdings einen Vergleich der Methoden und eine Einschätzung zur gegebenen Genauigkeit machen zu können, ist es nötig zuerst andere Fehlerquellen zu minimieren. Der Vergleich erfolgt mit einer analytische mithilfe von Greenschen Funktionen berechneten Referenzlösung. Zum Kontrollieren des Programmcodes wird ein Konvergenztest durchgeführt. Anschließend wird mit einer Dispersionkorrektur der Einfluss der zeitlichen und räumlichen numerischen Dispersion möglichst herausgerechnet, um überhaupt Genauigkeiten zu erreichen, in denen Unterschiede der Quellimplementierungsmethoden erkennbar und relevant werden. Die Berechnungen erfolgen anhand eines einfachen, homogenen, elastischen Models mit der 2D-FD-Methode auf einem geschachtelten Gitter mithilfe von Matlab.

2. Theoretische Grundlagen

2.1. 2D-Finite-Differenzen-Methode

Die Modellierung der Seismischen Wellen und die Berechnung der Seismogramme erfolgt mit der 2D-Finite-Differenzen-Methode auf einem geschachteltem Gitter nach Virieux (1986). Die Herleitung erfolgt aus der Spannungs-Dehnungs-Beziehung und den Bewegungsgleichungen

$$\frac{\partial \tau_{xx}}{\partial t} = (\lambda + 2\mu) \frac{\partial v_x}{\partial x} + \lambda \frac{\partial v_z}{\partial z}, \quad (2.1)$$

$$\frac{\partial \tau_{zz}}{\partial t} = (\lambda + 2\mu) \frac{\partial v_z}{\partial z} + \lambda \frac{\partial v_x}{\partial x}, \quad (2.2)$$

$$\frac{\partial \tau_{xz}}{\partial t} = \mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right), \quad (2.3)$$

$$\frac{\partial v_x}{\partial t} = \frac{1}{\rho} \left(\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} \right), \quad (2.4)$$

$$\frac{\partial v_z}{\partial t} = \frac{1}{\rho} \left(\frac{\partial \tau_{xz}}{\partial z} + \frac{\partial \tau_{zz}}{\partial x} \right). \quad (2.5)$$

mit den Geschwindigkeiten v_x, v_z , den Spannungen $\tau_{xx}, \tau_{zz}, \tau_{xz}$, der Dichte ρ und den Lamé-Koeffizienten $\lambda(x, z) = (v_p^2 - 2v_s^2)\rho$ und $\mu(x, z) = v_s^2 \cdot \rho$. Um diese Gleichungen numerisch zu lösen, muss eine zeitliche und räumliche Diskretisierung erfolgen. Dazu wird ein Zeitschrittintervall Δt festgelegt und das Modell auf einem Gitter mit dem in x- und z-Richtung gleichen Gitterknotenabstand Δh definiert. Somit gilt $t = n \cdot \Delta t, x = i \cdot \Delta h, z = j \cdot \Delta h$. Es wird ein geschachteltes Gitter bzw. standard staggered grid(SSD) verwendet, bei dem die verschiedenen Variablen an unterschiedlichen um $0,5\Delta h$ versetzt zwischen den Gitterknoten auf dem Gitter (vgl. 2.1) definiert werden, um die Genauigkeit zu erhöhen. Somit sind τ_{xx}, τ_{zz} bei (i, j) , die Scherspannung τ_{xz} bei $(i+0,5, j+0,5)$, v_x bei $(i+0,5, j)$ und v_z bei $(i, j+0,5)$ durch konkrete Werte festgelegt. Auch in der Zeit gibt es eine Verschachtlung, sodass Geschwindigkeiten bei n und Spannungen um ein halben Zeitschritt versetzt bei $n+0,5$ definiert sind.

Die partiellen Ableitungen werden nun mit einer Taylorentwicklung genähert und so diskretisiert, um die numerische Berechnung zu ermöglichen. Somit können nun schrittweise die Werte einer Variablen aus den vorherigen und umliegenden Werten näherungsweise berechnet werden. Die zeitliche partielle Ableitung wird durch die Taylorentwicklung erster Ordnung

$$\frac{\partial U(n\Delta t)}{\partial t} \approx \frac{U(n\Delta t) - U((n-1)\Delta t)}{\Delta t} + O(h) \quad (2.6)$$

bestimmt, während für die räumlichen Ableitungen höhere Ordnungen von $2M = 2, 4$ verwendet werden, also je M benachbarte Punkte in beide Richtungen mit in die Berechnung eines Gitterpunktes einfließen. Die notwendigen Taylor-Koeffizienten lassen sich mit

$$\sum_{m=1}^M a_m m^2 = 1, \sum_{m=1}^M a_m m^2 k = 0, k = 2, \dots, M \quad (2.7)$$

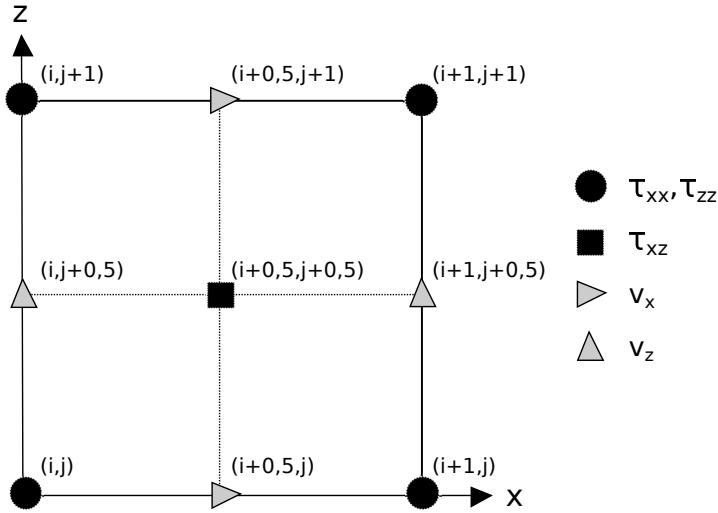


Abbildung 2.1.: Definition der elastischen Wellenfeldparameter auf einem verschachtelten Gitter. Schwarze Symbole stehen für Spannungen bei $t = (n + 0,5)\Delta t$ und graue Symbole für Geschwindigkeiten bei $t = n\Delta t$

M	a_1	a_2	a_3	a_4
1	1			
2	$4/3$	$-1/12$		
4	$8/5$	$-1/5$	$8/310$	$-1/560$

Tabelle 2.1.: Taylor-Koeffizienten für die FD-Modellierung verschiedene Ordnungen M berechnet nach Gleichung 2.7

berechnen. In Tabelle 2.1 sind diese für die Ordnungen $2M = 2, 4, 8$ berechnet.

Somit ergibt sich folgendes Aktualisierungsschema für die schrittweise Berechnung der Geschwindigkeiten und Spannungen in Abhängigkeit von der gewählten Ordnung:

$$v_{x[i+\frac{1}{2},j]}^{n+1} = v_{x[i+\frac{1}{2},j]}^n + \frac{\Delta t}{\rho \Delta h} \sum_{m=1}^M [a(m)(\tau_{xx}^{n+\frac{1}{2}}[i+m+1,j] - \tau_{xx}^{n+\frac{1}{2}}[i-m,j] \\ + \tau_{xz}^{n+\frac{1}{2}}[i+\frac{1}{2},j+\frac{1}{2}+m] - \tau_{xz}^{n+\frac{1}{2}}[i+\frac{1}{2},j-\frac{1}{2}-m])], \quad (2.8)$$

$$v_{z[i,j+\frac{1}{2}]}^{n+1} = v_{z[i,j+\frac{1}{2}]}^n + \frac{\Delta t}{\rho \Delta h} \sum_{m=1}^M [a(m)(\tau_{xz}^{n+\frac{1}{2}}[i+m+\frac{1}{2},j+\frac{1}{2}] - \tau_{xz}^{n+\frac{1}{2}}[i-m-\frac{1}{2},j+\frac{1}{2}] \\ + \tau_{zz}^{n+\frac{1}{2}}[i,j+\frac{1}{2}+m] - \tau_{zz}^{n+\frac{1}{2}}[i,j-\frac{1}{2}-m])], \quad (2.9)$$

$$\tau_{xx}[i,j]^{n+\frac{1}{2}} = \tau_{xx}[i,j]^{n-\frac{1}{2}} + \frac{(\lambda + 2\mu)\Delta t}{\Delta h} \sum_{m=1}^M [a(m)(v_x^n[i+m-\frac{1}{2},j] - v_x^n[i-m+\frac{1}{2},j])] \\ + \frac{\lambda \Delta t}{\Delta h} \sum_{m=1}^M [a(m)(v_z^n[i,j-\frac{1}{2}+m] - v_z^n[i,j-\frac{1}{2}-m])], \quad (2.10)$$

$$\tau_{zz}[i,j]^{n+\frac{1}{2}} = \tau_{zz}[i,j]^{n-\frac{1}{2}} + \frac{(\lambda + 2\mu)\Delta t}{\Delta h} \sum_{m=1}^M [a(m)(v_z^n[i,j-\frac{1}{2}+m] - v_z^n[i,j-\frac{1}{2}-m])] \\ + \frac{\lambda \Delta t}{\Delta h} \sum_{m=1}^M [a(m)(v_x^n[i+m-\frac{1}{2},j] - v_x^n[i-m+\frac{1}{2},j])], \quad (2.11)$$

$$\begin{aligned} \tau_{xz[i+\frac{1}{2}, j+\frac{1}{2}]}^{n+\frac{1}{2}} = & \tau_{xz[i+\frac{1}{2}, j+\frac{1}{2}]}^{n+\frac{1}{2}} + \frac{\lambda \Delta t}{\Delta h} \sum_{m=1}^M [a(m)(v_z^n_{[i+m, j+\frac{1}{2}]} - v_z^n_{[i-M+1, j-\frac{1}{2}]}) \\ & + v_x^n_{[i+\frac{1}{2}, j+m]} - v_x^n_{[i-\frac{1}{2}, j-m+1]})], \end{aligned} \quad (2.12)$$

Voraussetzung für diese Berechnungsmethode ist das Einhalten des Courant-Friedrichs-Lowy-Kriteriums bzw. Stabilitätskriteriums (vgl. Virieux (1986))

$$v_{p,max} \frac{\Delta t}{\Delta h} < \frac{1}{\sqrt{2}}. \quad (2.13)$$

Für die Wahl des Gitterabstandes und des Zeitschrittintervalls ist es zusätzlich wichtig folgendes Kriterium zu beachten, um die Dispersion gering zu halten:

$$\Delta h = \frac{\lambda_{min}}{n} = \frac{v_{min}}{n \cdot f_{max}}. \quad (2.14)$$

Dabei ist λ_{min} die kleinste Wellenlänge und n legt abhängig von der Taylor FD-Ordnung $2M$ die Mindestanzahl an Gitterpunkten pro Wellenlänge fest. .

2.2. Quellimplementation

Die Implementation einer vertikalen Kraftquelle erfolgt durch Addition des Quellsignals f_z zur Bewegungsgleichung der Geschwindigkeit v_z (2.5) bzw. dem Aktualisierungsschema(2.8). Bei einer Explosionsquelle erfolgt die Addition des Quellsignals gleichermaßen auf die Bewegungsgleichungen bzw. Aktualisierungsschema der Normalspannungen τ_{xx} und τ_{zz} . Im klassischen Fall liegt die Quelle an einem Ort, an dem die Variabel zu welcher ihr Signal addiert wird definiert ist, also bei einer vertikalen Kraftquelle bei $(i, j + 0, 5)$ oder bei einer Explosionsquelle bei (i, j) . Dann kann das entsprechende Quellsignal $s(n\Delta t)$ im Laufe der Berechnung mithilfe des Aktualisierungsschemas an diesem einen Punkt eingespeist werden. Liegt die Quelle allerdings an einem für die Implementierung ungünstigen Ort abseits eines Gitterpunktes, muss die Addition des Quellsignals auf die umliegenden Gitterknoten verteilt erfolgen. Dazu wird für alle mit einbezogenen Gitterknoten eine Gewichtungsfunktion $W(i\Delta h, j\Delta h)$ anhand ihres Abstands zur Quelle berechnet und dann das Quellsignal multipliziert mit der Gewichtungsfunktion nach obigem Vorgehen eingespielt. Die Verschiebung der Quelle um ax, az von einem Gitterknoten führt zur Position $(x_s, z_s) = ((i + ax)\Delta h, (j + az)\Delta h)$.

2.3. Sinc-Quellgewichtsfunktion

Die einfachste Methode der Implementierung einer Quelle abseits der Gitterknoten erfolgt durch Interpolation auf die umliegenden Gitterpunkte anhand ihres Abstands zur Quelle mithilfe einer sinc-Funktion. Die Formel zur Berechnung der Gewichte für die Gitterpunkte lautet folgendermaßen:

$$W(i\Delta h, j\Delta h) = \text{sinc}(x - x_s)\text{sinc}(z - z_s) = \frac{\sin \pi(x - x_s)}{\pi(x - x_s)} \frac{\sin \pi(z - z_s)}{\pi(z - z_s)} \quad (2.15)$$

Dabei werden die Berechnungsergebnisse genauer, je mehr Gitterpunkte mit einbezogen werden. In dieser Arbeit erfolgt die Implementierung der Quelle über alle Gitterpunkte, indem für alle die entsprechenden Gewichte berechnet werden. Bei Modellierungen für reale Daten ist dies nicht möglich, da sich die Materialparameter selbst an nahen Gitterpunkten stark unterscheiden können. Hier werden meist nur wenige benachbarte Gitterpunkte berücksichtigt und die Gewichtsfunktion mithilfe eines Kaiser-Fensters nach Hicks (2002) geglättet.

