

HANDBOOK OF GEOPHYSICAL EXPLORATION

SEISMIC EXPLORATION

VOLUME 1

BASIC THEORY OF EXPLORATION SEISMOLOGY

HANDBOOK OF GEOPHYSICAL EXPLORATION

SEISMIC EXPLORATION

Editors: Klaus Helbig and Sven Treitel

PUBLISHED VOLUMES

- 1984 – Mathematical Aspects of Seismology, 2nd Enlarged Edition
(M. Bäth and A.J. Berkhouit)*
- 1984 – Seismic Instrumentation (M. Pieuchot) ISBN 0-08-036944-8
- 1984 – Seismic Inversion and Deconvolution (a) Classical Methods (E.A. Robinson)*
- 1985 – Vertical Seismic Profiling (a) Principles, 2nd Enlarged Edition (B.A. Hardage)*
- 1987 – Pattern Recognition & Image Processing (F. Aminzadeh)*
- 1987 – Seismic Stratigraphy (B.A. Hardage)*
- 1987 – Production Seismology (J.E. White and R.L. Sengbush)*
- 1989 – Supercomputers in Seismic Exploration (E. Eisner)*
- 1994 – Seismic Coal Exploration (b) In-Seam Seismics (L. Dresen and H. Rüter)*
- 1994 – Foundations of Anisotropy for Exploration Seismics (K. Helbig)
ISBN 0-08-037224-4
- 1998 – Physical Properties of Rocks: Fundamentals and Principles of Petrophysics
(J.H. Schön) ISBN 0-08-041008-1
- 1998 – Shallow High-Resolution Reflection Seismics (J. Brouwer and K. Helbig)
ISBN 0-08-043197-6
- 1999 – Seismic Inversion and Deconvolution (b) Dual-Sensor Technology (E.A. Robinson)
ISBN 0-08-043627-7
- 2000 – Vertical Seismic Profiling: Principles, 3^d Updated and Revised Edition
(B.A. Hardage) ISBN 0-08-043518-1
- 2001 – Seismic Signatures and Analysis of Reflection Data in Anisotropic Media
(I. Tsvankin) ISBN 0-08-043649-8
- 2001 – Computational Neural Networks for Geophysical Data Processing (M.M. Poulton)
ISBN 0-08-043986-1
- 2001 – Wave Fields in Real Media: Wave Propagation in Anisotropic, Anelastic and
Porous Media (J.M. Carcione) ISBN 0-08-043929-2
- 2002 – Multi-Component VSP Analysis for Applied Seismic Anisotropy (C. MacBeth)
ISBN 0-08-0424439-2
- 2002 – Nuclear Magnetic Resonance, Petrophysical and Logging Applications
(K.J. Dunn, D.J. Bergman and G.A. LaTorra) ISBN 0-08-043880-6
- 2003 – Seismic Amplitude Inversion in Reflection Tomography (Y. Wang)
ISBN 0-08-044243-9
- 2003 – Seismic Waves and Rays in Elastic Media (M.A. Slawinski)
ISBN 0-08-043930-6
- 2004 – Quantitative Borehole Acoustic Methods (X. Tang and A. Cheng)
ISBN 0-08-044051-7
- 2004 – Seismic While Drilling – Fundamentals of Drill-Bit Seismic for Exploration
(F. Poletto and F. Miranda) ISBN 0-08-043928-4

* Book out of print.

SEISMIC EXPLORATION

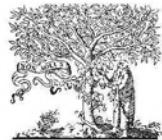
Volume 1

BASIC THEORY OF EXPLORATION SEISMOLOGY

by

John K. Costain and Cahit Çoruh
Department of Geological Sciences
Virginia Polytechnic Institute and State University
Blacksburg, VA, U.S.A.

2004



ELSEVIER

Amsterdam – Boston – Heidelberg – London – New York – Oxford
Paris – San Diego – San Francisco – Singapore – Sydney – Tokyo

ELSEVIER B.V.
Sara Burgerhartstraat 25
P.O. Box 211, 1000 AE
Amsterdam, The Netherlands

ELSEVIER Inc.
525 B Street
Suite 1900, San Diego
CA 92101-4495, USA

ELSEVIER Ltd.
The Boulevard
Langford Lane, Kidlington,
Oxford OX5 1GB, UK

ELSEVIER Ltd.
84 Theobalds Road
London WC1X 8RR
UK

© 2004 Elsevier Ltd. All rights reserved.

This work is protected under copyright by Elsevier Ltd., and the following terms and conditions apply to its use:

Photocopying

Single photocopies of single chapters may be made for personal use as allowed by national copyright laws. Permission of the Publisher and payment of a fee is required for all other photocopying, including multiple or systematic copying, copying for advertising or promotional purposes, resale, and all forms of document delivery. Special rates are available for educational institutions that wish to make photocopies for non-profit educational classroom use.

Permissions may be sought directly from Elsevier's Rights Department in Oxford, UK; phone (+44) 1865 843830, fax (+44) 1865 853333, e-mail: permissions@elsevier.com. Requests may also be completed on-line via the Elsevier homepage (<http://www.elsevier.com/locate/permissions>)

In the USA, users may clear permissions and make payments through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, USA; phone: (+1) (978) 7508400, fax: (+1) (978) 7504744, and in the UK through the Copyright Licensing Agency Rapid Clearance Service (CLARCS), 90 Tottenham Court Road, London W1P 0LP, UK; phone: (+44) 20 7631 5555; fax: (+44) 20 7631 5500. Other countries may have a local reprographic rights agency for payments.

Derivative Works

Tables of contents may be reproduced for internal circulation, but permission of the Publisher is required for external resale or distribution of such material. Permission of the Publisher is required for all other derivative works, including compilations and translations.

Electronic Storage or Usage

Permission of the Publisher is required to store or use electronically any material contained in this work, including any chapter or part of a chapter.

Except as outlined above, no part of this work may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the Publisher.
Address permissions requests to: Elsevier's Rights Department, at the fax and e-mail addresses noted above.

Notice

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein. Because of rapid advances in the medical sciences, in particular, independent verification of diagnoses and drug dosages should be made.

First edition 2004

Library of Congress Cataloging in Publication Data
A catalog record is available from the Library of Congress.

British Library Cataloguing in Publication Data
A catalogue record is available from the British Library.

ISBN: 0-08-037019-5
ISBN CD-ROM: 0-08-044589-6
ISSN: 0950-1401 (Series)

The paper used in this publication meets the requirements of ANSI/NISO Z39.48-1992 (Permanence of Paper).
Printed in The Netherlands.

**Working together to grow
libraries in developing countries**

www.elsevier.com | www.bookaid.org | www.sabre.org

ELSEVIER **BOOK AID**
International Sabre Foundation

Contents

1	Introduction	15
1.1	Acknowledgments	23
2	Groundwork	24
2.1	Complex numbers	24
2.1.1	Manipulation of Complex Numbers	25
2.1.2	Real and Complex Exponentials and Trigonometric Functions	28
2.1.3	Powers and Roots of a Complex Number	30
2.1.4	Logarithm of a Complex Number	31
2.1.5	Functions of a Complex Variable	33
2.1.6	Representation of Signals by Phasors	34
2.1.7	Linear equations	35
3	Fourier Transforms	39
3.1	Introduction	39
3.2	Signal Nomenclature	41
3.2.1	FT—Continuous time, continuous frequency	42
3.2.2	DFS—Continuous time, discrete frequency	43
3.2.3	DTFT—Discrete time, continuous frequency	43
3.2.4	DFT—Discrete time, discrete frequency	45
3.3	The Fourier Coefficients	48
3.3.1	Sign Convention	52
3.3.2	Determination of the Fourier coefficients	55
3.3.3	Fourier Coefficients from Linear Equations	55
3.3.4	Numerical Example	56
3.3.5	Fourier Coefficients from Orthogonality	57
3.3.6	Numerical Example	64
3.3.7	Dirichlet conditions	64
3.3.8	Summary	65
3.3.9	Sign Convention Revisited	74
3.3.10	Seismogram from the Atlantic Coastal Plain	75
3.3.11	Independence of the Fourier coefficients	77
3.4	From Fourier Series to Fourier Integrals	83
3.4.1	Complex forms and Fourier Integral	85

3.4.2	Fast Fourier Transform	89
3.5	Applications of Fourier Transforms	93
3.5.1	Time-shifting theorem	93
3.5.2	Time differentiation of the Fourier transform	97
3.5.3	Time integration of the Fourier transform	98
3.5.4	Introduction to the unit-impulse function	102
3.5.5	The sinc function	107
3.5.6	Definition of a linear System	109
3.5.7	Impulse Response	109
3.5.8	The time-convolution theorem	111
3.5.9	Autocorrelation and Crosscorrelation	120
3.5.10	The frequency-convolution theorem	123
3.5.11	Hilbert Transforms	129
3.6	<i>z</i> -transform	163
3.6.1	Factors of a finite, discrete function	165
3.6.2	Phase of minimum and maximum delay couplets	168
3.6.3	Amplitude and phase of a <i>z</i> -transform	169
3.6.4	Introduction to filters	175
4	Computational Considerations	185
4.1	Effect of Analysis Window on Fourier Spectrum	185
4.2	Aliasing	188
4.2.1	Sampling in the time domain – aliasing in the frequency domain	189
4.2.2	Example	190
5	Synthetics and Velocity Functions	195
5.1	Normal-incidence reflection coefficient	195
5.1.1	Example	199
5.2	Values of Reflection Coefficients	201
5.3	The Zoeppritz Equations	204
5.4	AVO and Zoeppritz equations in <i>T-X</i> domain	208
5.5	Synthetic Seismograms	214
5.6	The Reflectivity Function	218
5.7	Velocity functions	219
5.7.1	Impulse (thin bed)	221
5.7.2	Step	221
5.7.3	Ramp	223
5.7.4	Wavelet Tuning	231
5.7.5	Summary of velocity functions	238
5.8	Seismic trace attributes	242
6	Traveltime curves and velocity	253
6.1	Snell's law	254
6.1.1	The ray parameter <i>p</i>	257
6.2	Reflection Traveltime Curves	258

6.2.1	The root-mean-square velocity	266
6.2.2	Determination of interval velocities using Dix Equation	271
6.2.3	Effect of dip on reflection traveltimes curves	286
6.2.4	The normal moveout correction	290
6.3	Refraction traveltimes curves	293
6.3.1	Refractions from a single horizontal interface	296
6.3.2	Delay time	298
6.3.3	General expression for delay time	302
6.3.4	Two-layer model	302
6.3.5	General expression for multilayer refraction traveltimes curves from horizontal layers	303
6.3.6	Reflection and refraction traveltimes curves combined— comments	305
6.3.7	Effect of dip on refraction traveltimes curves	305
6.3.8	The Principle of Reciprocity	311
6.3.9	Refraction traveltimes curves over various geologic models .	312
6.3.10	Dipping plane interfaces	317
6.3.11	Linear increase in velocity with depth	322
6.3.12	Turning waves	325
6.4	Composite Refraction-Reflection Stacks	325
7	Seismic Source Wavelets	338
7.1	Energy Sources	342
7.1.1	Dynamite	342
7.1.2	Vibroseis	342
7.1.3	DinoSeis, Thumper, and others	352
7.1.4	Marine	352
7.2	Mathematical Descriptions of Wavelets	354
7.2.1	Ricker wavelet	355
7.2.2	Klauder wavelet	355
7.2.3	Comments	357
7.3	Wavelet z -transform representation	358
7.3.1	Physical requirements for real wavelets	359
7.3.2	A simple 2-point wavelet	361
7.3.3	Generation of wavelets	363
7.3.4	Partial energy of a wavelet	366
7.3.5	Roots plotted in the z -plane	367
7.3.6	Root on the unit circle	368
8	Wavelet Shaping and Deconvolution	370
8.1	Inverse infinite filters, finite input	373
8.1.1	Inverse filtering of a minimum-delay 2-term wavelet using z -transforms	379
8.1.2	Inverse filtering of a maximum-delay 2-term wavelet using z -transforms	384
8.1.3	Inverse filtering of a seismic trace using z -transforms . .	392

8.2	Inverse finite filters, infinite input	394
8.2.1	Exact filters for wave guides	394
8.3	Inverse filters and input each of finite length	418
8.3.1	General shaping and least-squares method	420
8.3.2	Predictive deconvolution	454
8.3.3	What does predictive deconvolution do?	480
8.3.4	Summary guidelines for predictive deconvolution	497
8.3.5	Predictive deconvolution—Conclusion	499
8.4	Spectral whitening	507
8.5	Further Applications of Hilbert Transforms	514
8.5.1	Relationship between the amplitude and phase spectrum of a causal function	514
8.5.2	Q	518
References		549
Index		564

List of Figures

1.1	Seismic record section	16
1.2	Crustal profile	18
1.3	Seismic data cube	19
1.4	Seismic record section—Coal	20
2.1	Graphical representation of a complex number	25
2.2	Polar coordinates	28
2.3	Roots of a complex number	32
3.1	Vibroseis seismogram obtained by the Virginia Tech seismic crew on the Atlantic Coastal Plain near Richmond, VA	40
3.2	Fourier series (DFS) application	45
3.3	A discrete-time signal	45
3.4	Relation between angular frequency and linear frequency	49
3.5	Frequencies required for a periodic series	50
3.6	Dirichlet conditions	66
3.7	Gibbs phenomena	67
3.8	Time-symmetric sine wave	68
3.9	Time-symmetric cosine wave	68
3.10	Sum of time-symmetric sine and cosine waves	69
3.11	Early shot record from Bellshill Lake, Alberta	70
3.12	A seismic trace	71
3.13	Fourier decomposition and synthesis: Rectangular pulse	73
3.14	Fourier decomposition and synthesis: Triangular pulse	74
3.15	Atlantic Coastal Plain seismic data	76
3.16	Fourier transform of infinite and windowed sinusoid	78
3.17	Fourier harmonics are independent of each other	79
3.18	Fourier harmonic interference	80
3.19	Effect of width of analysis window	81
3.20	Effect of shape of analysis window	82
3.21	Frequencies required for a periodic series	84
3.22	FFT procedure	94
3.23	Effect of time shift on the Fourier spectrum	96
3.24	Integrated seismic trace	99
3.25	A snapshot in time during wavelet integration	100
3.26	Comparison of trace integration	101

3.27	Spectra of differentiated and integrated Klauder wavelet	102
3.28	Seismic trace and its derivative	102
3.29	Integrated seismic trace	103
3.30	Spectra of integrated Trace 18	103
3.31	Temperature log and the geothermal gradient—effect of differentiation	104
3.32	Rectangular pulse centered about $t = 0$	105
3.33	Derivation of frequency content of Dirac delta function	105
3.34	A few cosine waves before summing to get Dirac delta function	107
3.35	Approximation to Dirac delta function by summation of a few cosine waves	107
3.36	Graphic display of derivation of frequency response of Dirac delta function	108
3.37	Relation between period of rectangular pulse and spacing of null points in the Fourier transform of the pulse	108
3.38	Definition of a linear system	109
3.39	Response of linear system to an impulse	110
3.40	Impulse response of a filter	110
3.41	Convolution caution	115
3.42	The reflection process	118
3.43	Source wavelet, reflection coefficients, and seismic trace	119
3.44	The Coherence Cube	122
3.45	Convolution in the frequency domain	127
3.46	Recovery of true spectrum is not possible	128
3.47	What is a Hilbert transform?	130
3.48	Why 90° ?	131
3.49	What if $h(t)$ is not causal?	133
3.50	Phasor rotation convention for Hilbert transform	134
3.51	Overview of Hilbert transform symmetry	135
3.52	Reconstruction of causal function from even part	137
3.53	Hilbert transform of sinusoid (a)	139
3.54	Hilbert transform of sinusoid (b)	140
3.55	Fourier transform of sign(t)	143
3.56	Hilbert transform of a continuous function using $-i \operatorname{sign}[\omega]$	144
3.57	Fourier transform of $1/\pi t$	145
3.58	Hilbert transform of sine wave using Hilbert transformer	149
3.59	Hilbert transforms of cosine wave - frequency domain	150
3.60	Graphic display of quadrature filter	151
3.61	Hilbert transform of continuous data by swapping	154
3.62	Hilbert transform of discrete data by swapping	155
3.63	$H_e(\omega)$ from $H_o(\omega)$	160
3.64	Continuous phase spectrum	162
3.65	Unit circle in the z-plane	165
3.66	Frequency response of $P(z) = -z_1 + z$	170
3.67	Complex conjugate roots	171
3.68	Single root plotted in z-plane	172

3.69	Unwrapped phase of maximum-delay wavelet	173
3.70	Vector plot of $P(z) = -z_1 + z$	173
3.71	Plot of minimum-phase $P(z) = -z_1 + z$	174
3.72	Frequency response of an averaging filter	177
3.73	Frequency response of a differencing filter	177
3.74	Simple notch filter	180
3.75	Simple low-pass filter	182
3.76	Butterworth low-pass filter	183
4.1	Fourier transform of complex exponential	186
4.2	Effect of truncation on Fourier spectrum	187
4.3	Effect of width of analysis window on Fourier spectrum	187
4.4	Effect of shape of analysis window on Fourier spectrum	188
4.5	Two sinusoids showing why aliasing occurs	191
4.6	Example of an aliased spectrum	192
4.7	How to check for aliasing	194
5.1	Velocities and densities of Atlantic Coastal Plain sediments . .	202
5.2	Domain of real reflection coefficients from Zoeppritz equations .	205
5.3	Angles of reflection and transmission for incident P -wave . .	206
5.4	Zoeppritz equations for various angles of incidence and velocity ratios	207
5.5	AVO common midpoint geometry and energy partition	210
5.6	Example 1 of amplitude variation with offset	213
5.7	Example 2 of amplitude variation with offset	214
5.8	Peterson Model for synthetic seismograms	216
5.9	Reflection coefficients obtained directly from impedance contrast versus approximation from derivative of $\text{Log}(v)$	220
5.10	The only three types of velocity functions needed to model a transversely isotropic earth	222
5.11	The thin bed velocity function	224
5.12	Thin-bed detection by integrated energy spectra	225
5.13	Analytic convolution of source wavelet with step function . .	227
5.14	The ramp velocity function	228
5.15	Integration of a source wavelet by a ramp velocity function .	229
5.16	Ratio of Fourier transform of differentiated source wavelet to transform of source wavelet	230
5.17	Integration of source wavelet by triangular velocity function .	231
5.18	Ricker wavelet—50 Hertz	233
5.19	Wavelet differentiation	234
5.20	Wavelet interaction as a differentiation	234
5.21	Effect of wavelet differentiation on amplitude spectrum	235
5.22	Analytically differentiated Ricker wavelet	235
5.23	Effect of layer thickness on reflection amplitude	236
5.24	Wavelet interaction as an integration	236
5.25	Effect of wavelet integration on amplitude spectrum	237

5.26	Analytically integrated Ricker wavelet	237
5.27	Ricker wavelet (80 Hz)	238
5.28	Velocity functions $w(t)$ from Sengbush et al. [161]	239
5.29	Velocity functions $w(t)$ and their transforms	241
5.30	Seismic attribute—quadrature trace	248
5.31	Seismic attribute—quadrature trace	249
5.32	Seismic attributes—trace, instantaneous phase and frequency	250
5.33	Seismic attributes—trace, envelopes	251
5.34	Hilbert transform of vibrator sweep	252
6.1	Physical significance of Snell's law	254
6.2	Derivation of Snell's law	255
6.3	Another application of Snell's law	257
6.4	Snell's law extended downward along a raypath	274
6.5	Apparent velocity	275
6.6	Emergence angle of wavefront and slope of traveltime curve	275
6.7	Hyperbolic travelttime curve from single interface	276
6.8	Reflection travelttime curves from 2-layer model	277
6.9	Notation for an n-layer reflection model	278
6.10	Travelttime curves from a 5-layer model	279
6.11	Travelttime curves from a 5-layer model with low-velocity layer	279
6.12	Wide-angle reflections to 10 km from 4-layers with low-velocity layer	280
6.13	Ray parameter versus offset for reflections from multilayer model	280
6.14	Ray parameter versus offset for reflections from model with uniform velocity change	281
6.15	Determination of accurate reflection traveltimes	281
6.16	Fictitious straight-line v_{rms} reflection raypath versus actual path	282
6.17	Comparison of travelttime computations using exact parametric equations, the rms velocity, and the third term in the power series expansion	283
6.18	Another comparison of travelttime computations using exact parametric equations, Equation (6.15), and the third term in the power series expansion	284
6.19	RMS velocity (ft/sec) distributions in major oil-producing sediments	284
6.20	Trace gather for an v_{rms} velocity search	285
6.21	Velocity spectrum	285
6.22	Velocity panel used to determine stacking velocities	286
6.23	Geometry for derivation of travelttime curve from a single dipping reflector	287
6.24	Effect of dip on the reflection travelttime curve from a single dipping interface	287
6.25	Effect of dip on reflection moveout	289
6.26	Normal moveout correction applied to reflections from zero-dip reflector	291

6.27	Normal moveout correction applied to 20° dip	292
6.28	Reflection and refraction arrivals	294
6.29	Refraction raypath geometry for single interface	297
6.30	Real-data examples of refraction head waves	299
6.31	The concept of delay time	300
6.32	Delay time for single segment in third layer of n-layer model	301
6.33	Three-layer refraction model	304
6.34	Reflection and refraction traveltimes curves combined	306
6.35	Down-dip refraction traveltimes over single dipping interface	307
6.36	Up-dip refraction traveltimes over single dipping interface	309
6.37	Reversed refraction traveltimes curves over single dipping interface	311
6.38	First arrival data for reversed refraction profile of Figure 6.37	313
6.39	Reversed refraction profile over the Price Mountain anticline	314
6.40	Reversed refraction profiles over vertical fault to surface	315
6.41	Reversed refraction profile over fault of large throw	316
6.42	Reversed refraction profile over fault of small throw	317
6.43	Reversed refraction profiles over buried high-velocity layer and buried vertical fault	318
6.44	Adachi's derivation of refraction traveltimes curves	319
6.45	Raypath geometry for linear change of velocity with depth	322
6.46	Geometry of refraction raypath for linear increase of velocity with depth	324
6.47	Traveltimes curve for linear increase in velocity with depth	324
6.48	Turning rays for normal reflection and for reflections from salt overhang	326
6.49	Refraction stacks—refraction moveout 1	329
6.50	Refraction stacks—refraction moveout 2	329
6.51	Refraction stacks—multifold refraction raypaths	330
6.52	Refraction stacks—dipping refractor	332
6.53	Refraction stacks—composite refraction-reflection stack	334
6.54	Refraction stacks—comparison with reflection	335
6.55	Refraction stacks—two refractors, same line	335
6.56	Refraction stacks—shallow comparison with reflection	336
7.1	Oscillatory character of a general pulse	339
7.2	Minimum-delay wavelet	340
7.3	Geophone arrays	341
7.4	Vibroseis method	343
7.5	Vibroseis method	344
7.6	Failing Y-1100A vibrator	345
7.7	Shear vibrator	346
7.8	Example of processing single-sweep vibrator data	347
7.9	Single-vibrator over crystalline terrane	349
7.10	Tapered versus untapered vibroseis sweep	350
7.11	Single-vibrator over Paleozoic shelf strata	351
7.12	Single-vibrator on Atlantic Coastal Plain sediments	353

7.13	Marine seismic source	354
7.14	Ricker wavelet Fourier transform pair	355
7.15	Klauder wavelet Fourier transform pair	357
7.16	Minimum-delay wavelet from impulsive source	359
7.17	Real seismic wavelet decays to zero	362
7.18	Modulus of $e^{-i\omega}$ is radius of unit circle.	363
7.19	Single root and its complex conjugate	368
7.20	Root of z -transform on unit circle	369
8.1	Complex 2-term wavelet	383
8.2	Positive-imaginary complex minimum-delay 2-term wavelet	384
8.3	Negative, real, minimum-delay 2-term wavelet	384
8.4	Inverse filtering of complex, maximum-delay 2-term wavelet	389
8.5	Plot of inverse-filter coefficients for 2-term wavelet	390
8.6	Complex time-domain equi-delay 2-term wavelet	391
8.7	Complex minimum-delay 2-term wavelet with root close to the unit circle	391
8.8	How to generate a synthetic seismic trace	393
8.9	Water-layer wave guide	394
8.10	The “effective source wavelet”	395
8.11	Response of wave guide, thickness 11 m	397
8.12	Response of wave guide, thicknesses of 15, 30, 60 meters	399
8.13	Effect of seismometer depth on wavelet character	400
8.14	Response of wave guide for water depths of 3, 6, and 9 meters	401
8.15	Water reverberations—Persian Gulf	401
8.16	Water reverberation spectra—Persian Gulf	402
8.17	Backus 2-point filter for long-period water-confined reverberations	404
8.18	Backus 2-point filter for short-period water-confined reverbera- tions	405
8.19	Backus 3-point filter for long-period deep reflection reverberations	409
8.20	Backus 3-point filter for short-period deep reflection reverbera- tions	410
8.21	Neidell's 2-layer reverberation model	411
8.22	Neidell's 2-layer reverberation model - Example 1	412
8.23	Neidell's 2-layer reverberation model - Example 2	419
8.24	Neidell's 2-layer model with convolved wavelet	419
8.25	Crosscorelation sign convention	425
8.26	Procedure for obtaining positive-lag crosscorrelation coefficients	426
8.27	Procedure for obtaining negative-lag crosscorrelation coefficients	427
8.28	Minimum-delay wavelet shaping by least-squares filter	429
8.29	Correlations using Figure 6-2 from Robinson and Treitel	430
8.30	More correlations from Robinson and Treitel Figure 6-2	434
8.31	Example (c) of Robinson and Treitel	436
8.32	Example (d) from Robinson and Treitel	437
8.33	Wavelet shaping using least-squares filter and maximum-delay input	438

8.34	Example (a) from Figure 6-2 of Robinson and Treitel	440
8.35	Wavelet shaping of mixed-delay input	441
8.36	Example from Figure 6-3 of Robinson and Treitel	444
8.37	Wavelet and roots of mixed-delay wavelet	445
8.38	Figure 6-3 of Robinson and Treitel[153] with correlations added	446
8.39	Figure 6-3 of Robinson and Treitel[153] with correlations and shift added	447
8.40	Inverse filtering of Vibroseis signal	452
8.41	Shaping of vibroseis wavelet	453
8.42	Predictive deconvolution of minimum-delay wavelet	456
8.43	Autocorrelation of wavelet and seismic trace	459
8.44	Comparison of wavelet and trace autocorrelations	460
8.45	Input and desired output for predictive deconvolution	466
8.46	Prediction filter output— 1	468
8.47	Prediction filter output— 2	469
8.48	Predictive deconvolution matrix comparison	474
8.49	Minimum-delay wavelet shaping by predictive deconvolution .	476
8.50	Spiking deconvolution of a minimum-delay wavelet. Prediction distance $\alpha = 1$. (PredictiveDeconvolution.nb → PredictiveDeconvolutionWavelets.cdr → PDmin2.wmf)	478
8.51	Predictive deconvolution of longer minimum-delay wavelet . .	479
8.52	Effect of predictive deconvolution on wavelet	480
8.53	Deconvolution to a “hanging wavelet”	483
8.54	Second zero-crossing	484
8.55	Second zero crossing?	485
8.56	Predictive deconvolution applied to non-minimum-delay wavelets	486
8.57	Predictive deconvolution of seismic trace	487
8.58	Trace energy after predictive deconvolution	488
8.59	Effect of design window width on predictive deconvolution .	490
8.60	Observable effects of wavelet truncation	491
8.61	An alternative to spiking deconvolution	492
8.62	Alpha versus thin beds	494
8.63	Effect of noise on predictive deconvolution	501
8.64	Removal of 1 st order long-period multiples by predictive deconvolution	502
8.65	Removal of 2 nd order long-period multiples	503
8.66	Removal of 1 st order short-period multiples	504
8.67	Backus filter coefficients from z-transforms and predictive decon	505
8.68	Autocorrelation of reverberations	506
8.69	Spectral separation by stretching	509
8.70	Model data after spectral separation by stretching	510
8.71	SAGC with and without amplitude balancing	511
8.72	Application of SAGC on real data	512
8.73	Spectral whitening of vibroseis data	513
8.74	Phase spectra of minimum-, mixed-, and maximum-delay wavelets	517
8.75	Absorption definition	522

8.76	Experimental data underlying linear-with-frequency assumption	523
8.77	Absorption coefficient α versus frequency ν	523
8.78	Absorption without dispersion	526
8.79	Dispersive velocity and Hilbert transform	531
8.80	Dispersion curve in Pierre Shale	536
8.81	Body wave dispersion curves for various values of Q	537
8.82	Discrete versus continuous formulation of dispersion	538
8.83	What if absorption is not linear-with-frequency?	540
8.84	Absorption impulse response of the earth for $Q = 20$, $a = 1500$ m	541
8.85	Comparison of absorptive impulse response computed from Gladwin and Stacey [78] with that from Equation (8.127)	542
8.86	Wavelet spreading for $Q = 30$	544
8.87	Wavelet shape change for $Q = 20$	548
8.88	Klauder wavelet shape change for $Q = 20$ at distance 100 m	548

List of Tables

3.1	Notation for the Fourier transform (FT)	42
3.2	Notation for discrete Fourier series (DFS)	44
3.3	Notation for the Discrete Time Fourier Transform (DTFT)	46
3.4	Notation for the Discrete Fourier Transform (DFT)	46
3.5	Fourier Transform Pairs	126
3.6	Examples of minimum, maximum, and mixed delay functions and their z -transforms	168
5.1	Velocities and densities of Paleozoic shelf strata	203
5.2	Notation for conversion of Zoeppritz equations to time-offset domain	211
5.3	Trigonometric relations for AVO forward modeling	212
7.1	Examples of 2-term wavelets	362
7.2	Examples of convolution of binomials	365
8.1	Summary of equations for absorption-dispersion pairs	539

This page is intentionally left blank

Chapter 1

Introduction

The material in this volume provides the basic theory necessary to understand the principles behind imaging the subsurface of the Earth using reflection and refraction seismology. The end products are a “record section” such as that shown in Figure 1.1 and, from a collection of “wiggly traces” that are recorded in the field (see Figure 3.1 on page 40 for an example of a wiggly 42-trace seismogram), derived information about subsurface structure and rock properties. For the most part, the principles of imaging are the same regardless of the depth to the target; i.e., the same mathematical background is necessary for targeting a shallow water table as for investigating the deep Mohorovitchic discontinuity (the “Moho”), which marks the bottom of the earth’s “crust” at a depth of 30-50 km.

It is generally accepted that of all the geophysical techniques reflection seismology provides the best resolution and the most information about the subsurface. It is also the most expensive method, and for this reason has not yet been widely applied to help define the geometry of near-surface aquifer systems; however, the potential is there. This method of exploring the subsurface has been developed and highly refined by the petroleum industry in the search for hydrocarbons onshore and offshore.

The images of the subsurface obtained by reflection seismology must generally be refined and sharpened before a geologic interpretation can be made, especially for data acquired on land. In this regard, Fourier theory is important in order to understand the physical meaning of the seismic “wiggly trace”, and the collections of these that we refer to as a “record section”.

Practical details of reflection or refraction data acquisition or processing are not included in this volume nor are field procedures and techniques of proper data acquisition; however, they are all equally important. The reader is referred to the excellent treatments of data acquisition and processing in this Series and elsewhere. Benyamin [16] provided *Mathematica* programs in his tutorial on key elements of total field design. A good image requires proper acquisition; the image can often be further improved (made more meaningful) by data processing. The basic theory that underlies such techniques is included in this volume.

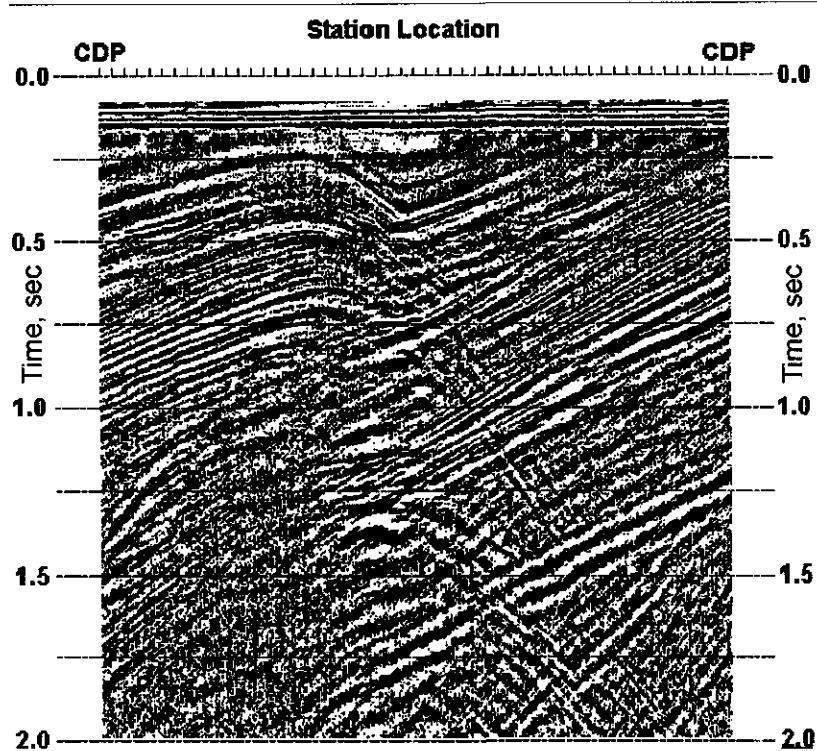


Figure 1.1: A 2-dimensional display of multi-trace seismic data called a “record section”. Such data are now acquired in such a way as to produce the 3-dimensional cube of data shown in Figure 1.3. Seismic data courtesy of Paradigm Geophysical.

We summarize the mathematical concepts required to understand reflection seismology data sets used in exploration geophysics. Because the authors have a combined experience of almost 100 years in reflection seismic multifold data acquisition, processing and interpretation, and because we operated our own field crew at Virginia Tech for ten years, we are able to include some results of our own applications of the theory discussed in this volume. Much of our research has been directed toward imaging the shallow and deep crust down to the Mohorovicic discontinuity with the objective of understanding the internal architecture and structural evolution of the Appalachian orogen in the eastern United States. We include illustrations of seismic images, many of which we generated ourselves, of the major geologic provinces of the Appalachian orogen. The Alleghanian plateau, folded Valley and Ridge, overthrust (allochthonous) Blue Ridge, the metamorphic core of the orogen (the Piedmont, Figure 1.2), and the younger sediments of the Atlantic Coastal Plain that conceal the eastward

extension of the exposed Piedmont, yield images of a complex tectonic history that are a necessary complement to the surface boundary conditions provided by geologic maps. We hope that including portions of seismic record sections of the internal geometry of a major orogen in this kind of a book on “basic theory” will help to convey the excitement that accompanies the merger of the disciplines of geology and geophysics. Although geologists and geophysicists work with completely different kinds of *basic field data*, they share a common interest and enthusiasm in understanding how these data collectively contribute to our understanding of the tectonic evolution of mountain belts.

In viewing a two-dimensional seismic “record section” or a three-dimensional volume of seismic data (Figure 1.3) it is desirable to have some appreciation of how different mathematical operations effect the basic data acquired in the field. The fundamental seismic “unit” is the seismic source wavelet that leaves the source point. The wavelet is reflected or refracted from the subsurface geometry and returns to the surface where it is recorded. After much data processing, the arrival times of reflected wavelets and the nature of the distortion of the recorded wavelets yield the seismic record section and the properties of the rocks that the wavelets have traversed.

This is not a book on processing techniques. There are many references (Yilmaz, [204]) to this subject including the massive two-volume series by Yilmaz [203]. Seismic data processing is a dynamic and changing field with new processing techniques continually emerging. In the course of interpreting reflection data acquired over the Appalachian orogen in the southeastern United States. Peavy [135] showed the advantages of introducing geology early in the processing sequence by summing reflection points along the direction of tectonic strike. In this volume we wish to illustrate more general topics such as how the shape of the source wavelet can be affected by, for example, “body wave dispersion” caused by intrinsic damping, and how we can use the Hilbert transform as a special application of Fourier theory to examine the effect that body wave dispersion has on changes in wavelet shape. Some appreciation of how certain velocity distributions in common geologic models can behave as mathematical operators to change the shape of the seismic wavelet somewhere between the source and receiver is useful to a geologist as well as a geophysicist. What effects does the geology have on the shape (frequency and phase) and propagation of the seismic wavelet because of the earth materials? In this volume we review reasons why only three simple “velocity functions” are required to generate *any* synthetic sequence of reflection coefficients, no matter how complicated the geology (page 219).

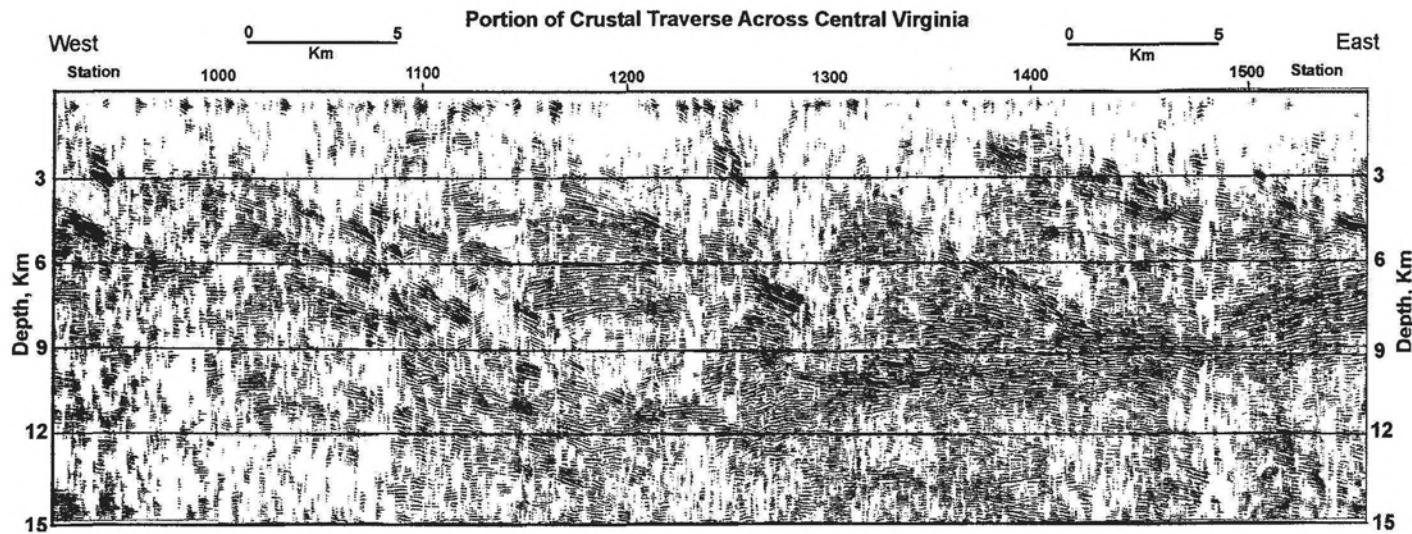


Figure 1.2: Portion of a conventional multifold (24-fold) seismic reflection line across the Appalachian orogen in central Virginia. The seismic signature of the arcuate/eastward-dipping folded and thrust-faulted reflectors is continuous along strike for over 400 km from Virginia to Georgia [50]. Crustal velocity is 6 km/sec. These structures cradle most of the earthquakes in the Central Virginia Seismic Zone. From Çoruh et al. [45], Pratt et al. [139], and Lampshire et al. [104]. Data acquired by the U.S. Geological Survey and reprocessed by Pratt [138].

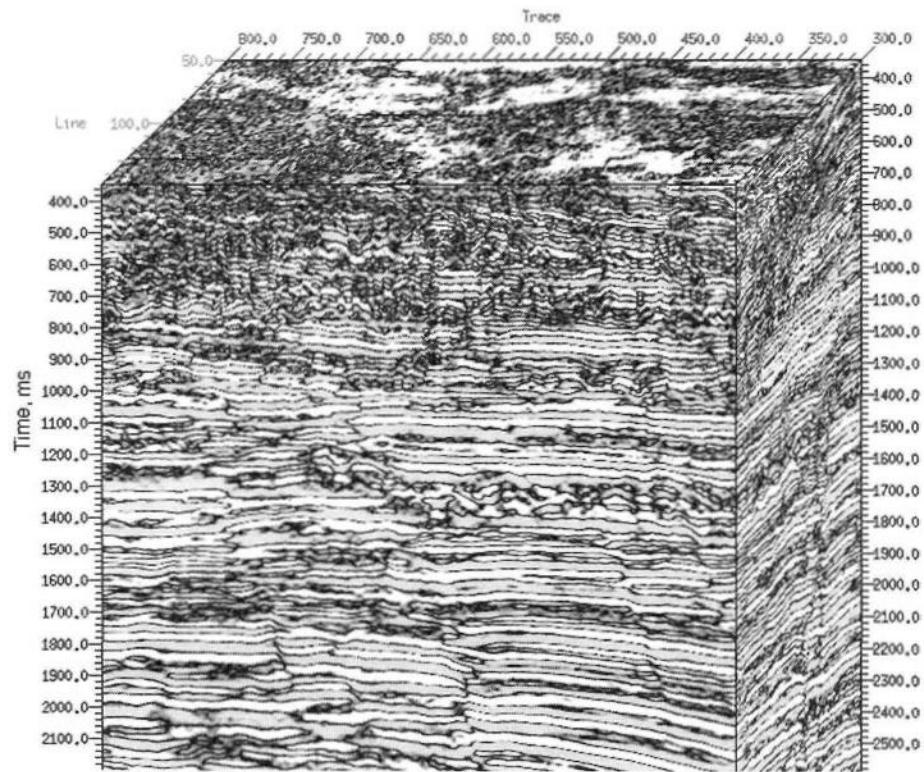


Figure 1.3: Modern seismic data acquisition and processing has matured to become a remarkably successful 3-dimensional effort. The 3-dimensional “cube” of data shown in this figure is the end product of 3-D data acquisition and processing. Data courtesy of Exxon.

The material in this volume is appropriate for shallow geophysical applications such as coal exploration (Figure 1.4) as well as for deeper studies. Indeed, the best recent images of ground penetrating radar (GPR) are the result of processing such shallow data in a manner similar to that used for tectonic studies and hydrocarbon applications. It is just a question of scale.

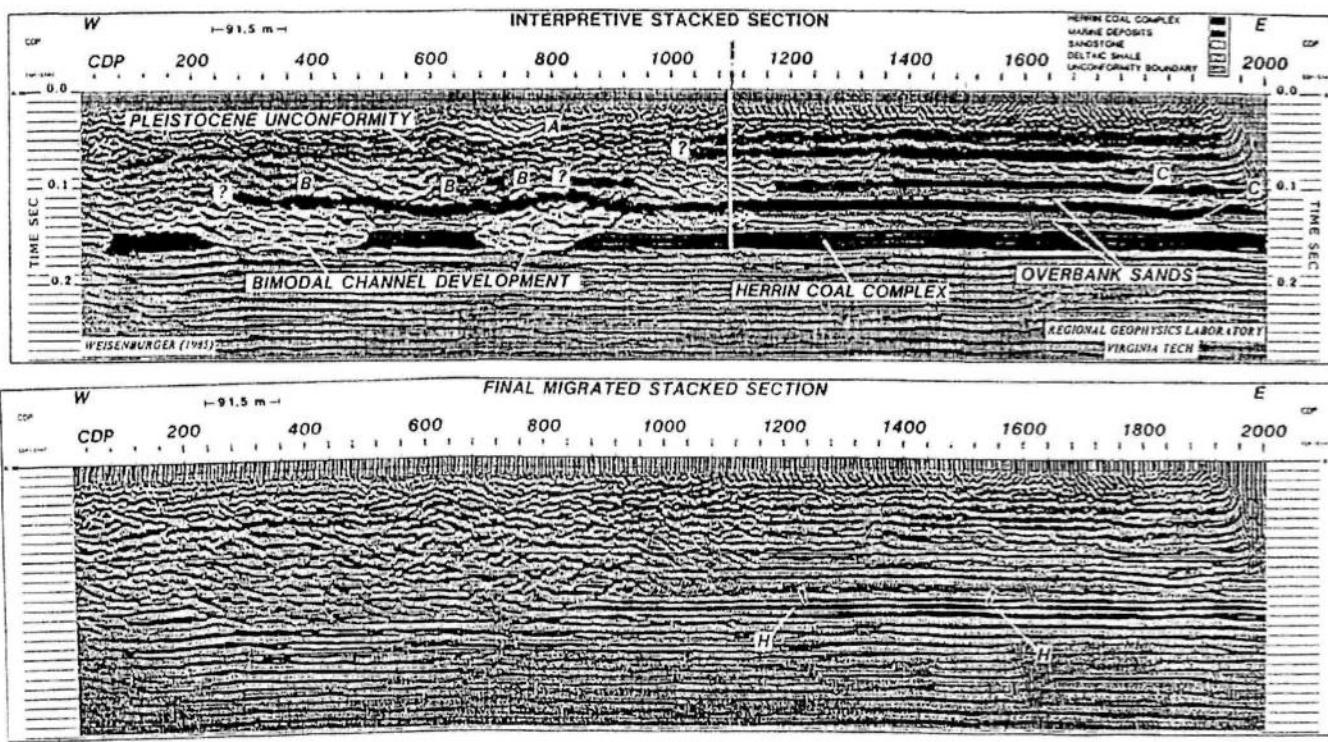


Figure 1.4: Top: An interpreted record section over shallow coal deposits in the Illinois Basin showing bimodal paleochannel development at about 150 ms two-way traveltimes in the shallow Herrin Coal Complex. The Herrin No. 6 seam, the major target in the area, is at a depth of about 160 m and is 2 to 3 m in thickness. Interruptions in seam thickness by paleochannel development have a direct bearing on the economic feasibility of mining operations. Bottom: Uninterpreted section. Field data courtesy of Conoco, Inc. Reprocessed and interpreted by Weisenburger [188].

Most of the material in this book has been offered in courses for geophysics majors at Virginia Tech. A prerequisite to these courses is a course commonly referred to as “Engineering Mathematics” where basic concepts such as complex numbers, vectors, differential equations, and Fourier theory are covered. We begin this volume with a brief summary of these subjects.

Fourier theory is used extensively in geophysics to analyze and process seismic and other kinds of data. The examples given in this volume apply the theory developed in the text. The intent of the presentation is to make the reader “computer literate” rather than “icon literate.” There is an advantage to having a background that enables one to work and think independently, to know both the theory and how to develop and implement algorithms. You can be sure of what someone else’s programs do, and be better positioned to test and challenge canned software. More important, when you get an idea about something theoretical or practical, you will be able to test it yourself. The philosophy behind the intensive use of programming by our students to illustrate and apply the lecture material can be summarized by the following:

Tell me and I’ll forget.

Show me and I may remember.

Involve me and I’ll understand . . . Chinese proverb

The elaborate and beautifully efficient commercially available icon-oriented software for production processing of reflection seismic data is often not the simplest way to develop an idea or educate oneself about the inner workings of what happens when you click on an icon. It is hoped that this volume will contribute to an understanding of what lurks beneath some of the icons. The courses taught by the authors at Virginia Tech required a background in Fortran programming, which in the last couple of years migrated to *Mathematica* [198].

Mathematica, first released in 1988, is a fully integrated environment for technical computing. Individual packages have long existed for specific numerical, algebraic, graphical and other tasks but the concept of *Mathematica* was to create a single system that could handle all the various aspects of technical computing in a coherent and unified way. *Mathematica* combines symbolic manipulation, numerical mathematical methods, outstanding graphics output suitable for publication, and a sophisticated programming language. For a user with even a modest background in programming, the transition from Fortran (or from anything else) to *Mathematica* is not difficult. Even without such a background, however, the learning curve is not that steep.

As a versatile scientific and technical computing language *Mathematica* has been chosen for this volume in order to focus the reader’s attention on concepts rather than programming. The programs are provided in the form of *Mathematica* statements and notebooks, which are easy to follow and adapt to a user’s needs. Complete *Mathematica* notebooks together with plot statements are on the included CD-ROM in the hope that the reader will explore and extend the examples given, and invent new ones. Readers can download a free *Mathematica* Notebook Reader for the computer platform they use (MathReader at

<http://www.wolfram.com/products/mathreader/>). In order to change the programs, however, a licensed version is necessary. This volume occasionally uses the *Mathematica* add-on package called *Signals and Systems* in order to facilitate some computations and generate a few of the plots. *Signals and Systems* requires a separate license.

The earth changes the shape of the seismic wavelet as it propagates away from the source. For example, wavelet shape is changed by virtue of its encounter with certain kinds of subsurface velocity distributions, which reflect back differentiated or integrated versions of itself. *Mathematica* movies of these processes are included on the CD-ROM to aid the reader in understanding how this works.

There is some advantage to writing Volume 1 in this Handbook of Geophysical Exploration for Seismic Exploration and at the same time being among the last to turn in the manuscript for publication. It gives us the opportunity to refer here to topics that are covered in greater detail in other volumes of this same Seismic Exploration series and we have attempted to point the reader to this more advanced material in this Series as well as elsewhere.

The manuscript for this volume was prepared using the L^AT_EX [103] software system, which is based on Donald Knuth's T_EX [98] language. Both were implemented using the PCTeX for Windows [88] software, Version 5.1.

John K. Costain
Cahit Çoruh

Blacksburg, Virginia
August 2004

1.1 Acknowledgments

Many funding agencies supported our research over the decades including the National Science Foundation, the Nuclear Regulatory Commission, the U.S. Geological Survey, Virginia Division of Mineral Resources, Maine Geological Survey, Sohio, Chevron, Southeastern Exploration and Production Corporation (SEPCO), and many others. From 1979-88 we operated a full-time Virginia Tech vibroseis data acquisition crew. In 1981-83, Sohio was able to take advantage of any available crew time. Ditto for SEPCO and Chevron. Each industry collaboration gave us a little more insight into the local or regional geology. From 1982-84 we formed the Virginia Tech Vibroseis Consortium (VTVC) and thought up projects that would be of interest to us as well as to industry and governmental subscribers. Our non-vibroseis "Thin-Bed Consortium" focused on the theoretical aspects of thin bed detection and the seismic response expected from sequences of thin beds as well as the potential of high resolution seismic data to examine and reinterpret the depositional environment of coal. Some of the results of this research are included in this volume.

Virginia Tech was the first university in the United States to install the then state-of-the-art DISCO (Digicon Interactive Seismic COnputer) processing software. Prior to this all of our seismic data processing was done on the university's IBM mainframe using homemade software written entirely in Fortran. This important turning point toward "icon-based" commercial reflection seismic data processing software has today evolved into the sophisticated processing and interpretation industry software now available at Virginia Tech. In the classroom, however, we also used the mainframe and Fortran so that our geophysics majors would be "computer literate", and not just "icon literate".

We gratefully acknowledge the many contributions to geophysics of Enders Robinson and Sven Treitel. Their tutorial and insightful imprint on the literature has hopefully been successfully passed along by us to our students. Their influence should be evident in many parts of this volume. John Costain takes pride in revealing that, in the jungles of Venezuela with a Socony-Mobil of Venezuela seismic crew, he taught Sven Treitel everything he knows about cable splicing and repair. Indeed, John is certain that it was this thought-provoking challenge in the early 50s that planted the seeds of intuition that now blossom as Sven's productive excursions into neural networks and their applications to geophysics. (In the world we live in you don't have to be right, you just have to be certain.)

Teaching undergraduate and graduate courses and doing research with students and faculty colleagues is one of the most intellectually stimulating environments around. We thank our students at the University of Utah (JKC: 1960-67), Virginia Tech (JKC: 1967-96 and CC: 1979-2004), Istanbul University (CC: 1976-83), and Istanbul Teknik University (CC: 1984-85) for all the discussions, for their enthusiasm, and for sharing their ideas. We are grateful to the many students who took our courses and it is to them that this volume is dedicated.

Chapter 2

Groundwork

Complex numbers are needed to describe and understand many aspects of exploration seismology. The study of forced vibrations due to a dynamic system, the behavior of a linear system, and the processing of data in the time and frequency domains all involve the use of complex numbers. A short summary of complex numbers and complex variables is therefore considered to be groundwork for this volume.

2.1 Complex numbers

A complex number is a quantity given in the form

$$z = x + iy$$

where x and y are real numbers and “ i ” is defined as

$$\begin{aligned} i &= \sqrt{-1} \\ i^2 &= -1 \end{aligned}$$

In this definition x is the real part of the complex number z and y is the imaginary part. Both x and y are real numbers. The notation for the real part $\text{Re}[]$ and imaginary part $\text{Im}[]$ of a complex number z is defined as

$$\begin{aligned} x &= \text{Re}[z] \\ y &= \text{Im}[z] \end{aligned}$$

respectively.

In a Cartesian coordinate system, complex numbers can be plotted with the real part of the complex number on the x -axis (the abscissa) and the imaginary part of the complex number (the ordinate) on the y -axis. In this definition the x -axis is called the axis of reals and the y -axis is called the axis of imaginaries. Such a two-dimensional space is known as the complex plane.

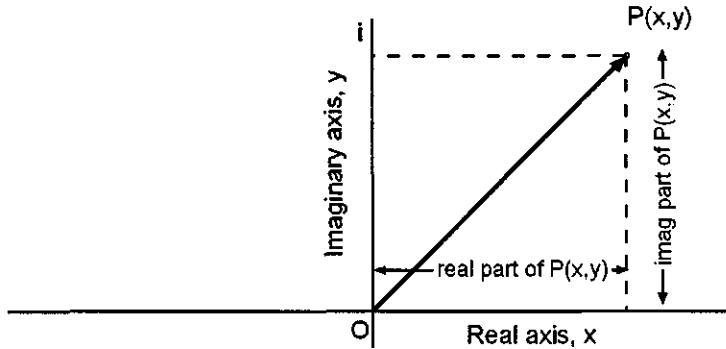


Figure 2.1: Graphical representation of the complex number, $P(x,y) = x + iy$. Projection of OP onto x -axis is the real part of the complex number $P(x,y)$. Projection of OP onto the y -axis is the imaginary part of the complex number. Distance $OP = \sqrt{x^2 + y^2}$ is called the modulus of the complex number. $i = \sqrt{-1}$. The vector defined by OP is called a *phasor*. Groundwork.cdr → GraphOfComplexNumber.wmf

A point P is shown in Figure 2.1 in the complex plane as defined by the coordinates of x and y . The x -axis corresponds to the axis of real numbers. The y -axis corresponds to the axis of imaginary numbers. The value z of the complex number located at the point P is defined as

$$z = x + iy$$

Any number can be considered to be a complex number. For example, a real number may be defined as a complex number for which the imaginary part of the complex number is zero. Correspondingly, an imaginary number is a complex number whose real part is zero. Therefore, the complex number system is a more general definition of a number system than the real number system.

There are two fundamental rules for the manipulation of complex numbers:

1. A complex number $z = x + iy$ is equal to zero if $x = 0$ and $y = 0$.
2. Complex numbers obey the ordinary rules of algebra. Note that $i^2 = -1$.

2.1.1 Manipulation of Complex Numbers

Using the above rules we give the rules for addition and multiplication of complex numbers. If $z_1 = x_1 + iy_1$, and $z_2 = x_2 + iy_2$ then the sum z is

$$\begin{aligned} z = z_1 + z_2 &= (x_1 + iy_1) + (x_2 + iy_2) \\ &= (x_1 + x_2) + i(y_1 + y_2) \end{aligned}$$

where the real parts have been added together and the imaginary parts have been added together to give the new complex number z . This is the addition

rule. The subtraction rule is given similarly

$$\begin{aligned} z = z_1 - z_2 &= (x_1 + iy_1) - (x_2 + iy_2) \\ &= (x_1 - x_2) + i(y_1 - y_2) \end{aligned}$$

The rule of multiplication is

$$\begin{aligned} z = z_1 z_2 &= (x_1 + iy_1)(x_2 + iy_2) \\ &= x_1 x_2 + iy_1 x_2 + iy_2 x_1 + i^2 y_1 y_2 \\ &= (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1) \end{aligned}$$

From the multiplication rule it is clear that if C is a real number

$$\begin{aligned} z = Cz_1 &= C(x_1 + iy_1) \\ &= Cx_1 + iCy_1 \end{aligned}$$

The quotient of the complex numbers z_1 and z_2 is given by

$$z = \frac{z_1}{z_2} = \frac{x_1 + iy_1}{x_2 + iy_2} = \frac{x_1}{x_2 + iy_2} + i \frac{y_1}{x_2 + iy_2} \quad (2.1)$$

If two complex numbers such as $z = x + iy$ and $z^* = x - iy$ differ only in the sign of their imaginary parts, they are called “complex conjugates” or simply “conjugates”. Thus $z^* = x - iy$ is called the conjugate of $z = x + iy$. If $z = x + iy$ and its conjugate is $z^* = x - iy$, then

$$\begin{aligned} z z^* &= (x + iy)(x - iy) \\ &= x^2 + y^2 \end{aligned}$$

This states that multiplication of a complex number with its conjugate results in a positive real number except when $z = 0$. From this result we define the absolute value of a complex number z by the non-negative square root of $(z z^*)$. Thus, the magnitude (absolute value) of z is

$$|z| = \sqrt{z z^*}$$

Using the above rules of manipulation of complex numbers the following can be stated. If $z = x + iy$:

$$\begin{aligned} (z^*)^* &= z \\ z + z^* &= 2 \operatorname{Re}(z) = 2x \\ z - z^* &= 2i \operatorname{Im}(z) = 2iy \\ |z^*| &= |z| \end{aligned}$$

If $z_1 = x_1 + iy_1$ and $z_2 = x_2 + iy_2$, then

$$\begin{aligned} (z_1 + z_2)^* &= z_1^* + z_2^* \\ (z_1 z_2)^* &= z_1^* z_2^* \\ |z_1 z_2|^2 &= |z_1|^2 |z_2|^2 \end{aligned}$$

In signal processing $z^{-1} = 1/z$ is important. Thus, if $z = (x + iy) \neq 0$

$$\begin{aligned}\frac{1}{z} &= \frac{z^*}{z z^*} \\ &= \frac{z^*}{|z|^2}\end{aligned}$$

or

$$\begin{aligned}\frac{1}{z} &= \frac{x - iy}{x^2 + y^2} \\ &= \frac{x}{x^2 + y^2} - i \frac{y}{x^2 + y^2}\end{aligned}$$

Using this result any quotient given as z_1/z_2 can be represented by multiplication of z_1 and $1/z_2$. Therefore

$$\begin{aligned}z_1 \frac{1}{z_2} &= (x_1 + iy_1) \left\{ \frac{x_2}{x_2^2 + y_2^2} - i \frac{y_2}{x_2^2 + y_2^2} \right\} \\ &= \frac{x_1 x_2 + y_1 y_2}{x_2^2 + y_2^2} + i \frac{x_2 y_1 - x_1 y_2}{x_2^2 + y_2^2}\end{aligned}$$

If we introduce polar coordinates then from Figure 2.2 we can write

$$\begin{aligned}x &= r \cos \theta \\ y &= r \sin \theta\end{aligned}$$

where r is the length of the vector representing the point P and θ is the angle between the real axis and the vector to P and is called the “phase angle”. The vector defined by OP in Figure 2.2 or 2.1 is called a *phasor*. In polar coordinates, the complex number $z = x + iy$ may be given by the phasor

$$\begin{aligned}z &= r(\cos \theta + i \sin \theta) \\ &= r \cos \theta + i r \sin \theta \\ &= x + iy\end{aligned} \tag{2.2}$$

which represents the polar form of the complex number.

The number r is always positive and is called the “modulus” of the complex number z . The modulus of the complex number z that gives the length of the line OP in Figure 2.2 is equal to the absolute value of the complex number. Thus

$$r = |z| = \sqrt{x^2 + y^2}$$

The coordinate θ is called the argument of the complex number z or the phase of the complex number z and is defined as

$$\begin{aligned}\theta = \arg z &= \tan^{-1} \left[\frac{y}{x} \right] \\ &= \cos^{-1} \left[\frac{x}{r} \right] \\ &= \sin^{-1} \left[\frac{y}{r} \right]\end{aligned} \tag{2.3}$$

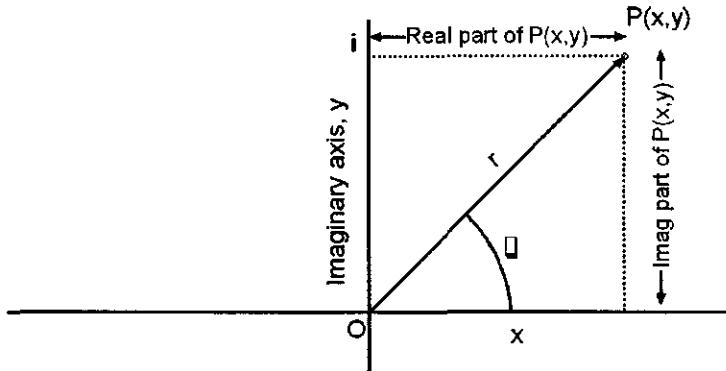


Figure 2.2: Components of the complex number $P(x,y)$ expressed in terms of polar coordinates r and θ . $x = r \cos \theta$; $y = r \sin \theta$. The vector defined by OP in this figure or Figure 2.2 is called a *phasor*.

2.1.2 Real and Complex Exponentials and Trigonometric Functions

From a Maclaurin expansion we have,

$$\begin{aligned}\sin(\theta) &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots \\ \cos(\theta) &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \dots \\ e^\theta &= 1 + \theta + \frac{\theta^2}{2!} + \frac{\theta^3}{3!} + \frac{\theta^4}{4!} + \dots \\ e^{i\theta} &= 1 + i\theta - \frac{\theta^2}{2!} - \frac{i\theta^3}{3!} + \frac{\theta^4}{4!} + \dots\end{aligned}$$

If we combine $\cos(\theta)$ and $\sin(\theta)$ as real and imaginary parts, respectively, and then do a series expansion we get

$$\begin{aligned}\cos \theta + i \sin \theta &= 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \frac{(i\theta)^4}{4!} + \dots \\ &= 1 + i\theta - \frac{\theta^2}{2!} - \frac{i\theta^3}{3!} + \frac{\theta^4}{4!} + \dots\end{aligned}$$

and thus we see that

$$e^{i\theta} = \cos \theta + i \sin \theta \quad (2.4)$$

an equation that Nobel Laureate Richard Feynman called [71, Volume 1, p. 22-10] “the most remarkable formula in mathematics”. Similarly

$$e^{-i\theta} = \cos \theta - i \sin \theta \quad (2.5)$$

From Equations (2.4) and (2.5) it follows that

$$\cos \theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

and

$$\sin \theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

These are known as Euler (pronounced “oiler”) formulas that show relations between exponential and trigonometric functions.

From the multiplication of series

$$e^x e^y = e^{x+y}$$

and therefore for a complex exponent we can write

$$\begin{aligned} e^{x+iy} &= e^x e^{iy} \\ &= e^x (\cos y + i \sin y) \end{aligned}$$

If we use Equations (2.2) and (2.4) it is possible to write

$$\begin{aligned} z &= x + iy \\ &= r(\cos \theta + i \sin \theta) \\ &= re^{i\theta} \end{aligned}$$

The polar form $z = re^{i\theta}$ makes convenient the multiplication or division of complex numbers. For example, if

$$z_1 = r_1 e^{i\theta_1}$$

and

$$z_2 = r_2 e^{i\theta_2}$$

then

$$z_1 z_2 = |r_1 r_2| e^{i(\theta_1 + \theta_2)} \quad (2.6)$$

That is, the multiplication of two complex numbers is carried out by multiplication of their moduli to get the modulus of the product and addition of their phases to get the phase of product. See Equation (5.35) in the discussion of seismic attributes for an application of (2.6) in the computation of “instantaneous frequency.”

Similarly if we perform division we get

$$\frac{z_1}{z_2} = \frac{r_1}{r_2} e^{i(\theta_1 - \theta_2)}$$

In this case we divide the moduli and subtract the phases.

Multiplication of two phasors corresponds to a convolution process while division of two phasors corresponds to a “deconvolution”. The fundamentals of convolution and deconvolution are discussed in Section 3.5 and Chapter 8, respectively.

2.1.3 Powers and Roots of a Complex Number

A simple way to determine the power of a complex number is to use the polar form of the complex number. If

$$z = r e^{i\theta}$$

then

$$z^n = (r e^{i\theta})^n = r^n e^{in\theta}$$

for any integer n . This is known as DeMoivre's theorem. When the modulus is equal to unity ($r = 1$) we obtain

$$\begin{aligned} (e^{i\theta})^n &= (\cos \theta + i \sin \theta)^n \\ &= \cos(n\theta) + i \sin(n\theta) \end{aligned}$$

This equation can be used to determine the formulas for $\sin 2\theta$, $\cos 2\theta$, $\sin 3\theta$, etc.

In a more general form, the complex number z may be given as

$$\begin{aligned} z &= r [\cos(\theta + 2k\pi) + i \sin(\theta + 2k\pi)] \\ &= r e^{i(\theta+2k\pi)} \end{aligned}$$

where k is any integer. This states an important feature of complex numbers. Any integer multiple of 2π can be added to the phase θ without altering the value of the complex number z ; the location of the point P in Figure 2.2 will not change.

If n is a positive integer then the complex number $z^{1/n}$ (the n th root of z) can be given as

$$\begin{aligned} z^{1/n} &= r^{1/n} e^{i(\theta+2k\pi)/n} \\ &= r^{1/n} \left[\frac{\cos(\theta + 2k\pi)}{n} + i \frac{\sin(\theta + 2k\pi)}{n} \right] \end{aligned} \quad (2.7)$$

The n th root of $z^{1/n}$ is a complex number whose n th power is z . There are in principle n possible values. The *Mathematica* software (and most humans) chooses a particular *principal root*. For example, the 3rd power of the complex number $(-1 + i\sqrt{3})$ is 8. *Mathematica*, however, will return the value 2 (the principal root) if you look up $8^{1/3}$. We can, though, obtain n distinct values of $z^{1/n}$ by giving k in Equation (2.7) integer values from 0 to $(n - 1)$. From Equation (2.7) for $n = 3$ and $k = 0$

$$z^{1/3} = r^{1/3} e^{i\theta/3} \quad (2.8)$$

So to find the polar coordinates of the cube root of the complex number $r e^{i\theta}$ find the principal root of r and divide the angle θ by 3. So the polar coordinates

of $z = 8^{1/3}$ are

$$r = 2 \quad (2.9)$$

$$\begin{aligned} \theta &= 0^\circ, 360^\circ/3, 720^\circ/3, 1080^\circ/3, \dots \\ &= 0^\circ, 120^\circ, 240^\circ, 360^\circ, \dots \end{aligned} \quad (2.10)$$

and referring to (2.8), (2.9) and (2.10) and knowing there are three distinct roots z_i then as input to *Mathematica* we have

$$\begin{aligned} z_1 &= 2 \cos(0) + i \sin(0) \\ z_2 &= 2 \cos\left(120 \frac{\pi}{180}\right) + i \sin\left(120 \frac{\pi}{180}\right) \\ z_3 &= 2 \cos\left(240 \frac{\pi}{180}\right) + i \sin\left(240 \frac{\pi}{180}\right) \end{aligned}$$

as shown in Figure 2.3. The value of z is the same for $\theta = 0^\circ$ and $\theta = 360^\circ$ so there are n distinct roots as k ranges from 0 to $n - 1$. A *Mathematica* program is

```
Solve[{z1 == 2(Cos[0] + I Sin[0]),
       z2 == 2(Cos[120*Pi/180] + I Sin[120*Pi/180]),
       z3 == 2(Cos[240*Pi/180] + I Sin[240*Pi/180])}, {z1, z2, z3}]
```

and the output is

$$\{\{z1 \rightarrow 2., z2 \rightarrow -1. + i \sqrt{3}, z3 \rightarrow -1. - i \sqrt{3}\}\}$$

as shown in Figure 2.3.

When the phase θ of the phasor is equal to zero, then $z = r = 1$. Therefore, in the general polar form with k an integer,

$$1 = e^{i(0+2k\pi)} = e^{i(2k\pi)}$$

and the root $r^{1/n}$ can be given as

$$1^{1/n} = e^{i(2k\pi/n)}$$

or

$$1^{1/n} = \cos(2k\pi/n) + i \sin(2k\pi/n)$$

for $k = 0, 1, 2, \dots, n - 1$.

From this result we see that the n roots of unity are located around the unit circle in the complex plane.

2.1.4 Logarithm of a Complex Number

In general if $z = e^\omega$ then

$$\omega = \ln z$$

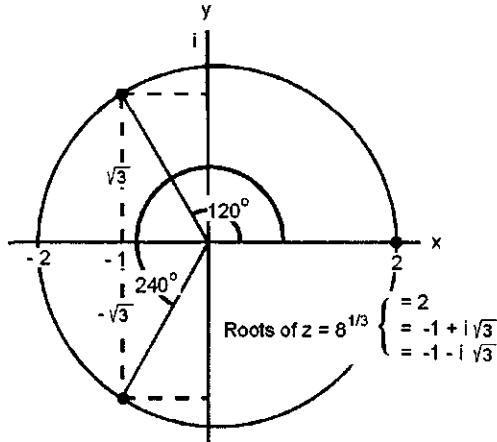


Figure 2.3: The three roots of $8^{1/3}$ plotted on a circle of radius $|z| = 8^{1/3} = 2$ and spaced 120° apart. cuberoot.nb \Rightarrow cuberoot.cdr \Rightarrow cuberoot.wmf

Using this definition properties of the logarithmic function may be given as follows.

$$\begin{aligned}\ln(z_1 z_2) &= \ln z_1 + \ln z_2 \\ \ln\left(\frac{z_1}{z_2}\right) &= \ln z_1 - \ln z_2 \\ \ln(z^n) &= n \ln z\end{aligned}$$

The logarithm of a complex number is another complex number, the real part of which is the logarithm of the modulus of z and the imaginary part is equal to the phase of z . If

$$\begin{aligned}z &= x + iy \\ &= r(\cos \theta + i \sin \theta) \\ &= re^{i\theta}\end{aligned}$$

then

$$\begin{aligned}\ln z &= \ln(re^{i\theta}) \\ &= \ln r + \ln e^{i\theta} \\ &= \ln r + i\theta\end{aligned}\tag{2.11}$$

For an application of (2.11) see page 514 and the discussion of recovering the phase spectrum θ of an unknown wavelet shape if only the modulus of the amplitude spectrum is known.

If $r = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}[y/x]$ then

$$\ln z = \frac{1}{2} \ln(x^2 + y^2) + i \tan^{-1}\left(\frac{y}{x}\right)$$

Here it is illustrated how to find the logarithm of a negative real number. Find the logarithm of -1 with the help of the unit phasor. If the phase angle is equal to π then

$$-1 = 1 e^{i\pi}$$

By taking the logarithm of this expression one gets

$$\ln(-1) = \ln(1) + i\pi$$

Since $\ln(1) = 0$ then

$$\ln(-1) = i\pi$$

Similarly, the logarithm of any real negative number $(-a)$ may be given as

$$\ln(-a) = \ln(a) + i\pi$$

In the general polar form, if

$$z = r e^{i(\theta+2k\pi)}$$

for $k = 0, \pm 1, \pm 2, \pm 3, \dots$, then the logarithm of the complex number is given by

$$\ln z = \ln r + i(\theta + 2k\pi) \quad \text{for } k = 0, \pm 1, \pm 2, \pm 3, \dots \quad (2.12)$$

According to Equation (2.12) the logarithm of any number, real or complex, can assume an infinite number of (complex) values, all with the same modulus but with different phase angles. This concept arises in the definition of the stability of dynamic systems and leads to the concept of minimum phase (see Section 3.5.11 on page 156).

For each value of z in (2.12) there is an infinite number of values for the function $\ln z$; however, if θ is allowed to vary only from 0 to 2π then the function $\ln z$ has just one value for each value of z . Such a function is called "single-valued" and is referred to as a *branch* of $\ln z$. When expressions such as $\ln z$ or \sqrt{z} are used in this volume, we are always referring to a single branch so that the function $\ln z$ or \sqrt{z} is "single-valued". From a practical standpoint, when we attempt to determine the minimum-phase spectrum of an unknown wavelet from its amplitude spectrum (Section 3.5.11), a computer is generally instructed to give results for the interval $0 \rightarrow 2\pi$, the *principal angle* θ where

$$\theta = \text{principal angle} + 2k\pi$$

2.1.5 Functions of a Complex Variable

Functions operating on the complex variable $z = x + iy$ are called functions of a complex variable. Functions of a complex variable are complex numbers of the form $u + iv$ where u and v are real functions of x and y . Thus, functions of a complex variable may be given as

$$w(z) = u(x, y) + iv(x, y)$$

The function $w(z)$ must have a definite derivative for a given value of the complex variable z . Following are the conditions that the function $w(z)$ must satisfy in order to have a unique derivative $w'(z)$:

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (2.13)$$

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad (2.14)$$

These equations are known as the Cauchy-Riemann equations and define the conditions that the real and imaginary parts of $w(z)$ must satisfy.

The Cauchy-Riemann equations state that if the real and imaginary parts of a function have continuous first derivatives and satisfy the equations in (2.14) then the function is called analytic. An *analytic signal* is a complex time-domain function that is analytic (see also the mathematical definition of a Hilbert transform in Section 3.5.11 on page 142). Hilbert transforms are discussed in more detail on page 129.

2.1.6 Representation of Signals by Phasors

As stated earlier, the vector defined by the line OP in the complex plane is called a phasor, as shown in Figure 2.2. The modulus and phase are called the elements of the phasor.

When the modulus r and the phase θ are real constant numbers, the phasor defined by $z = r e^{i\theta}$ is a fixed vector; however, in seismic data representation and analysis, as in most cases, the phasor is usually moving. It is a function of time. Under certain special conditions, the modulus is assumed to remain constant (see for example Figure 3.59 on page 150); in most cases the modulus, too, is a function of time. In the most general case, therefore, the phasor $z(t)$ is expressed as

$$z(t) = r(t) e^{i\theta(t)} \quad (2.15)$$

In this representation the length of the phasor is changing with time along with the time dependent phase. In some cases the length of the phasor may be constant while the phase changes with time as in

$$z(t) = r e^{i\theta(t)} \quad (2.16)$$

where r is a constant. With these definitions it is clear that amplitude modulated signals may be described by (2.15) whereas frequency modulated signals may be analyzed with (2.16).

In the case of digitized data, the moving and rotating phasors may be given, respectively, by

$$z(n\Delta t) = r(n\Delta t) e^{i\theta(n\Delta t)}$$

and

$$z(n\Delta t) = re^{i\theta(n\Delta t)}$$

where Δt is the sampling interval and n is a counter for the number of samples. Δt is commonly set = 1 time unit and therefore not explicitly shown in equations. The use of phasors or complex numbers makes it easy to understand the systems we often use.

2.1.7 Linear equations

Many problems in wave propagation require the solution of n equations in n unknowns. For example, plane-wave reflection coefficients are determined by solving a set of four linear equations called the *Zoeppritz equations* (Chapter 5.3).

Consider two equations in two unknowns f_a and f_b :

$$\begin{aligned} a_a f_a + a_b f_b &= g_a \\ a_b f_a + a_a f_b &= g_b \end{aligned}$$

where a_a , a_b , g_a , and g_b are known constants. Multiplication of the first equation by a_a and the second by a_b and then subtracting the second from the first gives

$$f_a = \frac{g_a a_a - g_b a_b}{a_a a_a - a_b a_b}$$

Similarly,

$$f_b = \frac{g_b a_a - g_a a_b}{a_a a_a - a_b a_b}$$

If we write

$$\begin{vmatrix} a_a & a_b \\ a_b & a_a \end{vmatrix} = a_a a_a - a_b a_b \quad (2.17)$$

the solutions for f_a and f_b can be written

$$f_a = \frac{\begin{vmatrix} g_a & g_b \\ a_b & a_a \end{vmatrix}}{\begin{vmatrix} a_a & a_b \\ a_b & a_a \end{vmatrix}}$$

and

$$f_b = \frac{\begin{vmatrix} a_a & a_b \\ g_a & g_b \end{vmatrix}}{\begin{vmatrix} a_a & a_b \\ a_b & a_a \end{vmatrix}}$$

Each of the scalar functions of the square arrays is called a *determinant*.

A 2×2 determinant is defined to be

$$\det \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

An array with n rows and n columns is called a determinant of order n . Determinants are always *square*; the number of rows is equal to the number of columns. The *elements* of the determinant are the constants a, b, c , and d . The most immediate and one of the most important applications of the theory of determinants is to solve systems of simultaneous linear equations, which can be used to describe Fourier theory (Chapter 3) and deconvolution (Chapter 8).

A determinant always has a *value*. The value of a two by two determinant is obtained by taking the product of the coefficients from upper left to lower right minus the product of the coefficients taken from upper right to lower left. A faster way is to let *Mathematica* do it using the built-in function called **Det** (or do it by hand for a 2×2 array). For example,

$$m = \{\{a, b\}, \{c, d\}\}$$

The answer returned by *Mathematica* using **Det[m]** is

$$-bc + ad$$

To determine the value of a 3^{rd} order determinant $|a_{ij}|$, delete any one row and any one column from the determinant. The row and column intersect at element a_{ij} . The remaining determinant is denoted M_{ij} and is called the *minor* of the element a_{ij} . The *cofactor* of the element a_{ij} is defined as $-1^{i+j} M_{ij}$. To determine the value of a 3rd order determinant, multiply each element of *one* column (or *one* row) by its cofactor and add the results. It makes no difference which row or column is used. This method of evaluation is called *Laplace's Development*. For the determinant $m = \{\{a, b, c\}, \{d, e, f\}, \{g, h, i\}\}$ written as

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

the value is

$$-ceg + bfg + cdh - afh - bdi + aei$$

A determinant of order n is written in the form:

$$\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

where the first and second subscripts are row and column indices, respectively. The notation $|a_{ij}|$ means the determinant whose elements are a_{ij} . In general,

a_{ij} does not equal a_{ji} , but there are important applications in seismic data processing for which this is the case. Evaluation of a determinant of order n is only slightly more complicated. If the determinant is of 4th order or higher, use the Laplace development until only determinants of second order remain. Then use Equation (2.17).

The following facts apply to determinants:

1. The value of a determinant is zero if
 - (a) all the elements in any row or in any column are zero,
 - (b) two rows or two columns are proportional to each other.
2. If each element of a single row or a single column of a determinant is multiplied by a constant, the value of the determinant is multiplied by the same constant.
3. The value of a determinant is multiplied by -1 if two rows, or two columns, are interchanged.
4. The value of a determinant is unchanged if
 - (a) rows are written as columns, and columns as rows,
 - (b) to each element of one row is added a constant times the corresponding element of another row. A similar statement holds for columns.

A *matrix*, like a determinant, is an ordered array of elements; however, a determinant has a single value. A matrix has no value, and is simply an ordered arrangement of n by m elements.

Although determinants are always square, the number of rows and columns in a matrix is not necessarily equal. A *row matrix* is a matrix consisting of a single row. A *column matrix* is a matrix with a single column. A *square matrix* has the same number of rows and columns. If all of the elements not on the principal diagonal are zero, the matrix is called a *diagonal matrix*. A diagonal matrix with equal elements on the principal diagonal is a *scalar matrix*. If the values of the elements on the principal diagonal of a scalar matrix are equal to unity, the matrix is an *identity* or less commonly a *unit* [201] matrix.

A *symmetric matrix* is a square matrix for which $a_{ij} = a_{ji}$. If $a_{ij} = -a_{ji}$, the square matrix is *skew symmetric*. A *null* or *zero* matrix is one in which each element $a_{ij} = 0$. Two matrices are equal if and only if they are identical, i.e., if they have the same number of rows, the same number of columns, and if $a_{ij} = a_{ji}$.

The addition (or subtraction) of two or more matrices with the same number of rows and columns is commutative and associative. The elements in the resulting matrix are simply the sums or differences of the elements in the corresponding locations in the given matrices.

One system of equations commonly encountered in least-squares filtering (Section 8.3.1) and predictive deconvolution (Section 8.3.2) is the set of "normal

equations" that is symmetric, positive definite, and *Toeplitz* as follows

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix}$$

where the coefficients r_i are *autocorrelation* coefficients determined from the input data, g_i crosscorrelation coefficients determined from the input data and user-defined desired output, and f_i are the unknowns (see Section 8.3.1). The symmetry of this system leads to a relatively fast solution by a recursive process described by Levinson [111]. The method is coded in Robinson [148, p. 44] as a Fortran subroutine called EUREKA. This and other Fortran subroutines have been converted to *Mathematica* modules in Appendix 2 of Robinson and Osman [152]. In the present volume systems of linear equations are solved by *Mathematica*'s built-in `LinearSolve` function, which requires no separate function in the *Mathematica* notebook but might be slower for some applications. We have found it to be satisfactory for systems consisting of over 1000 equations. See page 462 and Figure 8.45 for applications of `LinearSolve` to prediction and predictive deconvolution.

The addition of matrices is commutative and associative [201]. That is, for any matrices A , B , and C

$$\begin{aligned} A + B &= B + A \\ A + (B + C) &= (A + B) + C \end{aligned}$$

Multiplication is not defined for matrices that are not conformable.

Multiplication of matrices is distributive over addition. That is, if B and C are matrices that have the same number of rows and the same number of columns, and if A is a matrix conformable to B and C , then

$$A(B + C) = AB + AC$$

Multiplication of matrices is associative. For matrices that are suitably conformable,

$$A(BC) = (AB)C$$

In general, multiplication of matrices is not commutative. Even for matrices A and B that are conformable in either order. In general then,

$$AB \neq BA$$

If the product of two matrices is zero then this does not imply that one of the matrices is zero. For example [201, p. 582]

$$\begin{pmatrix} 6 & 4 & 2 \\ 9 & 6 & 3 \\ -3 & -2 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 & -2 \\ -1 & 0 & 3 \\ 2 & -3 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Chapter 3

Fourier Transforms

3.1 Introduction

Fourier theory is widely used in geophysics to analyze seismic as well as other kinds of geophysical data. The objectives of this chapter are to illustrate how and why Fourier theory is applied to geophysical data. Fourier theory can be applied to seismic data acquired over shallow targets, i.e., coal, the water table, etc., as well to data intended to image deeper targets; i. e., thrust decollements at depths of 9-10 km, duplexes in shelf strata, crustal imbrications formed in crystalline rocks, hydrocarbon deposits, the Moho, etc. Many of the basic mathematical tools required to sharpen the image of these and other geologic targets are found in the subject of Fourier transforms.

A shallow vibroseis seismogram is shown in Figure 3.1. The target to be imaged is a reflection from crystalline basement, which arrives at the receivers most distant from the energy source at a two-way traveltimes of about 220 ms. The continuity of the reflection is interrupted by Rayleigh surface wave noise, and the reflection cannot be traced across the seismogram. The noise is lower in frequency content. If we can determine which frequencies are associated with the surface waves and which are associated with the reflection signal, then we might be able to attenuate the amplitudes of those frequencies that are associated with the surface wave noise. We could zero out what we believe to be the noisy Fourier components, then do an inverse Fourier transform and return to the time domain. If the surface waves have indeed been attenuated then the continuity of the reflection from basement might be improved. The discussion that follows is concerned with the procedure for defining and computing the Fourier coefficients (transforms) that define the amplitude spectrum for each trace and so permit their modification.

This example of filtering is, of course, not the only application of Fourier transforms, but it is an instructive way to begin our discussion. The above simple example assumes that the frequencies associated with the noise are different than those associated with the signal. Under these conditions the attenuation

of the noise is relatively straightforward. No matter what the signal processing application, however, Fourier transforms play an important role.

We view seismograms in the *time domain*, and examine Fourier coefficients in the *frequency domain*. The mathematics of *Fourier transformation* is used to get from one domain to the other and back. Thus,

$$\begin{array}{ccc} \text{data in} & \left\{ \begin{array}{c} \text{Fourier transformation} \\ \Leftrightarrow \end{array} \right\} & \text{data in} \\ \text{time domain} & & \text{frequency domain} \end{array}$$

Wherever the symbol \Leftrightarrow appears in this volume it means the data on each side are a Fourier transform pair.

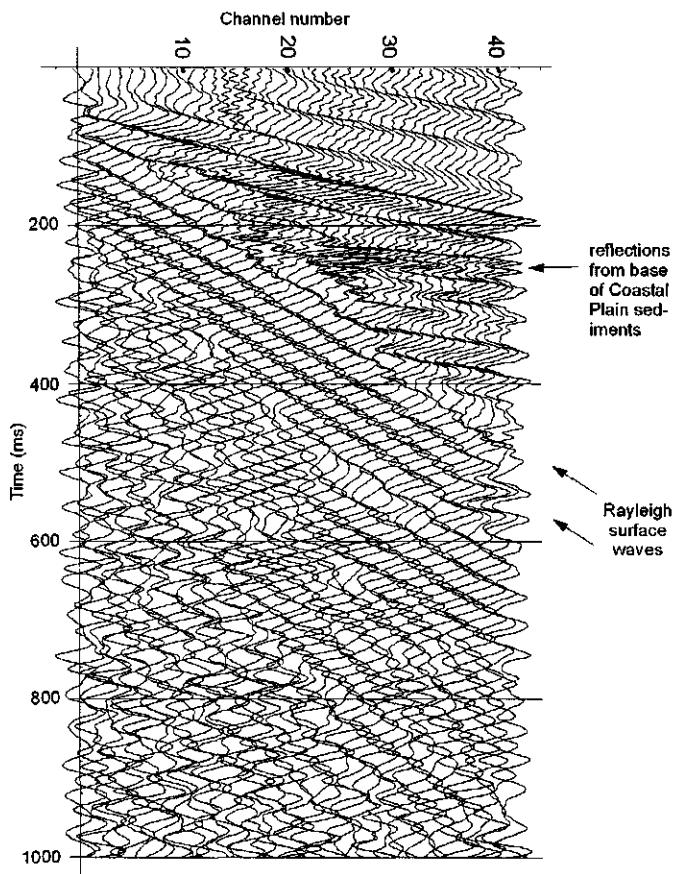


Figure 3.1: Rayleigh waves obscure the reflections from crystalline basement on a typical “wiggly trace” display of a seismogram. (cd → AtlanticCoastalPlain → ACPShotRecord.wmf)

3.2 Signal Nomenclature

A signal is a description of how one data parameter depends upon another parameter that is usually related to the domain of observation. For observational seismic data, one of the parameters is time and the other is the ground velocity or displacement at that time. Note that given any particular recording time the value of ground motion at that time can always be found, but the converse is not true. Time, therefore, is called the independent variable and the value of the ground velocity or displacement amplitude at that time is the dependent variable. Signals can be studied either in the observational domain (time and space) or in the corresponding mathematical Fourier domain (frequency and wavenumber).

Continuous signals that arrive at receivers are sampled at discrete intervals of time for further processing on a computer. For a seismic trace, the independent variable is usually time t ; the dependent variable is generally the ground displacement or velocity at that time. If t is the independent variable then you are said to be in the time domain; if frequency ω (radians/sec) or ν (cycles/sec, or Hertz) is the independent variable then you are in the frequency domain.

In the early 60s continuous recording on magnetic tape was replaced by digital recording. For seismic data the continuous function $f(t)$ arriving at a receiver is sampled at times $t = n\Delta t$, $n = 1, 2, 3, \dots$, where Δt is commonly either 1, 2, or 4 milliseconds (ms), or a fraction of a ms for shallow, high-resolution seismic data. See Chapter 4 for important considerations in sampling. If Δt is simply defined to be one time unit, then $t = n$. We can therefore express the independent variable as either t or n as convenient, keeping in mind any appropriate requirement to consider dimensions.

After continuous data are sampled at discrete intervals, the signal is called discrete. A discrete signal is one that is defined only for integer values of the independent variable(s); the dependent variable *could* take on a continuous range of values. For a digital signal, however, the dependent variable must also be discrete. Digital signals are thus considered to be a subset of discrete signals. Not only is the time quantitized, but so also is the amplitude.

There are four types of Fourier transforms in the nomenclature of conventional signal processing. They are

1. Fourier Transform (FT). Both time and frequency are continuous. All of the discrete types of Fourier transforms listed next are derived from the continuous Fourier transform.
2. Discrete-frequency Fourier Transform. Frequency is discrete, time is continuous. More commonly referred to as a Discrete Fourier Series (DFS) or simply a Fourier Series.
3. Discrete-Time Fourier Transform (DTFT). Time is discrete, frequency is continuous.
4. Discrete Fourier Transform (DFT). Both time and frequency are discrete. The introduction of the “fast Fourier transform” (FFT) version of the

Table 3.1: Notation for the Fourier transform (FT)

Analysis	Synthesis
$F(\nu) = \int_{-\infty}^{+\infty} f(t) e^{-i2\pi\nu t} dt$	$f(t) = \int_{-\infty}^{+\infty} F(\nu) e^{+i2\pi\nu t} d\nu$
$f(t)$ is continuous, aperiodic, and can be complex	$F(\nu)$ is complex, continuous, and aperiodic
$\infty \leq t \leq \infty$	$\infty \leq \nu \leq \infty$
$\infty \leq \nu \leq \infty$	$\infty \leq t \leq \infty$

DFT has revolutionized geophysics and other fields that deal with large data sets. It is the DFT that is by far the most widely used Fourier transform in geophysics.

Examples of each are given below. The notation $f(t)$ with curved brackets denotes a continuous function; the notation $f[n]$ with square brackets denotes a discrete signal.

3.2.1 FT—Continuous time, continuous frequency

Until the mid-50s seismic data were photographically recorded continuously with time on paper records, a recording medium that was later replaced by magnetic tape; the data, however, were still recorded continuously. Because the independent variable (time) and the dependent variable (ground velocity or displacement) are both recorded continuously, the signal is called a continuous signal. Continuous-time signals are also called *analog* signals. For example, the continuous function $F(\omega)$ defined by the following continuous integral is called the Fourier transform of the analog trace $f(t)$:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt$$

Commonly we refer to “analysis” and “synthesis” where analysis refers to transformation from the time domain to the frequency domain and synthesis just the opposite. A “Fourier transform” strictly applies to continuous signals in either the time or frequency domains, or in the space and wavenumber domains. Notation for the Fourier transform is shown in Table 3.1. We will commonly switch between angular frequency ω and frequency (cycles per sec), ν (Hertz), where $\omega = 2\pi\nu$.

As an example of a Fourier transform consider the continuous function

$$f(t) = \cos(t) e^{-t} u(t)$$

where $u(t)$ is the continuous step function. The Fourier transform $F(\omega)$ of $f(t)$ is

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

where the function $F(\omega)$ is also continuous.

3.2.2 DFS—Continuous time, discrete frequency

In the following form of a “Fourier series” $f(t)$ is a continuous, periodic signal

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a[n] \cos(2\pi \frac{n}{T} t) - 2 \sum_{n=1}^{\infty} b[n] \sin(2\pi \frac{n}{T} t) \quad (3.1)$$

The independent variable t on the right-hand side of (3.1) can take on any value, positive, negative, or zero, so time is continuous. In order for $f(t)$ to be periodic, however, only certain values of frequency n/T can be used, so $a[n]$ and $b[n]$ are discrete. The general notation for a discrete Fourier series (DFS) is shown in Table 3.2. We will start our discussion of Fourier theory in Chapter 3 with the Fourier series but a simple example will demonstrate the discrete nature of $a[n]$ and $b[n]$ and the continuous nature of t . Consider the continuous sinusoid

$$f(t) = \cos(2\pi\nu t + \theta)$$

that has been sampled at an interval Δt as labeled in Figure 3.2. The sampling interval is clearly too large to reproduce a smooth sinusoid when the sampled points (the dots) are connected. Note that the peaks and troughs themselves are not sampled. Using the analysis equation shown in Table 3.2 on the left we obtain the discrete series of coefficients $a[n]$ and $b[n]$ that define $F[n]$. With these coefficients available it is now possible to return to the time domain using Equation 3.1 or the equivalent synthesis equation shown in Table 3.2 and it is possible to return to the time domain at any time t because t is continuous. It is therefore possible to determine those values of $f(t)$ in between the original sampled intervals and locate the peaks and troughs of the original sampled sinusoid, as shown in Figure 3.2 where the continuous line $f(t)$ has been generated using (3.1).

3.2.3 DTFT—Discrete time, continuous frequency

Discrete-time signals are defined at discrete times; that is, the independent variable (time) has discrete values. Transformation to the frequency domain yields a function that is continuous in frequency. An example of a discrete-time signal is shown in Figure 3.3. An example of a discrete time series $f[n]$ is

$$f[n] = a^n u[n - 4]$$

where n is any integer, $u[n]$ is the “unit step sequence” and $u[n - 4]$ represents a unit step sequence delayed by 4 units.

Table 3.2: Notation for a discrete Fourier series (DFS).

Analysis	Synthesis
$F[n] = \frac{1}{T} \int_0^T f(t) e^{-i2\pi\frac{n}{T}t} dt$	$f(t) = \sum_{n=-\infty}^{+\infty} F[n] e^{+i2\pi\frac{n}{T}t}$
$f(t)$ is continuous, periodic, and can be complex	$F[n]$ is discrete, aperiodic, and can be complex.
$0 \leq t \leq T$	$-\infty \leq n \leq \infty$
$-\infty \leq n \leq \infty$	$0 \leq t \leq T$

From Table 3.3 the discrete-time Fourier transform (DTFT) $F(\omega)$ is

$$F(\omega) = \sum_{n=-\infty}^{+\infty} f[n] e^{-i\omega n} = \sum_{n=-\infty}^{+\infty} a^n u[n-4] e^{-i\omega n} \quad (3.2)$$

$$= \frac{a^4 e^{-3i\omega}}{e^{i\omega} - a} \quad (3.3)$$

Equation (3.2) can be evaluated to get (3.3) using the *Mathematica* `Sum[]` statement

$$\text{Sum}[a^n \text{UnitStep}[n - 4] e^{-i\omega n}, \{n, -\infty, \infty\}] = \frac{a^4 e^{-3i\omega}}{e^{i\omega} - a}$$

Note that (3.3) can be evaluated for any value of ω because $F(\omega)$ is continuous.

The discrete-time Fourier transform (DTFT) is “analogous” to the discrete Fourier series (DFS) in the sense that for the DFS a discrete number of Fourier coefficients is given from which it is possible to determine $f(t)$ at any time t ; $f(t)$ is continuous. For the discrete-time Fourier transform (DTFT) a discrete number of values of a time-domain signal is given from which it is possible to determine $F(\omega)$ at any frequency ω ; $F(\omega)$ is continuous. The DFS is useful for any application where it is desired to search for a peak (or trough) in a time series where the peak happens to fall somewhere within the sampling interval Δt of the signal (Figure 3.2). An example of this is a search for the peak (or trough) of a seismic wavelet that has undergone body wave dispersion (page 518) in a medium (a granite, say) where the absorption of energy is so small (high Q) that the spreading of the wavelet is less than one sample interval and a peak (or trough) lies between $f(t_0)$ and $f(t_0 + \Delta t)$. The DTFT is useful if you know the frequency of interest lies somewhere within the interval $F(\omega_0)$ and $F(\omega_0 + \Delta\omega)$ and you want to search for its maximum value.

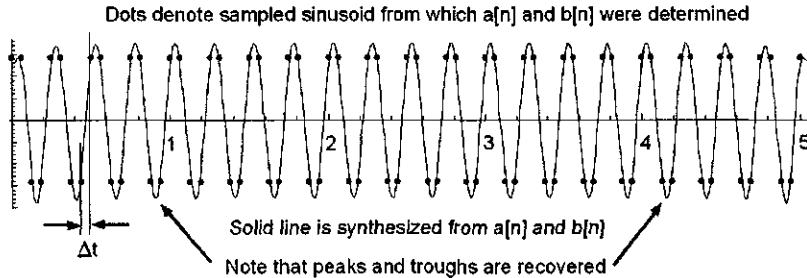


Figure 3.2: (a) The dots represent a sinusoid sampled at an interval too large to plot as a smooth sinusoid if the dots were to be connected. Using the discrete series $a[n]$ and $b[n]$ determined from the analysis equation shown in Table 3.2, the sinusoid has been synthesized at values of t using Equation (3.1) for values of $f(t)$ (the solid line) that now fall in between the original samples (the dots). The computed result made possible by this Fourier interpolation is a smooth sinusoid (the solid line). SampleSinusoid.nb \Rightarrow UnitStep.cdr \Rightarrow SampleSinusoid.wmf

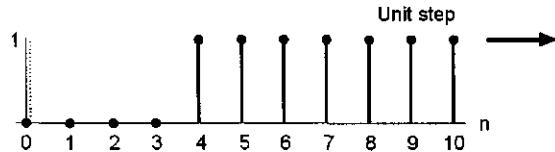


Figure 3.3: The unit step sequence, an example of a discrete-time signal. UnitStep.cdr \Rightarrow UnitStep.wmf

3.2.4 DFT—Discrete time, discrete frequency

For a discrete signal the dependent variable in general can be either continuous or discrete. A discrete signal is thus differentiated from a digital signal in that the dependent variable can take on a continuous range of values, while for a digital signal, the dependent variable must also be discrete. If both the independent and dependent variables are discrete then the signal is called a “digital signal” and the notation is that of Table (3.4). Digital signals are thus a subset of discrete signals. Notation for the “Discrete Fourier Transform” (DFT) is shown in Table 3.4 and a brief review of the “Fast Fourier Transform” is given on page 89.

A summary of the evolution of the recording of seismic data from analog paper records to modern digital methods can be found in Helbig [85]. Standard references for signal processing are Discrete-Time Signal Processing by Oppenheim and Schafer [128] and [129].

In this volume we do not adhere rigidly to the above standard notation for digital signal processing. For example, f_n is commonly substituted for $f[n]$ be-

Table 3.3: Notation for the Discrete Time Fourier Transform (DTFT).

Analysis	Synthesis
$F(\omega) = \sum_{n=-\infty}^{+\infty} f[n] e^{-i\omega n}$ <p>$f[n]$ is discrete, aperiodic, and complex.</p> <p>$-\infty \leq n \leq +\infty$</p>	$f[n] = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{+i\omega n} d\omega$ <p>$F(\omega)$ is continuous, periodic, and complex.</p> <p>$0 \leq \omega \leq 2\pi$</p>

Table 3.4: Notation for the Discrete Fourier Transform (DFT).

Analysis	Synthesis
$F[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] e^{-i2\pi \frac{k}{N} n}$ <p>$f[n]$ is discrete, periodic, and can be complex.</p> <p>$0 \leq n \leq N - 1$</p> <p>$0 \leq k \leq N - 1$</p>	$f[n] = \sum_{k=0}^{N-1} F[k] e^{+i2\pi \frac{k}{N} n}$ <p>$F[k]$ is discrete, periodic, and can be complex.</p> <p>$0 \leq k \leq N - 1$</p> <p>$0 \leq n \leq N - 1$</p>
See the Fast Fourier Transform (FFT) on page 89 for discussion of positive and negative times and frequencies	

cause it is easier to read. In addition, although seismic data are digital data, continuous integrals are often substituted for discrete summations because they are easier to write. Furthermore, digital signals are commonly derived from continuous ones. In addition, a “Fourier transform” really applies only to continuous signals in both the time and frequency domains but we will relax this definition and use it interchangeably with the term “discrete Fourier transform” (DFT), one of the work horses of geophysical data processing.

The signals dealt with in this volume are generally recorded on the surface of the earth or at sea. They are reflected and/or refracted from acoustic discontinuities in the subsurface. The objective is to reveal the subsurface geometry and ultimately relate this to the geology. The received signal is passed through an analog-to-digital converter (A/D) that must be able to convert the continuous ground motion to a numerical value—without clipping or distortion. This will depend upon how well the electronics can assign a value of ground motion to somewhere within a range between some maximum allowed value (a strong signal) and some minimum value (a weak signal) that lies above the electronic noise level of the system. The more “intervals” in the A/D converter then the closer the analog-to-digital conversion will approximate the actual continuous ground motion. If the ground motion is too large then the analog signal will be distorted by the receiver and the sampled value will be incorrect. If the ground motion is too small, then the signal will be lost in the background noise of the recording electronics. So there is always a certain dynamic range within which the sampled continuous signal will be a true, undistorted, representation of the continuous ground motion. The dynamic range is determined by system noise at the lower end and the onset of distortion at the upper end. A large dynamic range will permit the recording of weak reflections in the presence of strong signals (i.e., surface waves) that come in at the same time. The theoretical dynamic range of modern A/D converters for recording seismic data is about 126 dB. Compare this with the resolution of a paper record, about 30 dB. Some electronic noise is always present, so the dynamic range will generally be 10-20 dB less than the theoretical value. If a seismic amplifier measures a signal of 2 volts accurately and *at the same time* measure a signal of 0.000002 volts (2 μ V) accurately then the dynamic range of the system is

$$20 \log_{10} \left(\frac{2}{0.000002} \right) = 120 \text{ dB}$$

so signals with an amplitude difference of 1 million can be measured accurately *at the same time*. Of course the geophones or hydrophones must have a true dynamic range at least equal to this for faithful reproduction.

SUBROUTINE FT

As a nostalgic note, the following Fortran subroutine FT from Enders Robinson’s *Multichannel Time Series Analysis With Digital Computer Programs* [148] properly used (and easily translated for *Mathematica*) can approximate quite

well all of the above Fourier transformations for either complex or real transforms, including other members of the Fourier family such as Hilbert transforms. It is worth the effort to compare numerical results from this program with those from less transparent but faster, higher-level software such as *Mathematica*. Even though many of the Fortran subroutines in [148] have been converted to *Mathematica* modules in Appendix 2 of Robinson and Osman [152], mastering and using a Fortran or *Mathematica* version of FT from scratch is still a pleasant and rewarding experience.

```
SUBROUTINE FT(NX,X,W,C,S)
DIMENSION X(NX)
COSNW=1.0
SINNW=0.0
SINW=SIN(W)
COSW=COS(W)
S=0.0
C=0.0
DO 30 I=1,NX
C=C+COSNW*X(I)
S=S+SINNW*X(I)
T=COSW*COSNW-SINW*SINNW
SINNW=COSW*SINNW+SINW*COSNW
30 COSNW=T
RETURN
END
```

3.3 The Fourier Coefficients

We review first the mathematical notation for the definition of Fourier coefficients that produce functions that repeat themselves in time (*periodic* functions). Later, we examine what is necessary to generate functions that are *aperiodic* (non-periodic) and do not repeat themselves in time.

With reference to Figure 3.4, P moves at uniform speed around a circular path of radius A . Height PQ is given by:

$$PQ = A \sin \alpha = A \sin(\omega t)$$

where ω is radians per time unit and the initial phase angle is zero.

If at time $t = 0$, the initial phase angle is α , and if the period of revolution is T , then the formula for PQ is

$$\begin{aligned} PQ &= A \sin(\omega t + \alpha) \\ &= A \sin(2\pi\nu t + \alpha) \\ &= A \sin(2\pi \frac{1}{T} t + \alpha) \end{aligned}$$

where ν is cycles per time unit (1 Hertz = 1 cycle/sec).

Thus the ratio between the number of radians per time unit and the number of cycles per time unit is 2π . That is, $\omega = 2\pi\nu$.

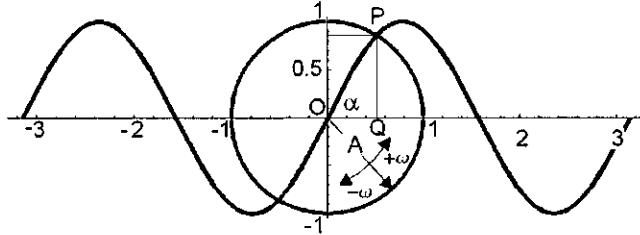


Figure 3.4: Relation between angular frequency ω , and linear frequency ν for a pure sinusoid. As the point P moves in a clockwise or counterclockwise circle until α has changed by an angular amount 2π , the point P traces out one cycle of a sinusoid. The height PQ above the base line through the center of the circle varies smoothly with time and describes a sinusoid of height PQ . The radius of the circle OP is a *phasor*. Counterclockwise rotation of the phasor OP is the convention for positive frequencies; clockwise for negative. (cd → fig1-1)

Consider a general function $f(t)$ of length T . Any such function $f(t)$ can be described by summing together sinusoids whose frequencies are integer multiples of the fundamental frequency $1/T$:

$$\dots - \frac{4}{T} - \frac{3}{T} - \frac{2}{T} - \frac{1}{T} - \frac{0}{T} + \frac{1}{T} + \frac{2}{T} + \frac{3}{T} + \frac{4}{T} + \dots \quad (3.4)$$

where T is the duration of the function. For each frequency n/T there is a sinusoid (a cosine or sine wave):

$$\begin{aligned} f(t) = & A_{-n} \sin \left(2\pi \frac{-n}{T} t + \alpha_{-n} \right) + \dots + \\ & A_{-2} \sin \left(2\pi \frac{-2}{T} t + \alpha_{-2} \right) + A_{-1} \sin \left(2\pi \frac{-1}{T} t + \alpha_{-1} \right) + \\ & A_0 + \\ & A_{+1} \sin \left(2\pi \frac{+1}{T} t + \alpha_{+1} \right) + A_{+2} \sin \left(2\pi \frac{+2}{T} t + \alpha_{+2} \right) + \dots + \\ & A_{+n} \sin \left(2\pi \frac{+n}{T} t + \alpha_{+n} \right) \end{aligned} \quad (3.5)$$

The values of A_n and α_n make up the *spectrum* of $f(t)$. A_n is the *amplitude spectrum* and α_n is the *phase spectrum*. The spectrum describes the amplitudes and phase angles of sinusoids that, when added together, give the function $f(t)$ over the time T .

For a function to repeat itself with period T the only frequencies ν that can be used in the summations are integer multiples of the fundamental frequency $\frac{1}{T}$. Such a Fourier summation using frequencies $1/T, 2/T, 3/T, \dots$ will converge to $f(t)$ as desired. That this must be so can be seen by examination of Figure 3.5.

At $t = 0$ and $t = T$ each of the sinusoids starts and ends at the same place so that for times $t < 0$ and $t > T$ the summation will repeat itself with period T .

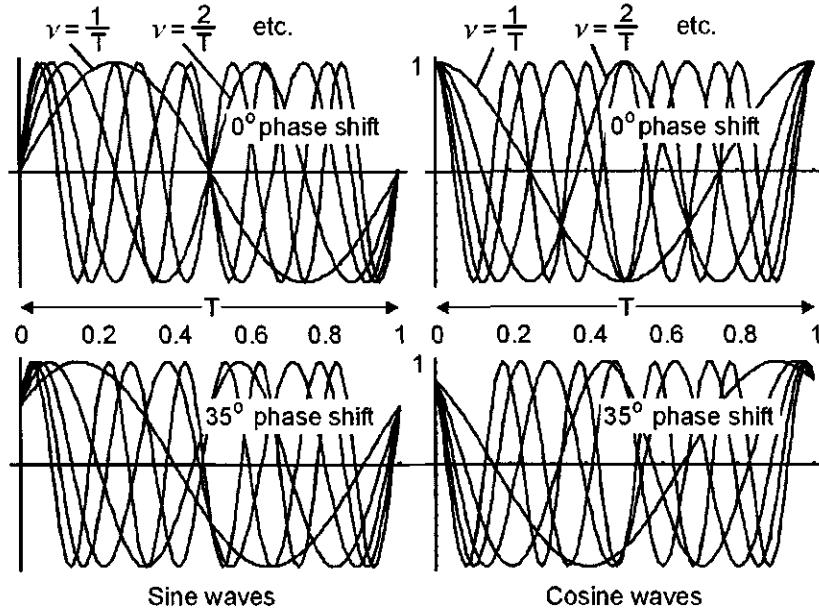


Figure 3.5: Sine waves left column, cosine waves right column. Top row: no phase shift. Bottom row: 35° phase shift. The time series $f(t)$ are T sec long. As long as the frequencies used in the Fourier series summation of Equation (3.33) are integer multiples of $1/T$ then the summations will start at the same place for $t = 0$ and $t = T$. At $t = 0$ and $t = T$ each of the sinusoids starts and ends at the same place for their respective phase shifts so that for times $t < 0$ and $t > T$ the summations will repeat themselves with period T . For all times $t < 0$ and $t > T$ $f(t)$ will therefore be periodic with period T .

Expanding each term on the right-hand side of Equation 3.5 by using the identity $\sin(\gamma + \beta) = \sin \gamma \cos \beta + \cos \gamma \sin \beta$, we get

$$\begin{aligned}
 A_n \sin\left(2\pi \frac{n}{T} t + \alpha_n\right) &= A_n \left[\sin\left(2\pi \frac{n}{T} t\right) \cos \alpha_n + \cos\left(2\pi \frac{n}{T} t\right) \sin \alpha_n \right] \\
 &= A_n \cos \alpha_n \sin\left(2\pi \frac{n}{T} t\right) + A_n \sin \alpha_n \cos\left(2\pi \frac{n}{T} t\right) \\
 &= b_n \sin\left[2\pi \frac{n}{T} t\right] + a_n \cos\left[2\pi \frac{n}{T} t\right]
 \end{aligned} \tag{3.6}$$

where, because $A_n \cos \alpha_n$ and $A_n \sin \alpha_n$ are constants, we can define $b_n = A_n \cos \alpha_n$ and $a_n = A_n \sin \alpha_n$.

This alternate formulation eliminates the need to consider phase angles. The unknowns are now a_n and b_n instead of A_n and α_n . Thus $f(t)$ can be synthesized

using the a_n and b_n terms:

$$\begin{aligned}
 f(t) = & a_{-n} \cos\left(2\pi \frac{-n}{T} t\right) + \dots + \\
 & a_{-3} \cos\left(2\pi \frac{-3}{T} t\right) + a_{-2} \cos\left(2\pi \frac{-2}{T} t\right) + a_{-1} \cos\left(2\pi \frac{-1}{T} t\right) + \\
 & a_0 + \\
 & a_{+1} \cos\left(2\pi \frac{+1}{T} t\right) + a_{+2} \cos\left(2\pi \frac{+2}{T} t\right) + a_{+3} \cos\left(2\pi \frac{+3}{T} t\right) + \dots + \\
 & a_{+n} \cos\left(2\pi \frac{+n}{T} t\right) + b_{-n} \sin\left(2\pi \frac{-n}{T} t\right) + \dots + \quad (3.7) \\
 & b_{-3} \sin\left(2\pi \frac{-3}{T} t\right) + b_{-2} \sin\left(2\pi \frac{-2}{T} t\right) + b_{-1} \sin\left(2\pi \frac{-1}{T} t\right) + \\
 & b_{+1} \sin\left(2\pi \frac{+1}{T} t\right) + b_{+2} \sin\left(2\pi \frac{+2}{T} t\right) + b_{+3} \sin\left(2\pi \frac{+3}{T} t\right) + \dots + \\
 & b_{+n} \sin\left(2\pi \frac{+n}{T} t\right)
 \end{aligned}$$

where we use the notation a_{-n} to denote the amplitude of the coefficient a_{-n} at the frequency $-n/T$, and the notation a_n to denote the amplitude of the coefficient a_n at the frequency $+n/T$.

In general a_{+n} is not equal to a_{-n} because $f(t)$ can be complex; however, if $f(t)$ is pure real, there is an important symmetry in the frequency domain between amplitudes of sinusoids at positive frequencies and amplitudes of sinusoids at negative frequencies:

$$\begin{aligned}
 a_{+n} &= a_{-n} \\
 b_{-n} &= -b_{+n}
 \end{aligned}$$

That is, a_n is an even function of frequency and b_n is an odd function of frequency.

It should be noted that Equation (3.7) holds for any value of time t regardless of whether t is positive, negative, or zero. Now, because cosines are even functions and sines are odd functions, we also have

$$\cos\left(2\pi \frac{-n}{T} t\right) = \cos\left(2\pi \frac{+n}{T} t\right)$$

and

$$\sin\left(2\pi \frac{-n}{T} t\right) = -\sin\left(2\pi \frac{+n}{T} t\right)$$

We also note that an even function times an even function is an even function, and an odd function times an odd function is an even function. That is, for $f(t)$

pure real, both

$$a_{+n} \cos\left(2\pi \frac{+n}{T} t\right) = a_{-n} \cos\left(2\pi \frac{-n}{T} t\right) \quad (3.8)$$

and

$$b_{+n} \sin\left(2\pi \frac{+n}{T} t\right) = b_{-n} \sin\left(2\pi \frac{-n}{T} t\right) \quad (3.9)$$

are even functions of frequency. Therefore, from Equations (3.7), (3.8), and (3.9), we can write

$$\begin{aligned} f(t) &= a_0 \cos\left(2\pi \frac{n}{T} t\right) + \dots \\ &+ a_3 \cos\left(2\pi \frac{3}{T} t\right) + a_2 \cos\left(2\pi \frac{2}{T} t\right) + a_1 \cos\left(2\pi \frac{1}{T} t\right) \\ &+ a_0 \dots \\ &+ a_1 \cos\left(2\pi \frac{1}{T} t\right) + a_2 \cos\left(2\pi \frac{2}{T} t\right) + a_3 \cos\left(2\pi \frac{3}{T} t\right) + \dots \\ &- a_n \cos\left(2\pi \frac{n}{T} t\right) \\ &- b_n \sin\left(2\pi \frac{n}{T} t\right) - \dots \\ &- b_3 \sin\left(2\pi \frac{3}{T} t\right) - b_2 \sin\left(2\pi \frac{2}{T} t\right) - b_1 \sin\left(2\pi \frac{1}{T} t\right) \\ &- b_1 \sin\left(2\pi \frac{1}{T} t\right) - b_2 \sin\left(2\pi \frac{2}{T} t\right) - b_n \sin\left(2\pi \frac{3}{T} t\right) - \dots \\ &- b_n \sin\left(2\pi \frac{n}{T} t\right) \end{aligned}$$

or simply

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) - 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (3.10)$$

This important summation is called a discrete Fourier series (DFS) as defined in Table 3.2 on page 44. It appears in different forms in various texts but (3.10) is as good as any.

3.3.1 Sign Convention

The choice of a negative sign before the b_n terms in the above equations is quite arbitrary, and we could just as well have written

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) + 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (3.11)$$

Equation (3.11) is, in fact, the notation universally found in mathematics textbooks for Fourier *series*, but not always for Fourier *integrals*. The reason for this is that mathematics textbooks do not switch back and forth between Fourier series (discrete summation) and Fourier integrals (continuous summation); the discussions of each subject are self-contained, and there may be few applications. For example, see the excellent text by Lee [108, p.5 and p.33]. The positive sign before the b_n terms appears to be almost always used in engineering textbooks. We will see later that the negative sign before the b_n terms is the preferred notation for an intuitive understanding of the Fourier integral. The sign convention is discussed further in Section 3.3.9. Throughout this volume we use a negative sign before the b_n terms and retain this convention when Fourier integrals are introduced, even though there we deal with continuous instead of discrete summations.

The z -plane is a complex plane with an imaginary and real axis that refer to the complex-valued variable z . The sign convention used here for z in a z -transform is

$$z = e^{-i\omega t} = e^{-i2\pi\nu t} \quad (3.12)$$

instead of

$$z = e^{+i\omega t} \quad (3.13)$$

as used by engineers. The distinction can be an important one in practical applications. *Numerical Recipes in C*[140] uses the convention of Equations (3.13) and (3.11). The definition given by Equation (3.12) is used by Robinson and Treitel in their many publications on geophysical applications of digital filtering. See for example Silvia and Robinson [164, p. 49]. Where it is important to recognize the sign convention for an application in this volume, the reader will be alerted. *Mathematica* conveniently allows the user to define the Fourier sign convention by defining a parameter called `FourierParameters`. In this volume we use the following *Mathematica* option:

```
FourierParameters->\{1,-1\}
```

The following *Mathematica* program called `FourierSignConvention.nb` can be used to determine which sign convention is used for the imaginary part of a Fourier transform in any other program (such as *Mathematica*'s `Fourier`). Execution of this program defines the sign convention used throughout this volume for the imaginary part of a Fourier transform and compares it with the one used by `Fourier` without the optional `FourierParameters` parameter, which is the opposite.

```
(* FourierSignConvention.nb *)
TestInput={-1,-1,-1,-1,1,1,1,1};
sign=-1;
(* The sign convention used in this volume is sign=-1 *)
dt=0.002;lti=Length[TestInput];period=lti*dt;
```

```

dfreq=1/period; Nyquist=1/(2*dt);
If[OddQ[ltil],L=Ceiling[ltil/2],L=lti/2];L;
w=2*Pi*dt;s=Table[0,{L}];
Do[sum=0;freq=(i-1)*dfreq;
Do[sum=sum+TestInput[[j]]*Sin[w*freq*(j-1)],
{j,1,lti}];s[[i]]=sign*sum,{i,1,L}];
Print["Sin transform for this volume:",Chop[s]];
Print["Sin transform from Mathematica:",
Take[Im[Fourier[TestInput]],L]*Sqrt[lti]];
Print["Change sign to +1 to see the difference"];

```

In order to use our sign convention with *Mathematica* change the above Take statement to

```

Take[Im[Fourier[TestInput,FourierParameters->\{1,-1\}]],
L]*Sqrt[lti];

```

The unknowns in Equation (3.10) are now a_n and b_n instead of A_n and α_n . In (3.10), because of the symmetry in the frequency domain, we need to deal only with positive frequencies. By definition, $\omega = d\theta/dt$. The angle θ is measured from the x -axis. If θ increases with time away from the x -axis toward the positive imaginary i -axis, then $d\theta > 0$. Positive values of frequency ω simply mean that the phasor rotates counterclockwise. Negative values of frequency mean that the phasor rotates counterclockwise.

Both positive and negative frequencies *must* always be accounted for in any Fourier synthesis; they are implicitly included in Equation (3.10) by the factor of “2” that multiplies each summation. As noted above, if the time-domain signal $f(t)$ is pure real, then the amplitudes of sine waves of negative frequency $-n/T$ are of opposite sign to sine waves of positive frequency $+n/T$.

The important point is to recognize that the sinusoids of negative frequency must always be included, one way or another, in the summation, or you must seek the justification for not using them. If, because of symmetry, only the positive frequencies need to be explicitly considered in computations, then any function $f(t)$ can be simply represented as

$$\begin{aligned}
f(t) = & \\
& A_0 + 2A_1 \sin\left(2\pi \frac{1}{T}t + \alpha_1\right) + 2A_2 \sin\left(2\pi \frac{2}{T}t + \alpha_2\right) + \dots \\
& + 2A_n \sin\left(2\pi \frac{n}{T}t + \alpha_n\right)
\end{aligned}$$

or by our preferred notation of Equation (3.10) where because of symmetry the negative frequencies *are* included by virtue of multiplication by “2”. Note that t can take on any value and for this reason the notation $f(t)$ is used in (3.10) instead of $f[t]$. See the example of Figure 3.2 on page 45.

In Equation(3.10), the units of a_n or b_n are the same as the units of $f(t)$ because trigonometric functions are dimensionless. If $f(t)$ is in microns, then a_n and b_n must be in microns.

Equation (3.10) is called an “inverse Fourier transform”. If the coefficients a_n and b_n are known, then the function $f(t)$ of length T can be obtained by adding together cosine and sine waves whose amplitudes are a_n and b_n , respectively, and whose frequencies n/T are integer multiples of the fundamental frequency $1/T$. The determination of the Fourier coefficients a_n and b_n is discussed in the following sections.

3.3.2 Determination of the Fourier coefficients

The coefficients a_n and b_n are called *Fourier*, or *Euler* coefficients. In this volume we determine the coefficients by using two general methods:

1. Determine the coefficients by solving a set of simultaneous linear equations, and
2. The conventional approach of introducing orthogonality, which results in integral expressions for the coefficients.

Examples are provided to show that either approach gives the same result.

3.3.3 Fourier Coefficients from Linear Equations

In discrete form, with a function of m points uniformly sampled over the period $T = m \Delta t$ and letting $\Delta t = 1$ time unit, then Equation (3.10) becomes

$$f[k] = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(\frac{n}{m} k\right) - 2 \sum_{n=1}^{\infty} b_n \sin\left(\frac{n}{m} k\right)$$

and we can write for $k = 0, 1, 2, 3, 4$ and $n = 1, 2$,

$$\begin{aligned} f[0] &= a_0 + 2a_1 \cos\left(2\pi \frac{1}{m} 0\right) + 2a_2 \cos\left(2\pi \frac{2}{m} 0\right) + 2b_1 \sin\left(2\pi \frac{1}{m} 0\right) + 2b_2 \sin\left(2\pi \frac{2}{m} 0\right) \\ f[1] &= a_0 + 2a_1 \cos\left(2\pi \frac{1}{m} 1\right) + 2a_2 \cos\left(2\pi \frac{2}{m} 1\right) + 2b_1 \sin\left(2\pi \frac{1}{m} 1\right) + 2b_2 \sin\left(2\pi \frac{2}{m} 1\right) \\ f[2] &= a_0 + 2a_1 \cos\left(2\pi \frac{1}{m} 2\right) + 2a_2 \cos\left(2\pi \frac{2}{m} 2\right) + 2b_1 \sin\left(2\pi \frac{1}{m} 2\right) + 2b_2 \sin\left(2\pi \frac{2}{m} 2\right) \\ f[3] &= a_0 + 2a_1 \cos\left(2\pi \frac{1}{m} 3\right) + 2a_2 \cos\left(2\pi \frac{2}{m} 3\right) + 2b_1 \sin\left(2\pi \frac{1}{m} 3\right) + 2b_2 \sin\left(2\pi \frac{2}{m} 3\right) \\ f[4] &= a_0 + 2a_1 \cos\left(2\pi \frac{1}{m} 4\right) + 2a_2 \cos\left(2\pi \frac{2}{m} 4\right) + 2b_1 \sin\left(2\pi \frac{1}{m} 4\right) + 2b_2 \sin\left(2\pi \frac{2}{m} 4\right) \end{aligned}$$

Simplifying,

$$\begin{aligned} f[0] &= a_0 + 2a_1 \cos\left(\frac{0\pi}{m}\right) + 2a_2 \cos\left(\frac{0\pi}{m}\right) + 2b_1 \sin\left(\frac{0\pi}{m}\right) + 2b_2 \sin\left(\frac{0\pi}{m}\right) \\ f[1] &= a_0 + 2a_1 \cos\left(\frac{2\pi}{m}\right) + 2a_2 \cos\left(\frac{4\pi}{m}\right) + 2b_1 \sin\left(\frac{2\pi}{m}\right) + 2b_2 \sin\left(\frac{4\pi}{m}\right) \\ f[2] &= a_0 + 2a_1 \cos\left(\frac{4\pi}{m}\right) + 2a_2 \cos\left(\frac{8\pi}{m}\right) + 2b_1 \sin\left(\frac{4\pi}{m}\right) + 2b_2 \sin\left(\frac{8\pi}{m}\right) \\ f[3] &= a_0 + 2a_1 \cos\left(\frac{6\pi}{m}\right) + 2a_2 \cos\left(\frac{12\pi}{m}\right) + 2b_1 \sin\left(\frac{6\pi}{m}\right) + 2b_2 \sin\left(\frac{12\pi}{m}\right) \\ f[4] &= a_0 + 2a_1 \cos\left(\frac{8\pi}{m}\right) + 2a_2 \cos\left(\frac{16\pi}{m}\right) + 2b_1 \sin\left(\frac{8\pi}{m}\right) + 2b_2 \sin\left(\frac{16\pi}{m}\right) \quad (3.14) \end{aligned}$$

This is a system of five equations in the five unknowns a_0 , a_1 , a_2 , b_1 , and b_2 . Thus, the determination of the unknown Fourier coefficients a_n and b_n , is nothing more than the solution of m simultaneous linear equations in m unknowns.

3.3.4 Numerical Example

Using *Mathematica* to solve this system of linear equations, and assuming an input function of $f[t] = 1, 2, 3, 4, 5$, we have

```
m=5;
Array[r,{m,m}]; MatrixForm[r]
v={1,2,3,4,5};
a0=Table[1,{i,1,m}];
a1=Table[2*Cos[i*Pi/m],{i,0,8,2}]//N;
a2=Table[2*Cos[i*Pi/m],{i,0,16,4}]//N;
b1=-Table[2*Sin[i*Pi/m],{i,0,8,2}]//N;
b2=-Table[2*Sin[i*Pi/m],{i,0,16,4}]//N;
r={a0,a1,a2,b1,b2};r=Transpose[r];
MatrixForm[r]
s=LinearSolve[r,v]
Clear[m,v,s,r,f]
```

The statement `MatrixForm[r]` (not followed by a semicolon) produces the output

$$\begin{array}{ccccc} 1 & 2. & 2. & 0. & 0. \\ 1 & 0.618034 & -1.61803 & -1.90211 & -1.17557 \\ 1 & -1.61803 & 0.618034 & -1.17557 & 1.90211 \\ 1 & -1.61803 & 0.618034 & 1.17557 & -1.90211 \\ 1 & 0.618034 & -1.61803 & 1.90211 & 1.17557 \end{array}$$

The output s from the *Mathematica* statement `s=LinearSolve[r,v]` is

$$s = \overbrace{3., -0.5, -0.5, 0.688191, 0.16246}^{a_0 \quad a_1 \quad a_2 \quad b_1 \quad b_2} \quad (3.15)$$

which corresponds as shown in (3.15) to the Fourier coefficients

$$a_0, a_1, a_2, b_1, b_2, \text{ respectively.}$$

Compare these numerical values to those obtained by the method described below and shown in Section 3.3.8.

In the above *Mathematica* example, we took advantage of the symmetry in the frequency domain that occurs if the function in the time domain is real; that is, all coefficients except a_0 were multiplied by 2. Taking advantage of symmetry, however, is not a requirement, and just results in a program that runs faster.

At this point we could end the discussion of how to determine the Fourier coefficients because we have shown that all you need is a good simultaneous linear

equation solver. Given a seismic trace consisting of, for example, 5,000 points, there will be 2,501 a_n and b_n coefficients for zero and positive frequencies, and 2,499 a_n and b_n coefficients for negative frequencies. See page 92 for a discussion of odd versus even number of points. Conceptually, the use of simultaneous linear equations is probably the easiest explanation of how to obtain the Fourier coefficients. It is not, however, the approach generally taken to determine the values of a_n and b_n even though the numerical values obtained by any other method are identical to those obtained by solving the linear equations. So we review the more conventional, but rather tedious, approach to determining the Fourier coefficients.

3.3.5 Fourier Coefficients from Orthogonality

To determine the coefficients a_n and b_n we will

- use the concept of the average value of a function $f(t)$, and
- examine the average values of sine and cosine functions, and products of such functions over one (or more) periods 2π .

The average value of a function

The average of k values of $f(t)$ is, by definition,

$$\frac{f(t_1) + f(t_2) + f(t_3) + \cdots + f(t_k)}{k} = \frac{1}{k} \sum_{i=1}^k f(t_i)$$

If these values are separated in time by an equal amount Δt then we change nothing by multiplying the numerator and denominator of the above by Δt to obtain

$$\frac{[f(t_1) + f(t_2) + f(t_3) + \cdots + f(t_k)] \Delta t}{k \Delta t} = \frac{1}{k \Delta t} \sum_{i=1}^k f(t_i) \Delta t$$

Now $k \Delta t = T$ is the length of time over which we average. For any finite value of T , if we allow $k \rightarrow \infty$ as $\Delta t \rightarrow 0$, the numerator becomes

$$\int_0^T f(t) dt$$

and the average value of $f(t)$ becomes

$$\frac{\int_0^T f(t) dt}{T} = \frac{1}{T} \int_0^T f(t) dt$$

Useful integrals

We will also need the following integrals:

$$\frac{1}{T} \int_0^T \cos\left(2\pi \frac{m}{T}t\right) \cos\left(2\pi \frac{n}{T}t\right) dt = \frac{1}{2} \left[\frac{\sin[2\pi(m-n)/T]t}{2\pi(m-n)/T} + \frac{\sin[2\pi(m+n)/T]t}{2\pi(m+n)/T} \right]_0^T n, m \neq 0 \quad (3.16)$$

$$\begin{aligned} \frac{1}{T} \int_0^T \sin\left(2\pi \frac{m}{T}t\right) \cos\left(2\pi \frac{n}{T}t\right) dt &= \\ -\frac{1}{2} \left[\frac{\cos[2\pi(m-n)/T]t}{2\pi(m-n)/T} + \frac{\cos[2\pi(m+n)/T]t}{2\pi(m+n)/T} \right]_0^T n \neq 0 \end{aligned} \quad (3.17)$$

$$\frac{1}{T} \int_0^T \sin\left(2\pi \frac{n}{T}t\right) \cos\left(2\pi \frac{n}{T}t\right) dt = \left[\frac{\sin^2(2\pi \frac{n}{T})}{2\pi \frac{n}{T}} \right]_0^T n \neq 0 \quad (3.18)$$

$$\begin{aligned} \frac{1}{T} \int_0^T \sin\left(2\pi \frac{m}{T}t\right) \sin\left(2\pi \frac{n}{T}t\right) dt &= \\ \frac{1}{2} \left[\frac{\sin[2\pi(m-n)/T]t}{2\pi(m-n)/T} - \frac{\sin[2\pi(m+n)/T]t}{2\pi(m+n)/T} \right]_0^T &= 0 \end{aligned} \quad (3.19)$$

We are now ready to proceed. To find a_0 , find the average value over the period T of each term in Equation 3.10.

$$\begin{aligned} \frac{1}{T} \int_0^T f(t) dt &= \frac{1}{T} \int_0^T a_0 dt + \frac{2}{T} \int_0^T a_1 \cos\left(2\pi \frac{1}{T}t\right) dt + \\ &\quad \frac{2}{T} \int_0^T a_2 \cos\left(2\pi \frac{2}{T}t\right) dt + \frac{2}{T} \int_0^T a_3 \cos\left(2\pi \frac{3}{T}t\right) dt + \dots - \\ &\quad \frac{2}{T} \int_0^T b_1 \sin\left(2\pi \frac{1}{T}t\right) dt - \frac{2}{T} \int_0^T b_2 \sin\left(2\pi \frac{2}{T}t\right) dt - \\ &\quad \frac{2}{T} \int_0^T b_3 \sin\left(2\pi \frac{3}{T}t\right) dt - \dots - \end{aligned} \quad (3.20)$$

All of the integrals on the right-hand side of Equation (3.20) are zero except the first, because the integrals are all of the form

$$\int_0^T \cos\left(2\pi \frac{n}{T}t\right) dt$$

or

$$\int_0^T \sin\left(2\pi \frac{n}{T}t\right) dt$$

We therefore have

$$\frac{1}{T} \int_0^T f(t) dt = \frac{1}{T} \int_0^T a_0 dt = \frac{1}{T} T a_0 = a_0$$

Therefore

$$a_0 = \frac{1}{T} \int_0^T f(t) dt \quad (3.21)$$

and a_0 represents the average value of $f(t)$ over T . The function $f(t)$ is completely known; it is, in fact, commonly the seismic trace. It is therefore possible to determine the coefficient a_0 even if we must resort to numerical integration.

To find the coefficient a_1 multiply both sides of Equation (3.10). by

$$\cos\left(2\pi\frac{1}{T}t\right)$$

and again integrate term by term to find the average value of each term. Thus,

$$\begin{aligned} \frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{1}{T}t\right) dt &= \frac{1}{T} \int_0^T a_0 \cos\left(2\pi\frac{1}{T}t\right) dt + \\ &\quad \frac{2}{T} \int_0^T 2a_1 \cos\left(2\pi\frac{1}{T}t\right) \cos\left(2\pi\frac{1}{T}t\right) dt + \\ &\quad \frac{2}{T} \int_0^T 2a_2 \cos\left(2\pi\frac{2}{T}t\right) \cos\left(2\pi\frac{1}{T}t\right) dt + \dots - \\ &\quad \frac{2}{T} \int_0^T 2b_1 \sin\left(2\pi\frac{1}{T}t\right) \cos\left(2\pi\frac{1}{T}t\right) dt - \\ &\quad \frac{2}{T} \int_0^T 2b_2 \sin\left(2\pi\frac{2}{T}t\right) \cos\left(2\pi\frac{1}{T}t\right) dt - \dots - \end{aligned}$$

This time, using Equation (3.16), all of the integrals on the right-hand side are zero except for the integral

$$\int_0^T \cos\left(2\pi\frac{1}{T}t\right) \cos\left(2\pi\frac{1}{T}t\right) dt = \int_0^T \cos^2\left(2\pi\frac{1}{T}t\right) dt = \frac{T}{2}$$

Solving for a_1 we get

$$\frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{1}{T}t\right) dt = \frac{1}{T} 2a_1 \frac{T}{2}$$

or

$$a_1 = \frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{1}{T}t\right) dt$$

To find the coefficient a_2 multiply both sides of Equation (3.10) by

$$\cos\left(2\pi\frac{2}{T}t\right)$$

and integrate term by term to find the average value of each term. Thus,

$$\begin{aligned}\frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{2}{T}t\right) dt &= \frac{1}{T} \int_0^T a_0 \cos\left(2\pi\frac{2}{T}t\right) dt + \\ \frac{2}{T} \int_0^T 2a_1 \cos\left(2\pi\frac{1}{T}t\right) \cos\left(2\pi\frac{2}{T}t\right) dt &+ \\ \frac{2}{T} \int_0^T 2a_2 \cos\left(2\pi\frac{2}{T}t\right) \cos\left(2\pi\frac{2}{T}t\right) dt &+ \dots + \\ \frac{2}{T} \int_0^T 2b_1 \sin\left(2\pi\frac{1}{T}t\right) \cos\left(2\pi\frac{2}{T}t\right) dt &+ \\ \frac{2}{T} \int_0^T 2b_2 \sin\left(2\pi\frac{2}{T}t\right) \cos\left(2\pi\frac{2}{T}t\right) dt &+ \dots +\end{aligned}$$

Again, using Equation (3.16), all of the terms on the right-hand side are zero except for the term

$$\frac{1}{T} \int_0^T \cos\left(2\pi\frac{2}{T}t\right) \cos\left(2\pi\frac{2}{T}t\right) dt = \frac{1}{T} \int_0^T \cos^2\left(2\pi\frac{2}{T}t\right) dt = \frac{T}{2}$$

Solving for a_2 we get

$$\frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{2}{T}t\right) dt = \frac{1}{T} 2a_2 \frac{T}{2}$$

or

$$a_2 = \frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{2}{T}t\right) dt$$

A general formulation for the a_n terms can be obtained by multiplying both sides of Equation (3.10) by

$$\cos\left(2\pi\frac{n}{T}t\right)$$

and then finding the average value of each term:

$$\begin{aligned}\frac{1}{T} \int_0^T f(t) \cos\left(2\pi\frac{n}{T}t\right) dt &= \\ \frac{1}{T} \int_0^T a_0 \cos\left(2\pi\frac{n}{T}t\right) dt + \frac{2}{T} \int_0^T 2a_1 \cos\left(2\pi\frac{1}{T}t\right) \cos\left(2\pi\frac{n}{T}t\right) dt &+ \\ + \dots + \frac{2}{T} \int_0^T 2a_n \cos\left(2\pi\frac{n}{T}t\right) \cos\left(2\pi\frac{n}{T}t\right) dt &+\end{aligned}$$

$$\begin{aligned} & \frac{1}{T} \int_0^T 2b_1 \sin\left(2\pi \frac{1}{T}t\right) \cos\left(2\pi \frac{n}{T}t\right) dt + \frac{2}{T} \int_0^T 2b_2 \sin\left(2\pi \frac{2}{T}t\right) \cos\left(2\pi \frac{n}{T}t\right) dt \\ & + \cdots + \frac{2}{T} \int_0^T 2b_n \sin\left(2\pi \frac{n}{T}t\right) \cos\left(2\pi \frac{n}{T}t\right) dt \end{aligned}$$

The result is

$$a_n = -\frac{1}{T} \int_0^T f(t) \cos\left(2\pi \frac{n}{T}t\right) dt \quad \text{for } n = 0, 1, 2, 3, \dots \quad (3.22)$$

This is an important equation and says that the discrete Fourier coefficient a_n can be found by integrating $f(t) \cos\left(2\pi \frac{n}{T}t\right)$ over the time interval from $t = 0$ to T . We also need an expression for the b_n terms.

The average values of sine and cosine functions, and products of sine and cosine functions over a period

Average values of some functions are zero. For example, the average value of $\sin(\theta)$ or $\cos(\theta)$ over a period is zero:

$$\frac{1}{2\pi} \int_0^{2\pi} \sin(\theta) d\theta = \frac{1}{2\pi} \int_0^{2\pi} \cos(\theta) d\theta = 0$$

or, in terms of cycles per time unit n/T ,

$$\frac{1}{T} \int_0^T \cos\left(2\pi \frac{n}{T}t\right) dt = \frac{1}{T} \left[\frac{T}{2\pi n} \sin\left(2\pi \frac{n}{T}t\right) \right]_0^T = 0 \quad (3.23)$$

and

$$\frac{1}{T} \int_0^T \sin\left(2\pi \frac{n}{T}t\right) dt = -\frac{1}{T} \left[\frac{T}{2\pi n} \cos\left(2\pi \frac{n}{T}t\right) \right]_0^T = 0 \quad (3.24)$$

where n is a positive or negative integer. That is, one cycle of a sinusoid integrated over one period is zero. This is true no matter what the initial phase angle. We note also that the average value (without at this point knowing just what the value is) of $\sin^2(\theta)$ over one period is the same as the average of $\cos^2(\theta)$ over one period (because the area under the curves is the same):

$$\frac{\int_0^{2\pi} \sin^2(\theta) d\theta}{2\pi} = \frac{\int_0^{2\pi} \cos^2(\theta) d\theta}{2\pi} \quad (3.25)$$

or simply

$$\int_0^{2\pi} \sin^2(\theta) d\theta = \int_0^{2\pi} \cos^2(\theta) d\theta$$

or, in terms of cycles per time unit

$$\int_0^T \sin^2\left(2\pi \frac{n}{T}t\right) dt = \int_0^T \cos^2\left(2\pi \frac{n}{T}t\right) dt \quad \text{for } n \neq 0$$

We can deduce what the value of either of these integrals is by observing that

$$\int_0^{2\pi} [\sin^2 \theta + \cos^2 \theta] d\theta = \int_0^{2\pi} [1] d\theta = 2\pi$$

or, in terms of cycles per time unit,

$$\int_0^T [\sin^2(2\pi \frac{n}{T}t) + \cos^2(2\pi \frac{n}{T}t)] dt = \int_0^T [1] dt = T \quad (3.26)$$

because $\sin^2 \theta + \cos^2 \theta = 1$.

From Equation (3.25a) and Equation (3.26) we deduce

$$\int_0^T \sin^2(2\pi \frac{n}{T}t) dt = \int_0^T \cos^2(2\pi \frac{n}{T}t) dt = \frac{1}{2}T = \frac{T}{2} \quad \text{for } n \neq 0 \quad (3.27)$$

To summarize, the average value over one or more periods T of

$$\sin^2\left(2\pi \frac{n}{T}t\right)$$

is equal to the average value over one or more periods T of

$$\cos^2\left(2\pi \frac{n}{T}t\right)$$

That is,

$$\begin{aligned} \int_0^T \sin^2\left(2\pi \frac{n}{T}t\right) \frac{dt}{T} &= \int_0^T \cos^2\left(2\pi \frac{n}{T}t\right) \frac{dt}{T} \\ &= \frac{1}{T} \int_0^T \sin^2\left(2\pi \frac{n}{T}t\right) dt = \frac{1}{T} \int_0^T \cos^2\left(2\pi \frac{n}{T}t\right) dt = \frac{T}{2} \quad \text{for } n \neq 0 \end{aligned}$$

A more direct approach leads to the same results:

$$\begin{aligned} \frac{1}{T} \int_0^T \cos^2\left(2\pi \frac{n}{T}t\right) dt &= \\ \left[\frac{t}{2} + \frac{\sin(4\pi \frac{n}{T}t)}{8\pi \frac{n}{T}} \right]_0^T &= \frac{T}{2} \quad \text{for } n \neq 0 \end{aligned}$$

and

$$\frac{1}{T} \int_0^T \sin^2\left(2\pi \frac{n}{T}t\right) dt = \left[\frac{t}{2} - \frac{\sin(4\pi \frac{n}{T}t)}{8\pi \frac{n}{T}} \right]_0^T = \frac{T}{2} \quad \text{for } n \neq 0 \quad (3.28)$$

The b_n coefficients are found in a similar manner. The coefficient b_0 is always zero. To find b_1 multiply both sides of Equation (3.10) by

$$\sin\left(2\pi \frac{1}{T}t\right)$$

and integrate term by term to find the average value of each term. Thus,

$$\begin{aligned} \frac{1}{T} \int_0^T f(t) \sin \left(2\pi \frac{1}{T} t \right) dt &= \\ \frac{1}{T} \int_0^T a_0 \sin \left(2\pi \frac{1}{T} t \right) dt + \frac{2}{T} \int_0^T 2a_1 \cos \left(2\pi \frac{1}{T} t \right) \sin \left(2\pi \frac{1}{T} t \right) dt + \\ \frac{2}{T} \int_0^T 2a_2 \cos \left(2\pi \frac{2}{T} t \right) \sin \left(2\pi \frac{1}{T} t \right) dt + \cdots - \\ \frac{2}{T} \int_0^T 2b_1 \sin \left(2\pi \frac{1}{T} t \right) \sin \left(2\pi \frac{1}{T} t \right) dt - \\ \frac{2}{T} \int_0^T 2b_2 \sin \left(2\pi \frac{2}{T} t \right) \sin \left(2\pi \frac{1}{T} t \right) dt - \cdots - \end{aligned}$$

Using Equation (3.27) all of the integrals on the right-hand side are zero except for the integral

$$\int_0^T \sin \left(2\pi \frac{1}{T} t \right) \sin \left(2\pi \frac{1}{T} t \right) dt = \int_0^T \sin^2 \left(2\pi \frac{1}{T} t \right) dt = \frac{T}{2}$$

Solving for b_1 we get

$$\frac{1}{T} \int_0^T f(t) \sin \left(2\pi \frac{1}{T} t \right) dt = -\frac{1}{T} 2b_1 \frac{T}{2}$$

or

$$b_1 = -\frac{1}{T} \int_0^T f(t) \sin \left(2\pi \frac{1}{T} t \right) dt$$

A similar procedure results in a general expression for the b_n terms:

$$b_n = -\frac{1}{T} \int_0^T f(t) \sin \left(2\pi \frac{n}{T} t \right) dt \quad \text{for } n = 0, 1, 2, 3, \dots \quad (3.29)$$

Note the negative sign introduced by our sign convention. In Equation (3.10) we chose a negative sign before the b_n summation because if this convention is used for periodic functions, then it will be consistent with Fourier integral notation for continuous functions as well when we later discuss the "time-shifting theorem".

For aperiodic Fourier theory, American mathematicians would also use a negative sign before the b_n coefficients. The choice of sign is important in a physical problem such as determining velocity in the frequency domain by using phase shifts instead of time shifts. The sign convention used in this book is the one used by Bracewell [26] and by mathematicians. That is, we use Equation (3.10) instead of equation (3.11). For the opposite notation for spatial Fourier transformation see Aki and Richards [3, p. 129-130]. Aki and Richards use our sign convention for Fourier transformation of time dependence.

So $f(t)$ could be synthesized if the amplitudes $2A_n$ of the sinusoids and the phase angles α_n are known, or if the coefficients a_n and b_n are known.

3.3.6 Numerical Example

Comparison of Fourier coefficients determined from simultaneous linear equations (3.14), (3.15) and *Mathematica*'s procedure called **Fourier** is shown below. The input is the same as that used in Section 3.3.2.

```
m=5;
Array[r, {m,m}];
v={1,2,3,4,5};
a0=Table[1,{i,1,m}];
a1=Table[2*Cos[i*Pi/m],{i,0,8,2}]//N;
a2=Table[2*Cos[i*Pi/m],{i,0,16,4}]//N;
b1=-Table[2*Sin[i*Pi/m],{i,0,8,2}]//N;
b2=-Table[2*Sin[i*Pi/m],{i,0,16,4}]//N;
r={a0,a1,a2,b1,b2};r=Transpose[r];
s=LinearSolve[r,v]
```

The output from **LinearSolve** is

$$s = \underbrace{\begin{matrix} a_0 \\ 3. \end{matrix}, \begin{matrix} a_1 \\ -0.5, \end{matrix} \begin{matrix} a_2 \\ -0.5, \end{matrix} \begin{matrix} b_1 \\ 0.688191, \end{matrix} \begin{matrix} b_2 \\ 0.16246 \end{matrix}}_{}$$

and that from **Fourier** is

$$s = \underbrace{\begin{matrix} a_0 + i b_0, \\ 3. + 0i, \end{matrix} \begin{matrix} a_1 + i b_1, \\ -0.5 + 0.688191i, \end{matrix} \begin{matrix} a_2 + i b_2, \\ -0.5 + 0.16246i, \end{matrix} \begin{matrix} a_{-2} + i b_{-2}, \\ -0.5 - 0.16246i, \end{matrix} \begin{matrix} a_{-1} + i b_{-1}, \\ -0.5 - 0.688191i \end{matrix}}_{}$$

After scaling the *Mathematica* output by $1/\sqrt{m}$ and taking the complex conjugate of the Fourier transform (because *Mathematica*'s default sign convention was used), the values of a_n and b_n are identical.

3.3.7 Dirichlet conditions

Given the coefficients a_n and b_n , the Dirichlet conditions, named after the German mathematician Peter Dirichlet, define the criteria required for convergence of a Fourier series summation to $f(t)$. The function $f(t)$ is assumed to be piecewise continuous, which means that $f(t)$ is continuous everywhere except at a finite number of points. The function shown in the left-hand side of Figure 3.6 is piecewise continuous ("nonsmooth", also called "piecewise smooth") with a single discontinuity. On the right is a discontinuity in a two-dimensional function shown as a surface.

If only a finite number of Fourier coefficients a_n and b_n are used in the summation of Equation (3.10) then near points of discontinuities a "ringing" occurs called the Gibbs phenomena because of the non-convergence of the finite Fourier summation. A pulse that starts at "0" and has an amplitude of "1" is shown

in Figure 3.7. The a_n and b_n coefficients are computed using Equations (3.22) and (3.29). If a small number of coefficients is used to synthesize the pulse then the Gibbs ringing is prominent. If a large number of Fourier coefficients is used to synthesize the pulse then the ringing is minimized as shown in the figure, although the series does not converge to either “0” or “1”; careful examination of Figure 3.7 shows that the series converge to exactly “1/2” regardless of the number of coefficients used. The series would converge to “1/2” even if it were possible to use an infinite number of a_n and b_n coefficients.

The Dirichlet conditions can be stated as follows:

1. The function is a piecewise continuous function with a finite number of discontinuities and a finite number of minima and maxima,
2. The function can be expanded in a Fourier series that converges to the function where it is continuous and converges to the mean of the positive and negative limits at the points of discontinuity.
3. The function is absolutely integrable:

$$\int_{-\infty}^{\infty} |f(t)| dt < \infty$$

Mathematica program to illustrate Gibbs phenomena.

```
(*          FourierSeries          *)
```

```
<< Calculus`FourierTransform`;
<< Graphics`MultipleListPlot`;
f = UnitStep[t + 0.25] - UnitStep[t - 0.25]; Plot[f, {t, -1, 1},
  AspectRatio -> 0.4]; ListPlot[Table[f, {t, -1, 1, 0.01}],
  PlotRange -> All, PlotJoined -> True, AspectRatio -> 0.4];
F = FourierSeries[f, t, 7]; fsum = Chop[Table[F, {t, -1, 1, 0.01}]];
times = Table[t, {t, -1, 1, 0.01}]; ListPlot[Transpose[{times, fsum}],
  PlotJoined -> True, PlotRange -> All, AspectRatio -> 0.4];
MultipleListPlot[Transpose[{times, Table[f, {t, -1, 1, 0.01}]}],
  Transpose[{times, fsum}], PlotJoined -> {True, True}];
```

3.3.8 Summary

In summary,

$$a_n = \frac{1}{T} \int_0^T f(t) \cos\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \quad (3.30)$$

$$b_n = -\frac{1}{T} \int_0^T f(t) \sin\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \quad (3.31)$$

(with a negative sign before the b_n integral). Equations (3.10), (3.30), and (3.31) are those associated with functions that are periodic. Equations (3.30) and (3.31) are known as Fourier cosine and sine transforms, respectively.

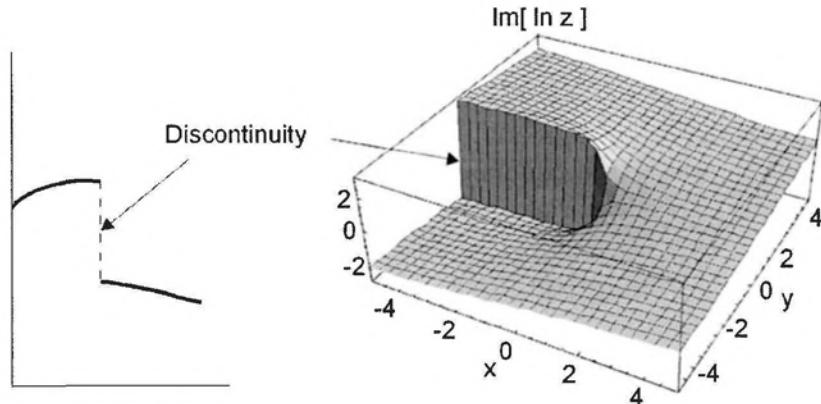


Figure 3.6: Left: A piecewise-continuous function with a single discontinuity. Right: A piecewise discontinuity in the two-dimensional function $z = x + iy$ plotted as the surface $\text{Im}[\ln z]$ versus x, y . Figure from *Mathematica* notebook by Weisstein [189]. Continuous.nb → Dirichlet.cdr → Dirichlet.wmf

A positive sine wave of frequency $\nu = 62.5$ Hertz is shown in Figure 3.8. The a_n , b_n , A_n coefficients and the phase angle spectrum for this sinusoid are shown below the trace. The spectrum has been computed for both positive and negative frequencies. The a_n coefficients are zero because the input sine wave is an odd function with respect to time. The first spike in the plot of the b_n coefficients corresponds to a frequency of $\nu = -62.5$ Hertz, and the second to a frequency of $\nu = +62.5$ Hertz. Note that $b_{-n} = -b_n$.

A cosine wave is shown in Figure 3.9 with corresponding coefficients a_n , b_n , and A_n . The spectrum has again been computed for both positive and negative frequencies (Figure 3.9). The first spike in the plot of the a_n coefficients corresponds to a frequency of $\nu = -62.5$ Hertz, and the second to a frequency of $\nu = +62.5$ Hertz. Note that $a_{-n} = a_{+n}$. The b_n coefficients are zero because the input cosine wave is an even function with respect to time.

The sum of a cosine wave plus a sine wave of the same frequency is shown in Figure 3.10 along with Fourier spectra. Spectra of other kinds of time series are treated similarly. A seismic trace (“Trace 3”) sampled at an interval of 1 ms is shown in Figure 3.12. The Fourier amplitude spectrum of Trace 3 is shown in Figure 3.12b. The trace was sampled at an interval of 1 ms and is a sampled version of one of the traces of the shot record from Bellshill Lake shown in Figure 3.11. Most of the energy lies between 0-110 Hertz. This interval is shown in more detail in Figure 3.12c.

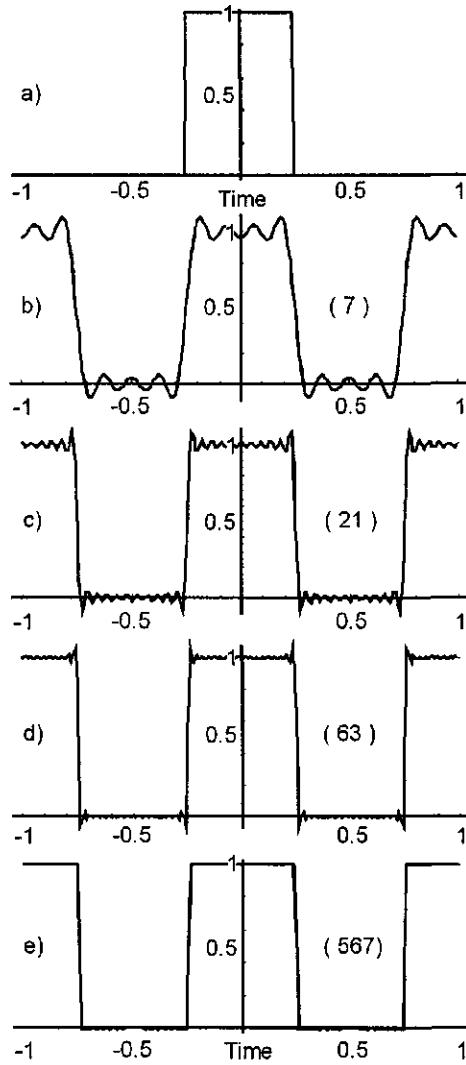


Figure 3.7: (a): A piecewise-continuous function (a pulse) with two discontinuities. Pulse starts at "0" and has a maximum value of "1". (b): The synthesized function using 7 Fourier coefficients and showing the Gibbs ringing. (c) 21 coefficients. (d) 63 coefficients. (e) 567 coefficients. Numbers in parentheses denote number of coefficients used in *Mathematica* "FourierSeries" program listed in text. All Fourier summations, regardless of number of coefficients used, converge to 0.5 at $t = \pm 0.25$. By the Dirichlet conditions, even if it were possible to use an infinite number of coefficients the summation would still converge to $(1 - 0)/2 = 0.5$ at $t = \pm 0.25$. GibbsPhenomena.cdr,nb → Gibbs.wmf

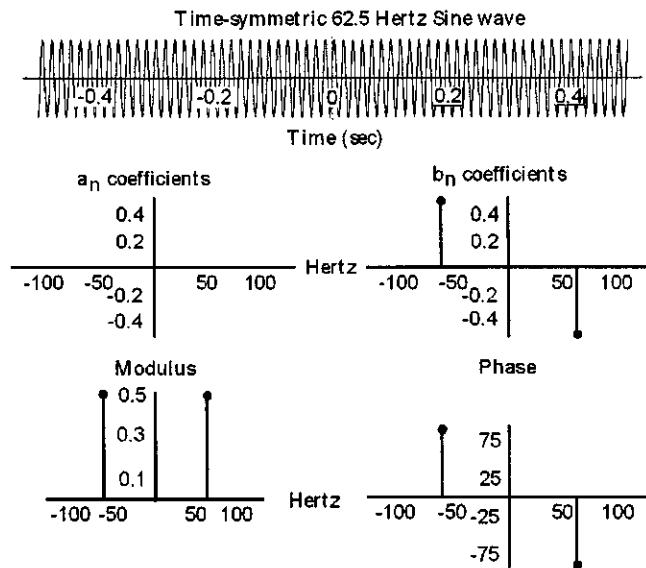


Figure 3.8: Top: Time-symmetric positive sine wave of 62.5 Hertz. Middle left: a_n (all zero). Middle right: b_n . Note $b_{-n} = -b_n$. Bottom left: Modulus. Bottom right: Phase angles in degrees. The phase angle $\theta = \tan^{-1}[b_n/a_n] = \infty = 90^\circ$. ($\sin 62.\text{ma} \rightarrow \sin 62.\text{cdr}$)

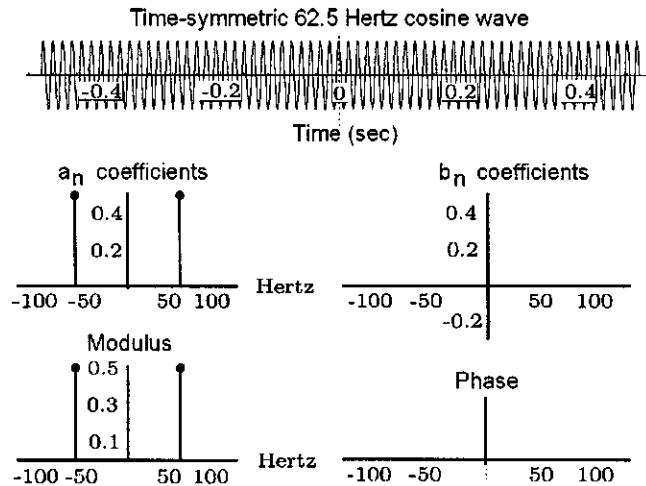


Figure 3.9: Top: Time-symmetric cosine wave of 62.5 Hertz. Middle left: a_n coefficients. Note $a_{-n} = a_n$. Middle right: b_n (all zero). Bottom left: Modulus. Bottom right: Phase angle spectrum in degrees (all zero). The phase angle $\theta = \tan^{-1}[b_n/a_n] = 0 = 0^\circ$. ($\cos 62.\text{ma}$)

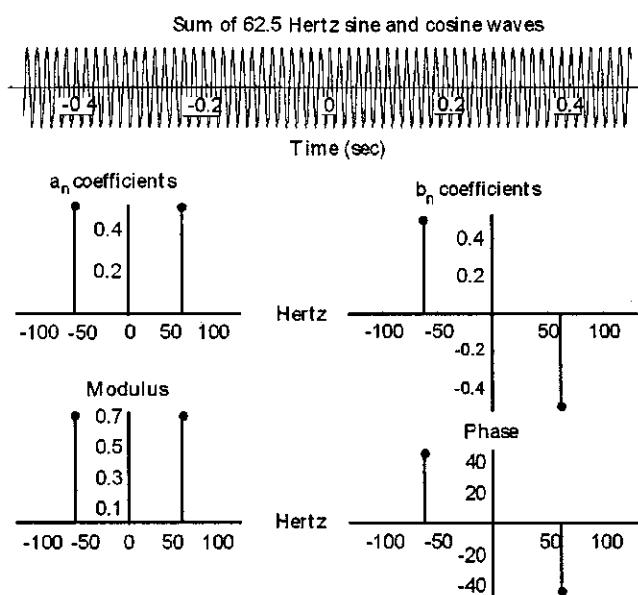


Figure 3.10: Top: Sum of time-symmetric sine and cosine waves each of frequency 62.5 Hertz. Middle left: a_n coefficients. Note $a_{-n} = a_n$. Middle right: b_n coefficients. Note $b_{-n} = -b_n$. Bottom left: Modulus. Bottom right: Phase angle spectrum in degrees. (sico62.ma → sico62.cdr)

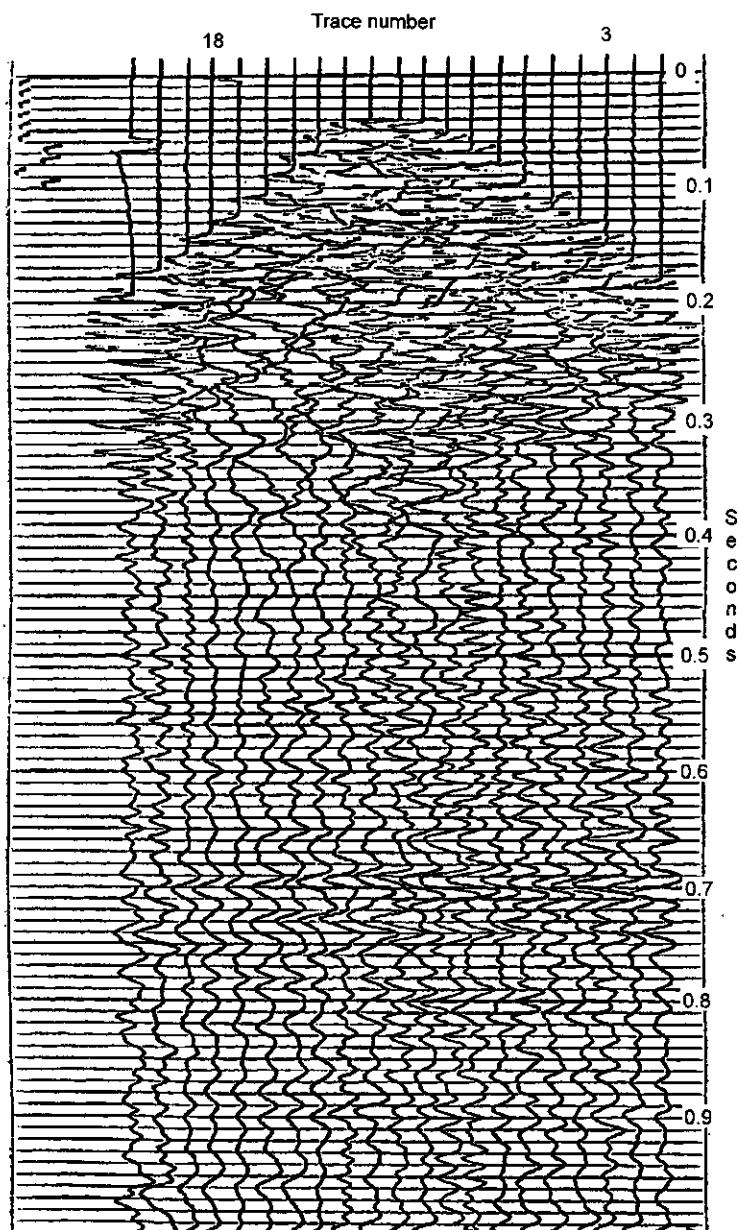


Figure 3.11: Early shot record from Bellshill Lake, Alberta. Courtesy of Geophysical Services, Inc. Shot depth is 150 feet. Shot number SP 22. (Bellshill-Lake.bmp)

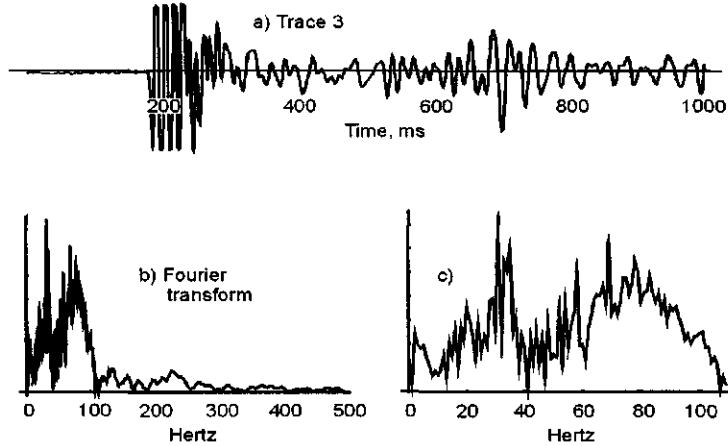


Figure 3.12: Top: An early analog seismic trace ("Trace 3"). Lower left: Modulus of Fourier spectrum (modulus) from 0-500 Hertz. Lower right: Expanded plot of lower left showing the frequency interval from 0-110 Hertz. (Trace3.mb \Rightarrow trace3.cdr)

Although we will be implementing Equations (3.30) and (3.31) on the computer by means of a finite summation, we nevertheless retain the notation of continuous integration (the integral sign) in equations

$$a_n = \frac{1}{T} \int_0^T f(t) \cos\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \quad (3.30)$$

$$b_n = -\frac{1}{T} \int_0^T f(t) \sin\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \quad (3.31)$$

and summation using only the discrete frequencies n/T in

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) - 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (3.10)$$

to emphasize that a Fourier synthesis using Equation (3.10) is *always* a discrete summation only over certain frequencies and *never* a continuous integration over all frequencies. Only certain frequency components $= n/T$ can be used.

Although (3.31) states that *all* values of time are used, in practice the field data are sampled at discrete time intervals Δt so that it is neither possible nor necessary to integrate over all values of time. On the other hand, the expression for exactly synthesizing a function that never repeats *must* use *all* values of frequency. Implementation of Equation (3.30) or (3.31) on the computer requires us to approximate *any* integral with a discrete summation, i.e., for

Equation (3.30)

$$a_n \approx \frac{\Delta t}{T} \sum_{t=0}^{t=T} f(t) \cos\left(2\pi \frac{n}{T} t\right) \quad (3.30a)$$

$$\approx \frac{\Delta t}{m\Delta t} \sum_{k=0}^m f(k\Delta t) \cos\left(2\pi \frac{n}{m\Delta t} k\Delta t\right)$$

$$\approx \frac{1}{m} \sum_{k=0}^m f(k\Delta t) \cos\left(2\pi \frac{n}{m} k\right) \quad (3.30c)$$

where m is the number of points in the time series.

Determination of the amplitude coefficients a_n and b_n gives a discrete “line spectrum” for periodic functions. The function $f(t)$ is synthesized by summing together cosines and sines of amplitude a_n and b_n , respectively, using only those frequencies equal to n/T , $n = 0, 1, 2, \dots$. These sinusoids are harmonics of each other because those that make up $f(t)$ must make one, two or any *exact whole number* of cycles in the given interval of time T so that $f(t)$ *will* repeat itself with period T . Such a set of sinusoids is called a “harmonic series,” and the sinusoids are called “harmonics.” The characteristic of such a series is that their frequencies are whole number multiples of the basic (fundamental) frequency $1/T$. For a discussion of early analog devices for the computation of Fourier series coefficients the reader is referred to the excellent little book by Barber [12].

Examples

As examples of Fourier decomposition followed by synthesis, we compute the Fourier coefficients for rectangular and triangular sampled continuous pulses. These sinusoids (cosine and sine waves), plotted to the same relative scale for a time duration equal to three times the original period are shown in Figures 3.13 and 3.14. The sinusoids (cosine and sine waves) are plotted below the input function. Cosine waves are solid lines; sine waves are dashed. Corresponding frequencies plotted are $0/T, 1/T, 2/T, \dots, n/T$. The algebraic sum of all of these sinusoids is shown below the sinusoids. The sums are seen to be a periodic repetition of one period T of the input data. Zero time is shown below the sum. The sinusoid that dominates the sums in Figures 3.13 and 3.14 is a low-frequency sinusoid. The amplitudes of some of the sinusoids are too small to be identified on the plot.

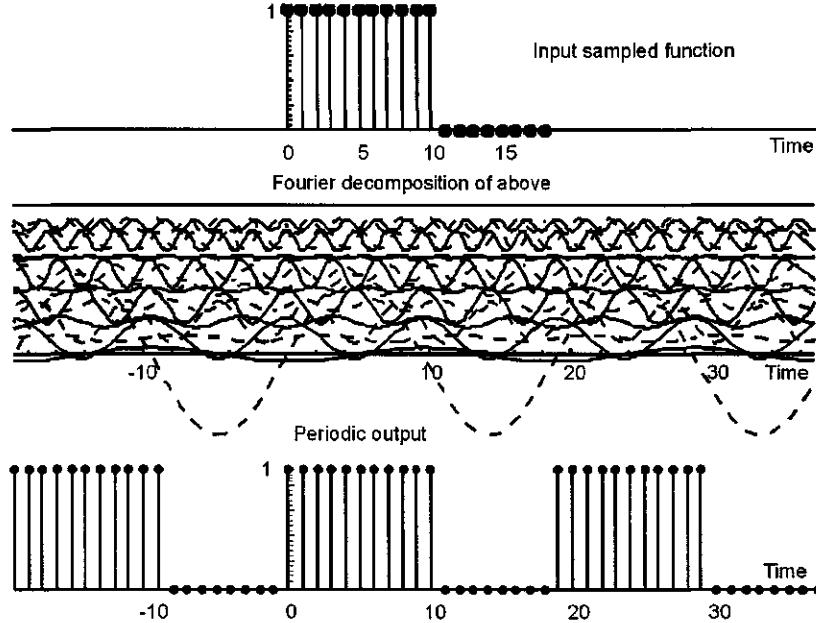


Figure 3.13: Top: Rectangular pulse. Middle: Corresponding harmonics plotted to correct relative scale. Bottom: Summation of these resulting in a rectangular pulse that is now periodic in time. Labla.nb \Rightarrow labla.cdr \Rightarrow labla.wmf

A *Mathematica* program for forward and inverse Fourier transformation using Equations (3.30), (3.30), and (3.31) is listed below. The input data are $f(t) = 1, 2, 3, 4, 5$.

```
ft={1,2,3,4,5};
nt=5;
dt=0.004;
period=nt*dt;
nyquist=1/(2*dt);
dfreq=1/period;
nf=nyquist/dfreq+1.1;
w=2*Pi*dfreq*dt;
cospart[n_]=Sum[(dt/period)*ft[[i]]*Cos[w*n*(i-1)],{i,1,nt}];
sinpart[n_]=Sum[-dt/period*ft[[i]]*Sin[w*n*(i-1)],{i,1,nt}];
an=Table[cospart[n-1],{n,1,nf}]
bn=Table[sinpart[n-1],{n,1,nf}]
synthesis[n_]=2*Sum[an[[i]]*Cos[w*n*(i-1)],{i,1,nf}]-2*Sum[bn[[i]]*Sin[w*n*(i-1)],{i,1,nf}]-an[[1]];
Table[synthesis[n],{n,0,4}];
Chop[%]
```

The output is

$$\begin{aligned} a_0, a_1, a_2 &= 3., -0.5, -0.5 \\ b_0, b_1, b_2 &= 0, 0.688191, 0.16246 \\ \text{synthesized } f(t) &= 1., 2., 3., 4., 5. \end{aligned}$$

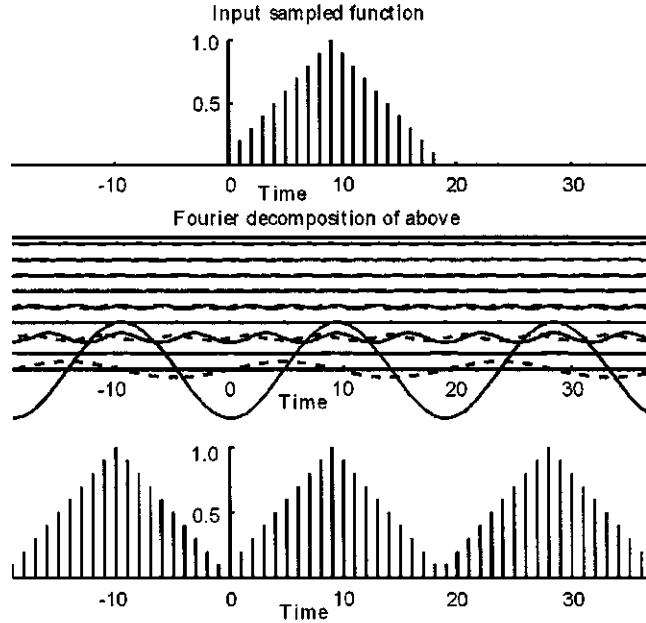


Figure 3.14: Top: Triangular pulse. Middle: Corresponding harmonics plotted to correct relative scale. Bottom: Summation of these resulting in a triangular pulse that is now periodic in time. lab1a.wmf and lab1d.wmf

3.3.9 Sign Convention Revisited

The importance of knowing which sign convention is used is nicely demonstrated by the *Mathematica* software itself. In *Mathematica* using the default `Fourier` parameters the sign of the b_n coefficients is positive as in

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) + 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (3.11)$$

In their Signals and Systems add-on package, however, it is negative as in

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) - 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (3.10)$$

The difference, therefore, is that their Standard package gives Fourier coefficients that are the complex conjugate of those from their Signals and Systems package, as shown from the following *Mathematica* program.

```
Needs["SignalProcessing`"]
list=Table[Sin[n], {n,0,10}];
Sqrt[Length[list]]*Fourier[list]
```

```
DiscreteFourierTransform[list]
```

The (scaled) output from Fourier is

```
{1.41119 +0. I, 1.89183 -0.898211 I, -2.85402+3.64178 I, -0.113634+0.725012 I,
0.149521 +0.316167 I, 0.220708 +0.0939228 I, 0.220708 -0.0939228 I,
0.149521 -0.316167 I, -0.113634-0.725012 I, -2.85402-3.64178 I,
1.89183 +0.898211 I}
```

while that from DiscreteFourierTransform is

```
{1.41119 +0. I, 1.89183 +0.898211 I, -2.85402-3.64178 I, -0.113634-0.725012 I,
0.149521 -0.316167 I, 0.220708 -0.0939228 I, 0.220708 +0.0939228 I,
0.149521 +0.316167 I, -0.113634+0.725012 I, -2.85402+3.64178 I,
1.89183 -0.898211 I}
```

This difference is of no consequence if you just want the amplitude (modulus) of a frequency component; i.e., if you want

$$\sqrt{a_n^2 + b_n^2}$$

The sign difference becomes important, however, if you want to compute the phase (argument) and use the *time-shifting theorem* (Section 3.5, page 93) in a practical application. For example,

$$\theta_n = \tan^{-1} \left[\frac{b_n}{a_n} \right]$$

will give results with different signs.

3.3.10 Seismogram from the Atlantic Coastal Plain

The seismogram from the Atlantic Coastal plain shown in Figure 3.1 is repeated at the top of Figure 3.15 along with the unfiltered Fourier spectra, the modified Fourier spectra after zeroing out those Fourier components primarily associated with the Rayleigh surface waves, and the inverse Fourier transforms of each trace. There is an improvement in the continuity of the basement reflection; however, we can do better. In Figure 3.15, the Fourier components that are associated primarily with the Rayleigh wave noise were simply zeroed out. This is generally never a good idea because of the abrupt effect on the amplitude spectrum. It is more desirable to zero out gradually the undesirable components using some kind of "window" that tapers on and off so that abrupt changes in the amplitude spectrum are avoided. This is discussed further below.

For clarity of plotting, each trace is scaled so that its variance is unity ("Standardize") and its mean is zero ("ZeroMean"). The forward transform is computed by Fourier and the inverse by InverseFourier. Portions of the amplitude spectrum were zeroed out by the *Mathematica* function ReplacePart. The sign convention for the imaginary part (the b_n coefficients) of the complex amplitude spectrum is not important in this application. The spectrum was smoothed slightly using the *Mathematica* function MovingAverage.

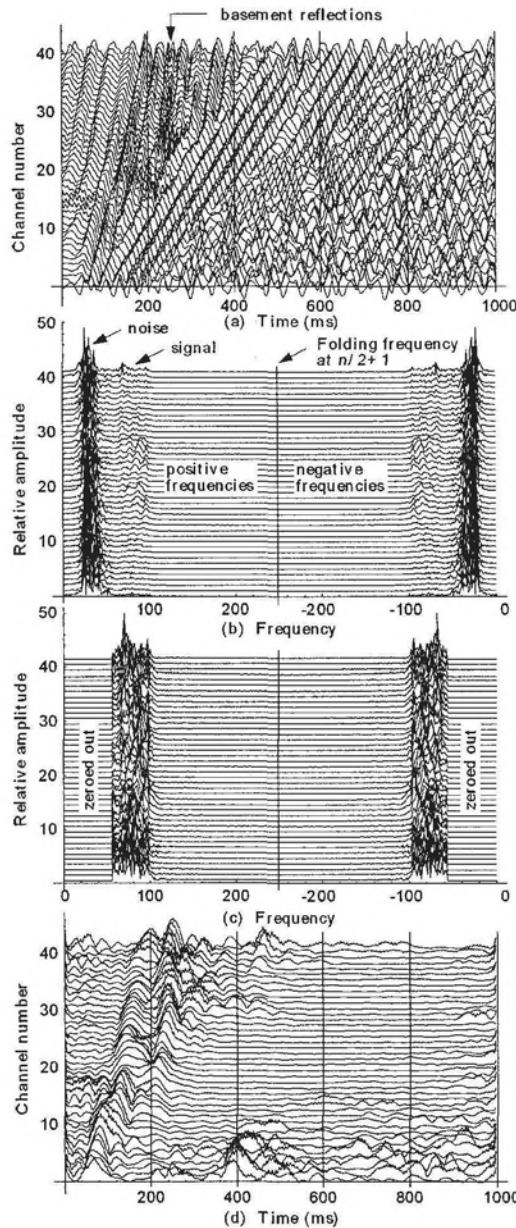


Figure 3.15: Shot record from Atlantic Coastal Plain near Richmond, VA. (a) Top: Rayleigh waves obscure continuity of reflections from crystalline basement. (b) Fourier amplitude spectrum of above seismogram. (c) Amplitude spectrum after zeroing out first 56 Fourier coefficients. (d) Bottom: Inverse Fourier transform now shows better continuity of reflections from crystalline basement. (ACPlain.cdr → FilteredACPSHOTRecord.wmf)

The *Mathematica* program that generated the results shown in Figure 3.15 is given below.

```
<<Statistics`DescriptiveStatistics`
<<Statistics`DataSmoothing`
data=ReadList["c:\\Book\\Data\\Seismic\\acpt042.txt",Table[Number,{500}]];
TracePlot=
  Table[ListPlot[Standardize[ZeroMean[Part[data,i]]]+(i-1),PlotJoined->True,
    DisplayFunction->Identity],{i,1,42}]
TimingLines=Table[{i_,j_},{i,100,500,100},{j,-1,44}];
TimingLines=
  Table[ListPlot[Part[TimingLines,i],PlotJoined->True,
    DisplayFunction->Identity],{i,1,5}]
tp=Show[TracePlot,TimingLines,DisplayFunction->$DisplayFunction]
Display["c:\\Book\\Figures\\ACPSection-Unfiltered.wmf",tp,"Metafile"]
UnfilteredFourierPlot=
  Table[ListPlot[Abs[Fourier[Standardize[ZeroMean[Part[data,i]]]]]+(i-1),
    PlotJoined->True,DisplayFunction->Identity],{i,1,42}]
ufp=Show[UnfilteredFourierPlot,DisplayFunction->$DisplayFunction]
Display["c:\\Book\\Figures\\ACPSection-OriginalFourierSpectrum.wmf",ufp,"Metafile"]
FilteredFourierPlot=
  Table[ListPlot[
    Standardize[
      Abs[ReplacePart[Fourier[Part[data,i]],0,
        Join[Table[{i}, {i,1,56}],Table[{i},{i,444,500}]]]]+i,
      PlotJoined->True,DisplayFunction->Identity],{i,1,42}]
ffp=Show[FilteredFourierPlot,DisplayFunction->$DisplayFunction]
Display["c:\\Book\\Figures\\ACPSection-ModifiedFourierSpectrum.wmf",ffp,"Metafile"]
SmoothedFilteredInverseFourierPlot=
  Table[ListPlot[
    Standardize[
      MovingAverage[
        Abs[InverseFourier[
          ReplacePart[Fourier[Part[data,i]],0,
            Join[Table[{i}, {i,1,56}],Table[{i},{i,444,500}]]],3]]+i,
        PlotJoined->True,DisplayFunction->Identity],{i,1,42}]
sfifp=Show[SmoothedFilteredInverseFourierPlot,TimingLines,
  DisplayFunction->$DisplayFunction]
Display["c:\\Book\\Figures\\ACPSection-Filtered.wmf",sfifp,"Metafile"]
```

3.3.11 Independence of the Fourier coefficients

A conventional Fourier "harmonic" analysis yields spectral amplitude coefficients at the frequencies

$$\dots - \frac{4}{T} - \frac{3}{T} - \frac{2}{T} - \frac{1}{T} + \frac{0}{T} + \frac{1}{T} + \frac{2}{T} + \frac{3}{T} + \dots$$

where T is the length of the analysis window. These frequencies are harmonics of each other and do not interfere with each other. That is, the harmonic at $1/T$ does not interfere with the one at $2/T$, and vice versa. The Fourier transform of the sum of any number of sinusoids would be a series of Dirac delta functions located exactly at the corresponding sinusoid frequencies with no interference if the analysis window is of infinite length. For example, the Fourier transform of

$$\cos(\omega_0 t)$$

is shown in Figure 3.16.

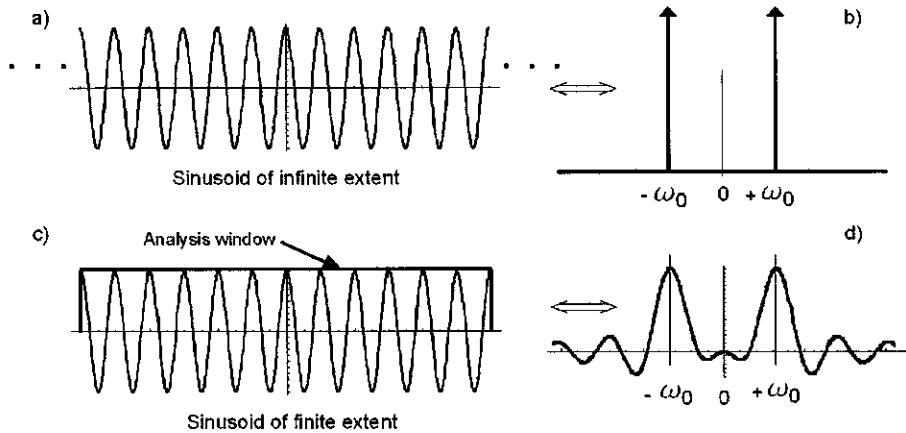


Figure 3.16: (a) A cosine wave of frequency ω_0 that extends from $-\infty$ to ∞ .
 (b) The Fourier transform of the cosine wave.
 (c) The sinusoid multiplied by an analysis rectangular time window; i.e., a sinusoid of finite length.
 (d) Fourier transform of the windowed cosine wave as defined by Equation (3.32). `IndependenceOffourierCoefficients.nb,cdr` ⇒ `SingleSinusoid.wmf`

Selecting just a portion of the signal, is, however, equivalent to multiplication of the sinusoid by a rectangular “analysis window”. Multiplication in the time domain is equivalent to convolution in the frequency domain as per Equation (3.65) on page 124. So the Fourier transform of the window is superimposed over the single Fourier component with the result as shown in Figure 3.16d. The Fourier transform becomes

$$F(\omega) = \tau \operatorname{Sign}(\tau) (1 + \operatorname{Sinc}[2\tau\omega]) \quad (3.32)$$

where τ is the width of the analysis window.

It is clear from Figure 3.16 that, although the maximum spectral amplitude is still at a frequency of ω_0 , frequencies on either side of ω_0 now have non-zero amplitudes, even though only a single sinusoid is present in the data. The effect of windowing has been to introduce energy throughout the entire spectrum. The single Fourier component ω_0 reaches out on each side of ω_0 and is superimposed on other components that would be present in a Fourier line spectrum. For Figure 3.16a there are no other components so there is no side lobe distortion or “window leakage” and the true spectrum of the single sinusoid can be determined.

Three sinusoids are shown in Figure 3.17. The result of truncation of the time series has been to introduce energy at other frequencies in the spectrum, but not to affect the frequencies equal to n/T , $n = 1, 2, 3, \dots$. Although these harmonics don't affect each other, if there really is energy in the data at non-harmonic frequencies (such as $2.4/T$), then this will affect the values of

the Fourier coefficients at $1/T$, $2/T$, $3/T$, etc., as shown in Figure 3.17. All

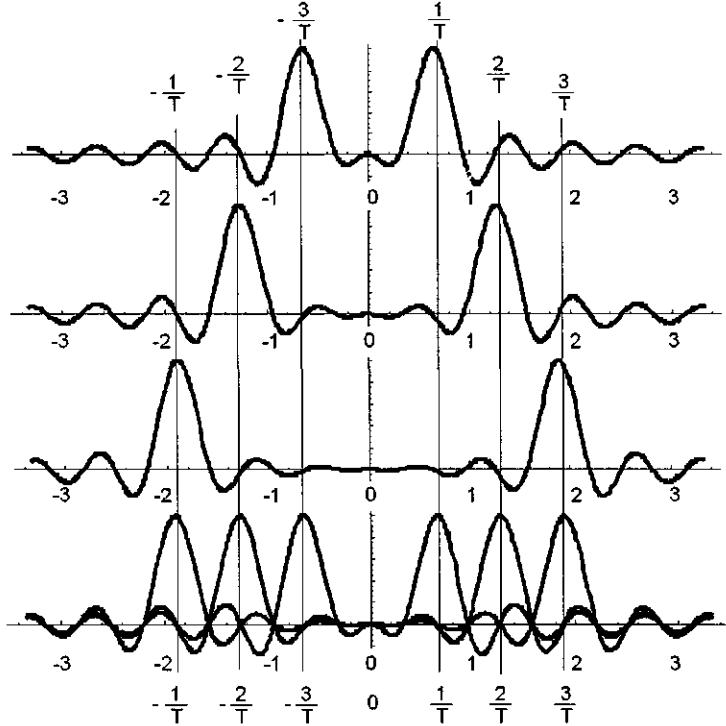


Figure 3.17: Three harmonics of frequency $\nu = 1/T$, $2/T$, and $3/T$ superimposed and showing that their side lobes do not interfere with each other at the frequencies $1/T$, $2/T$, and $3/T$. IndependenceOfFourierCoefficients.nb \Rightarrow IndependenceOfFourierCoefficients.nb \Rightarrow ThreeSinusoids.wmf

seismic data do, of course, contain frequencies that are not harmonics of each other. The effects of the addition of a single sinusoid of frequency $2.5/T$ to those shown in Figure 3.17 is shown in Figure 3.18. Now the true amplitudes of the harmonics $1/T$, $2/T$, and $3/T$ can no longer be recovered because of interference associated with the side lobes of the sinusoid of frequency $2.5/T$. This raises the whole question of seismic resolution; e.g., how long does the seismic trace have to be in order to resolve adjacent peaks in the Fourier spectrum, and would a different window shape help? The question of seismic resolution has been addressed in Volume 12 of this series by Berkhouit [18]. The effect of the side lobes can be decreased as shown in Figure 3.20 but only at the expense of the central lobe, which becomes wider. Thus, we are faced with a tradeoff between a small central lobe but with high side lobes, and small side lobes but a wider central lobe.

Window design includes those with a flat central part as well as those shown

in Figure 3.20. Side lobes associated with these windows are relatively prominent, however [18].

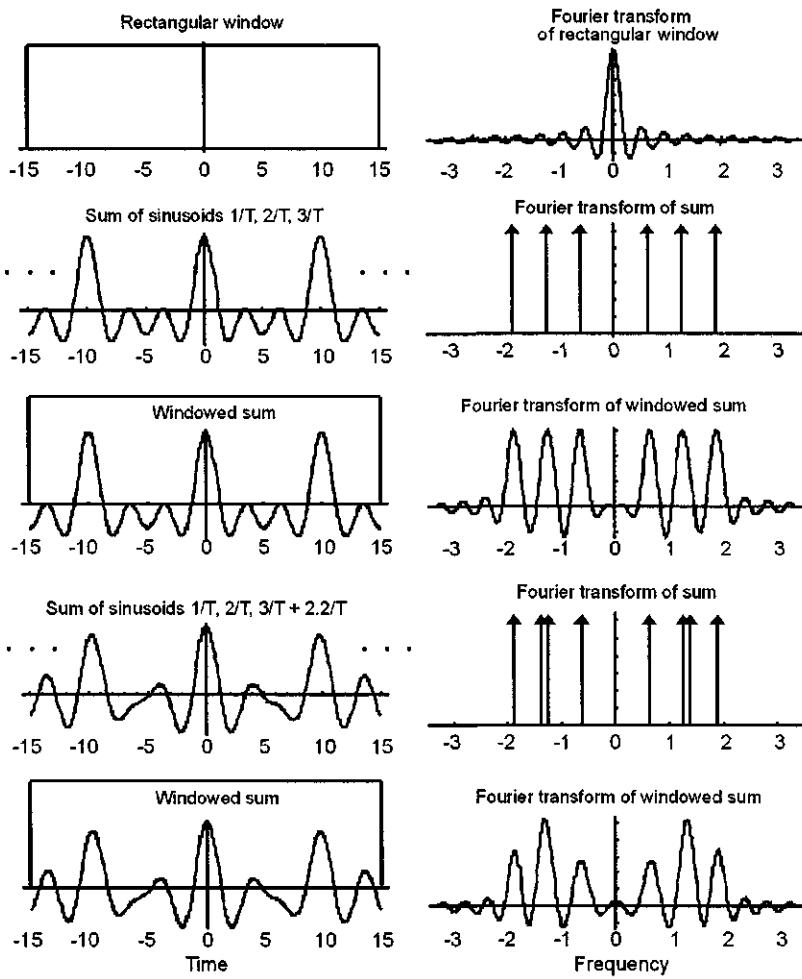


Figure 3.18: Three harmonics $1/T$, $2/T$, and $3/T$ plus a fourth $2.2/T$ superimposed and showing that the side lobes of $2.2/T$ interfere with each of the frequencies $1/T$, $2/T$, and $3/T$. IndependenceOfFourierCoefficientsThreePlusOne.nb \Rightarrow ThreeSinusoidsPlusOne.cdr \Rightarrow ThreeSinusoidsPlusOne.wmf

If the window is opened to a width wider than that shown in Figure 3.16, then the effect on adjacent frequencies is reduced, as shown in Figure 3.19, until the effect vanishes as shown in Figure 3.16b. The lesson here is that if you are only interested in the interpretation of a small portion, say 0.5 sec, of a seismic trace after enhancement by some bandpass or other cosmetic filter, then

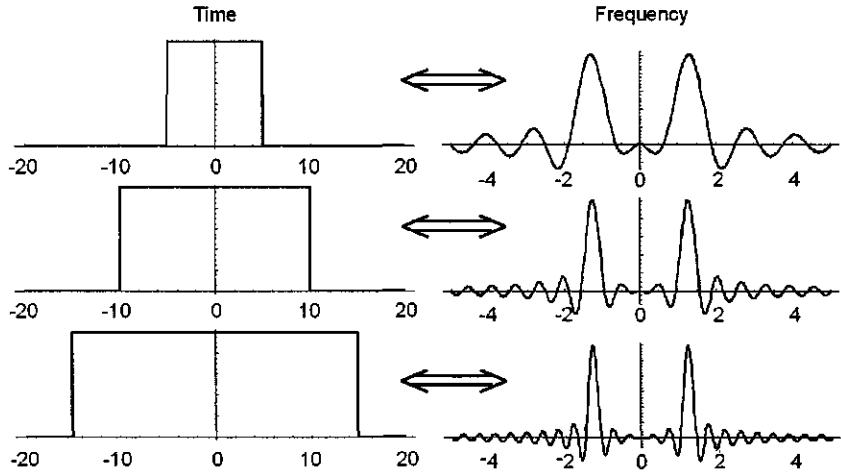


Figure 3.19: Left column: Various analysis window widths for windowed sinusoid of frequency $1/T$. Fourier transforms of left-hand column. As the window width is increased the magnitude of the Fourier transform increases until at a window of infinite width the magnitude of the Fourier transform $\rightarrow \infty \rightarrow \delta(1/T)$. VariousWindowWidths.nb,cdr \Rightarrow VariousWindowWidths.wmf

as much as possible of the entire trace should be filtered first before analysis of the smaller segment. This reduces the effects of the window on the data.

The result of truncation of the time series has been to introduce energy at other frequencies in the spectrum, but not to affect the frequencies equal to $n/T, n = 0, 1, 2, 3, \dots$. If there really is energy at other frequencies (such as $2.5/T$), however, then this will affect the values of the Fourier coefficients at both $1/T$ and $2/T$, as shown in Figure 3.18.

In summary,

1. The spectrum, $W(\nu_n)$, of a rectangular time-domain function has the shape

$$W(\nu_n) = \frac{\sin(2\pi\nu_n T/2)}{2\pi\nu_n T/2} = \frac{\sin(\pi \frac{n}{T} T)}{\pi \frac{n}{T} T}$$

which has zeros at the frequencies n/T for $n = 1, 2, 3, \dots$ as shown in Figure 3.17. Thus, if values of the Fourier coefficients are computed at frequencies

$$\frac{1}{T}, \frac{2}{T}, \frac{3}{T}, \dots$$

then these Fourier coefficients are independent of each other; i.e., an amplitude associated with one frequency component does not interfere with the other harmonics.

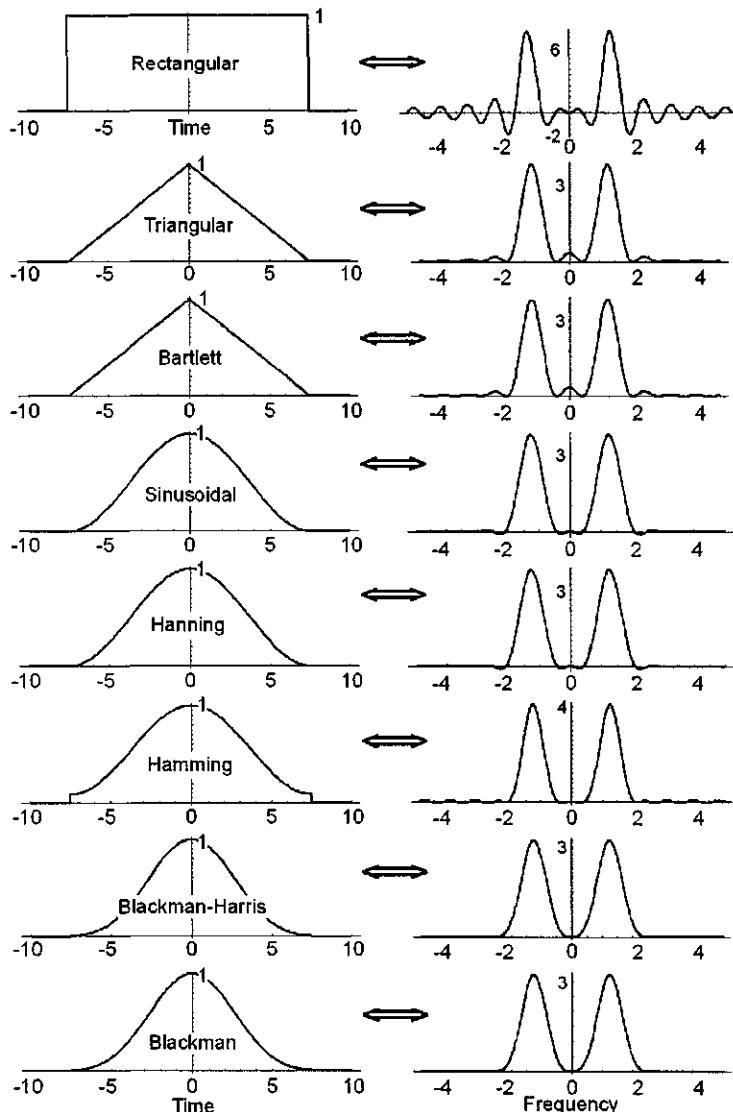


Figure 3.20: Left column: Various analysis window shapes for windowed sinusoid of frequency $1/T$. Right column: Fourier transforms of left-hand column.
 VariousWindowShapes.nb \Rightarrow IndependenceOfFourierCoefficients.cdr \Rightarrow VariousWindowShapes.wmf

2. Seismic data contain frequencies between $1/T, 2/T, 3/T, \dots$ that do interfere with those at $1/T, 2/T, 3/T, \dots$
3. Analysis of data through windows of various shapes reduces the amplitude of side lobes but at the expense of increasing the width of the central lobe.

Windows of various shapes such as those shown here are crafted for use either in the time-domain or the frequency-domain. A thorough discussion of the properties of side lobes and spectral analysis windows is beyond the scope of this volume. They have been explored extensively in the literature [89, 101, 195]. The leakage problem and therefore the problem of resolution is largely based on visual analysis. As noted by Berkhouit [18, p. 16], new computer-based techniques are being developed to realize, among other things, the separation of frequency components that are beyond the limits of direct interpretation. An example by Marangakis [116] of the detection of a thin-bed that is not visually apparent on the seismogram is shown in Figure 5.12 on page 225.

3.4 From Fourier Series to Fourier Integrals

The equation

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) - 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (3.33)$$

describes a function $f(t)$ that is periodic in the time domain with period T . In order to have this periodicity the coefficients a_n and b_n must be defined as

$$a_n = \frac{1}{T} \int_0^T f(t) \cos\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \quad (3.34)$$

$$b_n = -\frac{1}{T} \int_0^T f(t) \sin\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \quad (3.35)$$

where

$$\frac{n}{T} = \nu$$

Using the same definitions for average values of sine and cosine functions, and products of sine and cosine functions, as were used in Section 3.3.5 above, the following equations describe the Fourier transform pairs for functions of time $f(t)$ that do *not* repeat themselves:

$$f(t) = \int_{-\infty}^{\infty} a(\nu) \cos(2\pi\nu t) d\nu - \int_{-\infty}^{\infty} b(\nu) \sin(2\pi\nu t) d\nu \quad (3.36)$$

$$a(\nu) = \int_{-\infty}^{\infty} f(t) \cos(2\pi\nu t) dt \quad (3.37)$$

$$b(\nu) = - \int_{-\infty}^{\infty} f(t) \sin(2\pi\nu t) dt \quad (3.38)$$

Equation (3.33) states that for a function to repeat itself with period T the only frequencies ν that can be used in the summations on the right-hand side of (3.33) are integer multiples of the fundamental frequency $\frac{1}{T}$. Such a Fourier summation using frequencies $1/T, 2/T, 3/T, \dots$ will converge to $f(t)$ as desired. That this must be so can be seen by examination of Figure 3.21. At $t = 0$ and $t = T$ each of the sinusoids starts and ends at the same place so that for times $t < 0$ and $t > T$ the summation will repeat itself with period T . On the

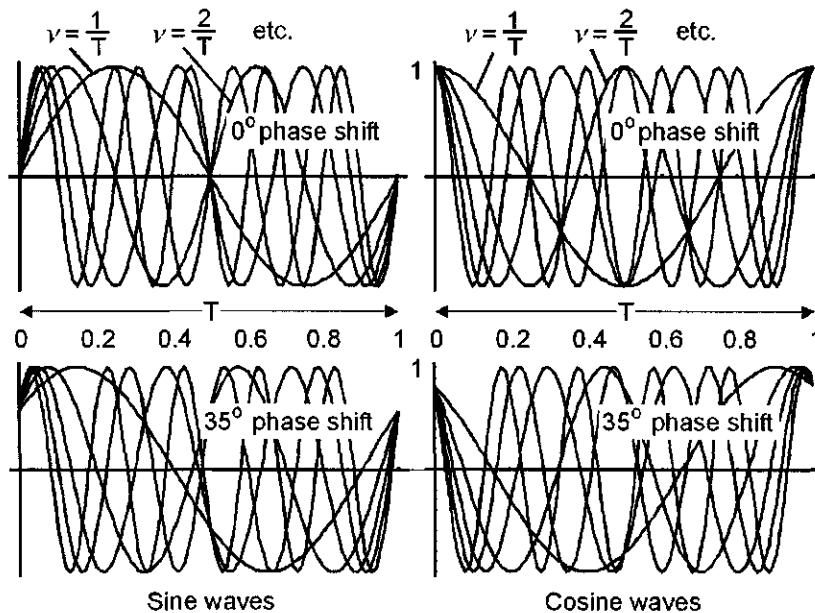


Figure 3.21: Sine waves left column, cosine waves right column. (Figure 3.5 repeated.)
Top row: No phase shift. Bottom row: 35° phase shift. The time series $f(t)$ are T sec long. As long as the frequencies used in the Fourier series summation of Equation (3.33) are integer multiples of $1/T$ then the summations will start at the same place for $t = 0$ and $t = T$. At $t = 0$ and $t = T$ each of the sinusoids starts and ends at the same place for their respective phase shifts so that for times $t < 0$ and $t > T$ the summations will repeat themselves with period T . For all times $t < 0$ and $t > T$ $f(t)$ will therefore be periodic with period T .

other hand if the function $f(t)$ is never to repeat then Equation (3.36) states that *all* frequencies must be used ($d\nu$) instead of just those that are an integer multiple of n/T .

3.4.1 Complex forms and Fourier Integral

A real, periodic function of time $f(t)$ and the Fourier coefficients a_n and b_n are described by (3.33), (3.34), and (3.35). If $f(t)$ is complex then we state without proof the corresponding Fourier transform pair for continuous, complex functions that do not repeat:

$$f(t) = \int_{-\infty}^{\infty} F(\nu)e^{+i2\pi\nu t} d\nu \quad (3.39)$$

$$F(\nu) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi\nu t} dt \quad (3.40)$$

where

$$f(t) = f_r(t) + i f_i(t) \quad (3.41)$$

$$F(\nu) = a(\nu) + i b(\nu) \quad (3.42)$$

That is, the real and imaginary parts of $f(t)$ are $f_r(t)$ and $f_i(t)$, respectively, and the real and imaginary parts of $F(\nu)$ are $a(\nu)$ and $b(\nu)$, respectively. Thus we can retain a similar notation and sign convention for a_n and b_n for the real and imaginary parts of the transforms that we used for periodic functions.

Recognizing that $\omega = 2\pi\nu$, Equations (3.39) and (3.40) can be rewritten as

$$\begin{aligned} f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{+i\omega t} d\omega \\ F(\omega) &= \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \end{aligned}$$

but because of the scalar $1/2\pi$, this formulation loses the symmetry in notation between the time and frequency domains (forward and inverse processes). Some texts recover this symmetry by defining

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

We often use the symmetrical notation of Equations (3.39) and (3.40).

The two functions $f(t)$ and $F(\nu)$ are called a *Fourier transform pair*. $F(\nu)$ is the *Fourier transform*, or *direct Fourier transform* of $f(t)$, and $f(t)$ is the *inverse Fourier transform* of $F(\nu)$. Note that the two Fourier integrals differ only in the sign of the Fourier kernel (exponential). Common shorthand mathematical notation for a Fourier transform pair is:

$$f(t) \iff F(\nu)$$

The distinction between repeating and non-repeating functions is clear from Equations (3.33) and (3.39) where for (3.39) the integration must be taken over

all frequencies ν from 0 to $\pm\infty$ instead of just integer multiples of $1/T$ as in Equation (3.33). Thus, if computations must be done by numerical integration, as is usually the case for geophysical applications, then $f(t)$ as defined by Equation (3.33) requires much less computational effort than that associated with Equation (3.39). Indeed, if $f(t)$ is the same function over the time interval T in both Equation (3.33) and (3.39) then it is pointless to undertake the extra numerical computation required by (3.39). If the distinction between (3.33) and (3.39) is not understood, however, then aliasing can result in serious computational errors as discussed later in the section on *aliasing*.

It is necessary to examine non-repeating (aperiodic) time-domain functions that are pure real. Using Euler's equations

$$\begin{aligned} e^{i2\pi\nu t} &= \cos(2\pi\nu t) + i \sin(2\pi\nu t) \\ e^{-i2\pi\nu t} &= \cos(2\pi\nu t) - i \sin(2\pi\nu t) \end{aligned}$$

and (3.42), we can expand (3.39):

$$\begin{aligned} f(t) &= \int_{-\infty}^{\infty} F(\nu)e^{+i2\pi\nu t} d\nu = \int_{-\infty}^{\infty} [a(\nu) + i b(\nu)] [\cos(2\pi\nu t) + i \sin(2\pi\nu t)] d\nu \\ &= \int_{-\infty}^{\infty} a(\nu) \cos(2\pi\nu t) d\nu - \int_{-\infty}^{\infty} b(\nu) \sin(2\pi\nu t) d\nu \\ &\quad + i \left[\int_{-\infty}^{\infty} a(\nu) \sin(2\pi\nu t) d\nu + \int_{-\infty}^{\infty} b(\nu) \cos(2\pi\nu t) d\nu \right] \end{aligned} \quad (3.43)$$

From this result it follows that if $f(t)$ is pure real then

$$f(t) = \int_{-\infty}^{\infty} a(\nu) \cos(2\pi\nu t) d\nu - \int_{-\infty}^{\infty} b(\nu) \sin(2\pi\nu t) d\nu \quad (3.44)$$

in close correspondence with (3.33) above for periodic functions.

If $f(t)$ is pure real, then from (3.44) and because of the frequency-domain symmetry defined by

$$\begin{aligned} a(\nu) \cos(2\pi\nu t) &= a(-\nu) \cos[2\pi(-\nu)t] \\ b(\nu) \sin(2\pi\nu t) &= -b(-\nu)(-\sin[2\pi(-\nu)t]) \end{aligned}$$

we can write, as we did for periodic functions,

$$f(t) = 2 \int_0^{\infty} a(\nu) \cos(2\pi\nu t) d\nu - 2 \int_0^{\infty} b(\nu) \sin(2\pi\nu t) d\nu \quad (3.45)$$

If $f(t)$ is pure imaginary, then from (3.43)

$$f(t) = 2 \int_0^{\infty} a(\nu) \sin(2\pi\nu t) d\nu + 2 \int_0^{\infty} b(\nu) \cos(2\pi\nu t) d\nu \quad (3.46)$$

Expanding Equation (3.40),

$$\begin{aligned} F(\nu) &= \int_{-\infty}^{\infty} f(t)e^{-i2\pi\nu t} dt \\ &= \int_{-\infty}^{\infty} [f_r(t) + i f_i(t)] [\cos(2\pi\nu t) - i \sin(2\pi\nu t)] dt \\ &= \int_{-\infty}^{\infty} f_r(t) \cos(2\pi\nu t) dt + \int_{-\infty}^{\infty} f_i(t) \sin(2\pi\nu t) dt \\ &\quad + i \left[\int_{-\infty}^{\infty} f_i(t) \cos(2\pi\nu t) dt - \int_{-\infty}^{\infty} f_r(t) \sin(2\pi\nu t) dt \right] \end{aligned}$$

or, if $f(t)$ is pure real,

$$F(\nu) = \int_{-\infty}^{\infty} f_r(t) \cos(2\pi\nu t) dt - \int_{-\infty}^{\infty} f_i(t) \sin(2\pi\nu t) dt \quad (3.47)$$

and so

$$a(\nu) = \int_{-\infty}^{\infty} f_r(t) \cos(2\pi\nu t) dt \quad (3.48)$$

$$b(\nu) = - \int_{-\infty}^{\infty} f_i(t) \sin(2\pi\nu t) dt \quad (3.49)$$

which, except for the $1/T$ factor are similar to Equations (3.30) and (3.31) for periodic functions.

Equation (3.48) is the *Fourier cosine transform* of $f_r(t)$; Equation (3.49) is the *Fourier sine transform*. The inverse Fourier transform is defined by Equation (3.44).

In Equation (3.43) if the units of $f(t)$ are in microns then the units of $a(\nu)$ and $b(\nu)$ must be in microns per unit of frequency interval $d\nu$. The functions $a(\nu)$ and $b(\nu)$ are referred to as *amplitude density functions*, and they are continuous functions of frequency ν .

The relationship between periodic and aperiodic Fourier coefficients is easily seen by comparing (3.30) with (3.48). We have

$$\begin{aligned} a_n T &= a(\nu) \\ b_n T &= b(\nu) \end{aligned}$$

That is, for $\nu = n/T$, the Fourier coefficients are directly proportional to one another with the constant of proportionality equal to the period T of the signal.

Computer implementation of the Fourier series and Fourier integral

Implementation of

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos(2\pi \frac{n}{T} t) - 2 \sum_{n=1}^{\infty} b_n \sin(2\pi \frac{n}{T} t)$$

where

$$\begin{aligned} a_n &= \frac{1}{T} \int_0^T f(t) \cos\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \\ b_n &= -\frac{1}{T} \int_0^T f(t) \sin\left(2\pi \frac{n}{T} t\right) dt \quad \text{for } n = 0, 1, 2, \dots \end{aligned}$$

is easily done if we use numerical integration to determine a_n and b_n . The same is true for

$$f(t) = \int_{-\infty}^{\infty} F(\nu) e^{+i2\pi\nu t} d\nu \quad (3.39)$$

$$F(\nu) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi\nu t} dt \quad (3.40)$$

and it is useful to examine the transition from periodic to nonperiodic functions using the same computer program. For nonperiodic functions, Equation (3.45) can be approximated by

$$\begin{aligned} f(t) &= 2 \int_0^{\infty} a(\nu) \cos(2\pi\nu t) d\nu - 2 \int_0^{\infty} b(\nu) \sin(2\pi\nu t) d\nu \\ &\approx 2 \sum_0^{\infty} a(\nu) \cos(2\pi\nu t) \Delta\nu - 2 \sum_0^{\infty} b(\nu) \sin(2\pi\nu t) \Delta\nu \\ &= 2\Delta\nu \sum_0^{\infty} a(\nu) \cos(2\pi\nu t) - 2\Delta\nu \sum_0^{\infty} b(\nu) \sin(2\pi\nu t) \end{aligned}$$

A program like Subroutine FT can be used with any increment $\Delta\nu$; however, fast Fourier transform algorithms impose a constraint on $\Delta\nu$ because the number of data points in the time and frequency domains needs to be the same. One can avoid this limitation by extending the data length in the time domain by adding zeros at the end of the data. We immediately see that our approximation to a nonperiodic function is itself periodic with period T , where

$$T = \frac{1}{\Delta\nu}$$

That is, large analysis windows in the time domain are associated with small frequency increments in the frequency domain.

Furthermore, if we were to use Fortran subroutine FT to implement

$$f(t) = 2\Delta\nu \sum_0^{\infty} a(\nu) \cos(2\pi\nu t) - 2\Delta\nu \sum_0^{\infty} b(\nu) \sin(2\pi\nu t)$$

then the above equation must be changed to

$$f(t) = \left[2\Delta\nu \sum_0^{\infty} a(\nu) \cos(2\pi\nu t) - 2\Delta\nu \sum_0^{\infty} b(\nu) \sin(2\pi\nu t) \right] - a(0)\Delta\nu$$

in order to be in agreement with

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos(2\pi \frac{n}{T} t) - 2 \sum_{n=1}^{\infty} b_n \sin(2\pi \frac{n}{T} t) \quad (3.33)$$

to acknowledge that our computer approximation to an aperiodic function is actually periodic with period $T = 1/d\nu$. As $d\nu \rightarrow 0$ then $T \rightarrow \infty$.

3.4.2 Fast Fourier Transform

Cooley and Tukey [42] proposed a fast algorithm for computing large complex Fourier transforms on a digital computer. With n a power of two, computing time for this algorithm is proportional to $n \log_2 n$, a major improvement over other methods where computing time is proportional to n^2 .

The “Fast Fourier Transform” (FFT version of the DFT—recall nomenclature on page 41) is considered to be one of the major breakthroughs in computational methods in general, and geophysics in particular, because it allows large data sets to be analyzed rapidly and accurately. It is the fastest known algorithm for calculating a discrete Fourier transform. The following is quoted from “The Hartley Transform” by Bracewell [25, p. 6]

When the FFT was brought into the limelight by Cooley and Tukey in 1965 it had an enthusiastic reception in the populous world of electrical signal analysis as the news spread via tutorial articles and special issues of journals. This ferment occasioned mild surprise in the world of numerical analysis, where related techniques were already known. Admirable sleuthing by M.T. Heideman, C.S. Burrus [30], and D.H. Johnson has now traced the origins of the method back to a paper of C.F. Gauss (1777-1855) written in 1805, where he says, “Experience will teach the user that this method will greatly lessen the tedium of mechanical calculation.”

A fascinating sidelight of the historical investigation is that Gauss’ fast method for evaluating the sum of a Fourier series antedates the work on which Fourier’s fame is based. We should hasten to add that Gauss’ paper was not published until much later (Collected Works, Vol. 3, Gottingen: Royal Society of Sciences, 1876), and we should remember that when Fourier introduced the idea of representing an arbitrary periodic function as a trigonometric series eminent mathematicians such as Lagrange resisted it.

A Fortran program NLOGN from Robinson [148] for computing forward and inverse fast Fourier transforms is included on the CD-ROM that accompanies this volume. This version of the FFT requires a data length of exactly 2^n . Other versions do not.

The discrete Fourier Transform can be given as a matrix multiplication [141], which requires n^2 complex multiplications where n represents the number of

sampled points. A discrete Fourier transform of length n can be given as the sum of two discrete Fourier transforms each with the length of $n/2$ (Danielson and Lanczos, see Press [141]). As stated in Press [141], in addition to Cooley and Tukey [42] and R.L. Garwin a number of researchers had independently discovered and implemented the Fast Fourier Transform. Following is a short discussion on the aspects of the FFT from a user point of view using a seismic trace as a discrete dataset. For a history of the FFT and further references the reader is referred to many excellent resources such as Brigham [27].

To emphasize the gain from the use of the Fast Fourier Transform (FFT) instead of the discrete Fourier series (DFS) a seismic dataset with 1000 shot gathers of 1024 traces per gather is considered. If the data length is 6 seconds recorded with a 2 ms sample rate, there will be about 3×10^9 data points. The Fourier transform of this data set requires about 9×10^{18} (n^2) complex multiplications. The Fast Fourier Transform reduces the number of complex multiplications to approximately 9×10^{10} ($n \log_2 n$). It would take about three years to complete the computations using FT but only 1 sec using FFT. Our thanks for fast computers and fast algorithms! The ratio between the numbers of complex computations in the Fourier Transform and Fast Fourier Transform is given by

$$\frac{n}{\log_2 n}$$

The main concept (Danielson and Lanczos, see Press [141]) behind the Fast Fourier Transform is that a discrete Fourier transform of a dataset $h[t]$ with length n can be represented by the sum of discrete Fourier transforms of the even-numbered and odd-numbered elements, each series with a length of $n/2$. The partial data sets with the even- and odd-numbered elements are given in the following form:

$$h^e[j] = h[2j] \quad j = 0, 1, \dots, n/2 - 1$$

and

$$h^o[j] = h[2j + 1] \quad j = 0, 1, \dots, n/2 - 1$$

Therefore, the Fourier transform of $h[j]$ can be given in two partial series in the following way:

$$H[k] = H^e[k] + H^o[k]$$

where $H^e[k]$ and $H^o[k]$ are the Fourier transforms of the partial data sets with even-numbered and odd-numbered points and k varies from 0 to $n/2$. These partial transforms are given by

$$H^e[k] = \sum_{j=0}^{n/2-1} h[2j] e^{i 2 \pi k 2j / n}$$

and

$$H^o[k] = \sum_{j=0}^{n/2-1} h[2j+1] e^{i2\pi k(2j+1)/n}$$

In a recursive fashion, one can continue subdividing data components all the way down to transforms of length 1 arrays. Different algorithms use different levels of subdivisions and can be based on decimation in time or in frequency depending on the data. Rearrangement for bit-reverse ordering is done before or after doing the transformation with $\log_2 n$ number of iterations. The speed gain can be obtained by eliminating some obvious multiplications for pre-known trigonometric computations with sine and cosine components, substituting complex computations with real arithmetic manipulations, and replacing floating point arithmetic with integer arithmetic using equivalent manipulations.

Most FFT algorithms transform input data of length n into the Fourier domain of length n . The Fourier domain components are ordered in the following way. For an odd number of data points there is no estimate at the Nyquist frequency and

$$\begin{aligned} H[k] & \quad \text{for positive } \nu \ (0 \leq \nu \leq \nu_N) \text{ is defined by } k = 1, 2, \dots, n/2 + 1 \\ H[k] & \quad \text{for negative } \nu \ (-\nu_N + d\nu \leq \nu \leq -d\nu) \text{ is defined by} \\ & \quad k = n/2 + 2, n/2 + 3, n/2 + 4, \dots, n - 1, n \end{aligned}$$

For input into an FFT program the signals in the time domain or the frequency domain must be positioned such that the value corresponding to $t = 0$ or $\nu = 0$ is the first element in the data vector.

For an even number n of points in a signal $x[1], x[2], x[3], \dots, x[n]$, $H[1]$ is the zero frequency (dc) component and

$$\begin{aligned} H[2] \dots H[n/2] & \quad \text{are the positive frequencies} \\ H[n] \dots H[n/2 + 2] & \quad \text{are the negative frequencies and} \\ H[n/2 + 1] & \quad \text{is the Nyquist frequency} \end{aligned}$$

For an odd number n of points $H[1]$ is the zero frequency component and

$$\begin{aligned} H[2] \dots H[(n+1)/2] & \quad \text{are the positive frequencies} \\ H[n] \dots H[(n+1)/2 + 1] & \quad \text{are the negative frequencies, and} \\ & \quad \text{there is no Nyquist frequency estimate} \end{aligned}$$

as shown in Figure 3.22c for an odd number (21) of points. It is obviously extremely important to know how an FFT program requires a data vector to be positioned for input and output. If the input is time-symmetric then the resulting cosine transform is pure real and symmetric. If the input is anti-symmetric then the resulting sine transform is pure imaginary. This will only be possible if the number of points in the time domain is odd. It is not possible to approximate a symmetric continuous function, symmetric about $t = 0$, by a discrete

function using an even number of points,; the resulting Fourier transform would be complex instead of pure real or pure imaginary.

Here is a numerical example (program `FFT0ddAndEven.nb`) using both Subroutine FT (see page 47) and *Mathematica*'s Fast Fourier Transform Fourier. First, an odd number of points. Assume a sampling interval of $\Delta t = 4$ ms so the folding frequency is $\nu_N = 1/(2\Delta t) = 125$ Hertz. Fortran Subroutine FT rewritten from Robinson [148] for *Mathematica* is

```
FT[x_, w_, k_] := (cosnw = 1; sinnw = 0;
sinw = Sin[w]; cosw = Cos[w]; s = 0; c = 0;
Do[c = c + cosnw*x[[i]]; s = s + sinnw*x[[i]];
t = cosw*cosnw - sinw*sinnw;
sinnw = cosw*sinnw + sinw*cosnw; cosw = t;
CosTransform[[k]] = c; SinTransform[[k]] = s,
{i, 1, Length[x]}]);
```

Let the input to FT be a real time-domain function $x = \{1, 2, 3, 4, 5\}$ with an odd number of points. The frequency increment $\Delta\nu$ between harmonics for a conventional Fourier spectral analysis is

$$\Delta\nu = \frac{1}{\text{period}} = \frac{1}{\text{Length}[x] \Delta t} = 50 \text{ Hertz}$$

The output from FT (for the zero and positive harmonic frequencies 0, 50, and 100 Hertz only, because of symmetry) is

```
Cosine transform from FT is {15, -2.5, -2.5}
Sine transform from FT is {0, -3.44095, -0.812299}
```

with the estimates corresponding to frequencies

$$\underbrace{-2.5 + 0.812299 i}_{\nu=-100}, \underbrace{-2.5 + 3.44095 i}_{\nu=-50}, \underbrace{15 + 0 i}_{\nu=0}, \underbrace{-2.5 - 3.44095 i}_{\nu=+50}, \underbrace{-2.5 - 0.812299 i}_{\nu=+100}$$

but no estimate at the folding frequency $\nu_N = 125$ Hertz (see Figure 3.22). The corresponding *Mathematica* usage is

```
x={1,2,3,4,5};
Fourier[x];
```

with identical output except for ordering:

$$\underbrace{15 + 0 i}_{\nu=0}, \underbrace{-2.5 - 3.44095 i}_{\nu=+50}, \underbrace{-2.5 - 0.812299 i}_{\nu=+100}, \underbrace{-2.5 + 0.812299 i}_{\nu=-100}, \underbrace{-2.5 + 3.44095 i}_{\nu=-50}$$

For an even number of points and the same sampling interval of $\Delta t = 4$ ms the folding frequency is still $\nu_N = 1/(2\Delta t) = 125$ Hertz but $\Delta\nu$ is different. Let the input x to FT be an even number of points $x = \{1, 2, 3, 4, 5, 6\}$. The frequency increment $\Delta\nu$ between harmonics is now

$$\Delta\nu = \frac{1}{\text{period}} = \frac{1}{\text{Length}[x] \Delta t} = 41.667 \text{ Hertz}$$

The output from FT (for zero and positive frequencies 0, 41.667, 83.333, and 125 only) is

Cosine transform from FT is {21, -3., -3., -3.}
 Sine transform from FT is {0, -5.19615, 1.73205, 0}

with the estimates corresponding to frequencies

$$\underbrace{21 + 0i}_{\nu=0}, \underbrace{-3 + 1.73205i}_{\nu=-83.333}, \underbrace{-3 + 5.19615i}_{\nu=-41.667}, \underbrace{21 + 0i}_{\nu=0}, \underbrace{-3 - 5.19615i}_{\nu=+41.6667}, \underbrace{-3 - 1.73205i}_{\nu=+83.333}, \underbrace{-3 + 0i}_{\nu=+125}$$

and now with a spectral estimate at the folding frequency $\nu_N = 125$ Hertz. The corresponding *Mathematica* program is

```
x={1,2,3,4,5,6};  
Fourier[x];
```

with identical output except for ordering:

$$\underbrace{21 + 0i}_{\nu=0}, \underbrace{-3 - 5.19615i}_{\nu=+41.6667}, \underbrace{-3 - 1.73205i}_{\nu=+83.333}, \underbrace{-3 + 0i}_{\nu=+125}, \underbrace{-3 + 1.73205i}_{\nu=-83.333}, \underbrace{-3 + 5.19615i}_{\nu=-41.667}$$

and this time with an estimate at the folding frequency $\nu = 125$ Hertz.

3.5 Applications of Fourier Transforms

A seismic wavelet undergoes many changes in shape and polarity as it propagates in the Earth. Two of these changes can be caused by simple geologic conditions: thin beds and linear changes in velocity with depth. These conditions can be nicely described mathematically by examining time differentiation and time integration of the Fourier transform, respectively, and are described in additional detail in Chapter 5. Another effect that changes wavelet shape is intrinsic damping, discussed on page 518.

The effect on the phase spectrum of a seismic wavelet can be described by the "time-shifting theorem." Simply stated, if a wavelet propagates without a change in its shape (without distortion), then the quantity

$$\sqrt{a_n^2 + b_n^2}$$

remains unchanged, but an amount equal to $-\omega t$ is added to its phase spectrum. This can be shown as follows.

3.5.1 Time-shifting theorem

A function $f(t)$ shifted toward positive time by an amount τ is denoted by

$$f(t - \tau)$$

The Fourier transform of this time-shifted function is, by definition

$$G(\nu) = \int_{-\infty}^{\infty} f(t - \tau) e^{-i2\pi\nu t} dt$$

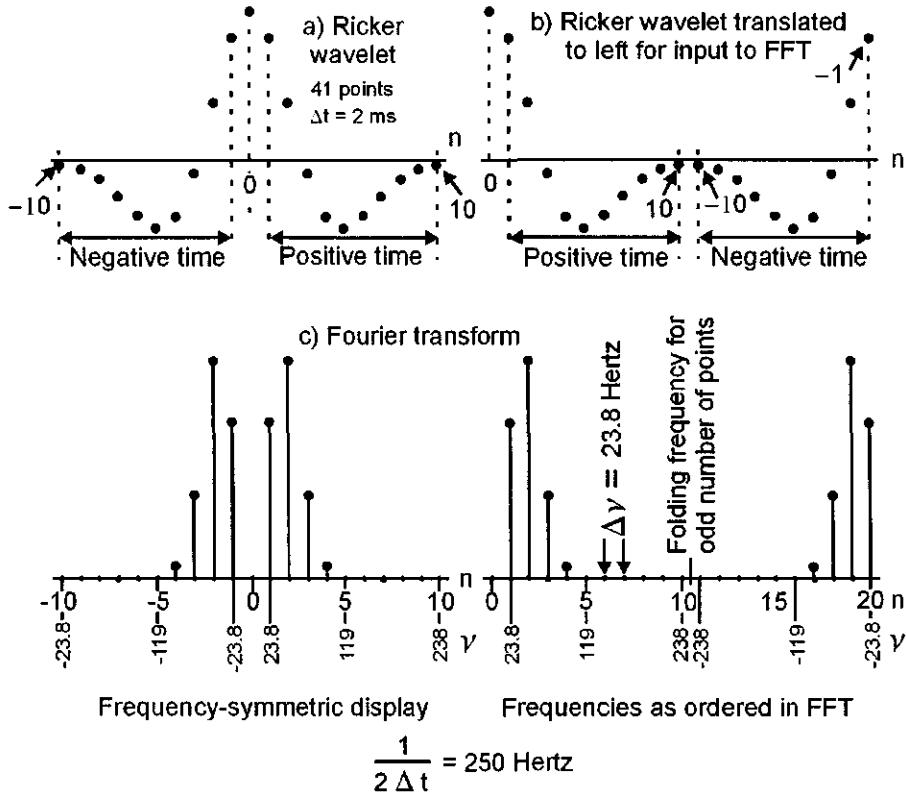


Figure 3.22: (a) A Ricker wavelet with a central frequency of 40 Hertz sampled at 2 ms. (b) Ricker wavelet rotated left with *Mathematica*'s `RotateLeft[]` module for proper input to an FFT program. (c) Spectral elements of transform as ordered in an FFT program. (d) Usual symmetric display of transform after FFT. The number of points is odd so there is no estimate at the Nyquist frequency. `RickerWaveletForFFT.nb` \Rightarrow `RickerFFTt.wmf`

Let $x = t - \tau$. Then $dt = dx$, because τ is a constant, and we have

$$\begin{aligned}
 G(\nu) &= \int_{-\infty}^{\infty} f(t - \tau) e^{-i2\pi\nu t} dt \\
 &= \int_{-\infty}^{\infty} f(x) e^{-i2\pi\nu(x+\tau)} dx \\
 &= \int_{-\infty}^{\infty} f(x) e^{-i2\pi\nu\tau} e^{-i2\pi\nu x} dx \\
 &= e^{-i2\pi\nu\tau} \int_{-\infty}^{\infty} f(x) e^{-i2\pi\nu x} dx
 \end{aligned}$$

$$= e^{-i2\pi\nu\tau} F(\nu)$$

$F(\nu)$ is the (complex) Fourier transform of the unshifted function. Rewriting,

$$\begin{aligned} G(\nu) &= e^{-i2\pi\nu\tau} F(\nu) \\ &= e^{-i2\pi\nu\tau} |F(\nu)| e^{i\phi(\nu)} \\ &= e^{i\phi(\nu)-i2\pi\nu\tau} |F(\nu)| \end{aligned}$$

where $|F(\nu)|$ denotes the modulus of $F(\nu) = \sqrt{a(\nu)^2 + b(\nu)^2}$ and $\phi(\nu)$ or $\phi(\omega)$ is the phase spectrum of $F(\nu)$. As usual,

$$\phi(\nu) = \tan^{-1} \left[\frac{b(\nu)}{a(\nu)} \right]$$

That is, the Fourier transform of a time-shifted function always has the same modulus $|F(\nu)|$ but the phase spectrum is continually changing as the amount $-\omega\tau = -2\pi\nu\tau$ is added to its phase spectrum after a traveltime of τ time units. In practice, we will see in Section 8.5.2 about Q on page 518 that *intrinsic damping* and the attenuation of higher frequencies by internal friction not only results in a continuous change in wavelet shape, but also introduces *body wave dispersion*, where different frequencies travel at slightly different body wave velocities.

The effect on the Fourier spectrum of multiplication by $e^{-i\omega\tau}$ is shown in Figure 3.23. Equation (3.50) has important consequences. The Fourier transform of a propagating seismic pulse that has traveled for a time τ has the same modulus but a different phase spectrum. Said another way, if the shape of the propagating seismic wavelet is time-invariant, then its Fourier transform must be described by Equation (3.50).

Implications

This conclusion states that if a source wavelet were to propagate in the Earth *without distortion*, then its Fourier spectrum (the word "spectrum" by itself refers to both amplitude and phase) would have to change only by addition of the linear term $-\omega\tau$ to its phase spectrum. The amplitude spectrum would remain unchanged. In reality, this is not possible for propagating seismic wavelets except in rocks that have an extremely high value of Q with constant velocity. Intrinsic damping and the absorption of higher frequencies changes the shape of the wavelet.

The time-shifting theorem is important. *If the phase angle spectrum does not change in a linear fashion as a real seismic wavelet propagates in time, then an inverse Fourier transform at a point in time (or at a point in space) will reveal that the wavelet shape is different for different times.*

The amplitude spectrum of the seismic wavelet is the square root of the sum of the squares of the real and imaginary parts of the Fourier transform:

$$A_n = \sqrt{a_n^2 + b_n^2}$$

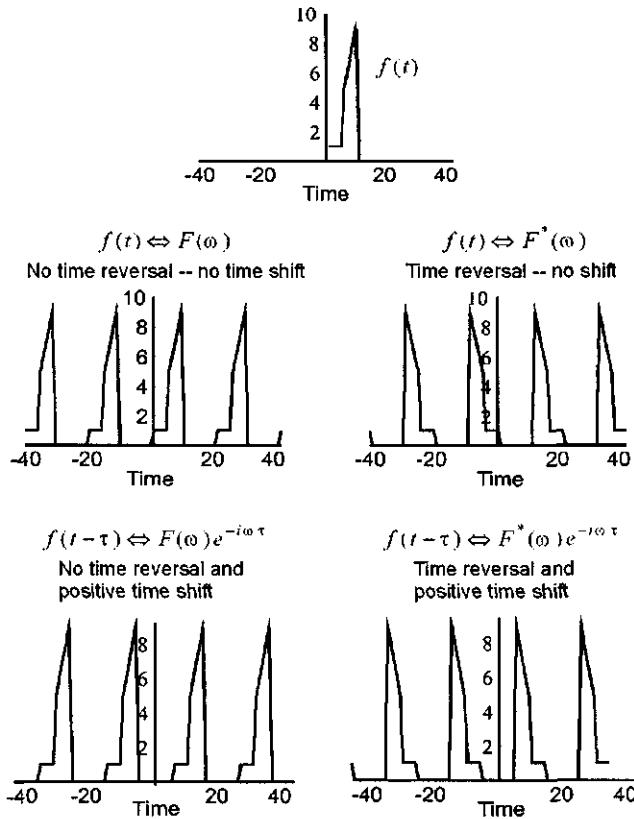


Figure 3.23: Top: original, unshifted function $f(t)$. Middle left: Notation for unshifted function. Middle right: notation for unshifted but time-reversed function $f(t)$. Bottom left: notation for Fourier transform multiplied by $\exp(-i\omega\tau)$ where τ is 6 time units resulting in positive-time shift. Bottom right: positive-time shifted and time-reversed $f(t)$. (FourierShiftedRight.mb → FourierShiftedRight.cdr → EffectOfTimeShift.wmf)

As the wavelet is shifted in time, the time-shifting theorem tells us that a_n and b_n both change, and so does

$$\tan^{-1} \frac{b_n}{a_n}$$

i.e., the phase angle $\phi(\nu)$ also changes, but A_n does not. If something is shifted in time (or space), and the time-shifting theorem is *not* obeyed, then some other physical process must be involved, such as intrinsic damping in the case of a real seismic wavelet. See Chapter 8.5.2.

Phase spectrum versus phase lag spectrum

As a wavelet propagates toward increasing time τ its phase spectrum $\phi(\nu)$ is continuously changing by virtue of the term $e^{-i\omega\tau}$ that multiplies the wavelet Fourier transform and therefore adds $-\omega\tau$ to the wavelet's phase spectrum $\phi(\nu)$, whatever that might be. The quantity $-\omega\tau$ is a phase *lag* because multiplication of the wavelet spectrum by $e^{-i\omega\tau}$ delays the time-domain pulse by an amount τ . That is, for the spectrum of an impulse (unity)

$$\delta(t - \tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (1) e^{-i\omega\tau} e^{i\omega t} d\omega$$

and the Dirac delta function is seen to be delayed by an amount τ . Delay of a signal in the time domain is equivalent to a phase lag in the frequency domain. If the phase spectrum is $\omega\tau$ then $-\omega\tau$ is the phase lag spectrum. So in general given a phase spectrum $\theta(\omega)$ then $-\theta(\omega)$ is the phase lag spectrum.

3.5.2 Time differentiation of the Fourier transform

Given the inverse Fourier transform:

$$f(t) = \int_{-\infty}^{\infty} F(\nu) e^{i2\pi\nu t} d\nu = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

Differentiate both sides with respect to t :

$$\frac{df(t)}{dt} = \frac{1}{2\pi} \frac{d}{dt} \left[\int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \right] = \frac{1}{2\pi} \int_{-\infty}^{\infty} i\omega F(\omega) e^{i\omega t} d\omega$$

Result:

$$\frac{df(t)}{dt} \iff (i\omega)F(\omega) \quad (3.50)$$

Differentiate again with respect to t :

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

Result:

$$\frac{d^2 f(t)}{dt^2} \iff (i\omega)^2 F(\omega)$$

In general:

$$\frac{d^n f(t)}{dt^n} \iff (i\omega)^n F(\omega) \quad (3.51)$$

Note that differentiation amplifies the higher frequencies with respect to the lower because the original spectrum $F(\omega)$ is multiplied by $i\omega$. As ω increases, whatever the original value of $F(\omega)$, it will be greater after multiplication by $i\omega$. Thus, differentiation of a noisy seismic trace results in an even nosier seismic trace, especially if the noise is toward the higher end of the frequency spectrum.

3.5.3 Time integration of the Fourier transform

Integrate both sides of

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

with respect to time:

$$f^{-1}(t) = \int_{-\infty}^t f(\tau) d\tau = \int_{-\infty}^t \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \right] d\tau = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{i\omega} F(\omega) e^{i\omega t} d\omega$$

Result:

$$f^{-1}(t) = \int_{-\infty}^t f(\tau) d\tau \iff \frac{1}{i\omega} F(\omega) \quad (3.52)$$

Further integration can be carried out with respect to time:

$$\int_{-\infty}^t \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{i\omega} F(\omega) e^{i\omega t} d\omega \right] dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{(i\omega)^2} F(\omega) e^{i\omega t} d\omega$$

In general

$$f^{-n}(t) dt \iff \frac{1}{(i\omega)^n} F(\omega) \quad (3.53)$$

where f^{-n} denotes the n th integration with respect to time t .

Time differentiation and integration are examples of high-pass and low-pass filters, respectively. The geology itself by means of certain velocity functions can differentiate or integrate a seismic wavelet, as shown in Chapter 5. Differentiation emphasizes the high frequencies over the low, and integration the low over the high. In general, given any kind of a function f , whether it be a function of time or a function of space, the locations of anomalies in time or space are much more obvious in the derivative of the function than in the function itself.

As an example of the effect of Equations (3.50) and (3.52) on data we show in Figure 3.24 a reflectivity function convolved with a Klauder wavelet (top), the same reflectivity function convolved with the differentiated Klauder wavelet (middle), and the reflectivity function convolved with the integrated Klauder wavelet (bottom). The integrated trace is enriched in low frequencies because of division by $i\omega$ (with an expected loss in resolution). Of course, the same result is obtained by differentiating or integrating the entire trace after convolution, as shown in Figure 3.26 for integration. Spectra of the Klauder wavelet, the differentiated Klauder wavelet, and the integrated Klauder wavelet are shown in Figure 3.27.

Differentiation of a seismic trace is straightforward and easy to visualize. Integration might not be as obvious so we provide Figure 3.25. In order to understand what division by $i\omega$ means, suppose a seismic trace is composed of a time-invariant source wavelet that is convolved with a reflectivity function, a

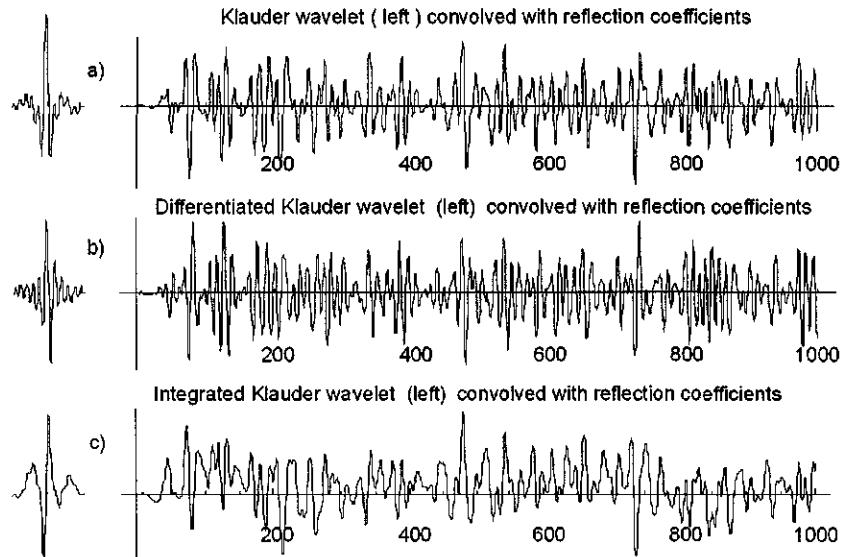


Figure 3.24: (a) Convolution of Klauder source wavelet (shown to the left of the trace) with reflection coefficients. (b) Differentiated Klauder wavelet (shown to the left of the trace) convolved with the same reflection coefficients. (c) Integrated Klauder wavelet (shown to the left of the trace) convolved with the same reflection coefficients. (SyntheticDiffAndInteg.nb \Rightarrow SyntheticDiffAndInteg.cdr \Rightarrow ComparisonDiffAndInteg.wmf)

generally-accepted good first approximation. If this trace is convolved with a unit step function generated by

```
UnitStepFunction = Table[1, Length[SeismicTrace]]
```

then the result is the same as you would get if an integrated source wavelet left the source and was reflected from those same reflection coefficients. This example is shown in Figure 3.24 using a Klauder wavelet as the source wavelet.