2.4. FD-konsistente Quellgewichtsfunktion

Eine andere Möglichkeit der Interpolation der Quelle ist die FD-konsistente Gewichtsfunktion nach Koene et al. (2020):

$$W(i\Delta h, j\Delta h) = \frac{\sum_{m=1}^M a_m(m-0, 5)(\text{sinc}(x - x_s + m - 0, 5) + \text{sinc}(x - x_s - m + 0.5))}{\Delta h} \\ \frac{\sum_{m=1}^M a_m(m-0, 5)(\text{sinc}(z - z_s + m - 0, 5) + \text{sinc}(z - z_s - m + 0.5))}{\Delta h} \quad (2.16)$$

Hierbei werden die Taylor-Koeffizienten (Tab. 2.1) benötigt und es wird nur auf M Punkte in alle Richtungen interpoliert. Es sollte aber eine höhere Genauigkeit bewirken.

2.5. Analytische Referenzlösung

Um eine Aussage über die Genauigkeit der FD-Methode treffen zu können ist ein Vergleich mit einer Referenzlösung nötig. Hierzu wird die analytische Seismogramme für jeden Empfänger (x_r, z_r) mithilfe der entsprechenden Greenschen Funktionen im Fourier-Raum berechnet, indem diese mit dem fouriertransformierten Quellsignal multipliziert und anschließen wieder rücktransformiert wird. Für den Fall des Druck-Seismogramms einer Explosionsquelle wird hierfür die akustische 2D Greensche Funktion nach Morse and Feshbach (1953)

$$G^{2D} = \sqrt{\frac{2\pi}{kr}} e^{-ikr - \frac{i\pi}{4}} \quad (2.17)$$

mit $k = \frac{\omega}{v_p}$ verwendet. Dabei ist $r = \sqrt{(x_r - x_s)^2 + (z_r - z_s)^2}$ der Abstand zwischen Empfänger und Quelle. Für die Berechnung der analytischen v_x - und v_z -Seismogramme für eine vertikale Kraftquelle werden die elastischen 2D Greenschen Funktionen von Carcione (2014) benötigt.

$$G_1(r, \omega, v_p, v_s) = -\frac{i\pi}{2} \left[\frac{1}{v_p^2} H_0^{(2)} \frac{\omega r}{v_p} + \frac{1}{\omega r v_s} H_1^{(2)} \frac{\omega r}{v_s} - \frac{1}{\omega r v_p} H_1^{(2)} \frac{\omega r}{v_p} \right], \quad (2.18)$$

$$G_2(r, \omega, v_p, v_s) = -\frac{i\pi}{2} \left[\frac{1}{v_s^2} H_0^{(2)} \frac{\omega r}{v_s} + \frac{1}{\omega r v_s} H_1^{(2)} \frac{\omega r}{v_s} + \frac{1}{\omega r v_p} H_1^{(2)} \frac{\omega r}{v_p} \right] \quad (2.19)$$

Aus diesen lässt sich durch

$$u_1(r, \omega, v_p, v_s) = \frac{F_0}{2\pi\rho} \frac{(x_r - x_s)(z_r - z_s)}{r^2} [G_1(r, \omega, v_p, v_s) + G_2(r, \omega, v_p, v_s)], \quad (2.20)$$

$$u_2(r, \omega, v_p, v_s) = \frac{F_0}{2\pi\rho} \frac{1}{r^2} [(z_r - z_s)^2 G_1(r, \omega, v_p, v_s) - (x_r - x_s)^2 G_2(r, \omega, v_p, v_s)]. \quad (2.21)$$

und anschließende Rücktransformation das Wellenfeld berechnen. Dabei ist $H_{0,1}^2$ die Hankelfunktion zweiter Art und nullter bzw. erster Ordnung und F_0 die Größe der Kraftquelle.

2.6. L2-Fehler

Der L2-Fehler lässt sich mit

$$f_{L2} = \frac{\sum_{n=1}^N f_{FD}(n) - f_{ref}(n)}{\sum_{n=1}^N f_{ref}(n)} \quad (2.22)$$

$$f_{L2} = \frac{\sqrt{\sum_{n=1}^N (f_{FD}(n) - f_{ref}(n))^2}}{\sqrt{\sum_{n=1}^N f_{ref}(n)^2}} \quad (2.23)$$

berechnen und zeigt, wie sehr sich die mit der FD-Methode berechneten Seismogramme f_{FD} von der analytisch berechneten Referenzlösung f_{ref} unterscheiden. Je kleiner dieser ist, desto genauer ist die Modellierung. Der L2-Fehler ist somit ein Mittel zur Quantifizierung und zum Vergleich der Güte verschiedener Modellierungen. Dabei wird dieser je für den Zeitraum der Ankunft der P-Welle von 0,02 – 0,05s und der S-Welle von 0,06 – 0,09s berechnet.

3. Modell und Geometrie

Das für alle Berechnungen verwendete Modell ist ein homogenes, isotropes, elastisches Medium mit einer Dichte von $\rho = 2000 \text{ kg/m}^3$, einer P-Wellen-Geschwindigkeit von $v_p = 2000 \text{ m/s}$ und einer S-Wellen-Geschwindigkeit von $v_s = 800 \text{ m/s}$. Das Gitter besteht für alle Berechnungen außer dem Konvergenztest aus $nx = nz = 260$ Gitterpunkten in einem Abstand von $dh = 1 \text{ m}$ in x- und z-Richtung. Die Zeitdiskretisierung erfolgt in Schritten von $\Delta t = 0,0001 \text{ s}$ über einen Zeitraum von $T = 0,1 \text{ s}$. Damit ist die Einhaltung des Stabilitätskriteriums (2.13) gewährleistet. Die zeitliche FD-Ordnung ist $M = 1$ und die örtliche FD-Ordnung ist je nach Angabe entweder $M = 1$ oder $M = 2$.

Die Quelle wird in der Mitte des Gitters auf den Gitterknoten bei den Koordinaten $(x_s, z_s) = (130, 130) \text{ m}$ platziert und für manche Berechnungen anschließend um $a_x \cdot dh, a_z \cdot dh$ verschoben.

Die 20 Empfänger befinden sich auf einem Kreis mit Radius $R = 50 \text{ m}$ um diese herum. Auf dem Kreis sind die Empfänger in gleichen Abständen verteilt und ihre Koordinaten entsprechen dem der so berechneten Position nächstens Gitterpunkt.

Als Quellsignal wird ein Ricker-Wavelet der Frequenz $f_c = 80 \text{ Hz}$ benutzt, welches sich durch

$$s_{Ricker} = \tau \cdot e^{-\tau^2} \quad (3.1)$$

mit

$$\tau = \pi \cdot f_c \cdot \left(t - \frac{1}{f_c} - t_{shift} \right)$$

berechnen lässt. Die Quelle wird für Druck-Seismogramme als Explosionsquelle und für v_x - und v_z -Seismogramme als vertikale Kraftquelle in z-Richtung implementiert.

4. Konvergenztest

Dieser Test wird durchgeführt, um die Implementierung der FD-Methode, sowie der analytischen Lösung zu kontrollieren. Für das Modell mit einer Quelle auf dem Gitterpunkt bei $(x_s, z_s) = (130, 130)$ m, welche mit einer sinc-Funktion über alle Gitterpunkte gewichtet eingefügt wird, werden Druck-, vx- und vz-Seismogramme mit der FD-Methode der Ordnung $M = 1, 2$ für alle 20 Empfänger um die Quelle zunächst mit $\Delta t = 0,0001$ s, $\Delta h = 1$ m berechnet. Anschließend wird diese Rechnung erneut mit zunächst mit im Vergleich zur Standardkonfiguration halbierten und dann geviertennten Δt und Δh , also kleinerer zeitlicher und räumlicher Diskretisierung.

Alle im Rahmen des Konvergenztestes berechneten Seismogramme werden zuletzt mit der entsprechenden analytischen Lösung, also der akustischen für den Druck und der elastischen für die Geschwindigkeiten verglichen. Bei korrekter Funktionsweise der FD-Methode sollte durch Verringerung der Zeitschritte und Gitterabstände das Ergebnis im Vergleich zur analytischen Lösung deutlich genauer werden und zu dieser konvergieren. Es ist anhand der L2-Fehler für den Fall $M = 1$ (Abb.4.1) und $M = 2$ (Abb.4.2) gut zu erkennen, dass diese stark abnehmen, je kleinschrittiger die Diskretisierung ist. Aufgetragen sind die L2-Fehler der Empfänger in den Abbildungen über dem Winkel ϕ von der Quelle zum jeweiligen Empfänger abgetragen ausgehend von der x-Achse. Auch ist hier zu sehen, dass die Genauigkeit der Druck-Seismogramme, also im akustischen Fall wie zu erwarten höher ist. Auch sind vor allem die Genauigkeit der v_x -Seismogramme deutlich von der Abstrahlrichtung in der sie aufgenommen wurden abhängig und die L2-Fehler bei Abstrahlung in Richtung der Achsen sind deutlich höher. Dies liegt vor allem am Einfluss der Dispersion, welcher im nächsten Kapitel genauer betrachtet und korrigiert wird. Noch deutlicher ist die Dispersion in Abb. 4.3 an der S-Welle zu erkennen. Allerdings ist auch zu sehen, dass in Richtung der Achsen kaum ein Ausschlag im Seismogramm zu erkennen ist und somit deshalb bei Vergleich der Differenzen der Lösungen mit den geringen Werten der Referenzlösung bei Berechnung des L2-Fehlers ein so großer Wert herauskommt. Weitere Abbildungen zum Konvergenztest, wie einzelne Darstellungen von Differenzen zwischen den FD- und analytischen Lösungen für verschiedene Abstrahlwinkel sind im Anhang zu finden. Dort lässt sich auch erkennen, dass die Dispersion im Fall höherer Ordnung $M = 2$ geringer ausfällt. Letztendlich ist aber eine Konvergenz an die analytischen Seismogramme aufgetreten, also funktionieren die FD-Methode, sowie die Berechnung der analytischen Lösung.

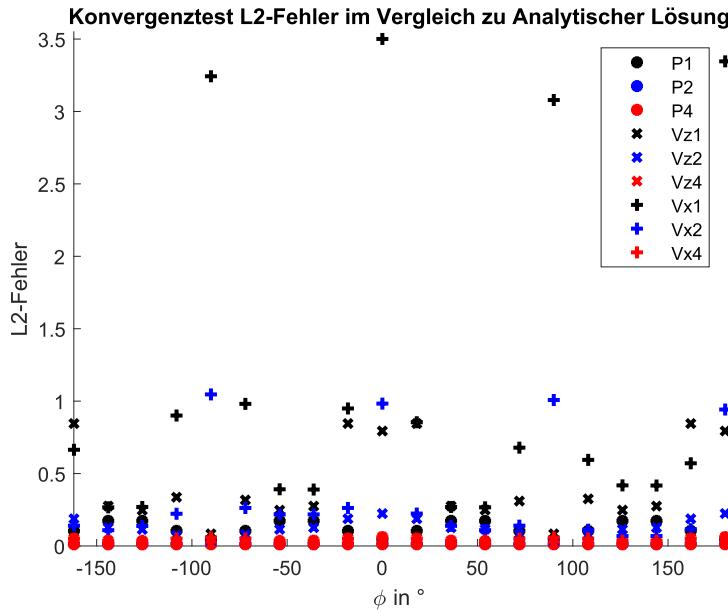


Abbildung 4.1.: L2-Fehlers des Vergleiches der im Konvergenztest mit schrittweise verringriger Diskretisierung berechneter Seismogramme der Ordnung $M = 1$ mit den analytisch berechneten Referenzlösungen. Die Seismogramme des Druckes P werden als Punkte, die v_x -Seismogramme als + und die v_z -Seismogramme als x dargestellt. Schwarze Symbole wurden mit der Standarddiskretisierung, blaue mit halbiert und rote mit geviertelter Schrittweite berechnet.