Figure 3.24c was obtained by first integrating the Klauder wavelet and then convolving the integrated source wavelet with the reflection coefficients. The integrated trace shown in Figure 3.26c is identical but was obtained by convolution of the trace with a unit step function. So the question naturally arises as to what sense it makes to convolve a step function with a trace in order to get a result that is equivalent to convolution of the reflectivity function with an integrated version of the source wavelet (a shape which, by the way resembles an offshore marine source). That is, does it make any physical sense to get an integrated source wavelet way down on the trace when this result is arrived at by summing all the points in the trace up to that point? This is what happens when you convolve the trace with a step function. What contribution does

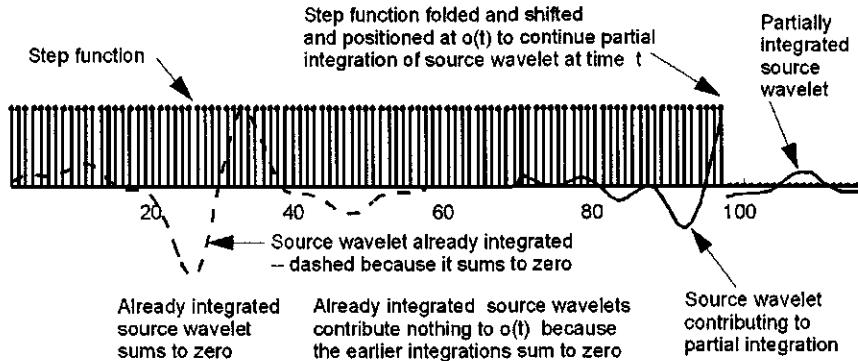


Figure 3.25: A snapshot in time while the source wavelet is undergoing partial integration. The seismic trace $o(t)$ in this case is simply one integrated source wavelet (not shown) from the top of the interval and one of reversed polarity from the bottom. For the situation shown, the integrated version from the bottom of the zone is not yet complete and a developing partial integration of reversed polarity is shown. In reality the partial integration would be reflected back up (to the left); the figure is intended to show the process of wavelet integration, not the complete seismic trace). Integration $f^{-1}(t)$ of wavelet accomplished by convolution of $f(t)$ with a unit step function $u(t)$ (the reflectivity function); i.e., $f^{-1}(t) = \int_{-\infty}^{\infty} f(\tau)u(t-\tau)d\tau$. Folded and shifted step function $u(t)$ moves across trace as convolution progresses. (A real, physical wavelet $f(t)$ would have a mean of zero.) Already integrated source wavelets $f^{-1}(t)$ sum to zero and contribute nothing to the convolution $f^{-1}(t)$ at time t . See Figure 5.14 on page 228 for a geological example of this integration process. (IntegrationAndCancellation.nb,cdr)

the early part of the trace make to the later part of the trace? The answer is provided on Figure 3.25 where it is seen that earlier integrated wavelets simply sum to zero, so they make no contributions at all to the later parts of the trace where, by partial integration, a new integrated waveform is being formed at $o(t)$. Compare this figure with Figure 5.14 and the accompanying discussion where it is shown that under certain conditions the earth actually performs this kind of integration.

A real seismic trace ("Trace 18") is shown in Figure 3.28 before and after differentiation (first derivative) by Equation (3.51). Notice the increase in high frequency content caused by the differentiation.

The same seismic trace integrated is shown in Figure 3.29 before and after integration. Notice the increase in low frequency content in Figure 3.29 after integration. The real data of length T after integration show a marked sinusoidal trend with a frequency near $1/T$. This trend is clearly the result of amplification of mathematical noise because the frequencies (≈ 1 Hertz) could not have been present in the recorded data. This interpretation is confirmed after spectral

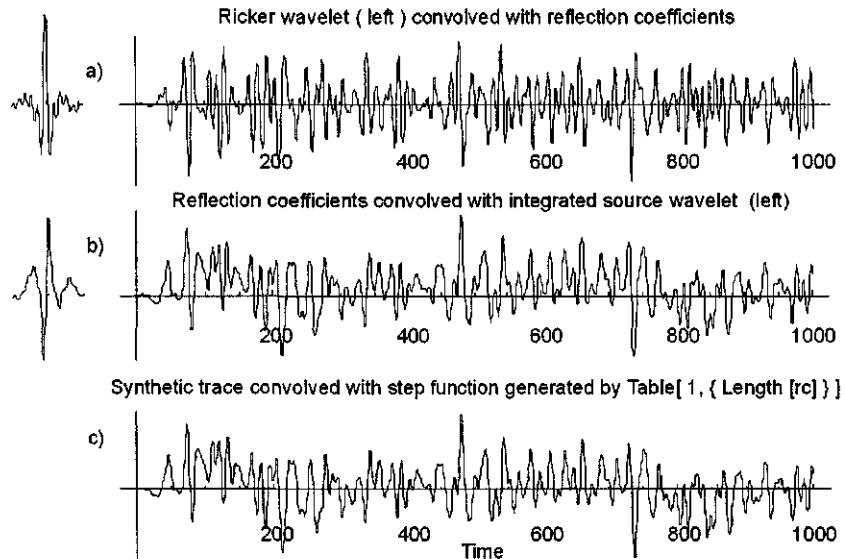


Figure 3.26: Top: Convolution of a Klauder wavelet with 1000 reflection coefficients (the random numbers) Source wavelet is to the left of the trace. Middle: Convolution of the same 1000 reflection coefficients with the integrated source wavelet shown to the left of the trace. Bottom: Convolution of the synthetic seismic trace (top) with a unit-step function. Note that (b) and (c) are identical. (syntheticdiffandinteg.nb \Rightarrow SyntheticDiffAndInteg.cdr \Rightarrow SyntheticIntegTwoWays.wmf)

analysis before and after integration as shown in Figure 3.30.

The integrated trace or wavelet is the output of a series of partial sums. The computations can be performed more efficiently by writing a module such as `foldcc` below that uses to advantage some of the built-in *Mathematica* functions.

```
foldcc[a_,b_]:=(la=Length[a];lb=Length[b];
lc=la+lb-1;
bb=Reverse[b];aa=PadRight[PadLeft[a,la+lb-1],lc+lb-1];
foldout=Table[Apply[Plus,Take[aa,{i,lb+i-1}]*bb],{i,1,lc}]);
```

A non-seismic but well-known example of the emphasis of anomalies by differentiation is shown in Figure 3.31. The Earth's geothermal gradient is the derivative of the temperature log versus depth.

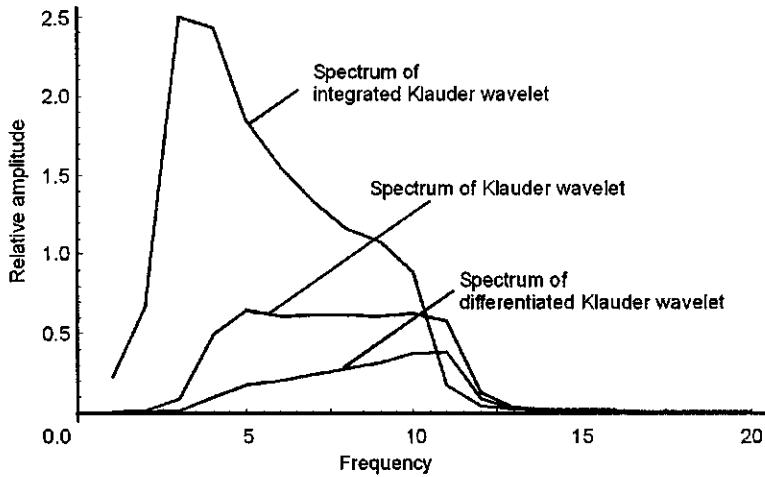


Figure 3.27: Spectra of differentiated and integrated Klauder wavelet plotted to same relative scale. (SyntheticDiffAndInteg2.nb \Rightarrow SyntheticDiffAndInteg.cdr \Rightarrow SpectraOfKlauderWavelet.wmf)

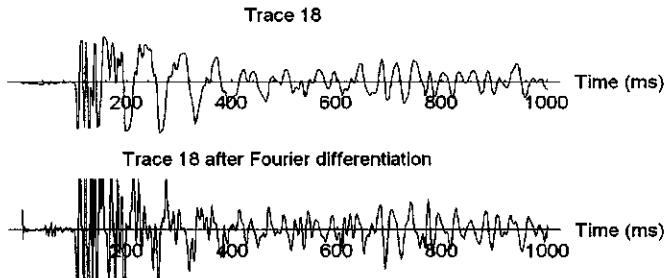


Figure 3.28: Trace 18 (top) and the derivative of trace 18 as obtained by Equation (3.51) (bottom). The increase in high frequency content is obvious. (deriv18)

3.5.4 Introduction to the unit-impulse function

A rectangular pulse of height E_m symmetric with respect to time is shown in Figure 3.32. The Fourier transform of this pulse is given by:

$$\begin{aligned} F(\nu) &= \int_{-\infty}^{\infty} f(t)e^{-i2\pi\nu t} dt \\ &= \int_{-T/2}^{T/2} E_m e^{-i2\pi\nu t} dt = E_m \int_{-T/2}^{T/2} e^{-i2\pi\nu t} dt \end{aligned}$$

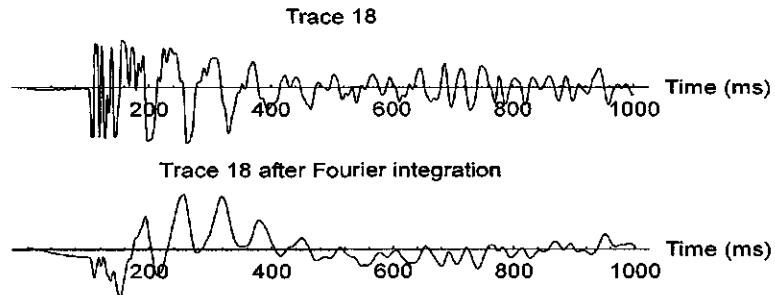


Figure 3.29: Trace 18 (top) and the integral of trace 18 (bottom). The decrease in high frequency content is obvious. (IntegrateTrace18.nb \Rightarrow int18.wmf)

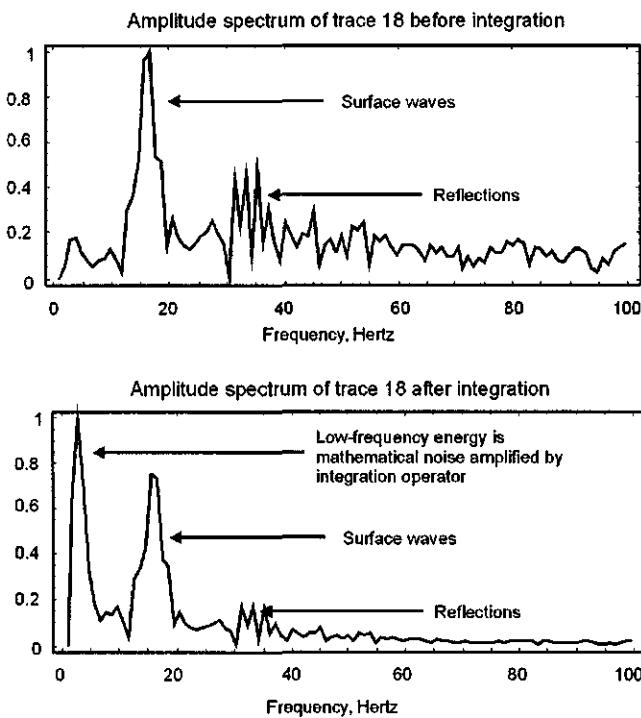


Figure 3.30: Top: Amplitude spectrum of trace 18 before integration. Bottom: Amplitude spectrum of trace 18 after integration. IntegrateTimeTestFinal1.nb \Rightarrow IntegrateTrace18.cdr \Rightarrow SpectraTrace18.wmf

$$= E_m \int_{-T/2}^{T/2} \cos(2\pi\nu t) dt - i E_m \int_{-T/2}^{T/2} \sin(2\pi\nu t) dt$$

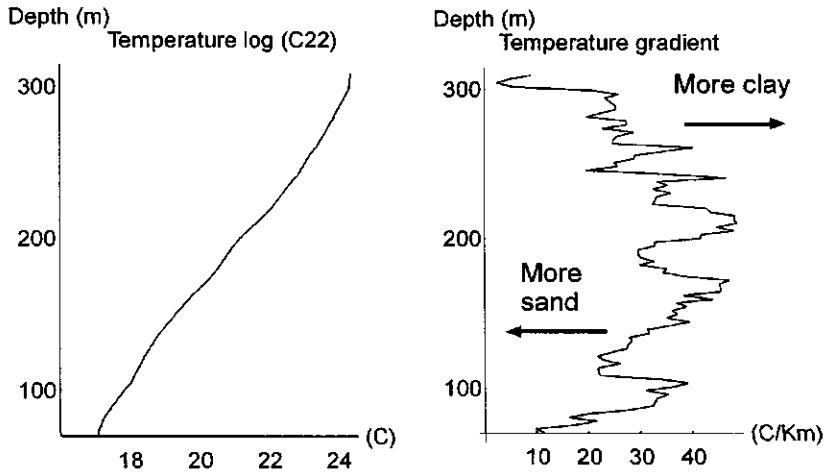


Figure 3.31: Temperature versus depth (left) and the derivative of this (right). The increase in the high frequency content of the geothermal gradient log is not noise. The temperature gradient (geothermal gradient) is directly proportional to subtle changes in lithology because the heat flow in the sediments is constant. A precision ($\pm 0.001 \text{ C}^\circ$ to $\pm 0.0001 \text{ C}^\circ$) temperature log is able to resolve these small differences. (logs.cdr \Rightarrow logs.wmf)

$$= \frac{E_m}{2\pi\nu} \left[\sin(2\pi\nu t) \right]_{-T/2}^{T/2} - 0$$

because sines are odd functions. Then, evaluating,

$$F(\nu) = \frac{E_m}{2\pi\nu} \left[\sin\left(2\pi\nu \frac{T}{2}\right) - \sin\left(2\pi\nu \frac{-T}{2}\right) \right] = \frac{E_m \sin(2\pi\nu \frac{T}{2})}{\pi\nu}$$

and so

$$F(\omega) = \frac{E_m \sin(\omega \frac{T}{2})}{\omega/2} = E_m T \left[\frac{\sin(\omega \frac{T}{2})}{\omega T/2} \right]$$

$E_m T$ is the area of the rectangular pulse. The spectrum of the rectangular pulse is

$$F(\omega) = E_m T \left[\frac{\sin(\omega \frac{T}{2})}{\omega \frac{T}{2}} \right] \quad (3.54)$$

or

$$F\left(\frac{n}{T}\right) = E_m T \left[\frac{\sin\left(2\pi \frac{n}{T} \frac{T}{2}\right)}{2\pi \frac{n}{T} \frac{T}{2}} \right] = E_m T \left[\frac{\sin(n\pi)}{n\pi} \right]$$

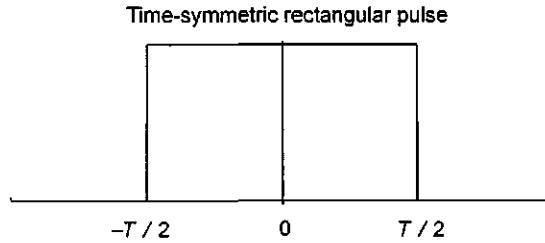


Figure 3.32: Rectangular function centered about $t = 0$ of width T and height E_m . (rectpuls)

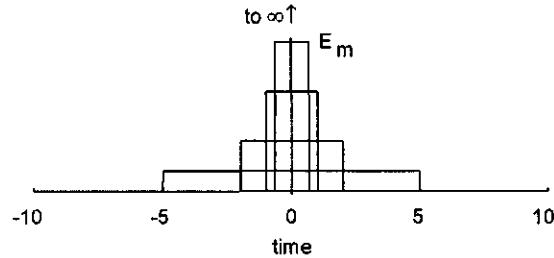


Figure 3.33: Area of rectangular pulse is kept constant as height of pulse increases. In the limit as $E_m \rightarrow \infty$ we arrive at the definition of the Dirac delta function $\delta(t)$, a singularity function whose amplitude is infinite and whose duration in time is zero. (ma rectpuls \rightarrow dirac2 \rightarrow cd \rightarrow dirac2)

Now keep the area $E_m T$ of the rectangular pulse constant as the width T of the pulse gets smaller and smaller, as shown in Figure 3.33.

This means that the height of the pulse must get larger and larger in order to keep the area equal to unity. As the width of the pulse becomes infinitesimally small, the height of the pulse approaches infinity, and in the limit, as $T \rightarrow 0$ we have

$$F(\omega) = \lim_{T \rightarrow 0} E_m T \left[\frac{\sin(\omega T/2)}{\omega T/2} \right]$$

because

$$\lim_{\theta \rightarrow 0} \frac{\sin \theta}{\theta} = 1$$

We can define the area $E_m T$ of the rectangular pulse to be anything. In particular we can define it to be unity so that we get a single impulse at the origin whose integral (area) is unity and whose height is infinite:

$$\begin{aligned} \int_{-\infty}^{\infty} \delta(t) dt &= 1 & t = 0 \\ &= 0 & t \neq 0 \end{aligned}$$

and

$$\begin{aligned}\delta(t) &= \infty & t = 0 \\ &= 0 & t \neq 0\end{aligned}$$

Such an impulse is called a Dirac delta function $\delta(t)$. Thus,

$$F(\omega) = \lim_{T \rightarrow 0} E_m T \left[\frac{\sin(\omega T/2)}{\omega T/2} \right] = E_m T = 1 \quad (3.55)$$

There is no imaginary part to $F(\omega)$ because $F(\omega) = 1$. Therefore the Dirac delta function $\delta(t)$ must be an even function of time with a phase angle spectrum equal to zero for all frequencies ω .

We have therefore deduced that $\delta(t)$ and 1 constitute a Fourier transform pair; i.e.,

$$\delta(t) \iff 1 \quad (3.56)$$

The observation that the spectrum of $\delta(t)$ is unity, that is

$$F(\omega) = 1 \quad (3.57)$$

is a statement that the unit impulse function contains *all* frequencies from $-\infty$ to $+\infty$, all with *equal* amplitudes and with the same phase angle (equal to zero). In other words,

$$\begin{aligned}\delta(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} [1] e^{i\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} [1 d\omega] e^{i\omega t}\end{aligned}$$

The amplitude of each sinusoid defined by the complex exponential $e^{-i\omega t}$ is $[1 d\omega]$, and there is an infinite number of them. Indeed, they are all cosine waves (even functions of time) because the antisymmetric sine waves cancel out. A few of the cosine waves before summing are shown in Figure 3.34 where they are centered about $t = 0$, as required by Equation (3.57). About 50 waves summed together are shown in Figure 3.35. Although it would require summing all frequencies ν to achieve $\delta(t)$, the use of just a few

$$\sum_{\nu=0}^{\nu=49} \cos(2\pi\nu t)$$

is clearly achieving the desired result. Spectra defined by Equation (3.54) are shown in Figure 3.36. Note that as the width of the rectangular function becomes narrower, the amplitudes of the high frequency components increase relative to the lower frequencies until, finally, the spectrum of a Dirac delta function is obtained.

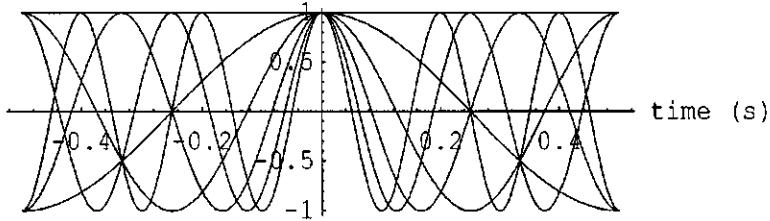


Figure 3.34: A few cosine waves centered about $t = 0$ before summing. They are exactly in phase only at the origin $t = 0$. (cosdirac)

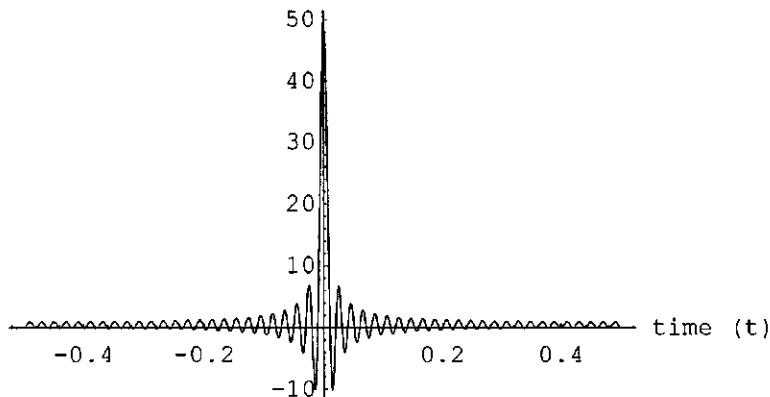


Figure 3.35: The Dirac delta function is an even function of time, and is obtained by summing cosine waves centered about $t = 0$. All frequencies are present in $\delta(t)$, and all have the same amplitude. A few of these cosine waves after summing are shown above. (cosdirac → fewdirac)

3.5.5 The sinc function

In the course of arriving at the definition of the Dirac delta function, we derived an important Fourier transform pair:

$$f(t) \iff \frac{\sin(\omega T/2)}{\omega T/2}$$

where $f(t)$ is the rectangular pulse of width T and amplitude unity. It is useful to display this transform pair graphically as in the following illustration, and to examine the relationship between the width T of the pulse and the locations of the null points in the frequency domain, as illustrated by Figure 3.37.

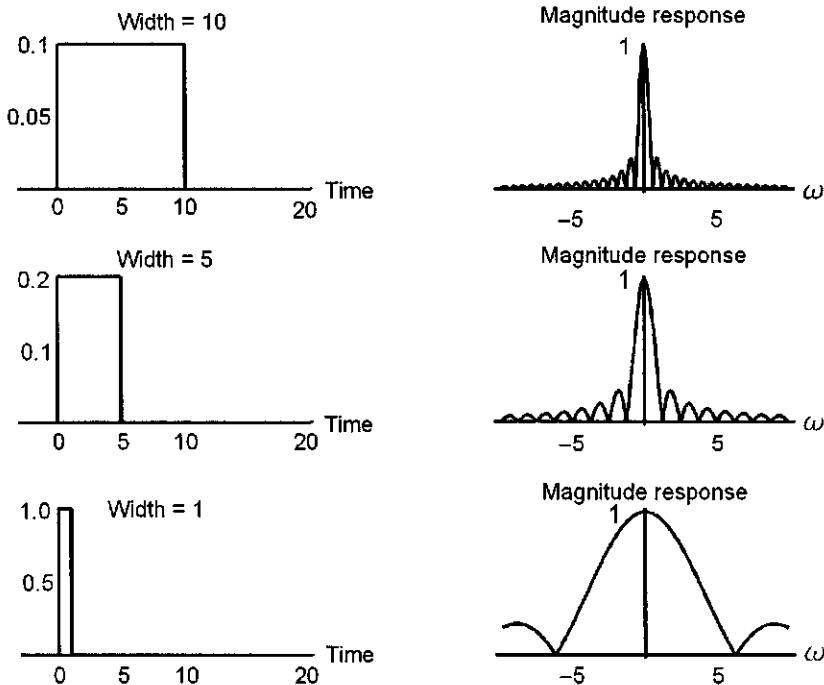


Figure 3.36: Graphic display of derivation of frequency response of Dirac delta function. Time domain on the left, frequency domain on the right. From top to bottom, width of rectangular pulse is decreasing (left column), resulting in spectra that approach a value of unity for all frequencies (right column). Area of rectangular pulse in time domain is kept constant and equal to unity. The rectangular pulse must therefore get taller as it gets narrower. (specpuls)

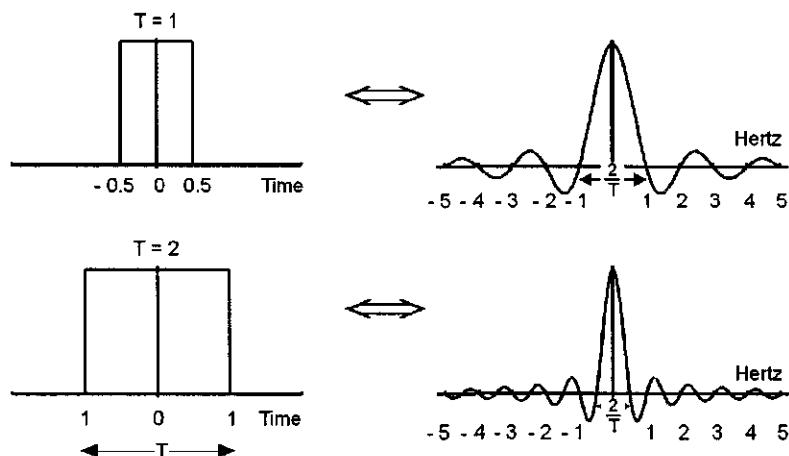


Figure 3.37: Relation between period T of rectangular pulse and the spacing of the null points (breadth) in the Fourier transform of the pulse. (ma, cd → square)

3.5.6 Definition of a linear System

Let $f(t)$ be the input to the system and $g(t)$ the output, as shown in Figure 3.38. If the system is linear, it has the following characteristics:

1. If $f(t)$ is delayed by time $t = \tau$, the output $g(t)$ will be delayed by the same amount. That is, if $f(t - \tau)$ is the input, then the output is $g(t - \tau)$.
2. If the input $f(t)$ is scaled by a constant C , then the output is scaled by the amount C ; i.e., if the input is $Cf(t)$, then the output is $Cg(t)$.
3. If the input is $f_1(t) + f_2(t)$, then the output is $g_1(t) + g_2(t)$.

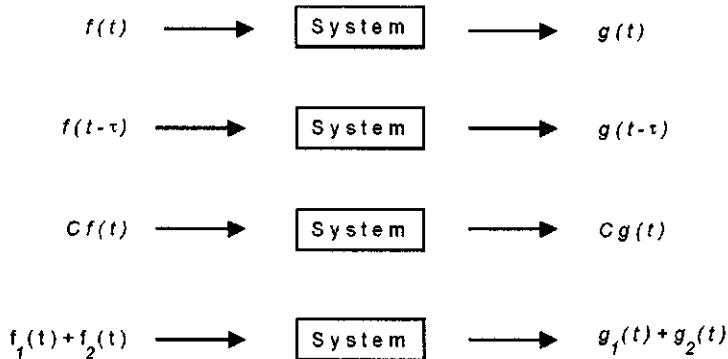


Figure 3.38: Definition of a linear system. Top: The input to the system is $f(t)$ and the output is $g(t)$. If the input $f(t)$ is delayed by τ time units, then the output $g(t)$ is delayed by the same amount. If the input $f(t)$ is scaled by a constant C , then the output is $Cg(t)$. Bottom: If the input is $f_1(t) + f_2(t)$, then the output is $g_1(t) + g_2(t)$. (cd → fold1)

3.5.7 Impulse Response

If the input to the system is a single impulse like the Dirac delta function $\delta(t)$, then the output $h(t)$ is called the *impulse response* of the system (Figure 3.39). Recalling that the Dirac delta function contains all frequencies from $-\infty$ to ∞ with the same amplitude and phase (zero) because

$$\delta(t) \iff 1$$

then it follows that the Fourier transform of $h(t)$ is exactly the *frequency response of the system*.

If $\delta(t)$ is the input to a system whose frequency characteristics are unknown, then the output from the system is a signal whose Fourier transform reveals the relative change of the amplitude of any sine or cosine wave input, as well as the change in the phase angle of the sinusoid.

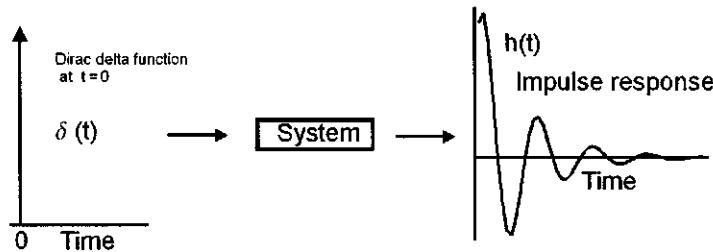


Figure 3.39: The response of a linear system to an impulse. The impulse is $\delta(t)$ and the time response is $h(t)$. The Fourier transform of $h(t)$ contains all the information about how the system responds to a sinusoid of any amplitude and frequency. (cd \rightarrow inout)

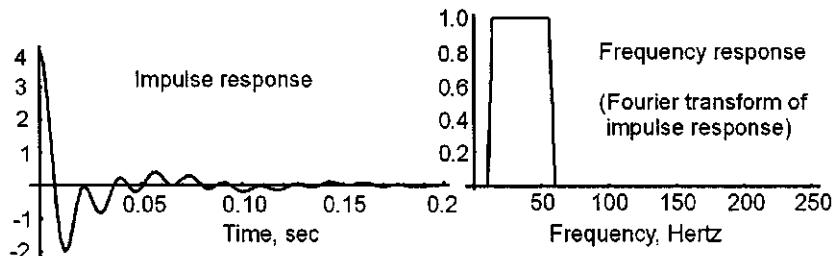


Figure 3.40: Impulse response of a filter (left). Only the early, non-zero, portion is shown. The modulus of the Fourier transform of this filter is shown in the righthand figure. The filter is a trapezoidal bandpass system with a bandwidth of 14-56 Hertz. (ma \rightarrow impresa, impresb \rightarrow cd \rightarrow impres)

With reference to Figure 3.38, we now examine the response of any *linear system* to an arbitrary input. For a Dirac delta function, the input to the system has an area of unity. That is,

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

where "1" is the area of the input.

Let the response of a system to an impulse $\delta(t)$ be $h(t)$, as shown in Figure 3.39. Commonly the term "impulse response" refers to the time response of some kind of filter when the input to the filter is impulsive; i.e., the input acts over a very short length of time when compared to the output, as shown in Figure 3.39. Eventually, the response vanishes to zero. The "system" in which we are most interested is the Earth. In this case, the impulsive input might be an energy source like dynamite, and the "impulse response" of the system is the

seismic trace, or seismogram.

The spectrum of a 14-56 Hertz trapezoidal bandpass filter is shown in Figure 3.40 (right). The inverse Fourier transform is also shown (left). This is the impulse response of the filter. The spectrum is similar to the bandwidth (14-56 Hertz) used by Virginia Tech for acquisition and processing of deep crustal reflection seismic data in the eastern United States.

3.5.8 The time-convolution theorem

Given any two functions $f_1(t)$ and $f_2(t)$, if we define a function $f(t)$ that represents the similar parts of both functions, then

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$$

where $f(t)$ is defined as the *convolution* of $f_1(t)$ and $f_2(t)$. Notation:

$$\begin{aligned} f_1(t) &\iff F_1(\omega) \\ f_2(t) &\iff F_2(\omega) \\ f(t) &\iff F(\omega) \end{aligned}$$

The *time convolution theorem*:

$$\left[f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau \right] \iff [F(\omega) = F_1(\omega) F_2(\omega)]$$

The theorem states that the Fourier transform $F(\omega)$ of the convolution $f(t)$ of two functions $f_1(t)$ and $f_2(t)$ is equal to the product of the Fourier transforms $F_1(\omega)$ and $F_2(\omega)$, i.e.,

$$F(\omega) = F_1(\omega) F_2(\omega)$$

To show this, we proceed as follows,

$$F(\omega) \iff f(t)$$

by definition, i.e.,

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau \right] e^{-i\omega t} dt \\ &= \int_{-\infty}^{\infty} e^{-i\omega t} \left[\int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau \right] dt \end{aligned}$$

As this equation stands, integration with respect to τ is carried out first to get a particular value of $f(t)$. Then integration with respect to t is carried out to

get the complex function $F(\omega)$, i.e., the value of the spectrum at a frequency of ω .

Change the order of integration. We can do this only if the limits of integration are not functions of the variable of integration:

$$F(\omega) = \int_{-\infty}^{\infty} f_1(\tau) \left[\int_{-\infty}^{\infty} e^{-i\omega t} f_2(t - \tau) dt \right] d\tau$$

Now the integration is first with respect to t then with respect to τ . But the quantity in brackets is the Fourier transform of the time-shifted function $f_2(t - \tau)$, and the Fourier transform of a time-shifted function is

$$f_2(t - \tau) \iff F_2(\omega) e^{-i\omega\tau}$$

so that the quantity in the bracket is equal to $F_2(\omega) e^{-i\omega\tau}$. Therefore,

$$\begin{aligned} F(\omega) &= \int_{-\infty}^{\infty} f_1(\tau) e^{-i\omega\tau} F_2(\omega) d\tau \\ &= F_2(\omega) \int_{-\infty}^{\infty} f_1(\tau) e^{-i\omega\tau} d\tau \end{aligned}$$

That is,

$$F(\omega) = F_1(\omega) F_2(\omega) \quad (3.58)$$

Conclusion: The Fourier transform of the convolution of two functions is equal to the product of the Fourier transforms of the two functions, as stated earlier.

We will see practical applications of convolution and the time-convolution theorem later. At this point we note two different methods of obtaining $f(t)$:

1. Convolve (in the time domain) $f_1(t)$ with $f_2(t)$.

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$$

2. Take the forward Fourier transforms of $f_1(t)$ and $f_2(t)$ to obtain $F_1(\omega)$ and $F_2(\omega)$. Multiply (in the frequency domain) $F_1(\omega)$ by $F_2(\omega)$. Then do an inverse Fourier transform to get back to the time domain:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega) F_2(\omega) e^{+i\omega t} d\omega$$

Implementation of Item 2 above requires caution. In order to multiply spectra in the frequency domain the frequency increment $d\omega$ must be the same for $F_1(\omega)$ and $F_2(\omega)$. Therefore, the time duration of $f_1(t)$ and $f_2(t)$ must be equal before taking their Fourier transforms. Furthermore, each function $f_1(t)$ and $f_2(t)$ must be padded with zeros such that $T \geq \text{Length}[f_1(t)] + \text{Length}[f_2(t)]$

so that the inverse Fourier transform will fit in the window T . For example, let us convolve the list

$$a = (1, 1, 1, 1, 1)$$

with the list

$$b = (1, 1, 1, 1, 1)$$

Convolution in the time domain (which we can do in our heads) results in

$$c = (1, 2, 3, 4, 5, 4, 3, 2, 1) \quad (3.59)$$

The length of c is `Length[a] + Length[b] - 1`. In our example the length of c is 9. If we do the convolution in the frequency domain by multiplication of the transforms of each function we must first pad each function with four zeros so that

$$\text{Length}[c] = \text{Length}[a] + \text{Length}[b] - 1 \quad (3.60)$$

is true. Thus

$$\begin{aligned} F_a &= \text{Fourier}[\text{Join}[a, 0, 0, 0, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \\ F_b &= \text{Fourier}[\text{Join}[b, 0, 0, 0, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \end{aligned}$$

Then

$$c = \text{InverseFourier}[F_a * F_b, \text{FourierParameters} \rightarrow \{1, -1\}]$$

and the answer is

$$c = (1, 2, 3, 4, 5, 4, 3, 2, 1)$$

as expected.

Now suppose we pad with only three zeros. Then,

$$\begin{aligned} F_a &= \text{Fourier}[\text{Join}[a, 0, 0, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \\ F_b &= \text{Fourier}[\text{Join}[b, 0, 0, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \end{aligned}$$

Then

$$c = \text{InverseFourier}[F_a * F_b, \text{FourierParameters} \rightarrow \{1, -1\}]; \quad (3.61)$$

and the answer is

$$c = (2, 2, 3, 4, 5, 4, 3, 2)$$

which is incorrect. The inverse transform no longer fits in the time-domain window because the inverse transform window of 8 points in length isn't long enough to accommodate the entire convolution, which is 9 points. So the last

value to the right (“1”) in Equation (3.59) wraps around and is added to the front end of the (periodic) function. The first value of c , which should be “1” becomes instead $1 + 1 = 2$, an incorrect value. Everything else in c is correct. The “wrapped-around” value has been *added* to the first value. It does not replace it.

Now pad with only two zeros. Then,

$$\begin{aligned} F_a &= \text{Fourier}[\text{Join}[a, 0, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \\ F_b &= \text{Fourier}[\text{Join}[b, 0, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \end{aligned}$$

Then

$$c = \text{InverseFourier}[F_a * F_b, \text{FourierParameters} \rightarrow \{1, -1\}];$$

and the answer is

$$c = (3, 3, 3, 4, 5, 4, 3, 2)$$

and the wraparound is more severe. Now the last two values to the right (“2” and “1”) in Equation (3.59) wrap around and are added to the front end of the (periodic) function. The second value of c , which should be “2” becomes instead $2 + 1 = 3$, and the first value which should be “1” becomes $1 + 2 = 3$. Everything else in c is correct. The two “wrapped-around” values have been *added* to the first two values. They do not simply replace the first two values.

Once more. Pad with only one zero. Then,

$$\begin{aligned} F_a &= \text{Fourier}[\text{Join}[a, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \\ F_b &= \text{Fourier}[\text{Join}[b, 0], \text{FourierParameters} \rightarrow \{1, -1\}]; \end{aligned}$$

Then

$$c = \text{InverseFourier}[F_a * F_b, \text{FourierParameters} \rightarrow \{1, -1\}]; \quad (3.62)$$

and the answer is

$$c = (4, 4, 4, 4, 5, 4,)$$

and the wraparound is more severe. Now the last three values to the right (“3”, “2” and “1”) in Equation (3.59) wrap around and are added to the front end of the (periodic) function. The third value of c , which should be “3” becomes instead $3 + 1 = 4$, the second value which should be “2” becomes $2 + 2 = 4$, and the third value which should be “3” becomes $3 + 1 = 4$. Everything else in c (very little left) is correct. The three “wrapped-around” values have been added to the first three values shown in Equation (3.59). They do not simply replace the first three values. See further `WraparoundExampleTimeDomain.nb` on the CD-ROM that accompanies this volume.

For a frequency-domain filtered seismic trace, if padding is not done then the incorrect result labeled “wraparound” shown in the bottom diagram of

Figure 3.41 is obtained. The first arrivals on “Trace 3” don’t arrive until about 200 ms and so the trace is essentially dead before this time thus allowing the overlapping data to be seen. If, however, there happened to be data over this quiet interval (the first arrivals arriving earlier), then this overlap would interfere adversely with the filtered data.

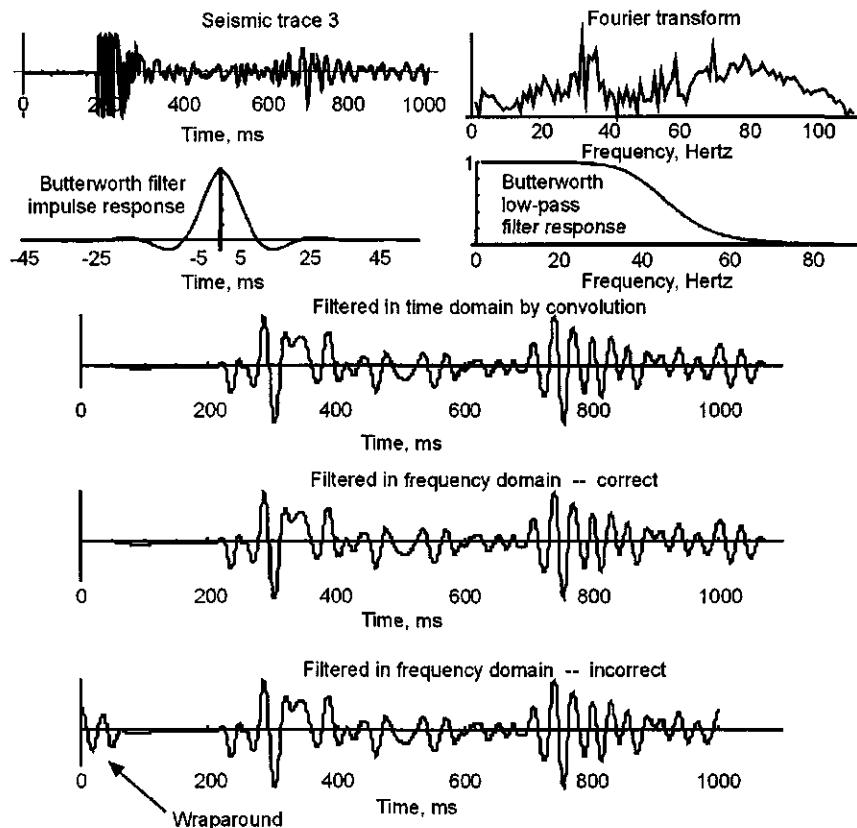


Figure 3.41: Top left: An unfiltered seismic trace (Trace 3). Top right: Fourier transform of seismic trace. Second row left: Impulse response of Butterworth low-pass filter. Second row right: Pass band of filter. Bottom three traces: (a) Time-domain convolution of filter with trace. (b) Filtered trace obtained by multiplication of spectra followed by inverse Fourier transformation as per Item 2. (c) Incorrect result obtained by not padding each function with zeros before Fourier transformation in order to make the length of the filtered output equal to the sum of the length of the input plus the length of the filter. See *Mathematica* program “TimeWraparoundExample.nb”. TimeWraparoundExample.nb \Rightarrow TimeWraparoundExample.cdr \Rightarrow TimeWraparoundExample.wmf

At first glance it would seem to require more computational effort to obtain a convolution by multiplication in the frequency domain instead of a single convolution in the time domain because one has to compute the Fourier transforms of $f_1(t)$ and $f_2(t)$, multiply these together in the frequency domain, and then perform an inverse Fourier transform of the product to get $f(t)$; however, it is often faster to do this for long time series because of the availability of the *Fast Fourier Transform*, a computational procedure that is considered to be one of the major breakthroughs in computational geophysics. The reader can see this by running *Mathematica* program *SyntheticDiffAndInteg.nb* (on the CD-ROM) where the integration is done in the frequency domain (using the Fast Fourier Transform, Section 3.4.2) as well as in the time domain by convolution of “trace 18” with a step function; the time-domain convolution takes considerably longer; however, it can be speeded up dramatically by using the following module instead of *fold* for convolution with the step function.

```
foldcc[a_,b_]:=(la=Length[a];lb=Length[b];
lc=la+lb-1;
bb=Reverse[b];aa=PadLeft[aa,la+lb-1],lc+lb-1];
foldout=Table[Apply[Plus,Take[aa,{i,lb+i-1}]*bb],{i,1,lc}]);
```

An application of the time-convolution theorem

The typical reflection seismic experiment is to generate a disturbance at the surface of the earth and record reflections from the interior of the earth at various offset distances. If $w(t)$ is the disturbance generated, then this “wavelet” from the surface source will be reflected from subsurface contrasts in acoustic impedance, and recorded at various receiver positions at the surface at successively later times, as shown in Figure 3.42. Reflected source wavelets $w(t)$ return to the surface at successively later times from deeper reflectors, or if reflected from the same interface, to farther offset distances.

Finding the model parameters (reflection coefficients and wavelet, for example) from the recorded data is one of the objectives of reflection seismology. Once the subsurface distribution of the reflection coefficients is known, we have imaged the geologic structure. Suppose we equate $w(t)$ with a wavelet generated by a source of energy at the surface of the Earth, and $r(t)$ with a (continuous) sequence of reflection coefficients. Then the seismogram $o(t)$ that is recorded is

$$o(t) = \int_{-\infty}^{\infty} r(\tau)w(t - \tau)d\tau$$

which is a convolution, or because $O(\omega) = R(\omega)W(\omega)$ from the time-convolution theorem,

$$o(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R(\omega)W(\omega)e^{i\omega t}d\omega$$

where

$$o(t) \iff O(\omega)$$

$$\begin{aligned} r(t) &\iff R(\omega) \\ w(t) &\iff W(\omega) \end{aligned}$$

as usual.

The convolution in discrete form may be given as

$$o[k] = \sum_{i=1}^{i=n} r[i] w[k-i]$$

where $k = t/\Delta t$, $i = \tau/\Delta t$, and Δt is the sampling interval.

The function most closely associated with the geology is $r(t)$, the time-sequence of reflection coefficients from successively deeper interfaces between materials of different acoustic impedance contrast. The process of convolution (superposition) has filtered $r(t)$ so that we record a “smeared-out” version of the reflection coefficients, as shown in Figure 3.43. That is, the reflection process of smoothing with $w(t)$ has filtered out the high frequencies present in the rapidly changing sequence of reflection coefficients (“spikes”). The process of convolution in the time is equivalent to multiplication in the frequency domain:

$$O(\omega) = R(\omega)W(\omega)$$

If we could recover $r(t)$ at each receiver station, and display each $r(t)$ side-by-side at their time coordinates, we would have a graphic image of the subsurface geology, complete with any faults and folds.

If the shape of the seismic wavelet $w(t)$ were known, then this might be possible because

$$R(\omega) = \frac{O(\omega)}{W(\omega)}$$

The sequence of reflection coefficients is then simply

$$r(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{O(\omega)}{W(\omega)} e^{i\omega t} d\omega \quad (3.63)$$

Alas, it is not that easy. Of course, this will be possible with noise-free synthetic data, as illustrated in Figure 3.43 where we have convolved a (known) source wavelet $w(t)$ with a reflectivity function $r(t)$ to give a synthetic seismogram $o(t)$; however, the spectrum $W(\omega)$ of the shot pulse is not generally known.

In the real world much more is recorded than just the reflections of interest from geologic interfaces (see Figure 7.3 on page 341 for an example). There are multiples, refractions, wind noise, surface waves, traffic noise, electrical interference from power lines, and a host of unidentified other signals, some of which could tell us much about the geology if only we could decipher them; however, there is some important conceptual insight to be obtained by examining what *might* be done with clean (synthetic) data.

Equation (3.63) is a simple example of an *inverse problem*. We have the recorded data $o(t)$ and we want the model parameters $r(t)$ that produced the

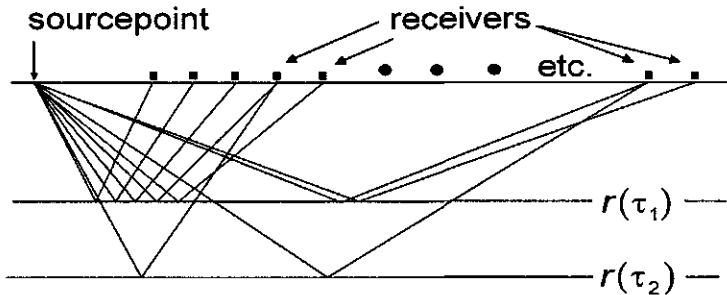


Figure 3.42: The source wavelet $w(t)$ leaves the source point, is reflected from subsurface discontinuities, and returns to the surface at successively later times from deeper reflectors and/or farther offset distances. If the reflection coefficient associated with the first reflector is $r(\tau_1)$, and that for the deeper reflector is $r(\tau_2)$, then the events recorded at a receiver are $r(\tau_1)w(t - \tau_1) + r(\tau_2)w(t - \tau_2)$. τ_1 and τ_2 are defined by the depth to the reflectors, the velocity of the media, and the distance between the source point and the receivers. For many reflectors at a particular offset, the events recorded as a seismic trace $o(t)$ are $o(t) = \sum r(\tau_i)w(t - \tau_i)$, which is a convolution. (hg explore → model1)

data. Those model parameters $r(t)$ are completely defined in our simple case by the *inverse filtering* operation of computing the inverse Fourier transform of the ratio

$$\frac{O(\omega)}{W(\omega)}$$

Recovery of $r(t)$ will require restoration of the high frequencies that are present in the random reflection coefficients (a Dirac delta function contains all frequencies of equal amplitude). In practice, it is necessary to resort to indirect methods to determine $w(t)$, but it can be done with impressive results.

The computational process of recovering $r(t)$ from the recorded data $o(t)$ where

$$o(t) = \int_{-\infty}^{\infty} r(\tau)w(t - \tau) d\tau$$

is called, appropriately, *deconvolution*, and is the subject of Chapter 8 on page 370.

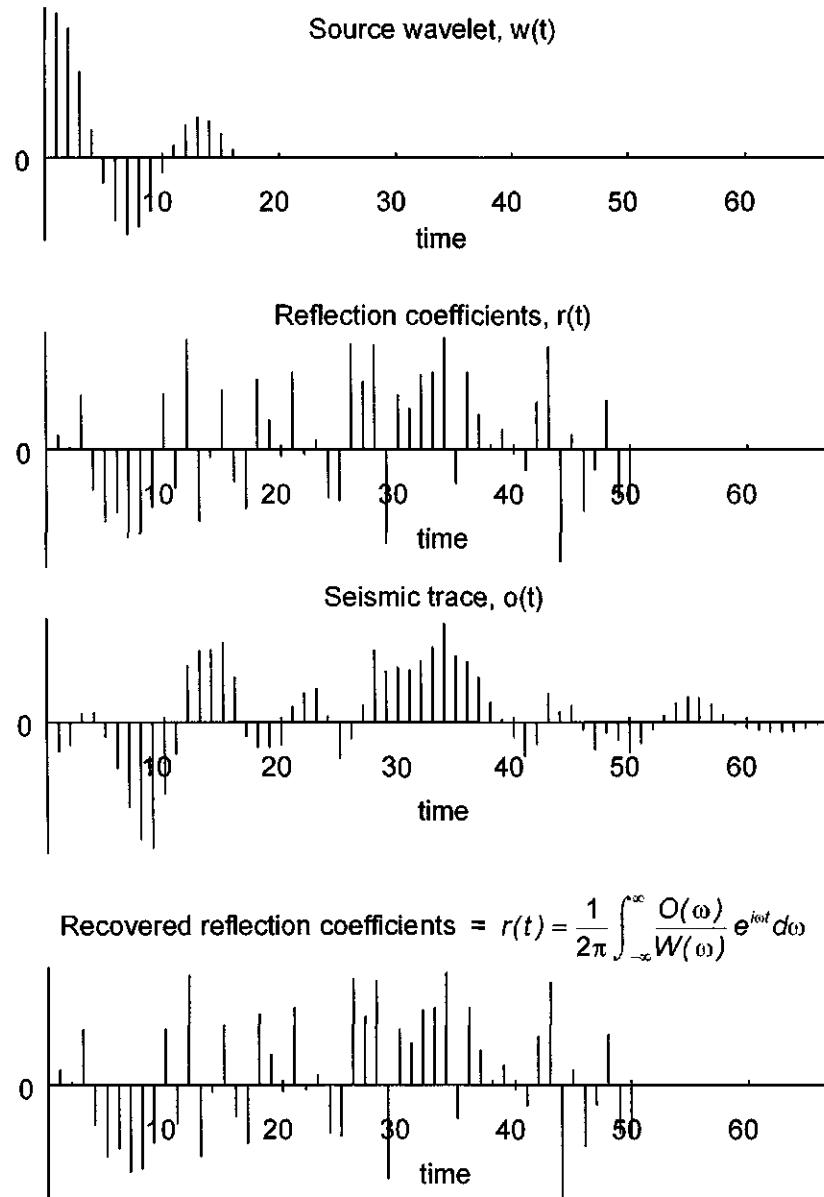


Figure 3.43: Source wavelet $w(t)$ (top) is convolved with reflection coefficients $r(t)$ resulting in seismic trace $o(t)$. Notice the random appearance of the reflection coefficients $r(t)$. Each reflection coefficient represents a subsurface discontinuity between two media of different velocity and/or density. With noise-free data, and if $w(t)$ is known and time-invariant, then it is a simple matter to recover the reflection coefficients $r(t)$ by computing the inverse Fourier transform of $O(\omega)/W(\omega)$. Alas, this simple model is not realistic. (ma → convolv1)

3.5.9 Autocorrelation and Crosscorrelation

Convolution is defined as

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$$

where $f_1(t)$ and $f_2(t)$ are two different functions and $f(t)$ is the convolution of the two. Convolution involves reversing one of the functions $f_1(t)$ or $f_2(t)$ before doing the integration. If neither of these functions is reversed then we have

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(\tau - t) d\tau$$

which graphically amounts to sliding one function without folding past the other to obtain $f(t)$. This is called the crosscorrelation of $f_2(t)$ and $f_1(t)$. It is a search for similarity between $f_1(t)$ and $f_2(t)$. Crosscorrelation is commonly denoted by a five-pointed star, or pentagram (\star), and is not to be confused with the asterisk ($*$) symbol used to denote convolution. That is,

$$f(t) = f_2(t) \star f_1(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$$

Crosscorrelation is similar to convolution but the functions are simply displaced with respect to each other without folding one of them first. For convolution

$$f_1(t) * f_2(t) = f_2(t) * f_1(t)$$

For crosscorrelation, however

$$f_1(t) \star f_2(t) \neq f_2(t) \star f_1(t)$$

Although convolution might be more difficult to visualize because of the combined operations of both folding *and* shifting, crosscorrelation involves a simple shift (lag) of one function relative to the other and is easily visualized. Clearly, if $f_1(t)$ is not equal to $f_2(t)$ then the result depends on the direction of relative shift between $f_1(t)$ and $f_2(t)$.

Changing the variable of integration results in

$$\begin{aligned} f(t) &= f_2(t) \star f_1(t) = \int_{-\infty}^{\infty} f_2(u - t) f_1(u) du = \int_{-\infty}^{\infty} f_2(u) f_1(u + t) du \\ f(-t) &= f_1(t) \star f_2(t) = \int_{-\infty}^{\infty} f_1(u - t) f_2(u) du = \int_{-\infty}^{\infty} f_1(u) f_2(u + t) du \\ f_2(t) \star f_1(t) &= f_1(-t) \star f_2(t) \end{aligned}$$

If $f_2(t) = f_1(t)$ then the operation is one of “autocorrelation” of the function $f_1(t)$ and

$$f(t) = f_1(t) \star f_1(t) = \int_{-\infty}^{\infty} f_1(u) f_1(u - t) du$$

A graphical interpretation of the autocorrelation integral is the sliding of the function $f_1(t)$ past itself as the integration is carried out to obtain $f(t)$ and $f(-t)$. Clearly, there is no difference in the result regardless of the direction of relative shift of $f_1(t)$ with respect to itself. The autocorrelation function $f(t)$ is therefore time-symmetric with respect to $t = 0$. In addition, the maximum value of $f(t)$ occurs at zero time shift (lag) and the value of $f(0)$ is proportional to the energy of the time series.

A *Mathematica* subroutine called `fold` that performs time-domain convolution, autocorrelation, or crosscorrelation was converted from Robinson [148] as follows

```
fold[a_, b_] := (lc = Length[a] + Length[b] - 1;
c = Table[0, {lc}];
Do[Do[k = i + j - 1;
c[[k]] = c[[k]] + a[[i]]*b[[j]], {j, 1, Length[b]},
{i, 1, Length[a]}]]);
```

The output of an operation is in the vector c . For convolution, the order of input functions is not important. For crosscorrelation it is (see for example Section 8.3.1 on page 428). For autocorrelation of $f_1(t)$ the input to `fold` is `fold[f_1,Reverse[f_1]]`.

Crosscorrelation gives a measure of the similarity between two different waveforms. It is an essential step in the early processing of vibroseis data (page 342) where essentially the pilot sweep searches for itself in the recorded uncorrelated seismic trace. At those times when it is superimposed on one of the many reflected pilot sweeps buried in the recorded trace then the crosscorrelation coefficient is high. At other times it is lower. Those times for which the crosscorrelation is high are identified as reflections. So crosscorrelation gives a measure of similarity. One of the more remarkable interpretive applications of a search for similarity between traces has to be the “coherence cube” as described by Bahorich and Farmer [11] and shown in Figure 3.44. Neidell and Taner [123] discussed the concept of semblance and crosscorrelation (“the most familiar of the likeness criteria”) and illustrated the differences obtained from crosscorrelation, semblance, and a simple sum. Semblance S is defined as [123]

$$S[k] = \frac{\sum_{j=k-N/2}^{j=k+N/2} \left[\sum_{i=1}^M f[i, j] \right]^2}{M \sum_{j=k-N/2}^{j=k+N/2} \sum_{i=1}^M f[i, j]^2}$$

where the trace is f , $f[i, j]$ is the j th sample of the i th trace in a window of width N centered at time k , and M is the number of traces.



Figure 3.44: Looking down on a horizontal time-slice in a 3-dimensional cube of seismic data. The coherence processing technique measures the similarity of waveform from trace to trace in a 3-dimensional cube of seismic data. If no fault discontinuities are present in the data cube then a high degree of similarity will be present in the waveform from trace to trace in any horizontal direction. Where a fault discontinuity is present, however, this offsets the traces on each side of the fault and the similarity decreases. The “coherence” can be measured in a number of ways but the key is the change in similarity across the discontinuity. For a cube of data the abrupt change in waveform similarity across even steep to vertical faults not only reveals the presence of the fault(s) but also clearly illustrates how they merge and change throw with depth as adjacent time slices are progressively viewed. Figure from Bahorich and Farmer [11].

3.5.10 The frequency-convolution theorem

The development of convolution in the frequency domain parallels the derivation of the *time-convolution theorem*. Given any two continuous functions of frequency $F_1(\omega)$ and $F_2(\omega)$, define a function $F(\omega)$ such that

$$F(\omega) = \int_{-\infty}^{\infty} F_1(\omega') F_2(\omega - \omega') d\omega'$$

where $F(\omega)$ is defined as the **convolution** of $F_1(\omega)$ and $F_2(\omega)$.

We have the usual notation:

$$\begin{aligned} f_1(t) &\iff F_1(\omega) \\ f_2(t) &\iff F_2(\omega) \\ f(t) &\iff F(\omega) \end{aligned}$$

The *frequency convolution theorem* then states:

$$\left[F(\omega) = \int_{-\infty}^{\infty} F_1(\omega') F_2(\omega - \omega') d\omega' \right] \iff [f(t) = f_1(t)f_2(t)]$$

which says that the inverse Fourier transform $f(t)$ of the convolution $F(\omega)$ of two functions $F_1(\omega)$ and $F_2(\omega)$ is equal to the product of the inverse Fourier transforms $f_1(t)$ and $f_2(t)$, i.e.,

$$f(t) = f_1(t)f_2(t)$$

The significance of this is immediately apparent. If $f_2(t)$ is an analysis window of some shape and $f_1(t)$ is the signal being analyzed, then analysis of the signal in an effort to recover its “true” spectrum means that each component affects the amplitudes of every other component by superposition of the Fourier transform of the analysis window, together with its side lobes, on each “true” spectral component. Therefore, the true spectrum cannot be recovered exactly.

To illustrate the frequency convolution theorem we proceed as follows,

$$f(t) \iff F(\omega)$$

by definition, i.e.,

$$\begin{aligned} f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{+i\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} F_1(\omega') F_2(\omega - \omega') d\omega' \right] e^{+i\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{+i\omega t} \left[\int_{-\infty}^{\infty} F_1(\omega') F_2(\omega - \omega') d\omega' \right] d\omega \end{aligned}$$

As this equation stands, integration with respect to ω' is carried out first to get a particular value of $F(\omega)$. Then integration with respect to ω is carried out to get the (in general) complex function $f(t)$.

We can change the order of integration if the limits of integration are not functions of the variable of integration:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega') \left[\int_{-\infty}^{\infty} e^{+i\omega t} F_2(\omega - \omega') d\omega \right] d\omega' \quad (3.64)$$

Now the integration is first with respect to ω then with respect to ω' .

Recalling the time-shifting theorem and making an analogy to obtain a “frequency shifting theorem”,

$$\begin{aligned} f(t - \tau) &\iff F(\omega)e^{-i\omega\tau} \\ F(\omega - \omega') &\iff f(t)e^{-i\omega't} \end{aligned}$$

it is seen that the quantity in brackets in Equation (3.64) is the inverse Fourier transform of the frequency-shifted function $F_2(\omega - \omega')$.

Then the Fourier transform pair for the frequency-shifted function $F_2(\omega)$ is

$$f_2(t) e^{-i\omega't} \iff F_2(\omega - \omega')$$

so that the quantity in the bracket in Equation (3.64) is equal to $f_2(t)e^{-i\omega't}$. That is,

$$f_2(t)e^{-i\omega't} = \int_{-\infty}^{\infty} e^{+i\omega t} F_2(\omega - \omega') d\omega$$

Therefore,

$$\begin{aligned} f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega') \left[\int_{-\infty}^{\infty} e^{+i\omega t} F_2(\omega - \omega') d\omega \right] d\omega' \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega') \left[f_2(t) e^{-i\omega't} \right] d\omega' \\ &= f_2(t) \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega') e^{-i\omega't} d\omega' \\ &= f_2(t) f_1(t) \end{aligned}$$

and we have

$$f(t) = f_1(t)f_2(t) \quad (3.65)$$

The notation above is parallel to that used to arrive at Equation (3.58) where we showed that multiplication in the frequency domain was equivalent to convolution in the time domain. In arriving at Equation (3.65) we conclude that multiplication in the time domain is equivalent to convolution in the frequency domain!

The effect of the analysis window revisited

Now we see more clearly the reason for what is commonly referred to as “leakage” when an attempt is made to determine the “true” Fourier spectrum of a signal. Selecting only a portion of the signal to analyze is equivalent to multiplication of the signal by an analysis window equal to zero outside of the window and unity inside the window; i.e., a rectangular window. In the frequency domain, this is equivalent to the superposition of the Fourier transform of just the window over each of the true spectral components, whatever they are. The transform of the window is superimposed on top of the amplitude of each true spectral component. The side lobes of each true spectral component therefore interfere with every other true spectral component. If the analysis window could be made infinitely wide, then, of course, there would be no interference at all because $1 \longleftrightarrow \delta(\omega)$. The Fourier spectra of the sum of three sinusoids is shown Figure 3.45. In Figure 3.45, the spectral components are separated by 10 Hertz. As this separation decreases it becomes apparent that recovery of the true relative amplitudes of spectral components is not possible (Figure 3.46). This is true regardless of the shape of the analysis window in the time domain.

Table 3.5: Fourier Transform Pairs

Mathematical Operation	Time domain, $f(t)$	Frequency domain, $F(\nu)$
Basic pair	$f(t)$	$F(\nu)$
Interchange of variable	$F(t)$	$f(-\nu)$
Negative argument	$f(-t)$	$F(-\nu)$
Complex conjugate	$f^*(t)$	$F^*(-\nu)$
Complex conjugate	$f^*(-t)$	$F^*(\nu)$
Scale time by constant a	$g(at)$	$\frac{1}{ a }F(\nu/a)$
Scale frequency by constant a	$\frac{1}{ a }g(t/a)$	$F(a\nu)$
Multiplication by constant, a	$a f(t)$	$a F(\nu)$
Sum of $f_1(t) + f_2(t)$	$f_1(t) + f_2(t)$	$F_1(\nu) + F_2(\nu)$
$f(t)$ shift in time τ	$f(t - \tau)$	$F(\nu) e^{-i2\pi\nu\tau}$
$F(\nu)$ shifted in frequency ν_0	$f(t) e^{+i2\pi\nu_0 t}$	$F(\nu - \nu_0)$
Convolution in time	$\int_{-\infty}^{+\infty} f_1(\tau)f_2(t - \tau) d\tau$	$F_1(\nu) F_2(\nu)$
Convolution in frequency	$f_1(t) f_2(t)$	$\int_{-\infty}^{+\infty} F_1(\nu_0) F_2(\nu - \nu_0) d\nu_0$
Autocorrelation in time	$\int_{-\infty}^{+\infty} f(\tau) f^*(\tau - t) d\tau$	$F(\nu) F^*(\nu)$
Autocorrelation in frequency	$f(t) f^*(t)$	$\int_{-\infty}^{+\infty} F(\nu_0) F^*(\nu_0 - \nu) d\nu_0$
Crosscorrelation in time	$\int_{-\infty}^{+\infty} f_1(t) f_2^*(t - \tau) d\tau$	$F_1(\nu) F_2^*(\nu)$
Crosscorrelation in frequency	$f_1(t) f_2^*(t)$	$\int_{-\infty}^{+\infty} F_1(\nu_0) F_2^*(\nu_0 - \nu) d\nu_0$
Differentiation in time	$d f(t)/dt$	$i 2\pi\nu F(\nu)$
Differentiation in frequency	$-i 2\pi t f(t)$	$d F(\nu)/d\nu$
Rectangular pulse $f(t)$	$f(t) = 1; -\frac{1}{2} < t < \frac{1}{2}$	$\frac{\sin(\pi\nu)}{\pi\nu}$
Rectangular spectrum, $F(\nu)$	$\frac{\sin(\pi t)}{\pi t}$	$F(\nu) = 1; -\frac{1}{2} < \nu < \frac{1}{2}$
Dirac delta function at $t = 0$	$\delta(t) = \infty$	1

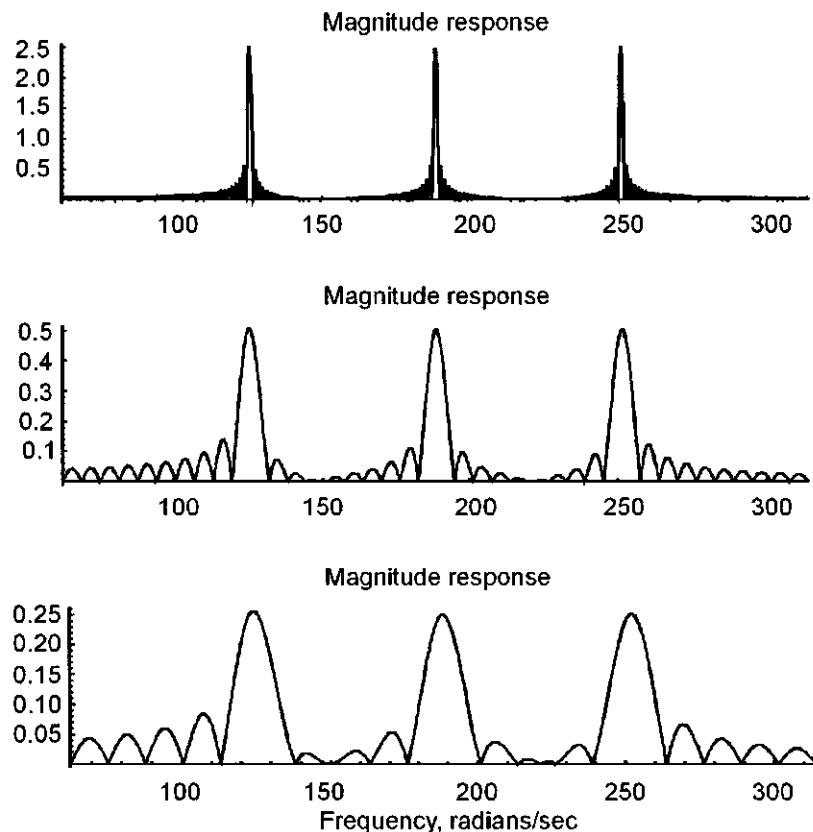


Figure 3.45: Continuous spectra showing superposition (convolution) in the frequency domain. The time-domain input signal is the sum of three sinusoids of 30, 40, and 50 Hertz, all of equal amplitude. Axes are labeled in radians/sec. Top: rectangular analysis window is 5 sec wide. Middle, 1 sec. Bottom: 0.5 sec. Each spectral component has superimposed upon it the modulus of the analysis window because multiplication in the time domain is equivalent to convolution (superposition) in the frequency domain. Depending upon the shape and width of the time domain analysis window, adjacent frequency components will interfere with each other to a greater or lesser extent. If the shape of the analysis window in the time domain is a simple "do nothing" rectangular analysis window, then interference from side lobes associated with the analysis window is at a maximum, as shown above, especially with the narrowest window. Note the progressive widening of the central and side lobes as the window narrows. (ma interf1 → cd → interf1)

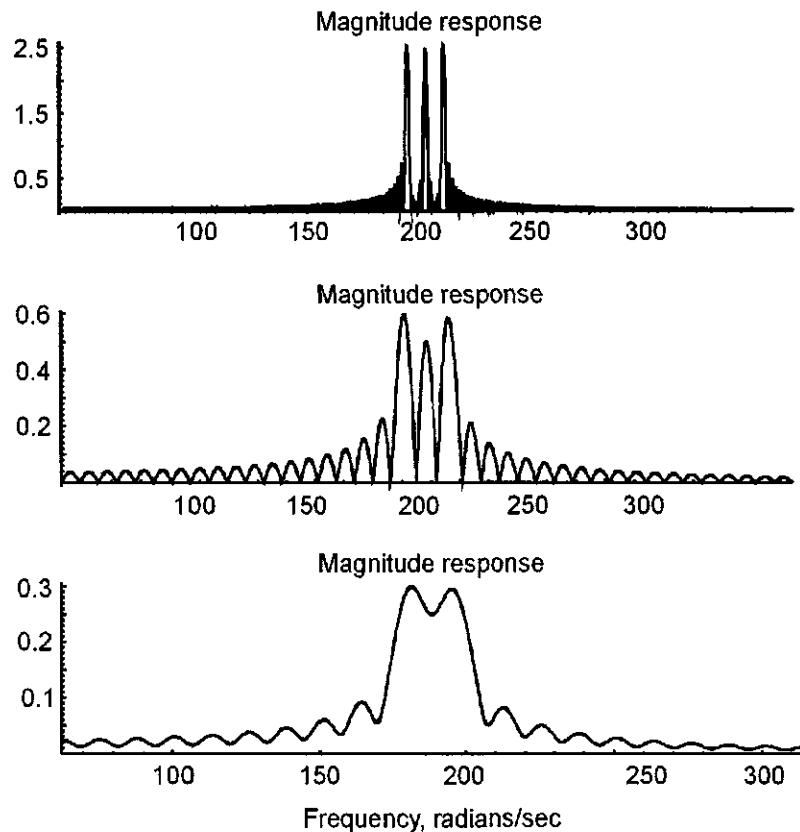


Figure 3.46: Signal is the sum of three sinusoids of 29, 30, and 31 Hertz, all of equal amplitude. For the same analysis windows used in Figure 3.45, a decrease in the separation of spectral components results in even more interference and illustrates that perfect recovery of true relative amplitudes is not possible, even if the only frequencies present are integer multiples of the fundamental frequency $1/T$. Analysis windows of 5, 1, and 0.5 sec. Axes are labeled in radians/sec. (ma interf2 → cd → interf2)

3.5.11 Hilbert Transforms

The Hilbert transform is a transformation that changes the phase of a Fourier spectral component depending on the sign of its frequency; the transform has no affect on the amplitudes of spectral components. The Hilbert transform converts cosine waves to sine waves and vice versa (see page 138), and because the conversion has no effect on the amplitude of the sinusoid a signal and its Hilbert transform have identical energy; only changes in amplitude can change the energy. The Hilbert transform relates the real part of the Fourier transform of a causal function to the imaginary part so causality is an important part of the Hilbert transform.

Any function $h(t)$ can be expressed as the sum of an even component $h_e(t)$ and an odd component $h_o(t)$. That is,

$$h(t) = h_e(t) + h_o(t) \quad (3.66)$$

If the function is causal as shown in Figure 3.47 then

$$h_e(t) = h_e(-t) \quad (3.67)$$

$$h_o(t) = -h_o(-t) \quad (3.68)$$

$$h(-t) = h_e(t) - h_o(t) \quad (3.69)$$

Solving (3.66) and (3.69) in terms of $h(t)$ and $h(-t)$ results in

$$h_e(t) = \frac{1}{2} [h(t) + h(-t)] \quad (3.70)$$

$$h_o(t) = \frac{1}{2} [h(t) - h(-t)] \quad (3.71)$$

So, as stated above

$$h(t) = \frac{1}{2} [h(t) + h(-t)] + \frac{1}{2} [h(t) - h(-t)] \quad (3.72)$$

which is easily shown to be true by expanding and simplifying the right-hand side of (3.72). Thus,

$$h(t) = h_e(t) + h_o(t) \quad \text{for all times } t \quad (3.73)$$

Equation (3.73) is true regardless of the shape of $h(t)$ or whether or not $h(t)$ is causal (whether or not $h(t)$ is zero for $t < 0$).

If $h(t)$ is causal, however, then a special symmetry and relationship exists between the even and odd parts of $h(t)$ as defined by (3.70) and (3.71) and illustrated in Figure 3.47. The function $h(t)$ shown in the top diagram of Figure 3.47 is causal. Because of the symmetry between the odd and even parts it is clear from the figure that if you know the even part then you can determine the odd part by multiplying the amplitude values of the even part *for negative times* by -1 and by multiplying the even part by $+1$ for *positive times*. From inspection of Figure 3.47 the causal function $h(t)$ is equal to

$$h(t) = 2 h_e(t) = 2 h_o(t) \quad \text{for } t \geq 0 \quad (3.74)$$

so that if all you know is the even part then you can completely recover the function $h(t)$ itself.

In the frequency domain the real and imaginary parts of the Fourier transform of a causal function are said to be Hilbert transforms of each other. That is, the Fourier transforms of the odd and even parts of a causal function constitute a Hilbert transform pair.

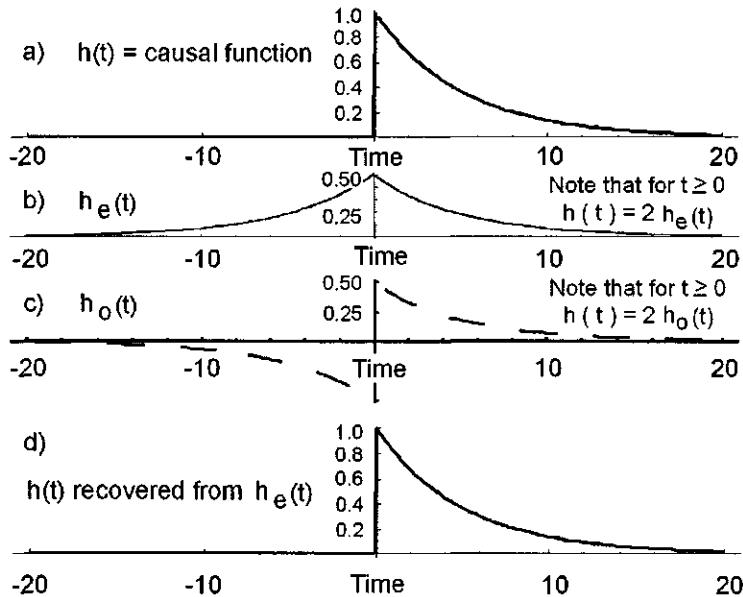


Figure 3.47: (a) Continuous causal function $h(t)$. (b) Even part $h_e(t)$ of the causal function. (c) Odd part $h_o(t)$. (d) $h(t) = 2h_e(t) = 2h_o(t)$ for $t \geq 0$. Note that $h(t) = h_e(t) + h_o(t)$ for all times t . (Hilbert Transform Continuous.nb → Hilbert Transform Continuous.cdr → Hilbert Transform Continuous.wmf)

In the frequency domain the conversion from $h_e(t)$ to $h_o(t)$ can be thought of as a 180° phase difference between the phase angles for the positive and negative frequencies because

$$e^{i\pi} = e^{-i\pi} = -1$$

This interpretation can be confirmed by taking the Fourier transform of the Sign function and observing its phase angle spectrum, which is shown in Figure 3.48; the phase angle difference between any positive and negative frequency is 180° . We can deduce, therefore, that Hilbert transformation can be accomplished by the addition of -90° to the Fourier phase angle spectrum that corresponds to the positive frequencies and by the addition of $+90^\circ$ to the phase angles for the negative frequencies. This gives the desired sum of 180° for the total phase difference between the phase angles for the positive and negative frequencies

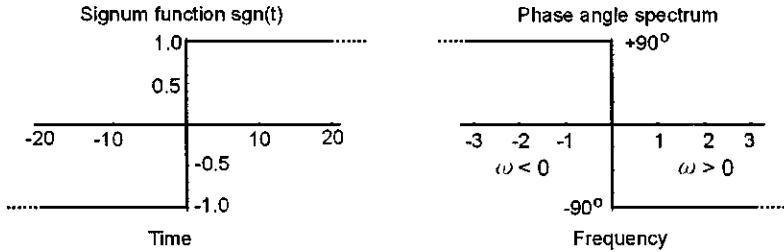


Figure 3.48: Left: The continuous non-causal time-domain function $\text{sgn}(t)$. Right: The Fourier phase spectrum for this function. Conclusion: The Hilbert transform of some real, causal function $h(t)$ can be obtained by taking the Fourier transform of the even part $h_e(t)$ and adding -90° to the phase angles for $\omega > 0$ and adding $+90^\circ$ to the phase angles for $\omega < 0$, and then taking the inverse Fourier transform. The result is the odd part $h_o(t)$ of $h(t)$. The sign of the phase angles for $\omega > 0$ and $\omega < 0$ is a direct consequence of the sign convention we chose for this volume (see page 52). (HilbertWhy90Degrees.nb → Why90Degrees.cdr → Reason90Degrees.wmf)

and at the same time provides the (odd) symmetry between the positive and negative frequencies. This operation does not change the magnitudes of any of the spectral components.

We cannot, however, subtract 60° from the phase angles for the positive frequencies and add 120° to the phase angles for the negative frequencies and expect to obtain a pure real time-domain function even though the overall phase difference would be 180° . Any unsymmetric phase manipulation instead of the phase angle spectrum shown in Figure 3.48 will cause a time shift or a complex inverse or both. That is, for a real, causal function whose frequency response is $H(\omega)$, if the phase angles for values of $\omega > 0$ all have -90° subtracted from them then the phase angles for frequencies $\omega < 0$ must all have $+90^\circ$ added to them. This modification to the phase spectrum is the phase spectrum $\phi(\omega)$ of the *signum* function $\text{Sign}(t)$, which is the time-domain operator that allowed us to determine $h_o(t)$ from $h_e(t)$. Therefore, a Hilbert transformation to obtain the odd part of a causal function from its known even part can be done in the Fourier domain by applying -90° and $+90^\circ$ phase shifts corresponding to the positive and negative frequencies, respectively. This can be accomplished by multiplying $H(\omega)$ by the appropriate left or right hand side of the equations

$$i = e^{+i\pi/2} \quad (3.75)$$

$$-i = e^{-i\pi/2} \quad (3.76)$$

Now we see why we chose to *subtract* 90° from the phase angles for positive frequencies and *add* 90° to the phase angles corresponding to negative frequencies instead of vice versa; this sign convention comes entirely from the one we

chose to adopt in this volume for the b_n Fourier series coefficients as discussed in Section 3.3.1 on page 52 and used in Equation (3.10).

What if $h(t)$ is not causal? An arbitrary function $h(t)$ defined by

$$h(t) = \sqrt{\pi} e^{-t^2}$$

is shown in Figure 3.49. This function is shown shifted with respect to $t = 0$ enough to the left in column (a) row 1 to make it non-causal. In column (b) it is causal and starts at $t = 0$. In column (c) it has been shifted to the right by some arbitrary amount τ . For the causal versions of $h(t)$ shown in columns (b) and (c) it is possible to determine the odd part $h_o(t)$ from the even part $h_e(t)$ by multiplication of $h_e(t)$ by $\text{Sign}[t]$ where $h_e(t)$ and $h_o(t)$ are defined by (3.70) and (3.71). If $h(t)$ is not causal then examination of Figure 3.49 column (a) row 5 shows that it is no longer possible to recover the odd part of $h(t)$ from the even part (or vice versa).

So the Hilbert transform defines the mathematical relationship between the Fourier transforms of the even and odd parts of a causal time-domain function. For a causal function the Fourier amplitude and phase spectra are also a Hilbert transform pair. The causal function will also be minimum-delay (minimum-phase in the frequency domain). More on this on page 156. A formal mathematical derivation of Hilbert transformation in the frequency domain to get the imaginary part from the real part and vice versa is provided on page 156 with a corresponding computer program (*hilbertF*) on page 530. Practical applications are introduced below and in Sections 5.8 and 8.5.2.

In the frequency domain the real and imaginary parts of the Fourier transform of our causal function are the components of a phasor as shown in Figure 3.50. The behavior of a phasor as 90° is subtracted from the phase spectrum for positive frequencies and 90° is added to the phase angles for negative frequencies is shown in Figure 3.50. This is equivalent to a complex multiplication as follows. If $a(\omega) + i b(\omega)$ represents the phasor as shown in Figure 3.50, then

$$\begin{aligned} i [a(\omega) + i b(\omega)] &= -b(\omega) + i a(\omega) \quad \text{for } \omega < 0 \\ -i [a(\omega) + i b(\omega)] &= b(\omega) - i a(\omega) \quad \text{for } \omega > 0 \end{aligned}$$

and because

$$\begin{aligned} i &= e^{+i\pi/2} \\ -i &= e^{-i\pi/2} \end{aligned}$$

then

$$[a(\omega) + i b(\omega)] e^{-i\pi/2} = b(\omega) - i a(\omega) \quad \text{for } \omega > 0 \quad (3.77)$$

$$[a(\omega) + i b(\omega)] e^{+i\pi/2} = -b(\omega) + i a(\omega) \quad \text{for } \omega < 0 \quad (3.78)$$

which simply says that 90° is added to the phase angle of the phasors for $\omega < 0$ and 90° is subtracted from the phase angle of the phasor for $\omega > 0$.

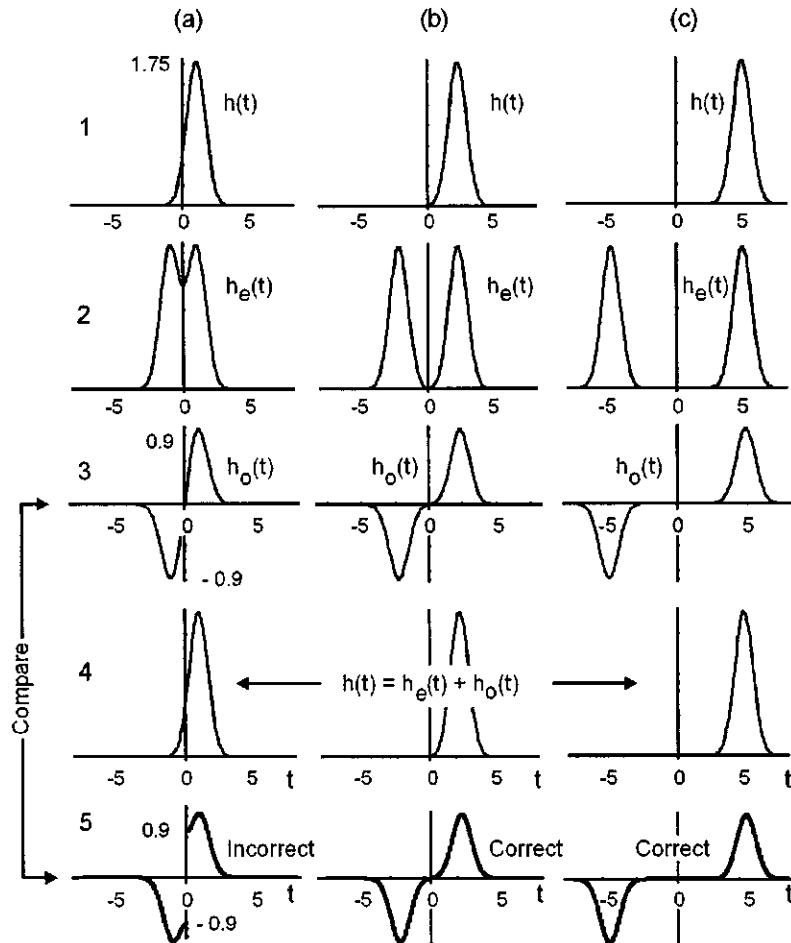


Figure 3.49: Last row in each column obtained by multiplication of even part of $h(t)$ (row 2 in each column) by $\text{Sign}[t]$. For each column compare row 3 with row 5. (a) Left column. Non-causal function $h(t)$. Although $h(t) = h_e(t) + h_o(t)$ for all t , it is no longer possible to recover $h_o(t)$ from $h_e(t)$ or vice versa by multiplication of the even or odd part by $\text{Sign}[t]$ as can be seen at column (a) row 5. (b) Middle column. $h(t)$ shifted to right by just enough to make it causal. Again $h(t) = h_e(t) + h_o(t)$ for all t and can now recover $h_o(t)$ from $h_e(t)$ or vice versa by multiplication of the even part by $\text{Sign}[t]$ as shown at column (b) row 5. (c) Right column. $h(t)$ shifted to right by arbitrary amount τ so it is still causal and again $h(t) = h_e(t) + h_o(t)$ for all t . Can recover $h_o(t)$ from $h_e(t)$ (or vice versa) by multiplication of the even part by $\text{Sign}[t]$ as shown at column (c) row 5. (HilbertTransformAndCausality.nb → HilbertTransformAndCausality.cdr → HilbertTransformAndCausality.wmf)

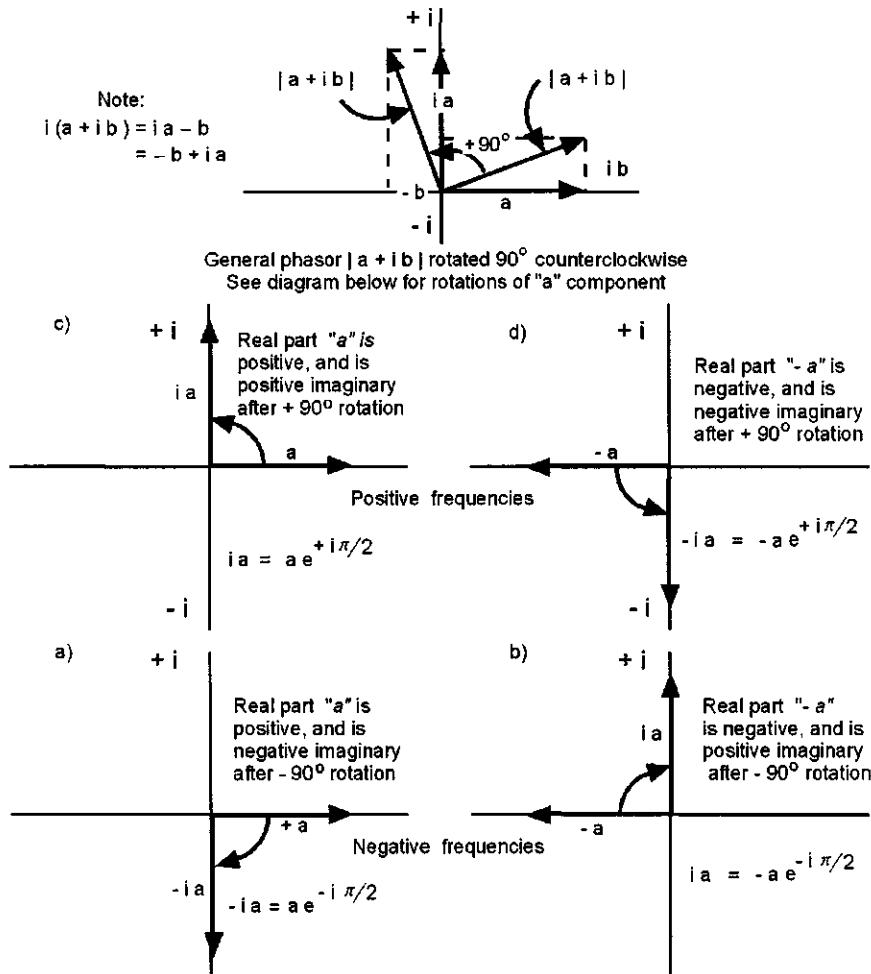


Figure 3.50: Convention. Top: shows projections of real and imaginary parts in complex plane. Positive ω : (a) Phasor with positive real part \rightarrow positive imaginary after Hilbert transformation ($+90^\circ$ rotation). (b) Phasor with negative real part \rightarrow becomes negative imaginary after Hilbert transformation ($+90^\circ$ rotation). Negative ω : (c) Phasor with positive real part \rightarrow negative imaginary after Hilbert transformation (-90° rotation). (d) Phasor with negative real part \rightarrow positive imaginary after Hilbert transformation (-90° rotation). Counterclockwise rotations denote positive phase shifts; clockwise shifts are negative. (HilbertTransforms.cdr \rightarrow RotatePhasors.wmf)

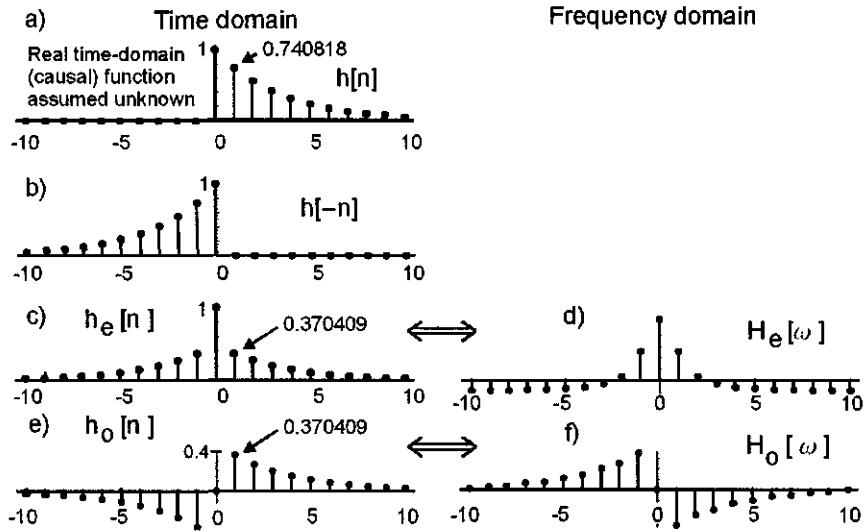


Figure 3.51: Left-hand column is time-domain, right-hand is frequency domain. Symbol \iff denotes Fourier transform pair. (a) A causal real function $h[n]$. (b) $h[-n]$. (c) Even part $h_e[n]$ of $h[n]$ and (d) Fourier transform $H_e[\omega]$. (e) Odd part $h_o[n]$ and (f) Fourier transform $H_o[\omega]$. (HilbertDiscreteEvenAndOdd.nb \rightarrow HilbertTransforms.cdr \rightarrow Tutorial1.wmf)

Shared point at $n = 0$. Determination of causal time-domain function $h[n]$ from $h_e[n]$ by suppressing the negative times of the even part and multiplying the values for $t > 0$ by 2.

The diagram in Figure 3.51a is a causal discrete function $h[n]$ which, because it is real, will have a Fourier transform whose real part has even symmetry, and whose imaginary part has odd symmetry. Its time-reverse $h[-n]$ is shown in Figure 3.51b. Next is the function $h_e[n]$ with even symmetry. The bottom diagram is the function $h_o[n]$ with odd symmetry. Examination of Figure 3.51 shows that the sign and magnitude of the even and odd parts is the same for positive times, but although the magnitude is the same for negative times, the sign is reversed. The real part of the Fourier transform of the real function will be even; the imaginary part will have odd symmetry (Figure 3.51).

The *Mathematica* program used to generate Figure 3.51 follows. Note *Mathematica*'s “replace” notation ($h/.n \rightarrow -n$).

```
(* Program HilbertDiscreteEvenAndOdd.nb *)
psize = 7;
h = UnitStep[n]*E^(-0.3*n);
ListPlot[Transpose[{Table[n,{n, -10, 10}],Table[h,{n, -10, 10}]}],
 PlotRange -> All, PlotLabel -> "h[n]", PlotStyle ->
 AbsolutePointSize[psize], AspectRatio -> 0.2, ImageSize -> 488];
```

```

ListPlot[Transpose[{Table[n, {n, -10, 10}], Table[UnitStep[-n]*
E^(+.3*n), {n, -10, 10}]}], PlotRange->All, PlotLabel->"h[-n]",
PlotStyle -> AbsolutePointSize[psize], AspectRatio -> 0.2,
ImageSize -> 488];
he = h + (h /. n -> -n)/2;
ListPlot[Transpose[{Table[n, {n, -10, 10}], Table[he, {n, -10, 10}]}],
PlotRange -> All, PlotLabel -> "he", PlotStyle ->
AbsolutePointSize[psize], AspectRatio -> 0.2, ImageSize -> 488];
ho = h - (h /. n -> -n)/2;
ListPlot[Transpose[{Table[n, {n, -10, 10}], Table[ho, {n, -10, 10}]}],
PlotRange -> All, PlotLabel -> "ho[n]", PlotStyle ->
AbsolutePointSize[psize], AspectRatio -> 0.2, ImageSize -> 488];
ListPlot[Transpose[{Table[n, {n, -10, 10}], Table[he + ho, {n, -10, 10}]}],
PlotRange -> All, PlotLabel -> "he[n] + ho[n]", PlotStyle ->
AbsolutePointSize[psize], AspectRatio -> 0.2, ImageSize -> 488];
RotateLeft[Table[he, {n, -10, 10}], Floor[21/2]];
He = Fourier[RotateLeft[Table[he, {n, -10, 10}], Floor[21/2]],
FourierParameters -> {1, -1}];
Ho = Im[Fourier[RotateLeft[Table[ho, {n, -10, 10}], Floor[21/2]],
FourierParameters -> {1, -1}]];
ListPlot[Transpose[{Table[n, {n, -10, 10}], RotateRight[He,
Floor[Length[He]/2]]}], PlotRange -> All, PlotJoined -> False,
PlotLabel -> "He", PlotStyle -> AbsolutePointSize[psize],
AspectRatio -> 0.2, ImageSize -> 488];
ListPlot[Transpose[{Table[n, {n, -10, 10}], RotateRight[Ho,
Floor[Length[Ho]/2]]}], PlotLabel -> "Ho", PlotRange -> All,
PlotJoined -> False, PlotStyle -> AbsolutePointSize[psize],
AspectRatio -> 0.2, ImageSize -> 488];

```

Inspection of $h_e[n]$ and $h_o[n]$ shows that $h_o[n]$ can be determined from $h_e[n]$ after multiplication of $h_e[n]$ by $\text{Sign}[n]$. We have determined the odd part $h_o[n]$ from the even part $h_e[n]$ without entering the frequency domain. $h[n]$ and $h[-n]$ have a common value at $n = 0$ ($t = 0$), and this value must therefore be divided by 2 ([129, Eqn. 10.4]).

In order to recover the causal function $h[n]$ itself using Equation (3.74) we therefore require a statement like

$$h[n] = 2 h_e[n] u[n] - h_e[0] \delta[n]; \quad (3.79)$$

where $u[n]$ is the unit step function and $\delta[n]$ is the Kronecker delta function. The *Mathematica* expression for this is

```

h=Table[2*he[[n]]*DiscreteStep[n-1]-  
he[[n]]*KroneckerDelta[n-1],{n,1,Length[h]}];

```

and the resultant output is

$$\begin{aligned}
h[n] = & \{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, \\
& 0.818731, 0.67032, 0.548812, 0.449329, 0.367879, \\
& 0.301194, 0.246597, 0.201897, 0.165299, 0.135335 \}
\end{aligned}$$

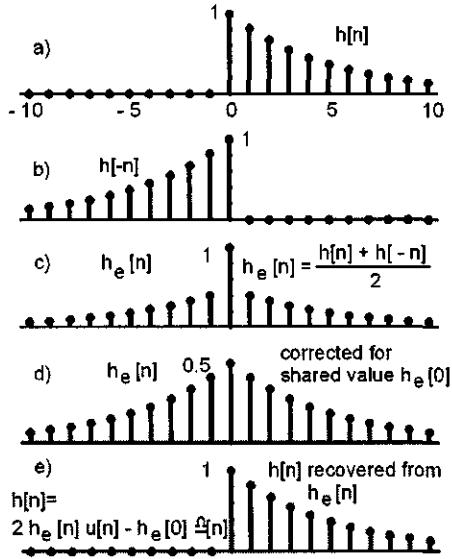


Figure 3.52: (a) Causal function $h[n]$. (b) Time reverse $h[-n]$. (c) Even part of $h[n]$ computed from Equation (3.70). (d) Even part $h_e[n]$ corrected by Equation (3.79) for the shared point $h_e[0]$. (e) Causal function $h[n]$ recovered from even part $h_e[n]$ using Equation (3.79). (HilbertTransformCausalFunctionFromEvenPart.nb → HilbertTransforms.cdr → hFromEven.wmf)

with the result that $h[n]$ has been successfully recovered from only its even part $h_e[n]$ as shown in Figure 3.52. Note, however, from Figure 3.51 that $h_o[0] = 0$, so we can recover $h[n]$ from $h_o[n]$ only for $n \neq 0$ [129, p. 665].

Summary

In order to generate the Hilbert transform of $f(t)$, where $f(t)$ is commonly the seismic trace,

1. Obtain $F(\omega)$ by taking the Fourier transform of $f(t)$.
2. Delay the phasor representing $F(\omega)$ by $-\pi/2$ radians for the positive frequencies,
3. Advance the phasor representing $F(\omega)$ by $+\pi/2$ radians for the negative frequencies, and
4. Obtain the Hilbert transform by taking the inverse Fourier transform of the phase-modified function $F(\omega)$.

The mathematical operations described above lead to a Hilbert transformation. After Hilbert transformation you end up in the same domain. If the Hilbert transform is used to determine the minimum phase spectrum of an unknown wavelet, then the conditions listed on page 499 must be met.

Hilbert transform of a sinusoid

The Hilbert transform converts sine waves to cosine waves and vice versa (except for sign, as described in the following paragraphs). In the frequency domain, the Fourier transform $F(\omega)$ of a (continuous) cosine wave of frequency ω_0 is

$$F(\omega) = \pi \delta(\omega - \omega_0) + \pi\delta(\omega + \omega_0)$$

as shown in Figures 3.53 and 3.54. A cosine wave of frequency ω_0 is represented by a non-zero amplitude at frequencies of ω_0 and $-\omega_0$. Results of Hilbert transform operations denoted by \mathcal{H} in Figures 3.53 and 3.54 are shown in the time domain (left column) and frequency domain (right column). The Hilbert transform of a cosine wave is a sine wave. This result and those that follow were arrived at by subtracting 90° from the phase angles for the positive frequencies, and adding 90° to the phase angles associated with the negative frequencies. A second Hilbert transformation on the sine wave gives a negative cosine wave, a third yields a negative sine wave, and finally not until the fourth Hilbert transformation do we arrive at the starting positive cosine wave.

In summary, and with reference to Figure 3.53,

$$\cos \omega t \implies \sin \omega t \implies -\cos \omega t \implies -\sin \omega t \implies \cos \omega t$$

Here is a *Mathematica* program that takes the Hilbert transform of the discrete function $\cos[\omega_0 t]$. Results of using this program are shown in Figures 3.53 and 3.54.

```
Print["Tests: HT of cosine = sine. HT of sine = -cosine."];
sinusoid = Table[Cos[2Pi*0.01*t], {t, 0, 1000, 0.01}];
FT = Fourier[sinusoid, FourierParameters -> {1, -1}];
L2 = Ceiling[Length[FT]/2];
Do[ FT[[i]] = FT[[i]]*E^(-I*Pi/2), {i, 1, L2}];
Do[ FT[[i]] = FT[[i]]*E^(+I*Pi/2), {i, L2 + 1, Length[FT]}];
HT = InverseFourier[FT, FourierParameters -> {1, -1}];
ListPlot[Re[HT], PlotJoined -> True];
```

The Do-statements listed above are incorporated into a subroutine called `hilbert` shown on page 162 for later applications in this section.

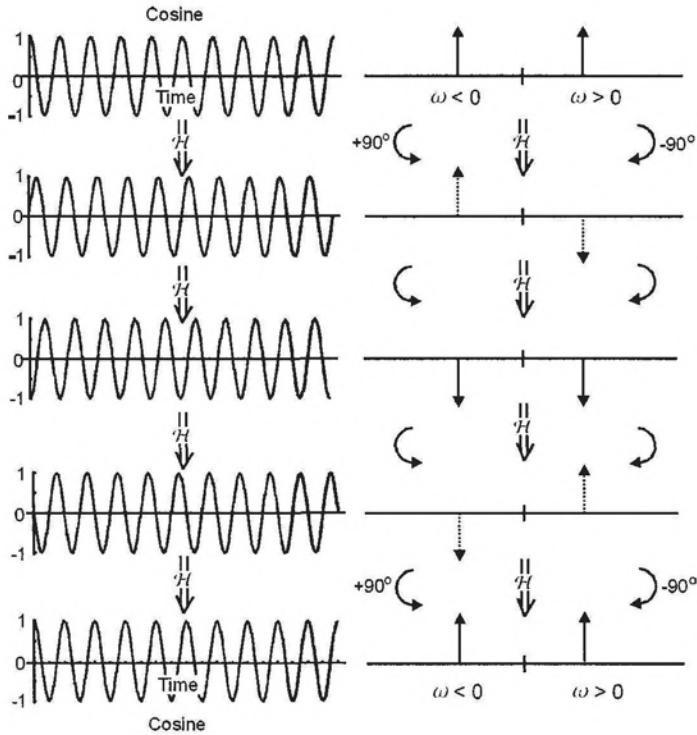


Figure 3.53: \mathcal{H} denotes Hilbert transformation. (a) Cosine wave. (b) Hilbert transform of cosine = +sine. (c) Hilbert transform of +sine = −cosine. (d) Hilbert transform of −cosine = −sine. (e) Hilbert transform of −sine = +cosine. For each transformation a phase angle of -90° has been added to the Fourier component for $\omega > 0$ and $+90^\circ$ for $\omega < 0$. (HilbertWhy90Degrees.nb,cdr → HTSinusoid.wmf)

Further discussion is divided as follows:

1. Fourier sign convention.
2. Properties of Euler functions
3. The “analytic signal”.
4. Mathematical definition of Hilbert transformation in the time and frequency domains. The “quadrature function.”
5. Hilbert transform of a seismic trace.
6. Phase spectrum of an unknown wavelet determined from its known amplitude spectrum.
7. Mathematical derivation of a Hilbert transform pair in the frequency domain using continuous functions.
8. Summary.

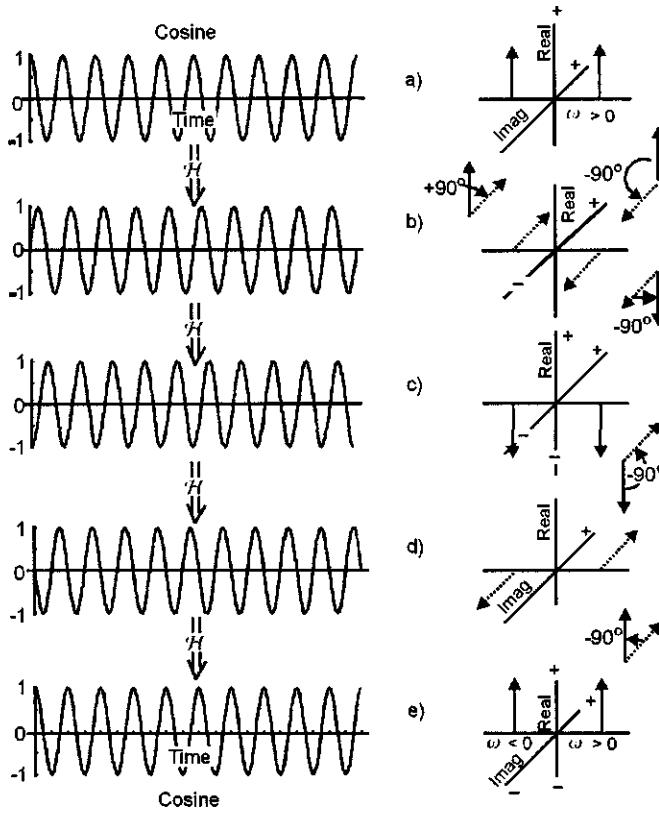


Figure 3.54: Same as Figure 3.53 but from a vector perspective. \mathcal{H} denotes Hilbert transformation. (a) Cosine wave. (b) Hilbert transform of cosine = +sine wave. (c) Hilbert transform of +sine = -cosine. (d) Hilbert transform of -cosine = -sine. (e) Hilbert transform of -sine = +cosine. For each transformation a phase angle of -90° has been added to the Fourier component for $\omega > 0$ and $+90^\circ$ for $\omega < 0$. (HilbertWhy90Degrees.nb → HilbertWhy90Degrees.cdr → HTVector.wmf)

Fourier sign convention

The sign convention for the exponent in a Fourier transform is restated here because of its importance in the examples that follow. The convention used in this volume is

$$\begin{aligned} H(\omega) &= \int_{-\infty}^{+\infty} h(t) e^{-i\omega t} dt \\ h(t) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(\omega) e^{+i\omega t} d\omega \end{aligned}$$

If you use a software package and are unsure of the sign convention used for Fourier transformation in the package then the *Mathematica* “calibration” program on page 53 can be used to confirm the sign of the imaginary exponent. Our convention corresponds to Fourier parameters $a = 1$ and $b = -1$ in *Mathematica* (Version 4.1).

Properties of Euler functions

The Fourier transform of a sinusoid contains both positive and negative frequencies, as shown in (3.80) and (3.81). That is,

$$\cos(\omega_0 t) \iff \pi \delta(\omega - \omega_0) + \pi \delta(\omega + \omega_0) \quad (3.80)$$

$$\sin(\omega_0 t) \iff i \pi \delta(\omega + \omega_0) - i \pi \delta(\omega - \omega_0) \quad (3.81)$$

In addition we have

$$e^{+i\omega_0 t} = \cos \omega_0 t + i \sin \omega_0 t \quad (3.82)$$

$$e^{-i\omega_0 t} = \cos \omega_0 t - i \sin \omega_0 t \quad (3.83)$$

Substituting (3.80) and (3.81) in the definition of $e^{+i\omega_0 t}$ we get

$$\begin{aligned} e^{+i\omega_0 t} &\iff \pi \delta(\omega - \omega_0) + \pi \delta(\omega + \omega_0) + \\ &\quad i [i \pi \delta(\omega + \omega_0) - i \pi \delta(\omega - \omega_0)] \\ &= \pi \delta(\omega - \omega_0) + \pi \delta(\omega + \omega_0) - \pi \delta(\omega + \omega_0) + \pi \delta(\omega - \omega_0) \\ &= 2 \pi \delta(\omega - \omega_0) \end{aligned} \quad (3.84)$$

$$e^{-i\omega_0 t} \iff 2 \pi \delta(\omega + \omega_0) \quad (3.85)$$

Equations (3.84) and (3.85) show that the Fourier transform of the complex exponential $e^{+i\omega_0 t}$ contains only the positive frequency ω_0 , and the Fourier transform of $e^{-i\omega_0 t}$ contains only the negative frequency $-\omega_0$.

The analytic signal

What is an analytic signal? The answer stems from looking at the seismic trace and wondering if it might be possible to extract from it an “instantaneous frequency” as a function of time, an “envelope” of the trace and an “instantaneous phase” as a function of time. We know that the frequency content of a seismic trace can change quickly with time (thinning beds, for example). It would therefore be useful to plot the frequency versus time as an aid to interpretation. This can be done by taking the Hilbert transform of the real trace and using this new trace as the imaginary part of a complex trace called the analytic signal. From this complex trace the “seismic attributes”, which include instantaneous frequency, can be determined.

Given any real function $f(t)$ then the “analytic signal” $a(t)$ is defined as

$$a(t) = f(t) + i \mathcal{H}[f(t)] \quad (3.86)$$

where $a(t)$ is complex and \mathcal{H} denotes Hilbert transformation.

A signal that has no negative-frequency components is an analytic signal. Thus, $e^{+i\omega_0 t}$ is an analytic signal as demonstrated by (3.84). $a(t)$ is the “analytic signal.” $\mathcal{H}[f(t)]$ is referred to as the “quadrature function.” The real part of the analytic signal is the original data and the imaginary part is its Hilbert transform. The imaginary part is the original data with a 90° phase shift. The imaginary part has the same amplitude and frequency content as the original data (the real part of $a(t)$). The phase spectrum depends on the phase of the real part. The discussion in this volume is for real, continuous or discrete causal sequences of time $f(t)$ or $f[n]$. It is also possible, however, to develop Hilbert transform relationships between complex functions.

Mathematical definition of Hilbert transformation in the time and frequency domains. The “quadrature function.”

We need to investigate the mathematical operators that will produce the desired results in the frequency and time domains. The discussion of Hilbert transforms below references discussions by Lee [108], Karl [93], Sheriff and Geldart [163], and Singh and Thompson [165] with our embellishments and *Mathematica* notebooks.

Recalling Equations (3.70) and (3.71) we have

$$\begin{aligned} h_e(t) &= \frac{1}{2} [h(t) + h(-t)] \\ h_o(t) &= \frac{1}{2} [h(t) - h(-t)] \end{aligned}$$

where $h_e(t)$ is an even function of time and $h_o(t)$ is an odd function of time, and for a causal function $h(t)$

$$h(t) = h_e(t) + h_o(t) \quad (3.87)$$

Multiplication in the time domain by `Sign[t]` corresponds to a convolution in the frequency domain so there must be some well-defined mathematical relationship in the frequency domain between the real and the imaginary parts of the Fourier transform of a causal function. This transform pair is shown in Figure 3.55. That is, we can convolve the real part of the Fourier transform of a causal function with some as yet unspecified function of frequency to obtain the imaginary part.

Following Sheriff and Geldart [163, p. 161], the desired transform pair can be determined analytically by taking the inverse Fourier transform of $-2i/\omega$:

$$\begin{aligned} \frac{-2i}{\omega} \iff \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{-2i}{\omega} e^{+i\omega t} d\omega &= \frac{-i}{\pi} \int_{-\infty}^{\infty} \left(\frac{\cos \omega t + i \sin \omega t}{\omega} \right) d\omega \\ &= \frac{2}{\pi} \int_0^{\infty} \frac{\sin \omega t}{\omega} d\omega \\ &= \text{sign}(t) \end{aligned}$$

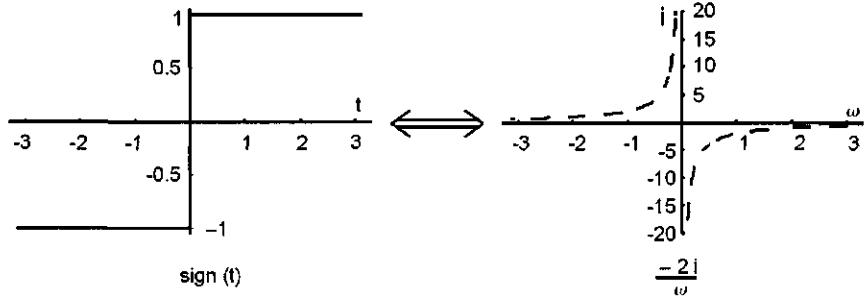


Figure 3.55: Left: $\text{sign}(t)$ function. Right: Fourier transform of $\text{sign}(t) = -2i/\omega$.
(FourierTransformEPiOver2.nb → HilbertTransforms.cdr → HilbertFreqConv1.wmf)

Note the “+” sign in the exponential. The final result is also conveniently provided by the *Mathematica* statement

$$\frac{2 \operatorname{Integrate}[\frac{\sin(\omega t)}{\omega}, \{\omega, 0, \infty\}]}{\pi}$$

or

$$\text{Sign}[t] = \text{InverseFourierTransform}[\frac{-2i}{\omega}, \omega, t, \text{FourierParameters} \rightarrow \{1, -1\}]$$

Therefore

$$\text{Sign}[t] \iff \frac{-2i}{\omega}$$

Multiplication in the time domain corresponds to convolution in the frequency domain. Therefore,

$$\begin{aligned} H_o(\omega) &= \int_{-\infty}^{\infty} H_e(\omega') \left(\frac{-2i}{\omega - \omega'} \right) d\omega' \\ &= H_e(\omega) * \frac{-2i}{\omega} \end{aligned} \quad (3.88)$$

Because of the singularity at $\omega = \omega'$ the Cauchy principal value (PV) must be taken wherever $\omega = \omega'$ so (3.88) is written

$$H_o(\omega) = \text{PV} \int_{-\infty}^{\infty} H_e(\omega') \left(\frac{-2i}{\omega - \omega'} \right) d\omega'$$

and then

$$h_o(t) \iff H_o(\omega)$$

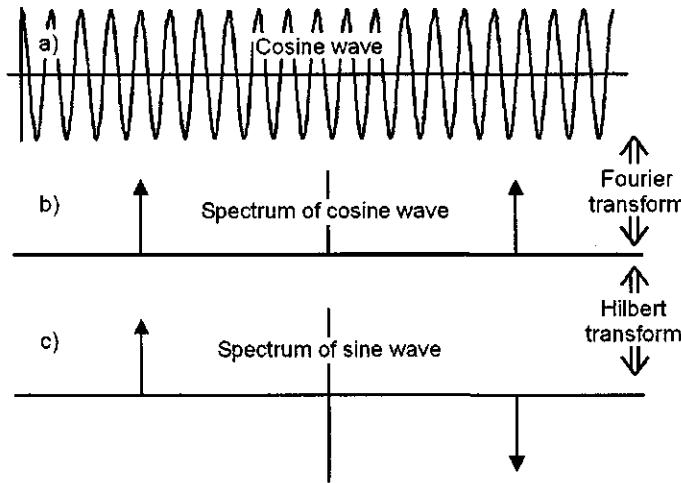


Figure 3.56: (a) Continuous cosine wave. (b) Fourier transform of (a). (c) Hilbert transform of (b) by multiplication of Fourier transform by $-i \text{sign}(\omega)$. (HilbertContinuousOddFromEven.nb, cdr, wmf)

$h_o(t)$ can also be determined from $H_e(\omega)$ by multiplication of the positive-frequency values of $H_e(\omega)$ by $-i$ because $e^{-i\pi/2} = -i$ and the negative-frequency values of $H_e(\omega)$ by $+i$ because $e^{+i\pi/2} = +i$ and then taking the inverse Fourier transform. That is,

$$h_o(t) \iff H_e(\omega) [-i \text{sign}(\omega)]$$

As a *Mathematica* example consider a continuous cosine wave of frequency 1 Hz as shown in Figure 3.56a. The real part of the Fourier transform of this is shown in Figure 3.56b. The imaginary part is zero. The Hilbert transform of the cosine wave is shown in Figure 3.56c and has been determined by multiplication of the real part of the spectrum of the cosine wave by $-i \text{sign}[\nu]$. The *Mathematica* statements used to obtain the Hilbert transform and generate the plots are

```
(* HilbertContinuousOddFromEven.nb *)
h = Cos[2 Pi 1 t];
Plot[h, {t, 0, 20}, PlotPoints -> 200]; (* Plot a *)
RealPart = Re[FourierTransform[h, t, Nu,
  FourierParameters -> {1, -1}]];
Plot[RealPart, {Nu, -10, 10}]; (* Plot b *)
Plot[-i*RealPart*Sign[Nu], {Nu, -10, 10}]; (* Plot c *)
```

The same result can be achieved by convolution of $1/\pi t$ with $h_e(t)$ because (Figure 3.57)

$$\frac{1}{\pi t} \iff -i \text{sign}(\omega) \quad (3.89)$$

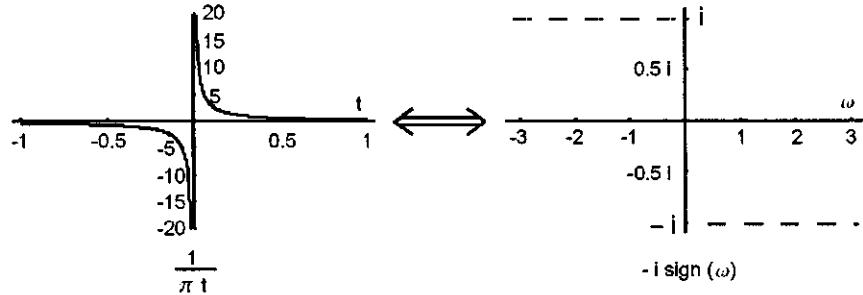


Figure 3.57: Left: $1/\pi t$ function. Right: Fourier transform of $1/\pi t$ is $-i \operatorname{sign}(\omega)$.
 (FourierTransformEPiOver2.nb → HilbertTransforms.cdr → HilbertFreqConv2.wmf)

That is,

$$\begin{aligned} h_o(t) &= h_e(t) * \frac{1}{\pi t} \\ &= \frac{1}{\pi} \operatorname{PV} \int_{-\infty}^{\infty} h_e(\tau) \left(\frac{1}{t-\tau} \right) d\tau \end{aligned} \quad (3.90)$$

where PV denotes the Cauchy principal value of the integral and $(*)$ denotes convolution.

Now assume a general complex time-domain function $a(t) = f(t) + i \mathcal{H}[f(t)]$. The imaginary part of $a(t)$ ($= \mathcal{H}[f(t)]$) is on the i -axis $\pm 90^\circ$ away from the real x -axis. The convolution $(*)$ of $1/(\pi t)$ with $f(t)$ gives the Hilbert transform of $f(t)$. That is,

$$a(t) = f(t) + i \mathcal{H}[f(t)] \quad (3.91)$$

$$= f(t) + i \frac{1}{\pi t} * f(t) \quad (3.92)$$

where $(*)$ denotes convolution.