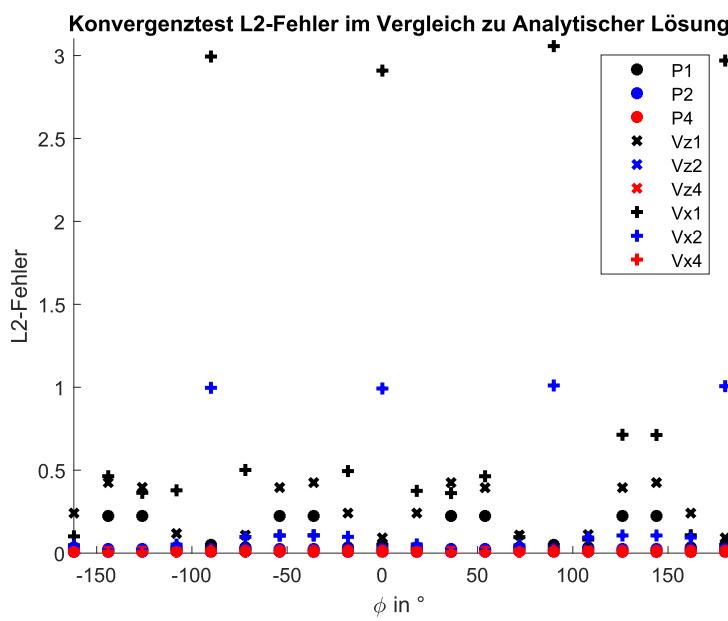


Abbildung 4.2.: L2-Fehlers des Vergleiches der im Konvergenztest mit schrittweise verringriger Diskretisierung berechneter Seismogramme der Ordnung $M = 2$ mit den analytisch berechneten Referenzlösungen. Die Seismogramme des Druckes P werden als Punkte, die v_x -Seismogramme als + und die v_z -Seismogramme als x dargestellt. Schwarze Symbole wurden mit der Standarddiskretisierung, blaue mit halbiert und rote mit geviertelter Schrittweite berechnet.

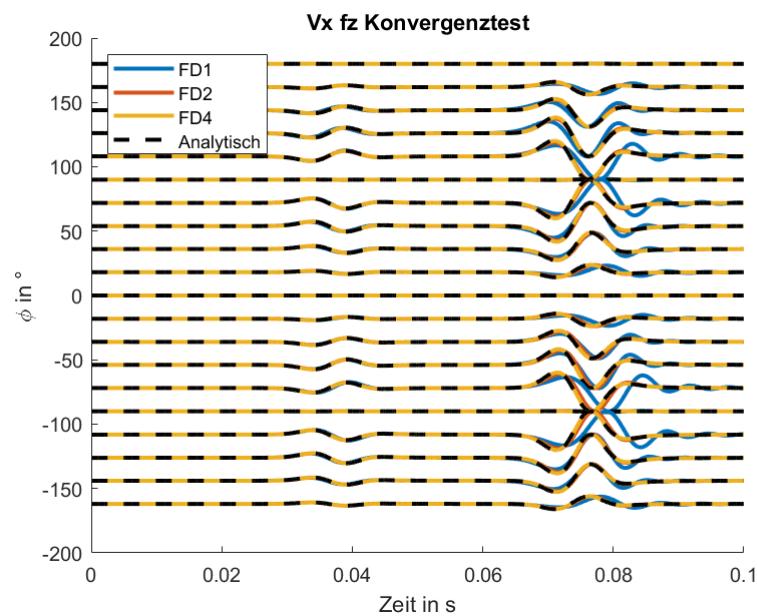


Abbildung 4.3.: Darstellung der für den Konvergenztest der Ordnung $M = 2$ berechneten v_x -Seismogramme im Vergleich mit der schwarz-gestrichelten analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

5. Dispersionskorrektur

Wie beim Konvergenztest eindeutig zu erkennen ist, ist es vor allem bei der Ordnung $M = 1$, sowie der vertikalen Kraftquelle unbedingt nötig den numerischen Dispersionsfehler in Abhängigkeit vom Abstrahlwinkel zu korrigieren, um genaue, aussagekräftige Seismogramme zu erhalten. Dafür muss zunächst die Geschwindigkeitsänderung durch die räumliche und zeitliche Dispersion bestimmt werden. Hergleitet werden kann diese durch Einsetzen des Ansatzes einer ebenen Welle in die Bewegungsgleichungen und die Aktualisierungsschema. Nach Liu and Sen (2011) ist dieses Verhältnis der numerischen zur tatsächlichen Geschwindigkeit gegeben durch

$$\frac{v_{FD}}{v} = \frac{2}{k\Delta hr} \arcsin r \sqrt{\left[\sum_{m=1}^M a_m \sin(m - 0,5) k \Delta h \cos \theta \right]^2 + \left[\sum_{m=1}^M a_m \sin(m - 0,5) k \Delta h \sin \theta \right]^2} \quad (5.1)$$

mit den Taylorkoeffizienten(2.1) a_m , dem Abstand zwischen Quelle und Empfänger r , sowie dem Abstrahlwinkel θ abgetragen ausgehend von der z-Achse. Ausgehend von dieser Geschwindigkeitsänderung durch die Dispersion kann nun eine Korrektur des Fehlers durchgeführt werden. Hierzu wird zunächst die nötige Zeitverschiebung

$$t_{shift} = r \Delta h \left(\frac{1}{2v - v_{FD}} - \frac{1}{v} \right) \quad (5.2)$$

berechnet und anschließend im Fourier-Raum eine Filterung des FD-modellierten Seismogrammes $S(\omega)$ durch

$$S(\omega)_c = S(\omega) \exp -it_{shift} \quad (5.3)$$

durchgeführt. Nach Rücktransformation ist das Seismogramm dispersionskorrigiert. Dieser Vorgang muss separat für die Aufzeichnung der P-Welle und der S-Welle erfolgen, da je die v_p bzw v_s für v in den Gleichungen benutzt wird. Somit ergeben sich je Seismogramme im Zeitbereich $t_p = 0,02 - 0,05$ s und $t_s = 0,06 - 0,09$ s.

Die mit Gleichung 5.1 berechenbare prozentuale numerische Geschwindigkeitsänderung durch die Dispersion ist für den Fall $M = 1$ in Abb. 5.1 dargestellt. Es fällt auf, dass die Dispersion für die S-Welle stärker ist.

In den Abbildungen 5.2, 5.3 sind die L2-Fehler der mit $M = 1$ und der berechneten Seismogramme mit und ohne Dispersionskorrektur im Vergleich zur analytischen Referenzlösung zu sehen für eine Quelle auf dem Gitterknoten mit $ax, az = 0$. Dabei ist zu erkennen, dass die Dispersionskorrektur die Genauigkeit deutlich steigert und somit notwendig ist, um die Genauigkeit der beiden Quellimplementierungen zu vergleichen. Die Verbesserung der Ergebnisse ist ähnlich der durch den Konvergenztest, also einer engeren Diskretisierung, welche mit deutlich mehr Rechenaufwand verbunden ist.

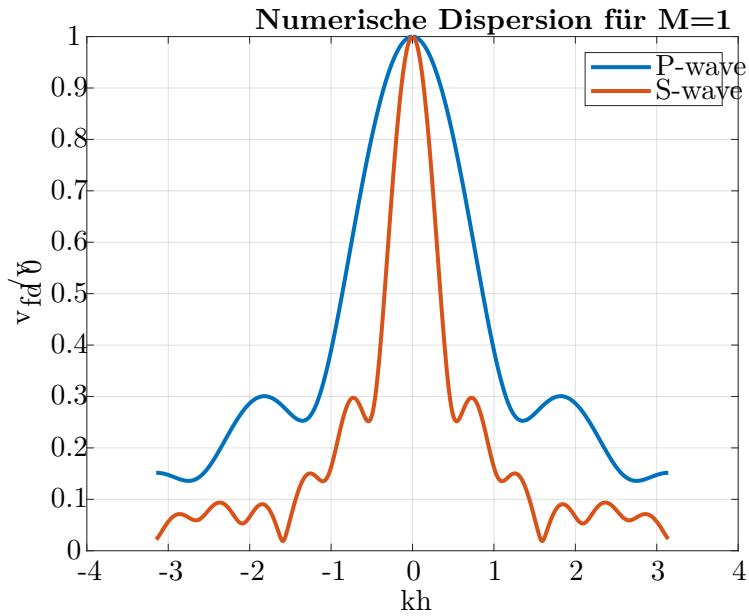


Abbildung 5.1.: Geschwindigkeitsänderung durch die numerische Dispersion bei $M = 1$ über $k\Delta h$

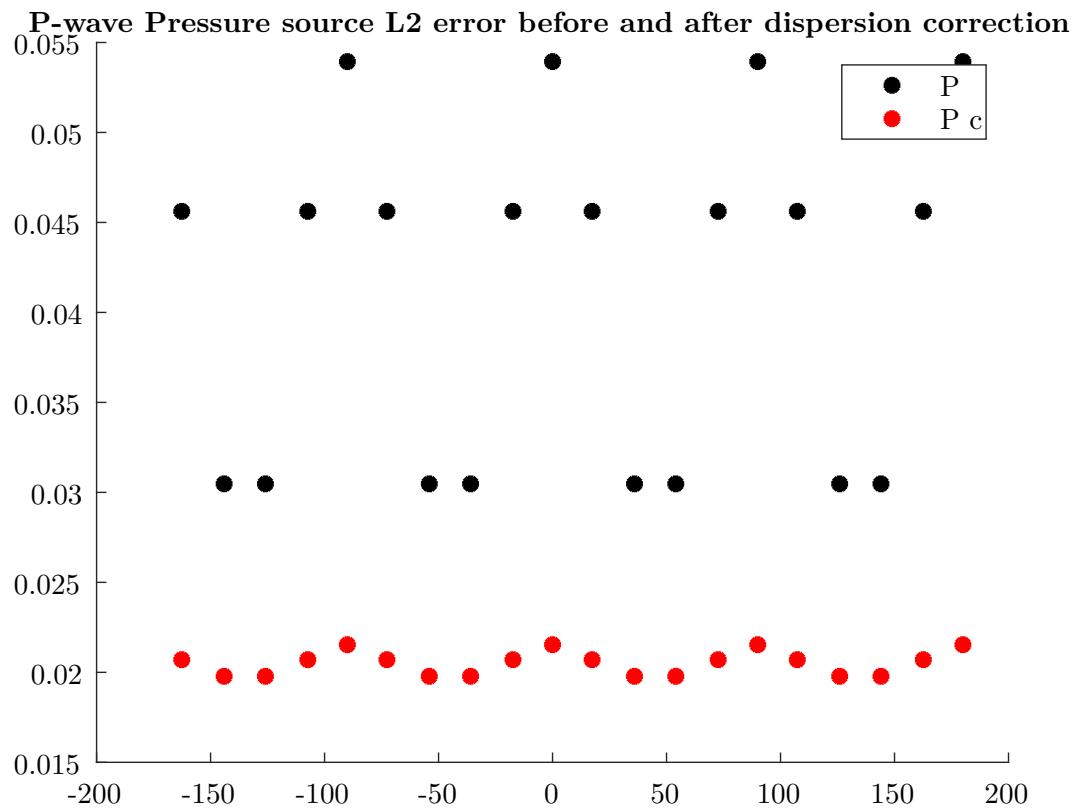


Abbildung 5.2.: L2-Fehler der Druck-Seismogramme mit $M = 2, ax = 0, az = 0$ mit(rot) und ohne(schwarz) Dispersionskorrektur

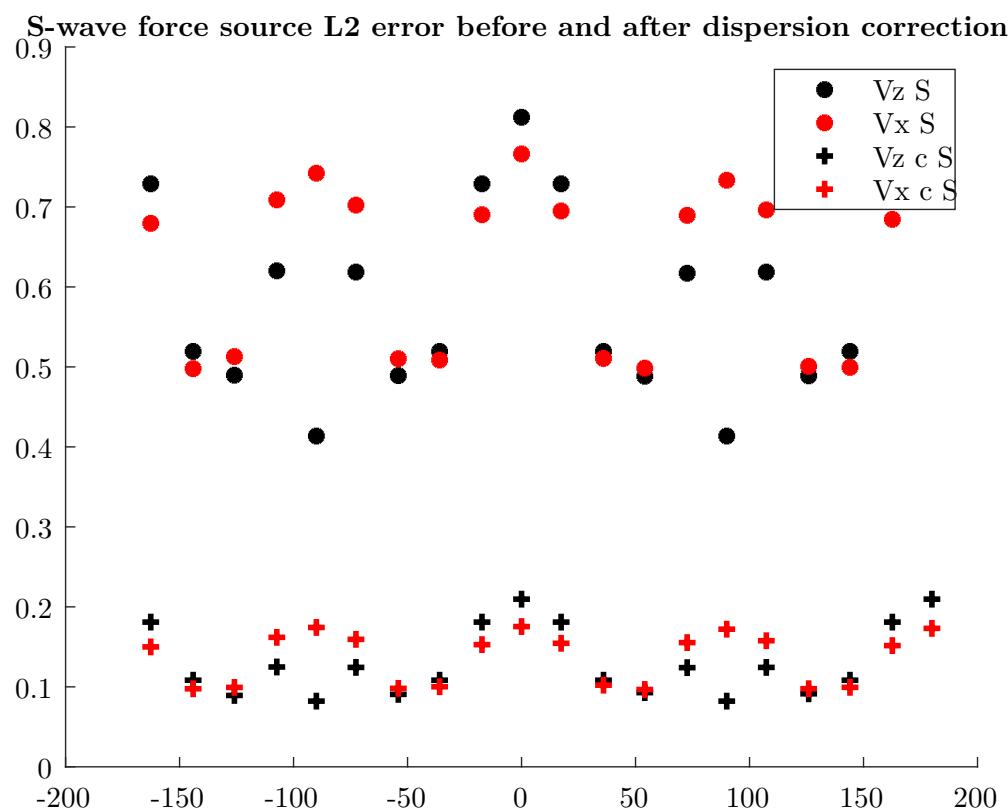


Abbildung 5.3.: L2-Fehler der Geschwindigkeits-Seismogramme mit $M = 2, ax = 0, az = 0$ mit und ohne Dispersionskorrektur für die S-Welle als + bzw. Punkt. v_z in schwarz, v_x in rot

6. Flexible Quellenpositionierung

Um die Genauigkeit der beiden Gewichtungsfunktionen zu vergleichen, wird zunächst der Fall $ax = az = 0,5$ betrachtet. Die Abbildungen 6.1 6.2 6.3 zeigen die Berechneten L2-Fehler für die flexibel und die auf dem Knotenpunkt platzierte Quelle mit der sinc- bzw FD-konsistenten Funktion. Bei den L2-Fehlern der Druck-Seismogrammen ist deutlich zu erkennen, dass die FD-konsistente Quellimplementierungsmethode ein genauereres Ergebnis auch bei flexibler Positionierung liefert, gefolgt von den mit der sinc-Funktion interpolierten Seismogrammen auf dem Gitterknoten, während die flexibel positionierten noch etwas ungenauer sind. Auch hier sieht man trotz der zuvor angewandten Dispersionskorrektur noch eine Abhängigkeit vom Abstrahlwinkel mit den größten Ungenauigkeiten in Richtung der Achsen. Bei den S-Wellen-Ausschnitten der Kraftquelle ist die Reihenfolge der Genauigkeiten gleich, nur die Unterschiede der verschiedenen Methoden sind nicht so markant. Bei den P-Wellen der Kraftquelle lässt sich keine konkrete Aussage treffen, welche Methode die genauesten Seismogramme ermöglicht, da es eine sehr große Streuung gibt. Bei Probieren weiterer Positionen ergab sich das Bild, das die sinc-Funktion der S-Wellen unabhängig von der Quellposition immer ähnlich gute Ergebnisse liefert. Während die FD-konsistente Methode sehr große Fehler hervorbringt, sobald die Quelle mehr in Richtung einer Achse, also nicht mittig verschoben wird. Dies kann aber möglicherweise auch an einem Fehler in der Implementierung liegen.