The complex signal $a(t)$ (could be $= e^{i\omega t}$ for example) is called the “analytic signal” and contains only positive frequencies. To see this we can use the Fourier transform pair

$$\frac{1}{\pi t} \iff -i \operatorname{sign}(\omega) \quad (3.93)$$

where the *Mathematica* sign convention for the pair is shown in the following two statements:

```
FourierTransform[1/(Pi*t), t, w, FourierParameters -> {1, -1}]
InverseFourierTransform[-I*Sign[w], w, t, FourierParameters -> {1, -1}]
```

Repeating (3.92) for convenience and taking the Fourier transform of both sides we get

$$\begin{aligned} a(t) &= f(t) + i \mathcal{H}[f(t)] \\ &= f(t) + i \frac{1}{\pi t} * f(t) \\ A(\omega) &= F(\omega) + i [-i \operatorname{sign}(\omega)] F(\omega) \\ &= F(\omega) [1 + \operatorname{sign}(\omega)] \end{aligned} \quad (3.94)$$

where $f(t)$ is commonly the real seismic trace and $F(\omega)$ its Fourier transform. Thus, from (3.94) the complex analytic trace $A(\omega)$ is

$$A(\omega) = \begin{cases} 2F(\omega) & \text{if } \omega > 0 \\ 0 & \text{if } \omega < 0 \end{cases} \quad (3.95)$$

and we have therefore confirmed that the complex function $A(\omega)$ has only positive frequencies. Furthermore, in arriving at Equation (3.95) it is clear that no computer program (such as *hilbert*) designed specifically for Hilbert transformation is necessary; only a procedure that gives a conventional Fourier transform is required. Even subroutine *FT* described on page 47 will do the job. $A(\omega)$ is commonly referred to as the amplitude spectrum of the “analytic trace”.

From (3.95) the analytic trace $A(\omega)$ can be determined from $F(\omega)$ by zeroing out the negative frequencies in $F(\omega)$ and multiplying by 2. An example follows (also on the CD-ROM). This example should be compared with the much more efficient program *hilbertF* on page 530.

```
(* A from 2F *)
(* Define Hilbert transform subroutine *)
(* A from 2F *)
hilbert[RealTrace_]:=({lh=Length[RealTrace];If[OddQ[lh],lh2=Ceiling[lh/2],lh2=1h/2];
fd=Fourier[RealTrace,FourierParameters->{1,-1}];
Do[fd[[i]]=fd[[i]]*E^{(i*Pi/2)}, {i, lh2+1, lh}];
Do[fd[[i]]=fd[[i]]*E^{(-i*Pi/2)}, {i, 1, lh2}]];
QuadratureTrace=Re[Chop[InverseFourier[fd,FourierParameters->{1,-1}]]];
(* End of Hilbert subroutine *)

CausalFunction=Table[UnitStep[t]*Cos[t]*E^{-0.2*t},{t,-20,20}];
(* ListPlot[%,PlotRange->All,PlotJoined->True,AspectRatio->0.2]; *)
If[OddQ[Length[CausalFunction]],L=Ceiling[Length[CausalFunction]/2],
L=Length[CausalFunction]/2];
Print["CausalFunction:",RotateLeft[CausalFunction,L-1]];
Print["First value (t=0) of causal function = ",
Part[RotateLeft[CausalFunction,L-1],1]];
F=Fourier[RotateLeft[CausalFunction,L-1],FourierParameters->{1,-1}];
(* Now zero out negative parts in transform of CausalFunction *)
Do[F[[i]]=0,{i,L+1,Length[F]}];
F=2F; F[[1]]=F[[1]]/2; (* Don't count the zero-frequency twice *)
Print["2F(w):",F];
Clear[QuadratureTrace]; RotateLeft[CausalFunction,L-1];
hilbert[RotateLeft[CausalFunction,L-1]];
Print["Quadrature trace computed from Hilbert transform program: ",
QuadratureTrace];
(* Form analytic signal a(t) and get its Fourier transform A(w) *)
AnalyticTrace=RotateLeft[CausalFunction,L-1]+I*QuadratureTrace;
A=Chop[Fourier[AnalyticTrace,FourierParameters->{1,-1}]];
Print["A(w):",A];
```

In the above listing compare the values of $2 F(\omega)$ with $A(\omega)$ as defined by (3.95). Also examine the numerical values for the inverse Fourier transform of $A(\omega)$, which is the complex analytic signal $a(t)$ as defined by Equation (3.91). The real part of the $a(t)$ is the causal function $f(t)$; the imaginary part of $a(t)$ is the quadrature function $\mathcal{H}[f(t)]$.

Here is a short version of the above program without output that is easier to follow.

```
signal = Table[Sin[7*t], {t, 0, 20.65, 0.01}];  
FT = Fourier[signal, FourierParameters -> {1, -1}];  
L2 = Ceiling[Length[FT]/2];  
Do[FT[[i]] = 0, {i, L2 + 1, Length[FT]}];  
A = 2 FT;  
QuadratureFunction = Im[InverseFourier[A, FourierParameters -> {1, -1}]];
```

From Equation (3.93) Hilbert transformation in the time domain can be accomplished by convolution of the time-domain operator $q(t)$:

$$q(t) = \frac{1}{\pi t} \iff -i \operatorname{sign}(\omega) \quad (3.96)$$

with the function. Convolution of $q[n]$ with a sine wave using (3.96) is shown in Figure 3.58. The input sine wave is at the top, the Hilbert transformer is in the middle, and the output (a negative cosine wave) of the time-domain convolution is at the bottom. Figure 3.58 was generated using Equation (3.89).

The same result is obtained in the frequency domain by using the program `hilbert`. As shown in Figure 3.59, the Hilbert transform of a sine wave is a negative cosine wave.

Saha et al. [154] caution that because the Hilbert operator $1/\pi t$ approaches zero asymptotically and possibly only at long operator lengths then, even though the coefficients are small, convolution in the time domain or in the frequency domain might not be the best way to compute a Hilbert transform for geophysical applications. They conclude that a frequency domain approach is preferred for routine seismic data processing; that is, an approach other than a discrete form of $q(t)$.

Determination of the imaginary part $h_o(t)$ by inverse Fourier transformation of the frequency-domain product of a “quadrature filter” $Q(\omega)$ and the Fourier transform of the real part of an “analytic” signal.

By definition,

$$\begin{aligned} h(t) &\iff H(\omega) \\ h_e(t) &\iff H_e(\omega) \\ h_o(t) &\iff H_o(\omega) \end{aligned}$$

We deduced (review page 129) that for Hilbert transformation we must *subtract* 90° from the phase angles for positive frequencies and *add* 90° to the phase angles corresponding to negative frequencies. This sign convention comes from the one we chose to adopt in this volume for the b_n Fourier series terms as discussed in Section 3.3.1 on page 52 and used in Equation (3.10). In the frequency domain phase shifts of $\pm 90^\circ$ can be accomplished by multiplying $H(\omega)$ by i or $-i$ where

$$i = e^{+i\pi/2} \quad (3.97)$$

$$-i = e^{-i\pi/2} \quad (3.98)$$

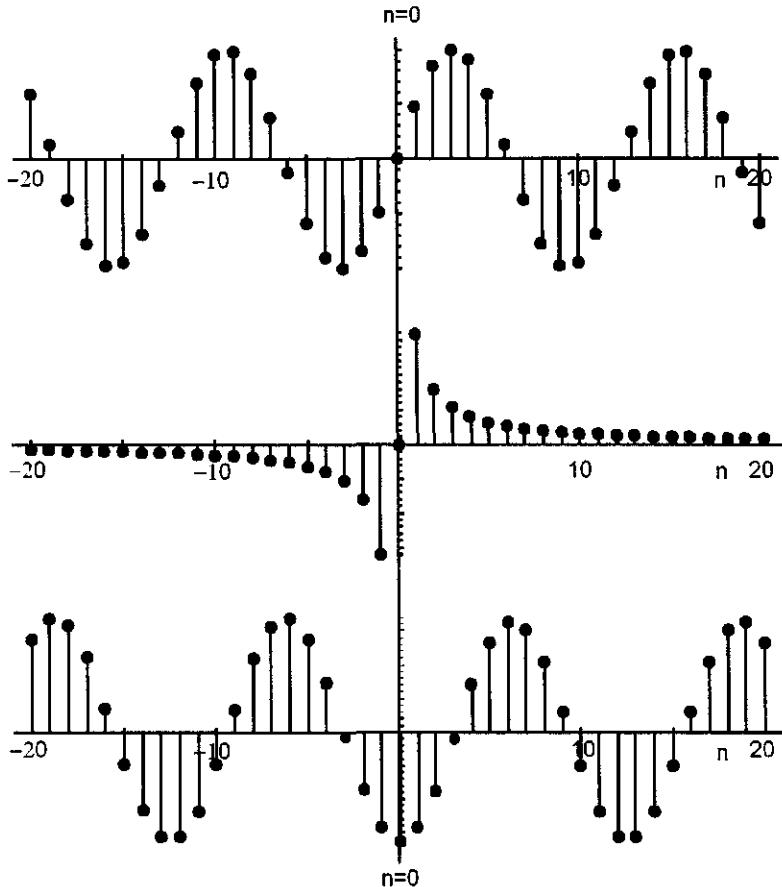


Figure 3.58: Top: Sine wave. Middle: Hilbert transformer of Equation (3.96), the inverse Fourier transform of $-i \operatorname{Sign}[\omega]$. Bottom: The convolution of the sine wave with the Hilbert transformer. Hilbert transform of sine wave = -cosine wave. (HilbertTransformerTimeDomain.nb → HilbertTransforms.cdr → HilbertTransformerTimeDomain.wmf)

Therefore, to obtain $h_o(t)$ from $h_e(t)$ multiply the spectrum of $H_e(\omega)$ by

$$-i \operatorname{sign}(\omega) = \begin{cases} +i & \text{if } \omega < 0 \\ -i & \text{if } \omega > 0 \end{cases} \quad (3.99)$$

where $\operatorname{sign}(x)$ is the **Sign** (pronounced “signum”) function defined as +1 for $x > 0$ and -1 for $x < 0$. Compare this with the equivalent qualitative discussion of Hilbert transformation at the start of this section on page 129.

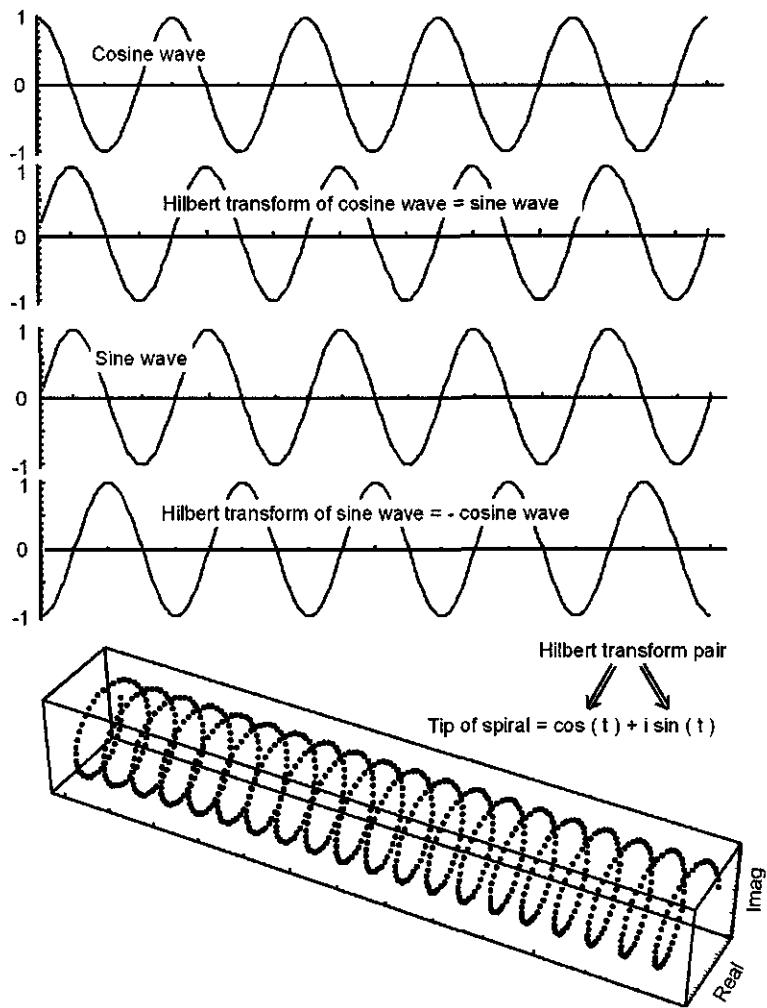


Figure 3.59: (a) Cosine wave. (b) Hilbert transform of cosine wave = sine wave. (c) Sine wave. (d) Hilbert transform of sine wave = $-\cosine$ wave. (e) The analytic signal is $\cos \omega t + i \sin \omega t$, which plots as the tip of a spiral of constant distance from the time axis; i.e., the modulus of the analytic signal is constant with time. (HTofSinusoid.nb → HilbertTransforms.cdr → HilbertTransformOfSinusoid.wmf)

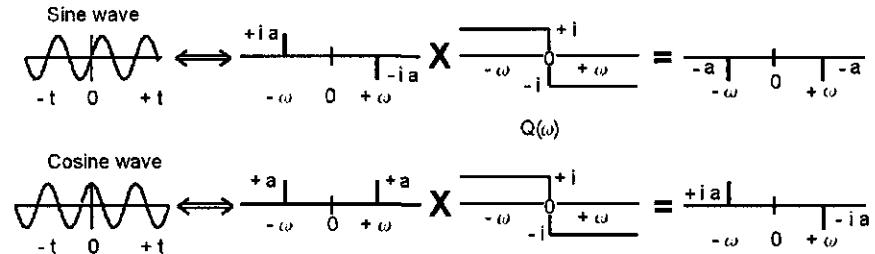


Figure 3.60: Illustration of quadrature filter of Equation (3.100). Top: Fourier transform of sine wave multiplied by $Q(\omega)$. Bottom: Fourier transform of cosine wave multiplied by $Q(\omega)$. Compare this figure with Figures 3.53 and 3.54. (QuadFilter.cdr → QuadFilter.wmf)

That is, the “quadrature filter” $Q(\omega)$ is

$$Q(\omega) = -i \operatorname{sign}(\omega) \quad (3.100)$$

The corresponding *Mathematica* statement is

```
Q=Table[-I*Sign[L-i],{i,1,Length[He]}];
```

where L is the number of positive frequencies including zero. Thus, the “quadrature trace” $h_o(t)$ is

$$h_o(t) \iff H_e(\omega) Q(\omega) = -i H_e(\omega) \operatorname{sign}(\omega) \quad (3.101)$$

$$= +i H_e(\omega) \operatorname{sign}(-\omega) \quad (3.102)$$

The complex “analytic trace” $h(t)$ is

$$h(t) \iff H_e(\omega) + i H_o(\omega)$$

Alternatively $h_o(t)$ can be determined using *Mathematica* subroutine *hilbert*, which has been referenced and listed on page 162. Both approaches are included in *Mathematica* program *HTofSinusoid.nb* on the CD-ROM that accompanies this volume. Application of Equation (3.100) is shown in Figure 3.60. Determination of the analytic signal and the associated “seismic attributes” is one of the useful applications of Hilbert transforms (page 242).

Hilbert transform of a seismic trace

Given a seismic trace we want its Hilbert transform. We don’t need to split the seismic trace into its even and odd parts. Instead, the Fourier transform of the trace is taken and we can work with the real and imaginary parts of the transform because

$$\begin{aligned} h_e(t) &\iff H_e(\omega) \\ h_o(t) &\iff H_o(\omega) \end{aligned}$$

With reference to Figure 3.62a the Fourier transform of the trace is shown in Figure 3.62b and c. The real and imaginary parts are denoted as H_e and H_o , respectively. Fourier components associated with H_e for the positive and negative frequencies are denoted as H_e^+ and H_e^- , respectively. Similar notation holds for H_o^+ and H_o^- . The Hilbert transform of the trace is obtained by application of Equations (3.97) and (3.98) to the Fourier transform of the trace, and then taking the inverse Fourier transform to get the quadrature trace. Note that the operation of taking the Hilbert transform of a time function can be viewed as taking the inverse Fourier transform of a Fourier transform so you end up in the starting domain. Before taking the inverse the phase angles are modified.

The straight lines on Figure 3.62 correspond to the following operations:

$$\begin{aligned} H_e^- \times e^{+i\pi/2} &= +iH_e^- = \text{negative-frequency imaginary part of quadrature trace} \\ H_e^+ \times e^{-i\pi/2} &= -iH_e^+ = \text{positive-frequency imaginary part of quadrature trace} \\ H_o^- \times e^{+i\pi/2} &= +iH_o^- = \text{negative-frequency real part of quadrature trace} \\ H_o^+ \times e^{-i\pi/2} &= -iH_o^+ = \text{positive-frequency real part of quadrature trace} \end{aligned}$$

In summary, it is seen that the real part of the spectrum of the quadrature trace is obtained from the imaginary part of the spectrum of the seismic trace, and the imaginary part of the spectrum of the quadrature trace is obtained from the real part of the spectrum of the seismic trace. An inverse Fourier transform of the spectrum shown in Figure 3.62 gives the time-domain quadrature trace, and the latter is the imaginary part of what we will refer to as the “complex seismic trace”. From this we can then derive the useful “seismic attributes” discussed in Section 5.8.

Using the above relations, the quadrature trace can be generated from the seismic trace $h[t]$ by the following steps as shown in Figure 3.62:

1. Take the Fourier transform $H[\omega]$ of the seismic trace $h[t]$.
2. Swap the real and imaginary parts of $H(\omega)$.
3. Multiply the real part of $H(\omega)$ by $+i$ just for the negative frequencies,
4. Multiply the imaginary part of $H(\omega)$ by $-i$ just for the positive frequencies,
5. Obtain the time-domain quadrature trace by taking the inverse Fourier transform of the modified $H(\omega)$.

Swapping the real and imaginary parts of the Fourier transform of the seismic trace and then multiplying by $\pm i$ as noted above is probably the fastest way to determine the quadrature trace.

An example of Hilbert transformation using the above steps on continuous data is shown in Figure 3.61. The *Mathematica* statements used to obtain the quadrature trace from the given function are shown on the figure.

Figure 3.62 is a graphic summary for discrete data of the following Hilbert transform *Mathematica* subroutine

```
hilbert[RealTrace_] := (lh = Length[RealTrace];
lh2 = Ceiling[lh/2];
fd = Fourier[RealTrace, FourierParameters -> {1, -1}];
Do[fd[[i]] = fd[[i]]*E^(+I*Pi/2), {i, lh2 + 1, lh}];
Do[fd[[i]] = fd[[i]]*E^(-I*Pi/2), {i, 1, lh2}];
QuadratureTrace = Re[Chop[InverseFourier[fd,
FourierParameters -> {1, -1}]]]);
```

The Fourier components that correspond to the positive frequency components of the real and imaginary parts of the Fourier transform of the seismic trace are each rotated by multiplication with $e^{-i\pi/2}$, as shown in Figure 3.62. The result yields the positive frequency components of the spectrum of the quadrature trace. The Fourier components that correspond to the negative frequency components of the real and imaginary parts of the Fourier transform of the seismic trace are each rotated by multiplication with $e^{+i\pi/2}$, as shown in Figure 3.62. The result yields the negative frequency components of the spectrum of the quadrature trace. Now, having the complete Fourier spectrum of the quadrature trace for both positive and negative frequencies one can take the inverse Fourier transform of the quadrature spectrum to obtain the time-domain quadrature trace. The complex seismic trace is

$$\text{complex seismic trace} = \text{seismic trace} + i \times \text{quadrature trace}$$

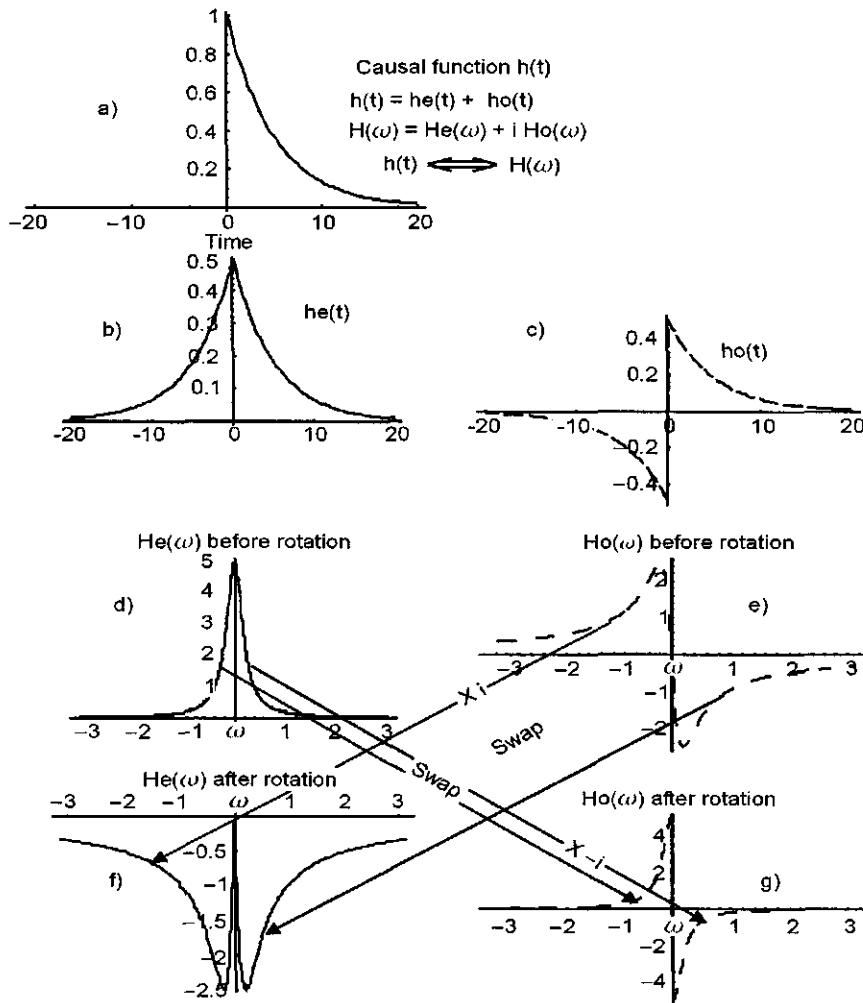


Figure 3.61: Top: (a) Causal input function $h(t)$. (b) and (c) are the real $h_e(t)$ and imaginary $h_o(t)$ parts of $h(t)$. $H(\omega)$ is the Fourier transform of the causal function. Lines from (d) and (e) to (f) and (g) show how the Hilbert transform can be obtained in a different but equivalent way than as described on page 129. The spectrum (f) and (g) of the quadrature trace can be obtained by swapping the real and imaginary parts of the Fourier transform of $h(t)$ and then multiplying the positive frequency components of the imaginary part of the swapped spectrum by $-i$ and the negative frequencies of the real part of the swapped spectrum by $+i$, as shown.

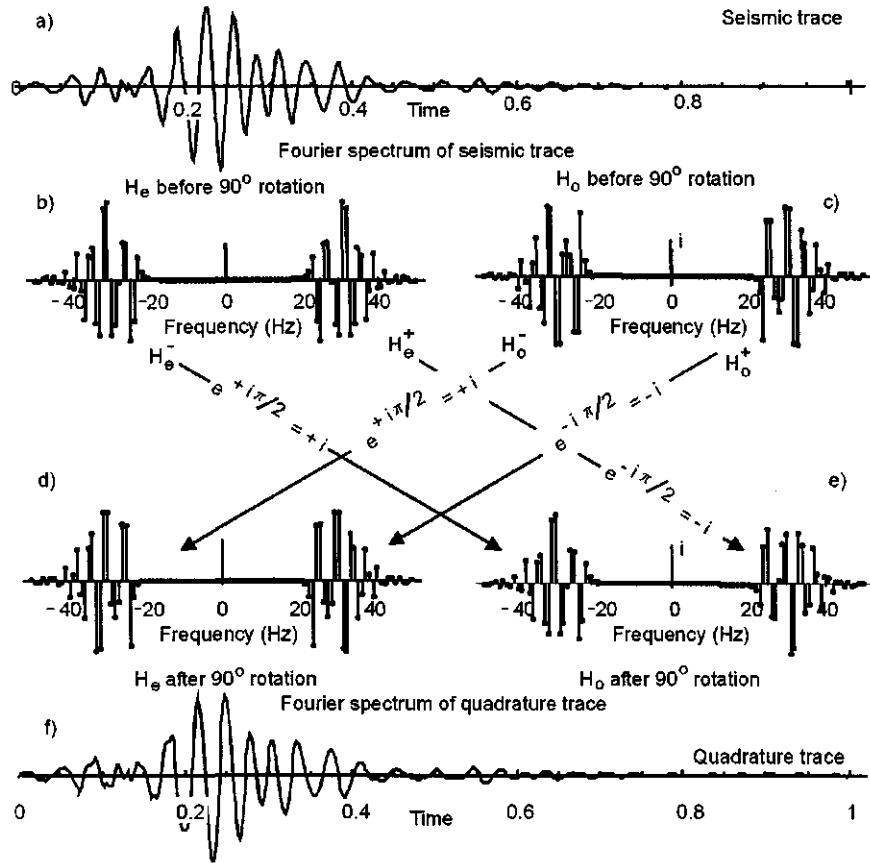


Figure 3.62: Top: (a) Seismic trace. (b) and (c) are real and imaginary parts of Fourier transform of trace. Lines from (b) and (c) to (d) and (e) show how Hilbert transform of trace can be obtained in a different but equivalent way than as described on page 129. The spectrum (d) and (e) of the quadrature trace (f) can be obtained by swapping the real and imaginary parts of the Fourier transform of the seismic trace and then multiplying the positive frequency components of the imaginary part of the swapped spectrum by $-i$ and the negative frequencies of the real part of the swapped spectrum by $+i$, as shown. That is, Fourier components for negative frequencies of (d) are obtained by multiplying the components (imaginary numbers) for the negative frequencies of (c) by $e^{+i\pi/2} = +i$, in accordance with Figure 3.60. So if a negative frequency component of (c) has a value ia then $ia \times i = -a$. Positive frequency components of (e) (imaginary numbers) are obtained by multiplying positive frequency components of (b) by $e^{-i\pi/2} = -i$, again as in Figure 3.60. So if a positive frequency component of (b) has a value a then $a \times -i = -ia$. On the other hand, if a negative frequency component of (b) has a value a then $a \times i = ia$, with no reversal in sign. Parts (d) and (e) are thus obtained by the phase shifts defined by Figure 3.60 and Equations (3.97) and (3.98). (HilbertTransformOfASeismicTraceAndSymmetryRelations.nb and HilbertSwapCheck.nb → HilbertTransforms.cdr → HilbertTraceSymmetry.wmf)

Minimum-phase spectrum of a wavelet determined from its known amplitude spectrum.

Assume that only the modulus $|A(\omega)|$ of the amplitude spectrum of a wavelet (or an impulse response) is known. For elaboration on this point refer to Section 8.5 on page 514. Its corresponding minimum-phase spectrum $\theta(\omega)$ is unknown. By definition

$$A(\omega) = |A(\omega)| e^{i\theta(\omega)}$$

Take the natural log of both sides:

$$\log A(\omega) = \log |A(\omega)| + i\theta(\omega) \quad (3.103)$$

For a minimum-phase wavelet, $\log |A(\omega)|$ and $\theta(\omega)$ are Hilbert transform pairs [132, 26]. According to Equation (3.86) if we take the Hilbert transform of the even part of $\log A(\omega)$, which is $\log |A(\omega)|$, we will obtain the spectrum $\theta(\omega)$ of the unknown causal function. Once $\theta(\omega)$ is known the complex function $A(\omega)$ can be constructed from (3.103) and the unknown minimum-phase wavelet can be determined by an inverse Fourier transform. Following the discussion on page 129, the Hilbert transform of $\log |A(\omega)|$ to find $\theta(\omega)$ guarantees that the wavelet will be causal. It will also be minimum-delay not because of the Hilbert transformation but because numerical evaluation of the multi-valued trigonometric functions involved in $\mathcal{H}(\log |A(\omega)|)$ gives the “principal value” for these trigonometric functions (if you use a calculator you get the same answer for $\cos 68^\circ$ as you do for $\cos [68^\circ + 360^\circ]$; i.e., we do not compute an “unwrapped” Fourier phase spectrum). The fact that the “principal value” of the trigonometric functions is what you get in the computations therefore guarantees that the wavelet will be “minimum-delay” (the computed phase angle spectrum will be closest to the real axis so it will be “minimum-phase”). Said another way, the phase spectrum obtained from a Fourier series is periodic; it cannot increase indefinitely with ω . See also Claerbout [40, p. 231]. White and O’Brien [193] concluded that a frequency-domain method using the discrete-time analog of a Hilbert transform was the most suitable method of computing minimum-phase wavelets from real seismic data.

Mathematical derivation of a Hilbert transform pair in the frequency domain using continuous functions.

Finally, we can formalize the qualitative discussion that we introduced on page 130 and investigate further what is happening with continuous functions in the frequency domain. Lee’s [108] analytical approach to the Hilbert transform is also an interesting application of Fourier transforms. First assume we are in the frequency domain and know only the function $H_o(\omega)$. We want to find $H_e(\omega)$ given $H_o(\omega)$; this derivation follows.

$H_e(\omega)$ from $H_o(\omega)$.

By definition

$$H(\omega) = \int_{-\infty}^{\infty} h(t) e^{-i\omega t} dt \quad (3.104)$$

and using (3.66) and (3.104), we can write

$$\begin{aligned} H(\omega) &= \int_{-\infty}^{\infty} [h_e(t) + h_o(t)] [\cos \omega t - i \sin \omega t] dt \\ &= \int_{-\infty}^{\infty} h_e(t) \cos \omega t dt - i \int_{-\infty}^{\infty} h_o(t) \sin \omega t dt \\ &= H_e(\omega) + i H_o(\omega) \end{aligned}$$

so that

$$H_e(\omega) = \int_{-\infty}^{\infty} h_e(t) \cos \omega t dt \quad (3.105)$$

$$H_o(\omega) = - \int_{-\infty}^{\infty} h_o(t) \sin \omega t dt \quad (3.106)$$

Equation (3.105) is an even function of ω and (3.106) is an odd function of ω . The inverse transforms of (3.105) and (3.106) are by definition

$$h_e(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H_e(\omega) \cos \omega t d\omega \quad (3.107)$$

$$h_o(t) = - \frac{1}{2\pi} \int_{-\infty}^{\infty} H_o(\omega) \sin \omega t d\omega \quad (3.108)$$

For our causal function $h(t)$,

$$h(t) = 0 \quad \text{for } t < 0 \quad (3.109)$$

and therefore (see Figure 3.47)

$$h_e(t) = h_o(t) \quad \text{for } t \geq 0 \quad (3.110)$$

Now, expressing (3.110) in terms of (3.107) and (3.108) we get

$$\int_{-\infty}^{\infty} H_e(\omega) \cos \omega t d\omega = - \int_{-\infty}^{\infty} H_o(\omega) \sin \omega t d\omega \quad \text{for } t \geq 0 \quad (3.111)$$

Equation (3.111) will hold for all values of t if $\sin \omega t$ on the right-hand side of (3.111) is replaced by $\sin \omega |t|$ to give

$$\int_{-\infty}^{\infty} H_e(\omega) \cos \omega t d\omega = - \int_{-\infty}^{\infty} H_o(\omega) \sin \omega |t| d\omega \quad (3.112)$$

which now holds for $-\infty < t < \infty$.

Now take the cosine transform of each side of (3.107):

$$\int_{-\infty}^{\infty} h_e(t) \cos \omega t dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \cos \omega t dt \int_{-\infty}^{\infty} H_e(u) \cos ut du \quad (3.113)$$

and after substituting (3.105) then (3.113) becomes

$$H_e(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \cos \omega t dt \int_{-\infty}^{\infty} H_e(u) \cos ut du \quad (3.114)$$

We want a relationship between $H_e(\omega)$ and $H_o(\omega)$. The right-hand integral of (3.114) is related to $H_o(\omega)$ by substituting (3.112) in (3.114) to obtain

$$H_e(\omega) = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \cos \omega t dt \int_{-\infty}^{\infty} H_o(u) \sin u|t| du \quad (3.115)$$

which can be written

$$\begin{aligned} H_e(\omega) &= -\frac{1}{2\pi} 2 \int_0^{\infty} \cos \omega t dt 2 \int_0^{\infty} H_o(u) \sin ut du \\ &= -\frac{4}{2\pi} \int_0^{\infty} \cos \omega t dt \int_0^{\infty} H_o(u) \sin ut du \\ &= -\frac{2}{\pi} \int_0^{\infty} \cos \omega t dt \int_0^{\infty} H_o(u) \sin ut du \end{aligned} \quad (3.116)$$

and this is the result we seek.

In other words, given $H_o(\omega)$ then $H_e(\omega)$ is the negative of the cosine transform of the inverse sine transform of $H_o(\omega)$. This is called a Hilbert transformation. Unlike a single Fourier transformation that transforms from the time domain to the frequency domain, the Hilbert transform—the cosine transform of the sine transform—ends up in the same domain. Equation (3.116) says that if you have the odd part $H_o(\omega)$ of the Fourier spectrum of a causal function then you are required to determine the even part $H_e(\omega)$ by (3.116). Now, having the entire spectrum $H_e + i H_o$, you can take the inverse Fourier transform and be assured of obtaining a causal result $h(t)$.

Equation (3.116) is perfectly general. There is no need to invoke the Cauchy principal value to solve for $H_e(\omega)$. Here is a *Mathematica* example that determines $H_e(\omega)$ from $H_o(\omega)$. To test (3.116) we need some function $H_o(\omega)$, which we can get by starting with a known causal function $h(t)$

$$\begin{aligned} h(t) &= u(t) e^{-\alpha t} \\ &= u(t) e^{-(2/10)t} \end{aligned}$$

where α is chosen to be $2/10$. For testing purposes $H_o(\omega)$ in (3.116) is found by starting with a known causal function $h(t)$ and determining the odd part $h_o(t)$ and its Fourier transform $H_o(\omega)$:

$$\begin{aligned} h_o(t) &= \frac{1}{2} [h(t) - h(-t)] \\ H_o(\omega) &\iff h_o(t) \end{aligned} \quad (3.71)$$

Now we want to determine $H_e(\omega)$ using the Hilbert transform equation (3.116) and given $H_o(\omega)$. Evaluate the outer integral in (3.116) noting that from (3.108)

$$h_o(t) = -\frac{2}{2\pi} \int_0^\infty H_o(\omega) \sin(\omega t) d\omega$$

(Compare this example with the much more efficient program *hilbertF* on page 530.) So the *Mathematica* statements are

$$\begin{aligned} h &= \text{UnitStep}[t] e^{-2/10 t}; \\ h_o &= \frac{h - (h/.t \rightarrow -t)}{2}; \\ H_o &= \text{FourierTransform}[h_o, t, w, \text{FourierParameters} \rightarrow \{1, -1\}] \\ &= \frac{5}{2(i - 5\omega)} - \frac{5}{2(i + 5\omega)} \end{aligned}$$

Normally we would not be given $h(t)$ and therefore neither $h_e(t)$ or $h_o(t)$. We are now ready to evaluate the outer integral in (3.116), which gives h_o .

$$\begin{aligned} h_o &= \frac{-2}{2\pi} \text{Integrate}[\text{Simplify}[H_o] \text{Sin}[\omega t], \{\omega, 0, \infty\}]; \\ &= \frac{1}{2} e^{-(t/5)} \text{Sign}[t] \text{Sign}[t] \end{aligned}$$

in agreement with Figure 3.47c. The inner integral of (3.116) is now

$$\begin{aligned} H_e(\omega) &= 2 \int_0^\infty h_o(t) \cos(\omega t) dt \\ \text{He} &= 2 \text{Integrate}[\text{Simplify}[H_e] \text{Cos}[\omega t], \{t, 0, \infty\}]; \\ H_e(\omega) &= \frac{5}{1 + 25\omega^2} \end{aligned} \tag{3.117}$$

and we have determined $H_e(\omega)$ from $H_o(\omega)$. As a check on this result the Fourier transform of $h(t)$ is shown in Figure 3.63.

$H_o(\omega)$ from $H_e(\omega)$

Equation (3.116) has an inverse. That is, given $H_e(\omega)$ it is possible to determine $H_o(\omega)$. It is this inverse that will be used to compute body wave dispersion and Q (page 518).

Take the sine transform of (3.108), which gives

$$\int_{-\infty}^\infty h_o(t) \sin \omega t dt = -\frac{1}{2\pi} \int_{-\infty}^\infty \sin \omega t dt \int_{-\infty}^\infty H_o(u) \sin u t du \tag{3.118}$$

and note that, by definition,

$$-\int_{-\infty}^\infty h_o(t) \sin \omega t dt = H_o(\omega) \tag{3.119}$$

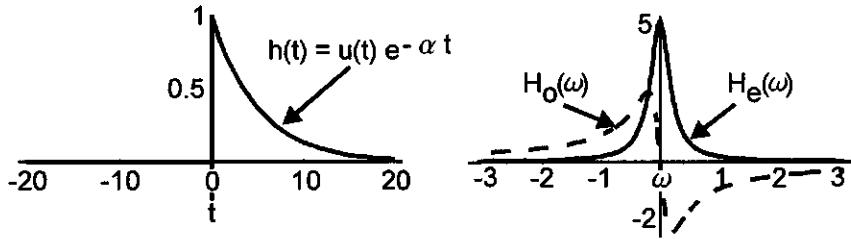


Figure 3.63: Left: Causal function $h(t)$ as defined in figure. Right: Fourier transform $H(\omega) = H_e(\omega) + i H_o(\omega)$ of $h(t)$. Dashed line is imaginary part $H_o(\omega)$.
(HeHoCheck.nb → HeHoCheck.cdr → HeHoCheck1.wmf)

Then (3.118) becomes

$$H_o(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega t dt \int_{-\infty}^{\infty} H_o(u) \sin u t du \quad (3.120)$$

We want $H_o(\omega)$ as a function of $H_e(\omega)$. If we replace t by $|t|$ in (3.120) the equation will still be true because the sign of $H_o(u)$ will not change, and if

$$H_o(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega t dt \int_{-\infty}^{\infty} H_o(u) \sin u t du$$

is true for $+t$ as well as $-t$, which it is, then

$$H_o(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega t dt \int_{-\infty}^{\infty} H_o(u) \sin u |t| du \quad (3.121)$$

will be true for $| -t |$. Thus, (3.121) can be written

$$H_o(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega |t| dt \int_{-\infty}^{\infty} H_o(u) \sin u |t| du \quad (3.122)$$

That is, for the right-hand side of (3.121)

$$\begin{aligned} & \frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega (-t) dt \int_{-\infty}^{\infty} H_o(u) \sin u (-t) du = \\ & \frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega t dt \int_{-\infty}^{\infty} H_o(u) \sin u t du \end{aligned} \quad (3.123)$$

because $\sin(-\omega t) = -\sin(\omega t)$.

Going back to (3.122), the right-hand integral of (3.122), which is

$$\int_{-\infty}^{\infty} H_o(u) \sin \omega |t| du \quad (3.124)$$

can be related to an integral involving $H_e(\omega)$ by reference to (3.112), which is repeated here:

$$\int_{-\infty}^{\infty} H_o(\omega) \sin \omega |t| d\omega = - \int_{-\infty}^{\infty} H_e(\omega) \cos \omega t d\omega$$

so (3.122) becomes

$$H_o(\omega) = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega |t| dt \int_{-\infty}^{\infty} H_e(u) \cos u t du \quad (3.125)$$

and this is the result we seek. Equation (3.125) is the one to be used in the determination of body wave velocity as a function of frequency when we discuss Q and its effect on the shape of the seismic wavelet on page 518.

Equation (3.125) is the companion equation to (3.116) and defines the required relationship between $H_e(\omega)$ and $H_o(\omega)$ for a causal function. That is, given $H_e(\omega)$ then $H_o(\omega)$ must be found using (3.125). This will allow us to determine the phase spectrum from the amplitude spectrum (page 156) where $H_e(\omega)$ is the known power spectrum and $H_o(\omega)$ is the unknown phase spectrum.

Here is a *Mathematica* example that determines $H_o(\omega)$ from $H_e(\omega)$. To test (3.125) we need some function $H_e(\omega)$. We get this by starting with the same causal function $h(t)$

$$\begin{aligned} h(t) &= u(t) e^{-\alpha t} \\ &= u(t) e^{-(2/10)t} \end{aligned}$$

where α is chosen to be 2/10. For testing purposes $H_e(\omega)$ in (3.125) is found by starting with a known causal function $h(t)$ and determining the even part $h_e(t)$ and its Fourier transform $H_e(\omega)$:

$$\begin{aligned} h_e(t) &= \frac{1}{2} [h(t) + h(-t)] \\ H_e(\omega) &\iff h_e(t) \end{aligned} \quad (3.71)$$

Now we want to determine $H_o(\omega)$ using (3.125) and given $H_e(\omega)$. Evaluate the outer integral in (3.125) noting that from (3.107)

$$h_e(t) = \frac{2}{2\pi} \int_0^{\infty} H_e(\omega) \cos(\omega t) d\omega$$

The *Mathematica* statements are

$$\begin{aligned} h &= \text{UnitStep}[t] e^{-2/10 t}; \\ he &= \frac{h + (h/.t \rightarrow -t)}{2}; \\ He &= \text{FourierTransform}[he, t, \omega, \text{FourierParameters} \rightarrow \{1, -1\}] \\ &= \frac{5 i}{2(i - 5\omega)} + \frac{5 i}{2(i + 5\omega)} = \frac{5}{1 + 25\omega^2} \end{aligned}$$

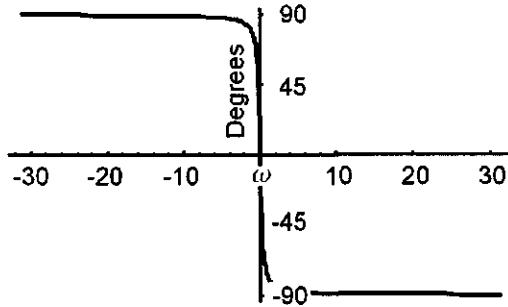


Figure 3.64: Plot is the $\tan^{-1}[H_o(\omega)/H_e(\omega)]$ where $H_e(\omega)$ and $H_o(\omega)$ are defined by Equations (3.117) and (3.126), respectively. (HilbertHoOverHe.nb → HeHoCheck.cdr → HeHoPhase.wmf)

Normally we would not be given $h(t)$ and therefore neither $h_e(t)$ or $h_o(t)$. We are now ready to evaluate the outer integral in (3.116) to give Figure 3.47b. Using *Mathematica* to obtain $h_e(t)$ we get

$$\begin{aligned} h_e &= \frac{2}{2\pi} \text{Integrate}[\text{Simplify}[He] \cos[\omega t], \{\omega, 0, \infty\}]; \\ &= \frac{1}{2} e^{-(t/5)} \text{Sign}[t] \end{aligned}$$

in agreement with Figure 3.47b. The inner integral of (3.125) is now

$$\begin{aligned} H_o(\omega) &= 2 \int_0^\infty h_e(t) \sin(\omega |t|) dt \\ Ho &= 2 \text{Integrate}[\text{Simplify}[he] \sin[\omega |t|], \{t, 0, \infty\}]; \\ H_o(\omega) &= \frac{-25\omega}{1 + 25\omega^2} \end{aligned} \quad (3.126)$$

and we have determined $H_o(\omega)$ from $H_e(\omega)$. As a check on this result the Fourier transform of $h(t)$ is shown in Figure 3.63.

Finally, the phase angle spectrum $\phi(\omega)$ is

$$\phi(\omega) = \tan^{-1} \left[\frac{H_o(\omega)}{H_e(\omega)} \right]$$

and this is shown in Figure 3.64.

A program for Hilbert transformation of a discrete time-domain function is:

```
hilbert[RealTrace_] := (lh = Length[RealTrace];
If[OddQ[lh], lh2 = Ceiling[lh/2], lh2 = lh/2];
fd = Fourier[RealTrace, FourierParameters -> {1, -1}];
Do[fd[[i]] = fd[[i]]*E^(+I*Pi/2), {i, lh2 + 1, lh}];
Do[fd[[i]] = fd[[i]]*E^(-I*Pi/2), {i, 1, lh2}];
QuadratureTrace = Re[Chop[InverseFourier[fd, FourierParameters -> {1, -1}]]]);
```

3.6 z-transform

From Equation (3.40),

$$\begin{aligned} F(\nu) &= \int_{-\infty}^{+\infty} f(t)e^{-i2\pi\nu t} dt \\ F(\omega) &= \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt \end{aligned} \quad (3.127)$$

As written, $f(t)$ is a continuous-time function. If this function is sampled at discrete time intervals Δt then it can be written as [166, 153]

$$f(t) = \sum_{n=-\infty}^{n=+\infty} f_n \delta(t - n\Delta t) \quad (3.128)$$

where n is integer. Thus, (3.127) can be written

$$F(\omega) = \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{n=+\infty} f_n \delta(t - n\Delta t) e^{-i\omega t} dt$$

This integral reduces to the sum

$$F(\omega) = \sum_{n=-\infty}^{n=+\infty} f_n \int_{-\infty}^{+\infty} \delta(t - n\Delta t) e^{-i\omega t} dt \quad (3.129)$$

$$= \sum_{n=-\infty}^{n=+\infty} f_n e^{-i\omega n\Delta t} \quad (3.130)$$

If $f(t)$ is written according to Equation (3.128), then integrals associated with continuous-time (aperiodic) theory can be used for functions that have been sampled at discrete time intervals. Thus,

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i2\pi\nu t} dt = \sum_{n=-\infty}^{n=+\infty} f_n e^{-i2\pi\nu n} \quad (3.131)$$

for $\Delta t = 1$ time unit and $n = \dots - 2, -1, 0, 1, 2, \dots$

Now make the substitution

$$z = e^{-i\omega} = e^{-i2\pi\nu} \quad (3.132)$$

Note the negative sign in the exponent. We choose this instead of

$$z = e^{+i\omega} = e^{+i2\pi\nu}$$

as used by electrical engineers. The choice of a negative sign is directly related to Equation (3.10) as discussed in Section 3.3.1. This difference in the

definition of z might or might not be important, depending on the physical application; however, one must be aware of which convention is used.

The dimensions of the exponent are correct in Equation (3.132) if we assume that Δt is *one time unit* and that, therefore, the dimensions of ν are *cycles per time unit*. After this substitution, Equation (3.131) becomes

$$F(z) = \sum_{n=-\infty}^{n=+\infty} f_n z^n \quad (3.133)$$

$F(z)$ is called the *z -transform* of $f(t)$. Again, widely-referenced engineering texts such as Oppenheim [129, p. 154] would write the z -transform as

$$F(z) = \sum_{n=-\infty}^{n=+\infty} f_n z^{-n}$$

The complex variable z

$$z = e^{-i\omega} = e^{-i2\pi\nu}$$

plots as a *unit circle* (Figure 3.65) with the origin at $z = 0$ as various values of ν are inserted and z is plotted in the complex z -plane. That is, the magnitude of

$$e^{-i\omega}$$

is 1 for any value of ω .

Thus, the z -transform of the function $f(t) = f_{-m}, \dots, f_{-3}, f_{-2}, f_{-1}, f_0, f_1, f_2, \dots, f_n$ is defined as

$$\begin{aligned} F(z) = & f_{-m} z^{-m} + \dots + f_{-3} z^{-3} + f_{-2} z^{-2} + f_{-1} z^{-1} \\ & + f_0 + f_1 z + f_2 z^2 + f_3 z^3 + \dots + f_n z^n \end{aligned} \quad (3.134)$$

where z is called an *indeterminate*. Many of the practical applications of z -transforms that involve multiplication and/or division of polynomials do not require that z take on a particular value; z never needs to be determined.

The z -transform is essentially the discrete equivalent of the Laplace transform. In general, z is a complex number and $F(z)$ is a complex function in the complex z -plane. If z has the value

$$z = e^{-i\omega}$$

then z can be thought of as a vector whose tip is confined to a circle of unit radius, as shown in Figure 3.65. The tip will spin around the circle depending on the value of ω and t ($t = n\Delta t$), but the modulus of z will always be on the unit circle.

Substituting:

$$z = e^{-i\omega}$$

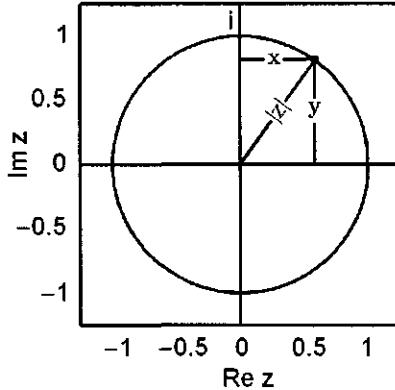


Figure 3.65: The unit circle in the z -plane. (ma → cdr → unitcir.wmf)

into Equation (3.134), and assuming times ≥ 0 , we have

$$\begin{aligned} F(\omega) &= f_0 + f_1 e^{-i\omega} + f_2 e^{-i2\omega} + f_3 e^{-i3\omega} + \cdots + f_n e^{-in\omega} \\ &= \sum_n f_n e^{-in\omega} \end{aligned}$$

which is the discrete representation of the Fourier integral

$$F(\omega) = \int_0^\infty f(t) e^{-i\omega t} dt$$

The Fourier transform of a function $f(t)$ is identical to the z -transform of the function when the z -transform is evaluated on the unit circle. The magnitude of

$$z = e^{-i\omega}$$

is unity whatever the value of ω .

We will discuss below the significance of values of $z = x+iy$ that plot *outside* of the unit circle, *inside* the unit circle, and *on* the unit circle. For now, we note only that the Fourier transform is simply a z -transform where z has been assigned values along the unit circle in the complex z -plane.

3.6.1 Factors of a finite, discrete function

Any polynomial $P(z)$ can be factored into a product of binomials. For example the polynomial

$$P(z) = 6 + 11z + 4z^2$$

where (6, 11, 4) are the (pure real) time-domain coefficients and the polynomial can be factored into

$$P(z) = (2 + z)(3 + 4z)$$

Each binomial factor can be thought of as a two-term polynomial (in this volume often referred to as a couplet). The term $(2 + z)$ is the z -transform of the time-domain couplet (2,1) and $(3 + 4z)$ is the z -transform of the couplet (3,4).

If the modulus of the first coefficient of a couplet is larger than the modulus of the second coefficient, the couplet is said to be *minimum-delay* (Robinson [147]). If the modulus of the second coefficient of a couplet is larger than the modulus of the first, the couplet is said to be *maximum-delay*. The terms refer to the onset of energy of the couplet; for a minimum-delay couplet (wavelet), the energy is delayed the least; most of it comes in with the first coefficient. For a maximum-delay couplet, most of the energy comes in with the second coefficient. The time-reverse of a minimum-delay wavelet is thus a maximum-delay wavelet.

The corresponding behavior in the frequency domain is described by the properties of the *phase* spectra of the couplets. Thus

$$\text{minimum-delay} \iff \text{minimum-phase}$$

so that minimum-delay and minimum-phase are terms that refer to time-domain and frequency-domain properties, respectively, so

$$\text{maximum delay} \iff \text{maximum phase}$$

For the couplet (2, 1), the first coefficient is larger in magnitude than the second, i.e., the energy appears early in the function. It is therefore *minimum-delay*. For the term (3, 4), the second coefficient is larger in magnitude than the first, i.e., the energy arrives later in the function. It is therefore *maximum delay*.

If either or both of the coefficients is complex, then the modulus of the coefficient is used to determine the delay properties of the couplet.

The term “minimum-phase” refers to the proximity of the entire phase angle spectrum $\theta(\omega)$ to the $\theta = 0$ axis in a plot of θ versus ω . For a minimum-phase couplet, $\theta(\omega)$ is as close as possible to the $\theta = 0$ axis for the given numerical values of the time-domain couplet. If this couplet were used as a filter and convolved with an arbitrary function, then the function would be shifted in time by an amount

$$\Delta t = \frac{\Delta\theta}{\Delta\nu}$$

where Δt is the slope of the curve θ versus ν . If the slope is constant, then θ versus ν is a straight line and the arbitrary function is shifted *without distortion* by an amount Δt . If the slope of $\theta(\nu)$ is zero, then the function is filtered, but not shifted at all. This follows from the time-shifting theorem.

If the coefficients of the two-term couplet are reversed, then the minimum-delay couplet becomes a maximum-delay couplet.

The phase spectrum, $\theta(\nu)$ versus ν , of the maximum delay couplet is now as far away as it can get from the $\theta = 0$ axis.

The concept of minimum- and maximum-delay is easily extended to functions having more than two terms. Given any function, construct the z -transform of the function and factor the z -transform polynomial into binomial factors (couplets). If each binomial in the factored polynomial is minimum-delay, then the function is minimum-delay. If each binomial in the factored polynomial is maximum delay, then the function is maximum delay. If there is a mixture of minimum and maximum delay binomials, then the function is called *mixed delay*.

For the polynomial

$$P(z) = 6 + 11z + 4z^2 = (2 + z)(3 + 4z) \quad (3.135)$$

The couplet $(2, 1)$ is minimum delay, and the couplet $(3, 4)$ is maximum delay. The time domain function $(6, 11, 4)$ is therefore mixed delay.

Inspection of Equation (3.135) shows that $P(z) = 0$ for $z = z_1 = -2$ and $z = z_2 = -3/4$. z_1 and z_2 are therefore the roots (zeros) of the polynomial.

The maximum-delay couplet $(3, 4)$ has for its z -transform $P_2(z)$:

$$P_2(z) = 3 + 4z$$

Dividing both sides by 4, we have

$$\begin{aligned} \frac{P_2(z)}{4} &= \frac{3}{4} + \frac{4}{4}z \\ &= \frac{3}{4} + z \end{aligned}$$

Equation (3.135) can therefore also be written

$$P(z) = 6 + 11z + 4z^2 = 4(-z_1 + z)(-z_2 + z) \quad (3.136)$$

where $z_1 = -2$ and $z_2 = -3/4$. In this form, it is clear that for $z = z_1$ or $z = z_2$, $P(z) = 0$.

If $P(z)$ is written in the form of Equation (3.136) then the z -transform pair for the first couplet is

$$\begin{aligned} (-z_1, 1) &\iff (-z_1 + z) \\ (2, 1) &\iff (2 + z) \end{aligned}$$

and for the second couplet,

$$\begin{aligned} (-z_2, 1) &\iff (-z_2 + z) \\ (3/4, 1) &\iff (3/4 + z) \end{aligned}$$

Thus, the z -transform of a finite discrete function $(a_0, a_1, a_2, \dots, a_m)$ can be factored into the form

$$\begin{aligned} A(z) &= a_0z^0 + a_1z^1 + a_2z^2 + a_3z^3 + \cdots + a_mz^m \\ &= a_m(z - z_1)(z - z_2)(z - z_3)\cdots(z - z_m) \\ &= a_m(-z_1 + z)(-z_2 + z)(-z_3 + z)\cdots(-z_m + z) \end{aligned}$$

Table 3.6: Examples of minimum, maximum, and mixed delay functions and their z -transforms. The couplets are $(-z_1, 1)$ and $(-z_2, 1)$

couplet	z -transform	Factors	Root(s)	Delay
(2, 1)	$2 + z$	(2, 1)	-2	minimum
(1, 2)	$1 + 2z$	(1, 2)	-1/2	maximum
(i, 0.5)	$i + 0.5z$	(i, 0.5)	-2i	minimum
(0.5, -i)	$0.5 - iz$	(0.5, -i)	-i/2	maximum
(2,15,34,24)	$2 + 15z + 34z^2 + 24z^3$	(1,2),(2,3),(1,4)	-1/2, -2/3, -1/4	maximum
(24,34,15,2)	$24 + 34z + 15z^2 + 2z^3$	(2,1),(3,2),(4,1)	-2, -3/2, -4	minimum
(12,35,24,4)	$12 + 35z + 24z^2 + 4z^3$	(1,2),(3,2),(4,1)	-1/2, -3/2, -4	mixed

where z_i , ($i = 1, 2, \dots, m$) are the zeros (roots) of the polynomial $A(z)$, and we have rearranged the binomial factors so that they are of the form $(-z_i + z)$ instead of $(z - z_i)$ emphasizing that each of these factors is just a polynomial of degree one (a couplet).

We have also normalized the magnitudes of the two couplet coefficients so that the second is always unity.

All or some of the roots might be real, imaginary, or complex. Examples are shown in Table 3.6.

3.6.2 Phase of minimum and maximum delay couples

In the time domain we described the properties of doublets as either minimum- or maximum-delay. In the frequency domain we can describe their properties by using the z -transform. Consider the z -transform of the couplet $(1, -1/z_0)$:

$$1 - \frac{z}{z_0} \quad \text{where } z_0 = r e^{i\omega_0} \quad \text{and } z = e^{-i\omega}$$

Both z and z_0 are in general complex numbers. The units of ω are *radians per time unit*. The z -transform $D(z)$ of the couplet $(1, -1/z_0)$ is

$$\begin{aligned} D(z) &= 1 - \frac{z}{z_0} \\ &= 1 - \frac{z}{r e^{i\omega_0}} \\ &= 1 - r^{-1} e^{-i\omega_0} z \\ &= 1 - r^{-1} e^{-i\omega} e^{-i\omega_0} \\ &= 1 - r^{-1} e^{-i(\omega + \omega_0)} \\ &= [1 - r^{-1} \cos(\omega + \omega_0)] - i [r^{-1} \sin(\omega - \omega_0)] \end{aligned}$$

where r is a real number.

For $r > 1$, $D(z)$ will be minimum-phase. This means that its phase spectrum will plot closer to zero phase compared with values of $D(z)$ for $r < 1$. This behavior for $r > 1$ is shown in Figure 3.66.

For $r < 1$, $G(z)$ will be maximum phase. This means that its phase spectrum will always increase without bound. *The phase spectrum of a sampled (digitized) maximum-delay function increases without bound. It is not periodic.* This can be seen on the polar plot of Figure 3.69.

The case for which $r > 1$ and for which

$$\phi(\omega) = \phi(\omega + 2\pi)$$

is called minimum-phase or, in the time domain, minimum-delay. When $r < 1$ the circular path encloses the origin. The phase curve is no longer periodic.

Signals $a_1, a_2, a_3, \dots, a_j$, $j = 1, m$ can have the same Fourier magnitude spectrum but different shapes in the time domain. But if a_3 , say, is a minimum-delay signal then

$$\sum_{i=1}^n |a_3[i]|^2 \geq \sum_{i=1}^n |a_j[i]|^2, \quad n = 1, 2, 3, \dots \text{ for any } j$$

That is, a minimum-delay wavelet will have its energy concentrated toward the beginning of the wavelet, but this does not mean that the first coefficient of the wavelet is the largest.

3.6.3 Amplitude and phase of a z-transform

Let z_1 be a root in the z -plane (Figure 3.66). Define z_1 as

$$z_1 = |z_1| e^{i\theta}$$

where $|z_1| = 1.3$ and $\theta = 135^\circ$. z_1 is the root of the z -transform of a two-term polynomial (a binomial) whose time-domain coefficients are

$$(-z_1, 1)$$

That is, the z -transform of this binomial is

$$\begin{aligned} P(z) &= -z_1 z^0 + 1 z^1 \\ &= -z_1 + z \end{aligned}$$

The frequency response (amplitude and phase spectrum) of this 2-term wavelet is obtained by evaluating its z -transform on the unit circle; i.e.,

$$\begin{aligned} P(z) &= -z_1 + z \\ P(\omega) &= -z_1 + e^{-i\omega} \\ P(\nu) &= -z_1 + e^{-i2\pi\nu\Delta t} \end{aligned} \tag{3.137}$$

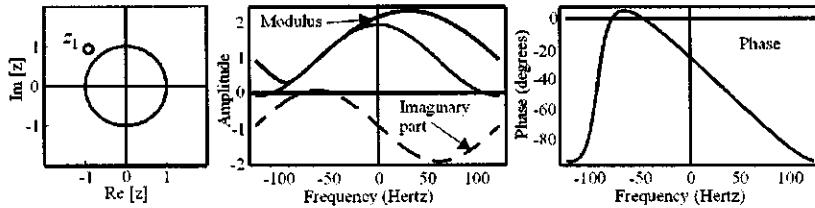


Figure 3.66: Single root z_1 and frequency response of the z -transform $P(z) = -z_1 + z$. Circle (left) shows location of the complex root z_1 . Modulus $|z_1| = 1.3$, $\theta = 135$ deg where $z_1 = |z_1| e^{i\theta}$. This is the root of the z -transform of a (complex) time-domain binomial $(-z_1, 1)$. The z -transform of this binomial is $P(z) = -z_1 z^0 + 1 z^1 = -z_1 + z$. The 2-term polynomial is minimum-delay [147] because the modulus of the first coefficient of the wavelet is larger than the second (the root is outside the unit circle). (Unitc1AndResp1AndPhsmin1.nb,cdr → Unitc1A.wmf)

As ν takes on values from

$$\frac{-1}{2\Delta t} < \nu < \frac{+1}{2\Delta t}$$

z completes one revolution of the unit circle in the z -plane. For $\Delta t = 4$ ms, the folding frequency is 125 Hertz. Therefore, the (complex) values of $P(\nu)$ define the frequency response of the two-point time-domain wavelet. The amplitude spectrum is shown in Figure 3.66 (center figure) for the frequency response defined by Equation (3.137). The time-domain binomial is complex; we therefore do not expect the same symmetry in the frequency domain that is found for a pure real time-domain function.

The z -transform $P(z)$ of the binomial $(-z_1, 1)$ is

$$P(z) = -z_1 z^0 + 1 z^1 = -z_1 + z$$

This is a minimum-delay 2-term polynomial because the first coefficient in the (complex) time-domain binomial is larger than the second. The phase response of $(-z_1, 1)$ is shown in Figure 3.66 (right). In the time domain we described the properties of doublets as either minimum- or maximum-delay. In the frequency domain we can describe their properties by using the z -transform.

Consider the z -transform $P(z)$ of the binomial $(-z_1, 1)$:

$$P(z) = -z^1 + z$$

where

$$z_1 = r e^{i\omega_1}$$

and

$$z = e^{-i\omega}$$

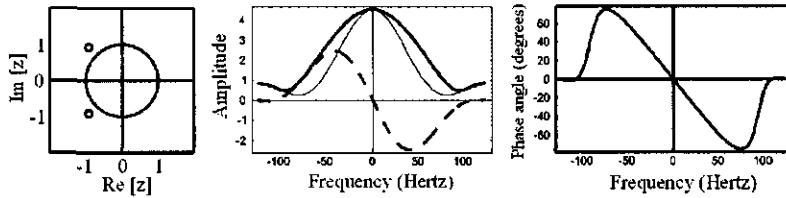


Figure 3.67: Complex conjugate roots plotted in the z -plane (left). Frequency response (center); dashed line is imaginary part. Phase angle spectrum (right). (unitc3.nb → unitc123.cdr → unitc123.wmf)

and r is some real number > 0 . Both z and z_1 are, in general, complex numbers. The units of ω are *radians per time unit* Δt .

For $|z_1| > 1$ $P(z)$ will be minimum phase. This means that its phase spectrum will plot closer to zero phase compared with values of $P(z)$ for $|z_1| < 1$. This behavior for $|z_1| > 1$ is shown in Figure 3.67. The concept of minimum-phase is due to Bode [23, 150]. For $|z_1| < 1$ $P(z)$ will be maximum phase. This means that its phase spectrum will increase without bound. *The phase spectrum of a sampled maximum-delay function increases without bound. It is not periodic.* This can be seen on the polar plot of Figure 3.68.

The case for which $|z_1| > 1$ and for which

$$\phi(\omega) = \phi(\omega + 2\pi)$$

is called *minimum-phase* or, in the time domain, *minimum-delay*. The concept of minimum delay is identical to the concept of minimum phase, except that “minimum delay” refers to the time-domain, whereas the term “minimum phase” refers to the frequency domain. When $|z_1| < 1$, then the circular path encloses the origin. The phase curve is no longer periodic.

The minimum-delay phase spectrum is *periodic*. The convolution of two binomials, one of which is the complex conjugate of the other, is shown in Figure 3.67. The modulus $|z_0| = 1.3$. $\theta = 135$ deg where $z_1 = |z_1| e^{+i\theta}$ and $z_2 = |z_2| e^{-i\theta}$. These are the roots of the z -transform of the (real) time-domain polynomial, which can be obtained by convolving the (complex time) binomial $(-z_1, 1)$ with the (complex time) binomial $(-z_1^*, 1)$. The result of the complex convolution is the real polynomial $(1.690, 1.838, 1)$. The z -transform of this polynomial is $P(z) = 1.690z^0 + 1.838z^1 + z^2 = 1.690 + 1.838z + z^2$. The roots (zeros) of this polynomial are z_1 and z_2 . *Although this is a minimum-delay 3-term polynomial the first coefficient in the time-domain polynomial (wavelet) is not the largest.*

The frequency response of the product of these two z -transform binomials is shown in the center diagram of Figure 3.67. This is the frequency response of the polynomial described by the roots of Figure 3.67. Modulus $|z_0| = 1.3$; $\theta = 135$ deg where $z_1 = |z_1| e^{+i\theta}$ and $z_2 = |z_2| e^{-i\theta}$. Note that the real part of $P(z)$ is even, and the imaginary part is odd. The z -transform of this polynomial

is

$$\begin{aligned} P(z) &= 1.690z^0 + 1.838z^1 + 1z^2 \\ &= 1.690 + 1.838z + z^2 \end{aligned}$$

The roots (zeros) of this polynomial are z_1 and z_2 . Although this is a minimum-delay 3-term polynomial the first coefficient in the time-domain polynomial (wavelet) is not the largest. The minimum-delay phase spectrum is *periodic*.

The complex root $z_1 e^{i\theta} = 0.7e^{i2.356}$ (phase angle of 135 deg) is shown in Figure 3.67. This is the root of the z -transform of the (complex) time-domain binomial $(-z_1, 1)$. The z -transform of this binomial is $P(z) = -z_1 z^0 + 1z^1 = -z_1 + z$. The root (zero) of this binomial is z_1 . This is a maximum-delay 2-term polynomial because the modulus of the first coefficient in the time-domain binomial (wavelet) is smaller than the second (the root is inside the unit circle).

The frequency response of this maximum-phase z -transform binomial is shown in Figure 3.68 (center). Modulus $|z_0| = 0.7$; $\theta = 135$ deg where $z_1 = |z_1| e^{+i\theta}$ and $z_2 = |z_2| -i\theta$.

The z -transform of the time-domain binomial $(-z_1, 1)$ is $P(z) = -z_1 z^0 + 1z^1 = -z_1 + z$. This is a maximum-delay 2-term polynomial because the first coefficient in the (complex) time-domain binomial (wavelet) is less than the second. Note that the real part of $P(z)$ is even, but the imaginary part is not odd. The root (zero) of $P(z)$ is z_1 . The phase response of the maximum-delay 2-term binomial is shown in Figure 3.68 (right). This is the (wrapped) phase spectrum of the z -transform binomial. The maximum-delay phase spectrum is *not periodic*.

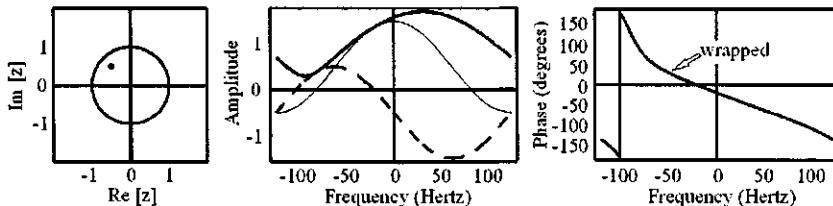


Figure 3.68: Left: Single root z_1 plotted in the z -plane. $z_1 = e^{i135^\circ}$. (Unitc3AndResmax1AndPhsmax1.nb,cdr,wmf)

The unwrapped phase of Figure 3.68 is shown in Figure 3.69. With reference to Figure 3.70, the z -transform binomial $P(z)$ is $P(z) = -z_1 + z = z - z_1$. $P(z)$ can be thought of as the sum of two vectors in the z -plane, a unit vector z that has a different direction for each value of $z = e^{-i\theta}$, and a constant vector $-z_1$. The tip of the resultant vector $P(z)$ defines the modulus and phase of the z -transform binomial $P(z)$. For a minimum-phase binomial $P(z)$ ($|z_1|$ outside the unit circle) the circle (dashed line) traced out by $P(z)$ *does not include the origin* $z = 0$ so that the phase of $P(z)$ must be *bounded and periodic*, as shown in

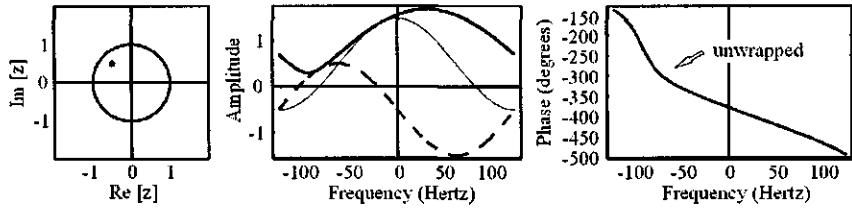


Figure 3.69: Unwrapped phase response of a maximum-delay 2-term wavelet. This is the unwrapped phase spectrum of the binomial $(-z_1, 1)$ whose z -transform is $P(z) = -z_1 + z$ where the root z_1 is shown in Figure 7.20. The maximum-delay phase spectrum is *not* periodic. (UnitCircleAndUnwrappedMaxPhase.nb,cdr,wmf)

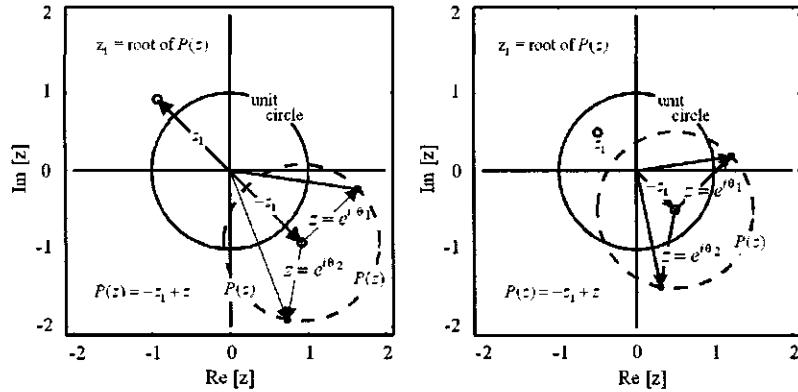


Figure 3.70: Vector plot of minimum-phase binomial (left) $P(z) = -z_1 + z$ where $z_1 = 1.3e^{j\theta_1}$ for $\theta_1 = 2.156$ radians. Two z -plane “vectors” are shown: $P_1(z) = e^{-i2\pi\nu_1 \Delta t} - z_1$ and $P_2(z) = e^{-i2\pi\nu_2 \Delta t} - z_1$ where $\nu_1 = -113.1$ Hertz, $\nu_2 = 49.6$ Hertz, and $\Delta t = 4$ ms. Maximum-phase z -transform shown on the right. (Vmin1AndVmax1.nb,cdr,wmf)

Figure 3.71. In Figure 3.70 (right), the root z_1 is less than 1. $z_1 = 0.7e^{j\theta}$ where $\theta = 135$ deg. The z -transform polynomial $P(z)$ is $P(z) = -z_1 + z = z - z_1$. Again, $P(z)$ can be thought of as the sum of two vectors in the z -plane, a unit vector z that has a different direction for each value of z , and a constant vector $-z_1$. The tip of the resultant vector $P(z)$ defines the modulus and phase of z -transform binomial $P(z)$. For a maximum-phase binomial $P(z)$ the circle (dashed line) traced out by $P(z)$ *includes the origin*, so that the phase of $P(z)$ must increase without bound, as shown in Figure 3.71 (right). The phase is not periodic. Convolution of two minimum-delay binomials results in a minimum-delay wavelet of length 3. Convolution of one or more minimum-delay binomials with one or more maximum-delay wavelets results in a *mixed-delay* wavelet of

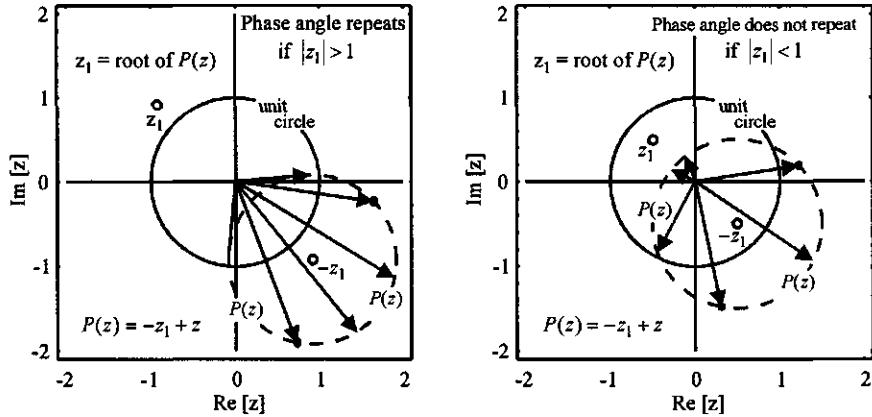


Figure 3.71: Plot of minimum-phase z -transform (left) where $P(z) = -z_1 + z$ and $z_1 = 1.3e^{i\theta_1}$ for $\theta_1 = 2.156$ radians. Several z -plane “vectors” are shown. It is clear that the phase angle of $P(z)$ must repeat because the dashed line defined by $P(z)$ does not include the origin. (vmin1.nb → Vmin1aAndVmax1a.cdr,wmf)

length 3.

The input and output functions of Subroutine FOLD must be defined as complex if the Fortran program FOLD is used to convolve the binomials. This is not necessary if the *Mathematica* program on page 365 is used.

The phase spectrum of a sampled (digitized) minimum-delay function is always periodic. Phase spectra of analog (continuous) functions are not periodic because the time-domain function itself is aperiodic.

The frequency response of the minimum-delay binomial wavelet is obtained by evaluating the z -transform on the unit circle; i.e.,

$$\begin{aligned} P(\omega) &= -z_1 + e^{-i\omega} \\ P(\nu) &= -z_1 + e^{-i2\pi\nu\Delta t} \end{aligned}$$

where $\Delta t = 0.004$ sec = 4 ms. As ν takes on values from

$$\frac{-1}{2\Delta t} < \nu < \frac{+1}{2\Delta t}$$

$$-125 < \nu < 125 \text{ Hertz}$$

the unit vector z completes one revolution of the unit circle in the z -plane. It is not, however, necessary to stop at $1/(2\Delta t)$. Reference to Figure 3.70 shows that as ν increases beyond $1/(2\Delta t)$, the tip of the vector $P(z)$ will continue to fall on the dashed circle. The phase angle associated with $P(\nu)$ is

$$\theta(\nu) = \tan^{-1} \frac{\text{Im}[P(\nu)]}{\text{Re}[P(\nu)]}$$

and the angle θ measured from the positive real axis increases without bounds as ν increases.

3.6.4 Introduction to filters

Complex numbers can be used to describe the concepts behind simple filter operations such as smoothing and differencing. Consider an averaging (smoothing) process that is commonly applied to observational data. If

$$a_0, a_1, a_2, \dots, a_m$$

is the observed data then we define a two-term smoothing as follows

$$b_n = \frac{a_n + a_{n+1}}{2}$$

If we assume that the weighting coefficients are equally important then the a_n and a_{n+1} coefficients can be represented by rotating phasors.

If

$$a_n = e^{in\omega}$$

then the output of a two-term smoothing is

$$b_n = \frac{e^{in\omega} + e^{i(n+1)\omega}}{2}$$

or

$$b_n = \frac{1 + e^{i\omega}}{2} e^{in\omega} \quad (3.138)$$

To determine the effect of the smoothing process, we need to analyze the term $(1 + e^{i\omega})/2$ in (3.138), which is the ratio of the output to the input.

Using the Euler formula we write

$$\frac{1 + e^{i\omega}}{2} = \cos\left(\frac{\omega}{2}\right) e^{-i\omega/2} \quad (3.139)$$

Equation (3.139) describes a new phasor where the modulus is

$$\left| \cos \frac{\omega}{2} \right|$$

and the phase is $-\omega/2$. Both the modulus and phase are functions of the angular frequency. The variation of the elements of this phasor as a function of angular frequency are shown in Figure 3.72, which represents a low-pass filter and shows the smoothing as a summing and/or averaging process that filters out high-frequency components in a tapering-out fashion.

A similar approach can be taken to examine the effect of a differencing filter. We define differencing as

$$b_n = a_{n+1} - a_n$$

Let a_n be a rotating phasor defined as

$$a_n = e^{in\omega}$$

Then the output of a two-term differencing is

$$b_n = e^{i(n+1)\omega} - e^{in\omega}$$

or

$$b_n = e^{in\omega} (e^{i\omega} - 1) \quad (3.140)$$

In order to determine the effect of the differencing process, we need to analyze the term $(e^{i\omega} - 1)$ in (3.140). Equation (3.140) is the system response that becomes

$$\frac{e^{i\omega/2} - e^{-i\omega/2}}{i} e^{-i(\omega/2 - \pi/2)}$$

which is equivalent to

$$2 \sin(\omega/2) e^{-i(\omega/2 - \pi/2)}$$

where the magnitude spectrum is

$$\left| 2 \sin\left(\frac{\omega}{2}\right) \right|$$

and represents a high-pass filter.

The phase spectrum is given by

$$-\left(\frac{\omega}{2} - \frac{\pi}{2}\right)$$

Since the phase spectrum changes between $-\pi/2$ and $\pi/2$, the phase spectrum is defined as

$$\begin{aligned} -\left(\frac{\omega}{2} - \frac{\pi}{2}\right) &\text{ for } \omega \geq 0 \\ -\left(\frac{\omega}{2} + \frac{\pi}{2}\right) &\text{ for } \omega \leq 0 \end{aligned}$$

From Figure 3.72a, we see that at smaller values of angular frequency we have the length of the phasor greater than the length of the phasor at higher angular frequencies. In terms of filtering this is equivalent to a low-pass filter (cosine type) with a linear phase change as shown in Figure 3.72b. Similarly, Figure 3.73 represents a high-pass filter. Figures 3.72 and 3.73 were generated using the *Mathematica* program *AveragingDifferencingCC.nb*, which is included on the CD-ROM that accompanies this volume. In both filters the position of the output along the t -axis has been shifted by an amount $\Delta t/2$.

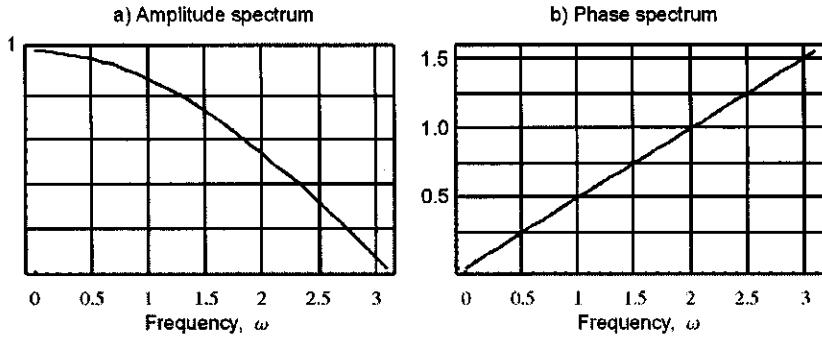


Figure 3.72: (a) Amplitude response of an averaging filter. It is a low-pass filter.
 (b) Phase response of an averaging filter. AveragingDifferencingCC.nb
 \Rightarrow AveragingDifferencing.cdr \Rightarrow Averaging.wmf

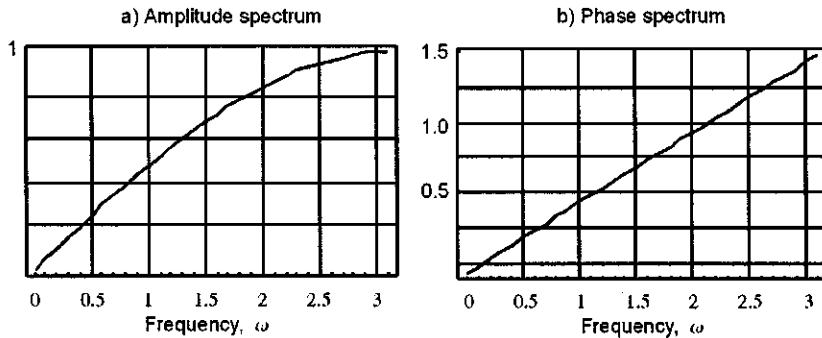


Figure 3.73: (a) Amplitude response of a differencing filter. It is a high-pass filter.
 (b) Phase response of a differencing filter. AveragingDifferencingCC.nb
 \Rightarrow AveragingDifferencing.cdr \Rightarrow Differencing.wmf

Numerical example

The average value of $a[n] + a[n + 1]$ is

$$\frac{a[n] + a[n + 1]}{2} = \frac{a[n]}{2} + \frac{a[n + 1]}{2} = 0.5 a[n] + 0.5 a[n + 1]$$

so the averaging of adjacent points in a time series can be viewed as the convolution (Section 3.5.8 on page 111) of the two-term sequence $(0, 0.5, 0.5)$ with the time series whose adjacent points are to be averaged. Thus,

$$(1, 2, 3, 4, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5) * (0.5, 0.5) = \\ (1.5, 2.5, 3.5, 4.5, 4.5, 3.5, 2.5, 1.5, 0.5, -0.5, -1.5, -2.5, -3.5, -4.5)$$

where (\star) —not to be confused with $(*)$ —denotes convolution and the end points of the filtered output have been discarded because of edge effects. That is, the correct result is obtained only if the input is operated on by the entire filter. Therefore, the longer the filter the larger the edge effect.

The frequency response of the 2-term filter $(0.5, 0.5)$ can be examined by taking its z-transform (see Section 3.6 on page 163) and evaluating it on the unit circle. The z-transform of $(0.5, 0.5)$ where $z = e^{-i\omega}$ is

$$0.5 + 0.5z = 0.5 + 0.5e^{-i\omega}$$

and evaluating this for $0 \leq \omega \leq 2\pi$ gives the amplitude and phase response shown in Figure 3.72. The amplitude and phase spectra are obtained by using *Mathematica*'s `Abs[]` and `Arg[]` functions, respectively. It is clear that an averaging process behaves as a low-pass filter.

A similar approach shows that a differencing process behaves as a high-pass filter. The difference between two adjacent points is

$$a[n+1] - a[n]$$

so the differencing of adjacent points in a time series can be viewed as the convolution of the two-term sequence $(1, -1)$ with the time series whose adjacent points are to be differenced. Thus,

$$(1, 2, 3, 4, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5) \star (1, -1) = \\ (1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1)$$

where (\star) denotes convolution and the end points of the filtered output have been discarded.

The frequency response of the 2-term filter $(1, -1)$ can be examined by taking its z-transform and evaluating it on the unit circle. The z-transform of $(1, -1)$ is

$$1 - z = 1 - e^{-i\omega}$$

and evaluating this for $0 \leq \omega \leq 2\pi$ gives the amplitude and phase response shown in Figure 3.73. It is clear that a differencing filter behaves as a high-pass filter.

Introduction to the pole-zero design of digital filters

We define a “filter element” as the two-term z-transform of a time-domain dipole. That is, the z-transform $P(z)$ of the time-domain dipole $(-z_1, 1)$ is

$$P(z) = -z_1 + z$$

where $z = e^{-i\omega}$. The frequency response $P(\nu)$ of this filter element is $P(z)$ evaluated on the unit circle:

$$\begin{aligned} P(\nu) &= -z_1 + e^{-i\omega\Delta t} \\ &= -z_1 + e^{-i2\pi\nu\Delta t} \end{aligned}$$

A notch filter element

The simplest 60-Hertz notch filter element for which $P(\nu) = 0$ is

$$\begin{aligned} P(z) = -z_1 + z &= -z_1 + e^{-i2\pi\nu\Delta t} \\ &= -z_1 + e^{-i2\pi(60)(0.004)} = -z_1 + e^{-i(1.50796)} = 0 \end{aligned}$$

from which

$$z_1 = e^{-i(1.50796)}$$

for a 60-Hertz notch filter and a sampling interval of 4 ms.

If the filter is to be real, then a second root, z_2 , is required:

$$z_2 = e^{+i(1.50796)}$$

The complete notch filter is therefore

$$P(60) = (-z_1 + z)(-z_2 + z) = (-e^{+i1.50796} + z)(-e^{-i1.50796} + z) \quad (3.141)$$

That is, for perfect rejection of 60 Hertz, the roots z_1 and z_2 lie on the unit circle and the phase angles are ± 1.50796 radians (± 86.4 deg).

The frequency response of Equation (3.141) is shown in Figure 3.74 (upper left). The rejection at 60 Hertz is excellent; however, the response of the filter is not flat at other frequencies. We might improve this by changing the filter $P(z)$ to

$$P(z) = \frac{(-z_1 + z)(-z_2 + z)}{(-z_3 + z)(-z_4 + z)} \quad (3.142)$$

and choose z_3 and z_4 in such a way that wherever the numerator of Equation (3.142) tends to decrease, there will be a corresponding decrease in the denominator of $P(\nu)$ such that the ratio of the numerator to the denominator tends to stay about the same.

One way to achieve this is to define the roots, z_3 and z_4 as

$$z_3 = r_3 e^{+i(1.50796)} \quad (3.143)$$

$$z_4 = r_4 e^{-i(1.50796)} \quad (3.144)$$

where r_3 and r_4 are < 1 .

Equation (3.142) evaluated on the unit circle, i.e., the frequency response $P(\nu)$, is shown in Figure 3.74 (upper right) for $r_3 = r_4 = 0.95$. The response is considerably better; however, it is about 1.1 instead of 1.0 at frequencies other than 60 Hertz. We can increase the values of r_3 and r_4 to, say, 0.99, which should cause $P(\nu)$ to decrease in value at frequencies other than 60 Hertz. The results are shown in Figure 3.74 (lower left). Expecting even further improvement for $r_3 = r_4 = 0.99999$, the results are shown in Figure 3.74 (lower right). The response at frequencies other than 60 Hertz is now quite flat, and the slope of the rejection filter is quite steep. As r_3 and r_4 approach the value of unity, the notch disappears completely, and

$$P(\nu) = 1$$

for any frequency, ν .

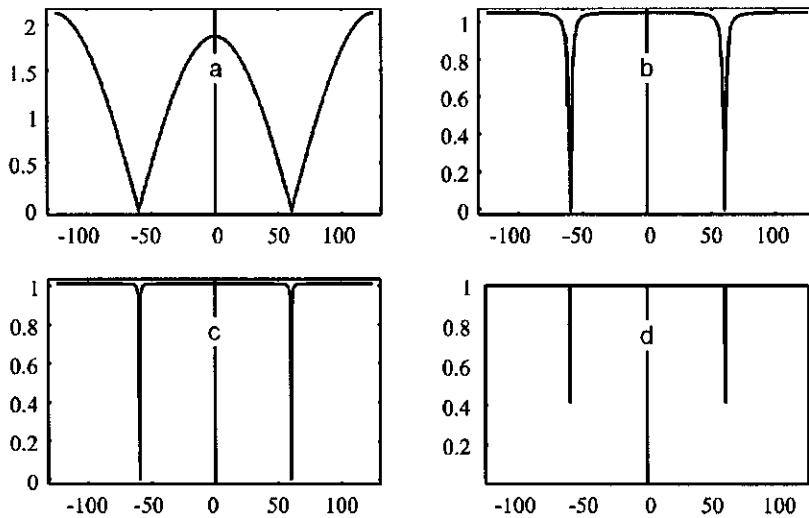


Figure 3.74: a) Simple notch filter. Frequency response computed using Equation (3.141) with the roots on the unit circle. Rejection at 60 Hertz is excellent, but the response at other frequencies is not flat. b) Better notch filter. Response computed using Equation (3.144). The response is now considerably flattened at frequencies other than 60 Hertz. $r_1 = r_2 = 1$ and $r_3 = r_4 = 0.95$. c) Still better notch filter. Response computed using Equation (3.144). $r_1 = r_2 = 1$. With r_3 and r_4 chosen = 0.99, the response is now closer to unity at frequencies other than 60 Hertz, and the notch is steeper. d) Even better(?) notch filter. Response computed using Equation (3.144). With r_3 and r_4 chosen = 0.99999, the response is now close to unity at frequencies other than 60 Hertz; however, at 60 Hertz the response is no longer zero. Indeed, as r_3 and r_4 approach unity, the notch begins to disappear completely. (n60abcd.nb → n60abcd.cdr → n60abcd.wmf)

A bandpass filter

Turning to single-frequency bandpass filters, we might expect to use the same logic to design a bandpass filter using z -transforms. The simplest 60-Hertz bandpass filter element for which $P(60) = 1$ is

$$\begin{aligned} P(z) = -z_1 + z &= -z_1 + e^{-i2\pi\nu\Delta t} \\ &= -z_1 + e^{-i2\pi(60)(0.004)} = -z_1 + e^{-i(1.50796)} = 1 \end{aligned}$$

from which, solving for z_1 ,

$$z_1 = e^{-i(1.50796)} - 1$$

for a 60-Hertz bandpass filter and a sampling interval of 4 ms.

If the filter is to be real, then a second root z_2 is required:

$$z_2 = z_1^*$$

The complete bandpass filter is therefore

$$P(60) = (-z_1 + z)(-z_2 + z) \quad (3.145)$$

That is, for perfect passing of 60 Hertz, the roots z_1 and z_2 lie on the unit circle and the phase angles are ± 1.50796 radians (± 86.4 deg).

Mooney [120] provided further insight into the pole-and-zero design of band-pass and band-reject digital filters.

Filters in parallel

Assume two signals $a(t)$ and $g(t)$ where

$$\begin{aligned} a(t) &\iff A(z) \\ g(t) &\iff G(z) \end{aligned}$$

The function $a(t)$ might be noise added to a seismic trace $g(t)$. Then

$$a(t) + g(t) \iff A(z) + G(z)$$

i.e., two polynomials $A(z)$ and $G(z)$ add when they are in parallel.

What are the delay properties of the sum of two functions in parallel? That is, what is the condition that $Y(z)$ be minimum phase where

$$Y(z) = A(z) + G(z) = A(z) \left[1 + \frac{G(z)}{A(z)} \right]$$

We have assumed that $A(z)$ is minimum phase. Is $1 + G(z)/A(z)$ minimum phase? A sufficient condition for $1 + G(z)/A(z)$ to be minimum phase is that the spectrum of $A(z)$ exceeds that of $G(z)$ at all frequencies, i.e., for any real value of ω , $|A| > |G|$.

If $G(z)/A(z)$ is plotted in the complex plane, then it must lie everywhere inside the unit circle and the quantity

$$1 + \frac{G(z)}{A(z)}$$

always has a positive real part. The curve cannot enclose the origin, therefore the phase must be that of a minimum-phase function. See page 408 for an application of filters in parallel.

Example of a low-pass filter—Butterworth

The famous Butterworth [31] filter is an excellent example of a filter with desirable characteristics in the frequency domain; that is, the transition from a

magnitude of “1” in the middle of the frequency passband to a value of zero in the stopband should be smooth, and the response within the passband should be as flat as possible. We introduce here only a basic introduction to a vast literature on filter design. For a more complete treatment of digital signal design and processing see the excellent works by Oppenheim and Schafer [128], [129], Steiglitz [168], Mitra and Kaiser [119], Steiglitz and others [169], and [133].

A rectangular function in the frequency domain nicely satisfies the “flatness” requirement in the passband but the transition from the passband to the stopband is certainly not smooth, and results in undesirable ripples in the time domain. One approximation to the desired filter characteristics might be

$$\frac{1}{1 + \frac{z-1}{z+1}} \quad (3.146)$$

Just by inspection of (3.146) it is clear that if $z = 1$ then the filter output is a maximum (“1”) at $\omega = 0$. If $z = -1$ then the filter output is zero at the Nyquist frequency ν_N because $z = -1 = e^{-i2\pi\nu_N\Delta t} = e^{-i\pi}$ and the denominator in the fraction of (3.146) goes to zero because the fraction goes to infinity. In between these two endpoints the amplitude response varies smoothly as shown in Figure 3.75. The squared-magnitude function $|H_a(i\omega)|^2$ for an analog Butterworth

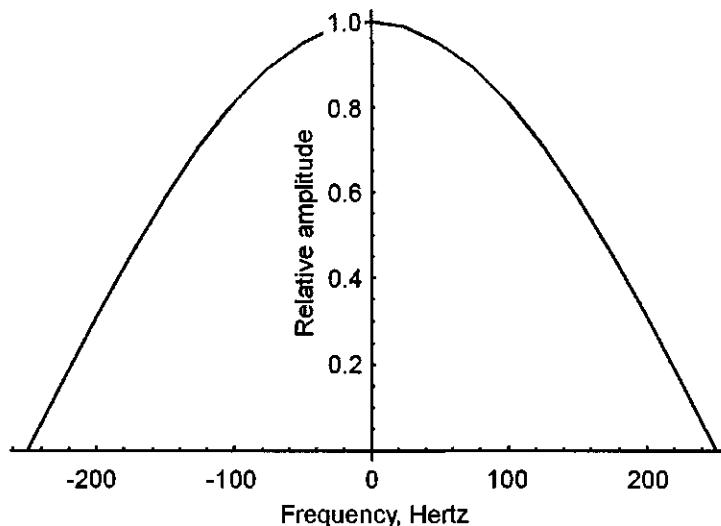


Figure 3.75: General diagram illustrating an approximation to a low-pass filter using Equation (3.146). (Butterworth.nb → Butterworth.cdr → Steiglitz21.wmf)

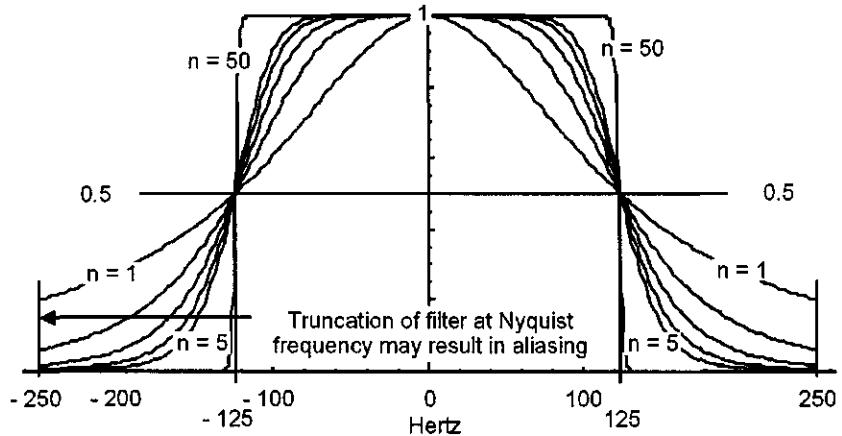


Figure 3.76: General low-pass Butterworth filter using Equation (3.147). (Butterworth.nb → Butterworth.cdr → OppSchaf.wmf)

filter is

$$|H_a(i\omega)|^2 = \frac{1}{1 + (\frac{i\omega}{i\omega_c})^{2n}} \quad (3.147)$$

where ω_c is the “cutoff” frequency and n is called the *order* of the Butterworth filter. The general frequency-domain filtering procedure would next be to take the Fourier transform of the desired filter, multiply it by the Fourier transform of the function (e.g., seismic trace) to be sampled, and then do an inverse Fourier transform to return to the time domain with the filtered result. Note, however, that for some values of n the filter response is not zero at the Nyquist frequency. If the Fourier transform of a time-domain function has a significant non-zero component at the Nyquist frequency ($1/2\Delta t$), then the time-domain data are probably undersampled and aliasing will be present in the frequency domain. A similar relationship holds when starting in the frequency domain. If the inverse Fourier transform of a frequency-domain function has a significant non-zero component at the folding time ($1/2\Delta\nu$), then the frequency-domain data are not sampled at a small enough interval, and aliasing is probably present in the time domain. Thus, in Figure 3.76, the Butterworth filters of order $n = 1$ and $n = 2$ for a sampling interval of $\Delta t = 2$ ms are unsatisfactory, but the higher orders converge to zero near the folding frequency of 250 Hertz. For a sampling interval of 1 ms, orders $2 < n < 5$ are satisfactory for a cutoff frequency of half the Nyquist frequency. For a sampling interval of 4 ms none of them are.

The reader is encouraged to run the *Mathematica* program below to design a satisfactory Butterworth filter.

```
(* Program Butterworth.nb *)
dt=0.002;Nyquist=1 over(2 dt);dfreq=Nyquist/100; nuCutoff=125;
MaxOrder=5;denominator=1+((I 2 Pi nu dt)/(I 2 Pi nuCutoff dt))^(2n);
Plots=Table[0,{MaxOrder}]; i=0;
frequency=Table[nu,{nu,-Nyquist,Nyquist,dfreq}];
Do[i=i+1; butterworth=Table[1/denominator,{nu,-Nyquist,Nyquist,dfreq}];
Plots[[i]]=ListPlot[Transpose[{frequency,Abs[butterworth]}],
PlotRange->{All,{0,1}},PlotJoined->True,DisplayFunction->Identity],
{n,1,MaxOrder}];
Show[Plots[[1]],Plots[[2]],Plots[[3]],Plots[[4]],Plots[[5]],
ImageSize->288]
```

Chapter 4

Computational Considerations

4.1 Effect of Analysis Window on Fourier Spectrum

Windows are special signals of finite extent that are designed for the purpose of modifying, for example, a seismic trace in order to enhance various forms of analysis, such as spectral analysis. For example, if you select from the middle of a seismic trace one second of a signal that lasts five seconds, then you have multiplied the seismic trace by zero except for the portion that you have selected (the analysis window), which has essentially been multiplied by unity. The selection process, as shown below, can be improved by some kind of taper at each end of the data analysis window in order to avoid the abrupt onset and termination of signal. The tapering can be done in either the time domain or frequency domain. The 21-sec vibroseis sweep used by the Virginia Tech seismic crew, for example, was tapered at each end with a 1-sec taper in order to improve spectral estimates of the data.

The Fourier transform $F(\nu)$ of a continuous function $f(t)$ is defined as

$$F(\nu) = \int_{-\infty}^{+\infty} f(t)e^{-i2\pi\nu t} dt$$

or, because $\omega = 2\pi\nu$,

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt$$

Any integral of this form, no matter what the nature of $f(t)$, is therefore a Fourier transform of $f(t)$.

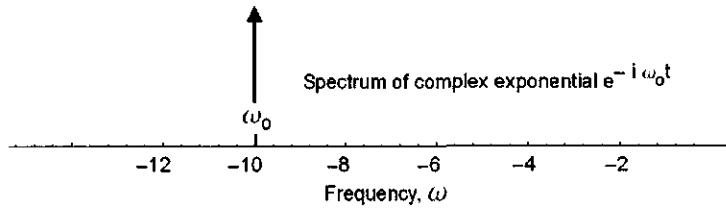


Figure 4.1: Fourier transform of $e^{-i\omega_0 t}$ with $\omega_0 = 10$ radians/sec.

The Fourier transform of the complex exponential $e^{-i\omega_0 t}$ of infinite length is given by

$$\int_{-\infty}^{\infty} e^{-i\omega_0 t} e^{-i\omega t} dt = i\delta(\omega + \omega_0) \quad (4.1)$$

where

$$\begin{aligned} \delta(\omega + \omega_0) &= \infty \quad \text{for } \omega = \omega_0 = -10 \\ &= 0 \quad \text{for } \omega \neq \omega_0 \end{aligned} \quad (4.2)$$

In other words, the *amplitude density* of $\delta(\omega)$, i.e., the amplitude per frequency interval, is infinite because there is only one sinusoid of finite amplitude (unity) over a frequency interval of, in the limit, zero. The *amplitude* of the single sinusoid is

$$\frac{\text{amplitude}}{\text{frequency interval}} \times \text{frequency interval} = \int_{-\infty}^{\infty} \delta(\omega - \omega_0) d\omega = 1$$

The result is shown in Figure 4.1 for $\omega_0 = 10$. If the window is not of infinite length, however, then the entire spectrum is affected by the finite length of the analysis window.

A portion of a sinusoid is shown in Figure 4.2. It is clear from Figure 4.2 that, although the maximum spectral amplitude is still at a frequency of 10 Hertz, frequencies on either side of 10 Hertz now have non-zero amplitudes, even though a single sinusoid is present in the analysis. The effect of windowing has been to introduce energy throughout the entire spectrum, and the "true" spectrum of just the sinusoid has not been recovered. This effect is often called "leakage".

If the window is opened to a width wider than that shown in Figure 4.2, then this effect on adjacent frequencies is reduced, as shown in Figure 4.3, until the side lobes vanish, as shown in Figure 4.1. The shape as well as the width of the analysis window has an affect on any attempt to recover the "true" spectrum. Various window shapes can be used to minimize the amplitudes of side lobes and to decrease the width of the central lobe. Several windows are illustrated in Figure 4.4.

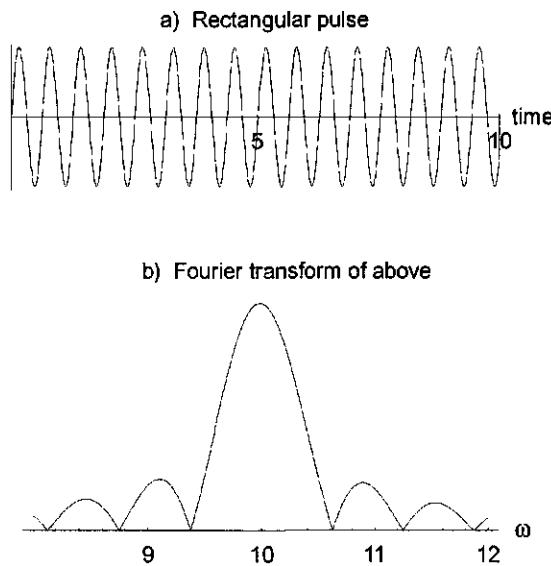


Figure 4.2: a) Truncated sinusoid of frequency $\omega_0 = 10$ radians/sec. b) Modulus of Fourier transform of above. Note the leakage (side lobes) on either side of $\omega = 10$ caused by the finite width of the analysis window. (cd \rightarrow rectsin)

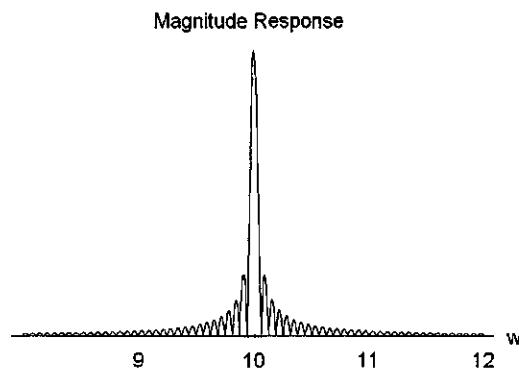


Figure 4.3: Fourier transform of a sinusoid of finite length, with window 10 times wider than that of Figure 4.2.

Clearly, in order to minimize the effect of side lobes, anything is better than the rectangular window of Figure 4.4. On the other hand, the rectangular window provides the best resolution (narrowest central peak). Further treatment

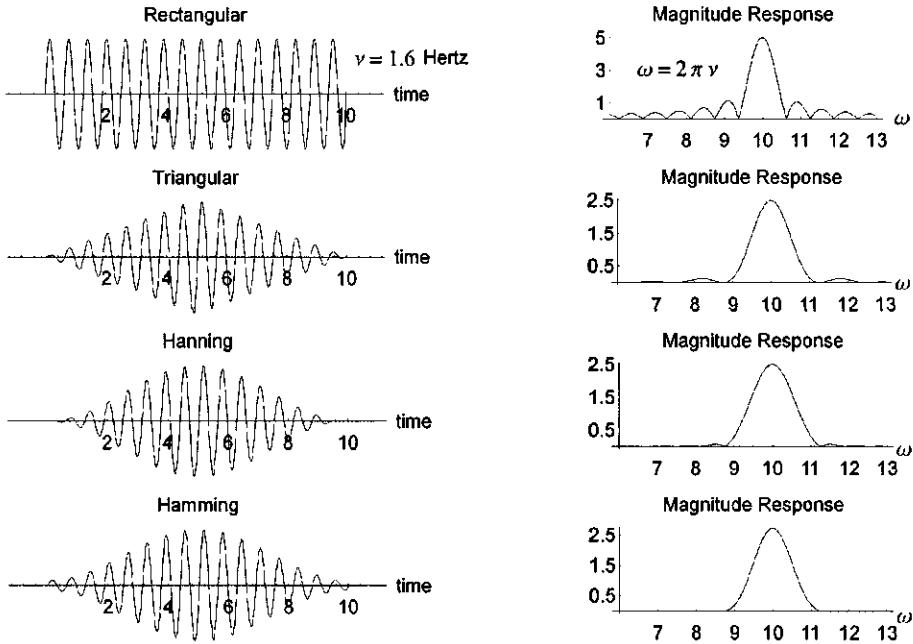


Figure 4.4: Truncation effects of various analysis window shapes as applied to a single sinusoid of frequency 1.6 Hertz. Left-hand column: time-domain. Right-hand column frequency domain. From top: Rectangular analysis window, triangular, Hanning, Hamming. The side lobes associated with the choice of the shape of the analysis window are decreased from top to bottom; however, this is done at the expense of the width of the central lobe, which gets wider from top to bottom. It is a trade-off. `Hanning.nb` \Rightarrow `Windows.cdr` \Rightarrow `HanningWindow.nb`

of the design and effects of analysis windows is beyond the scope of this discussion. The reader is referred to any of several excellent references. Suffice it to say that, in order to minimize the contamination of the true frequency spectrum with the spectrum of the analysis window, the longer the data set, the better.

4.2 Aliasing

A function $f(t)$ that is digitized, i.e., sampled, at a sampling interval Δt is sampled at times $0, \Delta t, 2\Delta t, 3\Delta t, \dots$. The sampling rate is $1/\Delta t$. We will show that, under these conditions, any frequencies that might be present in the signal

that are greater than $1/(2\Delta t)$ Hertz cannot be distinguished from frequencies in the range below the folding frequency, i.e., in the range 0 to $1/(2\Delta t)$ Hertz. This phenomenon is called “aliasing”, and is a conclusion of the Sampling theorem, or Shannon’s theorem, of information theory. Aliasing can occur in the time domain or in the frequency domain.

4.2.1 Sampling in the time domain – aliasing in the frequency domain

Following Susskind [174], if a signal is band-limited, which it is if it is sampled, then its value at any time t is determined by the values it takes at the times

$$k\Delta t \quad k = 1, 2, 3, \dots$$

The frequency $1/(2\Delta t)$ is called the *folding*, or *Nyquist* frequency, ν_N .

$$\nu_N = \frac{1}{2\Delta t}$$

The relation between angular frequency ω (radians/time unit) and circular frequency ν (cycles per time unit = Hertz), is:

$$\omega = 2\pi\nu$$

If ω_N is the Nyquist frequency in radians/sec, then

$$\omega_N = 2\pi\nu_N = 2\pi \left[\frac{1}{2\Delta t} \right] = \frac{\pi}{\Delta t}$$

If the data observed (recorded) actually do contain frequencies higher than ν_N , say up to ν_H , then aliasing occurs because the sampling interval is incorrectly chosen such that

$$\Delta t = \frac{1}{2\nu_N} < \frac{1}{2\nu_H}$$

Aliasing will cause two problems:

1. Amplitudes of frequencies beyond ν_N will not be recovered.
2. Amplitudes of frequencies between $2\nu_N - \nu_H$ will be incorrect because of aliasing. For example, If $\nu_N = 125$ Hertz and $\nu_H = 150$ Hertz then the amplitudes of frequency components between 100 Hertz and 125 Hertz will not be correct in addition to not having the frequency components higher than 125 Hertz.

To show why the sampling theorem is true, consider the sinusoid

$$\cos(\omega t + \phi) = \cos \left[\left(\frac{\pi}{\Delta t} - 2\pi\nu \right) t + \phi \right]$$

where $\pi/\Delta t - 2\pi\nu$ is any frequency below the folding frequency ω_N . We define this frequency below the folding frequency as ω_b :

$$\omega_b = \frac{\pi}{\Delta t} - 2\pi\nu = \omega_N - \omega$$

Let this sinusoid be sampled at times $t = k\Delta t$, $k = 0, 1, 2, 3, \dots$; i.e., whenever t is a multiple of the sampling interval. Then, because $\cos(x) = \cos(-x) = \cos(k2\pi \pm x)$,

$$\begin{aligned}\cos \left[\left(\frac{\pi}{\Delta t} - 2\pi\nu \right) k\Delta t + \phi \right] &= \cos [k\pi - k2\pi\nu\Delta t + \phi] \\ &= \cos [-k\pi + k2\pi\nu\Delta t - \phi] \\ &= \cos [k2\pi - k\pi + k2\pi\nu\Delta t - \phi] \\ &= \cos [k\pi + k2\pi\nu\Delta t - \phi] \\ \cos \left[\left(\frac{\pi}{\Delta t} - 2\pi\nu \right) k\Delta t + \phi \right] &= \cos \left[\left(\frac{\pi}{\Delta t} + 2\pi\nu \right) k\Delta t - \phi \right]\end{aligned}$$

where ϕ is any arbitrary phase angle.

Note that the frequency $\pi/\Delta t + 2\pi\nu$ lies *above* the folding frequency. Note also that

$$\omega_a = \frac{\pi}{\Delta t} + 2\pi\nu = \frac{2\pi}{\Delta t} - \frac{\pi}{\Delta t} + 2\pi\nu = \frac{2\pi}{\Delta t} - \left(\frac{\pi}{\Delta t} - 2\pi\nu \right) = 2\omega_N - \omega_b$$

or, in terms of cycles per time unit,

$$\nu_a = \nu_N + \nu = 2\nu_N - \nu_N + \nu = 2\nu_N - \nu_b$$

where $\nu_b = \nu_N - \nu$.

The conclusion is that a sinusoid of frequency $\pi/\Delta t + 2\pi\nu$ is equal to a sinusoid of frequency $\pi/\Delta t - 2\pi\nu$ whenever $t = k\Delta t$, i.e., at times for which t is a multiple of the digitizing interval (a phase difference between the two sinusoids is not important). The effect of sampling has made it impossible to separate the two frequencies.

Similar reasoning holds for resampling data that have already been sampled in order to reduce the size of the data set. Simply discarding every other sample in the data set might introduce a serious aliasing problem. Instead, the data should first be filtered using a high-cut filter (an anti-aliasing filter) that removes all frequencies above $1/(2\Delta t)$, where Δt is now the new sampling interval.

What this all means is that, if you choose to sample an analog signal using a sampling interval of Δt , then there had better not be any frequencies present in the signal that lie above the frequency $1/(2\Delta t)$. If there are indeed frequencies present above $1/(2\Delta t)$, then the spectrum that you compute for frequencies below $1/(2\Delta t)$ will be incorrect.

4.2.2 Example

As a numerical example suppose the sinusoid to be digitized (sampled) at a sampling interval of Δt has a frequency of $\pi/\Delta t + 2\pi\nu$ where $\Delta t = 4$ ms and

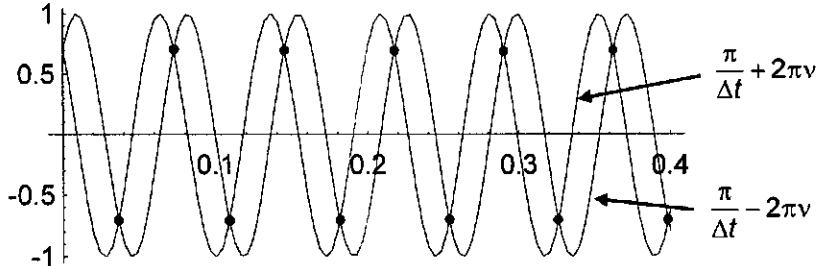


Figure 4.5: Plot of two cosine waves, one with a frequency of $\pi/\Delta t + 2\pi\nu$ and the other with a frequency of $\pi/\Delta t - 2\pi\nu$. The folding frequency is $\pi/\Delta t$. The two sinusoids are equal for times $t = k\Delta t$ (black dots). It is not possible to distinguish the two sinusoids of different frequencies because of aliasing. (ma + cd → alias1f)

$\nu = 13.8$ Hertz. The frequency of the sinusoid is, then,

$$\frac{1}{2\Delta t} + \nu = 125 + 13.8 \approx 139 \text{ Hertz}$$

The sinusoid having this frequency (and a phase angle of $\pi/4$) is shown in Figure 4.5 as $\cos[(\pi/\Delta t + 2\pi\nu)t + \beta]$. But there is another sinusoid of frequency

$$\frac{1}{2\Delta t} - \nu = 125 - 13.8 \approx 111 \text{ Hertz}$$

that is equal to the first for times $t = k\Delta t$. This sinusoid,

$$\cos[(\pi/\Delta t - 2\pi\nu)t - \beta]$$

is also shown in Figure 4.5.

The frequency $\pi/\Delta t - 2\pi\nu$ corresponds, in this example, to a frequency of approximately 111 Hertz. Thus, a sampling rate of $1/\Delta t$, or 250 samples per sec, does not distinguish between 139 Hertz and 111 Hertz. Or, for example, if the sample interval were $\Delta t = 4$ ms, then it would not be possible to distinguish between $\nu = 125 - 30 = 95$ Hertz and $\nu = 125 + 30 = 155$ Hertz.

What this means is that there had better not be any frequencies greater than the folding frequency ν_N present in the recorded data before sampling. Such frequencies will be “aliased” into the passband below ν_N , *even if there are no such frequencies below ν_N* . Furthermore, if the sinusoid of frequency $\pi/\Delta t + \nu$ is added to the sinusoid of frequency $\pi/\Delta t - 2\pi\nu$, and the resulting signal is observed only at the times $t = k\Delta t$, an incorrect waveform based on these observation results.

The effect of sampling, then, is to fold the frequency scale like an accordion with a fold at every multiple of $\pi\Delta t$. Frequency components on one side of a fold are “aliased”, or called by another name, on the other side of the fold.

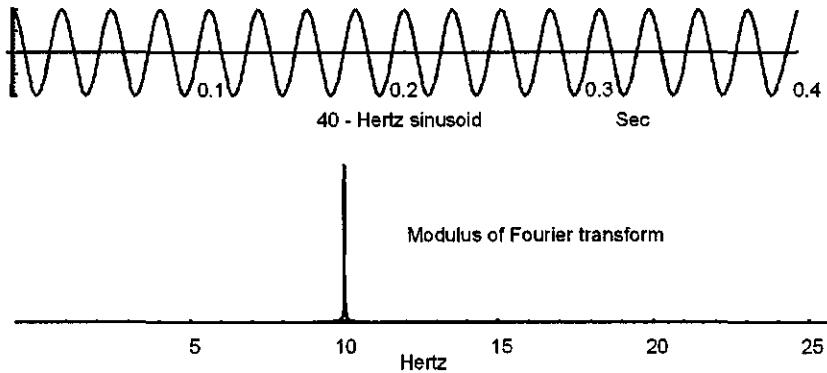


Figure 4.6: Example of an aliased spectrum. A sinusoid of frequency 40 Hertz (portion of the sinusoid shown in top diagram) has been under-sampled at an interval of 20 ms, which makes the folding frequency, $\nu_N = 1/(2\Delta t) = 25$ Hertz. The Fourier transform of this signal (bottom) has identified a spectral component of $2\nu_N - \nu = 50 - 40 = 10$ Hertz, a component that is not even present. The 40-Hertz peak is not there because the discrete computation is only carried out to 25 Hertz (the folding frequency). The sharpness of the 10-Hertz peak is a simply a consequence of using a signal of 100 sec length to get the Fourier transform; window effects, including side lobes, have therefore been minimized. (ma + cd → alias2f)

Suppose then, that the function $f(t) = A_1 \sin(2\pi\nu_1 t + \phi)$ is sampled at an interval of $\Delta t = 0.02$ s. The folding frequency, ν_N , is 25 Hertz. Let $\pi/\Delta t + 2\pi\nu$ be 40 Hertz. Then the sampling rate cannot distinguish between $25 + 15 = 40$ Hertz and $25 - 15 = 10$ Hertz.

With ν_1 chosen as 40 Hertz, the spectral analysis for the range 0-25 Hertz will see an aliased component with a frequency of $\pi/\Delta t - 2\pi\nu$ Hertz. The analysis does not extend to the true frequency but will identify a frequency of 50-40, or 10 Hertz, a component that does not even exist. The modulus of the aliased Fourier spectrum is shown in Figure 4.6. In general, the amplitude calculated for any frequency ν is the sum of the amplitudes for frequencies ν , $2\nu_N \pm \nu$, etc. The frequencies ν , $2\nu_N - \nu$, $2\nu_N + \nu$, $4\nu_N + \nu$, etc., are called *aliases* of one another, ν being the *principal alias*.

Aliasing is, of course, extremely serious but can be avoided by

1. Choosing a digitizing interval Δt small enough so that the folding frequency $1/(2\Delta t) = \nu_N$, is greater than any significant frequency in the original signal. If, after Fourier transformation and inspection of the amplitude spectrum, there is significant energy near the folding frequency then the data are probably aliased.

As an example of an aliased seismic spectrum, we choose "Trace 18", which was originally digitized from analog data at a sampling interval of 1 ms.

The data are shown in Figure 4.7. These data were then resampled at 2 ms, 4 ms, and 8 ms and Fourier transforms computed for each resampled trace. The resampling was done by simply discarding points; no filtering was applied before resampling. Results are shown in Figure 4.7.

2. Filtering out the high frequencies above the folding frequency *before* the signal is sampled. If frequencies above ν_N are not present in the signal to begin with, then they cannot be aliased into frequencies below ν_N . Thus, when recording seismic data, the selection of a sampling interval Δt by the observer for the recording instrumentation is commonly accompanied by the automatic introduction of *anti-alias* analog filters that guarantee that no signal above $1/(2\Delta t)$ will reach the analog to digital converters. It usually requires a non-trivial effort to circumvent the insurance provided by the instrumentation.