Für die Ordnung $M = 2$ ist bei $a_x = a_z = 0,5$ die FD-konsistente Methode bei der Kraftquelle, sowohl im Bereich der P-Welle als auch der S-Welle besser, allerdings ist beim Druck-Seismogramm der Fehler bei allen Varianten ähnlich. Die entsprechenden Abbildungen befinden sich im Anhang.

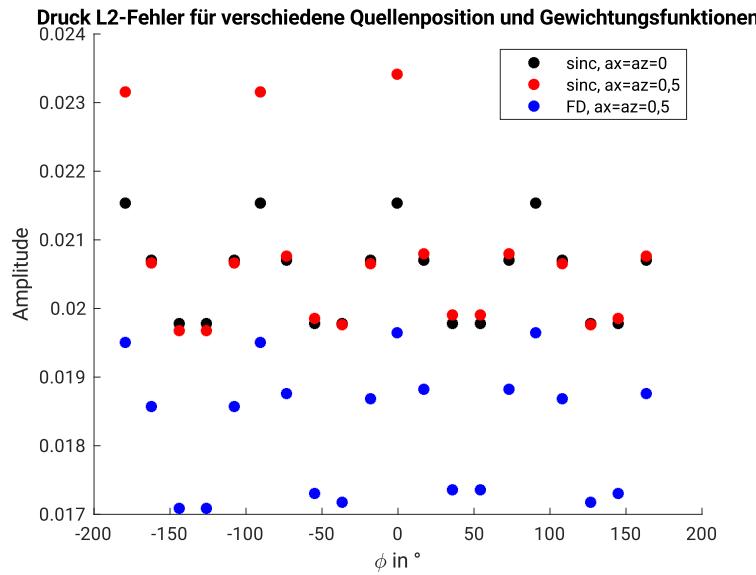


Abbildung 6.1.: L2 Fehler der Druck Seismogramme für Quellen auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

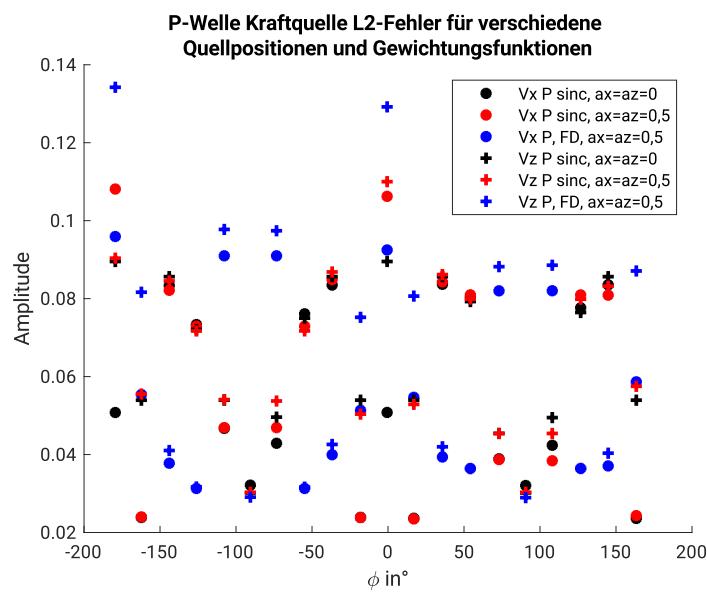


Abbildung 6.2.: L2 Fehler der P-Wellen Seismogramme für Kraftquellen auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

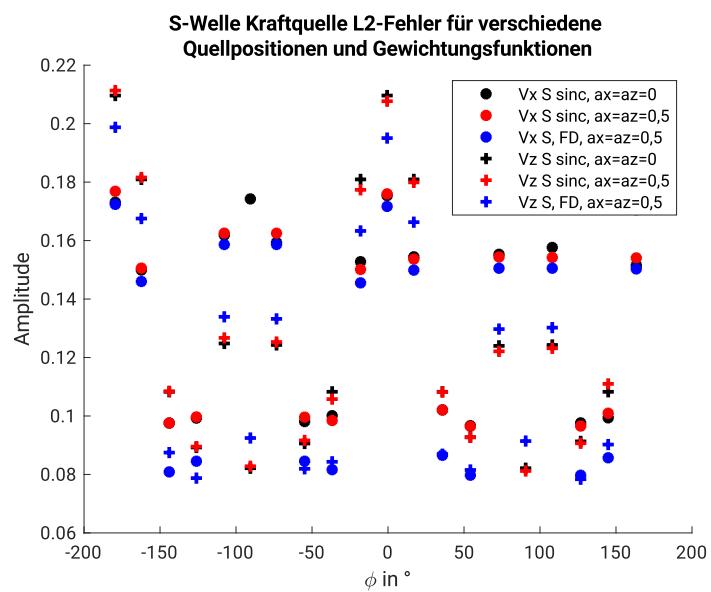


Abbildung 6.3.: L2 Fehler der S-Wellen Seismogramme für Kraftquellen auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

7. Schlussfolgerung

Demnach lässt sich zusammenfassend sagen, dass die Wahl der Gewichtungsfunktion zur Interpolation der Quellenposition zwischen der sinc- und der FD-konsistenten-Funktion keine zu großen Auswirkungen hat, da die Fehler bei Vergleich der beiden Funktionen eher sehr klein im Verhältnis zu den durch die numerische Dispersion in Raum und Zeit auftretenden Unsicherheiten ist. Nur bei sehr engen Gittern oder einer sehr großen FD-Ordnung, also nur wenig andere Unsicherheiten macht es vermutlich Sinn sich genau zu überlegen welche Variante besser geeignet ist. Vorteil der FD-konsistenten-Quelle ist, dass diese per Definition bereits nur umliegende Gitterpunkte im Umkreis von M miteinbezieht, während es bei der sinc-Funktion bei heterogenen Medien nötig ist eine angepasste Fensterfunktion zur Filterung zu Nutzen und sich selbst zu überlegen wie viele umliegenden Gitterpunkt zur Interpolation benutzt werden sollen. Da allerdings bei der Arbeit mit realen Daten kaum Dispersionskorrekturen möglich sind, ist in einem solchen Fall die Wahl der Variante wohl nicht sehr relevant.

Anhang

A. Zusätzliche Abbildungen

A.1. Konvergenztest

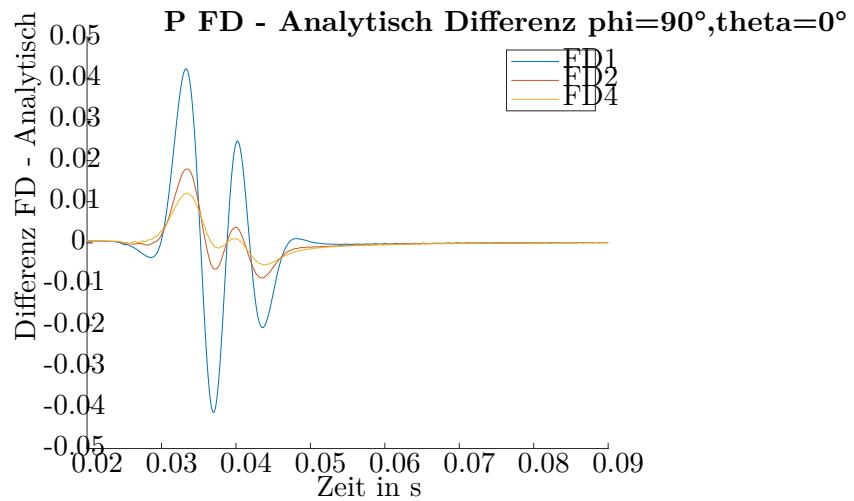


Abbildung A.1.: Für den Konvergenztest der Ordnung $M = 1$ berechneten Druck-Seismogramm des in z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbiert Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

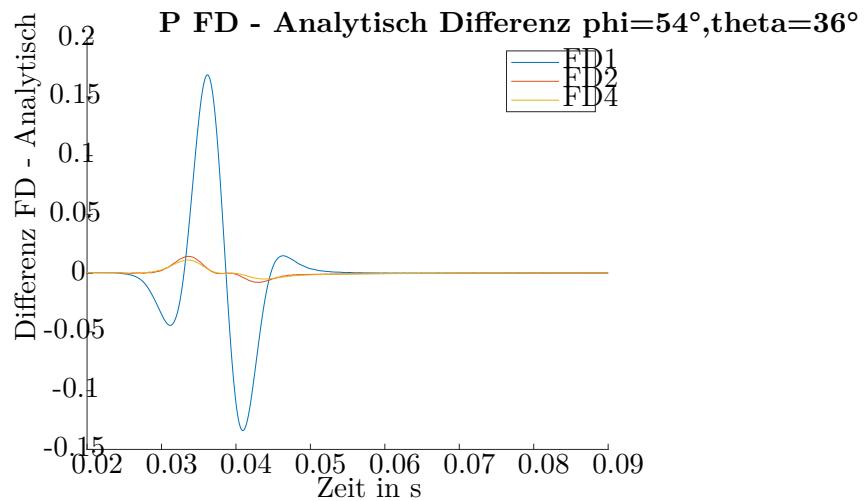


Abbildung A.2.: Für den Konvergenztest der Ordnung $M = 1$ berechneten Druck-Seismogramm eines zwischen x- und z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbiert Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

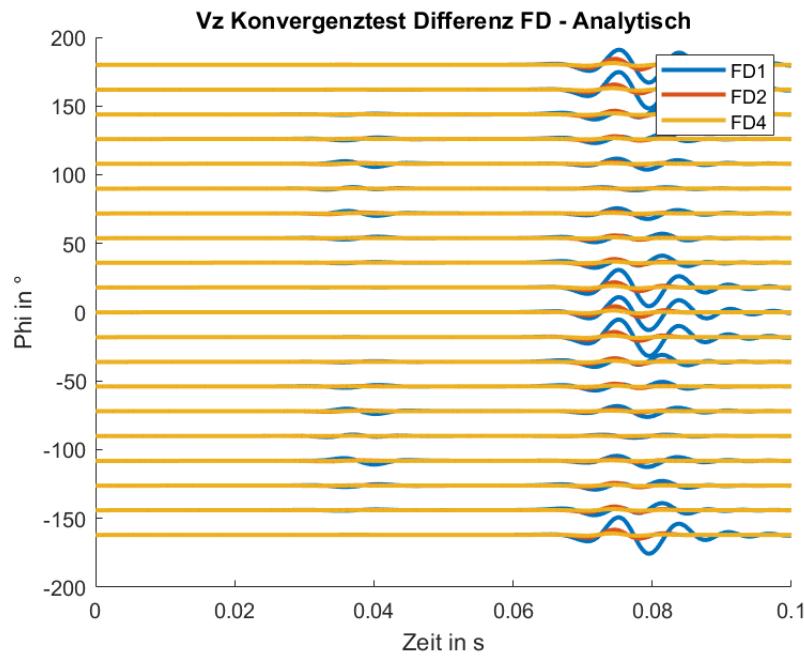


Abbildung A.3.: Darstellung der für den Konvergenztest der Ordnung $M = 1$ berechneten v_z -Seismogramme als Differenz mit der analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbiert Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