How to determine if aliasing is present

If the data have been improperly sampled (too large a value of Δt), then a Fourier transform of the data will show "significant" amplitudes of Fourier components near and at the folding frequency. This would make one suspicious about the quality of the data.

Another approach is to compute the amplitude spectrum $A_1(\omega)$ of the data with a sampling interval Δt_1 . Then resample the data with a sampling interval $\Delta t_2 = \Delta t_1/2$. Compute the amplitude spectrum $A_2(\omega)$ of the data with a sampling interval Δt_2 . Compare $A_1(\omega)$ with $A_2(\omega)$. A significant difference between these spectra, especially at high frequencies, will indicate an aliasing problem. This is essentially the approach shown in Figure 4.7.

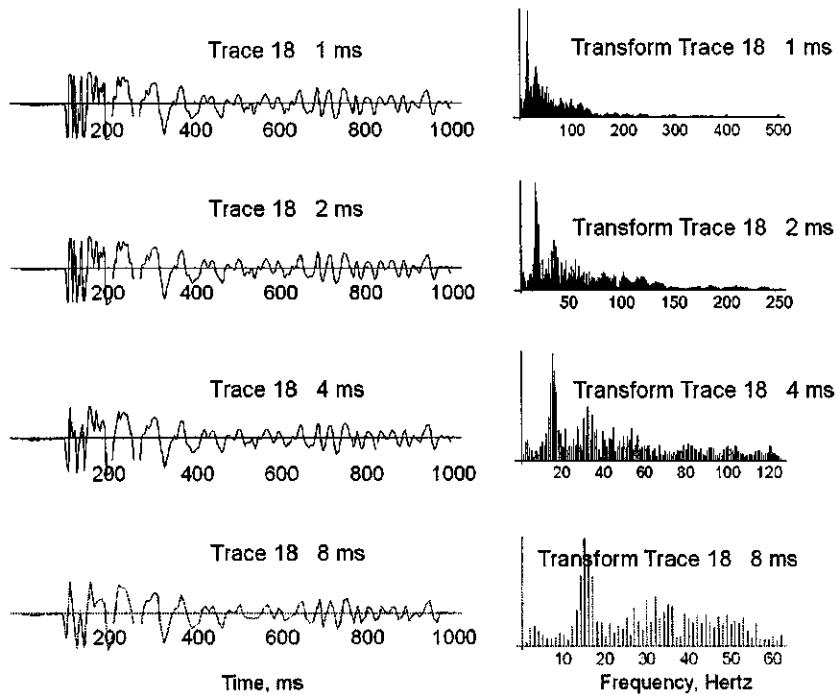


Figure 4.7: A check on aliasing. Top pair of figures shows Trace 18 (left) sampled from analog data at an interval of 1 ms. Figure to right is modulus of its Fourier transform. The energy clearly dies out well before the folding frequency of 500 Hertz. Next pair down is same data sampled at 2 ms by simply discarding every other point. No anti-alias filter was applied. Again, the spectrum (right) is essentially zero near the folding frequency of 250 Hertz. Third pair down is Trace 18 sampled at 4 ms. Energy is present near the folding frequency of 125 Hertz and the spectrum is probably aliased. Bottom diagram: Trace (left) resampled at 8 ms. Spectrum (right) is aliased. (ma → alias18a,b → cd alias18)

Chapter 5

Synthetics and Velocity Functions

One of the major breakthroughs in geophysics came with the ability to predict how a layered medium will respond to a general plane-wave impulsive input. The impulse response of the layered system is simply the recorded seismic trace. We examine first what would be reflected from a single interface that separates two media of different elastic properties. Our results will explain why a single very large shallow reflector does *not* mask reflection energy from a deeper interface.

5.1 Normal-incidence reflection coefficient

As an introduction to plane-wave reflection and transmission coefficients associated with general angles of incidence, we determine the amplitudes of reflected and transmitted waves for normal incidence at a plane boundary. The assumptions made are:

1. All media are linear, elastic, homogeneous, and isotropic, and
2. A plane wave sinusoid arrives at the boundary at normal incidence with a general frequency, wavelength, period, etc.

Picture a plane boundary between two media in the subsurface. The upper medium has a P -wave velocity of α_1 and a density of ρ_1 , and the lower medium has a velocity of α_2 and a density of ρ_2 . The *acoustic impedance* Z is defined as the product of density and velocity in each medium, or

$$Z_1 = \rho_1 \alpha_1$$

and

$$Z_2 = \rho_2 \alpha_2$$

In this example let's say $Z_2 > Z_1$. As a sinusoidal wave approaches, particles are set in motion. The work (energy) done in accelerating the particle of mass m from a velocity of zero to a velocity α_p is

$$E = \int_{x_1}^{x_2} F dx = \int_{x_1}^{x_2} m \bar{a} dx = \int_{x_1}^{x_2} m \alpha_p d \frac{\alpha_p}{dx} dx = \int_0^{\alpha_p} m \alpha_p d\alpha_p = \frac{1}{2} m \alpha_p^2$$

where \bar{a} is particle acceleration. The particle motion is parallel to the direction of propagation of the incident P -wave, and has a kinetic energy E that, from above, is defined as

$$E = \frac{1}{2} m \alpha_p^2$$

where α_p is the instantaneous velocity of the particle.

If the disturbance at the interface is a sinusoid, then the distance a of the particle from its undisturbed position at any time t is,

$$a = A \cos(\omega t)$$

where A is the maximum amplitude of the sinusoidal disturbance. When

$$a = A$$

then *all* of the energy of the particle is potential energy; the kinetic energy of the particle is zero. When $a = 0$, then *all* of the energy of the particle is kinetic energy; the potential energy of the particle is zero. These two energy states are equal.

The *particle velocity* α_p at any time t is the derivative of the above sinusoid with respect to time, i.e.,

$$\alpha_p = \frac{da}{dt} = -(\omega A) \sin(\omega t)$$

The magnitude of the *maximum* particle velocity is therefore

$$|\omega A|$$

When the particle has this velocity, then all of the energy of the particle has been accounted for, and the particle has no potential energy. Therefore, the *total* energy of the particle of mass m is simply

$$\frac{1}{2} m (\omega A)^2$$

If the energy isn't all kinetic or all potential, then it is *partly* potential and partly kinetic. But the sum is always the same (analogy with swing).

So we don't need to consider both potential and kinetic energy. The sum at any time t is always equal to the maximum kinetic energy of the particle. This constant amount of energy associated with the mass m and with all other

particles of mass m that are encountered along the direction of propagation of the seismic wavefront is transmitted across the interface with no loss of energy. We want to investigate changes in amplitude associated with reflection and transmission. There will be no changes in energy either along the ray path or across the boundary.

We now need to express the following:

1. How to express the propagation of this constant amount of energy along a ray path before it gets to the interface.
2. How to express the propagation of this constant amount of energy across the interface from the upper to the lower medium, or vice versa.

We want the energy per unit time per unit area to be constant along the ray path. That is, we want

$$\frac{\text{energy}}{\text{time-area}} = \text{constant}$$

So we have

$$\frac{\text{energy}}{\text{time-area}} = \frac{\frac{1}{2}m(\omega A)^2}{\alpha \Delta t} = \text{constant}$$

where Δt is some arbitrary small increment of traveltime spent along the ray path, and α is the area normal to the direction of propagation.

Now note that in the same length of time Δt the wave front would travel a distance

$$\Delta t = \frac{d_1}{\alpha_1}$$

in the upper medium, and

$$\Delta t = \frac{d_2}{\alpha_2}$$

in the lower medium. In this length of time Δt the wavefront will encounter a total mass m_1 in the upper medium, and a total mass m_2 in the lower medium.

Substituting for Δt in each medium, we get

$$\frac{\text{energy}}{\text{time-area}} = \frac{\frac{1}{2}m_1(\omega A_1)^2}{[d_1/\alpha_1]\alpha} = \frac{\frac{1}{2}m_2(\omega A_2)^2}{[d_2/\alpha_2]\alpha}$$

But

$$\frac{m_1}{d_1\alpha} = \rho_1$$

and

$$\frac{m_2}{d_2\alpha} = \rho_2$$

where ρ_1 is the density in the upper medium and ρ_2 is the density in the lower medium. Therefore,

$$\frac{\text{energy}}{\text{time-area}} = \frac{1}{2}\rho_1\alpha_1(\omega A_1)^2 = \frac{1}{2}\rho_2\alpha_2(\omega A_1)^2$$

where

$$\alpha = \alpha_1 \text{ in the upper medium}$$

and

$$\alpha = \alpha_2 \text{ in the lower medium}$$

Thus, the general notation

$$\frac{1}{2}\rho\alpha(\omega A)^2$$

describes the transport of particle motion energy at the *P*-wave velocity α of the medium, per unit time per unit area in either medium; i.e., the constant amount of energy that is being transported along the ray path at the velocity α . We have found an expression for condition 1 above.

In the discussion below, we will be equating terms of the form

$$\frac{1}{2}\rho v(\omega A)^2$$

where v represents a medium velocity and so the ω^2 and $1/2$ will cancel, leaving

$$\rho\alpha A^2 = ZA^2$$

where $\rho\alpha$ is a density times a velocity (= "impedance" Z).

The fact that ω cancels out means that a "plane wave reflection coefficient" is independent of frequency.

At the interface, the vertically incident energy is partitioned into reflected and transmitted energy

$$Z_1 A_1^2 = Z_2 A_2^2 + Z_1 [RC_d A_1]^2 \quad (5.1)$$

where RC_d is the reflection coefficient associated with the downward-going incident wave $Z_1 [(RC_d A_1)]^2$ is the energy reflected back upward into the incident medium, and $Z_2 A_2^2$ is the energy transmitted into the medium below the interface.

We start with an amplitude A_1 equal to 1. Solving Equation (5.1) for A_2 (the transmitted amplitude), we get

$$A_2 = \sqrt{\frac{Z_1 A_1^2 - Z_1 [RC_d A_1]^2}{Z_2}} \quad (5.2)$$

Going back up through the interface we would have a similar partitioning of energy:

$$Z_2 A_2^2 = Z_2 [RC_u A_2]^2 + Z_1 A_1^2$$

where A_1 is now the upward transmitted amplitude in the upper medium and RC_u is the reflection coefficient associated with the downward-reflected energy and defined by

$$RC_u = \frac{Z_1 - Z_2}{Z_1 + Z_2} \quad (5.3)$$

Solving for A_1 gives

$$A_1 = \sqrt{\frac{Z_2 A_2^2 - Z_2 [RC_u A_2]^2}{Z_1}} \quad (5.4)$$

where A_1 is now the transmitted amplitude in the upper medium.

To summarize, we start with some value for A_1 and, using Equation (5.2) we solve for A_2 , the transmitted amplitude. We then take this value and insert it into Equation (5.4) to get the transmitted value A_1 . Then compare this value of A_1 with what we started with. We see that even for a reflection coefficient of 0.3 (an extremely high value), we retain 91 per cent of the amplitude of what we started out with.

The conclusion is that a very strong shallow reflector does not mask even weak reflections from deeper reflectors.

5.1.1 Example

Assume a reflection coefficient of about 0.3 (a very large value); most reflection coefficients are less than 0.1. So let the reflection coefficient RC_d associated with the down-going wave be

$$RC_d = \frac{Z_2 - Z_1}{Z_2 + Z_1} = 0.30154$$

where we have arbitrarily defined

$$Z_1 = \rho_1 \alpha_1 = 2.4 \text{ (gm/cm}^3\text{)} (4.0 \text{ km/sec})$$

and

$$Z_2 = \rho_2 \alpha_2 = 2.67 \text{ (gm/cm}^3\text{)} (6.7 \text{ km/sec})$$

which is appropriate for the Rome shale/Shady dolomite impedance contrast in the Appalachian mountain system. The reflection coefficient is

$$RC_d = \frac{\rho_2 \alpha_2 - \rho_1 \alpha_1}{\rho_2 \alpha_2 + \rho_1 \alpha_1} = 0.30154$$

From Equation (5.2), the amplitude of the transmitted wave is 0.69846. Thus we have

$$\begin{aligned} Z_1 &= \rho_1 \alpha_1 = (2.4 \text{ gm/cm}^3) (4.0 \text{ km/sec}) = 9.6 \text{ gm/cm}^3 \\ Z_2 &= \rho_2 \alpha_2 = (2.67 \text{ gm/cm}^3) (6.7 \text{ km/sec}) = 17.889 \text{ gm/cm}^3 \\ TC_d &= \text{Sqrt} \frac{Z_1 A_1^2 - Z_1 [RC_d A_1]^2}{Z_2} = \end{aligned}$$

from which we can deduce

$$RC_d = \frac{Z_2 - Z_1}{Z_1 + Z_2} \quad (5.5)$$

$$\begin{aligned} TC_d &= 1 - RC_d \\ &= \frac{2 Z_1}{Z_2 + Z_1} \end{aligned} \quad (5.6)$$

The definition of the transmission coefficient (going down) TC_d is

$$TC_d = \sqrt{(1 - RC_d)^2} = 0.69846$$

where

$$TC_d + RC_d = 1$$

but we did not need to know this definition to get A_2 .

Going back up, the reflection coefficient is

$$RC_u = \frac{\rho_1 \alpha_1 - \rho_2 \alpha_2}{\rho_2 \alpha_2 + \rho_1 \alpha_1} = -0.30154$$

Using Equation (5.4) with $A_2 = 0.69846$, and solving Equation (5.4) for A_1 , the amplitude of the transmitted wave going back up after going through the interface from below, we get 0.90907, which is almost the same amplitude that we started with!

Said another way, by definition, the transmission coefficient (going up) TC_u is

$$TC_u = \sqrt{(1 - RC_u)^2} = 1.30154$$

where, again

$$TC_u + RC_u = 1$$

although we did not need this definition; however, notice that this up-going transmission coefficient is greater than 1.

The *two-way* transmission coefficient $TWTC$ is the product of the down and up transmission coefficients:

$$TWTC = (TC_d)(TC_u) = (0.69846)(1.30154) = 0.90907$$

and 0.90907 is almost what we started with so that the amplitude of the up going wave after transmission down through the reflector and back up through the reflector is still 91 per cent of the amplitude of the original incident wave. Of course, this value will be decreased by whatever the value of the deeper reflection coefficient from the second reflector.

The preceding was derived from basic principles and for normal incidence. Equations (5.5) and (5.6) also result from the solution of the Zoeppritz equations [206, 3] for normal incidence. The conclusion is that a strong shallow reflector does not mask even weak reflections from deeper reflectors.

5.2 Values of Reflection Coefficients

What are some typical values of reflection coefficients? If no well logs are available then measurements of velocity and density can be made on carefully prepared surface samples. For the Mississippian through Cambrian shelf strata and Precambrian basement rocks in the Valley and Ridge province near the Bane Dome in southwestern Virginia Edsall [69], Kolich [100], and Wells [190] reported on velocity and density values as measured in the laboratory or obtained from field studies. These rocks are well indurated but relatively unmetamorphosed. In order to close microcracks it is necessary to determine velocities in a pressure cell under hydrostatic pressure. Velocity measurements were made at hydrostatic pressures of 200, 400, and 600 atmospheres. Only the results for the highest confining pressures are shown in Table 5.1. The values of velocity and density yield acoustic impedance values that can then be used to compute expected values of reflection coefficients. From their results it is clear that the limestones, dolomites and crystalline basement have quite high velocities of about 6 km/sec, which is close to an average velocity that can be used for the entire Appalachian Valley and Ridge Paleozoic section. As shown by Kolich [100, p. 38] the largest reflection coefficient to be expected within the shelf strata is +0.29 between the Rome shale and Shady dolomite. This is an extremely large value and reflections from this part of the lithologic sequence are easily recognized and widespread. Most reflection coefficients are less than | 1 |.

The Paleozoic shelf strata are well indurated and relatively unmetamorphosed. On the other hand the relatively unconsolidated rocks of the Atlantic Coastal Plain and Gulf Coast have much lower velocities and densities as shown in Figure 5.1. In the eastern United States these sediments comprise a seaward-thickening wedge of unmetamorphosed, unconsolidated to semi-consolidated sediments, predominantly clastic, mostly sand, silt, clay, and limestone. Most of the sediments are of Cretaceous age and generally have an upward transition from fluvial and fluviodeltaic to marginal marine to (younger than Cretaceous) marine deposits. The crystalline rocks beneath these sediments are similar to those exposed to the west in The Piedmont and Blue Ridge provinces. The sedimentary wedge increases in thickness from zero to about 3 km beneath Cape Hatteras. The logs illustrated in Figure 5.1 show the densities and velocities of these sediments, and are typical for the entire Atlantic Coastal Plain. The

Crisfield drill site was targeted for potential geothermal applications because it was the location of the highest known temperatures at the shallowest depths in the Maryland-Delaware-Virginia region.

The sonic and density logs shown in Figure 5.1 are used later (page 218) to determine the reflection coefficients for this area of the Atlantic Coastal Plain. Synthetic seismograms are computed and compared using the method described by Sengbush et al. [161].

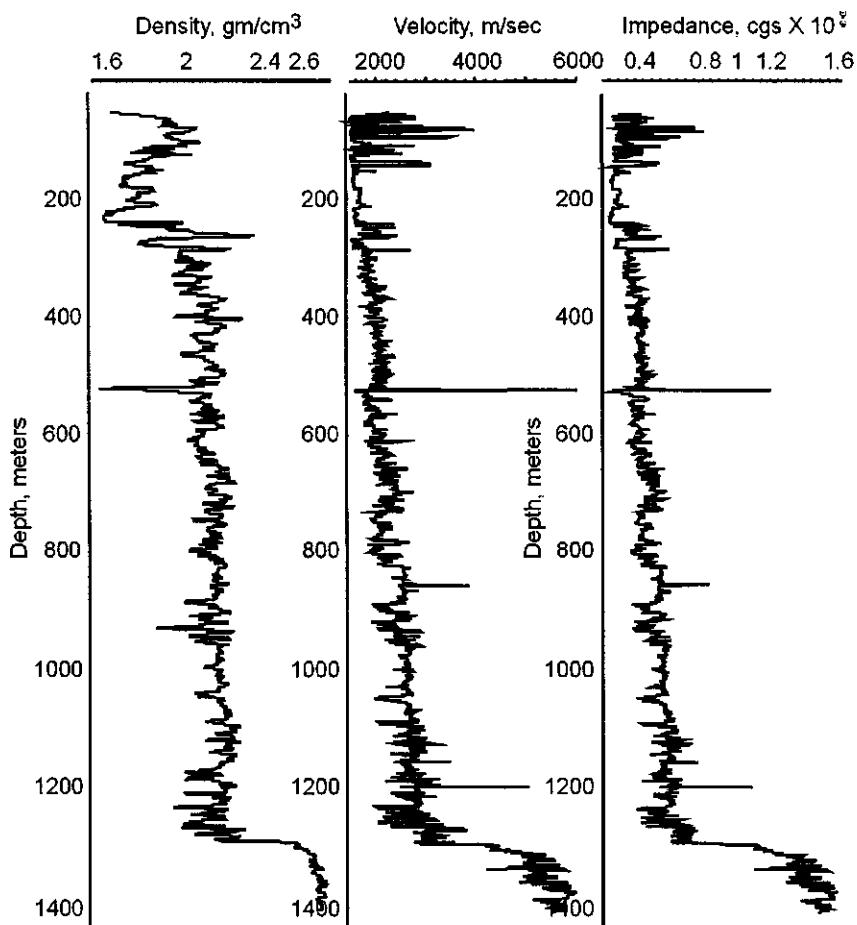


Figure 5.1: Site of the first deep geothermal test hole in the eastern U.S. in the Atlantic Coastal Plain sediments at Crisfield, MD. Left: Density in gm/cm^3 . Middle: Velocity in m/sec. Right Acoustic impedance in cgs units. (CrisfieldLogs)

Table 5.1: Velocity and density of Paleozoic shelf strata and crystalline basement in the Appalachian Valley and Ridge Province in southwestern Virginia. Number of samples in parentheses. Edsall* [69], Kolich [100].

Age	Formation (Number of samples)	Velocity, Km/sec	Density, gm/cm ³	Impedance X 10 ⁶ cgs
Mississippian	Price sandstone (5)	5.38 ± 0.234	2.67 ± 0.03	1.26-1.51
	post-Cloyd claystone (2)	4.75 ± 0.0636	2.72 ± 0.01	1.22-1.30
	Cloyd conglomerate (2)	5.51 ± 0.0283	2.58 ± 0.0	1.37-1.43
	Parrott sandstone (2)	5.58 ± 0.008	2.62 ± 0.0	1.41-1.48
Devonian	Chemung sandstone (4)	4.76 ± 0.138	2.61 ± 0.05	1.12-1.29
	Millboro shale (2)	4.71 ± 0.0141	2.74 ± 0.01	1.26-1.30
Silurian	Keefer sandstone (2)	5.74 ± 0.0283	2.64 ± 0.01	1.46-1.53
	Rose Hill sandstone (4)	5.41 ± 0.185	3.07 ± 0.03	1.60-1.73
	Tuscarora sandstone (4)	5.71 ± 0.0806	2.64 ± 0.01	1.42-1.53
Ordovician	Juniata sandstone (5)	5.05 ± 0.396	2.62 ± 0.03	1.12-1.43
	Martinsburg shale (2)	4.69 ± 0.205	2.70 ± 0.0	1.19-1.31
	Eggleson conglomerate (1)	5.35	2.65	1.42
	Moccasin shale (6)	5.01 ± 0.716	2.71 ± 0.03	0.95-1.51
	Bays sandstone (4)	5.34 ± 0.368	2.68 ± 0.01	1.27-1.53
	Witten limestone (2)	6.03 ± 0.0566	2.68 ± 0.1	1.55-1.63
	Liberty Hall shale (13)	5.07 ± 0.651	2.69 ± 0.02	1.11-1.68
	Lincolnshire limestone (5)	6.30 ± 0.0657	2.70 ± 0.01	1.66-1.73
	Five Oaks limestone (2)	6.38 ± 0.0	2.70 ± 0.0	1.72-1.74
	New Market limestone (2)	6.38 ± 0.0	2.72 ± 0.0	1.71-1.75
	Elway limestone (2)	6.19 ± 0.163	2.68 ± 0.01	1.59-1.69
	Upper Knox dolomite (2)	6.36 ± 0.0636	2.82 ± 0.0	1.71-1.80
	Kingsport dolomite (2)	6.35 ± 0.205	2.79 ± 0.0	1.64-1.81
	Longview limestone (1)	6.62	2.71	1.79
	Chepultepec limestone (2)	6.51 ± 0.0778	2.72 ± 0.0	1.73-1.78
	Chepultepec dolomite (2)	6.61 ± 0.0566	2.83 ± 0.0	1.73-1.88
Cambrian	Copper Ridge sandstone (2)	5.90 ± 0.156	2.71 ± 0.0	1.48-1.63
	Copper Ridge dolomite (2)	6.33 ± 0.007	2.81 ± 0.01	1.71-1.78
	Elbrook dolomite (4)	6.44 ± 0.220	2.80 ± 0.03	1.71-1.89
	Honaker dolomite (2)	6.95 ± 0.0212	2.85 ± 0.01	1.92-1.97
	Honaker dolomite* (4)	7.35 ± 0.238	2.83 ± 0.01	2.01-2.15
	Rome shale* (4)	3.88 ± 0.171	2.67 ± 0.02	0.88-1.09
	Shady dolomite* (3)	6.90 ± 0.0	2.84 ± 0.01	1.73-1.96
	Shady limestone* (2)	6.85 ± 0.0707	2.73 ± 0.01	1.85-1.91
	Erwin sandstone* (4)	5.70 ± 0.408	2.59 ± 0.04	1.32-1.58
	Hampton shale* (2)	5.4 ± 0.0	2.71 ± 0.0	1.43-1.49
	Unicoi sandstone* (2)	6.1 ± 0.141	2.67 ± 0.01	1.52-1.65
Precambrian	Augen gneiss* (2)	6.1 ± 0.141	2.7 ± 0.0	1.51-1.67
	Amphibolite* (2)	5.7 ± 0.0	3.00 ± 0.0	1.62-1.71
	Lynchburg amphibolite* (2)	6.25 ± 0.212	2.97 ± 0.03	1.78-1.90
	Lynchburg gneiss* (4)	5.18 ± 0.263	2.64 ± 0.08	1.22-1.46
	Grenvillian(?) gneiss* (4)	6.05 ± 0.031	2.66 ± 0.04	1.55-1.64

5.3 The Zoeppritz Equations

The Zoeppritz equations [206, 3] describe the amplitudes of reflected and transmitted compressional (longitudinal) and shear (transverse) elastic plane waves when a P -wave or S -wave is incident on a boundary between two media. Both model [199] and field experiments [186] have confirmed the practical application of these equations. They have been extensively discussed in the geophysical literature [182, 84, 3].

The geometry for an incident P -wave is shown in Figure 5.3. In general, two reflected and two transmitted waves result: a reflected longitudinal wave, a reflected transverse wave, a transmitted longitudinal wave, and a transmitted transverse wave. The energy of the incident wave is divided among the four waves in proportions that depend upon the elastic constants and densities of the two media, and on the angle of incidence. There are four equations because there are four boundary conditions that must be satisfied.

Zoeppritz [206] derived equations for the displacement amplitudes of the reflected and refracted waves with respect to the amplitude of the incident plane wave. At the boundary, the vector sum of the displacements associated with the P - and S -waves in the upper medium must equal the vector sum of the displacements associated with the P - and S -waves in the lower medium. Otherwise there would be relative slip between the two media. This requirement results in two equations – one for the horizontal and one for the vertical displacements:

$$\begin{aligned} A_1 \cos \theta_1 - B_1 \sin \lambda_1 + A_2 \cos \theta_2 + B_2 \sin \lambda_2 &= A_0 \cos \theta_1 \\ A_1 \sin \theta_1 + B_1 \cos \lambda_1 - A_2 \sin \theta_2 + B_2 \cos \lambda_2 &= -A_0 \sin \theta_1 \end{aligned}$$

At the boundary, the corresponding stress components in each medium must be equal. This requirement results in two additional equations – one for the normal stresses and one for the tangential stresses:

$$\begin{aligned} A_1 Z_1 \cos 2\lambda_1 - B_1 W_1 \sin 2\lambda_1 - A_2 Z_2 \cos 2\lambda_2 - B_2 W_2 \sin 2\lambda_2 &= \\ -A_0 \rho_1 \lambda_1 \cos 2\lambda_1 \\ A_1 \frac{\beta_1}{\alpha_1} W_1 + B_1 W_1 \cos 2\lambda_1 + A_2 \frac{\beta_2}{\alpha_2} W_2 \cos 2\lambda_2 = A_0 \frac{\beta_1}{\alpha_1} W_1 \sin 2\theta_1 \end{aligned}$$

Thus, these conditions result in a system of four linear equations whose solution results in two reflection and two transmission coefficients for a given angle of incidence θ_1 of a P -wave. The coefficients can be quite sensitive to the angle of incidence, and a relatively small change in the angle of incidence can result in a large change in a coefficient. It is apparent from Figure 5.3 that increasing the angle of incidence of the incident P -wave could eventually result in two critical angles. The first occurs when the angle of refraction of the transmitted P -wave reaches 90 deg; the second occurs when the angle of refraction of the transmitted S -wave reaches 90 deg. Similarly, inspection of a figure for an incident S -wave would show that three critical angles are possible.

As soon as a critical angle is reached, the wave type associated with that critical angle disappears, and the incident energy is distributed among the remaining waves. As soon as the first critical angle is reached, all reflection and transmission coefficients for the remaining wave types become complex. The physical amplitude corresponding to a remaining wave type is the square root of the sum of the squares of the real and imaginary parts of the coefficient. The domain of real reflection coefficients where polarity reversals can be displayed is shown in Figure 5.2.

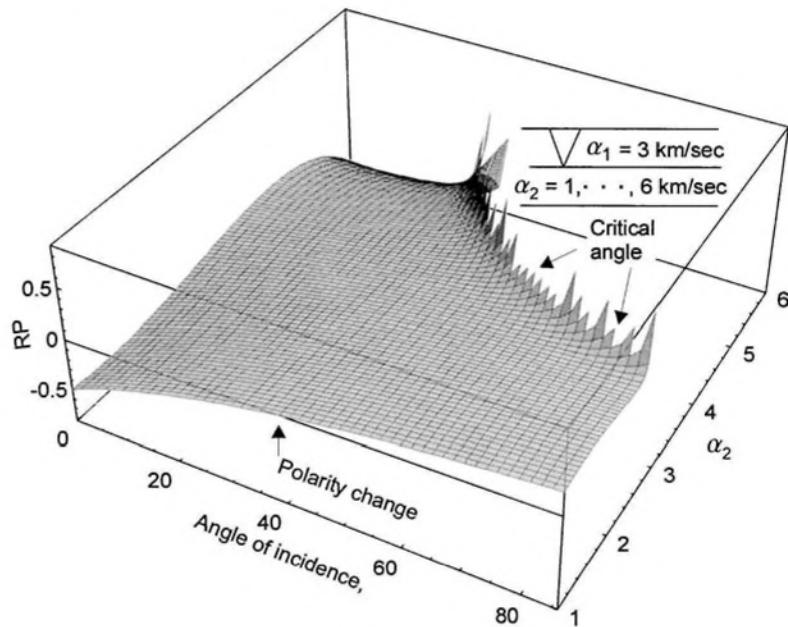


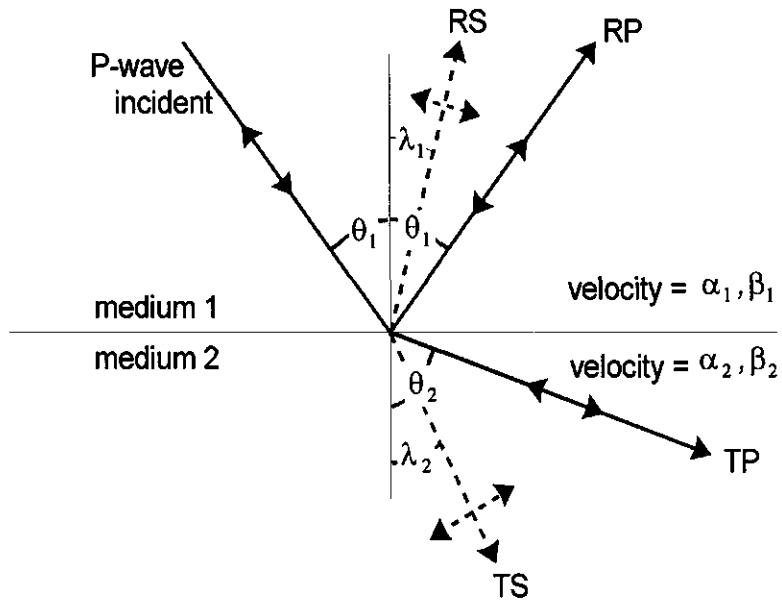
Figure 5.2: Amplitude of reflected P-wave (RP) for various angles of incidence and velocity ratios. The velocity α_1 was assumed to be 3 km/sec; α_2 varied from 1 to 6 km/sec. Polarity reversals where RP=0. For angles of incidence greater than about 30 deg, the reflection coefficients are complex and their amplitudes are given by the modulus of the complex reflection coefficient, as shown in Figure 5.4. (Zoeppritz3DRealRP.nb → ZoeppritzRealRP.cdr → ZoeppritzRealRP.wmf)

The “Zoeppritz equations” are thus

$$\begin{aligned}
 A_1 \cos \theta_1 - B_1 \sin \lambda_1 + A_2 \cos \theta_2 + B_2 \sin \lambda_2 &= A_0 \cos \theta_1 \\
 A_1 \sin \theta_1 + B_1 \cos \lambda_1 - A_2 \sin \theta_2 + B_2 \cos \lambda_2 &= -A_0 \sin \theta_1 \\
 A_1 Z_1 \cos 2\lambda_1 - B_1 W_1 \sin 2\lambda_1 - A_2 Z_2 \cos 2\lambda_2 - B_2 W_2 \sin 2\lambda_2 &= -A_0 Z_1 \cos(2\lambda_1) \\
 A_1 \frac{\beta_1}{\alpha_1} W_1 + B_1 W_1 \cos 2\lambda_1 + A_2 \frac{\beta_2}{\alpha_2} W_2 \cos 2\lambda_2 &= A_0 \frac{\beta_1}{\alpha_1} W_1 \sin 2\theta_1
 \end{aligned}$$

Solutions to these equations are shown in Figure 5.4, where the upper medium has been assigned a constant velocity of 3 km/sec. Poisson’s ratio is 0.25 for both media.

It is clear that the largest changes in amplitude versus angle of incidence occur for a low-velocity medium over a high-velocity medium and for angles of incidence greater than about 40°.



$$\frac{\sin \theta_1}{\alpha_1} = \frac{\sin \lambda_1}{\beta_1} = \frac{\sin \theta_2}{\alpha_2} = \frac{\sin \lambda_2}{\beta_2}$$

Figure 5.3: Nomenclature for reflected and transmitted waves for a *P*-wave incident at a plane boundary. (ZoeppritzIncidentP)

An excellent interactive tutorial about the Zoeppritz equations has been provided by the Consortium for Research in Elastic Wave Exploration Seismology (CREWES) in the form of the “CREWES Zoeppritz Explorer”. This Java applet allows the user to select an incident P-wave or an incident S-wave in either the upper or lower layer, as well as densities and velocities of the upper and lower layer. Results are displayed immediately and continuously as velocity or density values are changed. Both the phase and the magnitudes of the reflection coefficients are displayed. Critical angle(s) are annotated numerically and graphically. The CREWES Zoeppritz Explorer is currently located on the World Wide Web at

<http://www.crewes.org/Samples/ZoepExpl/ZoeppritzExplorer.html>

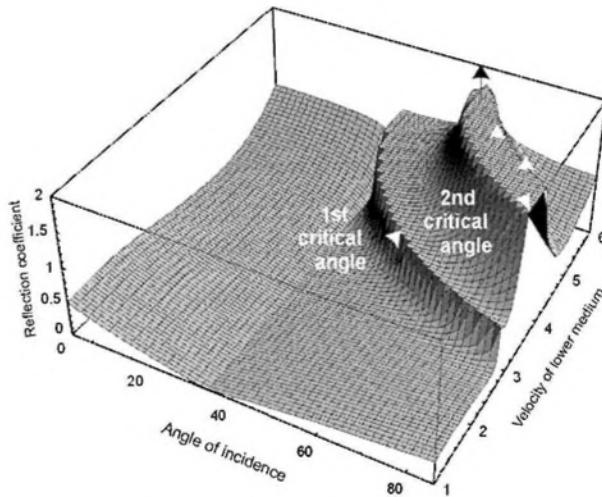


Figure 5.4: Amplitudes of reflection coefficients real or complex, positive or negative, are given by the modulus of the complex reflection coefficient. Velocity α_1 of upper medium is 3 km/sec and velocity α_2 of lower medium varies from 1 to 6 km/sec. (Zoeppritz3DPlotComplexRP.nb → Zoepritz3DPlotComplexRP.cdr → Zoepritz3DPlotComplexRP.wmf)

Physical Significance of a Complex Reflection Coefficient

Plane wave reflection and transmission coefficients for a single interface are independent of frequency. The coefficients apply to a single sinusoid of a single frequency incident at some angle on an interface. Before a critical angle is reached the only phase change possible for the incident sinusoid is a phase shift of 180° , which simply changes the polarity of the sinusoid; however, after a critical angle is reached one of the reflected or transmitted P to S waves exists no more (which one depends on the P-velocity ratio, the S-velocity ratio, the density contrast and the angle of incidence). All reflection coefficients for all remaining reflected and transmitted P or S waves become complex after any critical angle is reached. This simply means that the reflected or transmitted P or S sinusoid is shifted in phase *with respect to the incident sinusoid*. This phase shift is accompanied by a change in the amplitude of the reflected or transmitted sinusoid. Changing the phase of a sinusoid means shifting its peaks one way or the other (toward positive or negative time). Which way to go is simply given by the “time-shifting theorem” and is unambiguous.

Since any wavelet can be decomposed into its component sinusoids, we can use the complex reflection coefficient and thus the phase change, which is the same for each component sinusoid, for any wavelet. In the frequency domain the Fourier spectrum of the original wavelet can be multiplied by the complex reflection coefficient. This adds or subtracts the phase shift to the phase angle

spectrum. The modulus of the complex reflection coefficient scales the original modulus for that same frequency up or down. An inverse Fourier transformation to the time domain then shows the effect on the original wavelet.

A positive phase shift (counterclockwise into quadrant I from the positive real axis, or counterclockwise in quadrant III from the negative real axis) means a shift into negative time; i.e., the wavelet arrives earlier than would be expected. P.N.S. O'Brien [126] noted that

It has been objected that a phase lead, which necessitates a time lead, is physically impossible. There seems to have been no discussion of this point in the literature, but a possible explanation is as follows. When a plane pulse is incident on a boundary it acts as a line source and generates a disturbance which travels along the interface with the velocity of the lower medium. For angles of incidence greater than critical this velocity is greater than the trace velocity of the incident wave and, therefore, the interface disturbance will generate a wave in the upper medium which travels ahead of the reflected pulse predicted by ray theory. Because a plane wave means that the source lies at infinity the interface wave will provide an infinite forerunner to the pulse traveling along the reflected ray path.

In real systems, of course, the source emits curved wavefronts from a local source and in this case a head wave exists. As this always arrives earlier than the reflected pulse the phase lead associated with wide angle reflections merely means that the disturbance between the two events never completely dies away.

For a complex reflection coefficient there can be considerable distortion of the wavelet depending on the phase shift. If the signal to noise ratio is high enough then the observed distortion of the wavelet might be used to advantage. Additional scenarios can be examined by running the *Mathematical* program *ZoeppritzCriticalAngle.nb* on the CD-ROM that accompanies this volume. After running the program the graphics can be animated. A movie can be viewed by loading the *Mathematica* program *ZoeppritzCriticalAngleCC.mov*.

5.4 AVO and Zoeppritz equations in T - X domain

An overview of AVO (amplitude variation with offset) analysis is given by J.P. Castagna and M.M. Backus [34], editors of *Offset-Dependent Reflectivity—Theory and Practice of AVO Analysis*. In their introduction they quoted the following words spoken by W.J. Ostrander [130] at the 52nd Annual International Meeting of the Society of Exploration Geophysicists:

The P -wave reflection coefficient at an interface separating two media is known to vary with angle of incidence. The manner in which it varies is strongly affected by the relative values of Poisson's ratio of

the two media. For moderate angles of incidence, the relative change in reflection coefficient is particularly significant when Poisson's ratio differs greatly between the two media.

Theory and laboratory measurements indicate that gas sands tend to exhibit abnormally low Poisson's ratios. Embedding a low velocity gas sand into sediments having 'normal' Poisson's ratios should result in an increase in reflected P -wave energy with angle of incidence. This phenomenon has been observed on conventional seismic data recorded over known gas sands.

In reflection seismology, a CMP gather is the most suitable form of data for observation of the variation of reflection amplitude with offset. The recording geometry used to obtain a CMP (common midpoint) gather is well known. The Zoeppritz equations [206, 187] are normally written as a system of four simultaneous equations that involve angles of incidence, of reflection, and angles of transmission for the various types of P - and S -waves, as shown in Figure 5.5. The recording geometry for the acquisition of CMP (common midpoint) data suggests that the angles in these equations can be redefined as a function of the recording geometry, i.e., offset distance and depth to reflector. The Zoeppritz equations were accordingly converted into the time-offset domain by Demirbag and Çoruh [57] using the common-mid-point (CMP) geometry, Snell's law, and zero-offset two-way traveltimes. Once this has been done, the equations can be solved in the usual forward manner to examine the variation of amplitude with offset, but now as a function of reflection traveltimes and source-receiver offset instead of angle of incidence of the P -wave. The depths to reflectors can be defined by zero-offset two-way traveltimes and average velocities. Equally important, however, the inverse problem can be attempted, that of solving the equations for the rock density ratio and rock velocities, given the recording geometry and reflection amplitudes as measured on real data.

Forward modeling is used to observe the response of a physical system, that is, to view the synthetic traces that comprise an AVO gather. The modeled physical process is that of energy partition at a boundary that separates two elastic layers.

Conversion of the Zoeppritz equations to the time-offset domain

The model reviewed here from Demirbag [56] and Demirbag and Çoruh [57, 58] is a two-layer elastic medium with a single horizontal interface. Each layer is homogeneous and isotropic. In Figure 5.5, α_1, β_1 , and ρ_1 represent the P - and S -wave velocity and density for the upper layer, respectively, and α_2, β_2 and ρ_2 represent the same parameters for the lower layer. Assuming a plane P -wave to be incident on the boundary in the upper layer, the Zoeppritz equations predict the four types of waves that are generated at the boundary by the energy partitioning process that results in reflected and refracted P - and S -waves. The angle of incidence of the P -wave is θ_i , θ_r and ϕ_r are the P - and S -wave reflection angles, and θ_t and ϕ_t are the P - and S -wave

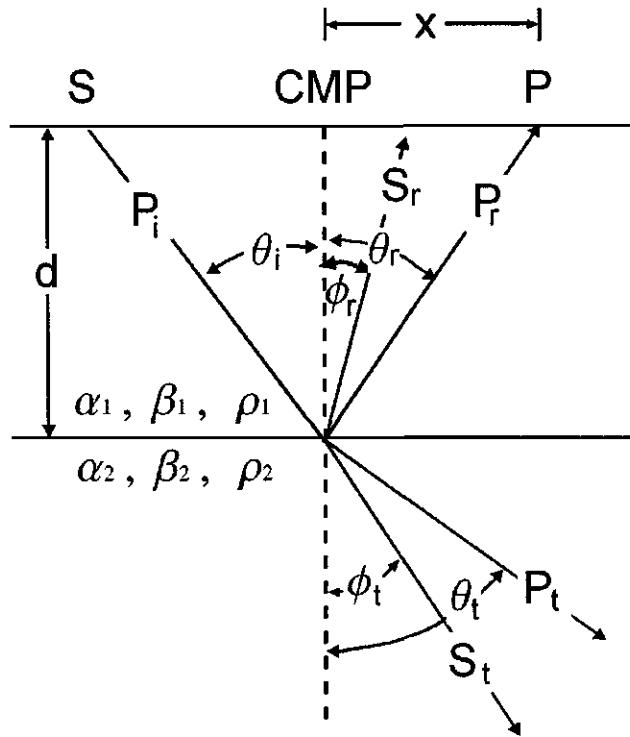


Figure 5.5: P -wave CMP (common midpoint) geometry and energy partition.
Demirbag-avo.cdr \Rightarrow avol.wmf

transmission angles, respectively.

From Figure 5.5, the time-offset relation for P -waves is

$$t_x = \sqrt{t_0^2 + \frac{4x^2}{\alpha_1^2}}$$

where t_x is the traveltime from S to P , t_0 is the zero-offset two-way travel time defined by $t_0 = 2d/\alpha_1$, and d and x are the depth of the reflector and half the source-receiver offset, respectively. In addition to the time-offset equation, Snell's law

$$\frac{\sin \theta_i}{\alpha_1} = \frac{\sin \theta_r}{\alpha_1} = \frac{\sin \theta_t}{\alpha_2} = \frac{\sin \phi_r}{\beta_1} = \frac{\sin \phi_t}{\beta_2} = \text{constant}$$

and the ratio of densities $\Delta\rho = \rho_2/\rho_1$ are used to convert the Zoeppritz equations into the time-offset domain. In the implementation, densities were assigned using Gardener's relation [73, 56]. The conversion of the Zoeppritz equations

into the CMP gather time-offset domain is achieved by replacing the trigonometric functions with their corresponding equalities given in terms of the P - and S -wave velocity of lower and upper layers, two-way traveltime, and CMP ray geometry. Referring to Figure 5.5 and using Snell's law and the general trigonometric relations, the definitions given in Table 5.2 and Table 5.3 can be substituted into the Zoeppritz equations [57, 58, 59].

Table 5.2: Notation for conversion of Zoeppritz equations to time-offset domain.

Reflected waves	Transmitted waves
$\sin \theta_r = \frac{2x}{\alpha_1 t_x}$	$\sin \theta_t = \frac{\alpha_2}{\alpha_1} \sin \theta_r = \frac{2x\alpha_2}{\alpha_1^2 t_x}$
$\cos \theta_r = \frac{2d}{\alpha_1 t_x} = \frac{t_0}{t_x}$	$\cos \theta_t = \sqrt{1 - \frac{4x^2 \alpha_2^2}{\alpha_1^4 t_x^2}}$
$\sin \phi_r = \frac{\beta_1}{\alpha_1} \sin \theta_r = \frac{2x\beta_1}{\alpha_1^2 t_x}$	$\sin \phi_t = \frac{\beta_2}{\alpha_1} \sin \theta_r = \frac{2x\beta_2}{\alpha_1^2 t_x}$
$\cos \phi_r = \sqrt{1 - \frac{4x^2 \beta_1^2}{\alpha_1^4 t_x^2}}$	$\cos \phi_t = \sqrt{1 - \frac{4x^2 \beta_2^2}{\alpha_1^4 t_x^2}}$
$\sin 2\theta_r = \frac{4xt_0}{\alpha_1^2 t_x^2}$	$\cos 2\theta_t = \frac{4x\alpha_2}{\alpha_1^2 t_x} \sqrt{1 - \frac{4x^2 \alpha_2^2}{\alpha_1^4 t_x^2}}$
$\cos 2\phi_r = 1 - \frac{8x^2 \beta_1^2}{\alpha_1^4 t_x^2}$	$\cos 2\phi_t = 1 - \frac{8x^2 \beta_2^2}{\alpha_1^4 t_x^2}$
$\sin 2\phi_r = \frac{4x\beta_1}{\alpha_1^2 t_x} \sqrt{1 - \frac{4x^2 \beta_1^2}{\alpha_1^4 t_x^2}}$	$\sin 2\phi_t = \frac{4x\beta_2}{\alpha_1^2 t_x} \sqrt{1 - \frac{4x^2 \beta_2^2}{\alpha_1^4 t_x^2}}$

The Zoeppritz equations in the time-offset domain have the following form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} R_p \\ R_s \\ T_p \\ T_s \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad (5.7)$$

where the solution vector that contains R_p , R_s , T_p , and T_s yields the amplitudes of the reflected P - and S -waves and transmitted P - and S -waves, respectively, for an incident P -wave of unit amplitude. The entries of the matrices are combinations of the known variables (x and t_0) and model parameters (α_1 , β_1 , α_2 , β_2 , and $\Delta\rho$). The elements of the matrices in Equation (5.7) are shown in Table 5.3. An example of using Equation (5.7) with the appropriate substitutions given in Table 5.3 is shown in Figure 5.6. For this example, the polarity of the reflected P -wave is always positive. A second example shown in

Table 5.3: Definitions of a and b . Refer to Equation (5.7).

$a_{11} =$ $2x/\alpha_1 t_x$	$a_{12} =$ $\sqrt{(1 - a_{22}^2)}$	$a_{13} = a_{11} \frac{\alpha_2}{\alpha_1}$	$a_{14} = \sqrt{(1 - a_{24}^2)}$
$a_{21} = -\frac{t_n}{t_x}$	$a_{22} = a_{11} \frac{\beta_1}{\alpha_1}$	$a_{23} = -(1 - a_{13}^2)^{1/2}$	$a_{24} = -a_{11} \frac{\beta_2}{\alpha_1}$
$a_{31} =$ $-2a_{11}a_{21}$	$a_{32} =$ $-a_{41} \frac{\alpha_1}{\beta_1}$	$a_{33} =$ $-a_{11}a_{23} \frac{2\Delta\rho\beta_2^2}{\beta_1^2}$	$a_{34} =$ $-\frac{\Delta\rho\alpha_1\beta_2}{\beta_1^2}(1 - 2a_{24}^2)$
$a_{41} =$ $1 - 2a_{22}^2$	$a_{42} =$ $-a_{12}a_{22} \frac{2\beta_1}{\alpha_1}$	$a_{43} =$ $-\frac{\Delta\rho\alpha_2}{\alpha_1}(1 - 2a_{24}^2)$	$a_{44} =$ $a_{14}a_{24} \frac{2\beta_2\Delta\rho}{\alpha_1}$
$b_1 = -a_{11}$	$b_2 = a_{21}$	$b_3 = a_{31}$	$b_4 = -a_{41}$

Figure 5.7 illustrates that a polarity change can occur within the CMP-gather. For these examples, Equation (5.7) was solved using the *Mathematica* function `LinearSolve`. The reader can explore other AVO scenarios using the *Mathematica* programs `avoExample.nb` found on the CD-ROM that accompanies this volume.

The inverse problem, on the other hand, attempts to recover the dependent variables of forward modeling. Input data to the inversion process are in the form of reflection amplitudes that are measured on seismic traces, such as those shown in Figure 5.6 or Figure 5.7. The inverse problem is to find the model parameters (velocities and density ratio) given the source-receiver offsets, zero-offset two-way travel time, and the amplitude variation with offset (AVO) data. The densities of the media appear in the ratio of ρ_2/ρ_1 in the Zoeppritz equations; therefore, only the ratio of densities can be estimated. The solutions from an inversion might make no physical sense unless some physical constraints are imposed on the solutions, and therefore they are constrained by allowing values for rock velocity and density to be limited to the range of values that are physically possible. The velocities must be consistent with Poisson's ratio falling in the range 0.05 – 0.45. The formulation of the inversion method for AVO data is based on pre-critical reflection data. Therefore, the maximum offset should be long enough to observe a significant change in the amplitude data while remaining in the pre-critical offset range.

When the offset-to-depth ratio ≥ 1.5 , the model parameters were recovered

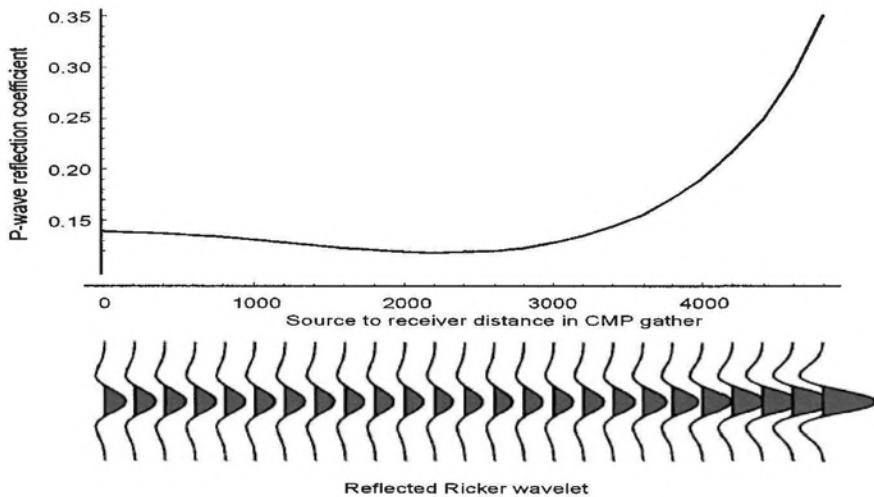


Figure 5.6: Example of amplitude variation with source-receiver distance using Equation (5.7) and Table 5.3. Model parameters are $\alpha_1 = 4000$ m/s, $\beta_1 = 2000$ m/s, $\rho_1 = 2465$ kg/m³, $\alpha_2 = 5000$ m/s, $\beta_2 = 2500$ m/s, and $\rho_2 = 2607$ kg/m³. Same parameters as [56, Figure 2, p. 13]. Reflection amplitudes are all positive. `avoExample1.nb`⇒ Demirbag-
`avo.cdr`⇒`avoExample1.wmf`

well in contrast to the results of ratios < 1.5. As a rule-of-thumb, model tests indicate that an offset-to-depth ratio of about 2 is necessary to obtain satisfactory results, depending upon the contrast in the layer seismic parameters. AVO data with an offset-to-depth ratio < 1 gives unsatisfactory results. In addition to a small offset-to-depth ratio, noise is the most disturbing factor in the inversion of AVO data. When the S/N and offset-to-depth ratios are small, noise obscures the change in reflection amplitude with offset and the parameters are estimated with larger errors. The mathematical details of the inversion process are beyond the scope of this volume. For details the reader is referred to Demirbag [56], Demirbag and Çoruh [57, 58], and Demirbag, Çoruh, and Costain [59].

Demirbag [56] reported results of AVO inversions using real multifold seismic

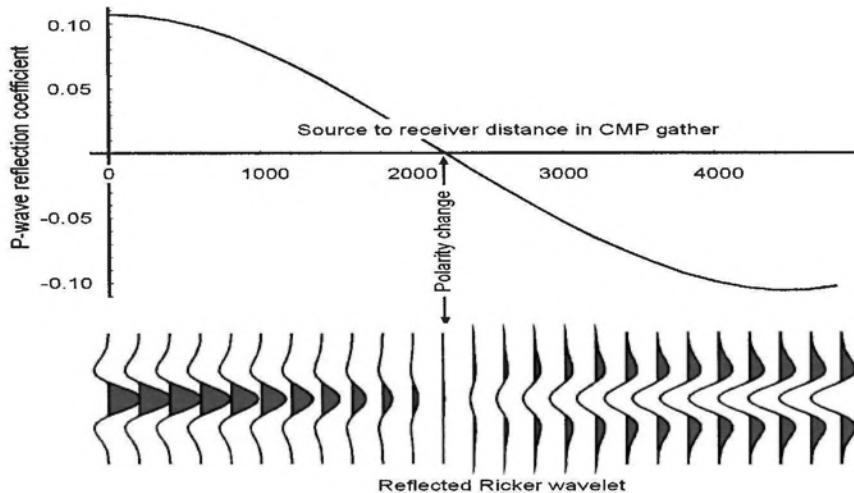


Figure 5.7: Example of amplitude variation with source-receiver distance using Equation (5.7) and Table 5.3. Model parameters are $\alpha_1 = 4000$ m/s, $\beta_1 = 1750$ m/s, $\rho_1 = 2465$ kg/m³, $\alpha_2 = 4750$ m/s, $\beta_2 = 3000$ m/s, and $\rho_2 = 2573$ kg/m³. Same parameters as [56, Figure 4, p. 15]. Reflection amplitudes are all positive for near distances and all negative for far distances. `avoExample2.nb` \Rightarrow `Demirbag-avo.cdr` \Rightarrow `avoExample2.wmf`

data. Velocities and density ratio were estimated from AVO data using common offset stacked CMP gathers of vibroseis data acquired by Virginia Tech on the Atlantic Coastal Plain near Charleston, SC [56, p. 84]. An excellent fit was obtained between the observed and calculated AVO data [56, Figure 23, p. 90]. Demirbag [56, Fig. 24, p. 91] also obtained consistent results for adjacent CMP gathers and reported on the distribution of inversion results in space.

5.5 Synthetic Seismograms

Synthetic seismograms (“synthetics”) are not direct recordings of seismic data. They are either generated directly from theoretical considerations, or indirectly from some other kind of data such as well logs. Comparison of synthetic seismo-

grams with actual seismograms is a widely-used method to test our understanding of the way the earth has responded, or might respond, to various kinds of short-term elastic and inelastic deformation. Synthetic seismograms are widely used to tie seismic data to subsurface geology. Propagation of seismic energy in the earth is clearly not described by plane wave fronts, yet this assumption is valid at a distance of about ten wavelengths from the source, and plane wave theory is considerably easier to formalize than curved wave fronts. The earth is neither perfectly elastic, isotropic or homogeneous, yet the assumption of elastic, isotropic, homogeneous media is easier to implement than the real inelastic anisotropic, inhomogeneous case. A disturbance propagating away from the source changes shape because of intrinsic damping (internal friction) that results in the gradual attenuation of higher-frequency Fourier components and therefore a gradual change in shape of the source wavelet. In spite of the simplistic assumptions commonly made about the propagation of seismic waves in the earth, the successful prediction of how the earth will behave is quite remarkable. As noted by Levin [110]

In view of the known complexity of real earth sections, why are the simple methods of exploration seismologists so successful in picturing what lies beneath our feet?

But marvelous as today's processing is, the processing can never be perfect, for it is based on imperfect models. By any reasonable standard, the success of reflection seismology is remarkable but the success is not completely reasonable.

To an excellent first approximation, the seismogram recorded at the surface of the earth is simply the algebraic sum of reflected source wavelets $w(t)$ as they return at different times from each subsurface reflector. As shown in Figure 5.8, the reflection from the shallowest reflector is the first to be recorded at the surface after a two-way time delay of τ_1 sec. The amplitude and possibly the polarity of $w(t)$ will be changed by the *reflection coefficient* associated with the reflectors. Thus, the event recorded at the surface is $R_1(\tau_1)w(t - \tau_1)$. The next reflected event originates from the next reflector and is $R_2(\tau_2)w(t - \tau_2)$, and so forth. The reflection process can therefore be written as a discrete summation of reflected wavelets each multiplied by the appropriate reflection coefficient,

$$o[t] = \sum_{i=1}^n R_i[\tau_i]w[t - \tau_i] \quad (5.8)$$

where $R_i[\tau_i]$ is the reflection coefficient associated with the two-way traveltime τ_i to the i_{th} reflector in a model with a discrete number of layers. This summation we recognize as a convolution, also called a superposition.

If instead of a discrete number of layers we have a continuous change of velocity with depth, then Equation (5.8) becomes a continuous convolution:

$$o(t) = \int_{-\infty}^{\infty} r(\tau)w(t - \tau)d\tau$$

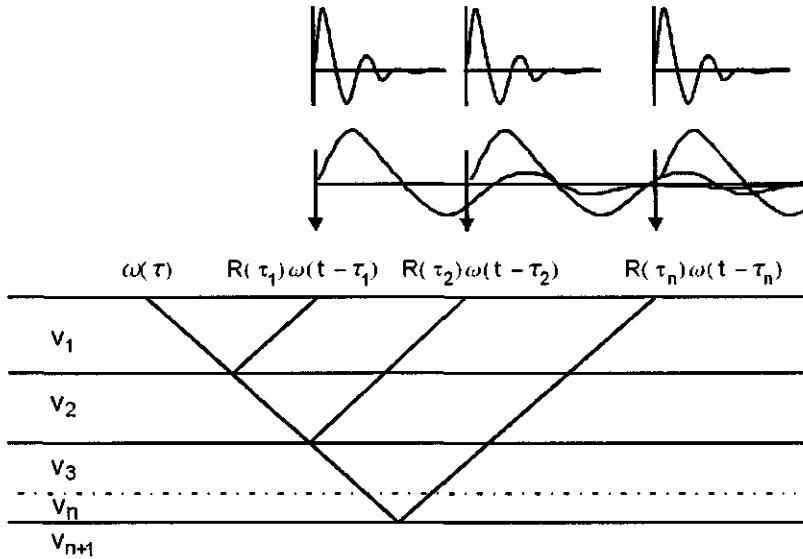


Figure 5.8: Peterson model. Reflection seismogram consists of many reflected source wavelets. If the layers are thick enough (or the wavelets of short enough duration), reflections will not interfere (top row). In practice the large number of layers results in constructive and destructive wavelet interference (second row). Reflection seismogram is the simple algebraic superposition of the overlapping reflected wavelets. (petmod.cdr → petmod.wmf)

and $r(t)$ is now a continuous *reflectivity function*.

For vertical incidence, the amplitude of the source wavelet reflected from an interface defined by a change in acoustic impedance (rock density times velocity) is given by the reflection coefficient R :

$$R = \frac{\rho_2 v_2 - \rho_1 v_1}{\rho_2 v_2 + \rho_1 v_1} \quad (5.9)$$

where v_1 and ρ_1 are the rock velocity and density, respectively of the upper layer, and v_2 and ρ_2 are the velocity and density of the lower. The product ρv is called the *acoustic impedance*.

Equation (5.9) comes from a solution of the Zoeppritz equations [206, 181], which describe the amplitudes of reflected and refracted plane waves for an incident plane wave at an arbitrary angle of incidence on an interface that separates rock types of different acoustic impedance. Equation (5.9) is the result of solving the Zoeppritz equations for normal incidence on the reflecting interface. A more elementary derivation of (5.9) can be found on page 195. The Zoeppritz equations are discussed further in Chapter 5.3 and a *Mathematica* computer program `avoExample` for their solution with examples in the time-offset domain can be found on the CD-ROM with this volume.

The magnitude of the reflection coefficient depends upon the angle of incidence at the interface, but is independent of the frequency content of the source wavelet. The coefficient is a *plane wave reflection coefficient*. If the source wavelet is incident on the interface at normal incidence, then the numerical value of the reflection coefficient is given by Equation (5.9). Values at other angles of incidence can be determined by solution of the Zoeppritz equations, a set of four linear equations [206, 181].

Many aspects of the reflection process were summarized for the interpreter of reflection seismic data in an insightful paper by Sengbush, Lawrence, and McDonal [161, 160]. We follow here their general presentation and elaborate where appropriate. The mathematical model used by Sengbush et al. was first defined by Peterson et al. [136]. The Peterson model is characterized by the following properties:

1. The layered model earth is transversely isotropic, and can be described in the vertical z -direction by a continuous velocity function $v(z)$. The density function $\rho(z)$ is related to $v(z)$ by

$$\rho(z) = k v^m(z)$$

where k and m are constants.

2. The source wavelet propagates as a plane wave and only in the z -direction, and is incident on each interface at right angles. Reflections can be considered to result from velocity changes only if a certain relationship between density and velocity is assumed. This relationship is discussed below.
3. Only primary reflections are included in the model -all types of "noise" including ground roll (surface waves), ghosts, and multiples are excluded.
4. The source wavelet is time-invariant; i.e., its shape does not change with traveltime, and its amplitude and/or polarity change only because of the magnitude and sign of the reflection coefficient. There is no attenuation as a result of internal friction or spherical divergence.

Peterson called the output of this model the *synthetic seismogram*, generally referred to as a *synthetic*. In spite of the assumptions inherent in the mathematical model, the synthetic is often remarkably similar to the actual field data.

The source wavelet $w(t)$ describes the disturbance at or near the surface of the earth that is propagated outward and downward, where it is reflected back up and recorded at some distance from the source. The reflectivity function $r(t)$ describes the continuous sequence of reflection coefficients. Each contrast in rock type results in a portion of the energy being reflected back to the surface and some being transmitted to deeper layers.

Although only the transversely isotropic model is considered in this volume. The reader is referred to excellent reviews and discussions by Vossler [185] and Cormier and Mellen [44] and references therein. Most rocks are transversely

isotropic to some extent [185] with the horizontal velocity component greater than the vertical velocity. This means that the drill will commonly encounter an exploration target at a shallower depth than predicted by simple conventional seismic velocity measurements obtained from just a horizontal recording profile.

The simple transversely isotropic model has been shown to be remarkably useful for predicting how the real earth behaves. It is therefore appropriate to examine the mathematical model together with its limitations.

5.6 The Reflectivity Function

If well control is available, the velocity distribution $v(z)$ in the earth can be obtained from *sonic logs*, which are continuous logs of velocity and density versus depth. For an n -layered model of discrete layers, the reflection coefficient R associated with an interface separating two layers is

$$R = \frac{\rho_2 v_2 - \rho_1 v_1}{\rho_2 v_2 + \rho_1 v_1} \quad (5.10)$$

If the contrast in acoustic impedance ρv across the interface is small, then R can be written as

$$R = \frac{\rho_2 v_2 - \rho_1 v_1}{\rho_2 v_2 + \rho_1 v_1} = \frac{\rho_2 v_2 - \rho_1 v_1}{2\rho_1 v_1 + \Delta(\rho v)} \approx \frac{\Delta(\rho v)}{2\rho_1 v_1} \quad (5.11)$$

where $\rho_2 v_2$ has been replaced by $\rho_1 v_1 + \Delta(\rho v)$.

If $\Delta\rho v$ is small compared to $\rho_1 v_1$, which will be the case for a continuous velocity log (CVL), then Equation (5.11) can be written

$$R \approx \frac{1}{2} \frac{\Delta(\rho v)}{\rho v} \approx \frac{1}{2} \Delta \log(\rho v) \quad (5.12)$$

where $\Delta \log(\rho v)$ is read “change in the log of ρv ”.

If the density is related to the velocity by the general expression

$$\rho = k v^m$$

where k and m are constants, then the reflectivity of the layered medium is related only to changes in velocity because the expression for acoustic impedance becomes

$$\rho v = k v^{m+1}$$

and (5.12) takes on the form

$$R \approx \frac{1}{2} \Delta \left[\log(k v^{m+1}) \right] = \frac{1}{2} \left[\Delta \log k + (m+1) \Delta \log v \right] \approx \frac{m+1}{2} \Delta \log v \quad (5.13)$$

Thus, if the density is related to the velocity by $\rho = k v^m$, then the reflection coefficient is proportional to a change in the log of the velocity.

If the velocity is continuous, then the reflectivity is continuous, and the set of reflection coefficients becomes the continuous *reflectivity function* designated $r(t)$. The reflectivity function $r(t)$ can be developed from (5.13) as follows.

$$\frac{R}{\Delta t} \approx \frac{1}{2} \frac{\Delta \log v}{\Delta t}$$

Taking the limit as $\Delta t \rightarrow 0$,

$$\begin{aligned} r(t) &\approx \lim_{\Delta t \rightarrow 0} \left(\frac{R}{\Delta t} \right) \approx \frac{1}{2v} \left(\frac{dv}{dt} \right) \\ &\approx \frac{1}{2} \frac{d[\log v(t)]}{dt} \end{aligned} \quad (5.14)$$

Thus, the *reflectivity function* $r(t)$ is the derivative of the velocity function where the *velocity function* is defined as the log of velocity versus time.

To illustrate the use of Equations (5.10) and (5.14) we use the density and sonic logs that were obtained from the first deep geothermal test hole in the eastern United States at Crisfield, MD, and shown in Figure 5.1 on page 202. The sampling interval for each log was one foot. The reflection coefficients obtained from (5.10) are shown in Figure 5.9a. Those obtained from (5.14) are shown in (b). Neither log was smoothed. A Ricker wavelet was convolved with each reflectivity sequence and the resulting seismograms are shown in (c). The agreement is generally quite good except at a few depth intervals where apparently the approximation fails. The good comparison means that if a density log is not available then the sonic log can be used to determine an approximation to the reflectivity function.

5.7 Velocity functions

Here we follow the terminology used by Sengbush et al. [161]. By definition, their “velocity function” $w(t)$ is the log of the velocity as a function of time:

$$w(t) = \log v(t) \quad (5.15)$$

and the reflectivity function $r(t)$ is the derivative of this:

$$r(t) = \frac{dw(t)}{dt} \quad (5.16)$$

We will show that only three kinds of velocity functions (Figure 5.10) and combinations of these are necessary to approximate any velocity distribution in the earth. These are:

1. Impulse. The velocity function impulsively changes from one value to another and then immediately returns to the former value. Mathematically, the velocity function $w(t)$ is therefore $w(t) = \delta(t)$. The reflectivity function associated with this velocity function is, by Equation (5.16),

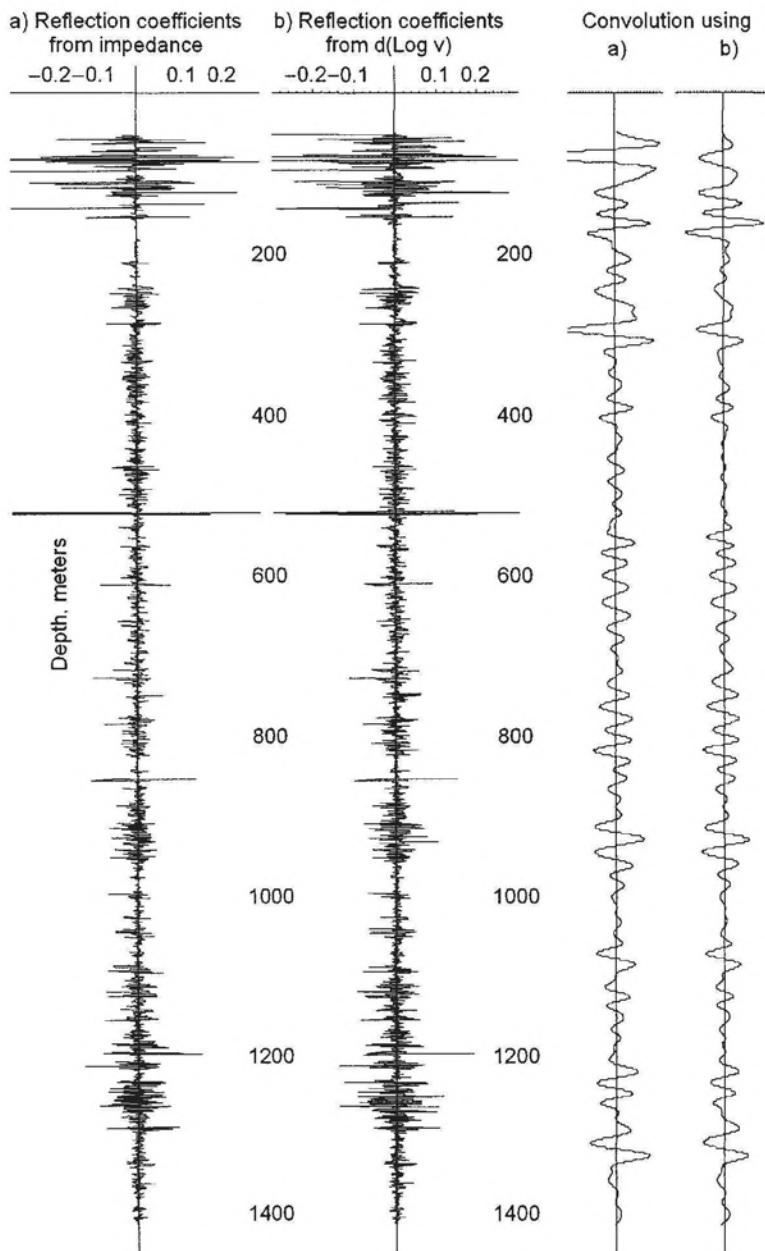


Figure 5.9: (a) Reflection coefficients from (5.10). (b) Reflection coefficients from (5.14). (c) Comparison of the two convolutions using reflection coefficients of (a) and (b). Source wavelet used for each convolution was a Ricker wavelet. The agreement is excellent. (SyntheticSeismogram.nb,cdr, \Rightarrow SengbushLogsExample.wmf)

$r(t) = dw(t)/dt = \delta'(t)$, as shown in Figure 5.11. This is a thin bed (thin with respect to two-way traveltime, not layer thickness). The thin layer can be a high- or a low-velocity layer. If it is low velocity, then the negative of the derivative of the source wavelet is reflected back to the surface.

2. Step. The velocity function $w(t)$ suddenly changes from one value to another. This is the single, isolated interface. The step velocity function changes the amplitude and possibly the polarity of the source wavelet, but not its shape. Mathematically, $w(t) = u(t)$.
3. Ramp. The velocity function $w(t)$ increases or decreases linearly and continuously with time. Mathematically, the ramp velocity function is

$$w(t) = \frac{A}{\Delta t} t u(t)$$

where $A/\Delta t$ is the change in the velocity function $w(t)$ per unit of two-way vertical travelttime. All three velocity functions are shown in Figure 5.10. We now examine these in more detail.

5.7.1 Impulse (thin bed)

The impulse velocity function is $w(t) = \delta(t)$. The reflection coefficients associated with this velocity function are, by Equation (5.9), two equal coefficients of opposite polarity, as shown in Figure 5.11. Use of the word “thin” means thin with respect to two-way travelttime, not layer thickness.

Assume a geologic layer of thickness two sample intervals (two-way time). A wavelet is reflected from the top of the thin bed and two sample intervals later followed by the wavelet reflected from the bottom, but reversed in polarity. The algebraic superposition of the two interfering wavelets results in the derivative of the source wavelet, returned as a single event to the surface. The differentiation is shown graphically in Figure 5.11. Thus, if a source wavelet is reflected from a thin bed then the derivative of the wavelet—a change in shape—is returned to the surface. Clearly, if the bed is thick with respect to travelttime, then the reflection from the top of the bed will be recorded without interfering with the reflection from the bottom, and we get the second type of velocity function, the single interface or step function.

5.7.2 Step

The mathematical expression for the single interface velocity function is proportional to the step function $u(t)$, where $u(t)$ is defined as

$$\begin{aligned} u(t) &= 1 \text{ for } t \geq 0 \\ &= 0 \text{ for } t < 0 \end{aligned}$$

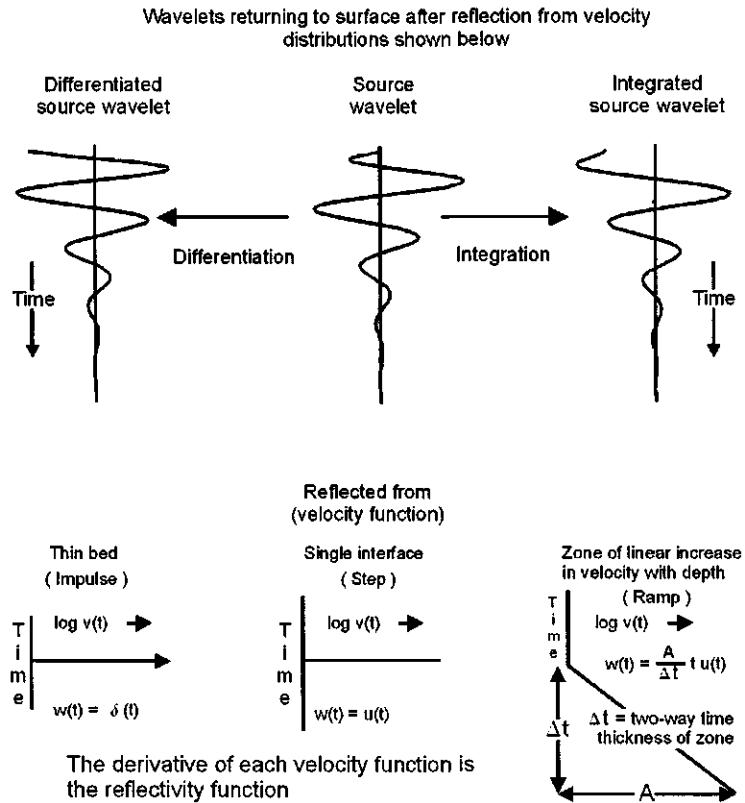


Figure 5.10: The three types of velocity functions $\log v(t)$ from which any velocity distribution $v(z)$ in the earth can be approximated. Left: thin bed. Middle: single interface. Right: continuously reflecting zone within which $v = v_0 + az$. For the latter, the reflection coefficient is constant as a function of time within the zone; i.e., $r(t) = \text{constant}$. Note that, although the geology is quite different for each of the three types of velocity functions, only one single wavelet event is reflected back and recorded at the surface from each model. Close inspection of the wavelets in the above figure shows that the "differentiated source wavelet" is enriched in higher frequencies with respect to the source wavelet, and the "integrated source wavelet" is enriched in lower frequencies. Although this is difficult to recognize in the time domain, it is easily confirmed in the frequency domain by examination of the ratios of the Fourier transforms of the wavelets. The information content contained in the shape of the wavelet is therefore of considerable importance if the geology is to be understood. (velfunc.cdr → velfunc.wmf)

This describes reflection from the top of a very thick bed (no bottom). If the bed is of finite two-way time thickness τ then as $\tau \rightarrow 0$ the sum of $u(t)$ and $-u(t-\tau)$ differentiates the waveform incident on the bed of finite thickness. The derivative of the step function is the Dirac delta function $\delta(t)$. Moving to the left in Figure 5.10, therefore, corresponds to differentiation. Thin beds can often be recognized by changes in instantaneous frequency or changes in wavelet shape. Marangakis [116] used an unconventional method called “integrated energy spectra” to detect the edges of a thin lens as well as the direction of thinning. Recognition of the lens was almost impossible in the time domain but was well defined by the integrated energy spectra, as shown in Figure 5.12. Bryan [29] investigated thin-bed resolution using cepstrum analysis.

5.7.3 Ramp

If the velocity increases linearly with depth, then the log of the velocity increases linearly with time. A linear increase of velocity with depth is

$$v(z) = v_0 + az$$

where v_0 and a are constants. Thus,

$$\frac{dz}{dt} = v_0 + az$$

from which

$$\begin{aligned} \frac{dz}{v_0 + az} &= dt \\ \int \frac{dz}{v_0 + az} &= \int dt \\ \frac{1}{a} \log(v_0 + az) &= t + \text{constant} \end{aligned}$$

When $t = 0$, then $z = 0$ so the constant of integration is $(1/a) \log v_0$. Then

$$\frac{1}{a} \log(v_0 + az) = t + \frac{1}{a} \log v_0$$

and

$$\log(v_0 + az) = at + \log v_0$$

If $v = v_0 + az$, then

$$\log v = at + \log v_0$$

or

$$w(t) = \log v(t) = \log v_0 + at$$

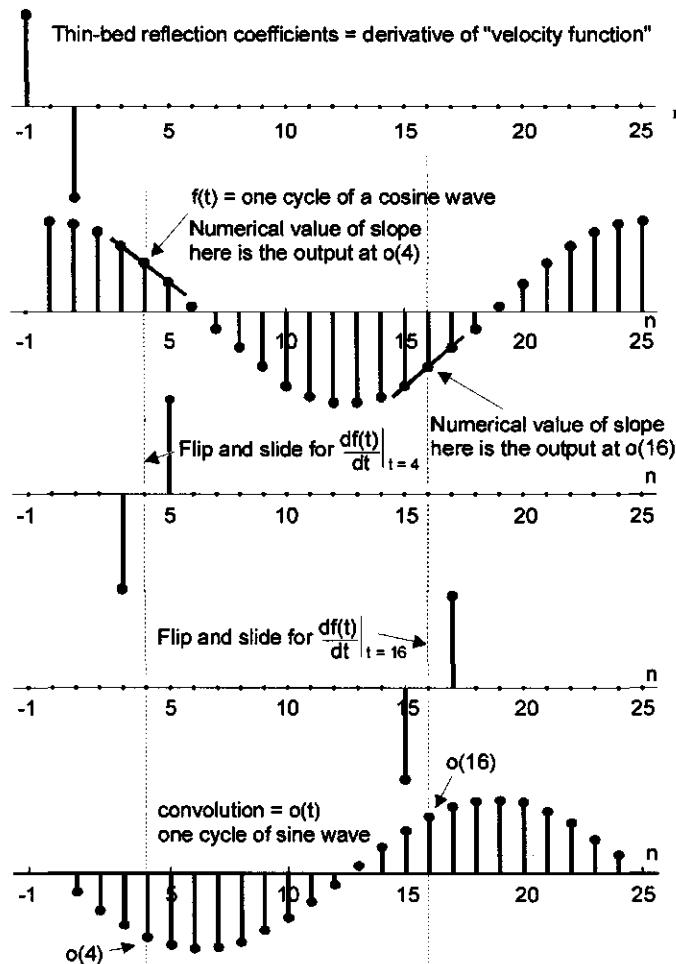


Figure 5.11: Thin bed. The “velocity function” is $\delta(t)$. The reflection coefficients associated with $\delta(t)$ are $\delta'(t)$ (top). Functions to be convolved are $\delta'(t)$ and one cycle of a cosine wave. Examples of computing the derivative of the cosine wave, at $f(4)$ and $f(16)$, are shown. If $\delta'(t)$ is flipped about $t = 0$ and slid to the desired output time for the complete convolution, the result will be the derivative of the cosine wave; i.e., a negative sine wave. Any source wavelet that is reflected from a thin bed is differentiated. Reflection returned to the surface is the derivative of the incident source wavelet. (ma, cd → thinbed)

which shows that the log of the velocity as a function of (one-way) time, i.e., the velocity function $w(t)$ is a linear function of time; i.e., a ramp. The constant a can be either positive or negative to accommodate a velocity increase or decrease.

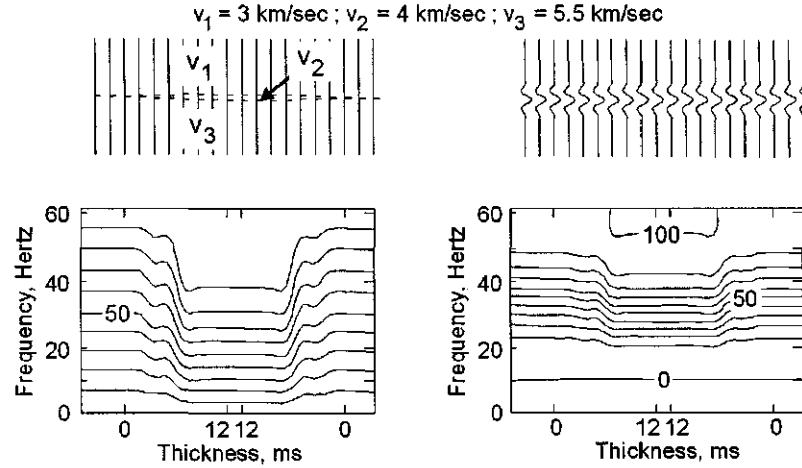


Figure 5.12: Thin-bed detection by integrated energy spectra. (a) Lefthand column: The reflectivity function only above the contour plot of integrated energy patterns. (b) Right-hand column: Convolution of the reflectivity function with the source wavelet plotted above the corresponding integrated energy patterns. The convolution smooths the reflectivity function but the lens is still clearly defined. Recognition of the lens is almost impossible in the time domain. The contours clearly indicate not only the presence of the thin bed but also the direction of thinning. Reproduced from Marangakis et al. [116]. (Marangakis.cdr → Marangakis.wmf)

Since the reflectivity function is, by definition, the derivative of the velocity function with respect to time, then:

$$r(t) = \frac{d(\log v(t))}{dt} = a \quad (5.17)$$

and we see that the reflectivity function is constant with respect to time. This means that within the medium where $v = v_0 + az$, we can approximate the reflectivity of the medium by a number of slices of constant time thickness, as shown in Figure 5.14, with the same reflection coefficient at the bottom of each time slice. As shown in Figure 5.14, this process of continuous reflection from the upper portion of the medium returns an integrated version of the incident source wavelet.

Another way of looking at this is to consider the sequence of constant reflection coefficients as equivalent to the *unit step function* $u(t)$ where

$$\begin{aligned} u(t) &= 1 & t \geq 0 \\ &= 0 & t < 0 \end{aligned}$$

The value of “1” is unimportant for the discussion; it could be, say, 0.05, more

like the value of a reflection coefficient. Letting $s(t)$ be a source wavelet then

$$\int_{-\infty}^{\infty} u(\tau)s(t - \tau)d\tau = \int_{-\infty}^{\infty} s(t)dt \quad (5.18)$$

so that convolution of $s(t)$ with a step function results in an integrated version of $s(t)$ with a corresponding change in shape of $s(t)$. In some examples below we'll let $s(t)$ be one cycle of a cosine wave to simulate a wavelet. For $s(t) = \cos(\omega t)$ Equation (5.18) becomes

$$\int_{-\infty}^{\infty} u(\tau) \cos[\omega(t - \tau)] d\tau = \sin(\omega t) \quad (5.19)$$

which is a convolution of a step function with the chosen source wavelet. In *Mathematica* language this convolution is written

```
ContinuousWavelet = Cos[omega tau];
IntegratedWavelet = Integrate[UnitStep[t - tau] *
Cos[omega*tau], {tau, -Pi, Pi}];
WindowedWavelet = UnitStep[tau*(Pi - tau)]*Wavelet;
WindowedIntegratedWavelet = UnitStep[t*(Pi - t)]*
IntegratedWavelet;
(* Transforms and plot statements are on CD-ROM *)
```

and the output of this notebook "Ramp Analytic Convolution.nb" is shown in Figure 5.13. As continuous sinusoids, the windowed sinusoids have the same basic frequency; that is, ω is the same for both the 1-cycle cosine "source wavelet" and the 1-cycle sine integrated version. Yet the integral of the windowed cosine wave has a spectrum that is shifted toward lower frequencies. This is caused by the abrupt truncation of the 1-cycle cosine wave at $t = 0$ and after a duration of one period. Higher frequencies are required to form the sharp onset at $t = 0$ and at its termination after one period. These high frequencies are not required for the integrated version, which is a sine wave with a less abrupt arrival time.

A different perspective of the same model is shown in Figure 5.15 where the reflection coefficients are represented as a series of spikes that approximate a rectangular pulse. Convolution of these with the source wavelet, one cycle of a cosine wave, results in the two reflections shown at the bottom. Each of these reflections is a cosine wave—one starting at the onset of the interval within which the velocity increases linearly with depth, and one starting at the termination of the linear interval and reversed in polarity. The ramp velocity model results in zero output from within the interval where the reflected wavelets can interact along their full length and sum to their mean value, which is zero. The polarity reversal takes place because as the incident wavelet emerges from the interval there will be more negative values than positive ones in the average that contributes to the integration. If the depth interval over which $v = v_0 + az$ and therefore over which $r(t) = \text{constant}$ is less than the duration of the source wavelet, then partial integration will take place returning a partially integrated

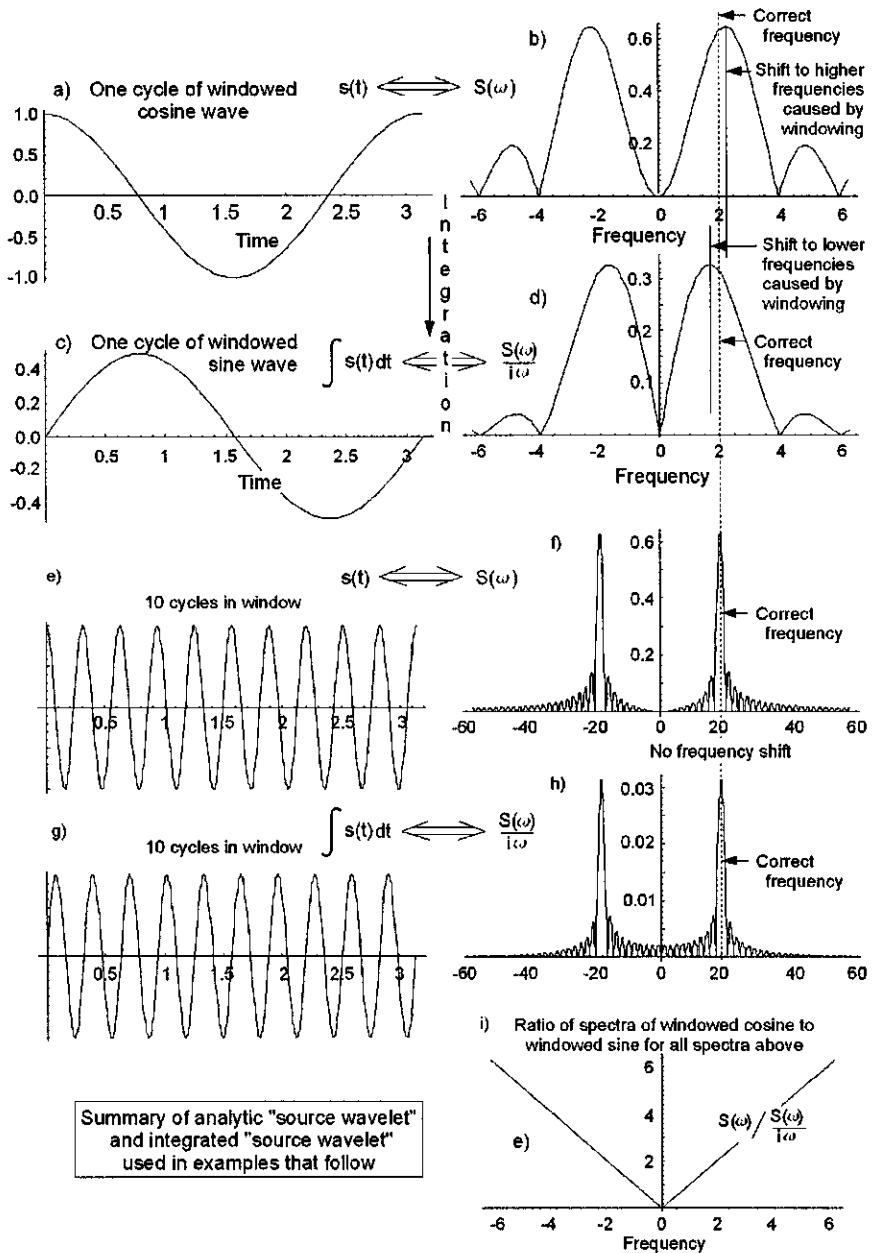


Figure 5.13: Convolution of wavelet with step function is equivalent to integration of wavelet. See (5.18). (a) One cycle of cosine wave taken as "source wavelet" to simplify concept of continuous reflection for later examples where $v = v_0 + a z$. (b) Fourier transform of windowed wavelet. Shift to higher frequencies due to windowing the sinusoid of infinite extent. (c) Analytic convolution of wavelet with step function. (d) Transform of windowed integrated wavelet. Shift to lower frequencies due to windowing. (e) 10 cycles of cosine wave. (f) Transform. Enough cycles of continuous waveform are in windowed sample to minimize shifts shown in (b) and (d). (g) Convolution of (e) with step function. (h) Transform. (i) Ratio of Fourier transform of any windowed source wavelet to windowed integrated source wavelets see (3.52) on page 98. (Ramp Analytic Convolution.nb,cdr → Ramp Analytic Convolution.wmf)

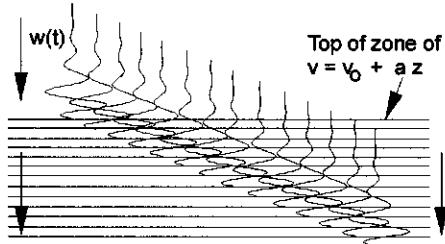
source wavelet from the top of the interval and a partially integrated one from the bottom. For the model wavelet shown in Figure 5.15, a cosine wave,

$$\int \cos(\omega t) dt = \sin(\omega t)$$

is fully integrated to a sine wave, as expected from the top zone of the reflecting interval.

a) Downward-travelling reflections.

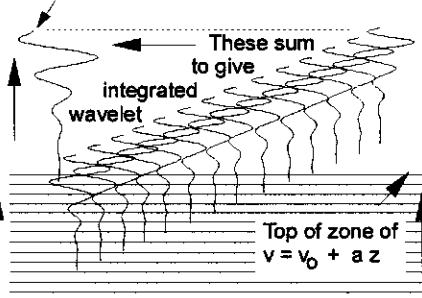
The downward travelling source wavelet $w(t)$ is first incident on the top of the continuously reflecting interval, then is incident on the top of the next thin layer, then is incident on the top of the third layer, etc. etc. The reflected result is shown to the right as the simple algebraic sum of the upward-travelling reflected wavelets. The resulting composite reflection is an integration as shown in the diagram to the right.



b) Upward-travelling reflections.

These reflected source wavelets add together on the way up to result in an integrated source wavelet.

Time of start of integrated wavelet corresponds to two-way traveltimes to top of continuous-reflecting zone



• Zone continues downward •

Figure 5.14: Ramp. Left: For a medium within which $v = v_0 + az$, the reflecting properties of the medium can be approximated by a number of thin layers of constant time thickness Δt equal to, say, one sample interval, which corresponds to the two-way time thickness of each thin layer. The top of each layer has the same reflection coefficient as a consequence of Equation (5.17). A source wavelet is reflected from the top of each of these layers. Each reflection is of the same polarity and amplitude because $r(t) = a = \text{constant}$. Right: The resulting up-going reflected source wavelets sum together to result in the superposition of all of the individual source wavelets reflected from the top of each thin layer; it is therefore an integration. Although vertical incidence is assumed, the downward-going and upward-going wavelets are separated horizontally for clarity. Conclusion: The single reflected event returned to the surface originates not just from the top of the medium, but continuously from the *upper portion* of the medium within which $v = v_0 + az$. Thus, a ramp velocity function reflects back a single composite wavelet that is the integral of the incident source wavelet. (timeslic2.cdr → timeslic2.wmf)

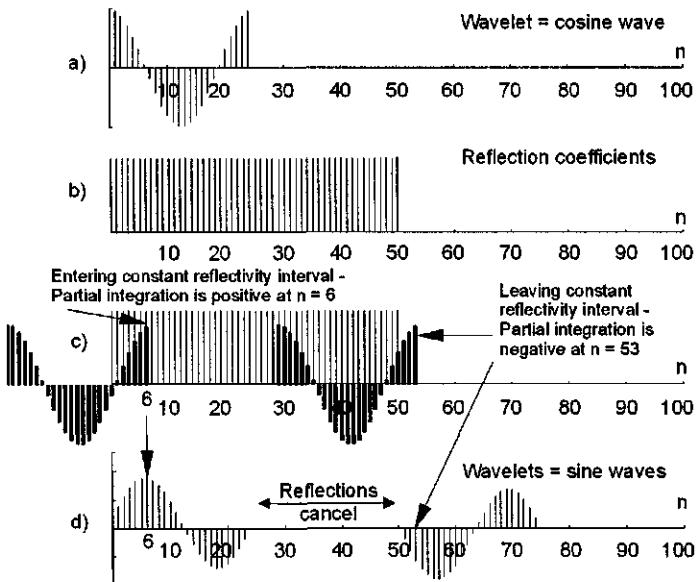


Figure 5.15: Integration by a ramp velocity function. Why the reflection from the bottom of the lithologic interval within which $v = v_0 + az$ is reversed in polarity. (a) Wavelet: One cycle of a cosine wave. (b) Reflection coefficients corresponding to a ramp “velocity function”. (c) The convolution. The reflection from the top of the interval within which $v = v_0 + az$ is a sine wave of the same polarity as the cosine wave because the partial integration shown for the arbitrary time index is becoming increasingly positive. The reflection developed at the end of the linear interval is the negative of the one from near the top because the sum for the partial integration at the index shown is starting out more negative than positive because the source wavelet is not entirely within the interval and there are more negative wavelet values than positive ones. These two reflections result from *continuous reflection* from within the linear interval near the top and bottom. (d) When the wavelet is entirely within the linear interval it is integrated out of existence! No reflections are returned from within the medium. Compare this with Figure 5.17. (ma, cd → ramp.wmf)

The source wavelet reflected from each of the three velocity functions shown in Figure 5.10 is a single event. Any information, therefore, that can be derived from the shape of the wavelet itself will be useful in an attempt to identify the geologic reason for the reflection. Inspection of the wavelets in Figure 5.10 shows that the “differentiated source wavelet” is enriched in higher frequencies with respect to the source wavelet, and the “integrated source wavelet” is enriched in lower frequencies. Although this might be difficult to recognize in the time

domain, it is easily confirmed in the frequency domain by examination of the ratios of the Fourier transforms of the wavelets, as shown in Figure 5.16. In addition, the use of the “instantaneous frequency” seismic attribute discussed in Section 5.8 clearly discriminates between these wavelet shapes.

The ratio of the Fourier transform of the derivative wavelet to the transform of the wavelet is $i\omega$. This should plot as a straight line, as shown. The ratio of the Fourier transform of the integrated wavelet to the transform of the wavelet itself is $1/(i\omega)$, and so the modulus of the ratio plots as shown in the right-hand part of the figure. Of course, the signal-to-noise ratio at various frequencies will govern how much of the ratio can be reliably interpreted. Reflection seismology today is involved in examining the shape of the seismic wavelet as well as its amplitude and arrival time. The information content contained in the shape of the wavelet is of considerable importance if the geologic model is to be understood. The three velocity functions in Figure 5.10 can be combined in any manner to

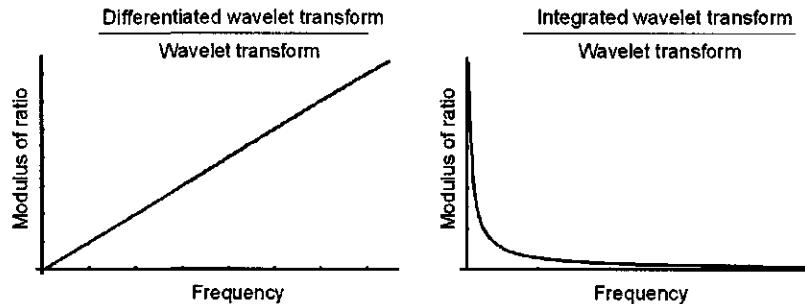


Figure 5.16: Left: The modulus of the ratio of the Fourier transform of the differentiated source wavelet shown in Figure 5.10 to the Fourier transform of the source wavelet itself. The ratio is theoretically equal to $i\omega$ and should plot as a straight line modulus $= |i\omega| = \omega$, which it does. Right: The modulus of the ratio of the Fourier transform of the integrated source wavelet shown in Figure 5.10 to the Fourier transform of the source wavelet itself. The modulus of the ratio should be inversely proportional ($|1/i\omega| = 1/\omega$) to the frequency, as shown. This non-linear frequency-domain process is equivalent to a partial superposition or summing in the time domain. (ma, cd → ratios.wmf)

form a more complicated velocity function as shown in Figure 5.28. For example, the sum of two ramp velocity functions is a triangular velocity function. Reflections from this more complicated arrangement are shown in Figure 5.17. For simplicity, one cycle of a cosine wave is used as the source wavelet. The reflections from the top and bottom are sine waves. The one from the middle is also a sine wave, but doubled in amplitude and reversed in polarity. The reader should “flip and slide” the reflection coefficients to get more insight into how these reflections originate.

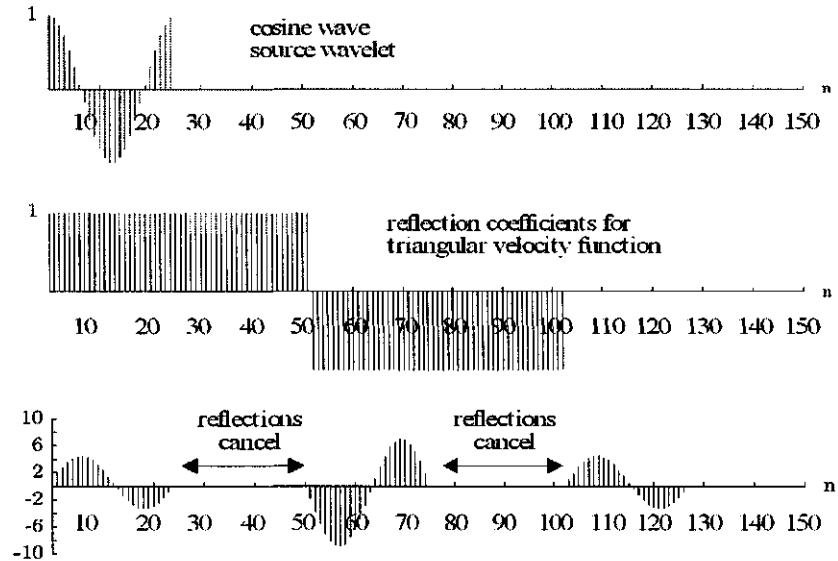


Figure 5.17: Integration by a triangular velocity function. Top: Wavelet, one cycle of a cosine wave. Middle: Reflection coefficients corresponding to a triangular “velocity function”. The first interval corresponds to $v = v_0 + az$, the second to $v = v_0 - az$. Bottom: The convolution. The reflection from the top of the interval within which $v = v_0 + az$ is a sine wave of the same polarity as the cosine wave. The reflection developed at the end of the linear interval within which $v = v_0 - az$ is the same as the one from the top. The reflection starting at the join of the two linear velocity functions is a sine wave with an amplitude twice that of the one at the top or the bottom, and with opposite polarity. Compare this with Figure 5.15. (triangvf.wmf)

5.7.4 Wavelet Tuning

The phenomenon of wavelet tuning is related to the interaction of the wavelets reflecting from the top and bottom of an interval. Tuning is controlled by the length of the wavelet with respect to the two-way time thickness of the interval. From this definition it is obvious that there will be no interaction when the thickness of the interval is greater than the length of the wavelet; however, when the thickness of the interval is smaller than the length of the wavelet, a partial interaction between the two wavelets will result in a composite waveform with a different shape as the response of the “thin bed.” An interval (bed) is defined to be “thin” if its thickness in two-way time is much less than the duration of the wavelet reflecting from its top and bottom. Furthermore, as discussed earlier, the thin interval acts as a differentiator when the top and bottom of the interval are represented by reflection coefficients of opposite polarity. Similarly, a thin

interval acts as an integrator when the polarity of the reflection coefficients is the same. There are excellent references in the literature on the thin-layer, wavelet tuning, and resolution [194, 92, 18]. Kallweit and Wood [92] reported on the limits of resolution of reflections from thin beds and Berkhouit [18, this Handbook Series, Volume 12, p. 48] on the resolution and detectability of thin beds. The zero-phase wavelet is required for resolution as well as detectability [18].

Let's examine a Ricker wavelet (zero-phase) defined by

$$\frac{1.0 - 2(\pi\nu_d t)^2}{e^{(\pi\nu_d t)^2}}$$

where ν_d is the dominant frequency.

A Ricker wavelet with a 50-Hertz dominant frequency and its amplitude spectrum are shown in Figure 5.18. The breadth of the Ricker wavelet in seconds is given by

$$\frac{\pi}{4\nu_d}$$

and is about 16 ms. It is obvious that the wavelets reflected from the top and bottom of an interval with opposite polarities will interact constructively when the two-way thickness of the interval is equal to the one half of the wavelet breadth. Similarly, the wavelets reflected from the top and bottom of an interval with the same polarity will interact destructively when the two-way thickness of the interval is equal to the one half of the wavelet breadth. Interaction of the wavelets with opposite polarity will result in differentiation where the effect of the differentiation is controlled by the ratio between the two-way layer thickness and one-half of the wavelet breadth.

Variation of the peak-to-trough amplitude of the composite wavelet formed by the interaction is shown in Figure 5.19. For a reflection coefficient of 0.25, the peak-trough amplitude of the composite wavelet quickly increases from 0 (no layer) to 0.44 while the ratio between the layer thickness and one-half breadth of the wavelet increases from 0 to 0.9. Then the peak-trough amplitude of the composite wavelet slowly decreases and asymptotically reaches the peak-trough amplitude (0.36) of the Ricker wavelet that is scaled by the reflection coefficient (0.25). The maximum tuned peak-trough amplitude of the composite wavelet corresponds to a 23% increase over the peak-trough amplitude of the interaction-free reflected wavelet. Obviously, this percentage gain is related to the type and shape of the wavelet. The increase in amplitude caused by thin-layer tuning will be greater with the relative increase in the side lobes of the wavelet.

Three snapshots from a movie (included on the CD-ROM that accompanies this volume as Wavelet Differentiation.movie) that simulates wavelet interaction as the layer thickness changes are shown in Figure 5.20. The curves, in order from low- to high-amplitude, correspond to ratios of 0.142, 1.0, and 1.57 between the two-way layer thickness and half-breadth of the wavelet. The change in the amplitude and shape of the composite wavelet demonstrate the importance of

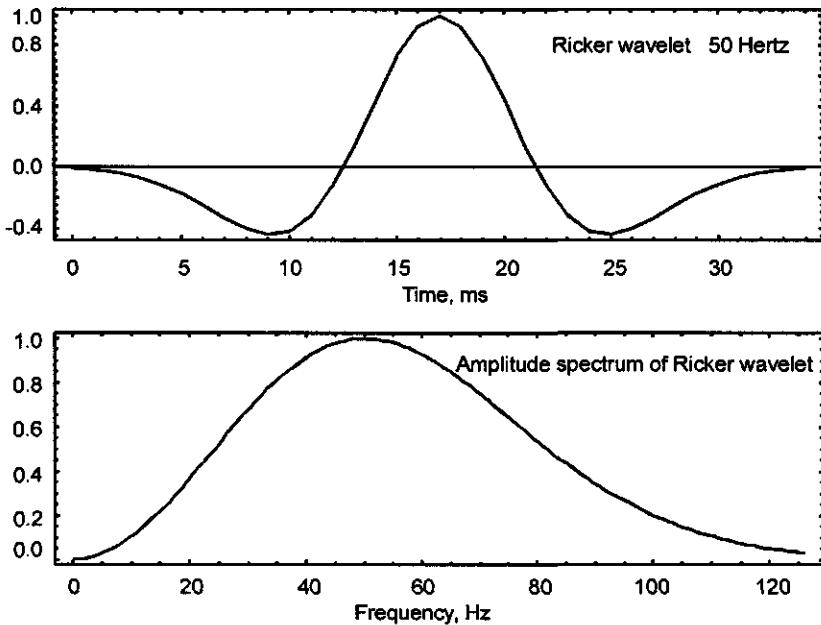


Figure 5.18: Ricker Wavelet. Top: Wavelet with a dominant frequency of 50 Hertz. Bottom: Its amplitude spectrum. Following figures related to tuning from *Mathematica* program *WaveletTuningFinal.nb* and *WaveletTuning.cdr*

the effect of the thin-layer and tuning. It can easily lead to misinterpretation such as false amplitude anomalies if the effect of the thin-layer and tuning is not recognized.

The amplitude spectra of the composite wavelets given in Figure 5.20 along with the spectrum of the original Ricker wavelet, which is represented by the curve that has its peak at 50 Hertz, are shown in Figure 5.21. The curve with a peak about 60 Hertz corresponds to the ratio of 0.142 between the two-way layer thickness and half-breadth of the wavelet while the curve with a peak around 40 Hertz represents the ratio of 1.57. The third curve with a peak at about 50 Hertz represents the ratio of 1.0. The general interpretation is that a thinning layer in the thinning direction will first shift the frequency band to lower frequencies and then shift it to higher frequencies. Therefore, recognition of the location of tuning via this change in the amplitude spectra might lead to the estimation of the layer thickness.

The analytically differentiated Ricker wavelet is given by

$$t \nu_d^2 e^{-(\pi \nu_d t)^2} (389.636364(\nu_d t)^2 - 59.2176264)$$

and is shown in Figure 5.22.

If the reflection coefficients from the top and bottom of a thin-layer have the same polarity, then the wavelet interaction acts as an integration. The

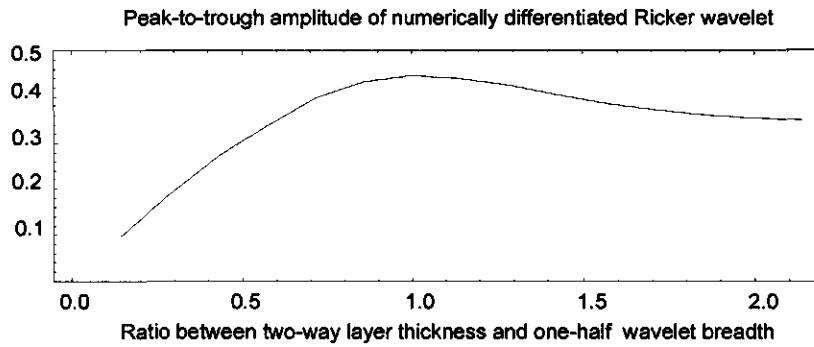


Figure 5.19: Wavelet differentiation. Peak-trough amplitude variation that results from differentiation of Ricker wavelets as a function of two-way layer thickness and one-half of the breadth of the wavelet.

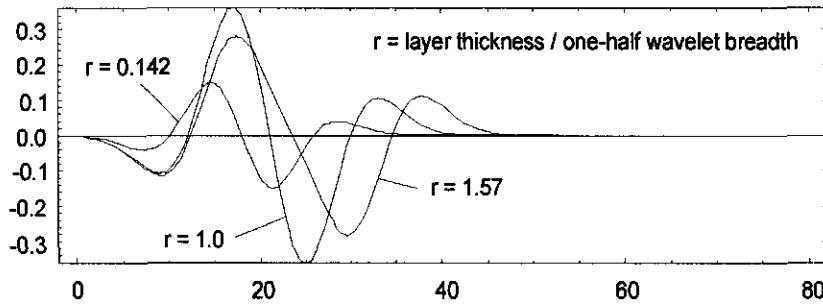


Figure 5.20: Wavelet interaction as a differentiation. (a) Peak-to-trough amplitude variation that results from differentiation of Ricker wavelets as a function of two-way layer thickness and one-half of the breadth of the wavelet.

variation in the peak-trough amplitude of the composite wavelet formed by such an interaction is shown in Figure 5.23. Again a reflection coefficient of 0.25 was considered at the top and bottom of the layer. The peak-trough amplitude of the composite wavelet reaches its maximum at the smallest thickness and decreases as the ratio between the two-way layer thickness and half-breadth of the wavelet increases. The peak-trough amplitude has its minimum near the ratio 1.0 and asymptotically reaches the peak-trough amplitude (0.36) of the Ricker wavelet that is scaled by the reflection coefficient (0.25). The change in the peak-trough amplitude between the minimum and maximum corresponds to a variation of about 158%. The change in the maximum peak-trough amplitude over the peak-trough amplitude of the Ricker wavelet scaled by the reflection coefficient (0.25) is about 75%. Obviously, these percentage gains are determined by the type and shape of the wavelet. The change in the peak-trough amplitudes will be greater with the relative increase in the amplitudes of the side lobes of the wavelet.

Three snapshots from a movie (included on the CD-ROM as "Wavelet Inte-

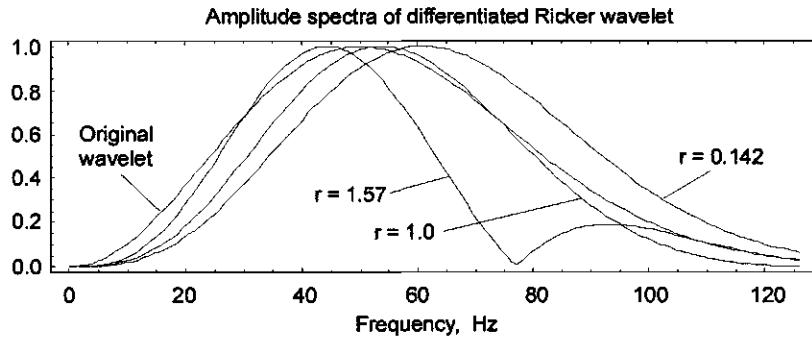


Figure 5.21: Effect of wavelet differentiation on amplitude spectrum. The curves identified with their peak from left to right represent the ratio of 1.57, 1.0, and 0.142 between the two-way layer thickness and half-breadth of the wavelet. The fourth curve with the peak at 50 Hertz represents the original Ricker wavelet.

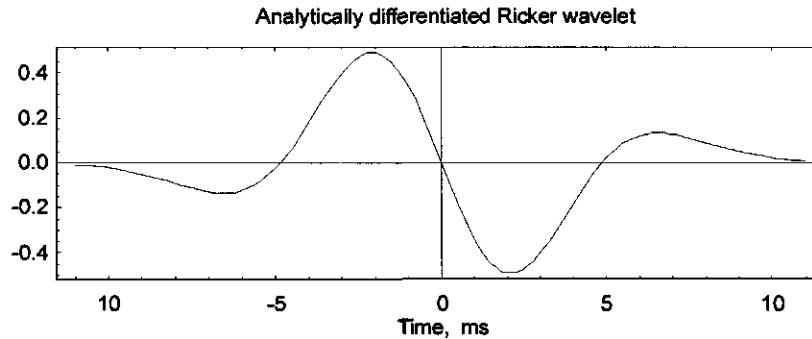


Figure 5.22: Analytically differentiated Ricker wavelet.

gration movie") that simulates wavelet interaction as the layer thickness changes are shown in Figure 5.24. The curves, in the order from shorter length to longer length of the composite wavelet, correspond to the ratio of 0.142, 1.0, and 1.57 between the two-way layer thickness and half-breadth of the wavelet. The change in the amplitude and shape of the composite wavelet exhibit the importance of the effect of the thin layer and tuning as an integration process. As in the case of wavelet differentiation, wavelet integration can easily lead to misinterpretation such as false amplitude anomalies if the effect of the thin layer and tuning is not recognized.

The amplitude spectra of the composite wavelets given in Figure 5.24 along with the spectrum of the original Ricker wavelet, which is represented by the curve that has its peak at 50 Hertz, are shown in Figure 5.25. The curve with a peak about 60 Hertz corresponds to the ratio of 0.142 between the two-way layer thickness and half-breadth of the wavelet, while the curve with a peak

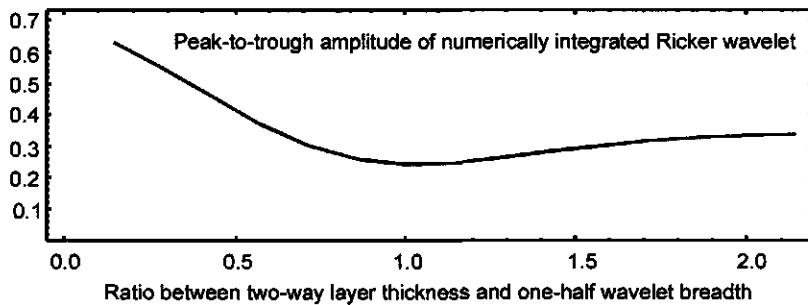


Figure 5.23: Peak-to-trough amplitudes in wavelet integration. Peak-to-trough amplitude variation that results from integration of Ricker wavelets as a function of two-way layer thickness and one-half of the breadth of the wavelet.

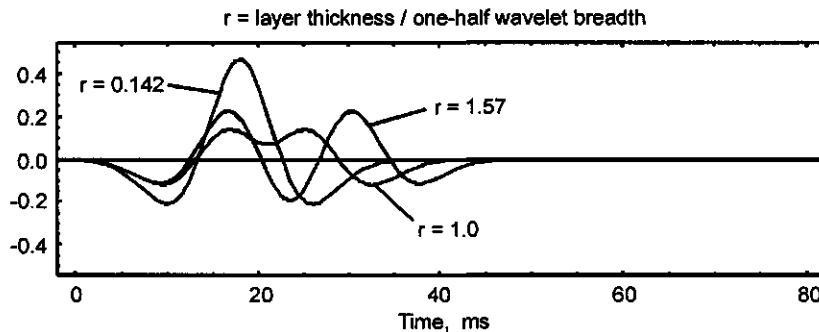


Figure 5.24: Wavelet interaction as an integration. The thinnest layer thickness causes the highest peak-trough amplitude with relatively shorter length of the composite wavelet. The smallest peak-trough amplitude of the composite wavelet occurs around the ratio of 1.0 between the two-way layer thickness and one-half of the breadth of the wavelet.

around 40 Hertz represents the ratio of 1.57. The third curve with a peak about 50 Hertz represents the ratio of 1.0. The general interpretation is that, as in the case of wavelet differentiation, a thinning layer in the thinning direction will first shift the frequency band to lower frequencies and then will shift it to higher frequencies. Wavelet integration results in a much more noticeable change in the amplitude spectra than does differentiation, as shown in Figure 5.25. Recognition of tuning by a change in amplitude spectra might lead to the estimation of layer thickness.

The analytically integrated Ricker wavelet is given by

$$t e^{-(\pi \nu_d t)^2}$$

and is shown in Figure 5.26.

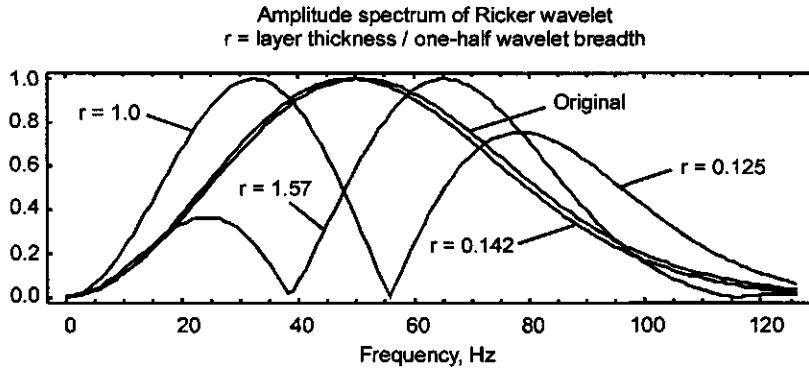


Figure 5.25: Effect of wavelet integration on amplitude Spectrum. The curves identified with their peak from left to right represent the ratio of 1.57, 1.0, and 0.142 between the two-way layer thickness and half-breadth of the wavelet. The fourth curve with the peak at 50 Hertz represents the original Ricker wavelet.

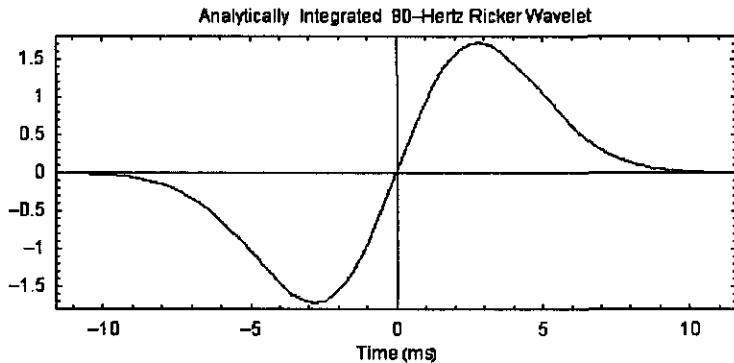


Figure 5.26: Analytically integrated Ricker wavelet.

The Ricker wavelet of 80 Hertz and its analytically differentiated and integrated versions are shown in Figure 5.27 for comparison. The differentiated wavelet has smaller amplitudes than the original wavelet while the integrated wavelet has larger amplitudes than the original wavelet. In terms of shape change, the differentiated wavelet has a phase shift of $-\pi/2$ while the integrated wavelet has $+\pi/2$ phase shift with respect to the original Ricker wavelet of zero phase shift.

Readers may run the movies provided on the CD-ROM to simulate wavelet differentiation and integration processes as the response of a thin layer.

The general conclusion from the above summary is that the shape and amplitude of the composite wavelet are determined by the layer thickness (two-way) and the breadth of the wavelet. Wavelet interaction caused by a thin layer can

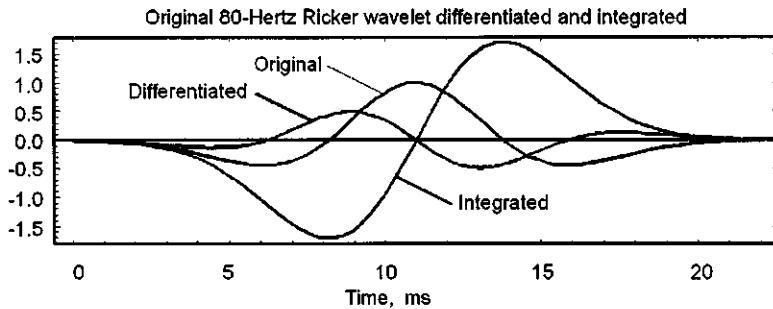


Figure 5.27: Ricker wavelet (80 Hz) and its analytically differentiated and integrated versions.

lead to misinterpretations; however, it can also be used as an interpretation tool.

5.7.5 Summary of velocity functions

Sengbush et al. [161] summarized the three basic “velocity functions” $w(t)$ required to describe any subsurface velocity distribution. They are

$$\text{Thin bed} \rightarrow w(t) = \delta(t) \quad (5.20)$$

$$\text{Single interface} \rightarrow w(t) = u(t) \quad (5.21)$$

$$v = v_0 + a z \rightarrow w(t) = \frac{A}{\Delta t} t u(t) = a t u(t) \quad (5.22)$$

These are illustrated in the top three figures in Figure 5.28, where $\Delta t = 500$ ms. Here Δt is not a sampling interval but rather the two-way traveltime thickness of a lithologic interval. These velocity functions were generated by the following *Mathematica* statements:

```
dt=0.5;A=1;
unitimpulse=
  SignalPlot[DiracDelta[t-0.5],{t,0,1}]
unitstep=Plot[UnitStep[t],{t,0,1}]
ramp=Plot[(A/dt)*t*UnitStep[t],{t,0,1}]
unitsquarepulse=
  Plot[UnitStep[t]-UnitStep[t-0.5],{t,0,1}]
ramptransition=
  Plot[A/dt*(t*UnitStep[t]-(t-dt)*UnitStep[t-dt]),{t,0,1}]
triangle=Plot[(A/dt)*(
  t*UnitStep[t]-2*(t-dt)*UnitStep[t-dt]+(t-2*dt)*
  UnitStep[t-2*dt]),{t,0,1}]
```

In order to convolve the reflection coefficients $r(t)$ with the source wavelet, the reader must recall that

$$r(t) = \frac{d [w(t)]}{dt} = \frac{d [\log v(t)]}{dt}$$

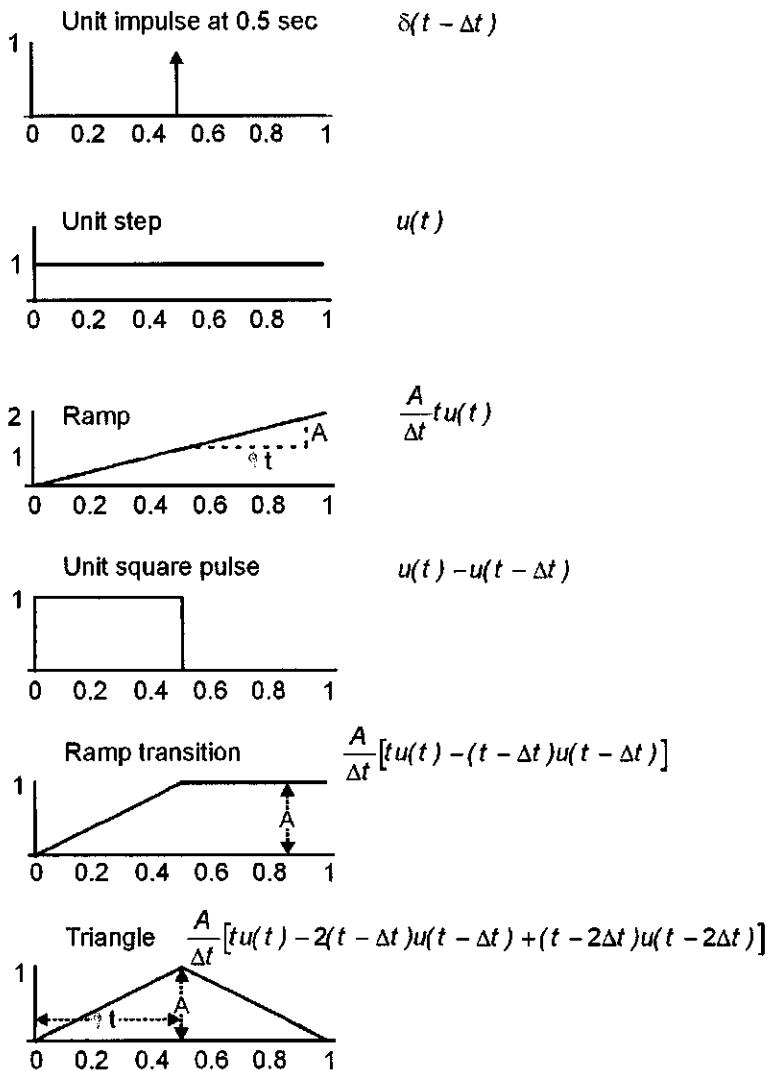


Figure 5.28: The top three functions are the basic velocity function building blocks. Simple combinations of these are shown below in the bottom three figures. After Sengbush, Lawrence and McDonal [161]. (SengbushVelocityFunctions)

Simple combinations of the above three velocity functions yield more general velocity functions as shown in Figure 5.28.

$$\text{Thick bed } \rightarrow w(t) = u(t) - u(t - \Delta t)$$

$$\begin{aligned} \text{Thick interval with } v = v_0 + az \rightarrow w(t) &= \frac{A}{\Delta t} \left[t u(t) - (t - \Delta t) u(t - \Delta t) \right] \\ v = v_0 + az \text{ above } v = v_0 - az \rightarrow w(t) &= \frac{A}{\Delta t} \left[t u(t) - 2(t - \Delta t) u(t - \Delta t) \right. \\ &\quad \left. + (t - 2\Delta t) u(t - 2\Delta t) \right] \end{aligned}$$

In general, the time thickness of a lithologic interval will be considerably less than the duration of a source wavelet. Nevertheless, any velocity distribution $v(z)$ can be approximated to any desired accuracy by using appropriate positive and negative values of a in Equation (5.22). Velocity functions, their Fourier transforms, $i\omega$ times the Fourier transform, and the reflection coefficient(s) associated with the velocity functions are summarized in Figure 5.29.

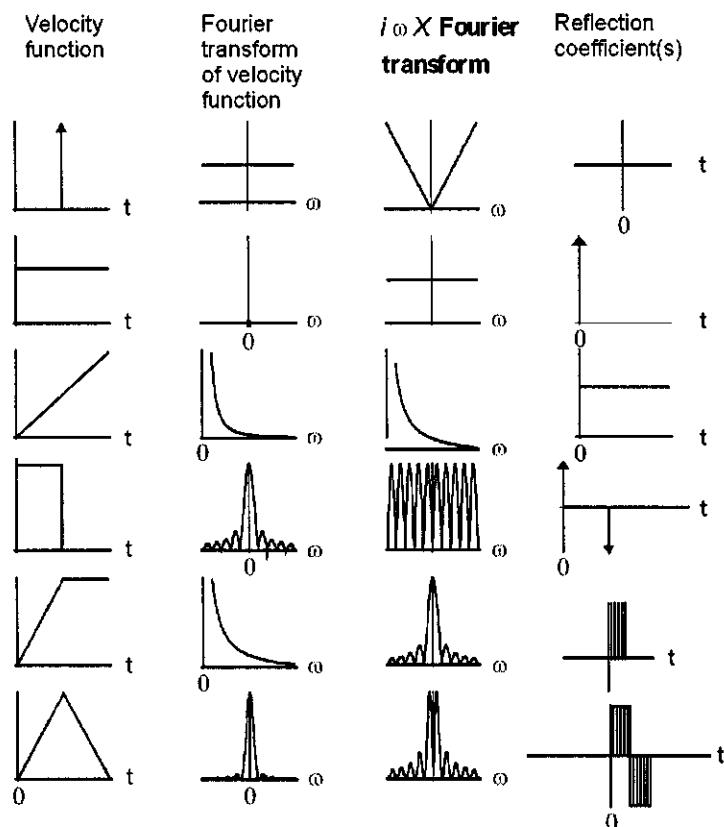


Figure 5.29: First column: Velocity functions of [161]. Second column: Fourier transform of the velocity function. Third column: $i\omega$ times the Fourier transform. Fourth column: The corresponding reflection coefficients. (SengbushVelocityPanel.cdr → SengbushVelocityPanel.wmf)

5.8 Seismic trace attributes

It is important to extract information from the seismic trace that might be directly related to variations in lithology. Parameters that provide information about reflectivity, phase, and frequency with respect to lithology are called “seismic trace attributes.” We saw in Chapter 5, page 229, that the seismic wavelet when reflected from a thin bed is differentiated, and when reflected from the top of a zone within which the velocity changes linearly with depth is integrated. This changes the shape of the reflected wavelet. Reflection from a thin bed enriches the reflected wavelet in high frequencies (Figure 5.10 and p. 229). Reflection from the top of a linear-with-depth velocity change enriches the reflected wavelet in low frequencies (p. 229). Clearly these are two different geologic models and it would be useful to be able to distinguish between them on the seismic trace. In order to do this we could transform the recorded trace into some new kind of trace from which we could obtain a direct measure of

$$\omega(t)$$

that is, a direct measure of the instantaneous frequency content of the trace continuously with respect to time. By definition $\theta(t) = \omega(t)t$. Therefore,

$$\omega(t) = \frac{\theta(t)}{t} \quad (5.23)$$

where $\omega(t)$ defines a time-average angular frequency from 0 to t . In order to extract higher resolution information directly related to lithology such as velocity, quality factor Q , etc., an “instantaneous” frequency, is defined as

$$\omega(t) = \frac{d\theta(t)}{dt}$$

We require some kind of a “complex” seismic trace, with a real part and an imaginary part, from which we can construct a phase angle spectrum $\theta(t)$. Differentiation of this with respect to time would then provide $\omega(t)$. Furthermore, from purely physical considerations, we would want such a trace to contain only positive frequencies.

Pursuing this goal, we observe that

$$\begin{aligned} e^{i\omega t} &= \cos(\omega t) + i \sin(\omega t) \\ &= \cos(\omega t) + i \mathcal{H}[\cos(\omega t)] \end{aligned} \quad (5.24)$$

where \mathcal{H} denotes Hilbert transformation as discussed in Section 3.5.11. Equation (5.24) is an example of generating a complex time-domain function ($= e^{i\omega t}$) from a real one, $\cos(\omega t)$, by taking the Hilbert transform $= \mathcal{H}[\cos(\omega t)]$ of the real one. That is, given any real (not necessarily causal) function $f(t)$ then

$$g(t) = f(t) + i \mathcal{H}[f(t)]$$

where $g(t)$ is complex. $g(t)$ is called the “analytic signal.” We call it the “complex seismic trace.” $\mathcal{H}[f(t)]$ is referred to as the “quadrature function.” It is the imaginary part of $g(t)$. As shown in Section 3.5.11, this is a general result for determining a complex time-domain function $g(t)$.

The conventional seismic trace $f(t)$ can thus be represented as the real part of a complex seismic trace [108, 24]. As shown in the figures that follow, this complex representation will allow us to see subtle changes in frequency, for example, that are present on a conventional time-domain trace but become more easily noticed by the interpreter when displayed as “instantaneous” frequency, especially if this “attribute” of the complex trace is displayed in color. The Hilbert transform discussed in Section 3.5.11 on page 129 is used to construct the “complex seismic trace”, also called the “analytic signal”.

Three “seismic attributes” can be defined using the complex seismic trace. They are the amplitude envelope $A(t)$, the instantaneous phase $\theta(t)$, and the instantaneous frequency $\nu(t)$ [24, 178]. Taking the Hilbert transform of a function of time is equivalent to taking the Fourier transform of a Fourier transform, so after Hilbert transformation you are returned to the same domain, in this case time. Recall the discussion on page 138 for further elaboration.

As discussed above and in Section 3.5.11, the imaginary part $\mathcal{H}[f(t)]$ of the complex seismic trace $g(t)$ can be computed by taking the Hilbert transform of the seismic trace $f(t)$. Following the discussion in Section 3.5.11 on the mathematical meaning of the Hilbert transform the following procedure can be used to compute the imaginary part of the trace:

1. Compute the Fourier transform $F(\omega)$ of the recorded seismic trace $f(t)$. Note the negative sign (by our convention) in the exponential.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (5.25)$$

2. Advance the phase $\theta(\omega)$ of $F(\omega)$ for negative frequencies by an amount $+\pi/2$ and by $-\pi/2$ for the positive frequencies. That is,

$$\theta'(\omega) = \begin{cases} \theta(\omega) - \pi/2 & \text{if } \omega > 0 \\ \theta(\omega) + \pi/2 & \text{if } \omega < 0 \end{cases} \quad (5.26)$$

so that $F_{mod}(\omega)$ with phase spectrum $\theta_{mod}(\omega)$ is the phase-modified version of $F(\omega)$.

3. Now obtain the imaginary part $\mathcal{H}[f(t)]$ of the complex seismic trace in the time-domain by an inverse Fourier transformation

$$\mathcal{H}[f(t)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_{mod}(\omega) e^{+i\omega t} d\omega \quad (5.27)$$

The complex or analytic seismic trace $g(t)$ in the time-domain can now be obtained by combining the real part $f(t)$ and the imaginary part $\mathcal{H}[f(t)]$ of the seismic trace. Thus

$$g(t) = f(t) + i\mathcal{H}[f(t)] \quad (5.28)$$

Using the complex trace $g(t)$ the envelope amplitude $A(t)$ is defined as the modulus, or absolute value, of the complex trace

$$\begin{aligned} A(t) &= |g(t)| \\ &= \sqrt{f(t)^2 + \mathcal{H}[f(t)]^2} \end{aligned} \quad (5.29)$$

The envelope function $A(t)$ is also called the “reflection strength” and is defined by the length of the phasor measured perpendicular to the time axis from the axis to the complex trace. This is the phase-independent amplitude that envelops the complex seismic trace. There is only one envelope, as defined by Equation (5.29).

The instantaneous phase $\theta(t)$ as a function of time is obtained from the real $f(t)$ and imaginary $\mathcal{H}[f(t)]$ parts of the complex trace $g(t)$. The instantaneous phase $\theta(t)$ in the time-domain is by definition

$$\theta(t) = \tan^{-1} \left[\frac{\mathcal{H}[f(t)]}{f(t)} \right] \quad (5.30)$$

Instantaneous phase $\theta(t)$ is independent of trace amplitude and therefore a display of instantaneous phase emphasizes reflection continuity better than a conventional amplitude display.

Instantaneous angular frequency $\omega(t)$ is defined as the rate of change of the instantaneous phase $\theta(t)$ with time. That is,

$$\omega(t) = \frac{d\theta(t)}{dt} \approx \frac{\Delta\theta(t)}{\Delta t} \quad (5.31)$$

$$= \frac{d}{dt} \left\{ \tan^{-1} \left[\frac{\mathcal{H}[f(t)]}{f(t)} \right] \right\} \quad (5.32)$$

where Δt is the sampling interval. Defining the instantaneous frequency in Hertz $\nu(t) = \omega(t)/2\pi$, we write

$$\begin{aligned} \nu(t) &= \frac{1}{2\pi} \frac{d\theta(t)}{dt} \approx \frac{1}{2\pi} \frac{\Delta\theta(t)}{\Delta t} \\ &= \frac{1}{2\pi} \frac{d}{dt} \left\{ \tan^{-1} \left[\frac{\mathcal{H}[f(t)]}{f(t)} \right] \right\} \end{aligned}$$

The phase $\theta(t)$ will need to be unwrapped before (5.31) can be used to compute the instantaneous frequency.

Unwrapping is not necessary if Equation (5.33) below is used. Using *Mathematica* to differentiate (5.32) and changing the sign of $\mathcal{H}[f(t)]$ because of the adopted sign convention (Section 3.3.1) yields

$$D[\arctan\{f(t), -\mathcal{H}[f(t)]\}, t]$$

from which

$$\omega(t) = \frac{d\theta(t)}{dt} = \frac{\mathcal{H}[f(t)] f'(t) - f(t) \mathcal{H}'[f(t)]}{f(t)^2 + \mathcal{H}[f(t)]^2} \quad (5.33)$$

where $f'(t)$ and $\mathcal{H}'[f(t)]$ are derivatives with respect to time.

We also suggest use of the average angular frequency $\omega_{ave}(t)$ as defined by Equation (5.23). Defining

$$\begin{aligned}\theta(t) &= \omega_{ave}(t)t \\ \omega_{ave}(t) &= \frac{\theta(t)}{t} \\ \nu_{ave}(t) &= \frac{\theta(t)}{2\pi t}\end{aligned}$$

In discrete form

$$\nu_{ave}[n] \approx \frac{\theta[n]}{2\pi n\Delta t} \quad (5.34)$$

A plot of a single trace of $\theta[n]$ versus $\nu[n]$ is shown in Figure 5.30. A record section of such a plot would show relative changes of frequency $\nu_{ave}[n]$ versus two-way traveltime $n\Delta t$.

Calculation of the instantaneous frequency requires the calculation of a difference $\Delta\theta$ in phase angles. In the presence of noise, however, it is preferable to determine $\Delta\theta$ as follows [197, 102, 115, 20, 21]. Given the time series

$$\begin{aligned}h[n] &= h_e[n] + i h_o[n] \\ h[n+1] &= |h[n+1]| e^{i\theta_{n+1}} \\ h[n] &= |h[n]| e^{i\theta_n} \\ h[n+1]h[n] &= |h[n+1]| |h[n]| e^{i(\theta_{n+1}+\theta_n)} \\ h[n+1]h^*[n] &= |h[n+1]| |h[n]| e^{i(\theta_{n+1}-\theta_n)}\end{aligned} \quad (5.35)$$

where $h^*[n]$ denotes the complex conjugate of $h[n]$. The phase difference $\theta_{n+1} - \theta_n$ (see Equation (2.6)) in Equation (5.35) is extracted as the argument of $h[n+1]h^*[n]$ using *Mathematica*'s `Arg[]` procedure applied to the left-hand side of Equation (5.35). This option using (5.35) to compute the phase difference is also included in *Mathematica* notebook `HilbertTransformOfTrace.nb` on the CD-ROM that accompanies this volume. A general discussion of phase unwrapping is given by Ghiglia and Pritt [77].

Instantaneous frequency $\nu(t)$ or $\omega(t)$ is not an abstract mathematical construct but a real measure of the local (in time) frequency content of the trace, unlike the amplitude spectrum. As discussed in Section 5.7.1, certain velocity distributions in the earth can perform the equivalent of the mathematical process of differentiation or integration of a source wavelet. This can result in the local enrichment of higher or lower frequencies, respectively. Attenuation by fluids can decrease frequency content by absorption (see Chapter 8.5.2). In addition, one expects the average value of $\nu(t)$ to be consistent with the information contained in $A(\nu)$. That is, the average numerical value of $\nu(t)$ for the trace must be prominent in the more conventional amplitude spectrum of the trace. We revisit this below with examples from real data.

Equations (5.29) and (5.30), and (5.31) or (5.32), provide the seismic attributes (envelope amplitude, instantaneous phase, and instantaneous frequency) that are widely used to identify and/or recognize subsurface structural and lithologic features such as faults, stratigraphic terminations (pinchouts), porosity and fracture changes, and other lateral changes and discontinuities. A smoothing filter can be applied to a seismic section plot of instantaneous phase in order to make regional trends more apparent. Additional insight into seismic attributes can be found in Taner and Sheriff [178] and Taner et al. [179].

A *Mathematica* function to compute the quadrature seismic trace is included in programs on the CD-ROM that accompanies this volume. The subroutine is:

```
hilbert[RealTrace_]:=({lh=Length[RealTrace];
  If[OddQ[Length[h]],lh2=Ceiling[lh/2],lh2=lh/2];
  fd=Fourier[RealTrace];
  Do[fd[[i]]=fd[[i]]*E^({+I*Pi/2},{i,lh2+1,lh});
  Do[fd[[i]]=fd[[i]]*E^({-I*Pi/2},{i,1,lh2});
  QuadratureTrace=Re[Chop[InverseFourier[fd]]]];
```

The phase defined by Equation (5.30) can be unwrapped using the following *Mathematica* function:

```
--Function--
unrapphase[data:_VectorQ,tol_:Pi,inc_:2 Pi]:=data+
inc*FoldList[Plus,0,Sign[Chop[Apply[
Subtract,Partition[data,2,1],{1}],tol]]];
--End of function--
data =the phase;
data1=unrapphase[data];
```

A seismic trace and its corresponding quadrature trace computed by the Hilbert transform are shown in Figure 5.31. The real and quadrature parts of a seismic trace are used to construct the complex seismic trace in the time-domain. The complex trace can be visualized as the trace that is drawn by the tip of a rotating phasor defined by $h(t) = f(t) + i\mathcal{H}[f(t)]$ perpendicular to the time axis and moving along the time axis. The seismic trace attributes, instantaneous phase $\theta(t)$ and instantaneous frequency $\nu(t)$, obtained from the complex seismic trace are shown in Figure 5.32. Several very slight irregularities in the instantaneous phase $\theta(t)$ are labeled on Figure 5.32 and correlated with the real seismic trace (Figure 5.32a) and the instantaneous frequency $\nu(t)$ (Figure 5.32d). The slight irregularities in the real trace and the instantaneous phase are prominent after the process of differentiation (Equation 5.31) used to obtain the instantaneous frequency $\nu(t)$. As in any process of differentiation, this emphasizes the higher frequencies and therefore amplifies any slight changes (including noise) that are present in the trace. This was also discussed on page 97 and shown in Figure 3.28. A similar result for a quite different application was shown in Figure 3.31 where the geothermal gradient is simply the derivative of the log of temperature versus depth and is highly sensitive to variations in sand/clay content. Seismic attributes are useful to emphasize subtle changes in the shape of composite wavelets that are not apparent on the seismic trace

itself. For example, an increase in instantaneous frequency is often associated with sedimentary pinchouts and thinning beds. The preferred method to view the seismic attributes is in color displays; i.e., higher frequencies in red, lower in blue. Most commercially available software packages for processing seismic data provide this valuable option.

In Figure 5.32d note that the average value of the instantaneous frequency $\nu(t)$ agrees approximately with the Fourier amplitude spectrum (Figure 5.32f) and with a statistical result from binning trace samples of instantaneous frequency (Figure 5.32g). The Fourier frequency increment and the width of the frequency bin are each $d\nu = 1$ Hertz. The average value of $\bar{\nu}(t)$ for this particular example is

$$\bar{\nu}[n] = \frac{1}{n} \sum_n \nu[n] \approx 29 \text{ Hertz} \quad (5.36)$$

The *Mathematica* program used to bin (collect) the trace samples of instantaneous frequency is

```
f1=0;f2=100;BarMidpoints=Table[f+1,{f,f1,f2-dfreq,dfreq}];  
bc=BinCounts[IFreq,{f1,f2,dfreq}];  
chart=Transpose[{bc,BarMidpoints}]//N;  
BarChart[chart,PlotRange->All]
```

In Figure 5.32d note how the instantaneous frequency attribute accentuates the high-frequency event even though this event is almost unrecognizable on the seismic trace itself.

The seismic trace envelope attribute $A(t)$ is shown in Figure 5.33. The reflection strength envelope shows more generally where maxima and minima are located on a seismic record section without direct concern for the detailed shape of the wavelet itself. Some displays reflect the envelope to envelop the negative troughs as well as the positive peaks, and this kind of display can be visualized for either Figure 5.33b or 5.33c. We show the envelope of $|h(t)|$ in Figure 5.33d. Taner and others [178, 179] discussed geologic situations for which this seismic attribute can be used to advantage. A useful example to illustrate the seismic attribute instantaneous frequency is provided by the conventional vibroseis sweep, a portion of which is shown in Figure 5.34a along with the Hilbert transform of the trace. In this example, $\nu(t)$ corresponds to the linearly varying sweep frequencies. The phase is shown in (b) and the instantaneous frequency in (c). This figure can be reproduced using the *Mathematica* notebook *HilbertTransformOfVibratorSweep* on the CD-ROM. The vibrator sweep is necessarily truncated; that is, it has a beginning and an end, which is equivalent to multiplication of the sweep by a rectangular window. Therefore, each frequency component will be convolved with the Fourier transform of the window resulting in ripples at each end of the curve of instantaneous frequency versus time. The sweep duration would have to be made very long (requiring long execution times), or tapered, to minimize the ripples. See the CD-ROM for the *Mathematica* notebook *HilbertTransformOfVibratorSweep.nb* used to generate Figure 5.34.

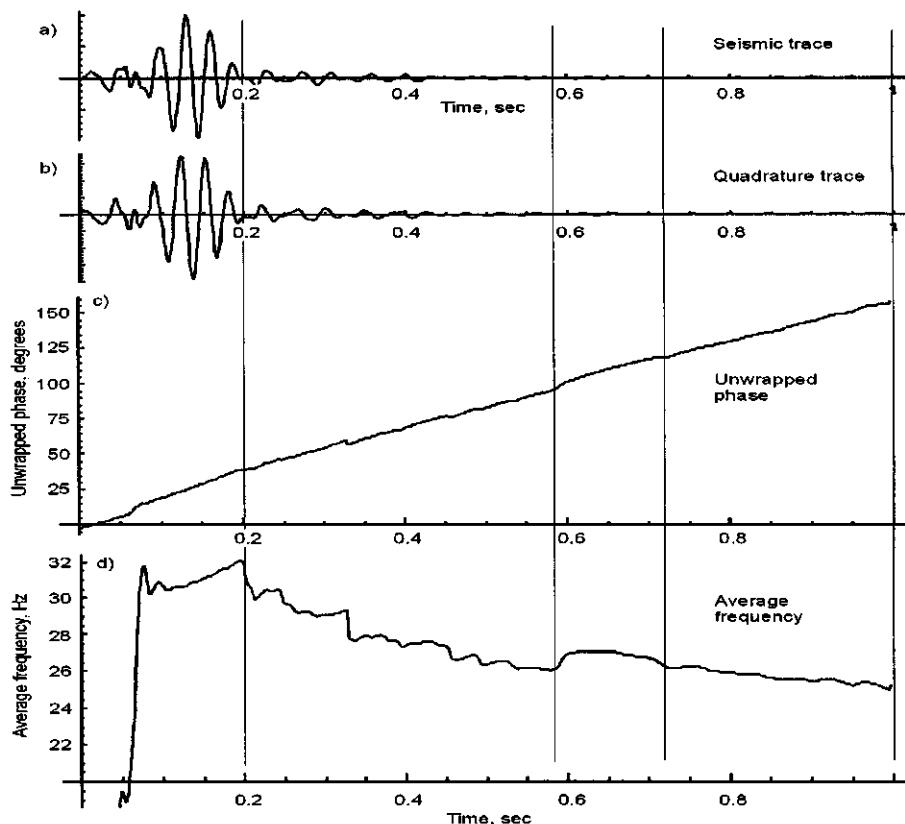


Figure 5.30: (a) A real seismic trace $f(t)$, (b) the quadrature trace $\mathcal{H}[f(t)]$ computed by taking the Hilbert transform of $f(t)$, (c) Unwrapped phase, and (d) Average frequency as defined by Equation (5.23). Dashed lines correlate features observed on seismic attributes but not obvious on the seismic trace. SeismicAttributesUnwrapped.nb \Rightarrow SeismicAttributes.cdr \Rightarrow NewDef.wmf

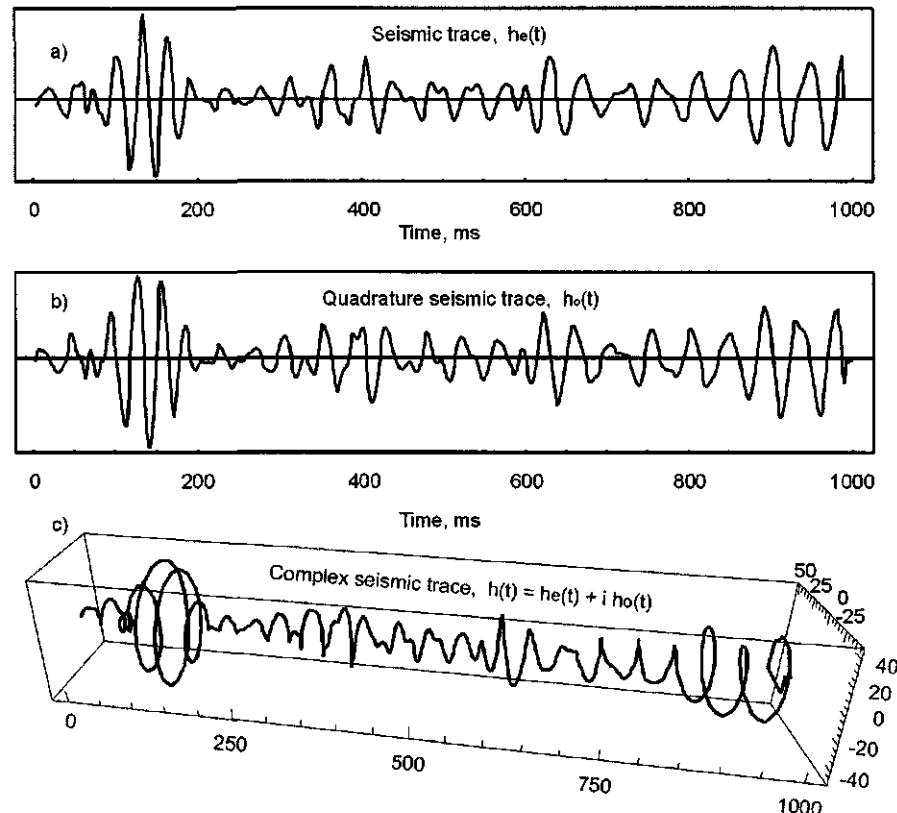


Figure 5.31: (a) A real seismic trace $f(t)$, and (b) the quadrature trace $\mathcal{H}[f(t)]$ computed by taking the Hilbert transform of $f(t)$. The tip of the phasor $A(t)$ defined by $A(t) = f(t) + i\mathcal{H}[f(t)]$ traces out the complex trace shown in (c). attr1.cdr \Rightarrow attr1.wmf

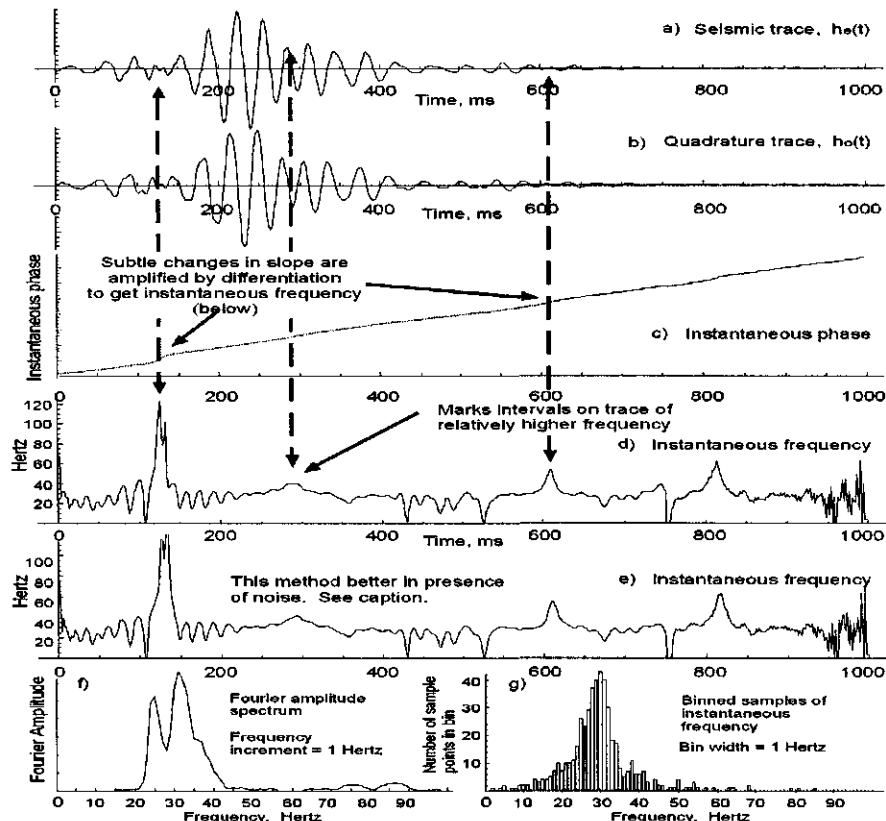


Figure 5.32: (a) The seismic trace. (b) Hilbert transform of trace. (c) Instantaneous phase. (d) Instantaneous frequency $\nu(t)$ from Equation (5.33). (e) Instantaneous frequency from Equation (5.35). (f) Fourier amplitude spectrum of trace. (g) Binning of trace sample points of $\nu(t)$. `HilbertTransformOfTrace.nb` \Rightarrow `SeismicAttributes.cdr` \Rightarrow `attr2.wmf`

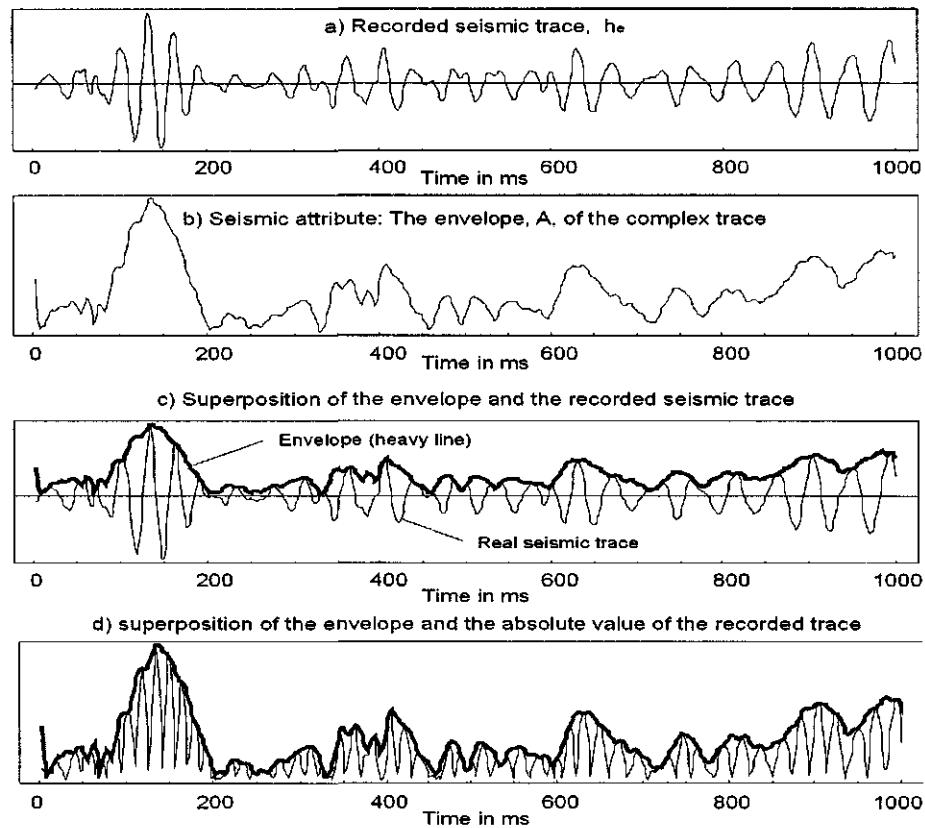


Figure 5.33: Seismic trace attributes computed using the real and quadrature parts of the seismic trace shown in Figure 5.31. (a) The seismic trace $f(t)$. (b) Envelope $A(t)$ of the real seismic trace $f(t)$. (c) Superposition of the envelope $A(t)$ and the real seismic trace $f(t)$. (d) Superposition of the envelope $A(t)$ and the absolute value $|f(t)|$ of the real seismic trace. SeismicAttributesUnwrapped.nb \Rightarrow SeismicAttributes.cdr \Rightarrow attr3.wmf

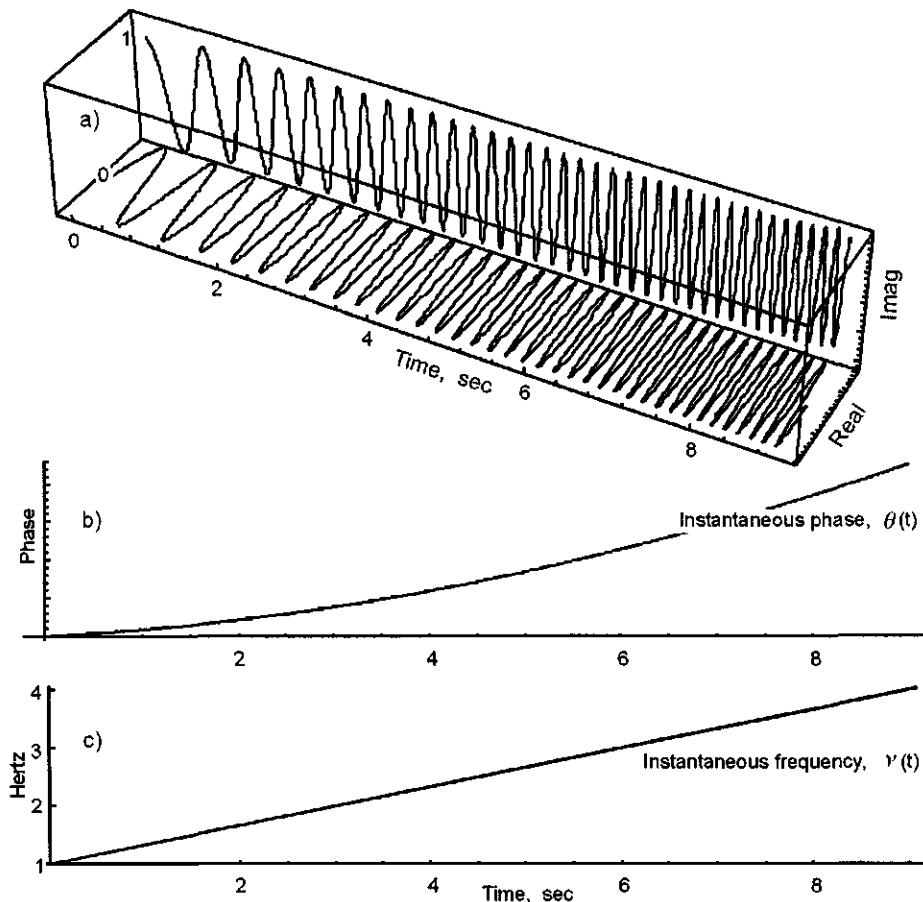


Figure 5.34: (a) Top: portion of vibrator sweep increasing linearly in frequency from 1 Hz to 4 Hz. Horizontal panel is real part $f(t)$ of complex sweep $h(t)$. Vertical panel is quadrature sweep $\mathcal{H}[f(t)]$ computed by taking Hilbert transform of $f(t)$. Sweep duration shown is 9 sec; however, sweep must be made long to minimize ripples at each end of instantaneous frequency curve. See text. (b) Unwrapped phase $\theta(t)$. (c) Instantaneous frequency $\nu(t)$ in Hertz recovered by using Hilbert transformation. `HilbertTransformOfVibratorSweep.nb` \Rightarrow `SeismicAttributes.cdr` \Rightarrow `instantvib.wmf`

Chapter 6

Traveltime curves and velocity

The immediate objective of exploration seismology is to determine the distribution of rock velocity versus depth in the earth. This is done by measuring the time it takes a disturbance to travel from some kind of an energy source to many *receivers*, called *geophones* on land and *hydrophones* offshore. Under the assumption that different rock types have different velocities, then if the distribution of changes in the subsurface variations in the speed of wave propagation in rocks can be determined, then the structural geometry of these rocks is also determined, thereby imaging folds, faults, depths to velocity discontinuities, to hydrocarbon accumulations, to the water table, to the base of the crust, etc., etc.

All efforts at velocity determination measure two quantities accurately: the time of travel of a seismic pulse from the source of a disturbance to receivers, and the distance on the surface of the earth between the source and the receivers. In one form or another, the data are displayed as seismic traveltimes versus distance from the energy source. For reflection data, the final displays are called *record sections*. For refraction data, the final displays are called *traveltime curves*, although some attempts have been made to present refraction data in the form of record sections (Çoruh et al. [46]) as discussed in Section 6.4 on page 325.

Although we will determine accurate velocity values versus depth, it is important to note that our data have generally been obtained on the surface of the ground, which is horizontal. The velocities that we measure, therefore, are *horizontal velocities* [86]. On the other hand, the drilling targets of interest at depth are in a vertical direction, and vertical velocities are required for accurate depth computations. Velocity anisotropy is common in rocks (Vossler [185]); for an ideal layered medium the horizontal velocity is always greater than the vertical. Therefore depth determinations made on the basis of surface measurements will yield velocities that are too high and depths that are shallower than expected when eventually confirmed by drilling.

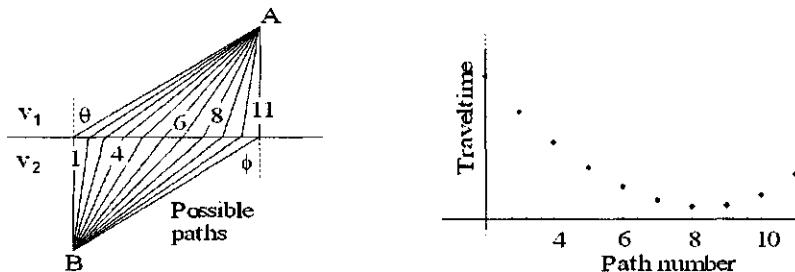


Figure 6.1: The physical significance of Snell's law is that it describes a least-time path and represents Fermat's principle. In order to go from point "A" to point "B" in the least possible time, the geometry of the raypath must obey Snell's law. Only one of the traveltimes shown (path 8) is a least-time path. (snell1a)

Before reviewing the different kinds of traveltimes we examine Snell's law, the fundamental law that governs the geometry of the least-time path between any source and any receiver. The curvature of the Earth and the implications for earthquake seismology are ignored; we assume a flat, horizontal surface on which the sources and receivers are placed.

6.1 Snell's law

The physical significance of Snell's law can be understood with reference to a simple application shown in Figure 6.1—to determine the length of time it would take a seismic wavefront to go from point A in medium 1 (upper medium) to point B in medium 2 (lower medium).

Points A and B are fixed with respect to the interface separating medium 1 of velocity v_1 from medium 2 of velocity v_2 . Without knowing which of the raypaths shown in Figure 6.1 is correct (i.e., which value of θ is correct) we measure with a ruler the amount of time spent along each segment, then plot the total time spent along any one path against some other parameter, the angle θ say, as shown in the figure and obtain the graph shown in Figure 6.1.

At the minimum value of traveltimes we find *experimentally* that

$$\frac{\sin \theta}{v_1} = \frac{\sin \phi}{v_2}$$

where θ and ϕ are the angles between the normal to the interface and the ray paths, and v_1 and v_2 are the velocities in medium 1 and medium 2, respectively. This is Snell's law, the general expression of which is

$$\frac{\sin \theta}{v} = \text{constant}$$

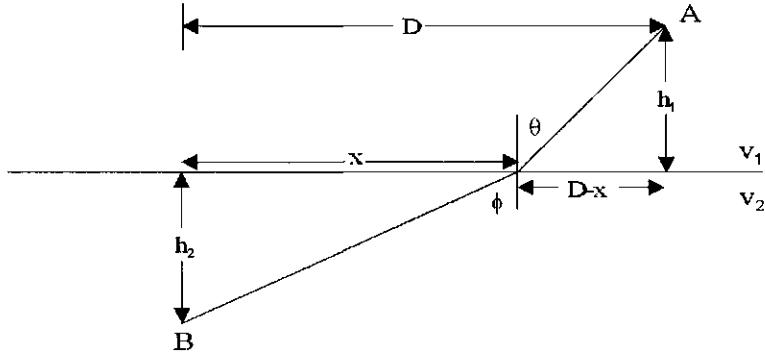


Figure 6.2: Figure for derivation of Snell's law. The distances h_1 , h_2 , and D are constant. The distance X is allowed to vary. The angle θ for which the traveltime is a minimum from A to B is associated with the path that obeys Snell's law. (snell.cdr → snell2)

The numerical value of the constant can be found at any point along the ray path by determining the value of the ratio $(\sin \theta)/v$ at the point on the raypath.

Snell's law is therefore a consequence of imposing a *least time* condition on the travel path; i.e., an object (or a wave front) traveling from A to B in the least amount of time *must* obey Snell's law. The principle can obviously be applied to more crooked travel paths composed of many segments corresponding to many velocity changes.

Snell's law is easily derived by showing that only *one* of the paths from A to B shown in Figure 6.1 is a minimum time path. The distances h_1 and h_2 are constant, but X is allowed to vary, thus defining travel paths of different total lengths. The total travelttime T along any one of these paths is

$$T = \frac{\sqrt{(D-X)^2 + h_1^2}}{v_1} + \frac{\sqrt{X^2 + h_2^2}}{v_2}$$

For T to be a minimum

$$\frac{dT}{dX} = \frac{-(D-X)}{v_1 \sqrt{(D-X)^2 + h_1^2}} + \frac{X}{v_2 \sqrt{X^2 + h_2^2}} = 0$$

or

$$\frac{(D-X)}{v_1 \sqrt{(D-X)^2 + h_1^2}} = \frac{X}{v_2 \sqrt{X^2 + h_2^2}}$$

from which inspection of Figure 6.2 gives

$$\frac{\sin \theta}{v_1} = \frac{\sin \phi}{v_2} \quad (6.1)$$

as before. Thus for T to be a minimum,

$$\frac{\sin \theta}{v} = \text{constant}$$

at all points along the ray path.

As in any problem dealing with minimization, it is often not necessary to examine the second derivative in order to determine whether the first derivative represents a maximum or a minimum. If any other travel path can be found that clearly takes *more* time to traverse than the Snell's law path defined by

$$\frac{\sin \theta}{v_1} = \frac{\sin \theta}{v_2}$$

then Snell's law must be associated with a minimum time path. In Figure 6.1, if "path 8" is the Snell's law path, then other paths can be drawn that would obviously take longer. A *Mathematica* program to solve for the quantity X in Figure 6.2 is given below. The velocities used are characteristic of low-velocity sediments of the Atlantic Coastal Plain overlying higher-velocity rocks of the crystalline basement. The velocities and thicknesses are in km/sec and km, respectively.

```
v1 = 2; v2 = 6; h1 = 1; h2 = 1; DD = 4; theta = 17*Pi/180;
Solve[{Tan[theta] == (DD - X)/h1,
Sin[theta]/v1 == Sin[phi]/v2}, {X, phi}] // N
(* Solution *)
{{X -> 3.69427, phi -> 1.06982}}
```

Examining Equation (6.1) and Figure 6.2, it is clear that if $v_2 > v_1$, then the ray in medium 2 will be refracted away from the normal to the interface. If $v_2 < v_1$, then the ray in medium 2 will be refracted toward the normal to the interface.

Seismology is just one example of a much more general application of minimization theory. Consider the situation shown in Figure 6.3 where the problem is to determine the cost of building a power transmission line from the power plant on one side of a river to a factory on land on the other side under the reasonable assumption that it will cost a different amount to build a transmission line over water as compared with the cost over land. In this case we are concerned with the least *cost* instead of a least *time*.

We would find that the least cost configuration of the transmission line must be such that

$$\frac{\sin \theta}{\text{miles}/\$ \text{over water}} = \frac{\sin \phi}{\text{miles}/\$ \text{over land}}$$

In the example shown, the number of miles obtained per dollar over land is less than the number of miles per dollar over water, hence the cost of building over water is less than over land. Reference to Figure 6.4 shows that we can extend the ray path downward, obeying Snell's law at every point along the raypath.

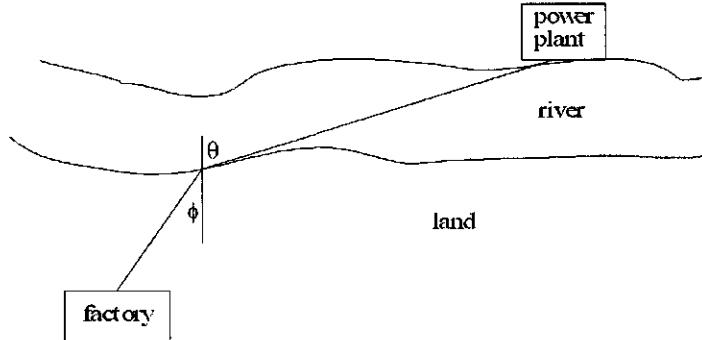


Figure 6.3: Snell's law is just a special case of a more general class of minimization problems. Here the objective is to minimize the cost of constructing a transmission line from the power plant to the factory rather than the time. (snell3)

The ratio $\text{Sin(angle)}/\text{velocity}$ is always equal to the constant value p . Clearly, the constant p is a number of great significance. We turn now to finding its numerical value.

6.1.1 The ray parameter p

Reversing the direction of travel and adding the ground surface as shown in Figure 6.5, the seismic wavefront must sweep over the distance ΔX in exactly the same time as it traverses the distance Δl . Therefore, from the geometry shown in Figure 6.5,

$$\sin \theta = \frac{\Delta l}{\Delta X} = \frac{\Delta l/\Delta t}{\Delta X/\Delta t} = \frac{v_1}{v_{app}}$$

where v_{app} is the apparent velocity as measured by the receivers shown deployed on the surface of the ground.

The relationship

$$\frac{\sin i}{v} = p \quad (6.2)$$

is valid everywhere along the raypath. Therefore, from Equation (6.2) and Figure 6.5,

$$\frac{\sin i}{v} = \frac{\sin \theta}{v_1} = \frac{1}{v_{app}} = p$$

and the numerical value of p is simply the slope of the travelttime curve. That is:

$$p = \frac{1}{v_{app}} = \frac{\Delta t}{\Delta X} \quad (6.3)$$

at the point of emergence of the wavefront. The relationships are summarized in Figure 6.6.

6.2 Reflection Traveltime Curves

Given a single horizontal interface (Figure 6.7) we want the traveltime T_1 of a reflection from the interface to a receiver located at a distance X . Application of the Pythagorean theorem to triangle SS'G in Figure 6.7 gives the following equation for the time-distance curve for a reflection from the first interface:

$$v_1^2 T^2 = 4h_1^2 + X^2$$

or

$$T^2 = T_0^2 + \frac{X^2}{v_1^2}$$

where $T_0 = 2h_1/v_1$ is the vertical two-way traveltime from the sourcepoint to a horizontal reflector and back to a receiver at the sourcepoint where $X = 0$.

Thus, the equation of a traveltime curve for a reflection from a single horizontal interface is the equation of a hyperbola with asymptotes

$$T = \pm \frac{X}{v_1}$$

The hyperbola is shown plotted above the model in Figure 6.7.

The quantity $T_1 - T_0$ is called *normal moveout*, abbreviated *nmo*.

A binomial expansion of $T_1 - T_0$ yields

$$\text{nmo} \approx \frac{X^2}{2T_0 v_1^2} \quad (6.4)$$

As a rule of thumb, for adequate determinations of stacking velocities from conventional multifold reflection seismic data, the normal moveout for the target reflection at the receiver most distant from the sourcepoint should be at least twice the wavelet duration; for example, if the target is at 2 sec two-way time, average velocity to the target is 6 km/sec, distance to farthest offset is 48 channels \times 70 m and the wavelet duration is 40 ms, then the normal moveout is about 80 ms.

Referring back to Figure 6.6, inspection of the emergence angle α of the ray path at the surface shows that

$$\sin \alpha = \frac{\Delta l}{\Delta X} = \frac{\Delta l/\Delta t}{\Delta X/\Delta t} = \frac{v_1}{\Delta X/\Delta t} = \frac{v_1}{v_X}$$

where Δt is the time required to travel the distance ΔX or Δl and v_X is the horizontal velocity. Thus,

$$\frac{\Delta t}{\Delta X} = \frac{\sin \alpha}{v_1}$$

This same result is obtained by differentiation of the time-distance hyperbola:

$$\frac{dT}{dX} = \frac{X/T}{v_1^2} = \frac{X}{v_1(v_1 T)} = \frac{\sin \alpha}{v_1}$$

From Figure 6.7:

$$\begin{aligned} X &= 2h_1 \tan \alpha \\ T &= \frac{2h_1 / \cos \alpha}{v_1} = \frac{T_0}{\cos \alpha} \end{aligned} \quad (6.5)$$

For each value of α there is a unique value of X and T . the traveltime curve of T versus X can be generated by assuming values of α from $-\pi/2$ to $+\pi/2$ and then generating values of X and T .

The two equations are said to be parametrically related by the *parameter* α . If the parameter is eliminated between these two equations, the result is

$$v_1^2 T^2 = 4h_1^2 + X^2$$

which is the equation of the traveltime hyperbola for a horizontal interface.

For any one raypath associated with the travelttime T to a receiver at a particular distance X the value of α is a unique constant. Hence, for that raypath

$$\frac{\sin \alpha}{v_1} = \text{constant} = p$$

this constant is the *ray parameter* p . Hence,

$$p = \frac{\sin \alpha}{v_1} = \frac{dT}{dX}$$

and we observe that *the slope of the travelttime curve at the point of emergence of a raypath at a distance X from the sourcepoint is equal to the value of the parameter p for that raypath*.

This is an important relationship between the slope of the travelttime curve at a distance X and the ray parameter p . the ray parameter p is thus a measurable number, and allows us to determine many of the physical properties of the Earth from the surface to its center at a depth of 6,371 km. We will also use this relationship for refraction travelttime curves.

Using the ray parameter notation we can write Equations (6.5) as

$$X = \frac{2h_1 p v_1}{(1 - p^2 v_1^2)^{1/2}} = 2h_1 \tan \alpha_{11}$$

$$T = \frac{2h_1}{v_1(1 - p^2 v_1^2)^{1/2}} = \frac{2h_1}{v_1 \cos \alpha_{11}}$$

A two-layer model is shown in Figure 6.8. Notation for angles is as follows: the first subscript refers to the number of the layer in which the segment of the raypath is located. The second subscript is the number of the deepest layer penetrated by the raypath, i.e., the number of the layer from the base of which the energy is reflected. For reflections from the second interface

$$\begin{aligned} X &= 2[h_1 \tan \alpha_{12} + h_2 \tan \alpha_{22}] \\ T_2 &= 2 \left[\frac{h_1 / \cos \alpha_{12}}{v_1} + \frac{h_2 / \cos \alpha_{22}}{v_2} \right] \end{aligned} \quad (6.6)$$

At the point of emergence of the raypath

$$\sin \alpha_{12} = \frac{\Delta l}{\Delta X} = \frac{\Delta l / \Delta t}{\Delta X / \Delta t} = \frac{v_1}{\Delta X / \Delta t}$$

where Δt is the time taken for the wavefront to sweep over the horizontal distance ΔX as it travels in the direction Δl . From the above,

$$\frac{\Delta t}{\Delta X} = \frac{\sin \alpha_{12}}{v_1}$$

Along the raypath to the deeper layer in Figure 6.8, by Snell's law

$$\frac{\sin \alpha_{12}}{v_1} = \frac{\sin \alpha_{22}}{v_2} = \text{constant}$$

The constant is the ray parameter p . Thus,

$$p = \frac{\sin \alpha_{12}}{v_1} = \frac{\sin \alpha_{22}}{v_2}$$

and, in terms of the parameter p Equations (6.6) become

$$\begin{aligned} X &= 2 \left[\frac{h_1 p v_1}{(1 - p^2 v_1^2)^{1/2}} + \frac{h_2 p v_2}{(1 - p^2 v_2^2)^{1/2}} \right] \\ T_2 &= 2 \left[\frac{h_1 / v_1}{(1 - p^2 v_1^2)^{1/2}} + \frac{h_2 / v_2}{(1 - p^2 v_2^2)^{1/2}} \right] \end{aligned} \quad (6.7)$$

It would be desirable to eliminate the parameter p from the above equations and obtain an equation of the form $T = f(X)$. Such an equation from a model with more than one interface would not be a hyperbola but would approximate a curve of higher order. In any event, it is not possible to find an *exact* expression $T = f(X)$ for the traveltimes from the base of more than one layer. It is simple, however, to deal with the parametric form of the traveltime curve, and any of the three methods described here can be used to generate traveltime curves.

Reference to Figure 6.9 shows that the time and distance equations for a reflection from the base of an n -layered stack are given by

$$\begin{aligned} T_n &= 2 \left[\frac{h_1 / \cos \alpha_{1,n}}{v_1} + \frac{h_2 / \cos \alpha_{2,n}}{v_2} + \dots + \frac{h_n / \cos \alpha_{n,n}}{v_n} \right] \\ X &= 2 [h_1 \tan \alpha_{1,n} + h_2 \tan \alpha_{2,n} + \dots + h_n \tan \alpha_{n,n}] \end{aligned}$$

or

$$\begin{aligned} T_n &= 2 \sum_{i=1}^n \frac{h_i / \cos \alpha_{i,n}}{v_i} \\ X &= 2 \sum_{i=1}^n h_i \tan \alpha_{i,n} \end{aligned} \tag{6.8}$$

From Snell's law:

$$\frac{\sin \alpha_{1,n}}{v_1} = \frac{\sin \alpha_{2,n}}{v_2} = \frac{\sin \alpha_{3,n}}{v_3} = \dots = \frac{\sin \alpha_{n,n}}{v_n} = \text{constant} = p$$

so that Equations (6.8) can be written

$$T_n = 2 \sum_{i=1}^n \frac{h_i}{v_i [1 - \sin^2 \alpha_{i,n}]^{1/2}}$$

and

$$\begin{aligned} X &= 2 \sum_{i=1}^n h_i \frac{\sin \alpha_{i,n}}{\cos \alpha_{i,n}} \\ &= 2 \sum_{i=1}^n h_i \frac{\sin \alpha_{i,n}}{(1 - \sin^2 \alpha_{i,n})^{1/2}} \end{aligned}$$

From Snell's law,

$$\sin \alpha_{i,n} = p v_i$$

so that

$$\begin{aligned} T_n &= 2 \sum_{i=1}^n \frac{h_i / v_i}{(1 - p^2 v_i^2)^{1/2}} \\ X &= 2p \sum_{i=1}^n \frac{h_i v_i}{(1 - p^2 v_i^2)^{1/2}} \end{aligned} \tag{6.9}$$

where the value of p ranges from $-\text{Min}[1/v_1, 1/v_2, \dots, 1/v_n]$ to $+\text{Min}[1/v_1, 1/v_2, \dots, 1/v_n]$. This definition of the limits for p will result in $-\infty \leq X \leq +\infty$

and $0 \leq T_n \leq +\infty$ regardless of whether the velocity increases or decreases with depth. A suitably small value of dp must be chosen. A *Mathematica* program *TraveltimeCurves* to compute and plot exact reflection traveltimes from a multilayered model with parallel interfaces using Equations (6.9) is

```
(* TraveltimeCurvesReflection.nb *)
(* Define velocities and thicknesses *)
v = {2, 3, 4, 5};
h = {0.5, 1, 1.5, 2};
NumberOfLayers = Length[h];
Plots = Table[0, {NumberOfLayers}];
Do[pmax = Min[Take[1/v, layer]]; dp = pmax/200;
  T = Table[0, {Ceiling[pmax/dp]}]; X = T; j = 0;
  Do[j = j + 1; sumT = 0; sumX = 0;
    Do[sumT = sumT + 2*(h[[i]]/v[[i]])/Sqrt[(1 - p^2*v[[i]]^2)];
     sumX = sumX + 2*p*(h[[i]]*v[[i]])/Sqrt[(1 - p^2*v[[i]]^2)], {i, 1,
      layer}]; T[[j]] = sumT; X[[j]] = sumX, {p, 0, pmax - dp, dp}];
  Print["Inverse slope (velocity) at Max[X] for Layer ", layer,
    " = ", (X[[j]] - X[[j - 1]])/(T[[j]] - T[[j - 1]]) // N, " at X = ",
    X[[j]] // N];
  Plots[[layer]] =
  ListPlot[Transpose[{X, T}], AxesOrigin -> {0, 0},
  DisplayFunction -> Identity, PlotRange -> {{0, 10}, Automatic},
  PlotJoined -> True], {layer, 1, NumberOfLayers}];
Show[Plots, DisplayFunction -> $DisplayFunction];
```

If the velocity increases with depth, then for long offsets the slope of the reflection traveltime curve for reflections from the base of the i^{th} layer will approach the interval velocity v_i of that layer. On page 266 we show that if T^2 is plotted against X^2 then the reciprocal of the slope of any of the resulting reflection traveltime curves gives a close approximation to the square of the “rms” velocity $v_{rms,i}^2$. Therefore, $v_{rms,i} \rightarrow v_i$ for long offsets. If there is a low-velocity layer of velocity v_i then the reciprocal of the slope of the travelttime curve for reflections from the base of the low-velocity layer will approach the value of the interval velocity v_{i-1} of the layer above. That is, $v_{rms,i} \rightarrow v_{i-1}$ for long offsets. This can be deduced by reference to Figure 6.12 and observing that for very long offsets most of the reflection path for reflections from the base of the low-velocity layer is spent in the layer above where the raypath eventually becomes essentially horizontal in the layer above the low-velocity layer (for very long offsets).

Equations (6.9) constitute the parametric form of the traveltime equation for a reflection from the base of an n -layered stack of horizontal reflectors. The parametric form, Equations (6.9), is the most accurate method to generate exact traveltime curves for any offset or velocity layering. Travelttime curves for four-layer models are shown in Figures 6.10, 6.11, and 6.12.

From the parametric equations, it is possible to extract a plot of p versus X for reflections from the bottom of each layer. Results for the model of Figure 6.12 are shown in Figure 6.13. For reflections from the base of the first layer, the

limiting value of p asymptotically approached by reflections to large offsets X is $p = 1/v_1$. For the model shown, the limiting value of p for large offsets for reflections from the base of the deepest layer should be about $p = 1/v_4$. A more uniform separation of the curves that describe $p(X)$ will be obtained if the velocity function is more uniform. For example, for the model shown in Figure 6.10, curves of p versus X are well separated, as shown in Figure 6.14.

The sigmoidal shapes are characteristic of $p(X)$ for any velocity layering and any reflection time.

Average velocity

The average velocity V_{ave} associated with a vertical raypath ($p = 0$) is the velocity determined during a well velocity survey by lowering a receiver on a cable into a well bore and measuring the traveltimes between a surface energy source and the receiver. The average velocity of wave propagation from the shot to the receiver is equal to the distance between the receiver and energy source divided by the observed traveltimes. Again, it is appropriate to use such an average velocity in a hyperbolic traveltime equation only for deep reflections recorded close to the sourcelpoint, i.e., only for small values of X/H .

Anticipating a later discussion of the kinds of velocity encountered in seismology, it is instructive at this point to examine further Equation (6.7). The raypath corresponding to $p = 0$ is the raypath leaving the sourcelpoint, traveling vertically downward, and returning to the sourcelpoint. For $p = 0$, Equations (6.7) become

$$\begin{aligned} X &= 0 \\ T_2 &= \frac{2h_1}{v_1} + \frac{2h_2}{v_2} = T_{0,2} \end{aligned}$$

where $T_{0,2}$ denotes the *two-way* traveltimes along a raypath that reflects at right angles from the base of the second layer. The *total* depth H to the base of the second layer is

$$H = h_1 + h_2 = \sum_{i=1}^2 h_i$$

the total traveltimes for the reflected wave is

$$\begin{aligned} T_2 &= 2 \left[\frac{h_1}{v_1} + \frac{h_2}{v_2} \right] \\ &= 2 \sum_{i=1}^2 \frac{h_i}{v_i} = 2 \sum_{i=1}^2 \tau_i \end{aligned}$$

We can thus define an *average velocity* V_{ave} as the total vertical distance traveled divided by the total traveltimes, i.e.,

$$V_{ave} = \frac{\sum_{i=1}^n h_i}{\sum_{i=1}^n h_i/v_i} = \frac{\sum_{i=1}^n v_i \tau_i}{\sum_{i=1}^n \tau_i} = \frac{\sum_{i=1}^n v_i \tau_i}{T_{0,2}/2}$$

where $\tau_i = h_i/v_i$ is the one-way traveltime through the i^{th} layer. This expression for the average velocity was derived under the assumption that $p = 0$. Hence, this velocity should not be used to generate hyperbolic reflection traveltimes except for deep reflections received close to the sourcepoint. If the ratio X/H is not small then the average velocity defined above will not accurately give the traveltime to the offset distance X . For large offset distances we require a different expression for the velocity used in the hyperbolic traveltime equation. First, however, we examine another approach to the concept of a least-time path—one that does not explicitly invoke Snell's law.

Traveltimes equations using Lagrangian multipliers

Another approach to the derivation of the traveltime T of a reflection from the base of an n -layer stack (Figure 6.15) is to use the method of Lagrange multipliers, or undetermined multipliers [175]. In general, such an approach involves the minimum of a function $f(x_i)$ and another function of the form $\phi(x_i) = \text{constant}$. First we note from calculus and the definition of a total differential that

$$\begin{aligned} df &= \left(\frac{\partial f}{\partial x_i} \right) dx_i \\ d\phi &= \left(\frac{\partial \phi}{\partial x_i} \right) dx_i \end{aligned}$$

Then we note that the total differential df is zero because we want to find the minimum of $f(x_i)$, and that $d\phi$ is zero because $\phi(x_i) = \text{constant}$.

Therefore,

$$\begin{aligned} df &= \left(\frac{\partial f}{\partial x_i} \right) dx_i = 0 \\ d\phi &= \left(\frac{\partial \phi}{\partial x_i} \right) dx_i = 0 \end{aligned}$$

Multiplying the $d\phi$ equation by λ (this is the undetermined multiplier) and adding it to the df equation gives

$$\left[\frac{\partial f}{\partial x_i} + \lambda \frac{\partial \phi}{\partial x_i} \right] dx_i = 0$$

Thus, λ must be chosen so that

$$\frac{\partial f}{\partial x_i} + \lambda \frac{\partial \phi}{\partial x_i} = 0$$

We thus have two equations in the two unknowns x_i and λ :

$$\frac{\partial f}{\partial x_i} + \lambda \frac{\partial \phi}{\partial x_i} = 0$$

$$\phi(x_i) = \text{constant}$$

and could, if we wanted, solve for the two unknowns x_i and λ .

At this point, however, we note that

$$\lambda = -\frac{\partial f/\partial x_i}{\partial \phi/\partial x_i}$$

In our particular problem, from Figure 6.15,

$$f(x_i) = 2 \sum_{i=1}^n \frac{s_i}{v_i} = 2 \sum_{i=1}^n \frac{(h_i^2 + x_i^2)^{1/2}}{v_i}$$

and

$$\phi(x_i) = 2(x_1 + x_2 + x_3 + \dots + x_n) = \text{constant} = X$$

so that

$$\frac{\partial f}{\partial x_i} = \frac{2x_i}{v_i(h_i^2 + x_i^2)^{1/2}} = \frac{\partial T_n}{\partial x_i}$$

and

$$\frac{\partial \phi}{\partial x_i} = 2$$

The Lagrange multiplier λ is

$$\lambda = -\frac{\partial f/\partial x_i}{\partial \phi/\partial x_i} = -\frac{\partial T_n/\partial x_i}{\partial \phi/\partial x_i} = \frac{-x_i}{v_i s_i}$$

Then

$$\lambda^2 = \frac{x_i^2}{v_i^2 s_i^2}$$

and

$$s_i^2 = \frac{x_i^2}{\lambda^2 v_i^2} = h_i^2 + x_i^2$$

or

$$x_i^2 = h_i^2 \lambda^2 v_i^2 + x_i^2 \lambda^2 v_i^2$$

$$= \frac{h_i^2 \lambda^2 v_i^2}{(1 - \lambda^2 v_i^2)}$$

Then

$$X = 2 \sum_{i=1}^n x_i = \pm 2\lambda \sum_{i=1}^n \frac{v_i h_i}{(1 - \lambda^2 v_i^2)^{1/2}}$$

where λ is allowed to take on either positive or negative values, and

$$T_n = 2 \sum_{i=1}^n \frac{s_i}{v_i} = 2 \sum_{i=1}^n \frac{x_i}{\lambda v_i^2} = 2 \sum_{i=1}^n \frac{h_i \lambda v_i}{\lambda v_i^2 (1 - \lambda^2 v_i^2)^{1/2}} \quad (6.10)$$

$$T_n = 2 \sum_{i=1}^n \frac{h_i/v_i}{(1 - \lambda^2 v_i^2)^{1/2}}$$

Summarizing:

$$\begin{aligned} X &= \pm 2\lambda \sum_{i=1}^n \frac{v_i h_i}{(1 - \lambda^2 v_i^2)^{1/2}} \\ T_n &= 2 \sum_{i=1}^n \frac{h_i/v_i}{(1 - \lambda^2 v_i^2)^{1/2}} \end{aligned} \quad (6.11)$$

Equations (6.11) constitute the parametric form of the time-distance equation for reflections from the base of an n -layer stack, and are the same equations as those found in Slotnick [167, p. 194, Equa. 5], and Taner and Koehler [175, p. 876, Equas. A6 and A7]. As values of λ vary from 0 to $1/v$ where $v = \max(v_1, v_2, v_3, \dots, v_n)$, X will vary from 0 to $\pm\infty$. Positive and negative values of λ are simply associated with traveltimes to receivers on each side of the sourcepoint.

The physical significance of the constant λ is obscured when the method of Lagrangian multipliers is used. The meaning of λ becomes apparent, however, when we consider that along any one reflected raypath Snell's law requires that

$$\frac{\sin \alpha_1}{v_1} = \frac{\sin \alpha_2}{v_2} = \frac{\sin \alpha_3}{v_3} = \text{constant}$$

But $\sin \alpha_i = x_i/s_i$ and therefore

$$\frac{x_i}{v_i s_i} = \text{constant} = \lambda$$

and the Lagrange multiplier λ is simply the *ray parameter*, more commonly designated p , which represents "slowness" in wave propagation. Each reflected ray path to a given distance X has associated with it a *unique* value of p . When λ is replaced by p then Equations (6.11) become (6.7) and (6.8).

6.2.1 The root-mean-square velocity

Implementing the parametric form of the traveltime equation is convenient as well as computationally efficient even though a value of X is given and the travelttime is desired. Values of p can be chosen that bracket the desired value of X , and an interpolation can be made to any desired precision.

The parametric representation is quite adequate for the generation of a travelttime curve for reflections from the base of a layered stack. It would, however, be simpler to have an explicit, although approximate, time-distance formula that expresses the travelttime T_n as a function of X . We can deduce such a formula from the parametric equations.

In the special case of a single layer the parametric equations are

$$X = \frac{2pv_1h_1}{(1-p^2v_1^2)^{1/2}}$$

and

$$T_1 = \frac{2h_1/v_1}{(1-p^2v_1^2)^{1/2}}$$

Elimination of p , which we can do for a single layer, gives

$$T_1^2 = \frac{4h_1^2}{v_1^2} + \frac{X^2}{v_1^2}$$

which is the equation of a hyperbola. Note that the equation is of the form

$$T_1^2 = \left[\frac{4h_1^2}{v_1^2} \right] X^0 + \left[\frac{1}{v_1^2} \right] X^2$$

It can be shown (Copson [43]) that in the general case there will exist a generalization of the above travelttime curve in the form of an infinite series

$$T_n^2 = c_1 X^0 + c_2 X^2 + c_3 X^4 + c_4 X^6 + \dots \quad (6.12)$$

where the constants c_1, c_2, c_3, \dots are functions of the layer thicknesses h_1, h_2, h_3, \dots and the velocities v_1, v_2, v_3, \dots

In order to evaluate the constants c_i , the quantities $T_n^2, X^2, X^4, X^6, \dots$, can be expressed as a power series in p^2 by using the binomial theorem and these quantities then substituted in the infinite series shown in Equation (6.12). The binomial theorem is

$$(a+b)^n = a^n + na^{n-1}b + \frac{n(n-1)}{2!}a^{n-2}b^2 + \frac{n(n-1)(n-2)}{3!}a^{n-3}b^3 + \dots$$

If n is a positive integer, the expansion contains only a finite number of terms and is valid for all values of a and b . If n is negative or a positive number not an integer, the expansion contains an infinite number of terms, and is valid if and only if $|a| > |b|$. That is, into Equation (6.12), we substitute for T_n and X :

$$\begin{aligned} T_n &= 2 \sum_{i=1}^n \frac{h_i/v_i}{(1-p^2v_i^2)^{1/2}} \\ X &= 2p \sum_{i=1}^n \frac{h_i v_i}{(1-p^2v_i^2)^{1/2}} \end{aligned}$$

The coefficients of like powers of p^2 on each side of the resulting equation are then equated, thus obtaining a system of equations that can be solved one at a time for c_1, c_2, c_3, \dots (See Taner and Koehler [175, 176]).

For the constant c_1 we obtain the value

$$c_1 = \left[2 \sum_{i=1}^n \frac{h_i}{v_i} \right]^2 = T_{0,n}^2 \quad (6.13)$$

and for c_2

$$c_2 = \frac{\sum_{i=1}^n h_i/v_i}{\sum_{i=1}^n v_i h_i} \quad (6.14)$$

Values of c_3, c_4, \dots can be determined from Taner and Koehler [175, Appendix A]. In some situations only c_1 and c_2 need to be considered, and sufficient accuracy for practical purposes is obtained from

$$T_n^2 = c_1 + c_2 X^2$$

This is the equation of a hyperbola and is of the same form as the exact expression for a one-layer case

$$T_1^2 = \frac{4h_1^2}{v_1^2} + \frac{1}{v_1^2} X^2$$

The constant c_2 has the dimensions of $1/(\text{velocity})^2$. The question is then: what is the difference between the velocity v_1 for a one-layer case and the velocity $1/\sqrt{c_2}$ for the n -layer case, and, in particular, what physical significance can be attached to the velocity $1/\sqrt{c_2}$?

Recalling from Equation (6.14) that

$$\frac{1}{c_2} = \frac{\sum_{i=1}^n v_i h_i}{\sum_{i=1}^n h_i/v_i}$$

The quantity $2 h_i/v_i$ is twice the one-way vertical traveltimes τ_i through the i^{th} layer, so that

$$\frac{1}{c_2} = \frac{\sum_{i=1}^n \tau_i v_i^2}{\sum_{i=1}^n \tau_i}$$

Hence $1/c_2$ is the weighted average of the squares of the interval (true) velocities v_i . The weights are equal to the interval velocities v_i .

The velocity $1/\sqrt{c_2}$ is defined as the *root-mean-square (rms)* velocity. We can thus write the travelttime hyperbola as

$$T_n^2 = c_1 + \left[\frac{1}{v_{\text{rms},n}^2} \right] X^2$$

where

$$v_{rms,n}^2 = \frac{\sum_{i=1}^n v_i h_i}{\sum_{i=1}^n h_i/v_i} = \frac{\sum_{i=1}^n \tau_i v_i^2}{\sum \tau_i} \quad (6.15)$$

and τ_i is the vertical traveltimes in the i^{th} layer. Equation (6.15) is the result that led to the famous *Dix equation* [61, Equation 12, p. 73] that allows determination of the interval velocities v_i from the rms velocities $v_{rms,i}$ by the inverse relationship

$$v_n^2 = \frac{v_{rms,n}^2 \sum_{i=1}^n \tau_i - v_{rms,n-1}^2 \sum_{i=1}^{n-1} \tau_i}{\tau_n} \quad (6.16)$$

As noted by Dix [61], errors made on earlier and later interfaces have no effect on the n^{th} interval velocity.

If, for example, $v_{rms,n-2}^2$ corresponded to a multiple but was included by mistake, this mistake would have no effect on the calculation of v_n .

Computations of velocities v_i using (6.16) must be made with caution unless the data quality is excellent and the reflecting interfaces are horizontal.

Equation (6.15) can be used to generate approximately hyperbolic traveltimes curves for use in applying the *normal moveout correction*.

Note that because $v_{rms,n}$ is the velocity that appears in the equation

$$T_n^2 = c_1 + \left[\frac{1}{v_{rms,n}^2} \right] X^2$$

we should then expect the hyperbolic track defining the (approximate) arrival times T_n of the reflection on the seismogram to be defined by the rms velocity. We also note from Equation (6.13) that

$$c_1 = T_{0,n}^2$$

where $T_{0,n}$ is the *two-way* vertical traveltimes of a reflection from the base of the n^{th} layer returning to the sourcepoint ($X = 0$). Therefore

$$T_n^2 = T_{0,n}^2 + \frac{X^2}{v_{rms,n}^2}$$

Equation (6.15) can be used to generate traveltimes curves or to apply the normal moveout correction as long as the ratio X/H , the ratio of offset to reflector depth, is not too large. As the offset decreases, the time difference decreases between that along the actual paths using the exact parametric equations and that along straight paths defined by the *rms* velocity. The actual paths get

straighter as the value of the rms velocity approaches the average velocity for which $p = 0$.

As an example of the computation of traveltime curves using the *rms* velocity, we use the model shown in Figure 6.16. The *rms* velocity for reflections from the base of the 4th layer is

$$v_{rms,4} = \sqrt{\frac{\sum_{i=1}^4 v_i h_i}{\sum_{i=1}^4 h_i/v_i}} = \frac{(2)(1) + (4)(1) + (2.5)(0.5) + (5)(1)}{1/2 + 1/4 + 0.5/2.5 + 1/5} = \sqrt{\frac{12.25}{1.15}} = 3.2637$$

so that the travelttime equation for reflections from the base of the 4th layer is

$$T_4 = \sqrt{\frac{X^2}{3.2637^2} + T_0^2}$$

which is the equation of a hyperbola, and is simpler, although less accurate, than the parametric form of the equations. Shah and Levin [162] described results of using the constant c_3 in the equation

$$T_n^2 = c_1 + c_2 X^2 + c_3 X^4 \quad (6.17)$$

where

$$c_3 = \frac{1 - [v_{rms}^4 / (v_{rms}^2)]^2}{4T_{0,n} (v_{rms}^2)^2}$$

and

$$v_{rms}^4 = \frac{\sum_{i=1}^n v_i^4 \Delta t_i}{\sum_{i=1}^n \Delta t_i}$$

The term in (6.17) containing the constant c_3 is normally used in seismic exploration processing today.

Here is a *Mathematica* program *VrmsComputation.nb* that computes the rms velocity associated with reflection from the bottom of each of the layers in the 4-layer model shown in Figure 6.16.

```

n = 4;
h = {1, 1, 0.5, 1}; v = {2, 4, 2.5, 5};
τ = Table[2 h[[i]]/v[[i]], {i, 1, n}];
T₀ = Table[Sum[2 h[[i]]/v[[i]], {i, 1, j}], {j, 1, n}];
Vrms2 = Table[Sum[τ[[i]] * v[[i]]2, {i, 1, j}]/Sum[τ[[i]], {i, 1, j}], {j, 1, n}];
Vrms = Sqrt[Vrms2];
Vrms = {Vrms,1, Vrms,2, Vrms,3, Vrms,4} Km/sec;

```

Various models generated using Equation (6.17) are shown in Figure 6.18. Ac-

tual rms velocities determined in major areas of oil exploration are shown in Figure 6.19.

6.2.2 Determination of interval velocities using Dix Equation

If we can observe and measure the arrival time of a reflection from the base of the n^{th} layer on a seismogram, then we can determine values of T_n for many values of X . Once these are known, we can compute depths to the reflectors. Values of T_n^2 and $T_{0,n}^2$ are measured directly from the seismogram. Values of X are also known. Note that if values of T_n^2 are plotted against values of X^2 , the result is a straight line with slope $1/v_{rms,n}^2$. We will then be able to compute values of v_i , the interval velocity. The above equation can therefore be solved for $v_{rms,n}$.

For reflections from the base of a single layer, $n = 1$. Then

$$T_1^2 = T_{0,1}^2 + \frac{X^2}{v_{rms,1}^2}$$

where $T_{0,1}$ is the two-way vertical traveltime to the base of the first layer, and $v_{rms,1}$ is the rms velocity associated with a reflection from the base of the first layer. It is apparent that this rms velocity is identically equal to the interval velocity of the first layer. For reflections from the base of the second layer, $n = 2$. Then

$$T_2^2 = T_{0,2}^2 + \left[\frac{1}{v_{rms,2}^2} \right] X^2$$

where $T_{0,2}$ is the two-way vertical traveltime to the base of the second layer, and $v_{rms,2}$ is the rms velocity associated with a reflection from the base of the second layer.

If T_2^2 is plotted against X^2 the slope of the resulting straight line is $1/v_{rms,2}^2$. Hence, $v_{rms,2}$ is determined from the data.

Returning to

$$v_{rms,n}^2 = \frac{\sum_{i=1}^n \tau_i v_i^2}{\sum_{i=1}^n \tau_i} \quad (6.18)$$

For a single layer $v_{rms,1} = v_1$. For $n = 2$ this becomes

$$v_{rms,2}^2 = \frac{\tau_1 v_1^2 + \tau_2 v_2^2}{\tau_1 + \tau_2}$$

The only unknown in this equation is v_2^2 , the square of the interval velocity of the second layer. The interval velocity of the second layer is given by

$$v_2 = \sqrt{\frac{v_{rms,2}^2 T_{0,2} - v_{rms,1}^2 T_{0,1}}{T_{0,2} - T_{0,1}}}$$

The extension to n layers is obvious and given by the Dix equation

$$v_n = \sqrt{\frac{v_{rms,n}^2 T_{0,n} - v_{rms,n-1}^2 T_{0,n-1}}{T_{0,n} - T_{0,n-1}}} \quad (6.19)$$

Thus, we have determined the interval velocities v_i , which can be more closely associated with the geology and with rock types than are the *rms* velocities.

With reference to the equation

$$v_{rms,2}^2 = \frac{\tau_1 v_1^2 + \tau_2 v_2^2}{\tau_1 + \tau_2}$$

we can now revisit Figure 6.16 (page 282) and show that, for very long offsets, the slope of the reflection traveltime curve for events reflected from the base of a low-velocity layer v_i is $1/v_{i-1}$. Note from the figure that, for very long offsets and because of the path being refracted downward toward the normal to the interface (Snell's law), the raypath will spend relatively little vertical time τ_2 in the low-velocity layer with respect to the vertical time τ_1 spent in the overlying layer. That is,

$$\tau_1 \gg \tau_2$$

and we get

$$\lim_{X \rightarrow \infty} v_{rms,2}^2 = \lim_{X \rightarrow \infty} \frac{\tau_1 v_1^2 + f_2 \beta_2^2}{\tau_1 + f_2} = \frac{\tau_1 v_1^2}{\tau_1} = v_1^2$$

and the reciprocal of the slope of the travelttime curve associated with reflections from the base of the low-velocity layer of velocity v_2 approaches v_1 , as deduced earlier from Figure 6.16.

If the reflecting interfaces are both parallel and horizontal, the velocities $v_{rms,i}$ are called *rms velocities*; if the interfaces are not horizontal they are called *stacking velocities* because they are the velocities used in data processing that result in the best-looking record section.

Velocity analyses on modern computers are commonly based on the relationship between travelttime and offset distance:

$$T_i^2 = \left[\frac{1}{v_{rms,i}} \right]^2 X^2 + T_{0,i}^2$$

where $v_{rms,i}$ is the rms or stacking velocity, and could be obtained by plotting T_i^2 against X^2 and taking the square root of the reciprocal of the slope of the resulting (approximately) straight line. Because of the large amounts of data that must be processed the older " $T^2 - X^2$ " graphical methods are simply replaced by coherency computer plots of *velocity spectra* that indicate the stacking velocity that results in the highest coherence—best looking—record section.

The procedure [175] is as follows. Choose a data window centered about a two-way normal-incidence reference time T_0 . Choose also reasonable (from

experience in the area) minimum and maximum values of v_{rms} for use in Equation 6.15. Keeping T_0 constant and choosing the minimum assumed value of v_{rms} let the data window sweep across the traces in the gather along a hyperbolic track defined by Equation 6.15 as shown in Figure 6.20.

The contents collected within the data window are then subjected to some kind of a test for coherence such as an unnormalized crosscorrelation, an energy-normalized crosscorrelation ("semblance"), or summing [123]. The "stacking velocity" v_{rms} is then varied at regular intervals between a minimum and maximum v_{rms} for a given area. For the correct value of v_{rms} the coherence between the traces in the gather will be a maximum, and that value of v_{rms} that indicates the highest coherence can then be used to apply a normal moveout correction corresponding to the time T_0 . A new T_0 is then chosen at, say, a time equal to half the time gate (data window) down the traces. Another sweep of predetermined values of v_{rms} is made at this new T_0 . This procedure is repeated down the record until the final analysis value of T_0 is reached. Each pair of values of T_0 and V_{rms} , including those sets that resulted in low coherence, corresponds to a particular hyperbolic track. A plot of the coherence obtained for each pair of T_0 and v_{rms} is called a velocity spectrum and is shown in Figure 6.21.

Direct examination of velocity spectra may be bypassed entirely, and the correct stacking velocity obtained by trial and error with the best-looking (most coherent) result (preferred geologic interpretation) chosen as the "correct" stacking velocity. The best-looking result is commonly associated with the value of the stacking velocity that results in the largest wavelet amplitude when values of the i^{th} reflection are summed along the hyperbola defined by

$$T_i = \sqrt{\left[\frac{1}{v_{rms,i}}\right]^2 X^2 + T_{0,i}^2}$$

The trial "best-fit" velocities are restricted to those satisfying the above relation between T_i , $T_{0,i}$, and X .

Interval velocities can be determined from Equation (6.18) if the reflectors are horizontal and there are no lateral variations in velocity. With real data, this is seldom the case and interval velocities computed using Equation (6.15) might be seriously in error. Furthermore, static corrections affect values of stacking velocities and therefore values of interval velocities.

The enormous amount of data involved in processing a multifold CDP seismic line necessitates an automated approach to the determination of stacking velocities. The method most commonly used is to generate *constant-velocity panels*, which are actually mini-stacks composed of several tens of traces at selected intervals along a seismic line. A typical velocity panel is shown in Figure 6.22.

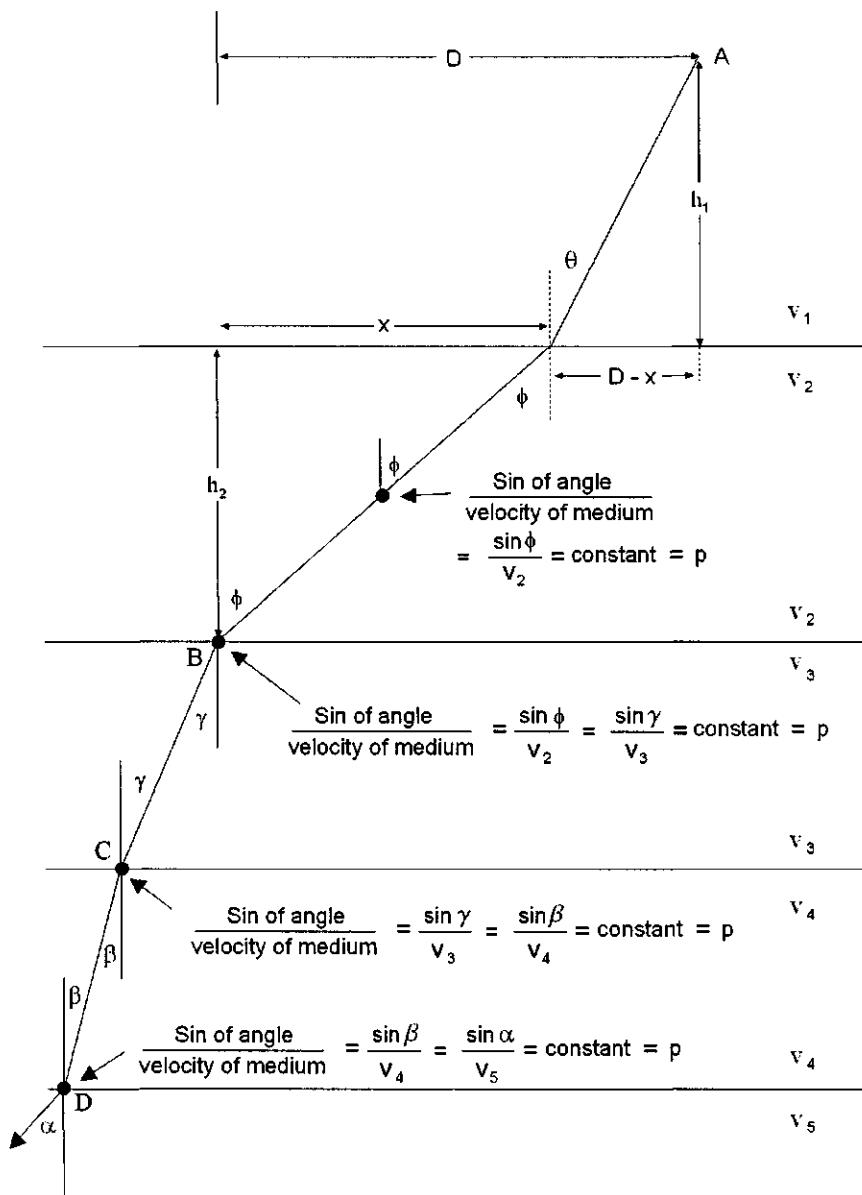


Figure 6.4: The raypath can be extended downward, obeying Snell's law — $\sin(\text{angle})/\text{velocity} = \text{constant}$ — at every point. The angle of incidence or refraction is between the normal to the interface and the raypath segment. The velocity is the velocity of the layer in which the raypath segment occurs. (snell.cdr \Rightarrow snell4.wmf)

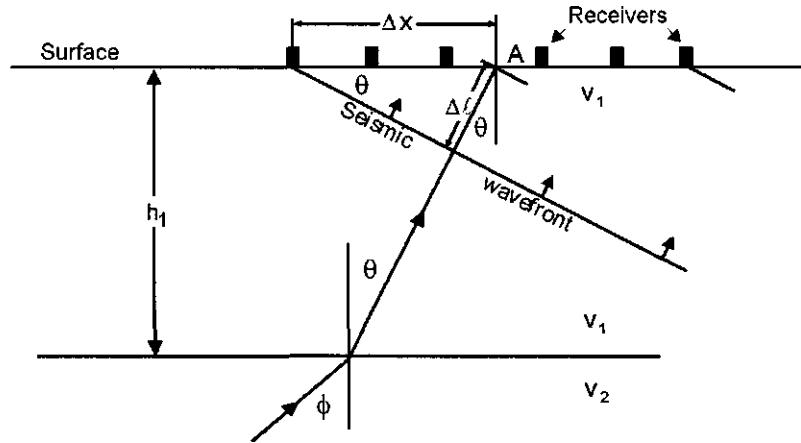


Figure 6.5: Apparent velocity. The upward-traveling seismic wavefront must traverse the distance ΔX in exactly the same time Δt as it traverses the distance Δl . Black rectangles on surface are receivers that record the time of arrival of the wavefront at each receiver. From the figure, $\sin \theta = \Delta l / \Delta X$. (snell.cdr == snell5.wmf)

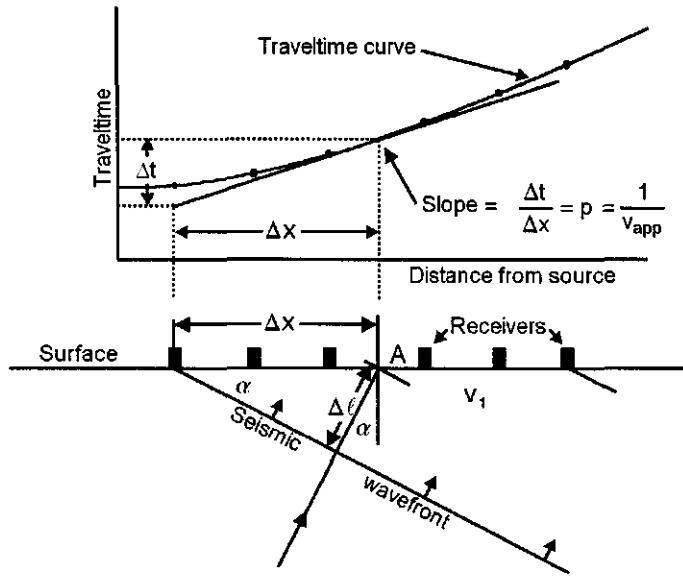


Figure 6.6: Relationship between the emergence angle α of the wavefront and the slope $\Delta t / \Delta X$ of the travelttime curve. The ray parameter p is equal to the slope $\Delta t / \Delta X$ of the travelttime curve as $\Delta X \rightarrow 0$. (snell6)

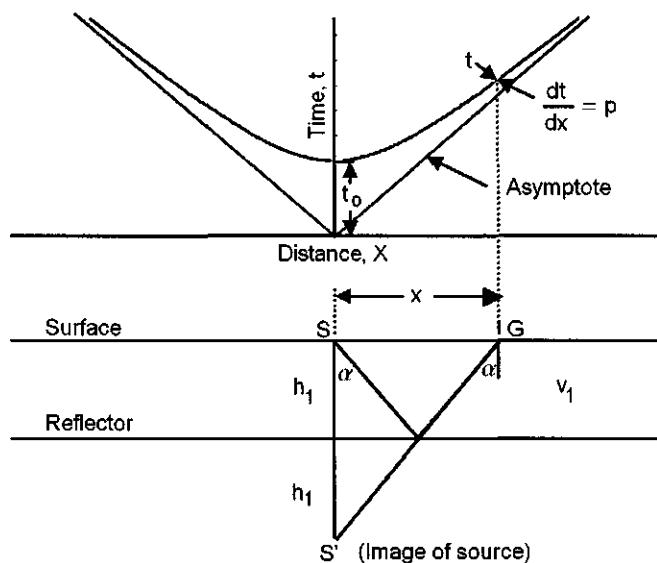


Figure 6.7: Reflections from a single interface define a hyperbolic traveltime curve.
(ssg)

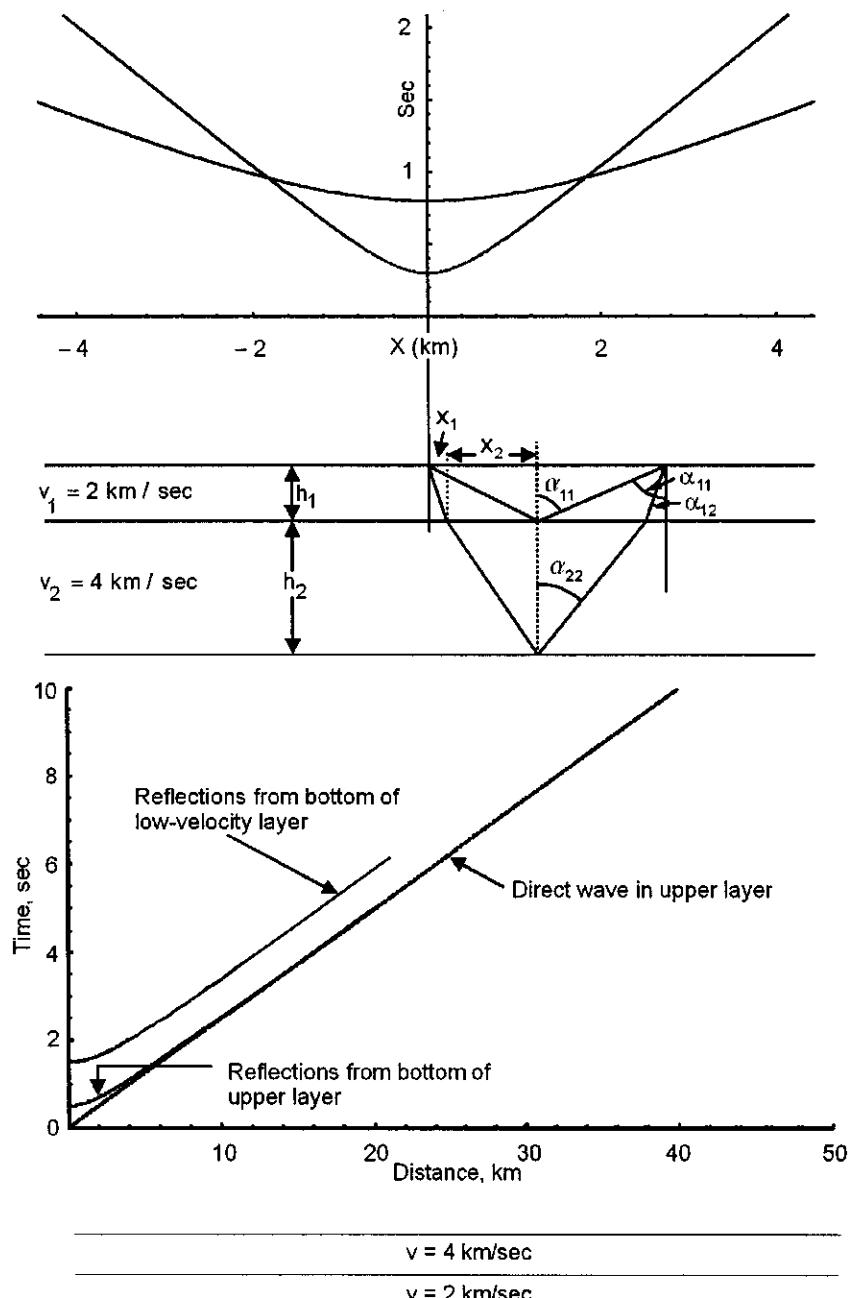


Figure 6.8: Traveltime curves from a 2-layer model. Top: If the velocity increases with depth they interfere with each other. Bottom: If the velocity decreases with depth they do not. (TraveltimeCurves2.nb, Snell, two, twol)

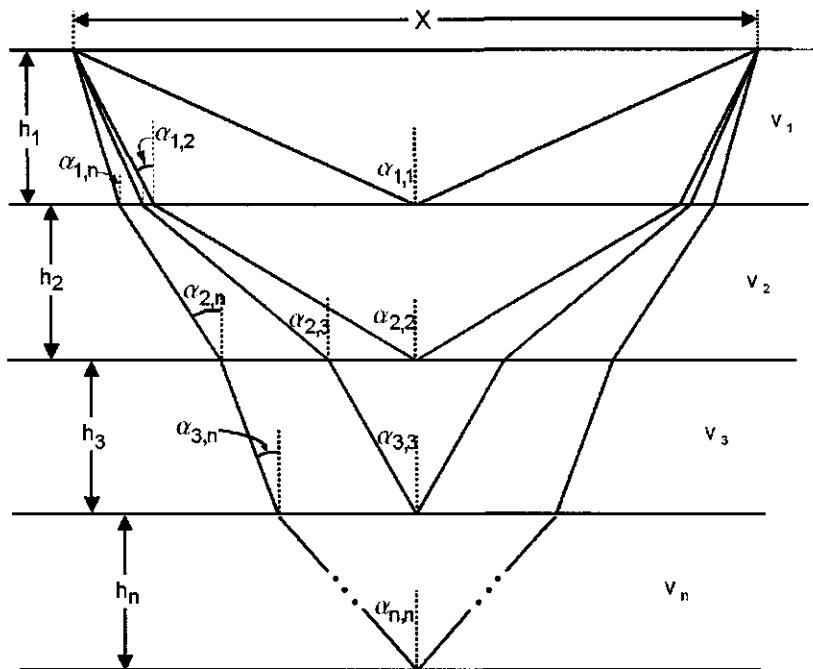


Figure 6.9: Angle and layer thickness notation for an n -layer reflection model. The first subscript on an angle refers to the layer number in which the ray segment is located; the second subscript refers to the deepest layer encountered along the entire ray path. (multin)

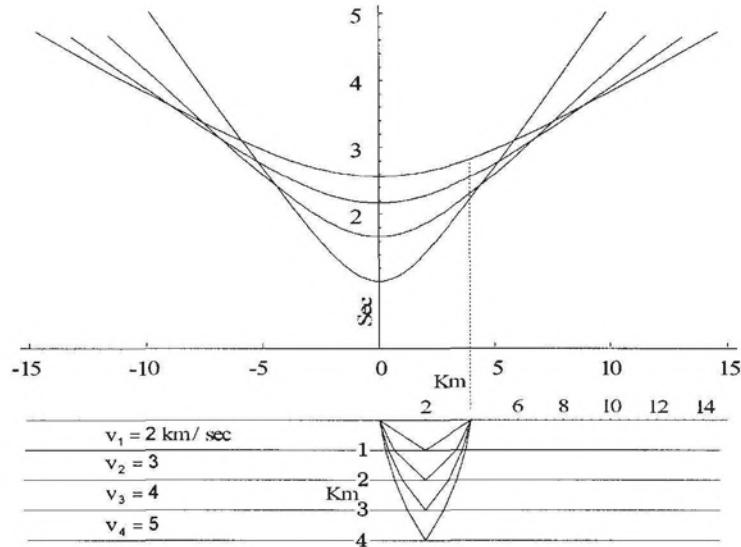


Figure 6.10: Traveltimes from a 5-layer model. The velocity of each layer increases with depth. Layer thicknesses each 1 km. Most reflection seismic data acquisition is in the region before the traveltimes curves start to cross, $-4 \text{ km} \leq X \leq +4 \text{ km}$ in the above figure. (multi4)

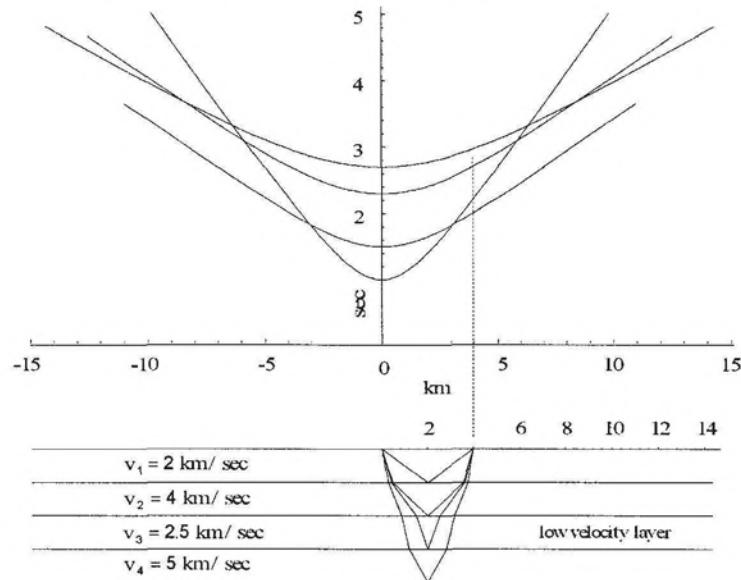


Figure 6.11: Traveltimes from a 5-layer model with a low-velocity layer. Unlike head waves (refractions), reflections are received from all interfaces, regardless of whether they are high-speed or low-speed layers. Layer thicknesses each 1 km. (multi4l)

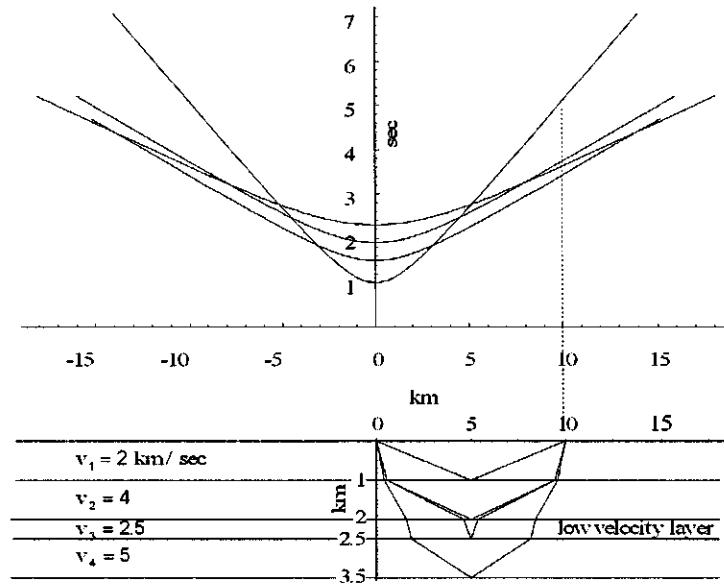


Figure 6.12: Wide-angle reflections to a receiver at 10 km from a 4-layer model with a low-velocity layer. Unlike head waves (refractions), reflections are received from all interfaces, regardless of whether they are high-speed or low-speed layers. Layer thicknesses as shown. (multi4la)

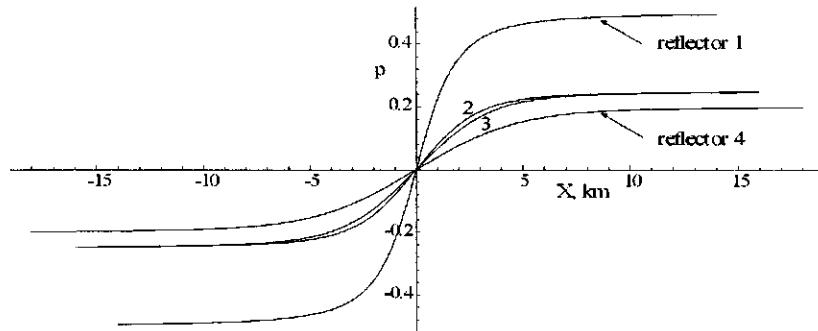


Figure 6.13: Plots of the ray parameter *p* versus offset *X* for reflections from the bottom of each layer of the velocity model shown in Figure 6.10. The sigmoidal shape is characteristic of all exact traveltime curves for any velocity layering. (pvsx.wmf)

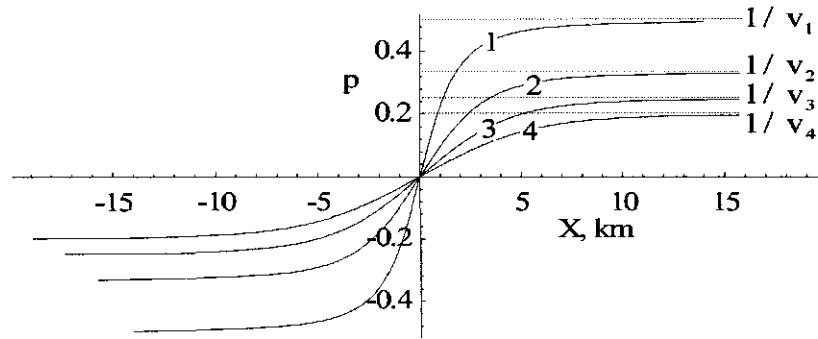


Figure 6.14: Plots of the ray parameter p versus offset X for reflections from the bottom of each layer of the velocity model shown in Figure 6.10. The curves are more uniformly separated if the velocity versus depth is relatively uniform. Numbers over curves refer to reflectors at bottoms of layers. Limiting values of p shown for each curve. (pvsx2)

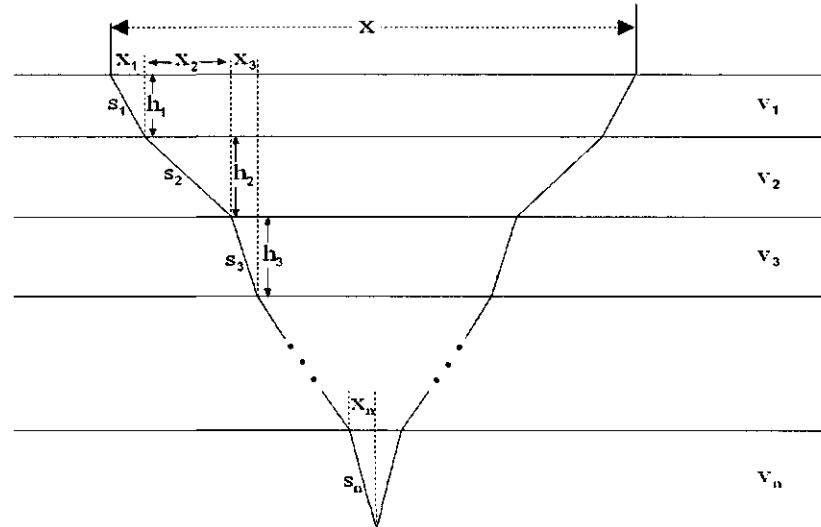


Figure 6.15: Reflections from a transversely isotropic multi-layer model (number of interfaces > 1) do not follow a hyperbolic travelttime curve, but nevertheless traveltimes can be determined to any desired degree of accuracy. (multi)

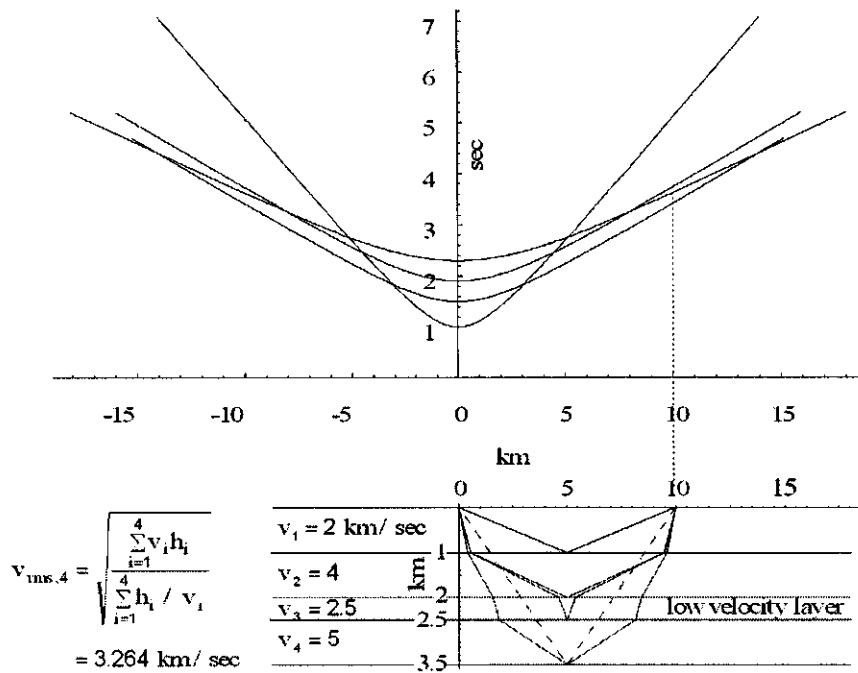


Figure 6.16: Computation of the rms velocity for reflections from the bottom of the 4th layer. The dashed straight line in the model from the sourcepoint to the reflection point is the fictitious traveltime path defined by Equation (6.15) that uses the rms velocity for the 4th layer. The traveltime curve defined by the rms velocity and the exact curve defined by the parametric equations are almost identical if the offset distance is not too large; i.e., if the offset/depth ratio X/H is not too large. If it is, then the next term in the power series expansion can be used, as shown in Figure 6.17. (multi4lr)

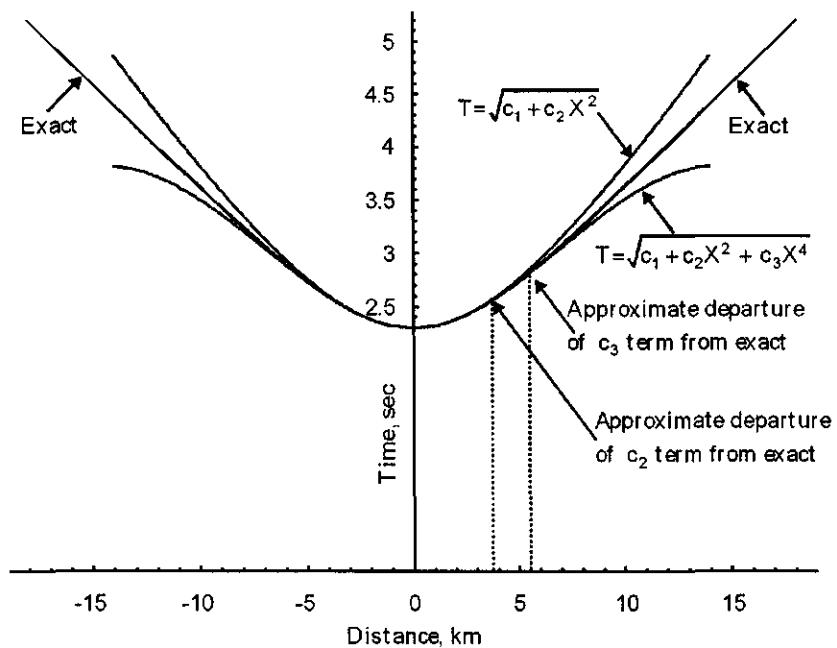


Figure 6.17: Comparison of traveltime computations using exact parametric equations, Equation (6.15), and the third term in the power series expansion of T^2 . The velocity model is shown in Figure 6.16. For this model, the rms velocity defined by Equation (6.15) is adequate for offset distances of about 4 km. Using three terms in the power series expansion extends the accuracy by about two km. (dixshah.wmf)

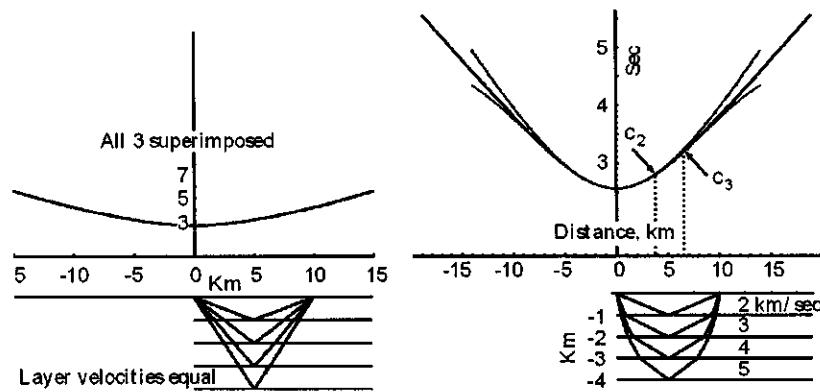


Figure 6.18: Traveltime curves from the base of the 4th layer comparing (1) exact results (parametric equations), (2) using Equation (6.15), and (3) the third term of the power series expansion. It is clear that for large values of the ratio X/H (shallow applications!) the exact parametric equations can provide a better estimate of normal moveout. (c3)

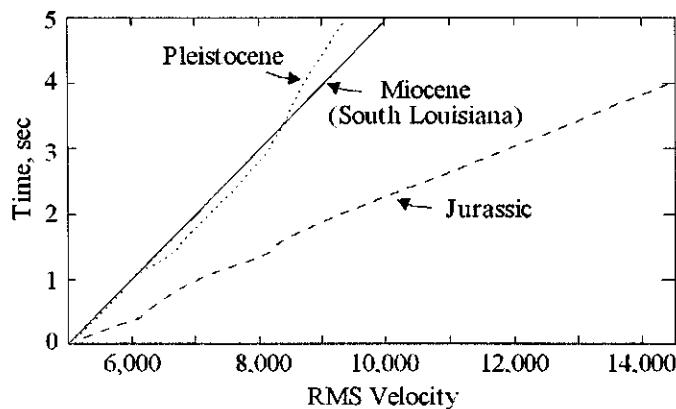


Figure 6.19: RMS velocity versus two-way traveltimes for Pleistocene, Miocene, and Jurassic sediments. Courtesy of Amoco Production Company (cd \Rightarrow VelocityAmoco)

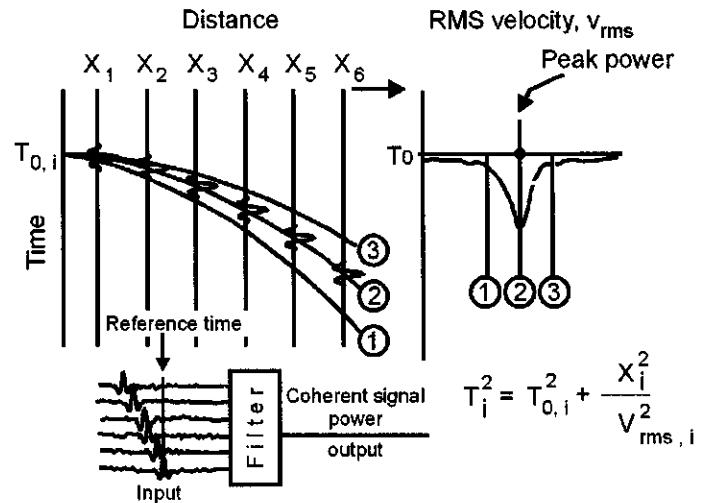


Figure 6.20: Trace gather for an *rms* velocity search. Figure from Taner and Koehler [175, Figure 6] (TanerKoehlerFig6.wmf)

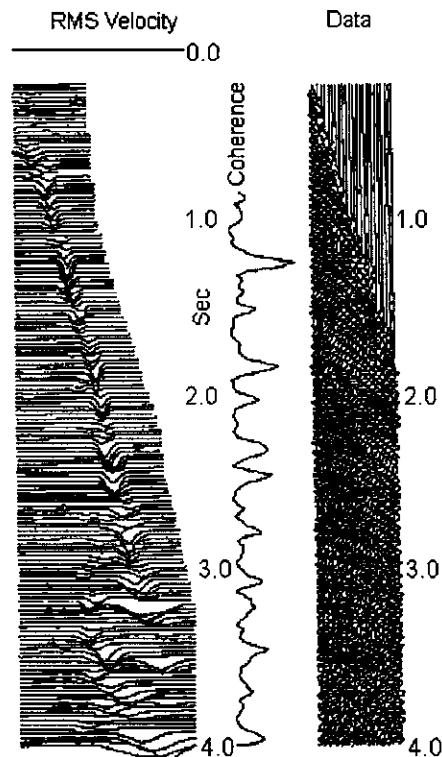


Figure 6.21: Velocity spectrum (left) obtained from data (right). Figure from Taner and Koehler [175, Figure 7]. (TanerKoehlerFig7.bmp)

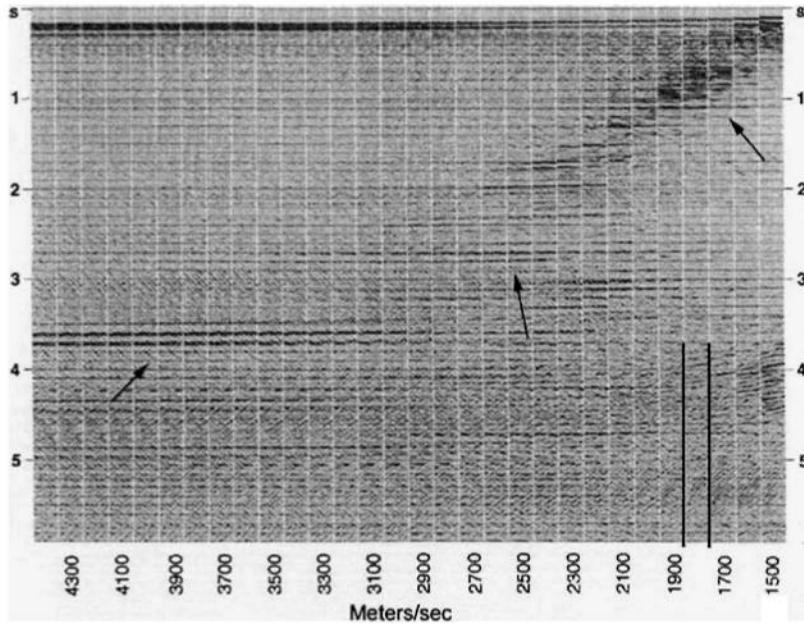


Figure 6.22: Velocity panel used to determine “stacking velocities”. A constant-velocity mini-stack is generated by using the same set of 24 CMP gathers and velocities from 1500 meters/sec to 4400 meters/sec. Two heavy vertical lines bracket the mini-stack for a stacking velocity of 1800 m/sec. Arrows mark a few locations where the stacking velocity used has brought out the best coherence and continuity for that mini-stack at that particular time. Such panels are generated many times along a line. From the velocity panels a stacking-velocity function can be defined for a final stack of the entire line on the basis of the visual picks of the best coherence seen on the individual mini-stacks. Figure modified from Yilmaz [203, p. 299].(cd → vpanel)

6.2.3 Effect of dip on reflection traveltime curves

The general effect of dip on reflection traveltime can be shown by applying the law of cosines to the triangle ($SS'X$) shown in Figure 6.23.