A.2. Dispersionskorrektur

A.3. Flexible Positionierung der Quelle

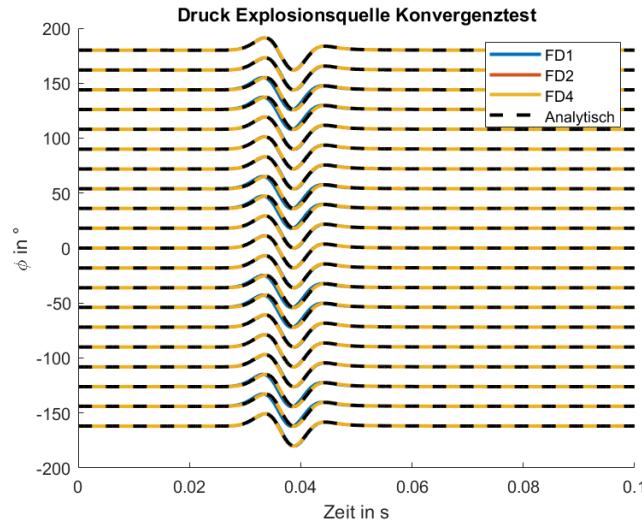


Abbildung A.4.: Darstellung der für den Konvergenztest der Ordnung $M = 2$ berechneten Druck-Seismogramme im Vergleich mit der analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

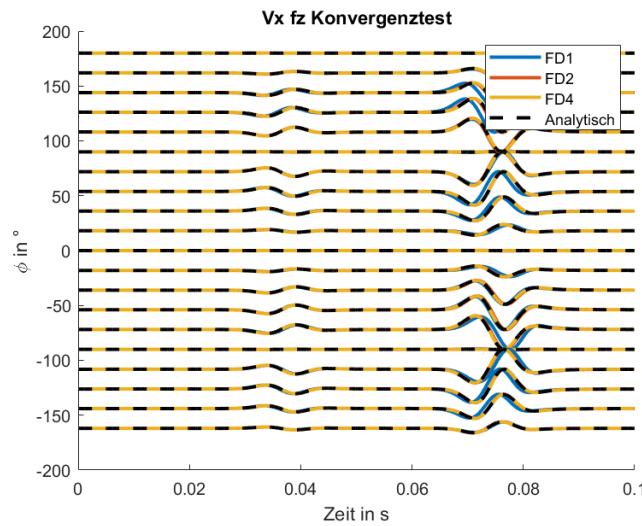


Abbildung A.5.: Darstellung der für den Konvergenztest der Ordnung $M = 2$ berechneten v_x -Seismogramme im Vergleich mit der schwarz gestrichelten analytischen Lösung über dem Abstrahlwinkel der Empfänger ϕ . Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

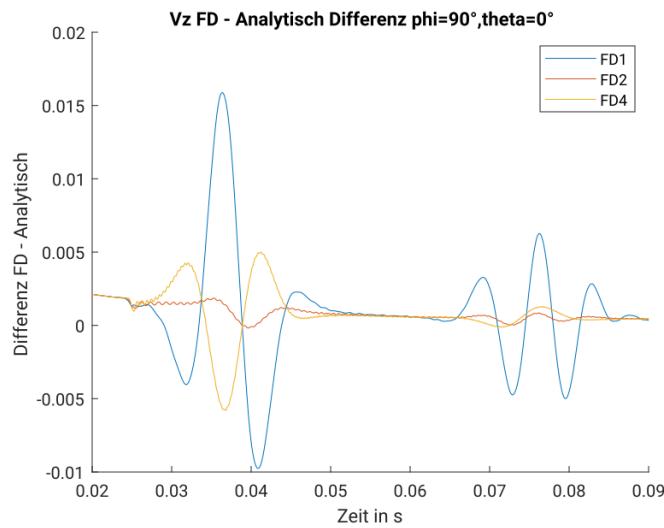


Abbildung A.6.: Für den Konvergenztest der Ordnung $M = 2$ berechneten v_z -Seismogramm des in z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

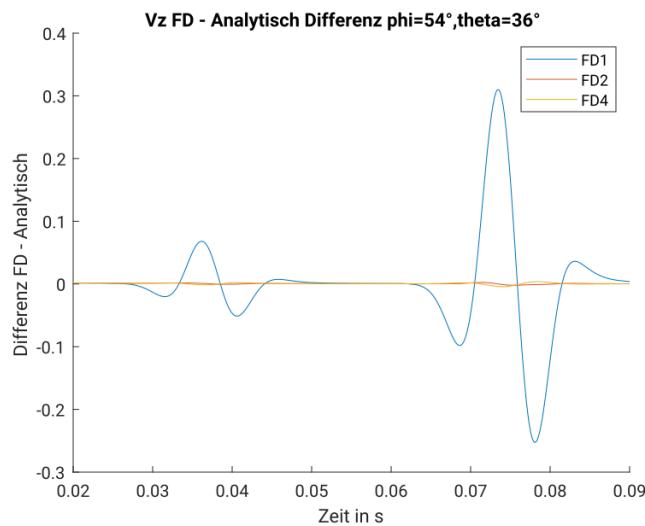


Abbildung A.7.: Für den Konvergenztest der Ordnung $M = 2$ berechneten v_z -Seismogramm eines zwischen x- und z-Richtung gerichteten Empfängers als Differenz mit der analytischen. Seismogramme der Standarddiskretisierung werden blau, mit halbierter Schrittweite orange und mit geviertelter Schrittweite gelb eingezeichnet.

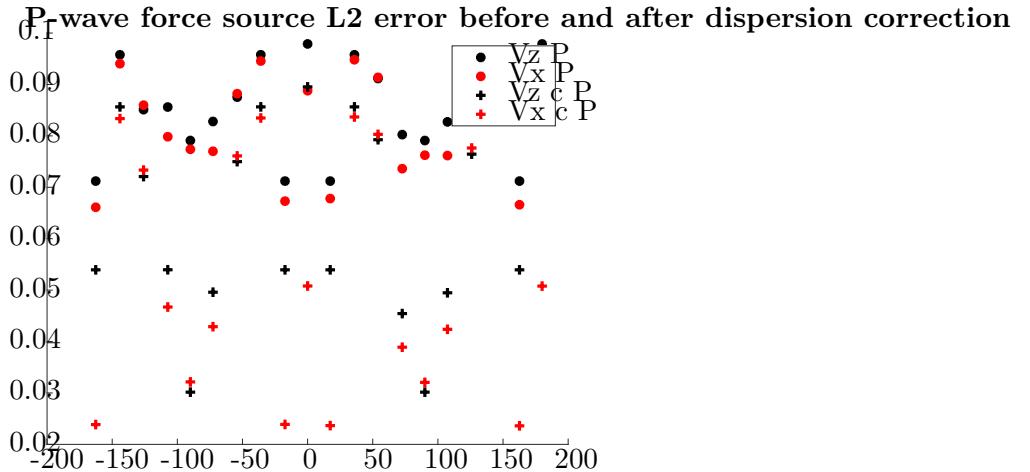


Abbildung A.8.: L2-Fehler der Geschwindigkeits-Seismogramme mit $M = 2, ax = 0, az = 0$ mit und ohne Dispersionskorrektur für die S-Welle als + bzw. Punkt. v_z in schwarz, v_x in rot

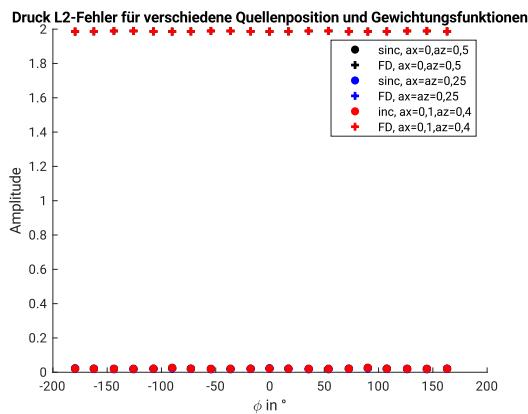


Abbildung A.9.: L2 Fehler der Druck-Seismogramme mit $M = 1$ für verschieden versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

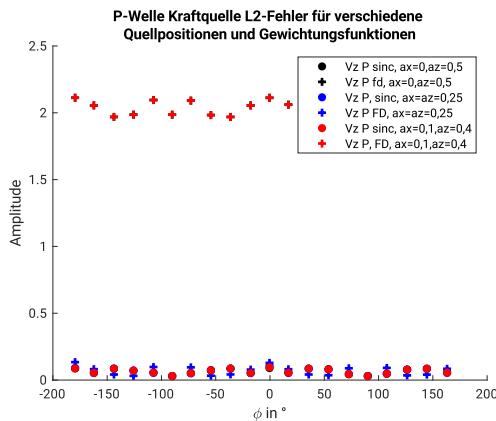


Abbildung A.10.: L2 Fehler der P-Wellen-Seismogramme einer Kraftquelle mit $M = 1$ für verschieden versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

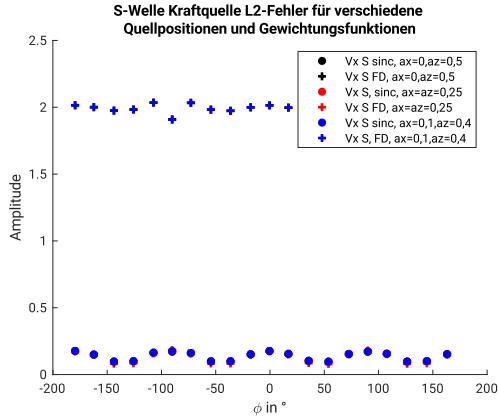


Abbildung A.11.: L2 Fehler der S-Wellen-Seismogramme mit $M = 1$ für verschieden versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

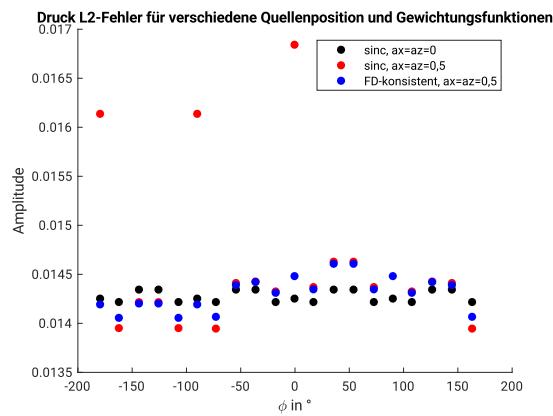


Abbildung A.12.: L2 Fehler der Druck-Seismogramme auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion bei $M = 2$

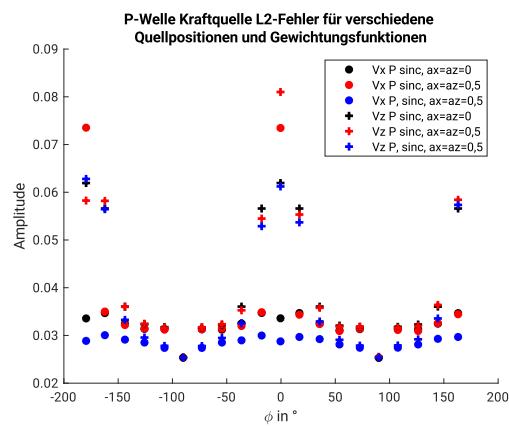


Abbildung A.13.: L2 Fehler der P-Wellen-Seismogramme einer Kraftquelle auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion bei $M = 2$

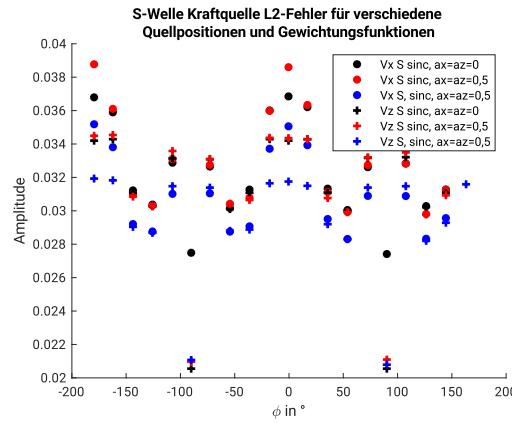


Abbildung A.14.: L2 Fehler der S-Wellen-Seismogramme auf einem Gitterpunkt und um $0,5\Delta h$ versetzt mit der sinc bzw. der FD-konsistenten Gewichtsfunktion bei $M = 2$

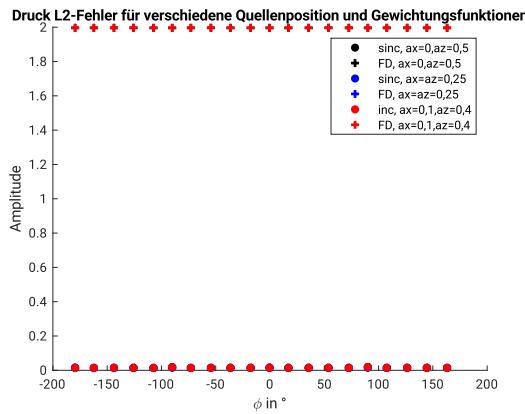


Abbildung A.15.: L2-Fehler der Druck-Seismogramme mit $M = 2$ für verschieden versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