For this simple geometry and a constant velocity v_1 the travelttime T is found to be

$$\begin{aligned} d^2 &= X^2 + (2h)^2 - 2(2h)(X) \cos(90^\circ + \theta) \\ T^2 &= \frac{X^2}{v_1^2} + \frac{4h^2}{v_1^2} - 2 \left(\frac{2h}{v_1} \right) \left(\frac{X}{v_1} \right) \cos(90^\circ + \theta) \end{aligned}$$

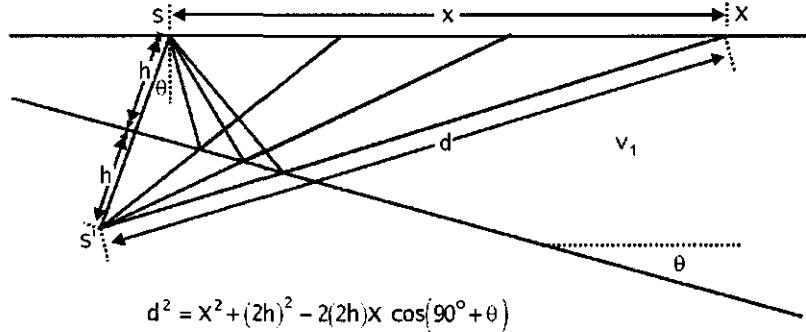


Figure 6.23: Reflection from a single dipping interface. (dipfig)

$$= \frac{X^2}{v_1^2} + \frac{4h^2}{v_1^2} + \frac{4hX}{v_1^2} \sin(\theta)$$

The effect of dip is to move the apex of the hyperbola updip (Figure 6.24), and to decrease moveout in that direction.

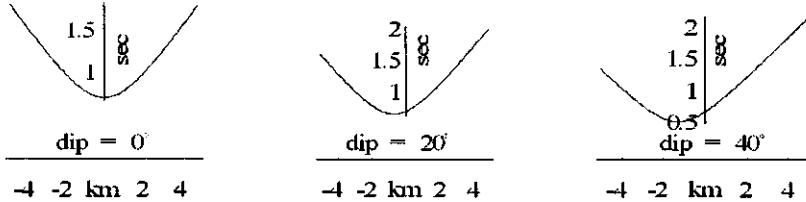


Figure 6.24: Effect of dip on traveltime curve. Single reflector, constant velocity. Reflector (not shown) dips to right and increases in dip from left to right. Apex of traveltme hyperbola moves up-dip. Normal moveout is decreased. (refldip)

The traveltme equation for reflection from a horizontal reflector from a source at S to a receiver at R is (see Figure 6.25a)

$$T = \sqrt{T_0^2 + \frac{X^2}{v^2}} \quad (6.20)$$

For a horizontal reflector the midpoint M at $X/2$ is directly above the reflection point labeled M' . For a dipping reflector this is no longer true, as shown in Figure 6.25b [167]. We wish to determine the traveltme T for the case in which a plane reflector dips at the angle θ as shown in Figure 6.25b. We will then compare this result with Equation 6.20. The desired result of finding T as a function of X in the presence of dip can be obtained by simply using the Law of Cosines and the trigonometric identity $\cos 2\theta = 2\cos^2 \theta - 1$.

From Figure 6.25b and applying the law of cosines ($c^2 = a^2 + b^2 - 2ab \cos C$) to the triangle $O - S' - R$ where $c^2 \equiv (S'R)^2$ and the angle $C = 2\theta$ we get

$$\begin{aligned} h &= \frac{vT_o}{2} \\ OR &= \frac{MM'/2}{\sin \theta} - X \\ OS &= OR + X \\ OS' &= OS \\ (S'R)^2 &= (OR)^2 + (OS')^2 - 2(OR)(OS') \cos 2\theta \\ T^2 &= \frac{(OR)^2 + (OS')^2 - 2(OR)(OS') \cos 2\theta}{v^2} \end{aligned} \quad (6.21)$$

$$(6.22)$$

Letting *Mathematica* simplify Equation 6.22 by executing

```
h=v*To/2;
OR=(MM/2)/Sin[theta]-x/2;
OS=OR+x;
OSp=OS;
OR^2+OSp^2-2*OR*OSp*Cos[2*theta]/v^2;
Simplify[%]
```

which gives

$$\begin{aligned} T^2 &= \frac{2(MM')^2 + X^2 + X^2 \cos 2\theta}{2v^2} \\ &= \frac{2(MM')^2 + X^2 + X^2(2\cos^2 \theta - 1)}{2v^2} \\ &= \frac{2(T'_o)^2 v^2 + 2X^2 \cos^2 \theta}{2v^2} \\ &= (T'_o)^2 + \frac{X^2 \cos^2 \theta}{v^2} \\ &= (T'_o)^2 + \frac{X^2}{V_{NMO}^2} \end{aligned} \quad (6.23)$$

where T'_o is the two-way time from the midpoint M along a line perpendicular to the reflector, and $V_{NMO} = v/\cos \theta$ is an apparent velocity greater than or equal to the true velocity v (Cressman [54]). This is equivalent to Equation (4) of Levin [109, p. 511]. The apparent velocity V_{NMO} will flatten the dipping reflections, but this apparent velocity is not the same as the true velocity v . The use of V_{NMO} allows one to stack common-offset data for steep or small dips. Equation (6.23) indicates that proper stacking of a dipping event requires a velocity that is greater than the velocity of the medium above the reflector [203, p. 286]. Equation (6.23) is known as Levin's dip correction [109].

Reflection curvature associated with normal moveout and reflector dip is “removed” by application of the normal moveout correction. As will become

evident, we do not "remove" normal moveout; rather we "apply a normal moveout correction". The consequences of this and how it affects wavelet shape are discussed in the next section.

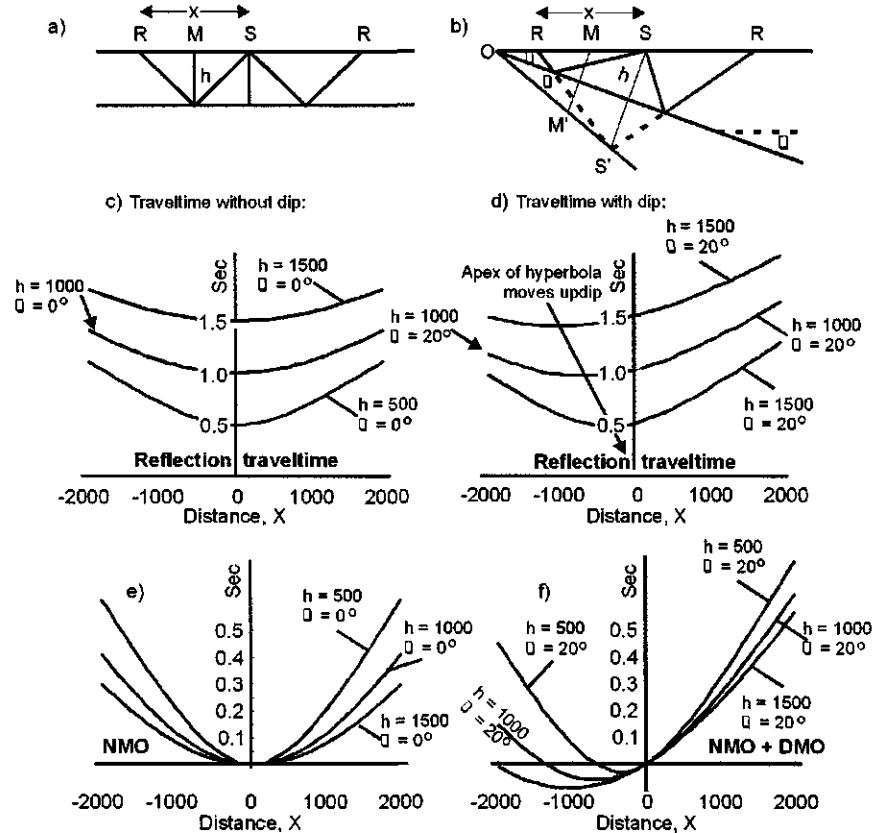


Figure 6.25: Effect of dip θ on reflection moveout. Single reflector, constant velocity $v = 2000$ m/sec. (a) Zero-dip model. (b) 20° -dip model. S = source. R = receiver. M = midpoint. S' = image of source. (c) Traveltime curves for no-dip model for depth h to reflector of = 500, 1000, 1500 m. (d) Traveltime curves for 20° dip as a function of offset and depth to reflector. Apex of traveltime hyperbola moves up-dip as depth to reflector increases. (e) Normal moveout (NMO). (f) NMO plus dip moveout (DMO). Moveout is decreased in updip direction, increased in downdip direction. (Reflection Traveltime Curve.nb \Rightarrow NMO.cdr \Rightarrow NMO.wmf)

6.2.4 The normal moveout correction

Recording configurations necessarily introduce reflection curvature even from a horizontal reflector. The curvature provides the time information required to determine the true subsurface velocity (rather than the horizontal component of the true subsurface velocity). But the curvature must be removed in order to provide an accurate image of the subsurface geometry. According to (6.4) normal moveout decreases with two-way traveltimes to the reflector, decreases as the square of the velocity, and increases as the square of the offset distance x . Two of these relationships are illustrated in Figure 6.25. The presence of dip introduces an additional factor. Several synthetic traces generated using the geometry of Figure 6.25b are shown in Figure ??a (no-dip) and b (20° dip).

First we apply the normal moveout correction to the reflections generated from the zero-dip model. This amounts to shifting each point to an earlier time by the amount NMO where

$$\begin{aligned} NMO &= T - T_o \\ &= \sqrt{T_o^2 + \frac{X^2}{v^2}} - T_o \\ &= \left(T_o + \frac{X^2}{2 v^2 T_o} + \frac{X^4}{8 v^4 T_o^3} + \frac{X^6}{16 v^6 T_o^5} + \frac{5 X^8}{128 v^8 T_o^7} + \dots \right) - T_o \\ &\approx \frac{X^2}{2 v^2 T_o} \end{aligned} \quad (6.24)$$

after a binomial expansion and retaining only the first two terms of the expansion. A more accurate approach, and that commonly used for the normal moveout correction, is obtained by retention of the X^4 term. The difference in traveltimes is shown in Figure 6.17 on page 283. The collection of traces used is that shown in Figure 6.26a generated by the program *NMOApplication.nb* on the CD-ROM that accompanies this volume. A minimum-delay wavelet (MinimumDelayWavelet.txt) was used.

In Figure 6.26b the normal moveout has been “removed” by the application of Equation 6.24 and accomplished by shifting each point in a trace by an amount of time given by Equation (6.24). For each value of T_o (corresponding to zero offset) the trace sample at offset X has been shifted to an earlier time by the amount NMO defined by Equation (6.24). This has resulted in wavelet stretching because the point-by-point application of (6.24) has been applied to a wavelet of finite duration causing the earlier parts of the wavelet to be shifted upward more than the later parts because the NMO correction is not linear. The reader is invited to explore this effect further using different wavelets, offsets, and T_o -times. For zero dip the NMO-corrected traces are now commonly considered to be ready for stacking. The result will not be a stack of the original wavelet shape, because of wavelet stretching, but generally an acceptable approximation to it.

In Figure 6.27a a 5-trace gather has been generated over a reflector with a dip of 20° . Equation (6.24) has again been used to apply the normal move-

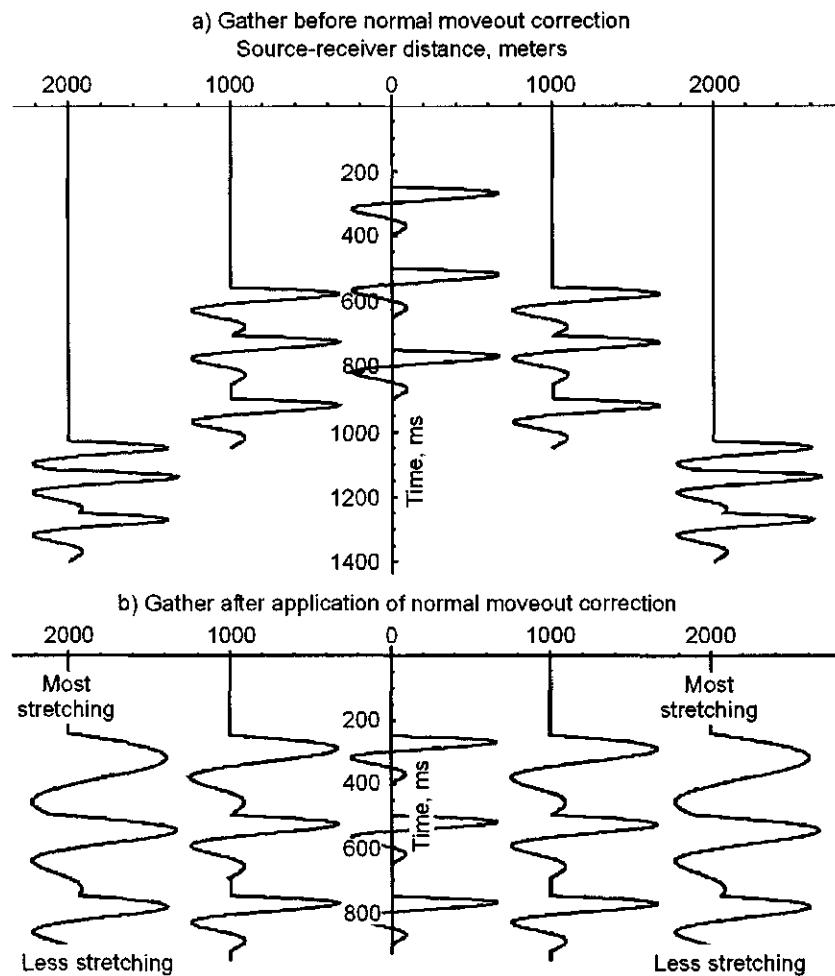


Figure 6.26: (a) Synthetic 5-trace gather acquired over zero-dip reflector. Recording geometry as in Figure 6.25a. A horizontal reflector is not recorded as a horizontal reflector. (b) Same traces after application of the normal moveout correction. Note stretching. After application of the NMO correction the onset reflection arrival times are correct. The effect of the recording geometry has been removed and a horizontal reflector is now recorded as a horizontal reflection, but the wavelet shapes now differ markedly. This is undesirable because traces in a gather are to be summed together in order to increase the signal-to-noise ratio. The optimum summation would be one for which the wavelet shape is the same for each trace in the gather for any arrival time. Note that the later-arrival wavelets are less affected by the stretching caused by application of the NMO correction. (NMOApplication.nb \Rightarrow NMO2.cdr \Rightarrow NMO2noDip.wmf)

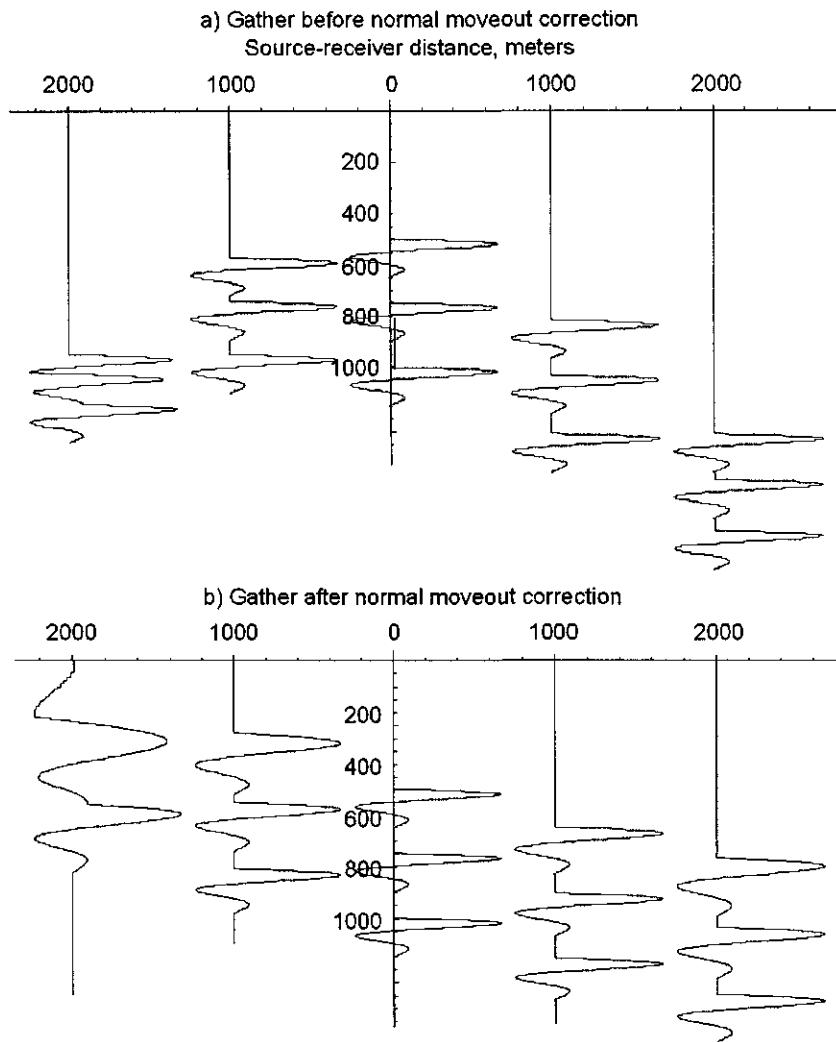


Figure 6.27: (a) Synthetic 5-trace gather acquired over a reflector dipping 20° . Recording geometry as in Figure 6.25b. After application of the same Equation (6.24) the onset reflection arrival times are not correct. The effect of the recording geometry has not been removed and wavelet stretching still occurs. (NMOApplication.nb \Rightarrow NMO2.cdr \Rightarrow NMODip.wmf)

out correction. This time the results are unsatisfactory. In the presence of dip the true rock velocity cannot be used (even if it were known) to flatten the reflection to a common offset time (Figure 6.27b). The velocities used to

stack reflection seismic data (“stacking velocities”) are commonly obtained by examining “constant-velocity panels”, which are mini-stacks of a dozen or so traces that are inspected visually to see which velocity produces the best looking (from a geological standpoint) and the most coherent (from a mathematical standpoint) result. An example was shown in Figure 6.22 on page 286.

The reader is referred to Yilmaz [203, p. 668-836], Claerbout [39, p. 160-163], and Stolt and Benson [170, This Series, Volume 5] for excellent in-depth discussions of the principles of the dip-moveout correction, a related discussion of prestack migration, and the theory and practice of seismic migration.

6.3 Refraction traveltimes curves

The seismic events of interest in refraction seismology are quite different from those of reflection, and are called *head waves*. The objectives of refraction seismology are the same, however: to determine how the velocity changes versus depth. Head waves are generated at the top of the boundary between two media, rather than being reflected *from* the boundaries. In order to record such waves, greater source-receiver offsets are generally required.

The general shapes and branches of refraction traveltimes curves, and their relation between reflection traveltimes curves, can be deduced by examining ray-paths associated with three general source-receiver offsets:

1. zero offsets
2. critical reflections
3. very long offsets

We first examine refraction traveltimes curves qualitatively to place them in context with reflection traveltimes curves. Then we derive the equations for the refraction traveltimes curves.

Assume a multilayer medium for which the velocity of each layer is constant within the layer, but the layer velocity increases with depth to the layer. The assumption of increasing layer velocity with depth is not required for reflection seismology, but is a limitation of refraction seismology, although, in general, the velocity does increase with depth in the earth.

The model is shown in Figure 6.28. Because of the increase in velocity of successively deeper layers, refraction events that travel deeper are able to arrive *ahead* of shallower refraction and reflection events that have actually traveled shorter paths; hence, refraction arrivals are commonly called *head waves*. The energy source is located at *S* in Figure 6.28. All of the seismic events to be described are *P-waves*, also called *compressional waves*, or *body waves*. The earliest events to arrive at any of the receivers on the surface follow a direct horizontal path (1) in the upper medium of velocity v_1 . They define a segment, or *branch* of a traveltimes curve whose slope is $1/v_1$, labeled as such in the figure. Representative raypaths for this and other arrivals are shown in Figure 6.28.

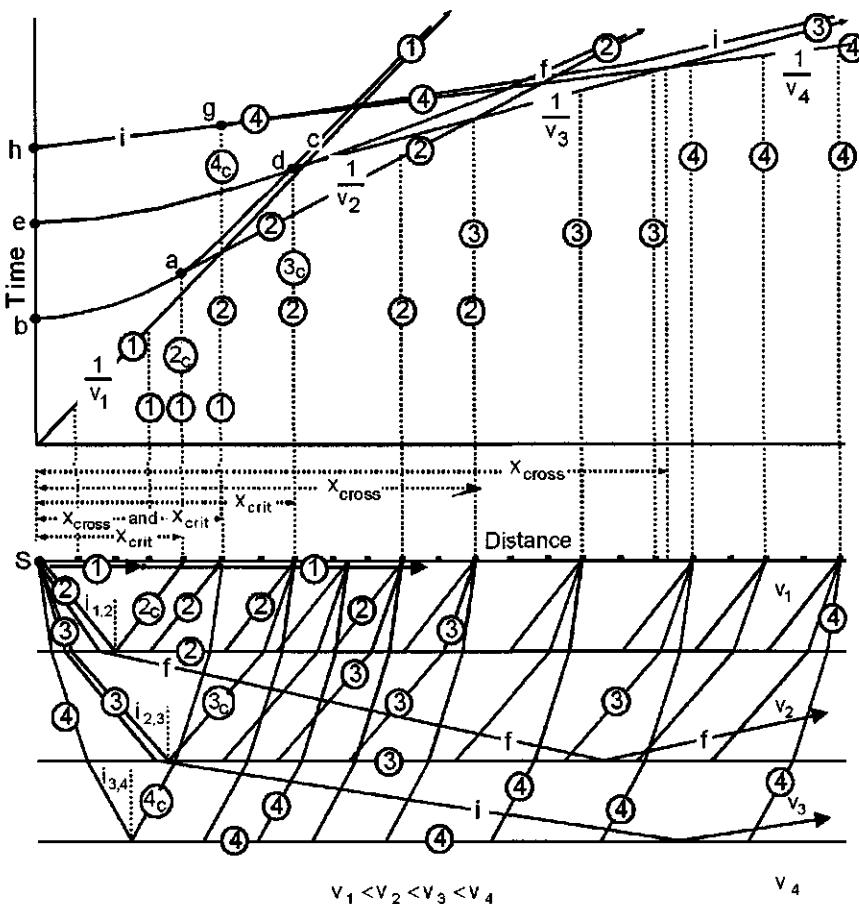


Figure 6.28: Reflection and refraction arrivals plotted on the same traveltime plot to show the relationship between the two. See text for discussion. (refr1)

At the sourcepoint S the downward-traveling wavefront eventually reaches the interface between layers 1 and 2. The particular raypath that is incident on medium 2 at the angle $i_{1,2} = \sin^{-1} v_1/v_2$ will, by Snell's law, be refracted in medium 2 such that the angle of refraction in medium 2 is 90° . Such an angle is called a *critical angle*, $i_{1,2} = i_{crit} = \sin^{-1} v_1/v_2$. This is a *critical reflection*. From this point on, a *head wave*, labeled "2" in Figure 6.28, is generated at the top of layer 2 and will travel along the top of layer 2 at the higher speed v_2 , continually radiating energy upward, back into layer 1. The slope defined by these head waves as they are recorded at the surface will be a straight line with a slope of $1/v_2$ because the receivers are deployed in the same direction as the direction of propagation of the head wave; i.e., parallel to the top of layer 2.

Since $v_2 > v_1$, the slope of this branch must be less than that for the direct waves that travel horizontally in layer 1.

How far back toward the source can refracted head waves be generated that will fall on this straight-line branch? Reference to Figure 6.28 shows that the ray that is incident at the critical angle $i_c = i_{1,2}$, will immediately return as a critical reflection, labeled “ 2_c ”, to location x_{crit} at time “a”. This distance is called a *critical distance*.

Between the critical distance x_{crit} and the crossover distance x_{cross} the direct arrivals traveling horizontally in the first layer will be recorded as *second arrivals*, as shown in Figure 6.28, until the *crossover distance* x_{cross} where head waves from the top of layer 2 arrive at the same time as the horizontally-traveling direct waves in layer 1. The distance x_{cross} is called a *crossover distance*.

Beyond x_{cross} , head waves that have traveled at the top of layer 2 are first arrivals until these are overtaken at another crossover distance x_{cross} by head waves that have been generated at the top of layer 3. Beyond this x_{cross} , because $v_3 > v_2$, the slope of the travelttime curve is therefore less than the branch defined by a slope of $1/v_2$. Beyond this crossover distance, first arrivals are head waves from the top of layer 4, and so forth.

How do the reflection traveltimes plot with respect to the refractions? One reflection from the base of layer 1 has already been identified: the raypath whose angle of incidence is critical and equal to $i_{1,2} = \sin^{-1}(v_1/v_2)$ (black dot labeled “a” in Figure 6.28). This angle is called the *critical angle* for the raypath that reflects from the interface that separates v_1 and v_2 . Another reflection raypath is a vertical one (not shown in the figure), where, clearly, it takes less time to travel a vertical path from S to the base of layer 1 than the slant path whose travelttime is “a”. The arrival time for a vertical reflection (ray parameter $p = 0$) from the base of layer 1 is therefore about at the time “b” on Figure 6.28. We now have two points on the travelttime curve for reflections from the base of layer 1.

To deduce other points on the travelttime curve, we observe that, for wide-angle reflections such as those marked “c” in Figure 6.28, the reflection paths are almost horizontal, and the slope must therefore be almost the same as that for the horizontal paths labeled “1” in the first layer, which define a slope of $1/v_1$. Reflections from the base of the first layer must therefore asymptotically approach that branch of the refraction travelttime curve whose slope is $1/v_1$. These reflection arrivals are labeled “c” in Figure 6.28, and the entire travelttime curve can now be sketched, as shown in Figure 6.28.

The travelttime curve for reflections from the base of layer 2 can be deduced in a similar manner. One point on this curve has already been defined –the time labeled “d” on Figure 6.28 where raypath “3” returns as a critical reflection (3_c). Clearly, it takes less time to reflect a vertical raypath from S (ray parameter $p = 0$) than the slant path labeled “ 3_c ”. This time is labeled “e” at $X = 0$.

To deduce other points on the travelttime curve, we observe that, for very wide-angle reflections such as those marked “f” in Figure 6.28, the reflection paths are almost horizontal in layer 2, and the slope must therefore be almost the same as that for the exactly horizontal refraction paths labeled “2” for head

waves from the top of the second layer. This slope is $1/v_2$. Reflections from the base of the second layer therefore asymptotically approach that branch of the refraction travelttime curve whose slope is $1/v_2$. These reflection arrivals are labeled "f" in Figure 6.28, and the entire travelttime curve is now defined.

Reflection traveltimes from the base of layer 3 can be deduced the same way. One point on this curve has already been defined—the point labeled "g" on Figure 6.28 where raypath "4" returns as a critical reflection (4_c). It takes less time to reflect a vertical raypath from S (ray parameter $p = 0$) than the slant path labeled " 4_c ". This time is labeled "h". (The vertical raypaths are not shown.)

To deduce other points on the travelttime curve, we observe that, for wide-angle reflections such as those marked "i" in Figure 6.28, the reflection paths are almost horizontal in layer 3, and the slope must therefore be almost the same as that for the exactly horizontal head waves labeled "3" from the top of the third layer. This slope is $1/v_3$. Reflections from the base of the third layer therefore asymptotically approach that branch of the refraction travelttime curve whose slope is $1/v_3$. These reflection arrivals are labeled "i" in Figure 6.28.

Reflection travelttime curves for reflections from deeper interfaces can be similarly deduced.

In the previous section we derived exact (parametric equations) and approximate (first several terms of a power series expansion) equations for reflection travelttime curves. We now derive exact equations for refraction traveltimes for selected geologic models.

6.3.1 Refractions from a single horizontal interface

It is possible to use a simplifying concept called "delay time" to derive equations for most refraction travelttime curves, even for dipping interfaces. Before doing this we derive refraction travelttime by a simple "brute force" summation of time spent along the various segments of the refraction path for a single refracting interface.

The model is shown in Figure 6.29. The refraction travel path is $abcd$. The travelttime T_2 of the head wave from the sourcepoint at a to a receiver at d is given by:

$$T_2 = \frac{ab}{v_1} + \frac{bc}{v_2} + \frac{cd}{v_1} \quad (6.25)$$

$$= \frac{2ab}{v_1} + \frac{bc}{v_2} \quad (6.26)$$

because $ab = cd$. From the figure, $\tan \alpha = eb/h_1$ and $\cos \alpha = h_1/ab$. Therefore

$$\begin{aligned} T_2 &= \frac{2h_1}{v_1 \cos \alpha} + \frac{X - 2h_1 \tan \alpha}{v_2} \\ &= \frac{2h_1}{v_1 \cos \alpha} + \frac{X - 2h_1 \sin \alpha / \cos \alpha}{v_2} \end{aligned}$$

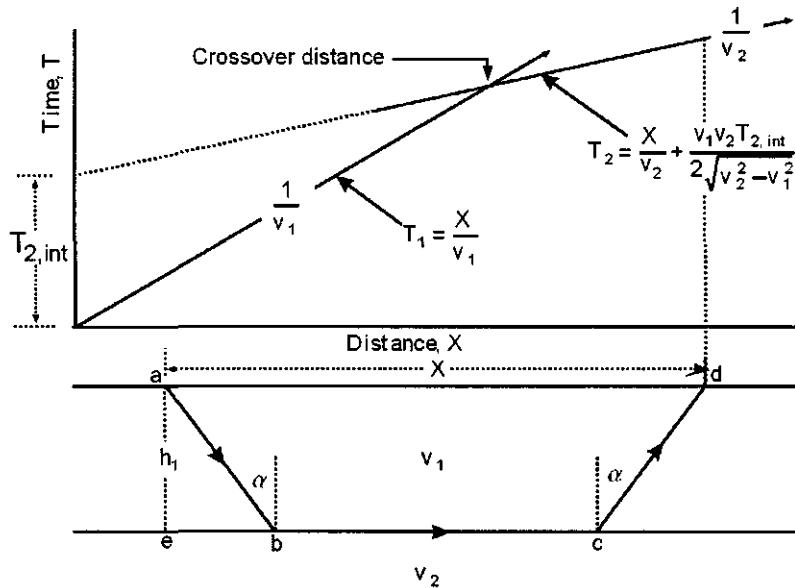


Figure 6.29: Refraction raypath geometry for a single interface. (refr2)

From Snell's law,

$$\frac{\sin \alpha}{v_1} = \frac{\sin 90^\circ}{v_2} = \frac{1}{v_2}$$

and therefore $\sin \alpha = v_1/v_2$. Substituting for v_2 :

$$\begin{aligned} T_2 &= \frac{2h_1}{v_1 \cos \alpha} + \frac{X}{v_2} - \frac{2h_1 \sin^2 \alpha}{v_1 \cos \alpha} \\ &= \frac{X}{v_2} + \frac{2h_1}{v_1 \cos \alpha} [1 - \sin^2 \alpha] \\ &= \frac{X}{v_2} + \frac{2h_1 \cos \alpha}{v_1} \end{aligned}$$

This is the equation of a straight line, and can be expressed in terms of the velocities v_1 and v_2 as

$$\begin{aligned} T_2 &= \frac{X}{v_2} + \frac{2h_1 \sqrt{1 - \sin^2 \alpha}}{v_1} \\ &= \frac{X}{v_2} + \frac{2h_1 \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \end{aligned}$$

The slope of the refraction travelttime curve is thus $1/v_2$ and the *intercept*

time $T_{2,int}$ (the time at $X = 0$), is

$$T_{2,int} = \frac{2h_1\sqrt{v_2^2 - v_1^2}}{v_1 v_2} \quad (6.27)$$

The quantities $T_{2,int}$, v_1 , and v_2 can be determined from the traveltime curve. The only unknown is therefore h_1 , and, from Equation (6.27), we have therefore determined the depth h_1 to the interface:

$$\begin{aligned} h_1 &= \frac{v_1 v_2 T_{2,int}}{2 \sqrt{v_2^2 - v_1^2}} \\ &= \frac{T_{2,int}}{2} \frac{v_1}{\cos \alpha} \end{aligned}$$

Illustrations of “first arrivals” for real data are shown in Figure 6.30 from a study done on the Atlantic Coastal Plain [52]. The objective of the study was to determine the depth to crystalline basement using refraction techniques. In the location of the field study relatively low-velocity (≈ 2) km/sec sediments comprising the Atlantic Coastal Plain overlie crystalline basement rocks (≈ 6 km/sec), so there is an excellent velocity contrast.

6.3.2 Delay time

A simpler derivation of refraction traveltimes that can also be used for multilayered systems, even with dipping interfaces, can be deduced using the concept of *delay time*.

Assume a fictitious, vertical, plane wavefront traveling in a horizontal direction and passing through the source at “a” and the receiver at “d” as shown in Figure 6.31. Make the (incorrect) assumption that the wave front traverses the entire distance X at the speed v_2 , then determine what is needed to correct this assumption; i.e.,

$$T_2 = \frac{X}{v_2} + \text{a correction term}$$

From Figure 6.31, what we have referred to as a “correction” is simply the difference between the actual time spent traversing the slant paths at the speed v_1 and the time spent along the projection of the slant paths in the direction traveled by the wavefront at the speed v_2 .

This difference, or “correction” is called *delay time* D , and for the downward slant path is D_d .

$$D_d = \text{delay time} = \frac{ab}{v_1} - \frac{eb}{v_2}$$

From Figure 6.29 and Snell’s law,

$$D_d = \frac{h_1}{v_1 \cos \alpha} - \frac{h_1 \tan \alpha}{v_2}$$

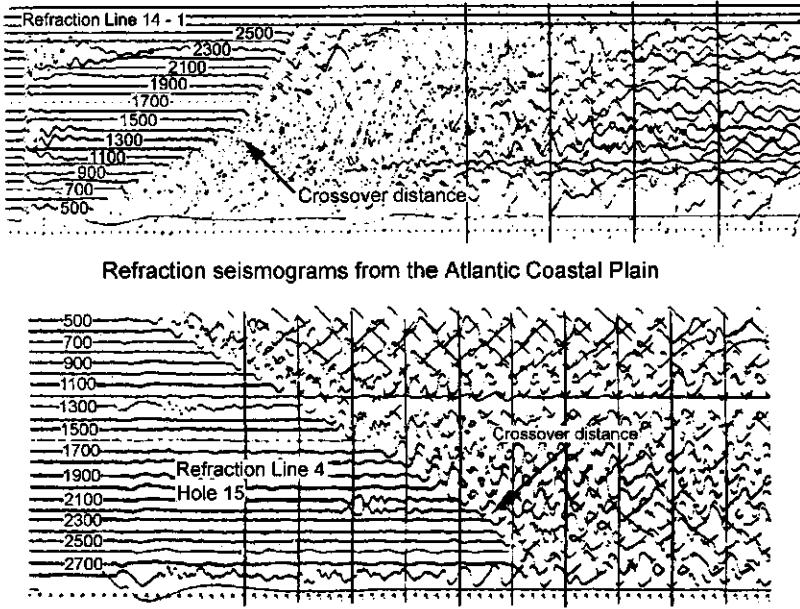


Figure 6.30: Refraction seismograms from the Atlantic Coastal Plain. Top: Site 1. Explosive charge consisted of 1 lb Nitramon primers at a depth of 20 feet. Bottom: Site 2. Explosive charge was a 6-foot length of Primacord at a depth of 20 feet. Annotations on field records indicate distances in feet from the shotpoint. From Costain and Robinson [52]. (CandR1.bmp)

$$\begin{aligned}
 &= \frac{h_1}{v_1 \cos \alpha} - \frac{h_1 \sin^2 \alpha}{v_1 \cos \alpha} \\
 &= \frac{h_1}{v_1 \cos \alpha} [1 - \sin^2 \alpha] = \frac{h_1 \cos \alpha}{v_1}
 \end{aligned}$$

Because the raypath is symmetrical in this model, the total delay time D for the down, D_d , and up, D_u , segments is twice this value:

$$\begin{aligned}
 D = D_d + D_u &= \frac{2h_1 \cos \alpha}{v_1} = \frac{2h_1 \sqrt{1 - \sin^2 \alpha}}{v_1} \\
 &= \frac{2h_1 \sqrt{v_2^2 - v_1^2}}{v_1 v_2}
 \end{aligned}$$

which is identical to the expression derived for the intercept time.

Thus, *delay time = intercept time*, and the traveltime T_2 is

$$T_2 = \frac{X}{v_2} + \text{total delay time (down and up)}$$

$$= \frac{X}{v_2} + \frac{2h_1\sqrt{v_2^2 - v_1^2}}{v_1 v_2} \quad (6.28)$$

and with reference to Figure 6.31 and Equation 6.28 the mathematical significance of the intercept time at $X = 0$ is the time spent along "ab" at the speed v_1 minus the time spent along "bc" at the speed v_2 plus the time spent along "ca" at the speed v_1 . There is no actual event that follows this path. The

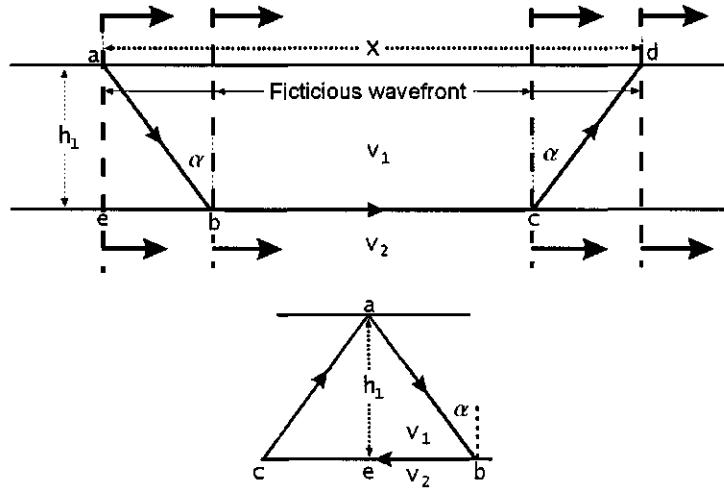


Figure 6.31: The concept of delay time. Upper diagram: Fictitious wavefront is assumed to travel from "a" to "d" at the speed v_2 . This is the " $T_2 = X/v_2$ " part of (6.28). Only the part of the path between "b" and "c" is traversed at this higher speed v_2 , however. Therefore a correction must be made to add more time. The time required is just the difference between the time actually spent along the slant path "ab" at the speed v_1 and the horizontal component ("eb") of the path traveling at the speed v_2 . This difference is called "delay time". For a horizontal refractor, the total delay time is the delay time associated with the down-going ray segment, ("ab"), plus the delay time associated with the up-going ray segment, ("cd"). There is no delay time associated with the segment "bc". Lower diagram: The mathematical significance of the intercept time at $X = 0$ is the time spent along "ab" at the speed v_1 minus the time spent along "bc" at the speed v_2 plus the time spent along "ca" at the speed v_1 . There is no actual event that follows this path "abc". For multiple layers α is not a critical angle (See Figure 6.32). (refr3.cdr, refr3a)

same definition of delay time can be applied to any segment of a raypath in a multilayered model. For example, consider the delay time associated with the down-going raypath segment in a deeper layer (Figure 6.32), say layer 3, in an n -interface model. The double subscript notation will be used, where the first number corresponds to the layer in which the segment or angle occurs, and the

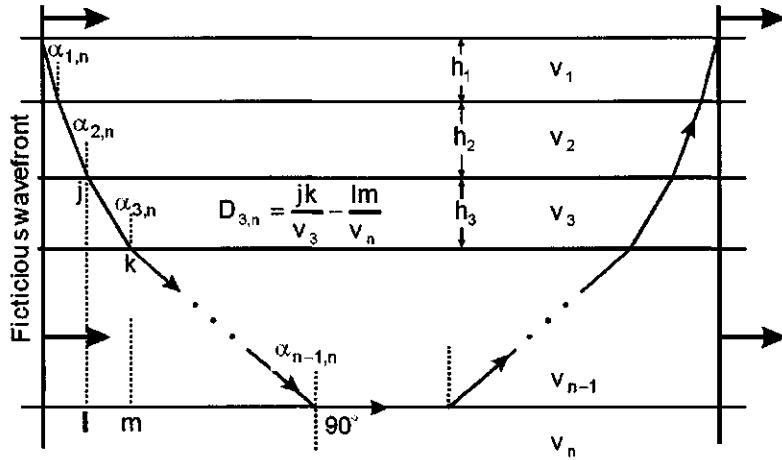


Figure 6.32: Delay time computed for a single segment in the third layer in an n -layer model. Total delay time along the entire ray path is the sum of delay times for each individual segment of the ray path. Note that the angle $\alpha_{3,n}$ is *not* a critical angle. (refr4)

second to the deepest layer encountered along the entire raypath.

Assume the head wave is generated at the top of the n^{th} layer. The delay time $D_{3,n}$ for this segment is, by definition,

$$\begin{aligned}
 D_{3,n} &= \text{delay time for down-going segment in layer 3} = \left[\frac{jk}{v_3} - \frac{lm}{v_n} \right] \\
 &= \frac{h_3}{v_3 \cos \alpha_{3,n}} - \frac{h_3 \tan \alpha_{3,n}}{v_n} \\
 &= \frac{h_3}{v_3 \cos \alpha_{3,n}} - \frac{h_3 \sin^2 \alpha_{3,n}}{v_3 \cos \alpha_{3,n}} \\
 &= \frac{h_3}{v_3 \cos \alpha_{3,n}} [1 - \sin^2 \alpha_{3,n}] \\
 &= \frac{h_3 \cos \alpha_{3,n}}{v_3}
 \end{aligned}$$

But from Snell's law:

$$\frac{\sin \alpha_{1,n}}{v_1} = \frac{\sin \alpha_{2,n}}{v_2} = \frac{\sin \alpha_{3,n}}{v_3} = \dots = \frac{\sin 90^\circ}{v_n} = \frac{1}{v_n}$$

So,

$$\frac{\sin \alpha_{3,n}}{v_3} = \frac{1}{v_n}$$

and therefore

$$D_{3,n} = \frac{h_3 \cos \alpha_{3,n}}{v_3} = \frac{h_3 \sqrt{1 - \sin^2 \alpha_{3,n}}}{v_3} = \frac{h_3 \sqrt{v_n^2 - v_3^2}}{v_3 v_n}$$

Note that the angle $\alpha_{3,n}$ in Figure 6.32 is *not* a critical angle, as it happened to be for the two-layer case.

6.3.3 General expression for delay time

The general expression for *delay time* $D_{m,n}$ for a single segment in the m^{th} layer is therefore seen to be

$$D_{m,n} = \frac{h_m \cos \alpha_{m,n}}{v_m} = \frac{h_m \sqrt{v_n^2 - v_m^2}}{v_m v_n} \quad (6.29)$$

6.3.4 Two-layer model

Using the concept of delay time and Equation (6.29) we can proceed with the derivation of refraction traveltimes from multilayered models (Figure 6.33). The traveltime T_3 from the top of the third layer can now be written

$$\begin{aligned} T_3 &= \frac{X}{v_3} + \text{total delay time (down and up)} \\ &= \frac{X}{v_3} + D_d + D_u \end{aligned}$$

where D_d is the *total* delay time associated with the down slant-path segments and D_u is the *total* delay time for the up segments.

Refer to Figure 6.32 and the definition of delay time applied to each segment of the raypath. Notation for the angles is the same as that used for reflection traveltimes: the first subscript refers to the number of the layer in which the segment of the refraction raypath is located. The second subscript is the index corresponding to the highest velocity encountered along the entire raypath.

$$D_d = \frac{h_1 \cos \alpha_{1,3}}{v_1} + \frac{h_2 \cos \alpha_{2,3}}{v_2}$$

and by symmetry

$$D_u = D_d$$

Thus,

$$T_3 = \frac{X}{v_3} + \frac{2h_1 \cos \alpha_{1,3}}{v_1} + \frac{2h_2 \cos \alpha_{2,3}}{v_2}$$

It remains to express the angles $\alpha_{m,n}$ in terms of layer velocities. From Snell's law and Figure 6.32:

$$\frac{\sin \alpha_{1,3}}{v_1} = \frac{\sin \alpha_{2,3}}{v_2} = \frac{\sin 90^\circ}{v_3} = \frac{1}{v_3}$$

Thus,

$$\frac{\sin \alpha_{1,3}}{v_1} = \frac{1}{v_3}$$

or, in general

$$\frac{\sin \alpha_{m,n}}{v_m} = \frac{1}{v_n}$$

and

$$\begin{aligned} T_3 &= \frac{X}{v_3} + 2 \sum_{m=1}^{n-1} \frac{h_m \cos \alpha_{m,n}}{v_m} \\ &= \frac{X}{v_3} + \frac{2h_1 \cos \alpha_{1,3}}{v_1} + \frac{2h_2 \cos \alpha_{2,3}}{v_2} \\ &= \frac{X}{v_3} + \frac{2h_1 \sqrt{1 - \sin^2 \alpha_{1,3}}}{v_1} + \frac{2h_2 \sqrt{1 - \sin^2 \alpha_{2,3}}}{v_2} \\ &= \frac{X}{v_3} + \frac{2h_1 \sqrt{v_3^2 - v_1^2}}{v_1 v_3} + \frac{2h_2 \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \end{aligned}$$

6.3.5 General expression for multilayer refraction traveltimes from horizontal layers

Using the concept of generalized delay time the equation for the travelttime T_n of a head wave originating from the top of the n^{th} layer is

$$\begin{aligned} T_n &= \frac{X}{v_n} + \text{total delay time (down and up)} \\ &= \frac{X}{v_n} + 2 \sum_{m=1}^{n-1} \frac{h_m \cos \alpha_{m,n}}{v_m} \\ &= \frac{X}{v_n} + 2 \sum_{m=1}^{n-1} \frac{h_m \sqrt{v_n^2 - v_m^2}}{v_m v_n} \end{aligned} \tag{6.30}$$

A *Mathematica* program `TraveltimeCurvesRefraction` to compute refraction traveltimes curves from an n -layer model based on Equation (6.30) is

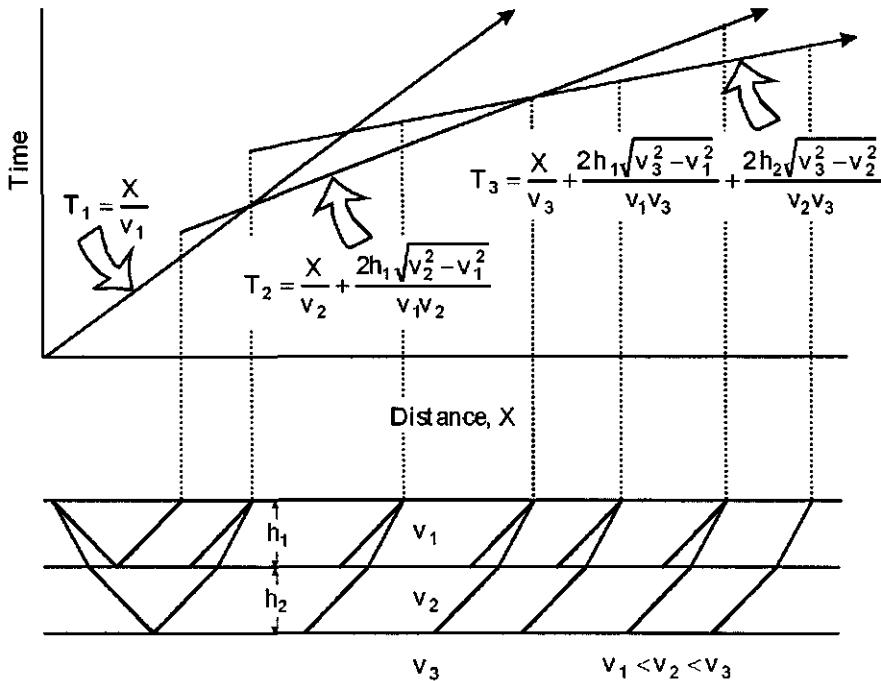


Figure 6.33: Two-interface (three layers) refraction model. Derivation of traveltime equations for head waves from the top of the second and third layers using the concept of delay time is straightforward. (refr5.cdr \Rightarrow refr5.wmf)

```
(* TraveltimeCurvesRefraction.nb *)(* Define velocities & thicknesses *)
v = {1, 2, 4, 8}; h = {1, 2, 4, 8};
NumberOfLayers = Length[h]; PlotToX = 25; PlotToT = 15;
Xmax = PlotToX; dx = .01; T = Table[0, {Ceiling[Xmax/dx]}]; X = T;
NumberOfInterfaces=Length[h]; RefractionPlots=Table[0,{NumberOfLayers}];
Do[DT = 0;
Do[DT=DT+(2h[[k]]*Sqrt[v[[interface]]^2-v[[k]]^2])/(v[[interface]]*
v[[k]]), {k, 1, interface - 1}]; j = 0;
Print["Delay time associated with layer ", interface, " is ", DT // N];
Do[j = j + 1; T[[j]] = x/v[[interface]] + DT;
X[[j]] = x, {x, 0, Xmax - dx, dx}];
RefractionPlots[[interface]] =
ListPlot[Transpose[{X, T}], AxesOrigin -> {0, 0},
DisplayFunction -> Identity, PlotRange -> {{0, PlotToX}, {0, PlotToT}},
PlotJoined -> True], {interface, 1, NumberOfInterfaces}];
Show[RefractionPlots, DisplayFunction->$DisplayFunction, ImageSize->488];
```

In this program the velocity of each layer must increase with depth.

6.3.6 Reflection and refraction traveltimes curves combined—comments

For long offsets reflected events and the refracted head waves arrive close together and overlap in time. Such “wide-angle shooting” requires non-standard data acquisition and processing techniques. Wide-angle recording configurations have long been used to explore the deep crust and the margins of continents. An example of using wide-angle reflection data to image crustal structure is provided by Kalachand and Kaila [90]. A general model illustrating the time relationship between reflected and refracted arrivals is shown in Figure 6.34.

6.3.7 Effect of dip on refraction traveltimes curves

We continue our discussion of refraction traveltimes by introducing dip into the model, which is shown in Figure 6.35. The energy source is at “a” and the receivers are deployed to the right as least as far as “d”, so the configuration is one of “shooting down-dip”. The objective of the experiment is to determine h_d , which is measured perpendicular to the interface between v_1 and v_2 .

First arrivals to receivers near the sourcepoint will travel horizontal paths in the upper medium and define a slope of $1/v_1$ as shown in Figure 6.35. At some distance it will become faster to travel the longer path down to the higher velocity medium, refract at the critical angle at the interface between v_1 and v_2 , and return to the surface as critically refracted head waves. We turn now to the derivation of the equation for head waves from the top of the lower medium. Using the concept of delay time, and again assuming a fictitious wavefront perpendicular to the interface between v_1 and v_2 , the time for the wavefront to travel the distance X can be written as

$$\begin{aligned}
 T_2 &= \frac{X \cos \theta}{v_2} + \text{delay time (down and up)} \\
 &= \frac{X \cos \theta}{v_2} + \frac{h_d \cos \alpha}{v_1} + \frac{[h_d + X \sin \theta] \cos \alpha}{v_1} \\
 &= \frac{X \cos \theta}{v_2} + \frac{h_d \cos \alpha}{v_1} + \frac{h_d \cos \alpha}{v_1} + \frac{X \sin \theta \cos \alpha}{v_1} \\
 &= \frac{X \cos \theta}{v_2} + \frac{X \sin \theta \cos \alpha}{v_1} + \frac{2 h_d \cos \alpha}{v_1} \\
 &= \frac{X \cos \theta \sin \alpha}{v_1} + \frac{X \sin \theta \cos \alpha}{v_1} + \frac{2 h_d \cos \alpha}{v_1} \\
 &= \frac{X}{v_1} [\cos \theta \sin \alpha + \sin \theta \cos \alpha] + \frac{2 h_d \cos \alpha}{v_1} \\
 &= \frac{X}{v_1} \sin(\alpha + \theta) + \frac{2 h_d \cos \alpha}{v_1} \\
 &= \frac{X \sin(\alpha + \theta)}{v_2 \sin \alpha} + \frac{2 h_d \sqrt{1 - \sin^2 \alpha}}{v_1}
 \end{aligned}$$

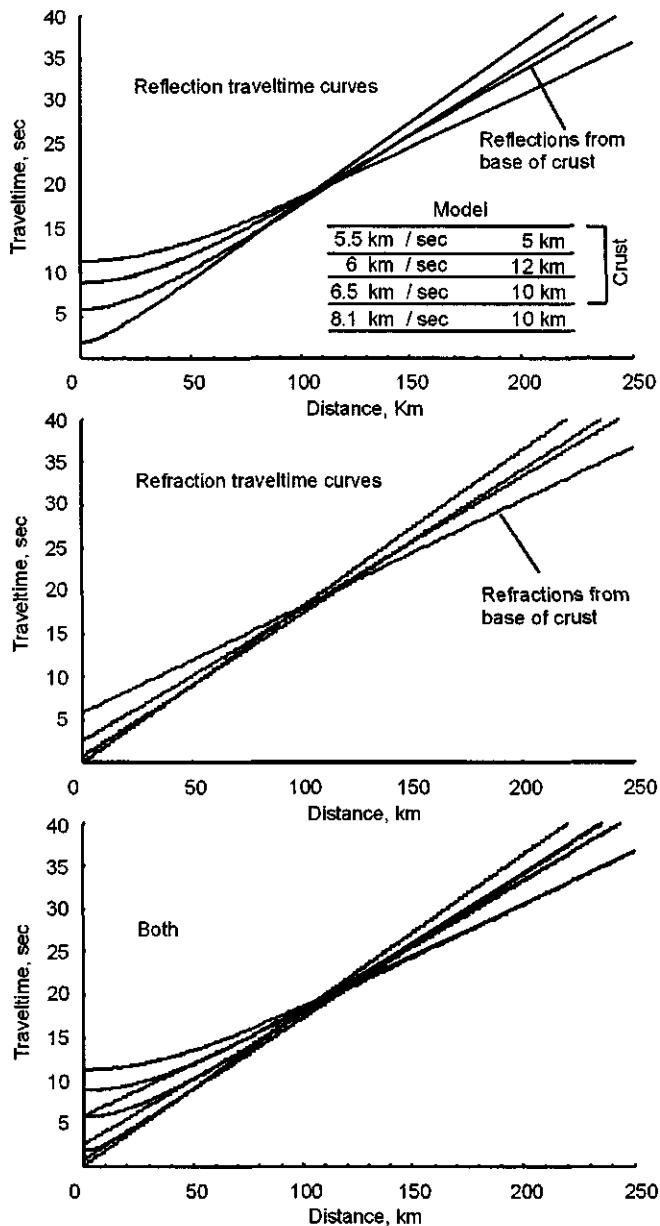


Figure 6.34: Top: Reflection traveltime curves. The velocities and layer thicknesses are shown at the top and are given in the *Mathematica* program *TraveltimeCurves.nb*. The model parameters were chosen to image the base of a thin crust at about 27 km. Middle: Refraction traveltime curves. These curves do not extend backward to $X = 0$ as shown. Refer to page 293 and Figure 6.3 where this is discussed in more detail. Bottom: Superposition of all traveltime curves. (ReflectionAndRefraction.cdr,wmf)

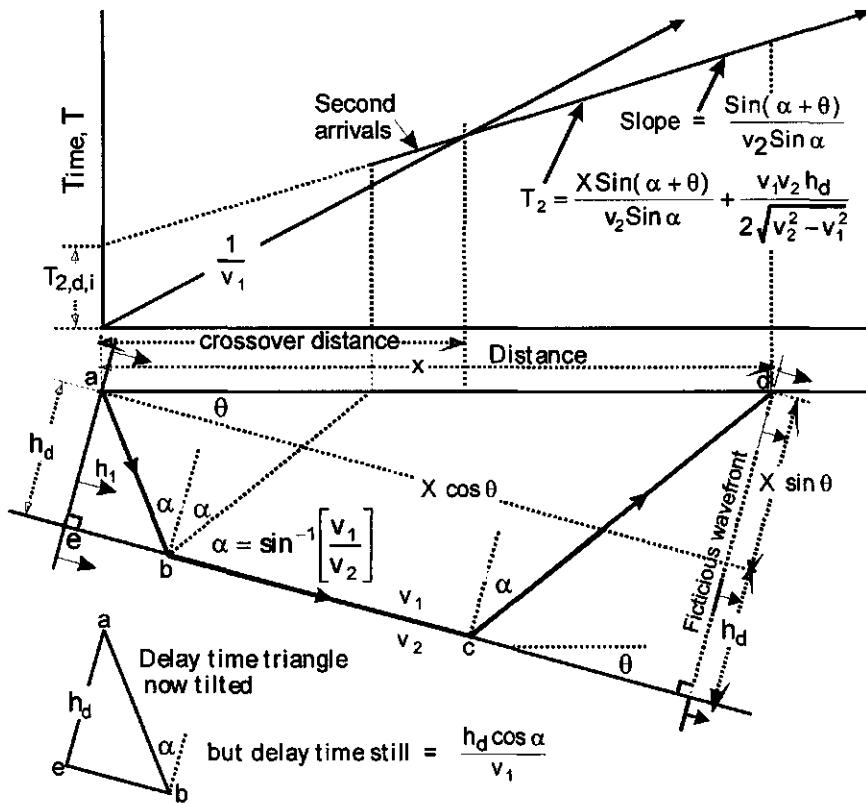


Figure 6.35: Refraction traveltimes over a single dipping interface: shooting down-dip from "a". The concept of delay time leads to an easy derivation (see text) of the traveltime curve. The slope of the refraction branch is no longer $1/v_2$. Therefore, the traveltime equation has two unknowns θ and v_2 . In order to determine h_d a second equation must be found, which is obtained by shooting a reversed (up-dip) profile, as shown in Figure 6.36. (refrdown)

$$= \frac{X \sin(\alpha + \theta)}{v_2 \sin \alpha} + \frac{2h_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \quad (6.31)$$

where we have used $\sin \alpha = v_1/v_2$ and $\sin(\alpha + \theta) = \cos \theta \sin \alpha + \sin \theta \cos \alpha$ to arrive at Equation (6.31), which is the equation of a straight line with

$$\text{slope} = \frac{\sin(\alpha + \theta)}{v_2 \sin \alpha} \quad (6.32)$$

$$\text{intercept} = \frac{2h_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \quad (6.33)$$

Note that if the angle of dip θ is zero then the slope of the refraction traveltime

curve is

$$\frac{1}{v_2}$$

as it must be for a horizontal interface.

The velocity v_1 is obtained directly from the slope of the first branch of the travelttime curve. If the dip is not zero, however, then we have two unknowns: θ and v_2 , but only one equation. The angle α is not an additional unknown because it is related to v_1 and v_2 by Snell's law: $\sin \alpha = v_1/v_2$. Without additional information, the depth h_d cannot be determined. Even more of a problem is the fact that a refraction profile over a dipping interface yields no information about whether the interface is dipping, because Equations (6.28) and (6.31) are both equations of straight lines.

A *reversed profile* must therefore always be obtained.

The required second equation comes from shooting in the opposite direction to obtain a *reversed profile*. The model geometry is shown in Figure 6.36. Again, we use the concept of delay time to obtain the travelttime T_2 when shooting updip. Using the delay time triangles and referring to Figure 6.36, we assume the fictitious wavefront travels the distance X from "a" to "d" at the speed v_2 .

$$\begin{aligned}
 T_2 &= \frac{X \cos \theta}{v_2} + \text{delay time (down and up)} \\
 &= \frac{X \cos \theta}{v_2} + \frac{h_u \cos \alpha}{v_1} + \frac{[h_u - X \sin \theta] \cos \alpha}{v_1} \\
 &= \frac{X \cos \theta \sin \alpha}{v_1} + \frac{h_u \cos \alpha}{v_1} + \frac{h_u \cos \alpha}{v_1} - \frac{X \sin \theta \cos \alpha}{v_1} \\
 &= \frac{X \cos \theta \sin \alpha}{v_1} - \frac{X \sin \theta \cos \alpha}{v_1} + \frac{2 h_u \cos \alpha}{v_1} \\
 &= \frac{X \sin(\alpha - \theta)}{v_1} + \frac{2 h_u \cos \alpha}{v_1} \\
 &= \frac{X \sin(\alpha - \theta)}{v_2 \sin \alpha} + \frac{2 h_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2}
 \end{aligned} \tag{6.34}$$

where we have used $\sin(\alpha - \theta) = \cos \theta \sin \alpha - \sin \theta \cos \alpha$ to arrive at Equation (6.34), which is the equation of a straight line with

$$\text{slope} = \frac{\sin(\alpha - \theta)}{v_2 \sin \alpha} \tag{6.35}$$

$$\text{intercept} = \frac{2 h_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \tag{6.36}$$

We now have two equations and two unknowns:

$$T_{2,d} = \frac{X \sin(\alpha + \theta)}{v_2 \sin \alpha} + \frac{2 h_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \tag{6.37}$$

$$T_{2,u} = \frac{X \sin(\alpha - \theta)}{v_2 \sin \alpha} + \frac{2 h_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \tag{6.38}$$

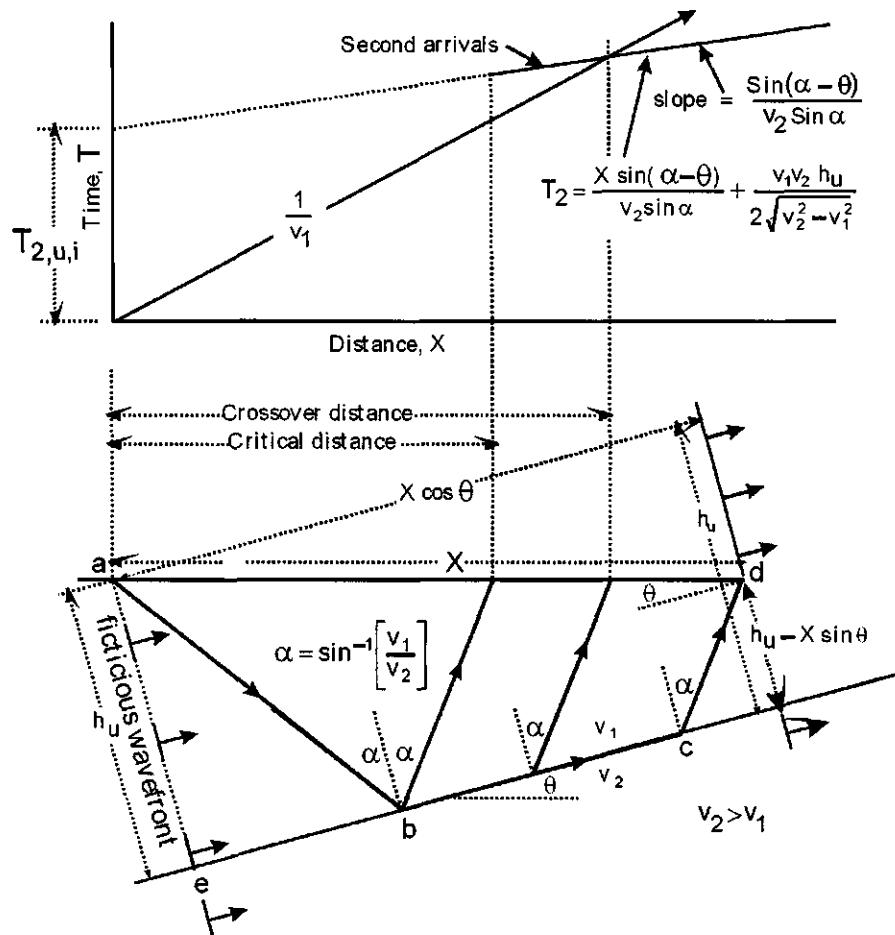


Figure 6.36: Refraction traveltimes over a single dipping interface: shooting up-dip from "a". The concept of delay time leads to an easy derivation (see text) of the travelttime curve, but the slope of the refraction branch is no longer $1/v_2$. Therefore, the travelttime equation has two unknowns θ and v_2 . In order to determine h_u a second equation must be found, which is that obtained by shooting a reversed (down-dip) profile, as shown in Figure 6.35. Note that the crossover distance is greater than that obtained when shooting down-dip. (refrup)

where $T_{2,d}$ and $T_{2,u}$ are the equations for the travelttime branches shooting down-dip and up-dip, respectively. Each is the equation of a straight line of the form

$$y = mx + b$$

The slopes m^+ and m^- of the respective traveltimes branches are

$$m^+ = \frac{\sin(\alpha + \theta)}{v_1} = \frac{\sin(\alpha + \theta)}{v_2 \sin \alpha} \quad (6.39)$$

$$m^- = \frac{\sin(\alpha - \theta)}{v_1} = \frac{\sin(\alpha - \theta)}{v_2 \sin \alpha} \quad (6.40)$$

Rearranging Equations (6.39) and (6.40), we have

$$\alpha + \theta = \sin^{-1}(v_1 m^+) \quad (6.41)$$

$$\alpha - \theta = \sin^{-1}(v_1 m^-) \quad (6.42)$$

Adding Equation (6.41) to (6.42), subtracting (6.42) from (6.41) and then rearranging, we get

$$\alpha = \frac{\sin^{-1}(v_1 m^+) + \sin^{-1}(v_1 m^-)}{2} \quad (6.43)$$

$$\theta = \frac{\sin^{-1}(v_1 m^+) - \sin^{-1}(v_1 m^-)}{2} \quad (6.44)$$

The values of m^+ , m^- , and v_1 can be obtained directly from the observed traveltimes data, and α and θ are therefore determined. From Snell's law,

$$v_2 = \frac{v_1}{\sin \alpha}$$

and from Equations (6.31) and (6.34), setting $X = 0$,

$$h_d = \frac{v_1 v_2 T_{2,d,i}}{2 \sqrt{v_2^2 - v_1^2}}$$

$$h_u = \frac{v_1 v_2 T_{2,u,i}}{2 \sqrt{v_2^2 - v_1^2}}$$

where $T_{2,d,i}$ and $T_{2,u,i}$ are the observed intercept times obtained when shooting downdip and updip, respectively, as shown in Figures (6.35) and (6.36).

The depths h_d and h_u are distances from the sourcepoints to the dipping interface as measured in a direction perpendicular to the interface. If the distances to the dipping interface directly beneath the sourcepoints are desired, these are simply

$$h_{\text{shallow end}} = \frac{h_d}{\cos \theta}$$

$$h_{\text{deep end}} = \frac{h_u}{\cos \theta}$$

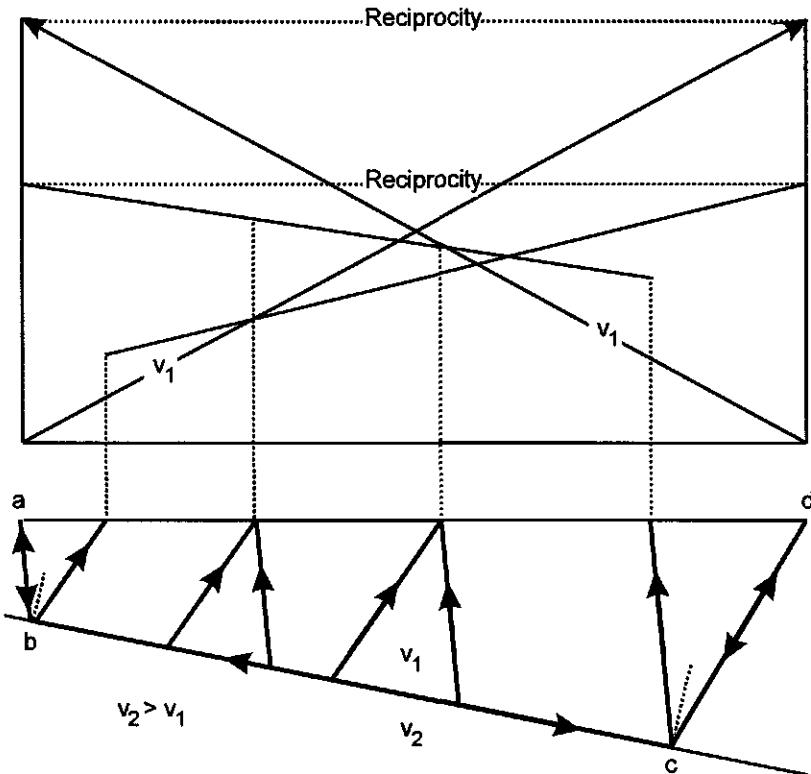


Figure 6.37: Reversed refraction traveltimes over a single dipping interface. The principle of reciprocity must be satisfied for all branches. Note that the crossover distance obtained when shooting up-dip is greater than the down-dip crossover distance. The down-dip slope is greater than the up-dip slope, consistent with Equations (6.39) and (6.40). (reverse)

6.3.8 The Principle of Reciprocity

In shooting a reversed profile, it will take the same time to travel the path $abcd$ as it does to travel the path $dcba$. These times at the sourcepoints must therefore be the same when the various branches are identified and the interpretation of the reversed refraction profile is made. As shown in Figure 6.37, this *principle of reciprocity* must be satisfied for each branch, or correct depths will not be obtained. Figure 6.37 has been duplicated in Figure 6.38, but only the first arrivals as recorded on a 24-channel data acquisition system are plotted.

The first arrivals for any branch rarely plot on a perfectly straight line. Different statics shifts beneath each receiver station, instrument problems that might result in no data at a receiver, near-surface velocity variations, and emergent rather than impulsive first arrivals all contribute to some scatter. Unlike conventional reflection data processing, the first step in the processing of re-

fraction data is an attempt to measure the emergent first motion of refraction events [131]:

The motion begins gradually and the instant when the kick can be recognized depends upon the amplification of the signal. If the amplification is low, then the event will be picked later than if the same signal were strongly amplified.

Where the same trace is displayed at several gains, the differences in arrival time can be of the order of several milliseconds [131, Figure 16-1]. See [131, This Series, Volume 13] for a discussion of the measurement of refraction arrival times and some successful approaches to this problem.

Crossover distances are therefore sometimes ambiguous. The interpreter must decide not only which first arrivals belong with which branch, but also how many branches are present. There must be one branch for each refractor in addition to well-defined first arrivals that yield a reliable value of v_1 . Additional data acquisition might be necessary to fulfill these requirements. The principle of reciprocity is a critical part of the interpretation.

Later arrivals might be distinct enough to assist where first arrivals are difficult to see (emergent), or scatter causes some ambiguity. Note, however, that as soon as the interpreter decides which first arrivals are associated with which branch, the interpretation is finished, even before any calculations are made!

The reflection study of Kolich [100] included a reversed refraction profile over the Price Mountain anticline, a structural window in Paleozoic shelf strata in southwestern Virginia. For this experiment three well-defined branches can be recognized (Figure 6.39).

6.3.9 Refraction traveltimes curves over various geologic models

Using modern computer graphics drawing programs, reversed refraction traveltimes curves can be quickly sketched over simple geologic models if the following rules of construction are followed:

- For horizontal interfaces, slopes associated with the same velocity are carefully constructed equal,
- The angles of incidence of critical angle reflections and head waves are accurately drawn,
- Reciprocity is rigorously obeyed.

A vertical fault or geologic contact that extends to the surface is shown in Figure 6.40. No head waves are generated; however, reciprocity between the horizontally traveling compressional waves must be satisfied.

Using the above rules, a reversed refraction profile over a fault of "large" throw [124, p. 272] is shown in Figure 6.41. Over the upthrown block and

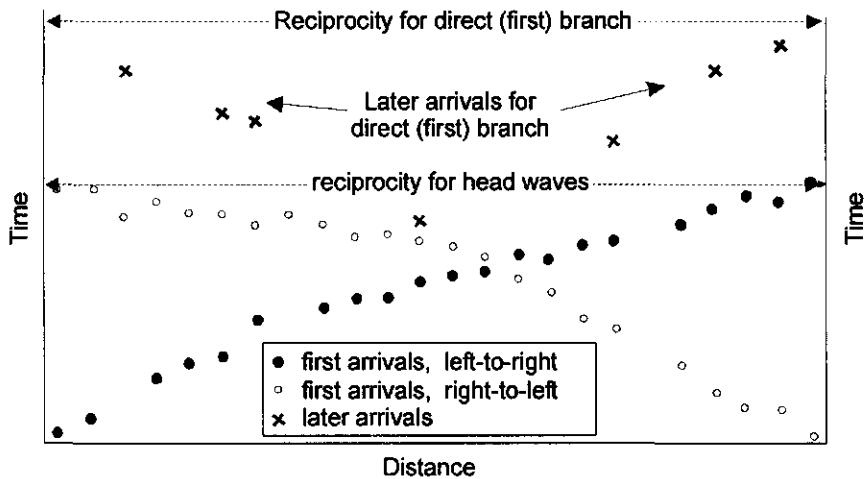


Figure 6.38: The first arrivals used to construct the reversed refraction profile shown in Figure 6.37. In general, the first arrival times will not fall on perfectly straight lines because of near-surface velocity irregularities. Refraction data should be corrected to a datum. The branches that are subsequently fitted by the interpreter to the datum-corrected first arrivals must satisfy reciprocity before depth computations are begun. For a simple two-layer case, the assignment of first arrivals to specific branches can be relatively straightforward. (reverse2.cdr \Rightarrow reverse2.wmf)

shooting from left to right the second (refraction arrivals) branch of the travelttime curve with slope equal to $1/v_2$ is linear to a distance X_n . Beyond this distance the slope of the first arrival branch gradually increases and asymptotically reaches a value of $1/v_1$. The time required to reach the part of the curve at a distance x that is asymptotic to $1/v_1$ can be quickly determined using the concept of delay time. It is

$$\begin{aligned} T &= \frac{S}{v_2} + \text{delay time down} + \text{delay time up} \\ &= \frac{S}{v_2} + \frac{Z \cos i}{v_1} + \frac{\sqrt{(X_n - S)^2 + Z^2}}{v_1} \end{aligned}$$

The linear portion of $1/v_2$ ceases at the distance X_n . If this point can be determined from the data then the distance S to the fault can be determined. It is

$$S = X_n - Z \tan i$$

Expanding the recording configuration away from the fault shown in Figure 6.41 we can deduce the reversed refraction profile that would be obtained if head waves are received from the downthrown block as well as from the upthrown. The result is shown in Figure 6.42. Shooting from left to right head

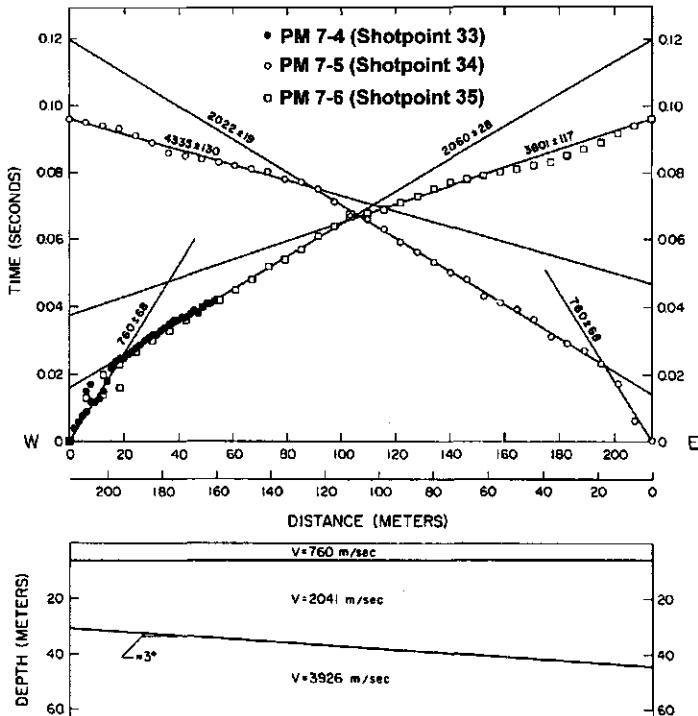


Figure 6.39: Reversed refraction profile over the Price Mountain anticline in southwestern Virginia. Note that the interpreter must choose branches that satisfy reciprocity *before* depth computations are made. Here three branches can be clearly recognized. This is a shallow refraction experiment over the same thick Paleozoic section for which velocities and densities are listed in Table 5.1 on page 203 and from which deep reflections were recovered as shown in Figure 7.3. From Kolich [100].

waves are now recorded at the surface from the downthrown block. The refracted branch of slope $1/v_2$ is now offset upward by an amount Δt . Shooting from right to left the crossover distance is greater than that observed from left to right because the downthrown block is deeper—the delay time is greater on the downthrown side. In addition, the slope of the refracted branch will approach $1/v_2$ instead of being equal to that value. Note that reciprocity must be obeyed even in the presence of faulting. Far from the fault the angle θ shown on the figure becomes small and if it is assumed to be almost zero then the throw ΔZ of the fault can be determined using the concept of delay time. Shooting from left to right the traveltime T_d to a receiver X_d on the downthrown

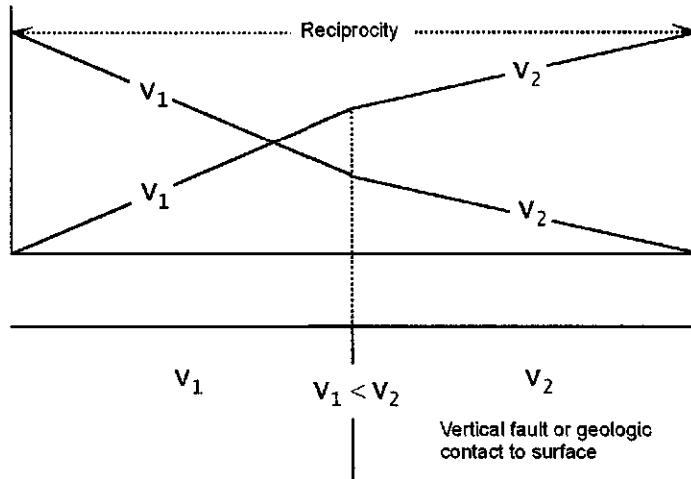


Figure 6.40: Reversed refraction profiles over a vertical fault or lithologic contact that extends to the surface. In this case there are no head waves generated, just direct compressional, P-waves. Reciprocity must still be satisfied. (refmod2)

side is

$$\begin{aligned}
 T_d &= \frac{X_d}{v_2} + \text{delaytime down} + \text{delay time up} \\
 &= \frac{X_d}{v_2} + \frac{Z_u \cos i}{v_1} + \frac{Z_d \cos i}{v_1} \\
 &= \frac{X_d}{v_2} + \frac{Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} + \frac{Z_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2}
 \end{aligned}$$

Shooting from left to right the traveltime T_u to a receiver X_u on the upthrown side is

$$\begin{aligned}
 T_u &= \frac{X_u}{v_2} + \text{delaytime down} + \text{delay time up} \\
 &= \frac{X_u}{v_2} + \frac{Z_u \cos i}{v_1} + \frac{Z_u \cos i}{v_1} \\
 &= \frac{X_u}{v_2} + \frac{2 Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2}
 \end{aligned}$$

The intercept times $T_{i,d}$ and $T_{i,u}$ (for $X = 0$) are

$$\begin{aligned}
 T_{i,d} &= \frac{Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} + \frac{Z_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \\
 T_{i,u} &= \frac{2 Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2}
 \end{aligned}$$

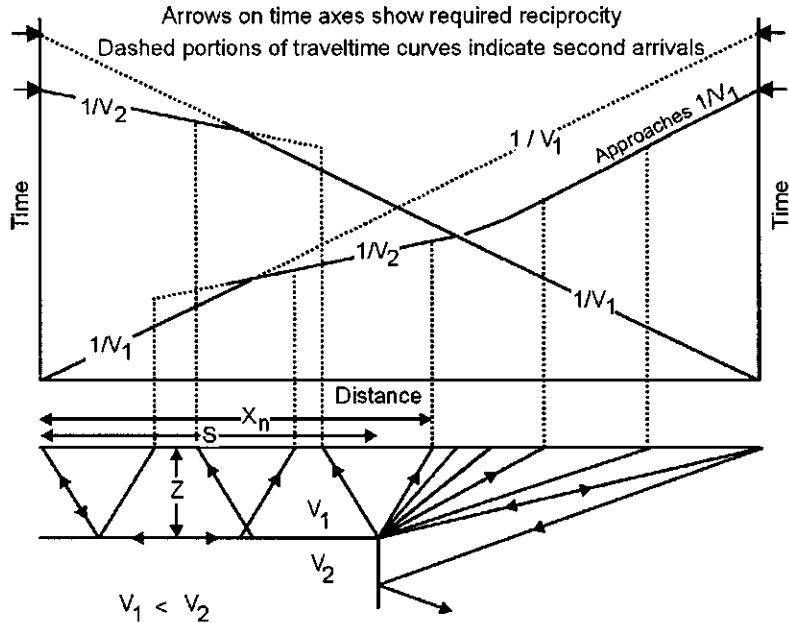


Figure 6.41: Reversed refraction profile over a fault of large throw. Alternatively, this could be a profile obtained close to the fault where no arrivals are obtained from the downthrown block. Compare with Figure 6.42. TraveltimesCurves.cdr —> RefractionLargeThrow.wmf

and the time difference is

$$\begin{aligned} T_{i,d} - T_{i,u} &= \Delta t = \frac{Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} + \frac{Z_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2} - \frac{2 Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \\ &= \frac{Z_d \sqrt{v_2^2 - v_1^2}}{v_1 v_2} - \frac{Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} = \frac{Z_d - Z_u \sqrt{v_2^2 - v_1^2}}{v_1 v_2} \end{aligned}$$

The throw $Z_d - Z_u = \Delta Z$ is therefore

$$\Delta Z = \frac{v_1 v_2 \Delta t}{\sqrt{v_2^2 - v_1^2}}$$

The ambiguity that can result from the interpretation of refraction data collected over a low-velocity layer is illustrated in Figure 6.43. Although head waves are generated at the top of the high-velocity layer because $v_2 > v_1$, they are not generated at the bottom of the high velocity layer because v_2 is also $> v_3$. If a layer of low velocity is the objective of a survey then an alternate geophysical method such as reflection should be considered as noted by Palmer [131] in Volume 13 of this Series. Undetected low-velocity layers and velocity inversions

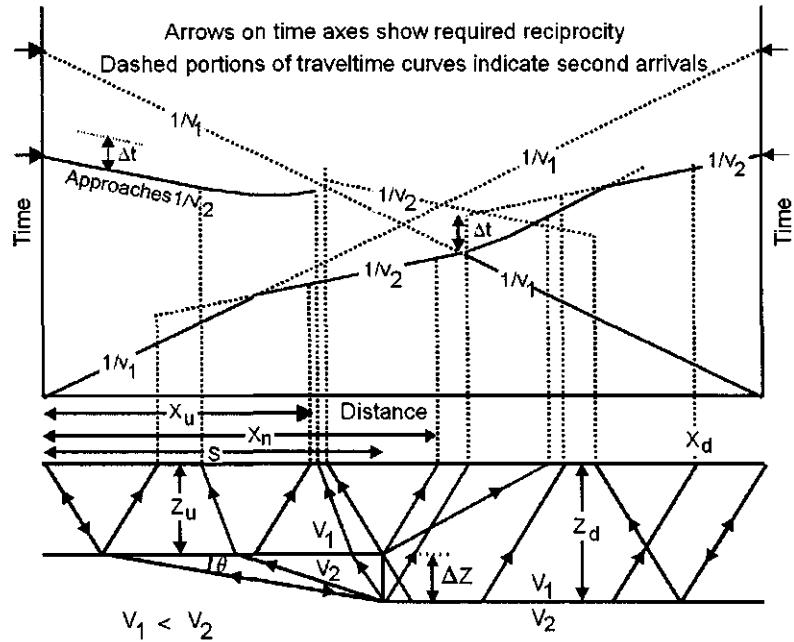


Figure 6.42: Reversed refraction profile over a fault of small throw. Compare with Figure 6.41. Shooting from left to right first arrivals are now recorded from the downthrown block. RefractionLargeThrow.cdr → Refraction-SmallThrow.wmf

highlight the fundamental shortcomings of the refraction method [131, p. 143]. The reader is referred to [131, Chapter 9] and references therein for further discussion of the “blind zone” problem.

All refraction (and reflection) data must be corrected to a datum. Other considerations are discussed by Palmer [131].

6.3.10 Dipping plane interfaces

The concept of delay time allows us to generalize the computation of refraction traveltimes from plane interfaces of arbitrary dip. Here we consider only plane interfaces. The interpretation of refraction data over irregular interfaces is discussed by Palmer [131] in Volume 13 of this Series.

Model geometry is shown in Figure 6.44 from Adachi [2] from which

$$\begin{aligned}\alpha_1 &= a_1 + \theta_2 \\ \beta_1 &= b_1 - \theta_2 \\ \alpha_i &= a_i + \theta_{i+1} \\ \beta_i &= b_i - \theta_{i+1}\end{aligned}$$

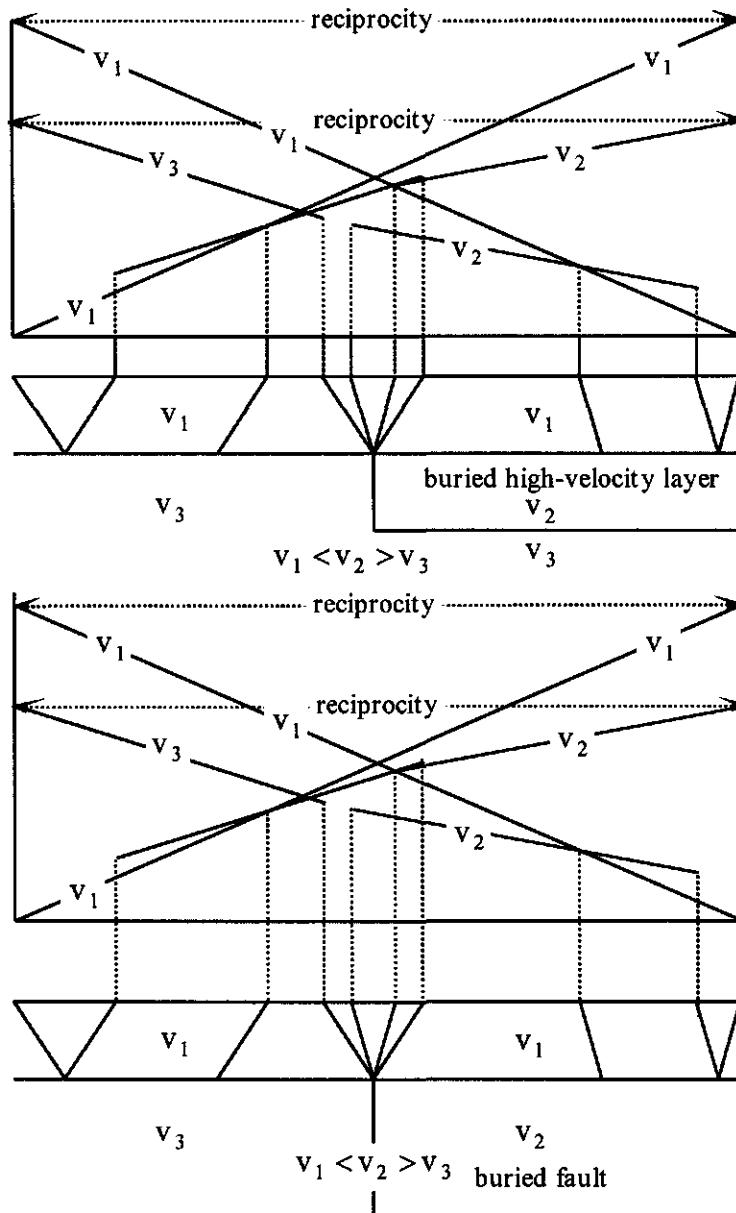


Figure 6.43: Reversed refraction profiles over a buried high-velocity layer (top) and a buried vertical fault (bottom). No head waves are generated from the bottom of the buried high-velocity layer because $v_2 > v_3$. The high-velocity layer is invisible to refraction, but would be revealed by reflection data. Although the geology is quite different, the reversed refraction profiles over these two geologic models are identical. (refmod1)

where the angles θ_i are with respect to the horizontal.

The fundamental formula given by Adachi is:

$$T_{A_n} = \frac{\sin \beta_1}{v_1} X + \sum_{i=1}^{n-1} \frac{h_{A_i}}{v_i} (\cos \alpha_i + \cos \beta_i)$$

where α and β are the angles between the raypath and the normal to an interface for the downgoing and upgoing rays, respectively. To prove this general equation

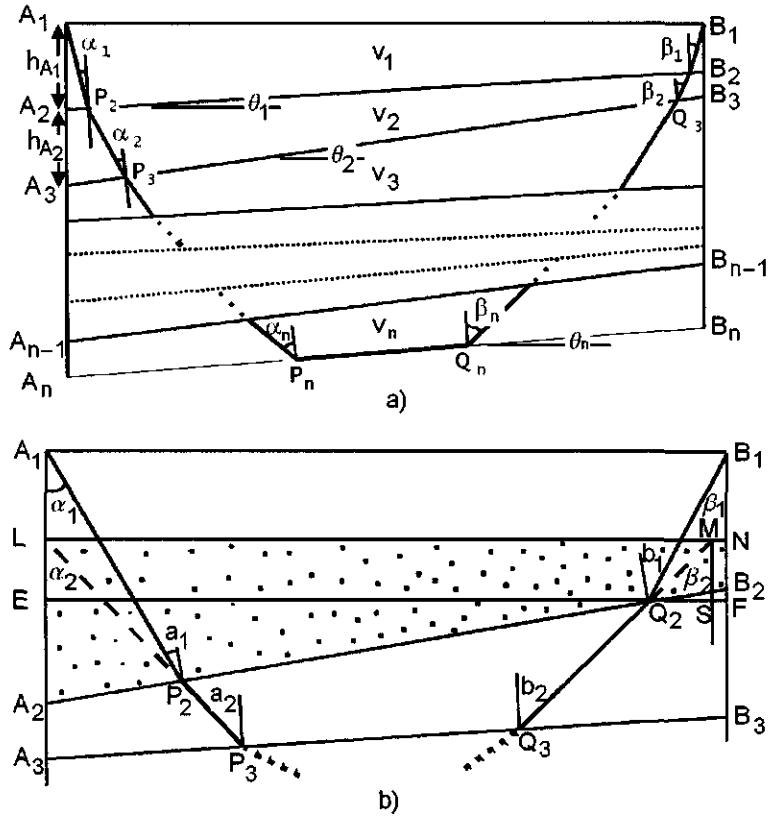


Figure 6.44: Top: a) Notation for the derivation of refraction traveltime curves over plane, dipping interfaces. Layer thicknesses h_i are measured in a vertical direction directly beneath the sourcepoints at A and B . Bottom: b) Replace layer 2 velocity with velocity of layer 1. Diagrams after Adachi [2]. (Adachi.cdr → adachild.wmf)

by induction, when $n = 2$ get:

$$T_{A_2} = \frac{\sin \beta_1}{v_1} X + \frac{h_{A_1}}{v_1} (\cos \alpha_1 + \cos \beta_1)$$

Consider the ray path $L P_2 P_3 \dots Q_3 Q_2 M$ as shown in Figure 6.44b. Denote the time spent along this path by T'_{A_n} . Assume the general equation to be true when $n = n - 1$. Then

$$T'_{A_n} = \frac{\sin \beta_2}{v_2} X' + \frac{h_{A_2} + L A_2}{v_2} (\cos \alpha_2 + \cos \beta_2) + \sum_{i=3}^{n-1} \frac{h_{A_i}}{v_i} (\cos \alpha_i + \cos \beta_i)$$

where $X' = LM$. We have stripped away part of v_1 and moved the datum plane down to LN .

And now:

$$T_{A_n} = T'_{A_n} + \frac{AP_2 + BQ_2}{v_1} - \frac{LP_2 + MQ_2}{v_2}$$

Let

$$T' = \sum_{i=2}^{n-1} \frac{h_{A_i}}{v_i} (\cos \alpha_i + \cos \beta_i)$$

T'_{A_n} can be rearranged as:

$$\begin{aligned} T'_{A_n} &= \frac{LA_2(\cos \alpha_2 + \cos \beta_2)}{v_2} + \sum_{i=2}^{n-1} \frac{h_{A_i}}{v_i} (\cos \alpha_i + \cos \beta_i) + \frac{\sin \beta_2}{v_2} X' \\ &= \frac{LA_2(\cos \alpha_2 + \cos \beta_2)}{v_2} + T' + \frac{\sin \beta_2}{v_2} X' \end{aligned}$$

so that

$$\begin{aligned} T_{A_n} &= \frac{LA_2(\cos \alpha_2 + \cos \beta_2)}{v_2} \\ &\quad + T' + \frac{\sin \beta_2}{v_2} X' + \frac{AP_2 + BQ_2}{v_1} - \frac{LP_2 + MQ_2}{v_2} \end{aligned}$$

Let

$$\Delta = AP_2 + BQ_2 + \frac{v_1}{v_2} [LA_2(\cos \alpha_2 + \cos \beta_2) + X' \sin \beta_2 - LP_2 - MQ_2]$$

Then

$$T_{A_n} = T' + \frac{\Delta}{v_1} \tag{6.45}$$

It can be shown that (left to the reader):

$$MN = Q_2 G - Q_2 S = \frac{\cos \theta_2 \sin \beta_1}{\cos b_1} h_{A_1} - \frac{\sin \theta_2 \sin \beta_1}{\cos b_1} X - MS \tan \beta_2$$

and

$$X' = \left(\frac{\sin \theta_2 \sin \beta_1}{\cos b_1} + 1 \right) X - \frac{\cos \theta_2 \sin \beta_1}{\cos b_1} h_{A_1} + MS \tan \beta_2$$

Therefore

$$\begin{aligned}\Delta &= \frac{\cos \theta_2}{\cos \alpha_1} h_{A_1} + \frac{\cos \theta_2}{\cos b_1} h_{A_1} - \frac{\sin \theta_2}{\cos b_1} X \\ &+ \frac{v_1}{v_2} \left[\frac{\cos(\alpha_2 - \theta_2) \sin \alpha_1}{\sin \alpha_2 \cos \alpha_1} (\cos \alpha_2 + \cos \beta_2) h_{A_1} + \sin \beta_2 \{X'\} - \frac{\cos \theta_2 \sin \alpha_1}{\sin \alpha_2 \cos \alpha_1} h_{A_1} - \frac{MS}{\cos \beta_2} \right]\end{aligned}$$

From Snell's law:

$$\frac{\sin \alpha_1}{v_1} = \frac{\sin(\alpha_2 - \theta_2)}{v_2}$$

$$\frac{v_1}{v_2} = \frac{\sin \alpha_1}{\sin(\alpha_2 - \theta_2)} = \frac{\sin b_1}{\sin(\beta_2 + \theta_2)}$$

Substituting this get:

$$\Delta = h_{A_1} (\cos \alpha_1 + \cos \beta_1) + X \sin \beta_1$$

Since, from above:

$$T_{A_n} = T' + \frac{\Delta}{v_1}$$

Then

$$T_{A_n} = \frac{\sin \beta_1}{v_1} X + \sum_{i=1}^{n-1} \frac{h_{A_i}}{v_i} (\cos \alpha_i + \cos \beta_i)$$

which indicates that the general equation is true for $n = n - 1$, therefore by mathematical induction the equation is generally true.

Adachi's formulation is easily programmed for use on a digital computer.

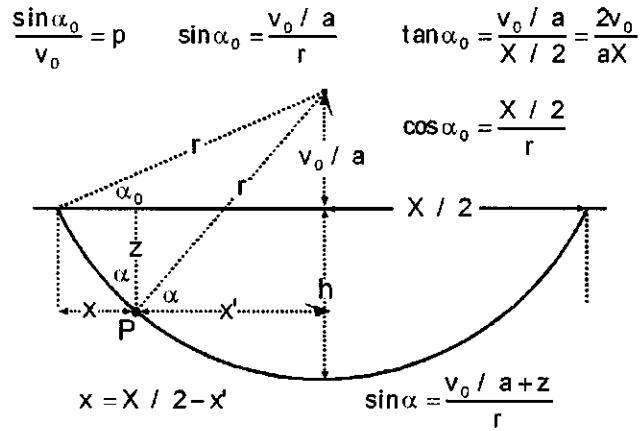


Figure 6.45: Raypath geometry for linear change of velocity with depth $v = v_0 + az$. Equations summarize procedure for determining (x, z) coordinates of raypath. (linear.ma → cd → linear1)

6.3.11 Linear increase in velocity with depth

The velocity distribution is described by $v(z) = v_0 + az$ where v_0 is the velocity at the surface ($z = 0$) and a is a constant with dimensions of, for example, km/sec per km of depth. A velocity gradient of 0.6 km/sec per km is typical of Gulf Coast sediments (Hale [83]). The general shape of the refraction raypath is shown in Figure 6.45.

In Figure 6.45 P is a point at a general depth z where the angle of incidence between the raypath and the vertical is i , for which h is the maximum depth of penetration, and whose ray parameter is p . At the maximum depth h , $\sin \alpha = 90^\circ$ and the raypath starts back up to return to the surface at the distance X . From Snell's law, at the maximum depth of penetration where the velocity v is v_m ,

$$\frac{\sin \alpha}{v(z)} = \frac{\sin 90^\circ}{v_m} = \frac{1}{v_m} = p$$

from which

$$\sin \alpha = \frac{v(z)}{v_m} = \frac{v_0 + az}{v_m} = \frac{v_0/a + z}{v_m/a} = p v(z)$$

The raypath is a segment of a circle of radius r where

$$r = \frac{v_m}{a}$$

with its center on the horizontal line a distance v_0/a above the surface. From

Figure 6.45,

$$r = \sqrt{\left(\frac{X}{2}\right)^2 + \left(\frac{v_0}{a}\right)^2}$$

The maximum depth of penetration h of the raypath is

$$h = r - \frac{v_0}{a} = \sqrt{\left(\frac{X}{2}\right)^2 + \left(\frac{v_0}{a}\right)^2} - \frac{v_0}{a}$$

The velocity at the bottom of the path is therefore

$$\begin{aligned} v_m &= v_0 + a h \\ &= v_0 + a \left[\left[\left(\frac{X}{2} \right)^2 + \left(\frac{v_0}{a} \right)^2 \right]^{1/2} - \frac{v_0}{a} \right] \\ &= \sqrt{\frac{a^2 X^2}{4} + v_0^2} \end{aligned}$$

The traveltime T is

$$\begin{aligned} T &= 2 \int \frac{dz}{v \cos \alpha} \\ &= 2 \int \frac{dz}{(v_0 + a z) \sqrt{1 - \left[\frac{(v_0 + a z)^2}{v_m^2} \right]}} \\ &= \frac{2}{a} \cosh^{-1} \left[\frac{v_m}{v_0} \right] \end{aligned}$$

because

$$\int \frac{dy}{y \sqrt{1 - (y^2/c^2)}} = -\cosh^{-1} \left| \frac{c}{y} \right|$$

Substituting

$$\frac{v_m}{v_0} = \sqrt{\left(\frac{aX}{2v_0} \right)^2 + 1}$$

and

$$\begin{aligned} T &= \frac{2}{a} \cosh^{-1} \sqrt{\left(\frac{aX}{2v_0} \right)^2 + 1} \\ &= \frac{2}{a} \sinh^{-1} \left[\frac{aX}{2v_0} \right] \end{aligned}$$

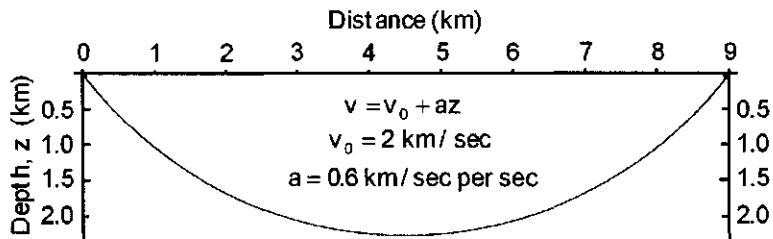


Figure 6.46: Typical refraction raypath for $v = v_0 + az$ for $v_0 = 2 \text{ km/sec}$ and $a = 0.6$. Large offsets are required for deep penetration. (linear.ma → cd → linear2)

The traveltime to a receiver at a distance X is

$$T = \frac{1}{a} \cosh^{-1} \left[\frac{a^2 (X^2 + z^2)}{2v_0 (v_0 + az)} + 1 \right]$$

for $a = 0.6 \text{ km/sec per sec}$ and $v_0 = 2 \text{ km/sec}$. Large offsets are required for deep penetration, as shown in Figure 6.47.

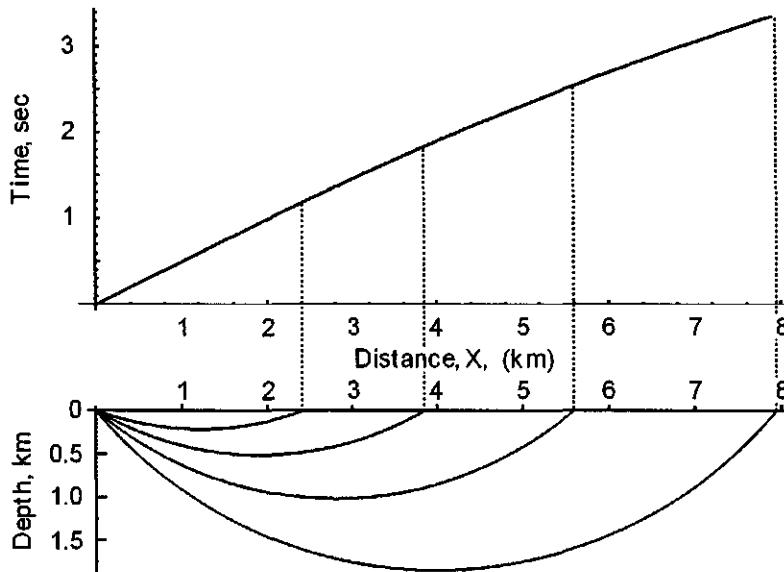


Figure 6.47: Raypaths for several different offsets X , and the traveltime curve for a medium in which $v = v_0 + az$, where $a = 0.6$ and $v_0 = 2 \text{ km/sec}$. Large receiver offsets are required for deep penetration. (pmovaref.ma → cd → pmovaref)

6.3.12 Turning waves

In Figure 6.48, the increase in velocity with depth has resulted in the seismic wave traveling downward and then upward before (and after) reflection from beneath an overhanging salt dome. This type of wave is called a *turning wave*, and is simply a continuous refraction (until reflected back). Recognition of these wave types and careful data processing have resulted in spectacular improvements in the imaging of the undersides of salt domes and intrusions of salt into vertical faults that radiate from domes (Hale [83]). An example is shown in Figure 6.48.

6.4 Composite Refraction-Reflection Stacks

Conventional processing techniques for reflection seismic data are not discussed in this volume. This subject has been covered extensively by Yilmaz [204] and many others, with more advanced applications in Volume 20 of this Handbook of Geophysical Exploration, Section I, Seismic Exploration [5]. Conventional processing techniques yield a “seismic record section”. The use of refraction data that are derived from unprocessed reflection data and then stacked in such a way that they can be merged with the reflection data is new, however, and this latter procedure is reviewed in this section.

Seismic data sets are generally processed to interpret subsurface features using either refracted or reflected arrivals, but not both. For example, in processing seismic reflection data, refracted arrivals are commonly discarded and considered to be noise during the early stages of the processing reflection data. Çoruh et al. [46] described a seismic data processing scheme that utilizes both reflected and refracted arrivals simultaneously in the common-midpoint gathers (collections of common midpoint data) to image shallow subsurface geologic features. Laughlin [106], Sen [159], Çoruh et al. [46, 47] used composite reflection-refracted stack sections to interpret shallower structures. The additional information obtained using the refracted arrivals together with the reflected arrivals allowed extended interpretations for shallower structures that cannot be imaged by the reflected arrivals alone.

In parallel with conventional reflection stacks, refraction stack sections can be generated using split-spread and/or off-end data after applying a linear-moveout correction that adjusts the refraction arrival times into modified intercept (delay) times that correspond to the related reflection times. Velocities necessary for the adjustment are obtained from constant-velocity display panels that are generated on the basis of the linear-moveout correction for the refracted first and later arrivals. The velocity obtained from the test panels is the actual velocity of a sub-horizontal refractor in which the critically refracted waves travel. In the case of a dipping refractor, the velocity obtained is an apparent velocity.

Refraction stacks produce extremely good refracted signals representing horizons that can be correlated with corresponding reflections and/or synthetic seis-

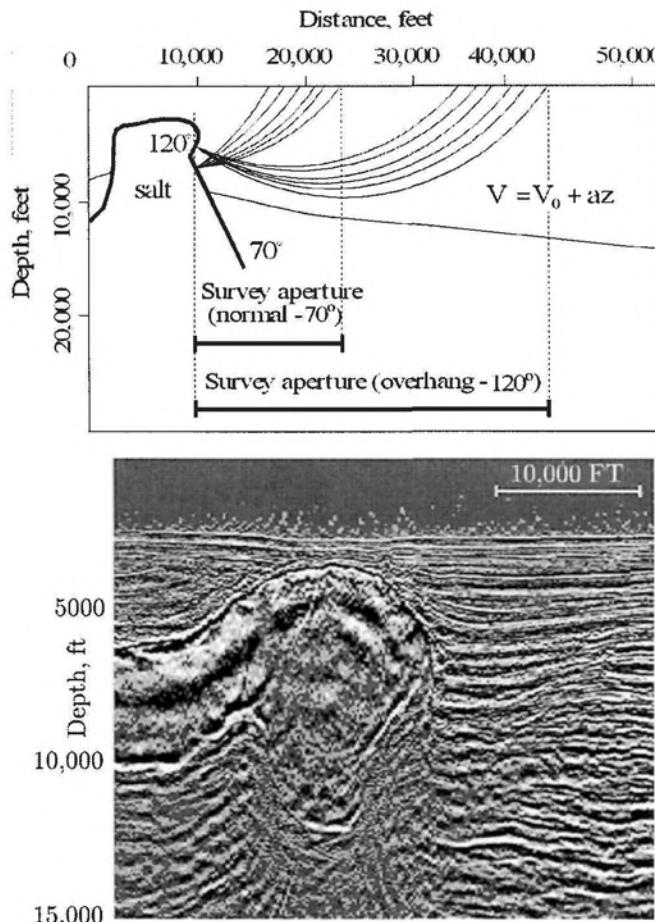


Figure 6.48: CMP (common midpoint) turning rays for normal reflection and for reflections from the salt overhang. Top: general schematic after Zhu [205]. Shown are the survey apertures and raypaths associated with one cmp on the overhang (120°) and one cmp on the steep flank of the salt dome (70°). Note that a much longer offset (larger aperture) is required to image the salt overhang as well as the normal steep dips with the same sourcepoint, but that the overhang can indeed be well imaged. Bottom: A similar overhang geometry imaged in a seismic section that incorporated turning ray processing (Abriel and Wright [1]). (cd → turnex1; ps → hj → seg)

mograms from well log data. This good correlation forms the justification for constructing composite refraction-reflection stack sections that combine the re-

flection and refraction arrivals in common midpoint gathers. The procedure minimizes the loss of seismic data and yields composite refraction—reflection stack sections that image shallow subsurface features.

Composite reflection-refraction stack sections can be used to image shallow structures for different purposes. Examples shown here were used to determine the upward penetration of basement faults into the Atlantic Coastal Plain sediments at South Carolina, near the Savannah River. The reflection stack sections successfully imaged faults in the sediments that extend upward from the crystalline basement. The faults show reverse motion and are interpreted to be reactivated faults of Paleozoic age. In addition to the reverse movement, drape (convex upward) folding in the Tertiary sediments is clearly imaged on the upthrown block close to the fault zone and is commonly associated with some of the deeper faults that penetrate upward into the overlying sediments. Displacement along faults as imaged in the reflection stack sections decreases rapidly upward from the basement and indicate growth faulting. The composite refraction-reflection stack sections show that the depth of upward penetration of the faults varies; most of them are associated with deformation at times as small as 50 ms two-way time (about 25 m).

The approach introduced by Coruh et al. [46] uses refracted arrivals in a stacking process that combines the results with conventional reflection seismic stack sections to allow better imaging of shallow subsurface structures. The traces that combine refraction and reflection arrivals make up the composite stack section. In conventional processing of multifold reflection data the refracted arrivals are generally suppressed during and/or before stacking. The use of split-spread data to minimize the effects of a dipping refractor was also introduced. Here we discuss the effects of dipping refractors and the use of refracted arrivals recorded using split- and off-end recording configurations.

The stacking process for refracted arrivals is similar to the conventional stacking process of reflected arrivals as discussed by Yilmaz [204]. Seismic data recorded using multi-offset spread setups includes reflected waves that follow hyperbolic arrival patterns as well as refracted waves that show linear arrival patterns. Reflected arrivals are recognized by hyperbolic time-delays versus offset, whereas refracted arrival patterns are characterized by linear time-delays versus offset. Because of its hyperbolic nature, the normal moveout correction of reflected waves results in a non-linear stretching of the reflected wavelets that necessitates a mute scheme to minimize the effect. On the other hand, the correction for the delay-time of refracted arrivals is a linear process and therefore does not generate stretching. The linear and non-linear corrections are combined to reduce both the refracted and reflected arrival times to zero-offset reflection time and adjusted delay-time. After these reductions multi-fold seismic data sorted into common-mid-point gathers can be stacked to produce a single trace.

The conventional use of refracted arrivals is limited to single-fold data that represent either planar or irregular refraction interfaces (Barry [13], Palmer [131], Lankston [105]). Imaging with refracted waves was addressed by Taner et al. [180]. Utilization of the generalized reciprocal method of Palmer [131] without first

break picking was introduced by Selvi [158]. Generally, the conventional interpretation of refracted arrival times is based on the use of delay-times of single-fold data in the form of source and/or receiver gathers (see page 298 and Figure 6.20, this volume). The refraction stack discussed here is also based on the use of refraction delay-times by utilizing multi-fold data acquired conventionally to record reflected waves at different offsets. The data sets are sorted to generate common-mid-point (CMP) gathers that represent a common-refractor-surface (CRS), the linear refraction moveout is removed, and the data are stacked. Here, we summarize the method and discuss its application for the case of a dipping layer.

The traveltime t_x for a critically refracted wave arriving at a source-receiver distance x (Figure 6.49a results in a travelttime that is given by

$$t_x = t_0 + \frac{x}{V_2} \quad (6.46)$$

where t_0 is the delay- or intercept-time and V_2 is velocity of the refractor. The delay-time t_0 is the time for the critically refracted ray to travel from S to E plus from F to R, and is

$$t_0 = \frac{2h}{V_1} \cos i_c \quad (6.47)$$

where h and V_1 are the thickness and velocity, respectively, of the first layer and

$$\cos i_c = \sqrt{1 - \sin^2 i_c} \quad (6.48)$$

where

$$\sin i_c = \frac{V_1}{V_2} \quad (6.49)$$

The delay time t_0 consists of a source term ($t_{SE} = SE/V_1$) that is the time it takes the critically refracted wavefront to travel from the source point to the refractor with an angle of incidence i_c , and a receiver term ($t_{FR} = FR/V_1$) that is the travelttime for the critically refracted wavefront to travel from the refractor to the receiver (Figure 6.49a). The delay time is therefore the two-way travelttime for the critically refracted wavefront from source to refractor and then to a receiver. Since the delay time is obtained from Equation (6.46) by defining $x = 0$, Figure 6.49b is obtained by moving the critically refracted raypaths without modifying the critical angle of incidence i_c to simulate $x = 0$. In Figure 6.49b the source and receiver points are placed at the mid-point between the source and receiver locations in Figure 6.49a.

The time difference between the arrival time of a critically refracted wave at an offset x and the delay time is called the refraction moveout (RMO) at offset x (Figure 6.50). For a planar refractor the refraction moveout is obviously linear. Removal of the refraction moveout transforms the arrival time of the critically refracted ray given in Figure 6.49a into the delay time as the two-way travel

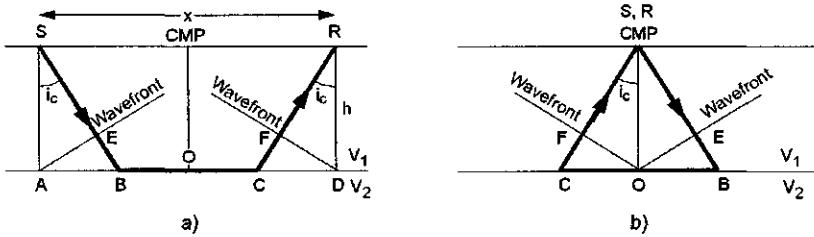


Figure 6.49: Critically refracted raypath before (a) and after (b) refraction moveout. CMP is the mid-point between source and receiver. $t_{SE} = SE/V_1$ and $t_{FR} = FR/V_1$ are the source and receiver terms of the delay time, respectively. (b) represents the critically refracted planar wavefront to travel from CMP point to the refractor and back to the surface CMP point. (RefractionStacks.cdr \Rightarrow f1.wmf)

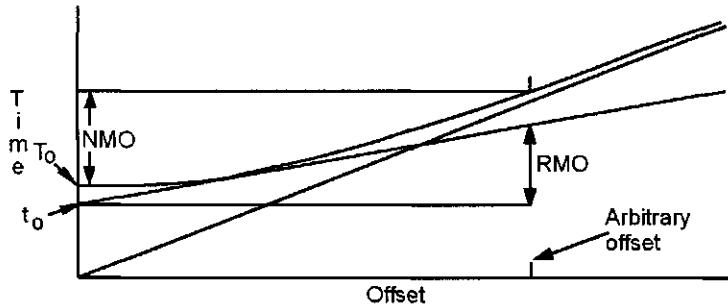


Figure 6.50: Definition of linear refraction and hyperbolic reflection moveout. RMO and NMO are the linear refraction and hyperbolic reflection moveout at an arbitrary offset.

time of the critically refracted ray at zero offset, as shown in Figure 6.49b. This reduction is a mathematical extrapolation because there is no physical refracted wave at zero offset. For the physical significance of the intercept time at zero offset see Figure 6.31 on page 300.

From Figure 6.49b the relation between the refraction delay time and the two-way zero-offset reflection time T_0 shown in Figure 6.50 is

$$T_0 = \frac{t_0}{\cos i_c} \quad (6.50)$$

where $T_0 \geq t_0$. The refraction delay times that are converted into reflection times by the cosine relation is called the adjusted delay-time, which represents time in the refraction stack sections.

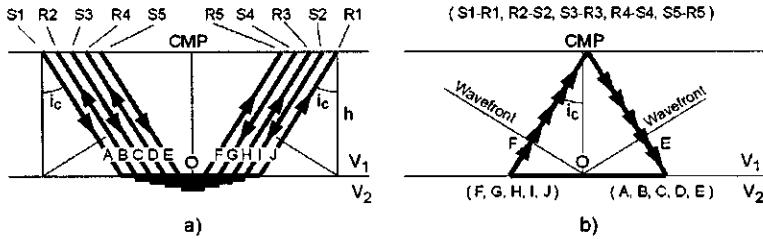


Figure 6.51: Multifold critically refracted raypaths before (a) and after (b) refraction moveout. (a) Split-spread data sorted by the mid-point coordinates forms ensembles with the common-refraction-surface raypaths that show 5-fold beneath the CMP point. (b) 5-fold common refraction raypaths after the refraction moveout.