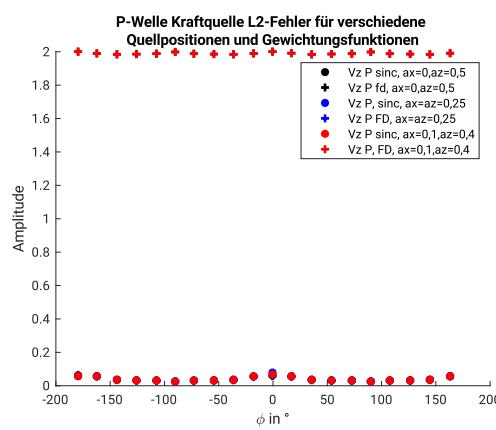


Abbildung A.16.: L2-Fehler der P-Wellen-Seismogramme mit $M = 2$ für verschieden versetzte Kraftquellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

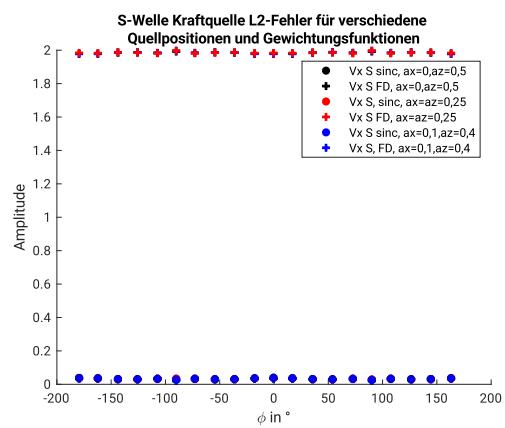


Abbildung A.17.: L2 Fehler der S-Wellen-Seismogramme mit $M = 2$ für verschieden versetzte Quellen mit der sinc bzw. der FD-konsistenten Gewichtsfunktion

B. Matlab Skripte

```

1 % Run FD and compare with analytical solution
2
3 close all; clear all;
4
5 fc=80.0 % center frequency of Ricker wavelet
6 M=1;      % LENGTH of Operator: Order=2M !!
7 Nr=1;
8
9 dispersioncorrection=1;
10
11 sourceimplementation=0; %0=sinc ,1=fd-consistent
12
13 % Shift of source in grid points (<=0.5)
14 ax=0.0; az=0.0;
15
16 r=130;
17
18
19 sourcetype=1;
20 elastic2d;
21 sourcetype=2;
22 elastic2d;
23 Vergleich_FD_analytic_2D;
24
25 ax=0.5;az=0.5;
26 sourcetype=1;
27 elastic2d;
28 sourcetype=2;
29 elastic2d;
30 Vergleich_FD_analytic_2D;
31
32 sourceimplementation=1;
33 sourcetype=1;
34 elastic2d;
35 sourcetype=2;
36 elastic2d;
37 Vergleich_FD_analytic_2D;

```

```

1 %close all;
2 %clear all;
3
4 % -----
5 % define grid parameters
6
7 %standard grid configuration
8 nx=260;      % Model size in grid points
9 nz=260;
10 dh=1.0;
11 dt=1.0e-4;
12 radius=50.0;    % radius of receiver circle in gridpoints
13

```

```
14 % denser grid (convergence test)
15 % nx=520;          % Model size in grid points
16 % nz=520;
17 % dh=0.5;           % Grid spacing [meter]
18 % dt=0.5e-4;        % time step [seconds]
19 % radius=100.0;     % radius of receiver circle in gridpoints
20 if Nr>1
21     nx=nx*Nr;
22     nz=nz*Nr;
23     dh=dh/Nr;
24     dt=dt/Nr;radius=50*Nr;
25 end
26
27 %-----
28 T=0.1;      % propagation time [s]
29
30 xs=round(nx/2);           % source position in grid
   points
31 zs=round(nz/2);
32
33 if ~exist('fc')
34     fc=80.0;           % center frequency of Ricker wavelet
35 end
36
37 nr=20;           % number of receivers in circle
38
39
40 if ~exist('M')
41     M=4;               % Length of FD operator: Order=2M
42 end
43
44 if ~exist('sourcetype')
45     sourcetype=1;       % =1: Pressure; =2: Vertical force;
   =3: Horizontal force
46 end
47
48 snap=1;           % output of snapshots
49
50
51 %model parameter
52 vp=2000.0; vs=800.0; rhov=2000.0;
53
54 partxt='Model 1';
55
56 framerate=0.010; % frames per second
57 framerate=round(framerate/dt);
58 %-----
59
60 % time and spatial vector
61 t=dt:dt:T;
62 x=(1:nx)*dh;
63 z=(1:nz)*dh;
```

```
64 nt=length(t);  
65  
66  
67 % Initialize wavefield matrix  
68 vx=zeros(nx,nz);  
69 vz=zeros(nx,nz);  
70 sxx=zeros(nx,nz);  
71 sxz=zeros(nx,nz);  
72 szz=zeros(nx,nz);  
73 p=zeros(nx,nz);  
74  
75 % Model 1  
76 c0=zeros(nx,nz);  
77 c1=zeros(nx,nz);  
78 rho=zeros(nx,nz);  
79 c0(:,:)=vp; % vp  
80 c1(:,:)=vs; % vs  
81 rho(:,:)=rhov;  
82  
83 % model 2  
84 %c0(:,1:40)=1000.0;  
85 %c1(:,1:40)=500.0;  
86 %rho(:,1:40)=1800;  
87  
88 bo=1./rho;  
89 lambda2mu=rho.*c0.*c0;  
90 mu=rho.*c1.*c1;  
91 lambda=lambda2mu-2*mu;  
92 cfl=c0.*dt/dh;  
93  
94 % check stability criterion  
95 cflnum=max(max(cfl));  
96 disp(['CFL number:',num2str(cflnum)]);  
97 if (cflnum>1) disp(['CFL number is greater than one. Stop.'])  
    ;  
98     return;  
99 end  
100  
101 % check numerical dispersion (minimum wavelength)  
102 lambda_p=min(min(c0)/(fc));  
103 disp([' Dominant P-wavelength is ',num2str(lambda_p), ' m']);  
104 disp([' Number of gridpoints per dominant wavelength is ',  
    num2str(lambda_p/dh)]);  
105 lambda_s=min(min(c1)/(fc));  
106 disp([' Dominant S-wavelength is ',num2str(lambda_s), ' m']);  
107 disp([' Number of gridpoints per dominant wavelength is ',  
    num2str(lambda_s/dh)]);  
108  
109 % define ricker wavelet  
110 rickerp=Ricker(t,fc,1.5*dt); % Pressure source  
111 rickerf=Ricker(t,fc,1.0*dt); % Force source  
112
```

```
113 dh2=1.0/(dh*dh);
114 dtx=dt/dh;
115 bo=bo*dtx;
116 lambda=lambda*dtx;
117 lambda2mu=lambda2mu*dtx;
118 mu=mu*dtx;
119
120
121 % calculate source and receiver coordinates
122 [xr,zr,phi]=circ(nr,radius);
123 % Discrete FD source positions
124 xr=round(xs+xr);
125 zr=round(zs+zr);
126
127
128
129 seispp=zeros(nt,nr);
130 seisvxp=zeros(nt,nr);
131 seisvzp=zeros(nt,nr);
132 seispf=zeros(nt,nr);
133 seisvxf=zeros(nt,nr);
134 seisvzf=zeros(nt,nr);
135
136 a=fdcoeff(M);
137
138 % b=4.18;
139 %W=source(xs,zs,ax,az,r,nx,nz);
140 if(sourceimplementation==0)
141     W=source(xs,zs,ax,az,r,nx,nz);
142 end
143 if(sourceimplementation==1)
144     W=sourceKoene(xs,zs,ax,az,r,nx,nz,a,dh,dh);
145 end
146 figure
147 imagesc(W'); colorbar; title('Source Distribution (zoom)')
148 zooms=r+2;
149 axis([xs-zooms xs+zooms zs-zooms zs+zooms])
150 xlabel('x axis'); ylabel('z axis');
151
152
153 filename='model1.gif';
154 if snap,
155     h=figure; imagenum=1;
156 end
157
158 % Loop over time steps and grid points
159 for n=1:nt
160     if (mod(n,framerate)==0)
161         disp(['Timestep: ', num2str(n), ' of ', num2str(nt)]);
162     end
163
164     switch sourcetype,
```

```

165      case 1
166          for i=M+1:nx-M
167              for j=M+1:nz-M
168                  sxx(i,j)=sxx(i,j)+W(i,j)*rickerp(n);
169                  szz(i,j)=szz(i,j)+W(i,j)*rickerp(n);
170              end
171          end
172          out=p; thred=0.1;
173      case 2
174          for i=M+1:nx-M
175              for j=M+1:nz-M
176                  vz(i,j)=vz(i,j)+W(i,j)*rickerf(n);
177              end
178          end
179          out=vz; thred=2.0e-2;
180      case 3
181          for i=M+1:nx-M
182              for j=M+1:nz-M
183                  vx(i,j)=vx(i,j)+W(i,j)*rickerf(n);
184              end
185          end
186          out=vx; thred=2.0e-2;
187      end
188
189
190      for i=M+1:nx-M
191          for j=M+1:nz-M
192              sxx_x=0.0; sxz_x=0.0; sxz_z=0.0; szz_z=0.0;
193              for m=1:M,
194                  sxx_x=sxx_x+a(m)*(sxx(i+m,j)-sxx(i-m+1,j));
195                  sxz_x=sxz_x+a(m)*(sxz(i+m-1,j)-sxz(i-m,j));
196                  sxz_z=sxz_z+a(m)*(sxz(i,j+m-1)-sxz(i,j-m));
197                  szz_z=szz_z+a(m)*(szz(i,j+m)-szz(i,j-m+1));
198              end
199
200              vz(i,j)=vz(i,j)+bo(i,j)*(sxz_x+szz_z);
201              vx(i,j)=vx(i,j)+bo(i,j)*(sxx_x+sxz_z);
202          end
203      end
204      for i=M+1:nx-M
205          for j=M+1:nz-M
206              vxx=0.0; vxz=0.0; vzx=0.0; vzz=0.0;
207              for m=1:M,
208                  vxx=vxx+a(m)*(vx(i+m-1,j)-vx(i-m,j));
209                  vxz=vzx+a(m)*(vz(i+m,j)-vz(i-m+1,j));
210                  vzx=vzx+a(m)*(vx(i,j+m)-vx(i,j-m+1));
211                  vzz=vzz+a(m)*(vz(i,j+m-1)-vz(i,j-m));
212              end
213              sxx(i,j)=sxx(i,j)+lambda2mu(i,j)*vxx+lambda(i,j)*
214                  vzz;
215              szz(i,j)=szz(i,j)+lambda2mu(i,j)*vzz+lambda(i,j)*
216                  vxx;

```

```

215     sxz(i,j)=sxz(i,j)+mu(i,j)*(vzx+vxz);
216     p(i,j)=sxx(i,j)+szz(i,j);
217   end
218 end
219
220 switch sourcetype,
221 case 1 % Pressure
222   for ntr=1:nr,
223     Seispp(n,ntr)=p(xr(ntr),zr(ntr));
224     Seisvxp(n,ntr)=vx(xr(ntr),zr(ntr));
225     Seisvzp(n,ntr)=vz(xr(ntr),zr(ntr));
226
227   end
228 case 2 % Force
229   for ntr=1:nr,
230     Seispf(n,ntr)=p(xr(ntr),zr(ntr));
231     Seisvxf(n,ntr)=vx(xr(ntr),zr(ntr));
232     Seisvzf(n,ntr)=vz(xr(ntr),zr(ntr));
233   end
234 end
235
236 if ((mod(n,framerate)==0)&&snap)
237
238   imagesc(out');
239
240   title([partxt,' snapshot at T=',num2str(n*dt),'s']);
241   clim([-thred thred]);
242   axis tight;
243   colormap(jet);
244   drawnow;
245   hold on
246   plot(xs,zs,'*')
247   for n=1:nr;
248     plot(xr(n),zr(n),'+')
249   end
250 %
251 %
252 %
253 %       frame=getframe(h);
254 %       im=frame2im(frame);
255 %       [imind,cm]=rgb2ind(im,256);
256 %       pause(0.1);
257 %       if(imagenum==1)
258 %           imwrite(imind,cm,filename,'gif','Loopcount',Inf);
259 %           imagenum=imagenum+1;
260 %       else
261 %           imwrite(imind,cm,filename,'gif','Writemode','append');
262 %           imagenum=imagenum+1;
263 %       end
264 end

```

```

265 %
266 %-----
267
268 function a=fdcoeff(M)
269 %Calculate FD-coefficients for the first derivative
270 A=zeros(M);
271 for l=1:M,
272     for k=1:M,
273         A(l,k)=(2*k-1)^(2*l-1);
274     end
275 end
276 c=zeros(1,M);
277 c(1)=1;
278
279 %A = sym(A); c = sym(c);
280 c = c';
281 a=inv(A)*c;
282 end
283
284
285 function [xr,zr,phi]=circ(nr,r)
286 dphi=2*pi/nr;
287 n=1;
288 xr=zeros(1,nr);
289 zr=zeros(1,nr);
290 phi=zeros(1,nr);
291 for p=0:dphi:2*pi-dphi,
292     phi(n)=p;
293     xr(n)=r*sin(p);
294     zr(n)=r*cos(p);
295     n=n+1;
296 end
297 end

```

```

1
2 % Calculate true locations of sources and receivers
3 % Thereby considering
4 % (1) FD staggered positions, (2) Source shift ax, az
5 % Remember: Discrete FD locations in grid points are xs,zs (
6 % source) xr, zr
7 % (receivers)
8
9 % Spatial shift
10 dh2=dh/2.0;
11
12 % First calculate source and receiver locations in meter
13 xr=xr*dh; zr=zr*dh; xs=xs*dh; zs=zs*dh;
14
15 % Apply Source shift ax,az
16 xs=xs+ax*dh; zs=zs+az*dh;
17

```

```

18 % Consider FD staggering (no staggering for p, staggering for
19 % vx, vy, fz)
20 % Pressure (no staggering)
21 xrp=xr; zrp=zr; xsp=xs; zsp=zs;
22 % Staggering for receivers recording vx
23 xrvx=xr+dh2; zrvx=zr;
24 % Staggering for receivers recording vz
25 xrvz=xr; zrvz=zr+dh2;
26 % Staggering for source fz
27 xsfz=xs; zsfz=zs+dh2;
28
29 %Calculate analytic signals-----
30 vx=zeros(nt,ntr);
31 vz=zeros(nt,ntr);
32 p=zeros(nt,ntr);
33
34 % analytical solutions assume source at (0,0). Therefore
35 % above FD coordinates must
36 % be shifted so that source is located at (0,0)
37
38 % Offsets for pressure source and pressure receivers (
39 % acoustic
40 % solution)
41 dxp=xrp-xsp; dzp=zrp-zsp;
42 R=sqrt(dxp.^2+dzp.^2);
43 % Calculate analytical solution for the acoustic case
44 for n=1:ntr,
45     [p1,t]=GF2Dac(fc, vp, R(n), dt, T);
46     p(:,n)=p1;
47 end
48 % Offsets for fz source and vx, vz receivers (elastic
49 % solution)
50 dxvxfz=xrvx-xsfz; dzvxfz=zrvx-zsfz;
51 dxvzfz=xrvz-xsfz; dzvzfz=zrvz-zsfz;
52
53 for n=1:ntr,
54     [vx1,vz1,t]=GF2Del(fc, vp, vs, rhov, dt, T, dxvzfz(n), dzvzfz(n));
55     vz(:,n)=vz1;
56     [vx1,vz1,t]=GF2Del(fc, vp, vs, rhov, dt, T, dxvxfz(n), dzvxfz(n));
57     vx(:,n)=vx1;
58 end
59
60 %-----
61 % Compare seismograms
62 tracenorm=1; % apply trace normalization
63 xcur=1.0; % amplitude scaling
64 phir=atan2(dzvzfz,dxvzfz);
65 phi=180.0*phir/pi;
66 xa=phi;
67 xp=xcur*(xa(2)-xa(1));

```

```
66
67
68 if tracenorm,
69 for n=1:ntr
70     norm1=max(abs([Seisvxf(:,n);Seisvzf(:,n)]));
71     norm2=max(abs([vx(:,n);vz(:,n)]));
72     Seisvxf(:,n)=Seisvxf(:,n)/norm1;
73     Seisvzf(:,n)=Seisvzf(:,n)/norm1;
74         vx(:,n)=vx(:,n)/norm2;
75         vz(:,n)=vz(:,n)/norm2;
76     norm1=max(abs([Seispp(:,n)]));
77     norm2=max(abs([p(:,n)]));
78     Seispp(:,n)=Seispp(:,n)/norm1;
79     p(:,n)=p(:,n)/norm2;
80
81
82 end
83 end
84
85 if dispersioncorrection,
86
87 f=(-nt/2:nt/2-1)/(dt*nt);
88 w=2.0*pi*f;
89 kp=w./vp;
90 ks=w./vs;
91
92 kph=kp*dh;
93 ksh=ks*dh;
94
95 rp=vp*dt/dh;
96 rs=vs*dt/dh;
97
98 %shift von quellen und empf nger auf theta
99 % phirp=atan2(dxp,dzp);
100 % phirvz=atan2(dxvzfz,dzvzfz);
101 % phirvx=atan2(dxvxfz,dzvxfz);
102
103 for n=1:ntr,
104     kphx=kph*sin(phir(n));
105     kphz=kph*cos(phir(n));
106     kshx=ksh*sin(phir(n));
107     kshz=ksh*cos(phir(n));
108
109 S1P=0;S2P=0;
110 S1S=0;S2S=0;
111 for m=1:M,
112     S1P=S1P+(a(m)*sin((m-0.5)*abs(kphx)));
113     S2P=S2P+(a(m)*sin((m-0.5)*abs(kphz)));
114     S1S=S1S+(a(m)*sin((m-0.5)*abs(kshx)));
115     S2S=S2S+(a(m)*sin((m-0.5)*abs(kshz)));
116 end
117
```

```

118 Dp=2.0*asin(rp*sqrt((S1P.*S1P)+(S2P.*S2P)))./(abs(kph)*rp
119 ) ;
120 Ds=2.0*asin(rs*sqrt((S1S.*S1S)+(S2S.*S2S)))./(abs(ksh)*rs
121 ) ;
122 dcp=vp*(1.0-Dp) ;
123 dcs=vs*(1.0-Ds) ;
124 tshiftp=radius*dh*((1.0./(vp+dcp))-(1.0./vp));
125 tshifts=radius*dh*((1.0./(vs+dcs))-(1.0./vs));
126 % Seisppc(:,n)=dispccorr(Seispp(:,n)',t,tshiftp);
127 % Seisvxfcs(:,n)=dispccorr(Seisvxf(:,n)',t,tshifts);
128 % Seisvzfcs(:,n)=dispccorr(Seisvzf(:,n)',t,tshifts);
129 % Seisvxfcp(:,n)=dispccorr(Seisvxf(:,n)',t,tshiftp);
130 % Seisvzfcp(:,n)=dispccorr(Seisvzf(:,n)',t,tshiftp);
131 Seisppc(:,n)=dispccorrection(Seispp(:,n)',t,dxp(n),dzp
132 (n),kph, rp , vp , a , dh );
133 Seisvxfcs(:,n)=dispccorrection(Seisvxf(:,n)',t,dxvxfz(
134 n),dzvxfz(n),ksh, rs , vs , a , dh );
135 Seisvzfcs(:,n)=dispccorrection(Seisvzf(:,n)',t,dxvzfz(
136 n),dzvzfz(n),ksh, rs , vs , a , dh );
137 Seisvxfcp(:,n)=dispccorrection(Seisvxf(:,n)',t,dxvxfz(
138 n),dzvxfz(n),kph, rp , vp , a , dh );
139 Seisvzfcp(:,n)=dispccorrection(Seisvzf(:,n)',t,dxvzfz(
140 n),dzvzfz(n),kph, rp , vp , a , dh );
141 end
142
143 figure;
144 kphr=pi*kph/max(kph);
145 kshr=pi*ksh/max(ksh);
146 plot(kphr,Dp,kshr,Ds,'linewidth',2.0)
147 xlabel('kh')
148 ylabel(' c_{fd}/c_0 ');
149 title([' Numerical dispersion for 1-D FD 0(2,',num2str(M
150 ','))]);
151 legend("P-wave","S-wave","p","location","northeast")
152
153 else
154 Seisppc(:,n)=Seispp(:,n);
155 Seisvxfcs(:,n)=Seisvxf(:,n);
156 Seisvzfcs(:,n)=Seisvzf(:,n);
157 Seisvxfcp(:,n)=Seisvxf(:,n);
158 Seisvzfcp(:,n)=Seisvzf(:,n);
159 end

```

```
162
163
164 % figure; hold on
165 %
166 % for trace=1:ntr
167 %     plot(t,Seisvzf(:,trace)*xp+xa(trace),'k--','linewidth'
168 %           ',2.0)
169 %     plot(t,Seisvzfcs(:,trace)*xp+xa(trace),'b-','linewidth'
170 %           ',2.0)
171 %     plot(t,vz(:,trace)*xp+xa(trace),'r-','linewidth',2.0)
172 % end
173 % legend("FD","FD corrected","Analytic","location",
174 %        "northeast")
175 % title('Vz fz S-wave Dispersion correction')
176 %
177 % hold off
178 %
179 %
180 % axis xy
181 % %axis([ min(t) max(t) min(xa)-2*xp max(xa)+2*xp])
182 % %xlim([0.065 0.085])
183 %
184 % %-----
185 % figure; hold on
186 %
187 % for trace=1:ntr
188 %     plot(t,Seisvzf(:,trace)*xp+xa(trace),'k--','linewidth'
189 %           ',2.0)
190 %     plot(t,Seisvzfcp(:,trace)*xp+xa(trace),'b-','linewidth'
191 %           ',2.0)
192 %     plot(t,vz(:,trace)*xp+xa(trace),'r-','linewidth',2.0)
193 % end
194 % legend("FD","FD corrected","Analytic","location",
195 %        "northeast")
196 % title('Vz fz P-wave Dispersion correction')
197 %
198 %
199 % hold off
200 %
201 %
202 % %-----
203 %
204 % figure; hold on
205 %
206 % for trace=1:ntr
207 %     plot(t,Seisvxf(:,trace)*xp+xa(trace),'k--','linewidth'
```

```
    ',2.0)
208 %     plot(t,Seisvxfcs(:,trace)*xp+xa(trace),'b-','linewidth
' ,2.0)
209 %     plot(t,vx(:,trace)*xp+xa(trace),'r-','linewidth',2.0)
210 % end
211 % legend("FD","FD corrected","Analytic","location",
"northeast")
212 % title('Vx fz S-wave Dispersion correction')
213 %
214 % hold off
215 %
216 % axis xy
217 % %axis([ min(t) max(t) min(xa)-2*xp max(xa)+2*xp])
218 % %xlim([0.065 0.085])
219 %     ylabel('Phi (deg)');
220 %     xlabel('time (s)');
221 %
222 % %-----
223 %
224 % figure; hold on
225 %
226 % for trace=1:ntr
227 %     plot(t,Seisvxf(:,trace)*xp+xa(trace),'k--','linewidth
' ,2.0)
228 %     plot(t,Seisvxfcp(:,trace)*xp+xa(trace),'b-','linewidth
' ,2.0)
229 %     plot(t,vx(:,trace)*xp+xa(trace),'r-','linewidth',2.0)
230 % end
231 % legend("FD","FD corrected","Analytic","location",
"northeast")
232 % title('Vx fz P-wave Dispersion correction')
233 %
234 % hold off
235 %
236 % axis xy
237 % %axis([ min(t) max(t) min(xa)-2*xp max(xa)+2*xp])
238 % %xlim([0.065 0.085])
239 %     ylabel('Phi (deg)');
240 %     xlabel('time (s)');
241 %
242 % %-----
243
244 figure; hold on
245
246 for trace=1:ntr
247     plot(t,Seispp(:,trace)*xp+xa(trace),'k--','linewidth'
,2.0)
248     plot(t,Seisppc(:,trace)*xp+xa(trace),'b-','linewidth'
,2.0)
249     plot(t,p(:,trace)*xp+xa(trace),'r-','linewidth',2.0)
250 end
251 legend("FD","FD corrected","Analytic","location","northeast")
```

```

    ")
252 title('P P P-wave Dispersion Correction')
253
254 hold off
255 %
256 axis xy
257
258 ylabel('Phi (deg)');
259 xlabel('time (s)');
260
261 %nwp1=1; nwp2=nt/2;
262 %nws1=nt/2; nws2=nt;
263 nwp1=0.02/dt; nwp2=0.05/dt;
264 nws1=0.06/dt; nws2=0.09/dt;
265
266 for trace=1:ntr,
267     l2errvzs(trace)=sqrt(sum((Seisvzf(nws1:nws2,trace)-vz(
268         nws1:nws2,trace)).^2))/sqrt(sum((vz(nws1:nws2,trace))-
269         .^2));
270     l2errvxcs(trace)=sqrt(sum((Seisvxf(nws1:nws2,trace)-vx(
271         nws1:nws2,trace)).^2))/sqrt(sum((vx(nws1:nws2,trace))-
272         .^2));
273     l2errp(trace)=sqrt(sum((Seispp(nwp1:nwp2,trace)-p(nwp1:
274         nwp2,trace)).^2))/sqrt(sum((p(nwp1:nwp2,trace)).^2));
275     l2errvzcs(trace)=sqrt(sum((Seisvzfcs(nws1:nws2,trace)-vz(
276         nws1:nws2,trace)).^2))/sqrt(sum((vz(nws1:nws2,trace))-
277         .^2));
278     l2errvxp(trace)=sqrt(sum((Seisvxf(nwp1:nwp2,trace)-vx(
279         nwp1:nwp2,trace)).^2))/sqrt(sum((vx(nwp1:nwp2,trace))-
280         .^2));
281     l2errvzcp(trace)=sqrt(sum((Seisvzfcp(nwp1:nwp2,trace)-vz(
282         nwp1:nwp2,trace)).^2))/sqrt(sum((vz(nwp1:nwp2,trace))-
283         .^2));
284     l2errvxcp(trace)=sqrt(sum((Seisvxfcp(nwp1:nwp2,trace)-vx(
285         nwp1:nwp2,trace)).^2))/sqrt(sum((vx(nwp1:nwp2,trace))-
286         .^2));
287
288 end
289 %-----
290 figure; hold on
291 plot(phi,l2errvzs,'k*',phi,l2errvxcs,'r*','linewidth',2.0)
292 plot(phi,l2errvzcs,'k+',phi,l2errvxcs,'r+','linewidth',2.0)
293 legend("Vz S","Vx S","Vz c S","Vx c S","location","northeast
   ")

```

```

284 title('S-wave force source L2 error before and after
285     dispersion correction')
286 hold off
287 %-----
288
289 figure; hold on
290 plot(phi,l2errvzp,'k*',phi,l2errvxp,'r*','linewidth',2.0)
291 plot(phi,l2errvzcp,'k+',phi,l2errvxcp,'r+','linewidth',2.0)
292 legend("Vz P","Vx P","Vz c P","Vx c P","location","northeast")
293     ")
294 title('P-wave force source L2 error before and after
295     dispersion correction')
296 hold off
297 %-----
298
299 figure; hold on
300 plot(phi,l2errp,'k*',phi,l2errpc,'r*','linewidth',2.0)
301 legend("P","P c","location","northeast")
302 title('P-wave Pressure source L2 error before and after
303     dispersion correction')
304 hold off
305
306
307
308
309
310
311 %-----
312
313 figure; hold on
314
315 for trace=1:ntr
316     plot(t,Seispp(:,trace)-p(:,trace),'k--','linewidth',2.0)
317     plot(t,Seisppc(:,trace)-p(:,trace),'b-','linewidth',2.0)
318 end
319 legend("FD - Analytic","FD corrected - Analytic","location
320     ","northeast")
321 title('Pressure Explosion source P-wave Dispersion
322     Correction, Difference to Analytic solution')
323 hold off
324 axis xy
325 ylabel('Amplitude');
326 xlabel('time (s)');

```

```

1 function [W] = source(xs,zs,ax,az,r,nx,nz)
2
3 %Calculate wieights for the spatial distribution of the

```

```
    source centered at
4 % xs ,zs
5
6 W=zeros(nx ,nz);
7 %W(xs ,zs)=1.0;
8
9 if ((abs(ax)>0.5) || (abs(az)>0.5))
10    disp("Warning from source.m: ax or az greater 0.5")
11 end
12
13
14 if r==0 ,
15    ax=0 .0 ;
16    az=0 .0 ;
17    r=1 ;
18 end
19
20 for k=xs -r+1:xs +r
21    for l=zs -r+1:zs +r
22        %Sinc function in x
23        x=k-xs -ax;
24        sx=sincc(x);
25
26        %Sinc function in z
27        z=l-zs -az;
28        sz=sincc(z);
29
30        W(k ,l)=sx*sz;
31    end
32 end
33
34
35
36 % for k=1:nx
37 %     for l=1:nz
38 %         %Sinc function in x
39 %         x=k+ax-xs ;
40 %         sx=sincc(x);
41 %
42 %         %Sinc function in z
43 %         z=l+az-zs ;
44 %         sz=sincc(z);
45 %
46 %         W(k ,l)=sx*sz;
47 %     end
48 % end
49
50
51
52 end
53
54
```

```

55 %
56 %
57 %           % Kaiser window in x
58 %           Kx=1.0;
59 %           if kaiser,
60 %               if (abs(x)<=r),
61 %                   y=b*sqrt(1-((x*x)/(r*r)));
62 %                   Kx=besseli(0,y)/besseli(0,b);
63 %                   %Kx=1.0;
64 %               else
65 %                   Kx=0.0;
66 %               end
67 %           end

```

```

1 function [W] = sourceKoene(xs,zs,ax,az,r,nx,nz,a,dx,dz) %FD-
    Konsistente-Methode(Koene)
2
3 %Calculate wieights for the spatial distribution of the
4 % source centered at
5 % xs+ax ,zs+az
6
7 W=zeros(nx,nz);
8 L=length(a);
9
10 if ((abs(ax)>0.5) || (abs(az)>0.5))
11     disp("Warning from source.m: ax or az greater 0.5")
12 end
13
14 for k=1:nx
15     for l=1:nz
16         x=k-xs-ax;
17         sx=0;
18         z=l-zs-az;
19         sz=0;
20         for i=1:L
21             %sx=sx+(a(i)*(i-0.5)*(sincc(pi*(x/dx+i-0.5))+%
22             %sincc(pi*(x/dx-i+0.5)))/dx; %(A1)
23             %sz=sz+(a(i)*(i-0.5)*(sincc(pi*(z/dz+i-0.5))+%
24             %sincc(pi*(z/dz-i+0.5)))/dz; %(A1)
25             sx=sx+(a(i)*(i-0.5)*(sincc(pi*(x+i-0.5))+sincc(pi*
26             *(x-i+0.5)))/dx; %(A1)
27             sz=sz+(a(i)*(i-0.5)*(sincc(pi*(z+i-0.5))+sincc(pi*
28             *(z-i+0.5)))/dz; %(A1)
29
30
31         W(k,l)=Wx*Wz;
32     end

```

```

33 end
34
35
36
37
38 end

```

```

1 function [p,t]=GF2Dac(fc,c,R,dt,T)
2
3 n2ft=T/dt;
4
5 t=(1:n2ft)*dt;
6 fnyq=1/(2*dt);
7
8 tsour=1/fc;
9 tau=pi*(t(1:(2*round(tsour/dt)))-tsour)/tsour;
10
11 delay=1.0/fc; shift=0.0/fc; ts=t-delay-shift; tau=pi*fc*ts;
12 pt0=(1.0-2.0*tau.*tau).*exp(-tau.*tau);
13
14 PT0=fftshift(fft(pt0,n2ft));
15
16 f=fnyq*(-n2ft/2:n2ft/2-1)/(n2ft/2);
17 w=2*pi*f;
18 w(n2ft/2+1)=w(n2ft/2);
19 k=w./c;
20
21 I2D=sqrt(2.0*pi./(k*R));
22 I2D=I2D.*exp(-i*k*R-pi/4.0);
23
24 P2D=I2D.*PT0;
25
26 p2D=real(ifft(fftshift(P2D),n2ft));
27 p=p2D;

```

```

1 function [vx,vy,t]=GF2Del(fc,vp0,vs0,rho,dt,T,x,y)
2
3 n2ft=T/dt;
4
5 t=(1:n2ft)*dt;
6 fnyq=1/(2*dt);
7
8 tsour=1/fc;
9 tau=pi*(t(1:(2*round(tsour/dt)))-tsour)/tsour;
10
11 delay=1.0/fc; shift=0.0/fc; ts=t-delay-shift; tau=pi*fc*ts;
12 s=(1.0-2.0*tau.*tau).*exp(-tau.*tau);
13 %s=tau.*exp(-tau.*tau);
14 t0=delay;
15
16 S=fftshift(fft(s,n2ft));
17

```

```

18 f=fnyq*(-n2ft/2:n2ft/2-1)/(n2ft/2);
19 w=2*pi*f;
20 w(n2ft/2+1)=w(n2ft/2);
21
22 vp=zeros(n2ft)+vp0;
23 vs=zeros(n2ft)+vs0;
24
25 F=1.0;
26
27 % correct for staggered FD position of vx shifted by +dh/2 in
28 % x
29 % and -dh/2 in z
30 % only for x-component u1
31 dh=1.0;
32
33 % x2=x;
34 % y2=y-dh/2;
35 % x1=x-dh/2;
36 % y1=y-dh/2;
37 % x1=x-dh/2;
38 % y1=y+dh/2;
39 x1=x;x2=x;
40 y1=y;y2=y;
41
42 r1=sqrt(x1^2+y1^2); % Distance to the receiver for u1
43 r2=sqrt(x2^2+y2^2); % Distance to the receiver for u2
44
45
46 %% Green's functions (CORRECTED, as appear in Carcione (2002)
47 )
48 for j=1:length(w)
49 if w(j)>0
50 G1=-i*pi/2*(1/vp(j)^2*besselh(0,2,(w(j)*r1/(vp(j))))+
51 ...
52 1. / (w(j)*r1*vs(j))*besselh(1,2,(w(j)*r1/(vs(j))))-
53 ...
54 1. / (w(j)*r1*vp(j))*besselh(1,2,(w(j)*r1/(vp(j)))) );
55 G2=i*pi/2*(1/vs(j)^2*besselh(0,2,(w(j)*r2/(vs(j))))-
56 ...
57 1. / (w(j)*r2*vs(j))*besselh(1,2,(w(j)*r2/(vs(j))))+
58 ...
59 1. / (w(j)*r2*vp(j))*besselh(1,2,(w(j)*r2/(vp(j)))) );
60
61 u1(j)=F/(2*pi*rho)*(x1*y1/r1^2)*(G1+G2);
62 u2(j)=F/(2*pi*rho)*(1/r2^2)*(y2^2*G1-x2^2*G2);
63 end
64 end
65
66
67

```

```

62 %Ricker type source
63 %S=sqrt(pi)*w.^2/(4*(pi*fc)^3).*exp(-i*w*t0).*exp(-w.^2/(4*(pi*fc)^2));
64
65
66 %% Construction of the w-space displacements
67 PHI_DX=u1.*S;
68 PHI_DY=u2.*S;
69
70 %% Construction of the w-space velocities
71 %PHI_VX=PHI_DX.*i.*w;
72 %PHI_VY=PHI_DY.*i.*w;
73
74 %% FFT of the PHI functions to obtain solution
75 Sol_x=real((ifft(fftshift(PHI_DX))))/dt;
76 Sol_y=real((ifft(fftshift(PHI_DY))))/dt;
77
78 %% Same for the velocities
79 %Vel_x=real((ifft(fftshift(PHI_VX))))/dt;
80 %Vel_y=real((ifft(fftshift(PHI_VY))))/dt;
81
82
83 vx=Sol_x;
84 vy=Sol_y;
85
86
87
88
89 %%%%%%
90 %% REFERENCES %%
91 %%%%%%
92 % (1) "WAVE PROPAGATION SIMULATION IN A LINEAR VISCOELASTIC
93 % MEDIUM", Carcione et al. (1988)
94 %
95 % (2) "WAVE FIELDS IN REAL MEDIA: WAVE PROPAGSTION IN
96 % ANISOTROPIC, ANELASTIC
97 % AND POROUS MEDIA", Carcione (2002)
98 %
99 % (3) "THE FINITE DIFFERENCE METHOD FOR SEISMOLOGISTS: AN
100 % INTRODUCTION",
101 % Moczo et al. (2005)
102 %
103 % (4) "INCORPORATION OF ATTENUATION INTO TIME-DOMAIN
104 % COMPUTATIONS OF
105 % SEISMIC WAVE FIELDS", Emmerich & Korn (1987)

```

```

1 function sc=dispcorrection(s,t,dx,dz,kh,r,v,a,dh)
2
3 M=length(a);
4 theta=atan2(dx,dz);
5 radius=sqrt(dx.^2+dz.^2);
6 khx=kh*sin(theta);

```

```

7 khz=kh*cos(theta);
8 S1=0;S2=0;
9 for m=1:M,
10     S1=S1+(a(m)*sin((m-0.5)*abs(khx)));
11     S2=S2+(a(m)*sin((m-0.5)*abs(khz)));
12 end
13
14 D=2.0*asin(r*sqrt((S1.*S1)+(S2.*S2)))./(abs(kh)*r);
15 dc=v*(1.0-D);
16 tshift=radius*dh*((1.0./(v+dc))-(1.0./v));
17
18 ns=length(s);
19 dt=t(2)-t(1);
20
21 f=(-ns/2:ns/2-1)/(dt*ns);
22
23 % Phase correction
24 w=2.0*pi*f;
25 FA=exp(-1i*w.*tshift);
26
27 FA(ns/2+1)=(FA(ns/2+2)+FA(ns/2))/2.0;
28
29
30 s=real(s);
31 S=fftshift(fft(s));
32
33 SC=FA.*S;
34 sc=ifft(ifftshift(SC));
35
36 sc=real(sc);
37
38 % if (size(s,1)~=1)      % Signal was passed in a column
39 %           vector
40 %           sc=sc';
41 % end

```

```

1 function s=Ricker(t,fc,shift)
2 ns=length(t);
3 s=zeros(ns);
4 delay=1.0/fc; ts=t-delay-shift; tau=pi*fc*ts;
5 %s=(1.0-2.0*tau.*tau).*exp(-tau.*tau);
6 % integrated:
7 s=tau.*exp(-tau.*tau);

```

Literaturverzeichnis

- Carcione, J. M. (2014). *Wave Fields in Real Media : Wave Propagation in Anisotropic, Anelastic, Porous and Electromagnetic Media.*, volume Third edition of *Handbook of Geophysical Exploration. Seismic Exploration*. Elsevier Science.
- Hicks, G. J. (2002). Arbitrary source and receiver positioning in finite-difference schemes using kaiser windowed sinc functions. *Geophysics*, 67(1):156–166.
- Koene, E., Robertsson, J., and Andersson, F. (2020). A consistent implementation of point sources on finite-difference grids. *Geophysical Journal International*, 223(2):1144–1161.
- Liu, Y. and Sen, M. K. (2011). Scalar wave equation modeling with time-space domain dispersion-relation-based staggered-grid finite-difference schemes. *Bulletin of the Seismological Society of America*, 101(1):141–159.
- Morse, P. M. and Feshbach, H. (1953). *Methods of theoretical physics*. McGraw-Hill.
- Virieux, J. (1986). P-sv wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51(4):889–901.