Refraction Stack

Before the effect of a dipping refractor in common-mid-point sorted gathers that represent a common-refractor-surface is discussed, we summarize the refraction stack. Conventional multifold reflection data acquisition also results in common refractor surface coverage by multifold critically refracted raypaths because source and receiver pairs that represent each raypath are symmetric about a common-mid-point (CMP), as shown in Figure 6.51a. This figure represents split-spread recorded traces sorted by their common-mid-point coordinates. CMP or CRS sorted data from non-split recorded traces would give the same result. From Figure 6.51a it is obvious that a common-mid-point gather sorted prior to stacking the reflected waves can be considered to be a common-refractor-surface (CRS) ensemble for the critically refracted arrivals. CRS ensembles sorted by offset exhibit arrival times that are characterized by linear delays, i.e., the refraction moveout. As shown in Figure 6.51a multiple refracted raypaths cover the same refractor surface beneath the surface location of the CMP. The delay times for these raypaths that cover the same surface can be obtained, and represent the delay time for the common-mid-point (CMP). Because of the multi-fold nature of the data sets one can use stacking to combine and reduce the data sets to represent the CMP point of that gather. The stacking process is subject to some smoothing effects because of the different lengths of the refractor surfaces, as shown in Figure 6.51a.

The number of raypaths covering the same part of the refractor is a maximum at the center directly beneath the common-mid-point (CMP) and tapers out in both directions. This suggests that the wavelet after stacking and the delay time (now the arrival time of the wavelet) represent the subsurface beneath the CMP point. The delay time mapped in the refraction stack section can be related to the thickness of the top layer and its velocity because removal of the refraction moveout reduces all arrival times in a CRS ensemble to a common delay time, as shown in Figure 6.51b. After the removal of refraction

moveout, all the critically refracted wavelets will appear at the same time if the refractor is planar and the velocities of the first and second layer are relatively constant over the limited area traversed by the raypaths. If the refractor is irregular, the wavelets will arrive around an average delay time and increase the effect of smoothing. This background forms the basis for using common refraction surface ensembles to remove the linear refraction moveout and stack data to obtain refraction sections, which are then combined with the reflection stacks. It is also possible to include removal of the linear refraction moveout and hyperbolic reflection moveout within a single processing step.

As discussed in Çoruh et al. [46], the refraction stack sections are constructed in the ($CMP-t_0$)-domain and are therefore suitable for direct visual interpretation. The refraction stacking process, in contrast to the conventional use of refracted arrivals, utilizes first refracted arrivals as well as later refracted arrivals. The processing flow of a refraction stack is similar to that of a conventional processing flow for reflected arrivals. The refractor velocity required to remove the refraction moveout can be obtained from a velocity analysis that is based on removal of linear refraction moveout. A general processing flow for a refraction stack from a multifold data set might include: refraction datum statics, amplitude balancing, spectral whitening, sorting, determination of refractor velocity, removal of refraction moveout, stacking, amplitude balancing, filtering, etc. The flow might include spectral whitening after stack and surface consistent residual statics. A partial migration effect can be obtained if the removal of refraction moveout can be incorporated with the generalized reciprocal method [131] as discussed by Taner et al. [180] and Selvi [158].

It is expected that the refraction stack will enhance critically refracted arrivals while attenuating noise and other events. The stacking process can also be used to scan for the velocity required for the refraction moveout corrections. In the case of a planar sub-horizontal refractor the velocity obtained from such an analysis will be the actual refractor velocity (see page 293 and following), which is

$$V_2 = \frac{V_1}{\sin i_c} \quad (6.51)$$

In the case of a dipping refractor the velocity determined will be an apparent velocity. Since the concept of apparent velocity is also valid for critically refracted arrivals in a common source domain, we attempt to show the difference. Apparent velocities in a common source domain are recognized on a reversed refraction profile (page 317 this volume). The two apparent velocities are

$$V^- = \frac{V_1}{\sin(i_c + \alpha)} \quad (6.52)$$

and

$$V^+ = \frac{V_1}{\sin(i_c - \alpha)} \quad (6.53)$$

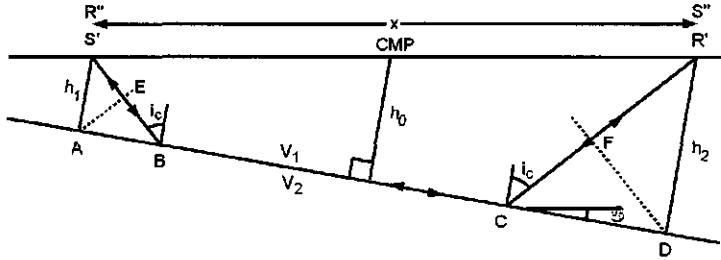


Figure 6.52: Critically refracted raypath from a dipping refractor. Using Equation (6.56) the critically refracted arrival can be reduced to a delay time that represents h_0 thickness beneath the common-mid-point.

where α is the slope of the refracting surface. These apparent velocities obtained from a reversed refraction profile can be averaged to approximate the velocity of the refracting medium. The velocity of the refractor from the down-dip and up-dip measured apparent velocities is

$$V_2 \approx 0.5 \left[\frac{V_1}{\sin i_c - \alpha} + \frac{V_1}{\sin i_c + \alpha} \right] \quad (6.54)$$

In the case of a dipping-refractor and common-mid-point sorted data that simulate a common refractor surface one can write the following from Figure 6.52. The arrival time of the critically refracted wave between point S' and R' is

$$t_x = \frac{x \cos \alpha}{V_2} + \frac{h_1 + h_2}{V_1} \cos i_c \quad (6.55)$$

This is independent of the direction of recording. Using the layer thicknesses h_1 , h_2 at the source and receiver, and the layer thickness h_0 at the mid-point CMP one can write

$$h_1 = h_0 - dh$$

and

$$h_2 = h_0 + dh$$

Therefore Equation 6.55 can be rewritten as

$$t_x = \frac{x \cos \alpha}{V_2} + \frac{2h_0}{V_1} \cos i_c \quad (6.56)$$

Equation 6.55 indicates that an apparent velocity defined by

$$V_{apr} = \frac{V_2}{\cos \alpha}$$

can be used to remove from the arrival time the time associated with the path between A and D or between E and F. This results in the delay time. This delay time is associated with h_0 beneath the CMP point. Therefore, the use of the actual velocity will result in an incorrect delay time, while use of the apparent (incorrect) velocity will result in a correct delay time. This discussion implies that refraction stack sections from common-mid-point gathers can be obtained from data sets recorded with either split or off-end recording configurations. It is apparent that the direction of wave propagation with respect to the dip direction is not important even if off-end recording is used.

Composite Stack Sections

The refraction stacked traces from the CRS ensembles represent two-way traveltimes (delay times) for critically refracted travel paths beneath the common-mid-point and therefore can reveal subsurface structure. In spite of the smoothing effect of refraction stacking, the portion of the refractor that contributes most to the stack is approximately beneath the common-mid-point, and can be a relatively small area depending on the offset range used. In reducing the arrival time of refracted waves to delay times one needs only to know the refraction moveout velocity. In reducing to the modified delay times, in addition to the refraction moveout velocity, the velocity of the top layer (V_1) is also required. Both velocities are determined from the velocity analysis that scans for velocities using the linear moveout. Using these velocities and Equation (6.50), the delay times can be converted to the modified delay times that represent the zero-offset reflection times. The conversion is more critical when the contrast between seismic velocities is larger.

The first example is a comparison given by Çoruh et al. [46] and shown in Figure 6.53 where an original reflection stack section (left) is compared with the composite refraction-reflection stack section. The critically refracted waves are limited to the shallow part of the section and marked accordingly.

As reported by Çoruh et al. [46], the reflection stack section on the left in Figure 6.53 was originally recorded and processed to interpret subsurface faults in the Atlantic Coastal Plain sediments in South Carolina. The original seismic data and its interpretations revealed a reverse fault that penetrates upward into the Coastal Plain sediments. Recognition of this fault made it necessary to determine the other faults and their depths of upward penetration in order to determine the latest tectonic activity that produced displacement in the area. A reprocessing effort that was targeted to recover reflections shallower than those in the seismic section shown on the left in Figure 6.53 was carried out (Sen [159], Çoruh et al. [48, 46], Domoracki [64]). The reprocessing recovered additional shallower reflections and helped to determine the upward penetration of the faults more accurately, as shown on the right in Figure 6.53. Interpretation of the recovered reflections showed that the border fault (the Pen Branch Fault) between Triassic strata of the Dunbarton basin and crystalline basement penetrated upward to as shallow a depth as the shallowest reflection. In order to trace this fault and others at shallower depths the composite refraction-reflection

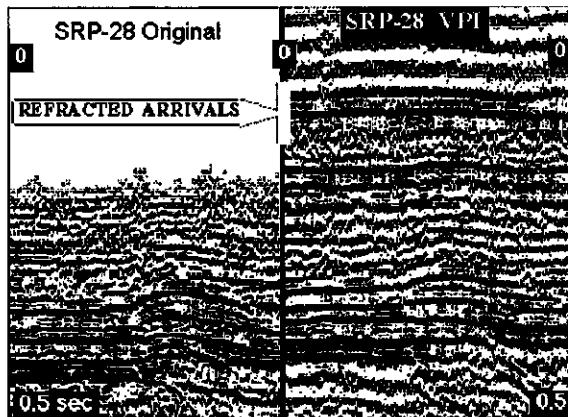


Figure 6.53: Composite refraction-reflection stack section - Line SRP-28. Left half: Original reflection stack from a conventional processing to trace the upward extent of faults penetrating from the basement into the Atlantic Coastal Plain sediments. Right half: Composite stack from a processing scheme that recovers reflected and critically refracted arrivals from shallower depths (From Çoruh et al. [46]).

stack sections were produced. After velocity analysis, the recovered refracted arrivals were combined with reflected arrivals after stacking. The composite refraction-reflection stack section is shown in Figure 6.53 (right).

A comparison of a conventional reflection stack versus a refraction stack for the same reflector/refractor is shown in figure 6.54. The refraction stack is clearly superior. Two refraction stacks from the same line that image two different refractors at different depths are shown in Figure 6.55. Each refractor is well-defined and each exhibits the characteristic convex-upward deformation of the hanging wall that is associated with the major Dunbarton Triassic Basin border fault (Pen Branch). This fault, initially caused by extension during basin formation, subsequently became a reverse fault that penetrated upward into the sediments of the Atlantic Coastal Plain from crystalline basement where it has been clearly imaged [63]. The refraction stacks therefore suggest that deformation associated with the basement fault has propagated upward from the crystalline basement into the overlying sediments all the way to the surface.

A final example shown in Figure 6.56 is a comparison of the shallowest reflector and refractor that could be imaged by reflected and refracted events. It demonstrates how the refraction stack can successfully image a shallow refractor that is not even apparent on the conventional reflection stack.

Interpretation of the composite refraction-reflection stack sections were found to be useful in determining the upward penetration of faults imaged in the crystalline basement, the Dunbarton Triassic basin and the Cretaceous-Tertiary Atlantic Coastal Plain sediments in South Carolina. In general, as reported

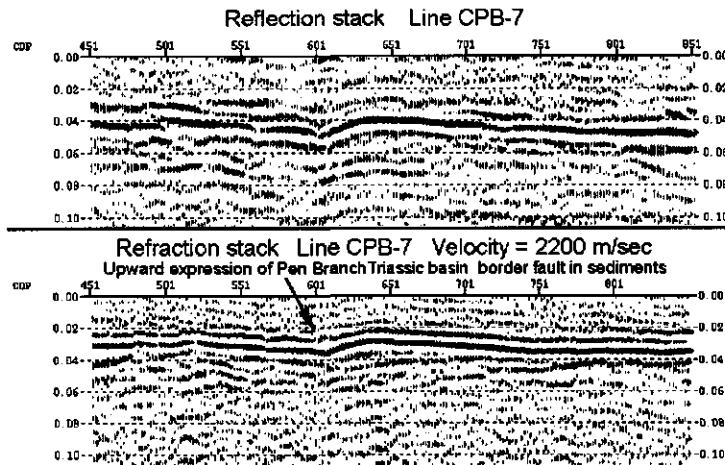


Figure 6.54: The same reflector/refractor has been imaged by both processing methods. Better resolution was obtained from the refraction stack (bottom). Arrow shows interpreted location of the Pen Branch fault, the upward penetration of the major boundary fault of the Dunbarton Triassic basin.

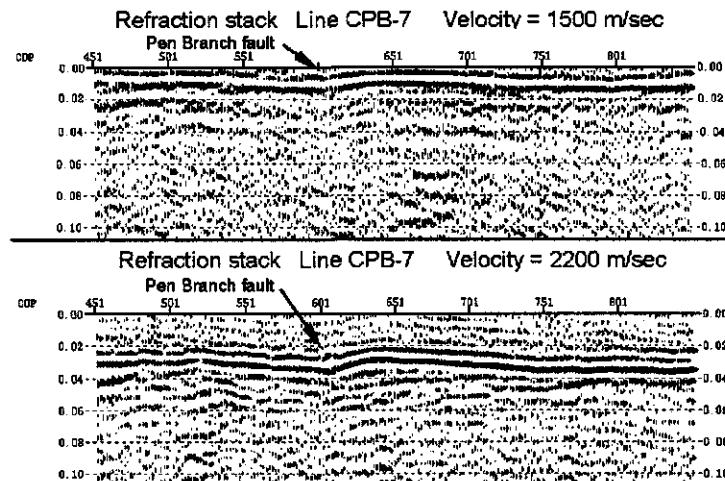


Figure 6.55: Two refraction stacks that image two different refractors at different depths on the same line. Note the different stacking velocity required for each depth. The same characteristic convex-upward deformation (between CDPs 600-750) of the deeper and shallower refractor suggests that this deformation extends all the way up to the surface.

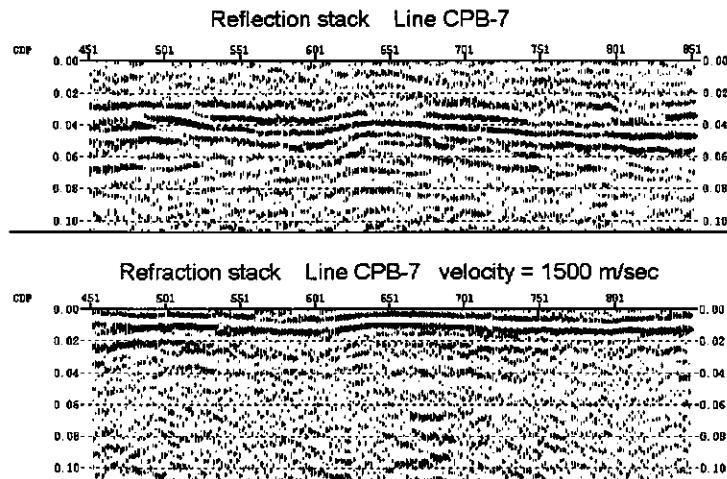


Figure 6.56: The same portion of the line has been processed as a reflection stack (top) and a refraction stack (bottom). The very shallow refractor (bottom), which was the target of interest because of the question of the extent of upward penetration of the major basin boundary fault into the sediments from crystalline basement, was not even imaged on the reflection stack. See also Figure 6.55.

by Çoruh et al. [46] and Domoracki et al. [64], the faulting in the area is not limited to the Triassic Dunbarton basin, which is interpreted to be bounded by faults reactivated under compression since the late Cretaceous. The other faults imaged in the sediments extend from basement upward into the sediments. Deformation without faulting imaged in Tertiary sediments can be associated with shallower upward penetration of faults that are imaged at deeper depths. Displacement imaged along faults decreases rapidly upward from the basement indicating growth faulting during the late Cretaceous and Tertiary. The composite refraction-reflection stack sections indicate that the depth of upward penetration of the faults varies; most of the faults are associated with deformation at times as small as 50 ms two-way time (≈ 25 m).

In conclusion, it is suggested that use of combined reflected and refracted arrivals in producing seismic sections can offer key images for interpreting shallower structures. This approach requires an attempt to retain as much recorded data as possible in producing seismic sections. In addition to retaining reflected arrivals, the refracted arrivals can be included in the final stack sections, i.e., the composite sections. The refracted events recovered by stacking in the composite refraction-reflection sections offer important information about shallower structures that is useful for different purposes. The examples presented in this paper imaged shallow structures that are important to determine the upward penetration depth of the faults in the Atlantic Coastal Plain sediments. As a

final comment, the arrival times of refracted events from refraction stack sections can also be used directly to correct/refine statics created by near-surface velocity variations.

Chapter 7

Seismic Source Wavelets

Dix [137, 62] discussed the “minimum oscillatory character of a pulse” and suggested that the primary seismic pulse leaving the sourcepoint must be relatively simple. An oversimplified disturbance from an impulsive (e.g., explosive) point source is shown in Figure 7.1. After the disturbance has passed, the medium returns to its original state with no permanent deformation. If the initial disturbance at the onset of the pulse is a compression (+), then there must be a rarefaction (−) later in the wavelet. In reality, a real wavelet leaving an impulsive source is more oscillatory than that shown in Figure 7.1. A more realistic wavelet is shown in Figure 7.2. Away from the source the deformation of the medium still returns to zero [137]. Any physical seismic wavelet must have a zero mean because the transmitting medium suffers no permanent strain (away from the source). If there are several compressional maxima in the medium, then there must also be several later rarefactions in order to produce zero deformation. The ultimate seismic wavelet with the best resolution would be the Dirac delta function $\delta(t)$ or its discrete equivalent, the Kronecker delta, defined as $\delta_{ij} = 1$ for $i = j$, $= 0$ for $i \neq j$. Either lasts an infinitesimal amount of time with no possibility for overlap or interference between successive reflected arrivals. Alas, the earth will not transmit the uniform and constant frequency spectrum associated with either $\delta(t)$ or δ_{ij} , and the relative attenuation of their higher frequency components quickly increases the duration of the reflected wavelet. Those wavelets that arrive at about the same time therefore interfere with each other and their shape is concealed from direct observation. The wavelet shown in Figure 7.2 is used for many of the numerical computations in this volume.

Before processing, seismic wavelets vary greatly in length and frequency content primarily because of reverberations that commonly increase their time duration. Deconvolution, a common data processing step to shorten wavelet length and shape them in the direction of a Dirac delta function, increases their bandwidth and therefore the resolution of the seismic method.

Surface waves of large amplitude generated by the source itself interfere with the reflected source wavelets. Surface waves propagate in a horizontal

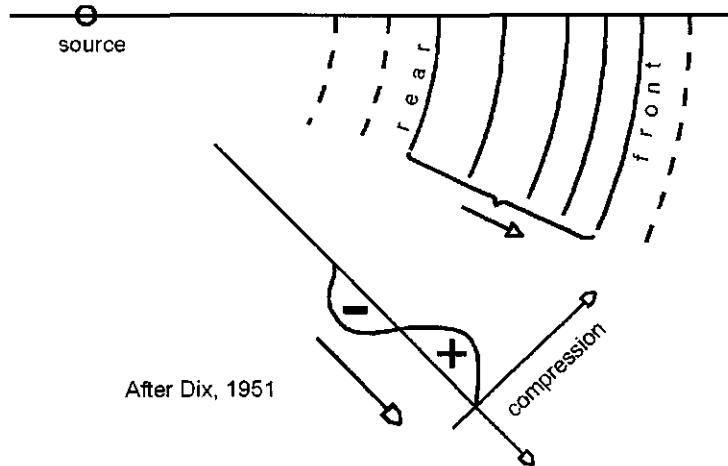


Figure 7.1: The oscillatory character of a general pulse. Any wavelet from an impulsive source must have a zero mean because the transmitting medium suffers no permanent strain. If there are several compressional maxima in the medium, then there must be several later rarefactions in order to produce zero deformation. After Dix [62]. Dix.cdr → dix342.wmf

direction; reflections travel upward in a more vertical direction. This difference in propagation direction can be exploited to attenuate the low-frequency surface waves that arrive at the same time as deeper reflections of interest. Judicious spacing of the receivers can attenuate the horizontally-travelling surface waves that travel outward from the source. The design of data acquisition surveys is not the purpose of this volume but an example from Kolich [100] is shown in Figure 7.3 to illustrate the importance and effectiveness of the use of receiver arrays on land where surface waves are a problem. The data shown in the figure were acquired on the topographically high Bane Dome, a structural window near Blacksburg, VA, where the thickness of Paleozoic shelf strata is about 10 km. The “reflected surface waves” are reflected from steep topography. Noise from surface waves is just one more example of why the changing shape of the seismic source wavelet is difficult to obtain.

After processing, most seismic wavelets familiar to the petroleum industry tend to persist in time for somewhere between 40-100 ms, and have a smooth amplitude spectrum that peaks somewhere between 15 and 30 Hertz, with the energy in the spectrum concentrated at frequencies between 8 and 55 Hertz. For high resolution, near-surface studies the vibrator energy spectrum is shifted upward to about 40-350 Hertz. During data acquisition, it is desireable to maintain a frequency bandwidth of at least two octaves to minimize the amplitude of side lobes in the wavelet.

If the shape of the reflected source wavelet were known at all times on a

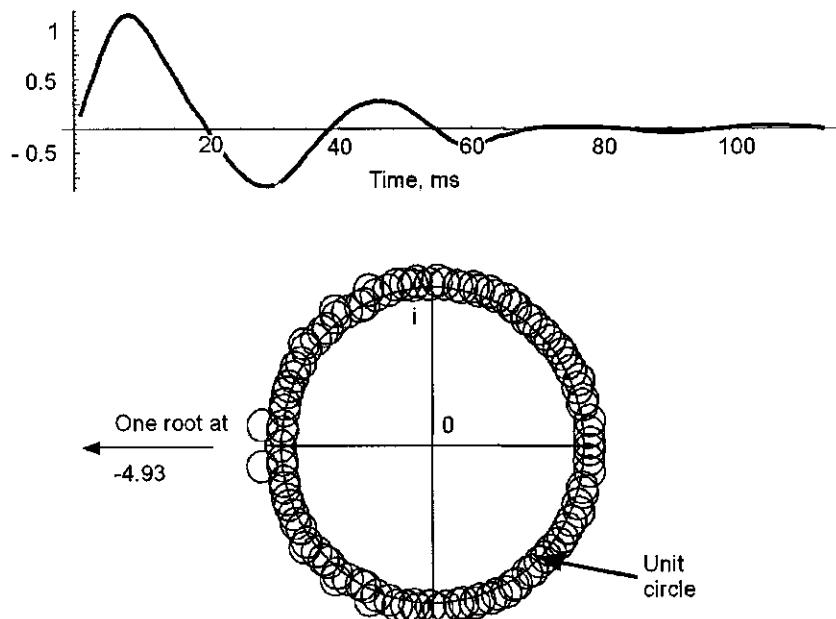


Figure 7.2: Top: Approximate shape of a typical minimum-delay wavelet. Sampling interval is 2 milliseconds. Bottom: Roots plotted on unit circle. MinimumDelayWavelet.cdr \Rightarrow MinimumDelayWavelet2.wmf

seismic trace, then it would be a simple matter to use linear filter theory to recover the reflection coefficients that are more directly associated with the geology. Wavelet shape, however, must usually be deduced indirectly from the entire seismic trace by some statistical approach (e.g., autocorrelation). Isolated, time-invariant wavelets do not appear on seismic data because of the nature of interfering and superimposed reflected wavelets that make up the seismic trace. The remarkable success of “predictive deconvolution” (page 454) to alter the shape of the unseen wavelet in time and thereby increase resolution suggests that, to a first approximation, the seismic reflection process is indeed a simple one.

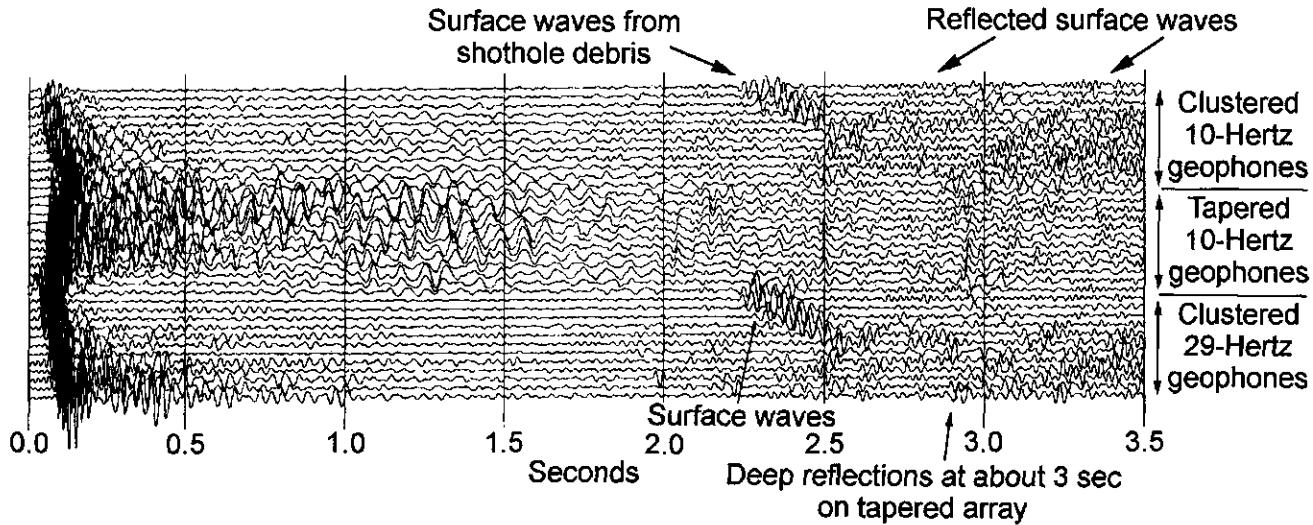


Figure 7.3: Energy source was ten pounds of Unigel at a depth of 76 m. Arrays of 12 clustered 29-Hz, 12 clustered 10-Hz, and 12 10 Hz receivers, the latter having a tapered response, occupied the same surface coverage. Tapered array was weighted by using 1 – 2 – 3 – 4 – 5 – 4 – 3 – 2 – 1 receivers with 10.7 m spacing between weights; overall length of tapered array was 82 m. The effectiveness of the tapered array is evident especially over the interval from 2 to 3 sec where reflections from deep Paleozoic shelf strata are clearly recorded but are absent on the two clustered segments where shothole debris and reflected refractions dominate that interval. Average velocity of Paleozoic section is 6 km/sec. From Kolich [100].

7.1 Energy Sources

Dynamite and the vibroseis energy source are now the two most commonly used methods for geophysical exploration on land. The use of a vibrator has the lowest environmental impact (but a vibrator sometimes leaves rectangular pad depressions in a roadbed that might not have been properly constructed).

7.1.1 Dynamite

Dynamite, a relatively uncontrolled impulsive energy source, was the first source used for seismic exploration [6]. When buried in a borehole dynamite provides an impulsive source with a wide bandwidth. A specially formulated product called *Dynoseis* is a high-velocity (6.7 km/sec), high-density (1.78 gm/cm^3) and high-energy explosive designed for geophysical exploration. Five pounds would be considered to be a small charge. High velocity means the explosive column will detonate faster, thus approaching an impulse $\delta(t)$. A higher density means you can use a shorter length of charge in a borehole. *Dynoflow* is a blasting agent that can be pumped into boreholes. Whatever the buried explosive agent it is important wherever possible to 1) place the charge below the weathered layer or at least in relatively unweathered material, and 2) to place the charge in the same lithologic unit below the weathered layer if such lithologic continuity can be recognized in the area. This is important for the calculation of “static” time corrections required for processing the data [10, 177, 117]. A seismogram obtained using an explosive source is shown in Figure 7.3.

7.1.2 Vibroseis

Vibroseis is a less energetic but more controlled surface energy source. During the 1990s vibroseis was the energy source for over half of the land 2-D and 3-D geophysical crews operating worldwide [125], which translates to about 75 percent of the mi/mi² of seismic data for petroleum exploration or reservoir delineation.

The “vibroseis” method of seismic prospecting was invented by Conoco and has been described by Crawford, Doty, and Lee [53]. Much has been written about this source and only a brief summary is given here. One or more continuous signal vibrators in a source array provide the energy, thus replacing the dynamite source. The frequency bandwidth can be controlled, offering possibilities for improving the signal-to-noise ratio in areas where surface waves are a problem. The vibroseis energy source is the most common seismic source for land use.

Pads mounted beneath trucks or trailers are placed in contact with the surface of the ground and set into vibration starting at one frequency ν_1 and ending at ν_2 . The time it takes for the transition from ν_1 to ν_2 is called a *sweep* or *pilot sweep*. The transition from ν_1 to ν_2 is commonly linear with time, although convincing arguments have been made by Pritchard [142] that non-linear sweeps can and should give better results.

A unique, nonrepeating signal with a duration of many seconds is generated by the vibrator. The source wavelet is thus considerably longer (6,000 - 24,000 ms) than the final processed seismic wavelet. The pilot signal is reflected from the subsurface and recorded in the usual way; however, few conclusions can be drawn by merely observing the recorded signal because it is composed of hundreds of pilot sweeps (source wavelets) that overlap in the usual way. The recorded vibroseis sweep "wavelet" can be, in fact, 32,000 ms long!

The seismic trace recorded in the field using a vibroseis source is the superposition of many reflected pilot sweeps. The pilot sweep is crosscorrelated with the overlapping sweeps reflected from the subsurface. The crosscorrelation process is summarized in Figures 7.4 and 7.5. The pilot sweep is shifted (without folding) one sample point at a time, and after a shift multiplied with the recorded trace point-by-point. The products are then summed (Figure 7.4). A single sum yields a crosscorrelation coefficient for that time shift (lag). The resulting collection of all crosscorrelation coefficients is the Klauder wavelet. The maximum value of the crosscorrelation (autocorrelation in this case) is for a lag of $\tau = 0$, when the copy of the pilot sweep "finds itself" exactly superimposed over the pilot sweep. The field seismogram as recorded is uninterpretable (Figure 7.5c) until the additional step of crosscorrelation of the recorded trace with the pilot sweep results in compression of the sweep into a Klauder wavelet (Figure 7.5c).

The additional computing step called crosscorrelation is therefore required before vibroseis data can be interpreted, leading to the much shorter *Klauder wavelet*, as discussed in Section 7.2.2.

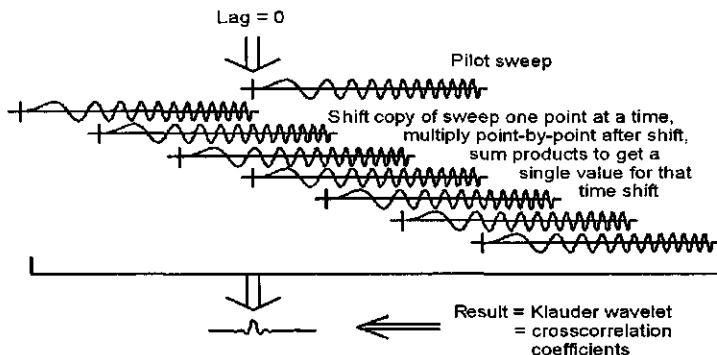


Figure 7.4: As the pilot sweep is moved along a reflected version of itself the "matched filter" process of crosscorrelation results in a Klauder wavelet with a central lobe that coincides exactly with the arrival time of the reflected pilot sweep. (VibroseisRecordingProcess.nb → VibroseisRecordingProcess.cdr → VibroseisRecordingProcess2.wmf)

The conventional procedure used to acquire vibroseis data is to sum in the field the contributions from several vibrator sources that make up a source array. Belcher [15] discussed an alternative method of recording each sweep

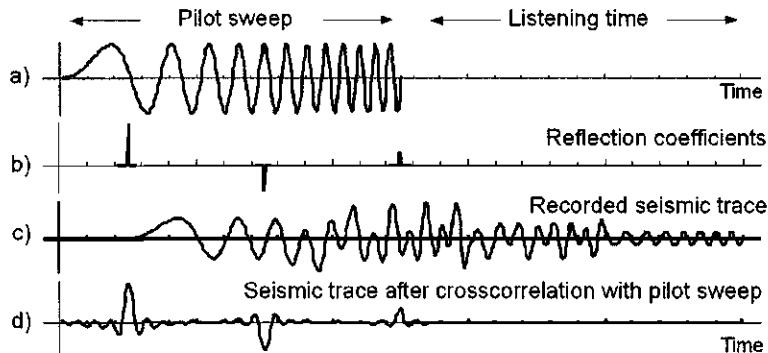


Figure 7.5: Top: (a) pilot sweep. (b) Three reflection coefficients. (c) Recorded seismic trace = the convolution of (a) and (b). The field seismogram as recorded is uninterpretable until the additional step of crosscorrelation of the recorded trace with the pilot sweep results in compression of the recorded trace into Klauder wavelets. (d) The final result after crosscorrelation of the pilot sweep with the recorded seismic data. As the pilot sweep is moved without folding along the recorded seismic trace, the “matched filter” process of crosscorrelation has identified those trace times that correlate most closely with a reflected pilot sweep. The result is three Klauder wavelets with central lobes that coincide exactly with the reflection coefficient times. Usually the sweep is tapered on each end to reduce the amplitudes of the side lobes (see Figure 3.20 on page 82). For deep crustal research, the Virginia Tech seismic crew commonly used a pilot sweep duration of 24 sec, a listening time of 8 sec (giving a total record length of $24 + 8 = 32$ sec), a sampling interval of 2 ms, an upsweep bandwidth of 14 – 56 Hz, and a 1-sec taper on each end. VibroseisRecordingProcess.nb → VibroseisRecordingProcess.cdr → VibroseisRecordingProcess1.wmf)

that provides more flexibility in the processing stage. The output from each pad position in the source array is recorded separately rather than summing in the field. He showed that if cancellation of surface waves by a source array is not a requirement, then processing each sweep as a separate sourcepoint can result in increased lateral resolution. This procedure was applied to seismic data acquired by Virginia Tech over a buried rift basin in the southeastern United States. The results are shown in Figure 7.8.

Although vibrators are almost always used in a source array of three to five vibrators, this is not always the best approach. The Virginia Tech crew used a single Failing Y-1100A vibrator to image deeper crustal targets as well as relatively shallow objectives. Use of vibrators over crystalline terrane is difficult under any conditions but the single vibrator results shown in Figure 7.9 over the crystalline terrane in central Virginia are clearly superior to those from a four-vibrator source array. The difference here is not in processing because the same processing steps were used in each case. The average velocity of the crystalline



Figure 7.6: Top left: Single Failing Y-1100A vibrator (Bill Davis, crew foreman). These are commonly deployed in a linear array of 4-5 vibrators. Top right: closeup of vibrating pad. Bottom left: Visiting delegation from China. Bottom right: recording instrument truck left, vibrator right. Vibroseis-RecordingProcess.nb → VibroseisRecordingProcess.cdr → VibroseisPictures1.bmp)

rocks in this location is 6 km/sec so the depth range shown is from 6 to 9 km.

Resolution is affected by bandwidth. A three-octave bandwidth is better than two octaves. The reflectivity of the Paleozoic shelf strata in the southeastern United States is excellent, as shown in Figure 7.11. The site is a structural dome in Paleozoic strata in the relatively unmetamorphosed Valley and Ridge province in southwestern Virginia. These data were first acquired using a single Y-1100A vibrator (left) and a sweep of 14 – 56 Hertz and later re-acquired with the same vibrator over the same roadbed but using a wider sweep of 10–80 Hertz and a longer, by 30%, sweep. The use of a longer sweep should theoretically



Figure 7.7: Shear vibrator. Pyramid-shaped teeth protruding from vibrator plate sink into ground. Plate motion transverse to direction of truck generates shear waves.

improve the signal-to-noise ratio by over 30%. The higher data quality of the stack on the right is believed to be due primarily to the wider bandwidth of the three-octave sweep as opposed to two, which might also have resulted in better residual statics [65, p. 8]. The group interval was 70 m and the sampling interval 2 ms for both lines. Both sections were acquired on the same roadbed over clastics as opposed to carbonates. Carbonates at the surface scatter vibroseis data because of Karst topography. The sections were processed at Virginia Tech using the same software.

As noted (page 188 and Figure 4.4), oscillations in the frequency domain are associated with sharp corners in the time domain. It is therefore desirable to “taper” the sweep at the beginning and the end. That is, an untapered sweep looks like that shown in Figure 7.10a where a rectangular window is shown superimposed on an untapered sweep signal. This will introduce undesirable oscillations into the Fourier amplitude spectrum and the autocorrelation function. A better approach to sweep generation is shown in Figure 7.10b where a tapered window multiplies the sweep amplitudes. For deep crustal vibroseis recording the Virginia Tech seismic crew generally used a bandwidth of 14–56 Hz, a sweep length of 24 sec, and a 1-sec taper on each end of the sweep to minimize oscillations in the frequency domain.

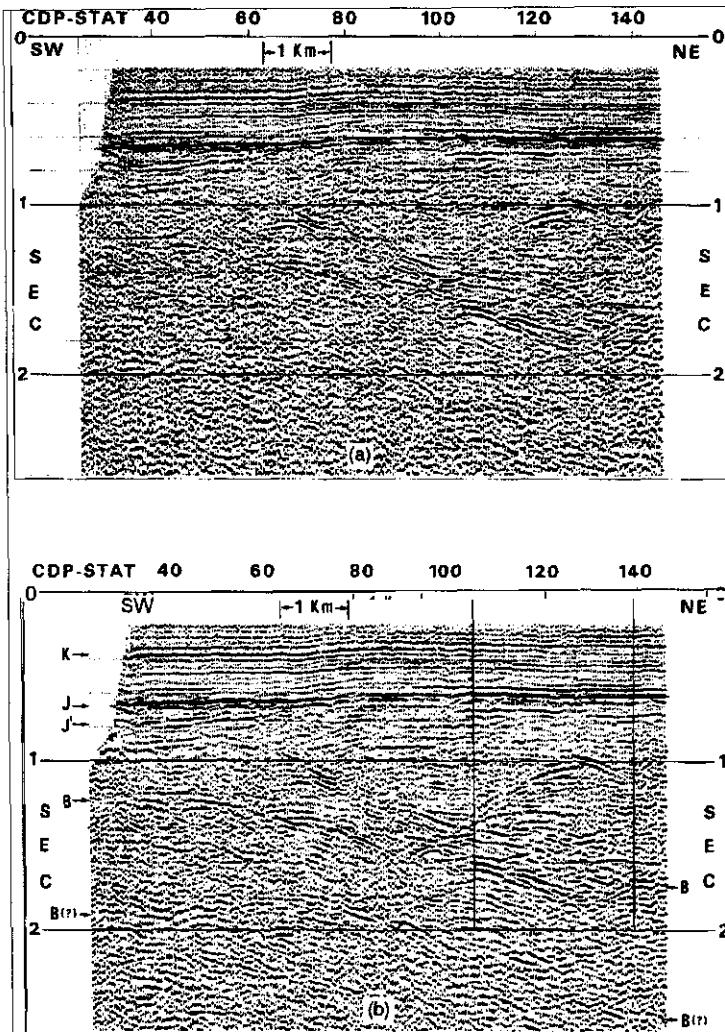


Figure 7.8: Two stacked sections of line VT-5 over a buried Mesozoic basin in South Carolina. Horizontal reflections are from the sediments of the Atlantic Coastal Plain. Dipping reflections between 1 and 2 sec originate from within the concealed Mesozoic basin beneath the ACP sediments. (a) Top: With automatic gain control applied *after* individual sweeps were summed but *before* correlation. (b) Bottom: Automatic gain control applied to individual sweep records *before* correlation and summing. Note improvement in the reflections at 1 sec and between 1.8 and 2.6 sec (B) and (B?) in the lower record section. Figure from Belcher et al. [15, Figure 2, p. 1738].

The amplitude function $a(t)$ of a cosine-tapered sweep is [4]

$$a(t) = \begin{cases} (A/2) \left[1 - \cos \left(\frac{\pi t}{T_1} \right) \right], & \text{if } t < 0 \\ A, & \text{if } T_1 \leq t \leq T_2 \\ (A/2) & \text{if } t > T_2 \end{cases} \quad (7.1)$$

where A is the maximum amplitude of the taper, T_1 is the duration of the taper at the start of the sweep, and T_2 is the duration of the taper at the end of the sweep.

A *Mathematica* version of Equation (7.1) is

$$\begin{aligned} \text{Taper} = & \text{If} \left[t \leq T_1, \frac{A}{2} \left[1 - \cos \left(\frac{\pi t}{T_1} \right) \right], \right. \\ & \left. \text{If} \left[t \leq \text{SweepDuration} - T_2, A, \frac{A}{2} \left[1 + \cos \left(\frac{\pi (t - \text{SweepDuration} + T_2)}{T_2} \right) \right] \right] \right]; \end{aligned}$$

where T_1 and T_2 are the durations of the taper at the beginning and end, respectively, of the sweep, and A is the maximum amplitude of the taper. The function “Taper” is then multiplied by the untapered sweep to obtain the sweep shown in Figure 7.10b.

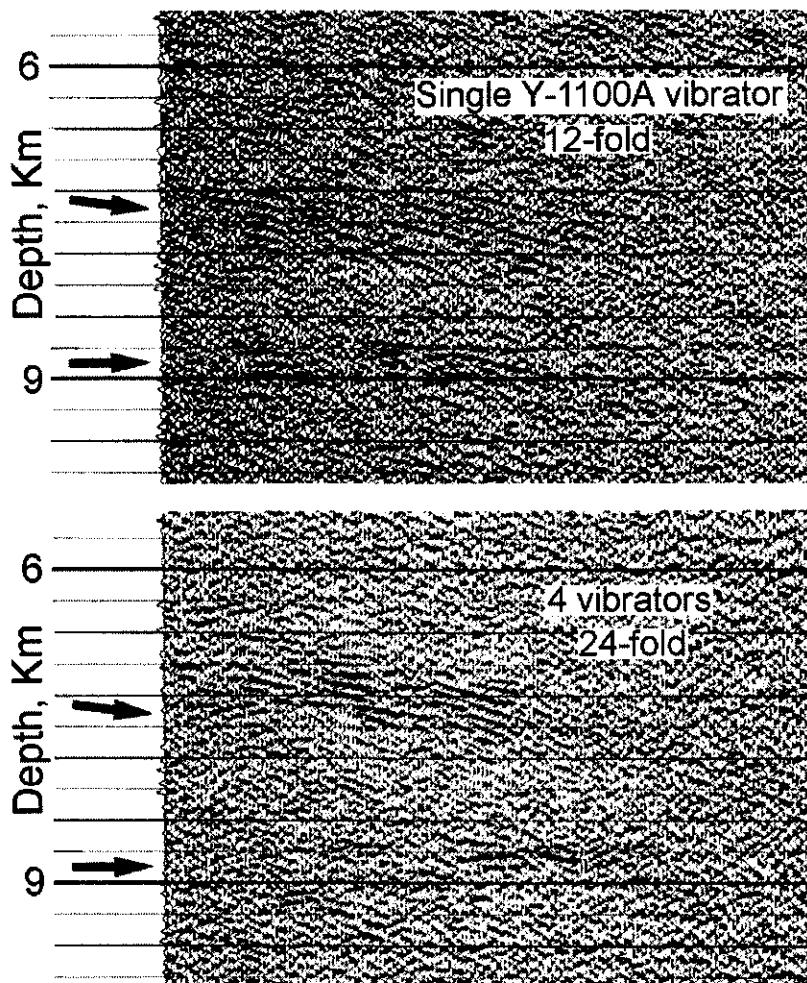


Figure 7.9: Comparison of a single vibrator versus a four-vibrator source array over a depth interval of 6 to 9 km (~ 20,000 to 30,000 ft) in the metamorphosed crystalline terrane of the central Virginia Piedmont. Single vibrator source and multi-vibrator source array were deployed over the same road bed. Top: Single vibrator. Bottom: Array of four vibrators. Arrows indicate reflection zones for comparison. Although the reflectivity and reflector continuity of these metamorphosed rocks is typically poor, the single-vibrator results are superior overall in spite of the higher fold used for the multivibrator source array. The average velocity of the crystalline rocks is 6 km/sec. Horizontal scale equals vertical scale. Not migrated. Both data sets processed with a similar procedure and software at Virginia Tech.

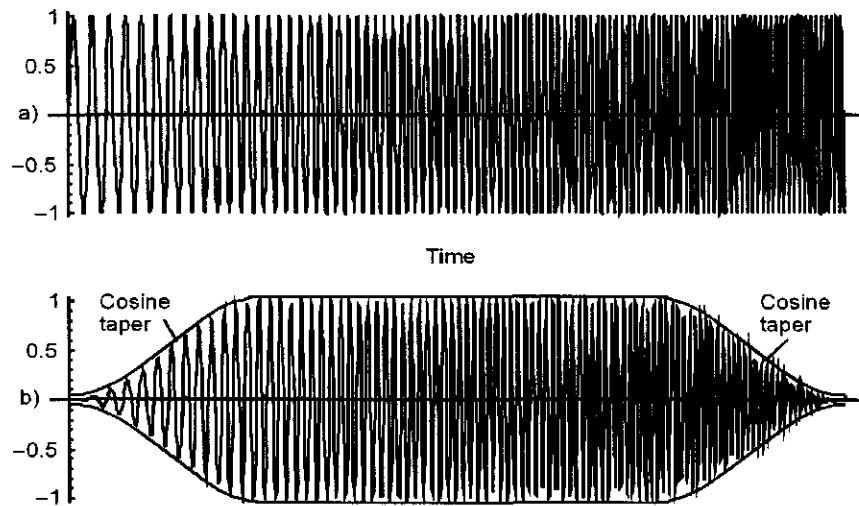


Figure 7.10: a) Untapered sweep showing rectangular window superimposed over the vibroseis sweep signal. Tapering the sweep is multiplication in the time domain, which corresponds to convolution in the frequency domain. Therefore, each frequency component in the true spectrum of the sweep reaches out to adjacent sweep frequencies and contaminates each of the “true” frequency components in the vibroseis sweep. b) The same sweep now tapered by the cosine taper defined by Equation (7.1). Contamination (“leakage”) still takes place but the side lobes of the window spectrum are smaller in amplitude.

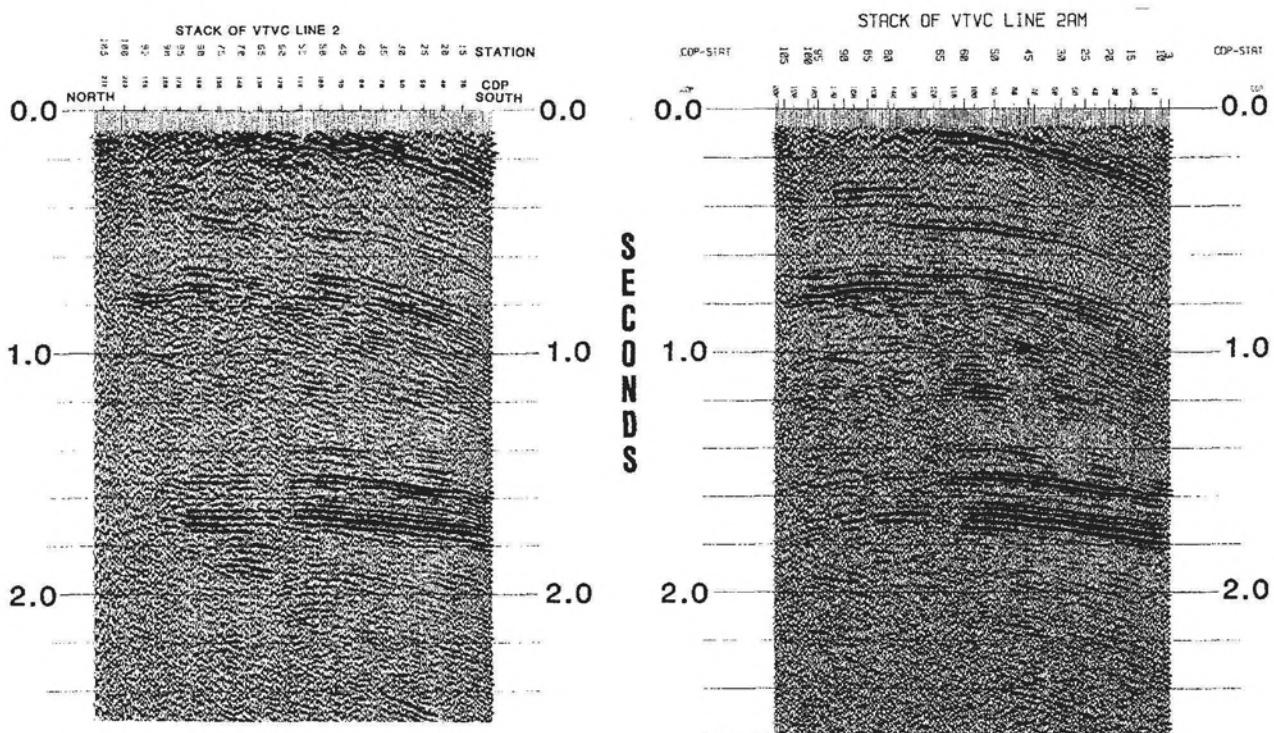


Figure 7.11: Data acquired by Virginia Tech over the Bane structural dome in southwestern Virginia using a single Y1100-A vibrator. Geologic setting is one of folded and thrust-faulted Paleozoic shelf strata. Left: Sweep of 14 – 56 Hertz (two octaves). Figure from Gresko and Costain [80, 81]. Right: Data re-acquired by Virginia Tech over the Bane Dome using the same vibrator over the same road bed but with a sweep bandwidth of 10 – 80 Hertz (three octaves). From Domoracki, Costain and Çoruh [65]. The average velocity of the Paleozoic shelf strata is 6 km/sec. Horizontal scale equals vertical scale. Not migrated. BaneDome.cdr \Rightarrow BaneDome.wmf

The reflectivity of the sediments of the Atlantic Coastal Plain is excellent. This sedimentary wedge thickens to about 3 km beneath Cape Hatteras, NC. Several studies have been made for various reasons other than exploration for hydrocarbons. Single versus multivibrator results are shown in Figure 7.12 from Costain and Çoruh [49]. The reflectivity seen on seismic data depends upon a large number of factors, including the source, recording parameters, processing, line orientation, and surface and subsurface geologic factors. Reflection data acquired in the same area are often different even though similar recording and processing parameters are used. Examples are shown by Costain and Çoruh [49], Scott [157], and Hubbard et al. [87].

7.1.3 DinoSeis, Thumper, and others

The DinoSeis energy source was a low-power large truck-mounted surface energy source invented and marketed by the Sinclair Oil Company. Sinclair used dinosaur cartoons in its advertising and that accounts for its name [114]. The unit sits on a firing chamber filled with propane and oxygen and the mixture is fired by a spark plug to produce one pulse, which produces one seismic record. Normally several pulses are fired at each sourcepoint to increase the signal-to-noise ratio. The output is summed to produce one record. Comparisons between dynamite and DinoSeis are shown by Godfrey et al. [79]; the results compare favorably. The DinoSeis pulse was about equivalent to one 10-sec vibroseis sweep [113]. Lindsey [114] noted that two problems led to its demise:

1. “Flameouts” that burned several thousand acres of prime west Texas grassland and made it difficult to permit in some areas, and
2. Poor firing time accuracy because of spark plug fouling making it difficult to synchronize source elements in an array. Vibroseis units on the other hand can be properly synchronized.

The “Thumper” is “...the ultimate in simplicity: Take a 7000 lb slab of old steel plate, lift it nine feet, and drop it on the ground.” [114]. The Thumper was used in Australia as recently as 1985 [107].

The reader is referred to Lindsey [114] for a historical summary of innovative seismic energy sources used for exploration on land. See also Bay [14].

7.1.4 Marine

Most reflection seismic data are now obtained offshore in such a way that the end result of acquisition and processing yields a 3-dimensional cube of data from which time-slices can be extracted with just about any orientation. The seismic resolution is generally far superior to that obtained on land for two reasons:

1. The hydrophones are located in a medium (water) that has uniform properties with identical receiver coupling throughout the long recording arrays. For marine data there is no near-surface weathered layer with its

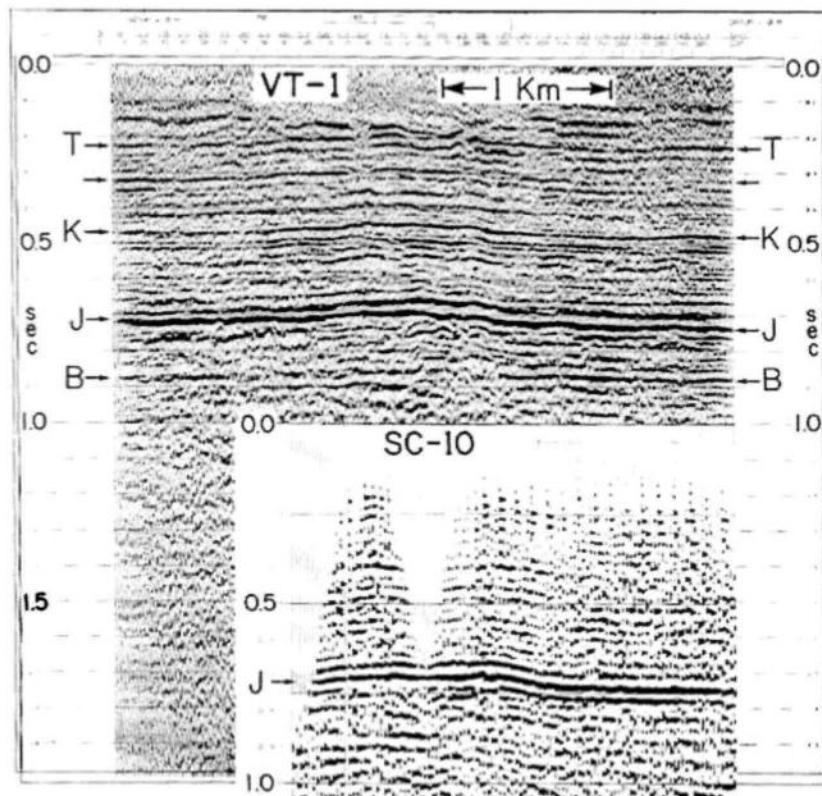


Figure 7.12: Comparison of a single vibrator (Line VT-1) acquired by the Virginia Tech crew (10-80 Hertz, group interval 35 m, 24-fold) versus a 3-vibrator source array (10-60 Hertz, group interval 60 m, 12-fold). Insert positioned over the same surface coverage on the same road bed over sediments of the Atlantic Coastal Plain. Several stratigraphic markers are noted: "J" is a prominent regional reflection from the top of Jurassic volcanics at about 0.7 sec, which corresponds here to a depth of about 700 m. Apparent arching of J-reflector probably due to higher velocity Eocene channel at shallow depths. The difference in resolution between the two lines is attributed to differences in field/recording parameters including the receiver group interval, near and far offsets, receiver and source arrays, sweep frequencies, and the number of vibrators. Differences in processing include editing, muting, statics, velocities, and spectral whitening. Details of the Virginia Tech data acquisition, processing and interpretations are available from Yantis [202] and references therein, Costain and Çoruh [49], and Belcher [15]. Figure from [49].

commonly extreme lateral variations in velocity that cause static shifts

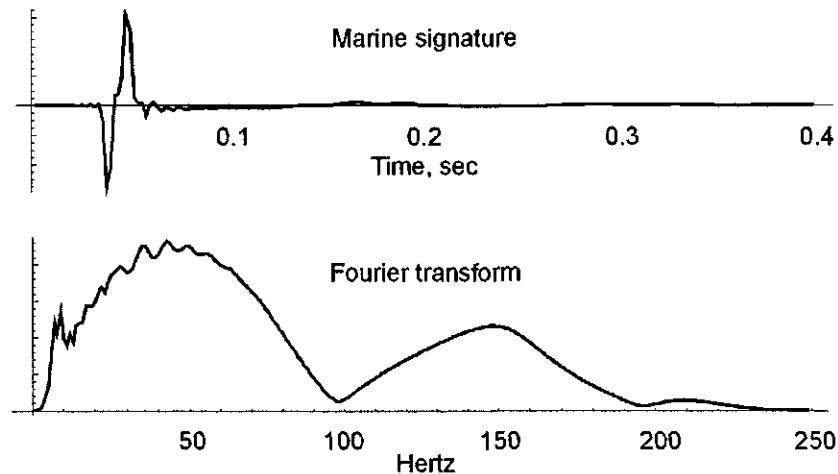


Figure 7.13: A typical marine near-field seismic signature, courtesy of Joe Reilly, Exxon Mobil. `MarineSourceAnalysis.nb` \Rightarrow `MarineSources.cdr` \Rightarrow `MarineSource.wmf`

and variable coupling.

2. The marine seismic source generates an impulsive signature of extremely short duration and excellent repeatability as shown in Figure 7.13.

The marine seismic source gun array signature consists of

1. The primary gun energy.
2. Bubble pulse.
3. Ghost reflected from the water surface.

These contributions all add to the total signature, which is about 500 ms; however, the largest contribution comes from the primary and is only about 25 ms in duration with a shape that resembles an integrated Klauder wavelet (see Figure 5.26 on page 237). Statistical gapped predictive deconvolution (Section 8.3.2) has been extremely effective in reducing short-period multiple energy associated with a shallow water bottom, and in removing residual bubble pulse energy. Reilly and Hobbs [143] in a report on the evaluation of source signature estimation technology provided excellent illustrations of marine signatures.

7.2 Mathematical Descriptions of Wavelets

Two wavelets commonly referenced in the literature are the Ricker wavelet and the Klauder wavelet. The autocorrelations of these wavelets are quite sim-

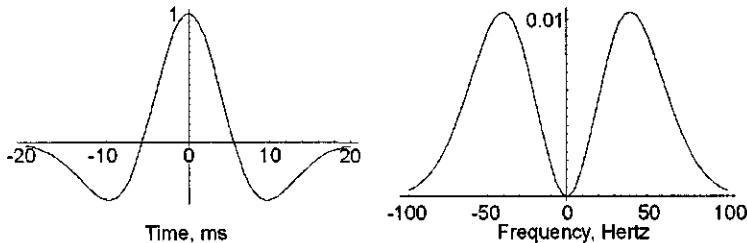


Figure 7.14: The classic Ricker wavelet transform pair with a peak frequency of $\nu_0 = 50$ Hertz. A plot of the transform pair of Equation (7.2). Left: Time domain. Right: Frequency domain. `RickerWavelet.nb` \Rightarrow `Ricker.cdr` \Rightarrow `ricker.wmf`

ilar [60] and we would therefore expect similar results from a deconvolution filter designed from autocorrelation coefficients (Section 8.3.2).

7.2.1 Ricker wavelet

An early attempt to quantify the shape of the seismic wavelet was given by Ricker [144] in a classic paper describing the wavelet shape that now bears his name. The Ricker wavelet bears some resemblance to an actual physical seismic wavelet but is actually the second derivative of the error function. The Fourier transform pair for the Ricker wavelet $f(t)$ is

$$f(t) = [1 - 2\pi^2\nu_0^2t^2] e^{-\pi^2\nu_0^2t^2} \Leftrightarrow \left[\frac{2}{\sqrt{\pi}} \right] \left[\frac{\nu^2}{\nu_0^3} \right] e^{-\nu^2/\nu_0^2} = F(\nu) \quad (7.2)$$

and is shown in Figure 7.14 for a peak frequency ν_0 of 50 Hertz. See also Lindsey [112]. The effect of absorption and dispersion on a Ricker wavelet is shown in Figure 8.87.

7.2.2 Klauder wavelet

The vibroseis pilot sweep is “cross-correlated” with the recorded data in order to reduce the time duration of the sweep. This process can be described as sliding (without folding) the pilot sweep along the recorded seismic trace. Whenever the pilot sweep “sees” a reflected version of itself on the recorded seismic trace, a high degree of correlation (positive or negative) results, which corresponds to a relatively high value of a correlation coefficient. For example, where the pilot signal is exactly superimposed over a returned reflection, the correlation coefficient would be, say, +1. Where the pilot signal is exactly superimposed over a reversed-polarity version of itself, the correlation coefficient would be -1. In between these exact superpositions, the correlation will be associated with a correspondingly lower correlation coefficient. This process of the pilot signal seeking an image of itself in the recorded data is called “vibroseis

crosscorrelation" of the known pilot sweep with the recorded data. It is easily accomplished on a computer. If only one reflected pilot sweep were recorded, the cross-correlation would be called an *autocorrelation*. The autocorrelation reaches a maximum value when the pilot is exactly superimposed over the reflection because all products are positive and the sum therefore reaches a large positive value. As the pilot is slid past its image, the correlation is quickly degraded because negative products are introduced into the sum of products, and the value of the correlation coefficient becomes lower. The correlation drops off rapidly as the pilot sweep shape moves along the recorded trace, defining a correlation function called a "Klauder wavelet", one for each reflected vibroseis sweep. Thus, after crosscorrelation, the seismic trace is no longer a direct measure of ground velocity, but is actually a "crosscorrelogram", with high trace values corresponding to peaks (or troughs, in the case of negative reflection coefficients) in the Klauder wavelets. Crosscorrelation of the pilot sweep with itself (autocorrelation) considerably reduces the effect time duration of the pilot sweep. The vibroseis "source wavelet" becomes the Klauder wavelet and the recorded trace is now something that can be interpreted. The Klauder wavelet can be defined as

$$K(t) = \frac{\sin[(L-t)B] \cos(Pl)}{BL}$$

where $P = \pi(\nu_1 + \nu_2)$, $A = \pi(\nu_2 - \nu_1)/L$, and $B = At$. The frequencies ν_1 and ν_2 are, respectively, the start and end of the vibroseis sweep. The duration of the sweep is L sec. The recording parameters used by Virginia Tech for deep crustal exploration were $\nu_1 = 14$, $\nu_2 = 56$, and $L = 24$ sec.

Mathematically, the process of crosscorrelation of the pilot sweep $p(t)$ with the recorded trace $s(t)$ is given by, for continuous functions,

$$\Phi_{ps}(\tau) = \int_{-\infty}^{\infty} s(t) p(t + \tau) dt$$

so that the resultant trace is a measure of correlation rather than some kind of ground motion.

The autocorrelation of the sweep is called the *Klauder wavelet* [95, 76] and is given by the above equation or alternatively

$$f(t) = \frac{\sin[\pi\{(\nu_2 - \nu_1)/L\}(L-t)t] \cos[\pi(\nu_1 + \nu_2)t]}{\pi(\nu_2 - \nu_1)Lt}$$

where $\nu = [(\nu_2 - \nu_1)/T]t$ and T is the sweep length in seconds.

Multiple vibrators are commonly used, primarily to inject more energy into the earth and improve the signal-to-noise ratio. For the acquisition of shallow, high-resolution seismic data, a single vibrator might be preferred. Many comparisons between single and multiple vibrator sources have been made. The depth of penetration of the energy from a single vibrator depends primarily upon the characteristics of the near-surface in the area of study. Factors such as vibrator weight, number of vibrators, ambient noise, and sweep length do,

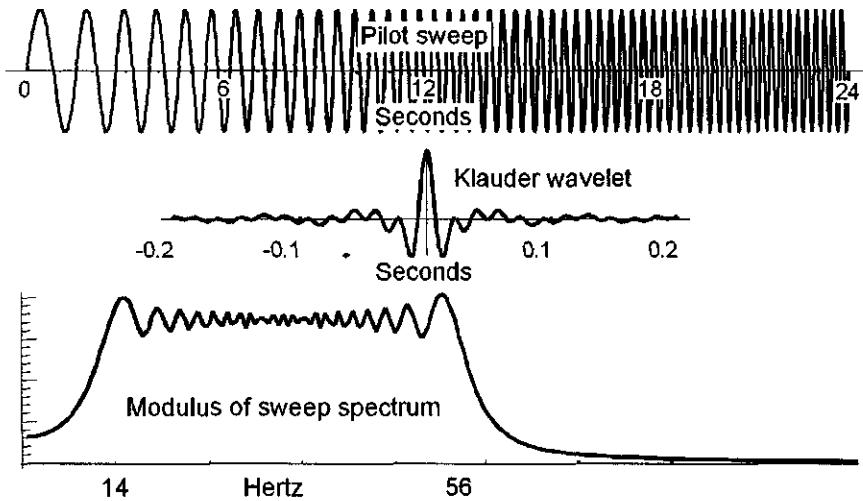


Figure 7.15: The Klauder wavelet is the autocorrelation of the vibroseis sweep. Top: pilot sweep. Middle: autocorrelation of pilot sweep. Bottom: Fourier transform of pilot sweep. (klauder)

of course, enter into the quality of the final record section; however, impressive results have been obtained from shallow and intermediate crustal depths using a single large vibrator.

The vibroseis pilot sweep commonly used by the Virginia Tech seismic crew for deep crustal research in the Appalachians is shown in Figure 7.15. The autocorrelation of the sweep is shown below the sweep. The process of autocorrelation shortens the sweep; it is the autocorrelation of the sweep that is actually viewed on a vibroseis record, and not a source wavelet. The untapered, linear pilot sweep is simply

$$p(t) = \sin(2\pi\nu t)$$

where

$$\nu = \nu_1 + \frac{\nu_2 - \nu_1}{2L} t$$

L is the length of the sweep and ν_1 and ν_2 are the starting and ending frequencies, respectively. We used a tapered sweep, tapered on each end by 1 sec to minimize side lobes.

7.2.3 Comments

There is no magic formula for the choice of an energy source or the design of a survey—on land at least—but there is plenty of guidance. See especially

Volumes 3 (Seismic Field Techniques) and 7 (Seismic Noise Attenuation) in this Handbook of Geophysical Exploration and Evans [70]. Data quality depends on many factors [113]. So much more than the energy source is involved, as implied by Figures 7.3 and 7.12. How large should the charge be? How many vibrators? How many sweeps? What duration? How many pulses? “How many thumps would a thumper thump if a thumper could thump thumps?” [107]. In rugged terrain or in areas where the roads are crooked (i.e., the Appalachians) it is not possible to use vibrators along straight lines for long distances. On the other hand using helicopters and explosives to obtain straight-line coverage on land and avoid crooked-line processing can be prohibitively expensive, but this is commonly done and with good reason by the petroleum industry. Implicit in the cost of data acquisition are concerns like having to re-acquire the data because of initial poor quality, the time spent in interpreting poor data, and so forth. On land at least, one clear example of the value of doing it right simply and early with a seismic line(s) is the famous Brevard Zone in the southeastern United States. This linear structural feature extends for over 400 km from Georgia to Virginia and can be seen from orbit. Over twenty interpretations [22] had been made about the tectonic significance of this feature before seismic lines acquired perpendicular to the lineament eliminated all but one and showed that the zone is simply the surface expression of an eastward-dipping thrust fault similar (in geometry) to those observed in the folded Appalachians to the west [41, 51].

As any wavelet propagates away from the energy source many effects act to change its frequency content and therefore its shape. Some of the more interesting of these effects are reflection from a thin bed, which returns the derivative of the incident wavelet, and reflection from a geologic interval within which the velocity changes linearly with depth, which returns the integral of the source wavelet. In addition, energy loss due to intrinsic damping (absorption) continuously changes the frequency content of the wavelet along its propagation path. These effects are discussed in more detail in Sections 5.7 and 8.5.2.

7.3 Wavelet z -transform representation

An approximation to a wavelet generated from an impulsive source is shown in Figure 7.16. The sampling interval is two milliseconds. A frequency spectrum and a pole-zero plot are shown in the same figure. The following minimum-delay wavelet coefficients are used for many of the numerical computations in this volume.

```
wavelet={0.132984,0.342512,0.548545,0.744799,0.907561,1.04204,1.12834,1.16132,1.14894,
1.07934,0.981866,0.870631,0.753987,0.641438,0.534492,0.422551,0.300063,0.196714,
0.099406,-0.0225922,-0.125436,-0.215479,-0.306348,-0.384165,-0.45418,-0.516001,
-0.557161,-0.584517,-0.589772,-0.579596,-0.54334,-0.492505,-0.425367,-0.349645,
-0.265986,-0.181963,-0.100738,-0.0363371,0.0281819,0.0917981,0.151973,0.195556,
0.232337,0.257646,0.271141,0.277681,0.272208,0.267031,0.250813,0.217928,0.171922,
0.119909,0.0633082,0.0171123,-0.03006,-0.0776557,-0.12405,-0.153802,-0.168138,
-0.166966,-0.155284,-0.136898,-0.113179,-0.0910421,-0.0715249,-0.0540389,-0.0337813,
-0.0130044,-0.000170491,0.010792,0.0176374,0.0211389,0.0197672,0.0179044,0.0211134,
0.0206213,0.0152477,0.0159675,0.0136121,0.0123929,0.00978807,0.00622238,-0.00373036,
-0.00802724,-0.0172832,-0.0264725,-0.0327029,-0.033411,-0.0401914,-0.0381475,
-0.0384762,-0.0348436,-0.0293412,-0.0217437,-0.0167644,-0.00920205,-0.00332008,
0.00399458,0.00993271,0.0162817,0.0222198,0.0267815,0.0290488,0.0287685,0.0299113,
```

```
0.0301194, 0.0297948, 0.0272967, 0.0234607, 0.018213, 0.0125574, 0.00656805, 0.000936034};
```

The roots of the wavelet are shown in Figure 7.16 and were determined using the root finder *Mathematica* module `RootsFrom` as listed in [152]. The amplitude spectrum was computed using *Mathematica's Fourier*.

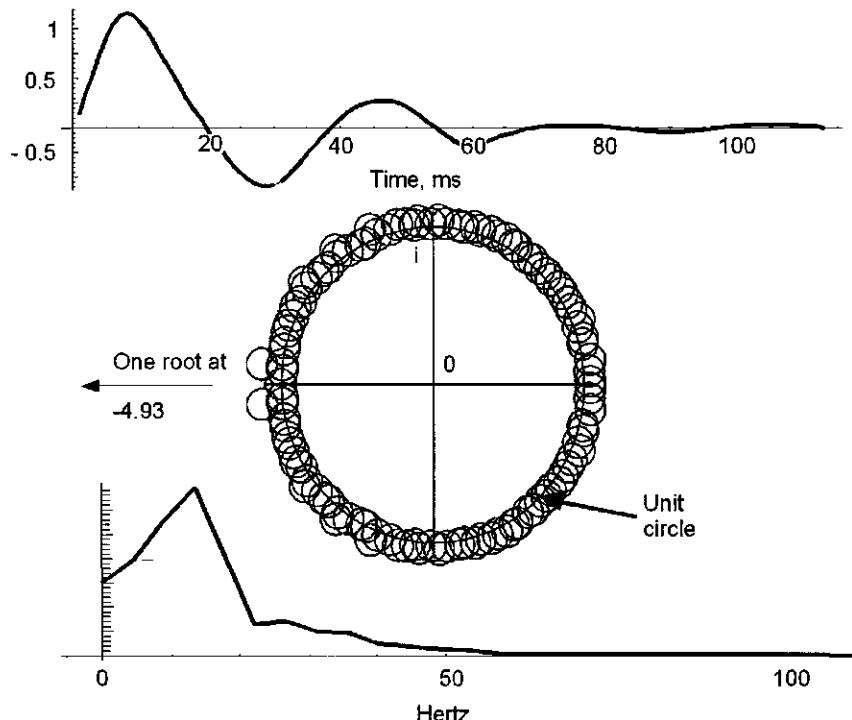


Figure 7.16: Top: Minimum-delay wavelet. Sampling interval is 2 milliseconds. Center: A pole-zero plot of the roots. Bottom: Frequency content of the wavelet. This is not a realizable physical wavelet, i.e., from an explosive source, because its mean value is not zero. `MinimumDelayWavelet.nb` \Rightarrow `MinimumDelayWavelet.cdr` \Rightarrow `MinimumDelayWavelet.wmf`

7.3.1 Physical requirements for real wavelets

For the wavelet mean to be zero (a necessary condition for a real, physical wavelet) you must have at least one of the roots on the unit circle; i.e., one of the roots z_i must be $z_i = 1$ so that the couplet $(-z_i, 1)$ becomes $(-1, 1)$. This will be but one of the many 2-term factors in the z -transform of the wavelet to be constructed. If $z_1 = 1$ then $(-z_1, 1)$ becomes $(-1, 1)$, which is recognized

from Chapter 5 as a differencing operator. The convolution of

$$(-1, 1)$$

with

$$(-z_2, 1) * (-z_3, 1) \dots (-z_n, 1)$$

whatever that polynomial is, removes the zero-frequency “dc” (constant) component where “ $*$ ” denotes convolution. Thus, the z -transform of the wavelet we are building becomes

$$\begin{aligned} W(z) &= (-z_1 + z)(-z_2 + z)(-z_3 + z) \dots (-z_n + z) \\ &= (-1 + z)(-z_2 + z)(-z_3 + z) \dots (-z_n + z) \end{aligned}$$

For the inverse z -transform

$$w(t) = (-1, 1) * (-z_2, 1) * (-z_3, 1) \dots * \dots (-z_n, 1)$$

Also, we want to avoid aliasing so if we choose one of the roots z_2 , say, to be $z_2 = -1$ then the couplet $(-z_2, 1)$ becomes $(1, 1)$ and its z -transform is $1 + z$. We can see why the frequency content of the wavelet will be zero at the Nyquist frequency by making a substitution for the definition of z in the z -transform $1 + z$:

$$\begin{aligned} 1 + z &= 1 + e^{-i2\pi\nu\Delta t} \\ &= 1 + e^{-i2\pi(\frac{1}{2\Delta t})\Delta t} \\ &= 1 + e^{-i\pi} = 1 - 1 \\ &= 0 \end{aligned}$$

where the time interval between the two points in the couplet is the sampling interval Δt and the folding frequency has been substituted for the frequency, ν . This is the second root of the many 2-term factors that, when convolved together, will produce the wavelet. Thus, the z -transform of the wavelet we are building now becomes

$$\begin{aligned} W(z) &= (-z_1 + z)(-z_2 + z)(-z_3 + z) \dots (-z_n + z) \\ &= (-1 + z)(1 + z)(-z_3 + z) \dots (-z_n + z) \end{aligned}$$

and for its inverse z -transform

$$w(t) = (-1, 1) * (1, 1) * (-z_3, 1) \dots * \dots (-z_n, 1)$$

It remains to select the other roots keeping in mind the delay characteristics of the desired wavelet. Neidell [122] suggested that the seismic wavelet after processing is relatively simple. His heuristic analysis indicated that the roots of seismic wavelets are essentially all on the unit circle and the real axis and that these attributes seem to characterize seismic wavelets rather well.

7.3.2 A simple 2-point wavelet

Suppose p_0 and p_1 are the only two sampled points in a time series $p(t)$. Then, from Equation (3.133), the z -transform $P(z)$ of the continuous-time function $p(t)$ is

$$P(z) = p_0 z^0 + p_1 z^1 = p_0 + p_1 z$$

Such a two-term time series can be referred to as a *couplet*, and the z -transform of the time series is a z -transform *binomial*.

We might define two more z -transforms as

$$Q(z) = q_0 z^0 + q_1 z^1 = q_0 + q_1 z$$

and a third z -transform as the ratio of these two:

$$w(z) = \frac{P(z)}{Q(z)} = \frac{p_0 + p_1 z}{q_0 + q_1 z} \quad (7.3)$$

$w(z)$ is a mapping function that transforms from the complex z -plane into the complex w -plane. The coefficients of one time series are p_0 and p_1 ; q_0 and q_1 are the coefficients of the other time series. The coefficients p_0, p_1, q_0 , and q_1 can be zero, real, imaginary, or complex.

The properties of binomials and the ratios of two binomials will be examined. Accelerated interest (because of computers) in the field of iterative function systems (IFSs) and fractals relies heavily on ratios defined by Equation (7.3), known as the Möbius transformation. It is one of the most important classes of conformal transformations (mapping). Equation (7.3) is also known as the *bilinear*, or *linear fractional* transformation.

Thus, for many reasons, the mathematical behavior of the “binomial” and the ratio of two binomials in the complex z -plane is important.

In Equation (7.3), we must have $p_1 q_0 - p_0 q_1 \neq 0$ because

$$\frac{dw}{dz} = \frac{p_1 q_0 - p_0 q_1}{(q_1 z + q_0)^2}$$

and therefore if $p_1 q_0 = p_0 q_1$, $dW/dz = 0$, and every point of the z -plane is a *critical point* (a critical point is a point in the z -plane where $dW/dz = 0$).

Note that if $p_1 q_0 = p_0 q_1$ then

$$\frac{p_1}{q_1} = \frac{p_0}{q_0}$$

and the numerator and denominator of Equation (7.3) are proportional to each other. As a result w is independent of z and the entire z -plane is mapped (transformed) into the same point in the w -plane.

The simplest “wavelet” is a two-point “binomial” such as shown in the right-hand side of Figure 7.17. Examples of binomials are listed in Table 7.1, and their z -transforms are plotted in Figure 7.18. These are values that, for example, the constants p_0, p_1, q_0, q_1 , or a, b, c , or d might take in Equation (7.3) (as long as $p_1 q_0 - p_0 q_1 \neq 0$). Either coefficient of any binomial can be pure real, pure imaginary, or complex. The coefficients of these are plotted in Figure 7.18 to emphasize that they can be real, imaginary, or complex.

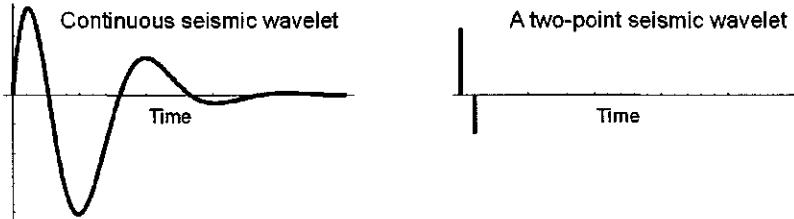


Figure 7.17: Left: A continuous seismic wavelet decays to zero as $t \rightarrow \infty$. Right: A simple two-point “wavelet”, used to illustrate the general properties of more complex wavelets such as the one shown on the left. Many of these 2-point “wavelets” convolved together make up an approximation to the wavelet shown on the left. (wave2pt.nb → cd → wave2pt.wmf)

Table 7.1: Examples of 2-term wavelets

The coefficients are plotted in Figure 7.18		
Binomial number	First coefficient p_0	Second coefficient p_1
1	$-1 + i0$	$-0.5 + i0$
2	$1 + i0$	$2 + i0$
3	$0 + i1$	$0.5 + i0$
4	$1.5 + i1.9$	$1 + i0.5$
5	$1.2 + i0.9$	$0 - i0.2$
6	$1.7 + i0$	$-1.5 + i0$
7	$1.5 + i1.5$	$-1.7 - i1.5$

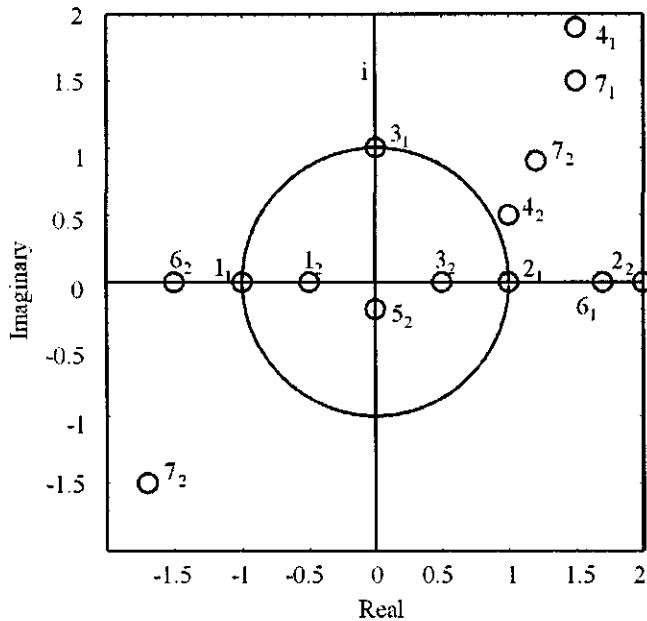


Figure 7.18: The modulus of the complex number $e^{-i\omega}$ is 1 = radius of unit circle. The numbers beside the plotted points refer to the binomial and the first or second coefficient listed in Table 1. Notation 5_2 refers to the 2nd coefficient of the 5th binomial. (DipoleExamples.nb,cdr → DipoleExamples.wmf)

7.3.3 Generation of wavelets

The convolution of binomials of the form $(-z_i, 1)$ results in a wavelet whose delay properties depend on the delay properties of the individual binomials. Thus, if z_1 and z_2 are outside of the unit circle, the 3-term wavelet that results from the convolution

$$(-z_1, 1) * (-z_2, 1)$$

will be minimum-delay (minimum-phase in the z-transform domain).

If z_1 and z_2 are inside the unit circle, then the 3-term wavelet that results from the convolution

$$(-z_1, 1) * (-z_2, 1)$$

will be maximum-delay (maximum phase). If one root is inside and one outside of the unit circle, the 3-term wavelet that results from the convolution

$$(-z_1, 1) * (-z_2, 1)$$

will be mixed-delay (mixed-phase in the z -transform domain). If a binomial has a root somewhere *on* the unit circle then the binomial is called “equi-delay”. It is neither minimum-delay nor maximum-delay. If z_1 and z_2 form a complex conjugate pair of roots then the associated wavelet is real. If not, the wavelet will be complex in the time domain. Examples are shown in Table 7.2.

In general,

$$\begin{aligned} F(z) &= (-z_1 + z)(-z_2 + z) = (-z_1)(-z_2) + (-z_1 - z_2)z + z^2 \\ &= (-z_1)(-z_2)z^0 + (-z_1 - z_2)z^1 + z^2 \end{aligned} \quad (7.4)$$

If $z_2 = -z_1$ then Equation (7.4) becomes

$$\begin{aligned} F(z) &= (-z_1 + z)(+z_1 + z) = (-z_1)(+z_1) + 0z + z^2 \\ &= (-z_1)(+z_1)z^0 + 0z^1 + z^2 \end{aligned}$$

and the moduli of the time-domain coefficients have values

$$|z_1|^2, 0, 1$$

with the coefficient for $t = \Delta t$ equal to zero.

The modulus of the complex convolution of any two equi-delay 2-term wavelets has even symmetry with respect to the second coefficient of the convolution. The equi-delay binomial is not of physical significance with regard to the generation or examination of real seismic wavelets; it never decays as a function of time because each of its coefficients has the same value.

Note (Table 7.2 that, for a wavelet greater than two points in length it is not possible to determine its delay (or phase) characteristics by simple examination of the relative magnitudes of the coefficients. *The first coefficient of a 3-point minimum-delay wavelet is not necessarily the largest coefficient. The last coefficient of a 3-point maximum-delay wavelet is not necessarily the largest coefficient.*

It follows that convolution of m binomials will result in a wavelet of length $m + 1$. If all m binomials are minimum delay then the resulting wavelet will be minimum delay. If m binomials are minimum delay and n are maximum delay then the mixed-delay convolution will be of length $m+n-1$. If n maximum-delay binomials are convolved together then the resulting wavelet will be maximum delay and of length $n + 1$.

In general, an individual 2-term “wavelet” can be complex. Therefore, the convolution must be done with a complex version of, for example, SUBROUTINE FOLD.

Table 7.2: Examples of the convolution of two binomials

The binomials are $(-z_1, 1)$ and $(-z_2, 1)$						
$ z_1 $	θ_1 deg	Real and imag parts of $-z_1$	$ z_2 $	θ_2 deg	Real and imag parts of $-z_2$	Complex con- volution of binomials
1.3	135	$0.919 - i0.919$	1.3	-135	$0.919 + i0.919$	1.69, 1.84, 1
1.3	90	$0 - i1.3$	1.3	-90	$0 + i1.3$	1.69, 0, 1
1.3	0	$-1.3 + i0$	1.3	0	$-1.3 + i0$	1.69, -2.6, 1
1.3	0	$-1.3 + i0$	1.3	180	$1.3 + i0$	-1.69, 0, 1
1.3	90	$0 - i1.3$	1.3	90	$0 - i1.3$	-1.69, -i2.6, 1
1	0	-1	1	180	1	-1, 0, 1
0.7	0	$-0.7 + i0$	0.7	180	$0.7 + i0$	-0.49, 0, 1
0.7	90	$0 - i0.7$	0.7	-90	$0 + i0.7$	0.49, 0, 1
0.9	130	$0.579 - i0.689$	0.9	-130	$0.579 + i0.689$	0.81, 1.158, 1
0.7	130	$0.45 - i0.536$	0.7	130	$0.45 - i0.536$	-0.085 - i0.48, 0.9 - i1.07, 1
1.3	130	$0.836 - i0.996$	0.7	130	$0.45 - i0.536$	-0.1577 - i0.896, 1.286 - i1.532, 1
1.3	130	$0.836 - i0.996$	0.7	-130	$0.45 + i0.536$	0.91, 1.29 - i0.46, 1

If roots z_i occur in complex conjugate pairs then as each binomial is successively convolved with the preceding convolution the intermediate convolutions may be complex; however, after the last binomial has been convolved the coefficients of the resulting wavelet in the time domain must be real. The following *Mathematica* program can be used to convolve together 2-term wavelets (couplets) of the form $(-z_i, 1)$ where z_i can be real, complex, or imaginary; however, if a root z_i is chosen to be complex then another must be included that is its complex conjugate in order for the convolution to result in a pure real wavelet.

```
(* Subroutine converted from Fortran of Robinson, 1967,
   Multichannel Time Series Analysis With Digital
   Computer Programs *)
fold[a_, b_] := lc = Length[a] + Length[b] - 1;
c = Table[0, {lc}];
Do[Do[k = i + j - 1;
  c[[k]] = c[[k]] + a[[i]]*b[[j]], {j, 1, Length[b]},
  {i, 1, Length[a]}];
(*-----End of subroutine-----*)
a = {1, 1, 1}; b = {1, 1, 1};
fold[a, b];
c
(*-----End of output-----*)
```

7.3.4 Partial energy of a wavelet

The continuous autocorrelation function $\phi(\tau)$ is defined as

$$\phi(\tau) = \int_{-\infty}^{+\infty} w(\tau)w(t+\tau)d\tau$$

The discrete formulation is

$$\phi(\tau) = \sum_{-\infty}^{+\infty} w(\tau)w(t+\tau)$$

Given four wavelets:

- A: (4, 0, -1)
- B: (2, 3, -2)
- C: (-2, 3, 2)
- D: (-1, 0, 4)

Each of these wavelets has the same autocorrelation function but different delay properties. For wavelet A the autocorrelation function $\phi(\tau)$ is

$$\begin{aligned}\phi(-2) &= (4)(0) + (0)(0) + (-1)(4) = -4 \\ \phi(-1) &= (4)(0) + (4)(0) + (-1)(0) + (-1)(0) = 0 \\ \phi(0) &= 4^2 + 0^2 + -1^2 = 17 \\ \phi(+1) &= (4)(0) + (4)(0) + (-1)(0) + (-1)(0) = 0 \\ \phi(+2) &= (4)(0) + (0)(0) + (-1)(4) = -4\end{aligned}$$

In the frequency domain, the condition that each wavelet in the set have the same autocorrelation function becomes the condition that each wavelet in the set has the same amplitude spectrum. This follows because the Fourier transform of the autocorrelation is the power spectrum. *So each of the above wavelets has the same amplitude spectrum, but each has a different phase spectrum.*

The “cumulative energy buildup” [150, p. 112-113]

$$(a_0^2, a_0^2 + a_1^2, a_0^2 + a_1^2 + a_2^2)$$

of each of these wavelets is

- A: (16, 16, 17)
- B: (4, 13, 17)
- C: (4, 13, 17)
- D: (1, 1, 17)

Wavelet “A” has the quickest energy buildup. This is the “minimum-delay” wavelet. The energy delay of wavelet A is a minimum compared to that of any other wavelet in the above set. The above wavelets make up a set because they all have the same autocorrelation function. The maximum delay wavelet has the

slowest energy buildup. Most of its energy is concentrated at the end (wavelet "D"). Wavelets "B" and "C" are mixed delay.

The minimum-delay condition becomes the condition that the phase-lag spectrum is a minimum. Thus the term "minimum-energy-delay" is equivalent to "minimum-phase-lag." (The phase-lag spectrum is the negative of the phase spectrum.)

7.3.5 Roots plotted in the z-plane

An example of a root z_i plotted in the complex z -plane is shown in the left-hand figure of Figure 7.19. The z -transform of the (complex) time-domain binomial

$$(-z_1, 1)$$

is

$$-z_1 z^0 + 1 z^1 = -z_1 + z$$

If another root, say z_2 , is the complex conjugate of z_1 then convolution of the coefficients of the two binomials

$$(-z_1, 1) \text{ and } (-z_2, 1)$$

will result in a pure real time domain function.

If the coefficients a_0, a_1, \dots, a_m are real then the roots must appear in complex conjugate pairs, as shown in Figure 7.19.

The z -transform polynomial is obtained by convolving the (complex time) binomial $(-z_1, 1)$ with the (complex time) binomial $(-z_2, 1)$. The result is the real time series $(1.690, -2.2252, 1)$. The z -transform of this time series is

$$\begin{aligned} P(z) &= 1.690z^0 - 2.252z^1 + 1z^2 \\ &= 1.690 - 2.252z + z^2 \end{aligned}$$

The roots (zeros) of this polynomial are z_1 and $z_2 (= z_1^*)$ and are shown plotted in Figure 7.19. It is useful to plot the roots of the polynomial

$$a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \dots + a_m z^m$$

in the complex z -plane.

From the definition of minimum and maximum delay functions it is apparent that if all the zeros, or roots, of the z -transform polynomial lie outside of the unit circle in the z -plane, then the function $(a_0, a_1, a_2, \dots, a_m)$ is minimum delay. If some of the roots lie inside the unit circle, then the function is mixed delay. If all of the roots lie inside the unit circle, the function is maximum delay.

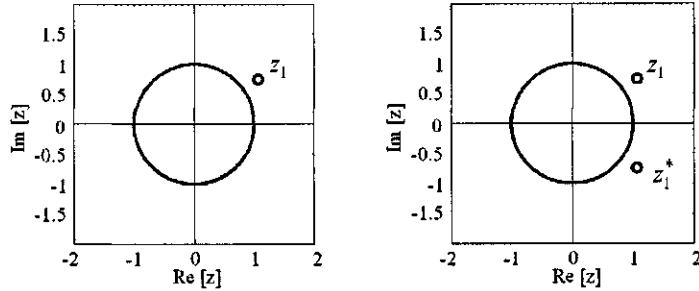


Figure 7.19: Single root z_1 , left, and together with its complex conjugate, right. Circles show locations of complex roots $z_1 = re^{i\theta}$. Modulus $r = 1.3$, $\theta = 35$ deg. The corresponding (real) time-domain binomial is $(1.690, -2.2252, 1)$. Roots1AndRoots2.nb → Roots1AndRoots2.cdr → Roots1A.wmf)

7.3.6 Root on the unit circle

If the root z_1 of the z -transform $P(z) = -z_1 + z$ is on the unit circle then

$$|z_1| = 1$$

A complex root on the unit circle is shown in Figure 7.20. A root on the unit circle means that the time-domain 2-point wavelet is $(1,1)$. That is, the wavelet does not decay with time –the value at $t = 0$ is 1 and the value at the later time $t = \Delta t$ is still 1. It is neither minimum-delay nor maximum delay, and because it does not decay with time this simple 2-point wavelet is not “physically realizable”.

The concept of the distribution of energy in a wavelet as a function of time will be explored further when wavelets longer than two sample points are discussed; however, generalizations that hold for wavelets of any length can already be stated from our investigation of 2-point wavelets:

Given the two-point wavelet $(-z_1, 1)$ where $|z_1| = 1.3$, the squared amplitudes are:

$$(1.3 \times 1.3, 1 \times 1) = (1.69, 1)$$

so the energy decreases with time. If the coefficient $-z_1$ is real, then this is the kind of binomial that is physically realizable.

For the maximum-delay wavelet $(1, -z_1)$ where $|-z_1| = 1.3$,

$$(1 \times 1, 1.3 \times 1.3) = (1, 1.69)$$

Even if the generally complex number z_1 had only a real part, this would not be a physically realizable wavelet because its energy increases with time.

For the wavelet $(-z_1, 1)$ where $|-z_1| = 1$, i.e., the root is on the unit circle, the wavelet energy does not decay at all:

$$(1 \times 1, 1 \times 1) = (1, 1)$$

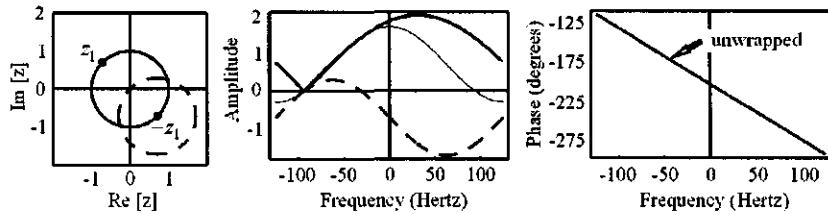


Figure 7.20: A root $|z_1|=1$ on the unit circle (left) together with the frequency and phase response of the z -transform (middle and right). A root on the unit circle means that the time-domain binomial wavelet is $(-z_1, 1)$. That is, the wavelet does not decay with time because $|-z_1|=1$. The wavelet is neither minimum-delay nor maximum delay, and because it does not decay with time it is not physically realizable. The z -transform of such a wavelet is, as usual, $P(z) = -z_1 z^0 + 1 z^1 = -z_1 + z$. (RootOnUnitCircle.nb,cdr,wmf)

Such a wavelet is not physically realizable.

The above conclusions, with a little help using the concept of the partial wavelet energy, can be extended to wavelets of any length.

Chapter 8

Wavelet Shaping and Deconvolution

In the review articles [35, 33] *Geophysics in the new millennium*, Cary [33] opened his discussion of seismic signal processing with comments that can serve as an introduction to this chapter:

In the introduction to his comprehensive SEG textbook *Seismic Data Processing*, Oz Yilmaz [204] selects deconvolution, common-midpoint stacking and migration as being the three principal processes that are applied during routine seismic processing. Since Yilmaz' tome was first published in 1987, a vast number of papers have been published and conference presentations have been given on virtually every aspect of seismic processing. However, I think it is still accurate to say that the same three processes dominate the processing flow of the vast majority of seismic data that is processed now, at the beginning of the twenty-first century.

The present volume is one about “basic theory” and as such might justifiably exclude more than a mere mention of data acquisition techniques that lead to that essential step of common-midpoint stacking. Furthermore, the many approaches to the migration of seismic data constitute more than “basic” theory. In any event migration is discussed extensively elsewhere in this Series. That leaves deconvolution, the first of Yilmaz’ three principal processes, and some basic theory of deconvolution is certainly appropriate for this volume.

The general terms “wavelet shaping” and “deconvolution” are associated with “inverse filtering” where the filter operator is designed to change the shape of the primary source wavelet. Usually, the objective is to shorten the duration of the wavelet, which for the case of a wave guide can be accomplished by removing reverberations. Or maybe it is desired to change the shape of a recorded mixed-delay wavelet to one that is minimum-delay but of the same duration. Some form of wavelet shaping is one of the most common steps in

seismic data processing. Predictive deconvolution is a form of wavelet shaping commonly used to remove reverberations “attached” to the primary source wavelet. Removal of the reverberations shortens the length of the wavelet.

All attempts to change the wavelet shape involve, conceptually at least, two distinct steps:

1. Convert the input to an impulse at $t = 0$,
2. Convolve the impulse with the desired output.

It is Step 1 above that requires a knowledge of the wavelet shape, because unless the delay properties (minimum, maximum, or mixed) of the wavelet are known, the inverse filter cannot be properly designed; i.e., it won't converge to zero as the filter length increases. The delay properties of the desired output are not important and so Step 2 above is always just a simple convolution. Of course, the wavelet shape is, in general, not known. It is instructive, however, to shape wavelets and deconvolve under the assumption that the shape is known, in which case filter design is referred to as a “deterministic” process. If the wavelet shape is not known then it can be estimated from the seismic trace itself, and filter design becomes a “statistical” process dependent upon the assumed statistical properties of the sequence of reflection coefficients as discussed on page 455. Predictive deconvolution is a deterministic process when applied to a known wavelet, but a statistical one when applied to a seismic trace.

In practice, the design of an inverse filter is not separated into two steps; however, doing so in a conceptual sense focuses on the conditions required for proper convergence of the inverse filter operator. The conditions, simply stated and elaborated upon below, are:

- If the input is minimum-delay, inverse filter coefficients are required only for times $t \geq 0$.
- If the input is maximum-delay, inverse filter coefficients are required only for times $t < 0$.
- If the input is mixed-delay, inverse filter coefficients are required for times negative, zero, and positive.

There are many ways to organize this chapter. Because of its importance predictive deconvolution might justify a separate chapter of its own instead of just a section in this one. Wave guides are an interesting phenomenon but the exact finite filters that remove reverberations so well from synthetic data are not easily implemented on real data. But these and other filtering methods do share a common thread that suggests they all belong in the same chapter. We chose the following organization because it seems to accommodate both deterministic and statistical methods of filter design. The above conditions on filter coefficients can be accommodated within the framework of three general subdivisions:

1. Inverse filters of infinite (or at least relatively long) length and input (a wavelet) of finite (relatively short) length. A seismic source wavelet, by definition, eventually decays to zero. That is, the numerical values of the coefficients that define the wavelet eventually become so small in value that for all practical purposes the wavelet becomes one of finite length. So must the filter, of course, but theoretically it would be of infinite length, so the longer the filter the better the approximation to the desired output. Examples include
 - (a) Wavelet shaping of minimum- and maximum-delay 2-point wavelets using Fourier transforms.
 - (b) Wavelet shaping of minimum- and maximum-delay 2-point wavelets using z -transforms.
 - (c) General wavelet shaping, including predictive deconvolution
 - (d) Inverse filtering of a seismic trace using z -transforms.
2. Inverse filters of finite length and input of infinite length
 - (a) Water-layer “Backus” and other wave guides. Removal of reverberations.
3. Inverse filters of finite length and input of finite length (the least-squares filters). The above two classes in practice revert to this class because of the practical requirement of implementing convolution on a computer. Examples include
 - (a) General wavelet shaping
 - (b) Predictive deconvolution

We will examine the properties of various kinds of filter-input combinations and see that a few simple conceptual ideas can provide simplification and understanding. In particular, the well-known “binomial expansion” is the keystone that leads to a better understanding of filter “stability”.

Suppose we wish to design an inverse digital filter that, after convolution with any arbitrary input, will convert the input to a unit impulse at $t = 0$. The input that we are commonly concerned with is the *seismic source wavelet*. If the time-duration of the seismic source wavelet can be decreased, then increased resolution in the recorded seismic trace will result. For perfectly noise-free data with the wavelet shaped to a unit impulse, the result of inverse filtering (or “deconvolution”) will be the *reflectivity function* itself.

The seismic trace $s(t)$ in its simplest form is the convolution of the reflectivity function $r(t)$ and the wavelet $w(t)$:

$$s(t) = \int_{-\infty}^{+\infty} r(\tau)w(t - \tau)d\tau$$

This is commonly referred to as the “convolutional model”. In the frequency domain, this is expressed as

$$S(\omega) = R(\omega)W(\omega)$$

Ideally (we will eventually abandon this unattainable goal –but see Figure 8.61) we wish to recover $R(\omega)$. If $w(t)$ is time-invariant and if we know what it looks like, and if $s(t)$ is noise-free, then $r(t)$ is easily obtained from

$$r(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{S(\omega)}{W(\omega)} e^{+i\omega t} d\omega$$

The function $r(t)$ is about as close as possible to the “geology” because it contains information about changes in the acoustic impedance of the subsurface as a function of reflection traveltime. If the density $\rho(z)$ is related to the velocity $v(z)$ by

$$\rho(z) = kv(z)^m \quad (8.1)$$

where z is depth, and k and m are constants, then $r(t)$ depends only on changes in rock velocity instead of acoustic impedance (Sengbush et al. [161]). Equation (8.1) holds very well with $k = 0.23$ and $m = 0.25$ for the sediments of the Gulf Coast.

The concept of the “ z -transform” was introduced on page 164 for a time series. Continuous-time functions are commonly described by Fourier and Laplace transforms. Discrete-time functions are described by z -transforms (pioneered by Zadeh Lofti). The introduction of the z -transform to design digital filters carries with it the terminology of “stability” and “instability” of the digital filter time series, concepts that we have not confronted when dealing with Fourier theory. One might with good reason ask why filter design could not be carried out using Fourier theory (using a subroutine like FT, for example) instead of z -transform theory. We will attempt to clarify why some methods are easier to use than others, and with fewer pitfalls.

Many if not all of the methods of “inverse filtering” used to change the shape of the wavelet are necessarily related because they are just approximations to exact filters. For example, the numerical values of the coefficients of an approximate least-squares filter of finite length must approach the numerical values of the coefficients of an exact filter of infinite length as the length of the filter is allowed to increase. A filter of infinite length is neither practical nor necessary but a discussion of a filter of infinite length does provide insight into filter design. We therefore begin with a discussion of exact inverse filters of infinite length that operate on input (the source wavelet, say) of finite length.

8.1 Inverse infinite filters, finite input

This is the first subdivision of deconvolution filters listed on page 372 and includes those inverse filters of infinite length and input of finite length. As we

turn from the Fourier transform to the z -transform, it is instructive to recall comparisons between the two approaches. The relevant equations are

$$f(t) = a_0 + 2 \sum_{n=1}^{f_{\max}} a_n \cos(2\pi \frac{n}{T} t) - 2 \sum_{n=1}^{f_{\max}} b_n \sin(2\pi \frac{n}{T} t) \quad (8.2)$$

$$a_n = \frac{1}{T} \int_0^T f(t) \cos(2\pi \frac{n}{T} t) dt \quad (8.3)$$

$$b_n = -\frac{1}{T} \int_0^T f(t) \sin(2\pi \frac{n}{T} t) dt \quad (8.4)$$

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

$$f(t) = \int_{-\infty}^{\infty} F(\nu) e^{+it2\pi\nu} d\nu \quad (8.5)$$

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt \quad (8.6)$$

$$F(\nu) = \int_{-\infty}^{+\infty} f(t) e^{-i2\pi\nu t} \quad (8.7)$$

where in general $f(t)$ and $F(\omega)$ are complex. Expanding Equation (8.5) and separating into real and imaginary parts:

$$f(t) = \int_{-\nu_{\max}}^{+\nu_{\max}} \operatorname{Re}[F(\nu)] \cos 2\pi\nu t d\nu - \int_{-\nu_{\max}}^{+\nu_{\max}} \operatorname{Im}[F(\nu)] \sin 2\pi\nu t d\nu \quad (8.8)$$

$$+ i \left(\int_{-\nu_{\max}}^{+\nu_{\max}} \operatorname{Im}[F(\nu)] \cos 2\pi\nu t d\nu + \int_{-\nu_{\max}}^{+\nu_{\max}} \operatorname{Re}[F(\nu)] \sin 2\pi\nu t d\nu \right) \quad (8.9)$$

which, for a pure real time function, becomes

$$f(t) = 2 \int_0^{\nu_{\max}} \operatorname{Re}[F(\nu)] \cos(2\pi\nu t) d\nu - 2 \int_0^{\nu_{\max}} \operatorname{Im}[F(\nu)] \sin(2\pi\nu t) d\nu \quad (8.10)$$

where

$$\operatorname{Re}[F(\nu)] = \int_0^T f(t) \cos(2\pi\nu t) dt \quad (8.11)$$

$$\operatorname{Im}[F(\nu)] = - \int_0^T f(t) \sin(2\pi\nu t) dt \quad (8.12)$$

The function $f(t)$ in Equation (8.5) can be either real or complex. Fortran Subroutine FT below from Robinson [148] can handle either case with equal ease:

```
SUBROUTINE FT(NX,X,W,C,S)
DIMENSION X(NX)
COSNW=1.0
SINNW=0.0
SINW=SIN(W)
COSW=COS(W)
S=0.0
C=0.0
DO 30 I=1,NX
C=C+COSNW*X(I)
S=S+SINNW*X(I)
T=COSW*COSNW-SINW*SINNW
SINNW=COSW*SINNW+SINW*COSNW
30 COSNW=T
RETURN
END
```

It is instructive to understand how to use this subroutine, or at least some code that is equally transparent, because if you do you will be able to predict how other, less open, computer programs should perform. Although this code isn't as fast as an elegant "fast Fourier transform" program, it does everything they do and it's plenty efficient enough on today's platforms for the user to experiment with a new idea or compare numerical output with other software. It is relatively speedy because it uses trigonometric recursion (the "T" stands for "temporary"). There are only two trigonometric lookups. Note that the sign convention is that used by engineers. To switch to the convention of mathematicians and that preferred in this volume, change $s=s+sinnw*x[[i]]$ to $s=s-sinnw*x[[i]]$.

A *Mathematica* version of Fortran FT together with numerical output for comparison with *Mathematica*'s Fourier is

```
-----Mathematica subroutine -----
FT[x_,w_,k_]:=(cosnw=1;sinnw=0;sinw=Sin[w];cosw=Cos[w];s=0;c=0;
Do[c=c+cosnw*x[[i]];s=s+sinnw*x[[i]];t=cosw*cosnw-sinw*sinnw;
sinnw=cosw*sinnw+sinw*cosnw;cosnw=t;CosTransform[[k]]=c;
SinTransform[[k]]=s,{i,1,Length[x]}];
-----End of Mathematica subroutine -----
-----Input data -----
x={1,2,3,4,5};
-----End of input data -----
w=2*Pi*freq*dt;dt=0.004;period=Length[x]*dt;
foldfreq=1/(2*dt);dfreq=1/period;n=Floor[foldfreq/dfreq+1];
CosTransform=Table[0,{n}]; SinTransform=Table[0,{n}];
Do[freq=(j-1)*dfreq;w=2*Pi*freq*dt;FT[x,w,j],{j,1,n}];
-----Compare with Mathematica's Fourier -----
```

```

FromFourier=Fourier[x]*Sqrt[Length[x]];
Print["Cosine transform from Subroutine FT      ",CosTransform];
Print["Cosine transform from Mathematica's Fourier ",
      Take[Re[FromFourier],Ceiling[Length[x]/2]]];
Print["Sine   transform from Subroutine FT      ",SinTransform];
Print["Sine   transform from Mathematica's Fourier ",
      Take[Im[FromFourier],Ceiling[Length[x]/2]]];
----- output -----
Cosine transform from Subroutine FT      {15, -2.5,-2.5}
Cosine transform from Mathematica's Fourier {15.,-2.5,-2.5}
Sine   transform from Subroutine FT      {0, -3.44095,-0.812299}
Sine   transform from Mathematica's Fourier {0.,-3.44095,-0.812299}
-----
```

To review briefly, if $f(t)$ is real then Equation (8.5) reduces to (8.10). The function $f(t)$ is commonly the seismic trace. The coefficients a_n and b_n in (8.3) and (8.4) are real in this case, and they possess an important symmetry in the frequency domain:

$$a_n(-\nu) = a_n(\nu) \quad (8.13)$$

and

$$b_n(-\nu) = -b_n(\nu) \quad (8.14)$$

where

$$\nu = \frac{n}{T}$$

The same symmetry, under the same assumptions, applies to z -transforms. *Recall that if the symmetry defined by Equations (8.13) and (8.14) is not present in the frequency domain before returning to the time domain, then $f(t)$ will be complex, not real.*

In SUBROUTINE FT, we would require one call to transform from the time domain to the frequency domain, and two calls to transform from the frequency domain to the time domain. Each call returns two integrations (one to the argument C and one to the argument S). For a forward transform of a pure real time function, we call C the "cosine transform" and S the "sine transform". Thus, for a pure real function $f(t)$ the forward transform (one call) returns (after multiplication by appropriate constants outside of the subroutine)

$$\begin{aligned} C &= \text{Re}[F(\nu)] = \int_0^T f(t) \cos(2\pi\nu t) dt \\ S &= \text{Im}[F(\nu)] = - \int_0^T f(t) \sin(2\pi\nu t) dt \end{aligned}$$

The negative sign for S can be added outside of the subroutine. The inverse transform (two calls) returns, for the first call,

$$C = \int_0^{\nu_{\max}} \text{Re}[F(\nu)] \cos(2\pi\nu t) d\nu$$

$$S = \int_0^{\nu_{max}} \operatorname{Re}[F(\nu)] \sin(2\pi\nu t) d\nu$$

and for the second call

$$\begin{aligned} C &= \int_0^{\nu_{max}} \operatorname{Im}[F(\nu)] \cos(2\pi\nu t) d\nu \\ S &= \int_0^{\nu_{max}} \operatorname{Im}[F(\nu)] \sin(2\pi\nu t) d\nu \end{aligned}$$

But from Equation (8.5)

$$\left[\int_{-\nu_{max}}^{+\nu_{max}} \operatorname{Im}[F(\nu)] \cos 2\pi\nu t d\nu + \int_{-\nu_{max}}^{+\nu_{max}} \operatorname{Re}[F(\nu)] \sin 2\pi\nu t d\nu \right] = 0$$

for a real-time function, and so, for the inverse transform, we ignore S from the first call and C returned from the second call because, from Equation (8.9), *they must sum to zero for a pure real time function.*

SUBROUTINE FT is not an efficient use of computer time if it yields more than we actually use. A Hartley transform pair defined as

$$\begin{aligned} H(\nu) &= \int_{-\infty}^{\infty} h(t) \cos(2\pi\nu t) + h(t) \sin(2\pi\nu t) dt \\ h(t) &= \int_{-\infty}^{\infty} H(\nu) \cos(2\pi\nu t) + H(\nu) \sin(2\pi\nu t) d\nu \end{aligned}$$

is computationally faster because for a real time function the unnecessary computations are not made. The Hartley transform produces real output for a real input. SUBROUTINE FT is completely general, however, and can be used for complex time as well as real time. The reader is encouraged to use FT, or a program like it, for all computations that use Fourier, Hilbert, or Hartley transforms. It is an excellent tutorial program that will generate the same numerical values as faster, but less transparent, programs such as NLOGN.

As we use z -transforms, we should continue to recall and compare their properties with Fourier transforms. We are still simply transforming from a (possibly) complex plane (time) to a (possibly) complex plane (frequency).

To repeat, all attempts to change wavelet shape involve, conceptually at least, two distinct steps: (1) Convert the input to an impulse at $t = 0$, and (2) Convolve the impulse with the desired output. In order to “spike out” a wavelet we need to examine the convergence properties of the inverse filters that can be designed. For a simple deconvolution, we want to design a filter such that the time-domain 2-term wavelet

$$(-z_1, 1)$$

(where the value of $-z_1$ occurs at $t = 0$) is deconvolved to

$$1, 0, 0, 0, 0, \dots$$

where the “1” occurs at $t = 0$.

The requirements for the design of filter coefficients required to convert a 2-term wavelet (see 7.3.2 on page 361) to a unit impulse at $t = 0$ can be simply stated:

- The inverse filter required to spike out a minimum-delay wavelet converges only for zero and positive time; i.e., *it must have filter coefficients defined only for zero and positive time*. This is stated without proof here, but is later shown to be the case by examination of Equation (8.20). Filter coefficients for negative values of time must be zero. Such a filter is said to have only a *memory* component.
- The inverse filter required to spike out a maximum-delay wavelet is defined (converges) only for negative time (not zero). Such a filter is said to contain only an *anticipation* component (Robinson and Treitel [153, page 105]).
- The inverse filter required to spike out a mixed-delay wavelet has filter coefficients defined for negative, zero, and positive times. Such a filter has both a memory component and an anticipation component.

From a Fourier transform point of view, $D(\omega)$, the transform of a two-term wavelet (a “couplet”) $d(t)$ is

$$D(\omega) = \int_{-\infty}^{+\infty} d(t)e^{-i\omega t} dt$$

where

$$d(t) \iff D(\omega)$$

We wish to design a filter $f(t)$ such that

$$\int_{-\infty}^{+\infty} f(\tau)d(t - \tau)d\tau = 1 \quad \text{at } t = 0$$

This is the simplest possible case—to convert two points in the time domain into the number “1” occurring at $t = 0$.

Convolution in the time domain is equivalent to multiplication in the frequency domain. Therefore,

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

Thus, the Fourier transform of the desired inverse filter $f(t)$ is

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{1}{d(\omega)} e^{+i\omega t} d\omega \quad (8.15)$$

It would be a simple matter to design a filter using, for example, SUBROUTINE FT or its *Mathematica* equivalent to determine $f(t)$ from Equation (8.15); however, in (8.15) or (8.5) the value of “ t ” can, in general, be negative, zero, or positive. According to our three guidelines (undeveloped at this point), however, we are restricted in the time domain by the delay properties of the wavelet whose inverse we desire. Thus, to spike out a minimum-delay couplet by convolution with an inverse filter, we evaluate the filter coefficients only at zero and positive time. To spike out a maximum-delay wavelet evaluate the coefficients only for values of $t < 0$. To spike out a mixed-delay wavelet evaluate the filter coefficients f_i for negative, zero, and positive values of time. If these rules are followed, the resulting convolution will always result in a spike at $t = 0$. Instability problems will never arise; the resulting inverse filter will not “blow up” and become infinitely large. The convolution will be stable and predictable.

Although Fourier theory (and SUBROUTINE FT) can be successfully used to spike out a wavelet of any arbitrary delay characteristics, it gives us no insight into *what* guidelines must be followed in order to obtain convergence; i.e., *at what times do we evaluate the inverse filter $f(t)$ in order that the filter will not become unstable*. To determine the convergence of a series of filter coefficients, we must take into account the delay properties of maximum- and minimum-delay wavelets and their inverses.

In z -transform notation and given the complex 2-term wavelet $f(t)$

$$f(t) = (-z_1, 1) \iff -z_1 + z$$

we wish to design an inverse filter $g(t)$. That is,

$$g(t) = \int_{-1/2\Delta t}^{+1/2\Delta t} \frac{1}{-z_1 + e^{-i2\pi\nu\Delta t}} e^{+i2\pi\nu t} d\nu \quad (8.16)$$

If $(-z_1, 1)$ is minimum delay, then Equation (8.16) is evaluated only for times $t \geq 0$. If $(-z_1, 1)$ is maximum delay, then Equation (8.16) is evaluated only for times, $t < 0$. If $f(t)$ is mixed delay, then Equation (8.16) is evaluated for times $-\infty < t < \infty$.

Each of the inverse filters will be of infinite length, and each will spike out the complex time-domain couplet at $t = 0$. Of course, Equation (8.16) holds whether the time-domain couplet is complex, real, or imaginary. The (complex) inverse filter $f(t)$ will be stable, and, when convolved with the couplet, will reshape it to “1” (real) at $t = 0$.

8.1.1 Inverse filtering of a minimum-delay 2-term wavelet using z -transforms

Determine the “inverse” filter $f(t)$ that will convert the two-term wavelet

$$(-z_1, 1)$$

to the number “1” at $t = 0$. z_1 is a complex number in the z -plane. That is, the z -transform $P(z)$ of the wavelet is

$$P(z) = -z_1 + z$$

We require the inverse z -transform of

$$\frac{1}{-z_1 + z}$$

because

$$1 \iff (-z_1 + z) \frac{1}{-z_1 + z}$$

Power series obtained by a successive differentiation process are called *Maclaurin series* or *Taylor series*[19, p. 22-23] about the origin. A Taylor series is a series of powers of $(z - z_1)$, where z_1 is some constant. For a given function, there is *only one convergent power series* of the form

$$\sum_{t=-\infty}^{t=+\infty} f_t z^t \equiv \sum_{n=-\infty}^{n=+\infty} f_n z^n$$

no matter how the series is obtained (that is, by using z -transform theory, Fourier theory, or predictive deconvolution). We wish to find these coefficients $f_t \equiv f_n$.

The Taylor series for “ $F(z)$ about z_1 ” (e.g., Boas [19, p. 22-24]) can be found by successive differentiation. z_1 can be real or complex. A Maclaurin series is the same as a Taylor series but with $z_1 = 0$. The theorem for a Taylor series is

$$\begin{aligned} F(z) &= F(z_1) + (z - z_1)F'(z_1) + \frac{1}{2!}(z - z_1)^2 F''(z_1) + \cdots + \frac{1}{n!}(z - z_1)^n F^{(n)}(z_1) \\ &= \sum_{n=0}^{\infty} \frac{F^{(n)}(z_1)}{n!} (z - z_1)^n \end{aligned} \quad (8.17)$$

The infinite series converges to $F(z)$. A proof of the theorem can be found in Churchill [38, p. 129-131]. The expansion of a binomial (two-term polynomial) of the form $(1 + z)^{-1}$ results in a power series called the “binomial series” (Churchill [38, p. 131], Boas [19, p. 24]):

$$(1 + z)^p = 1 + pz + \frac{p(p - 1)}{2!} z^2 + \frac{p(p - 1)(p - 2)}{3!} z^3 + \cdots \quad (8.18)$$

where p is any real number, positive or negative, and z is a complex number in the z -plane.

For this series to converge, $|z|$ must be less than 1. If z is a real number, then we refer to an *interval of convergence* along the real axis in the z -plane; if z is a complex number, then we refer to a *circle of convergence* in the z -plane. See Chapter 2.

The power series expansion of a binomial can be determined by using Equation (8.18) as long as the binomial is in the form

$$(1+z)^p$$

before expansion, where $|z| < 1$. Otherwise, the series will not converge.

Given a z -transform 2-term wavelet

$$F(z) = z - z_1 = -z_1 + z$$

then the present application is to design a time-domain filter (called an “inverse filter”) that, when convolved with the wavelet

$$(-z_1, 1)$$

will reshape it to a spike (the number “1”) at the origin ($t = 0$). *For all other times, we want the inverse z -transform to be zero.*

Seismic resolution will therefore have been increased because the “wavelet” will be shortened in time from two points to one, and this shorter wavelet will not interfere with parts of other wavelets arriving at the same time. The z -transform of the inverse filter that will accomplish this is

$$\frac{1}{(-z_1 + z)} = (-z_1 + z)^{-1} \quad (8.19)$$

Equation (8.19) can be written

$$\frac{1}{(-z_1 + z)} = \frac{1}{-z_1(1 - z/z_1)} = -z_1^{-1}[1 - z/z_1]^{-1} \quad (8.19a)$$

where the quantity in brackets is now in the form of Equation (8.18), with $p = -1$ and with z replaced by $-z/z_1$. The Taylor series will converge as long as

$$\left| \frac{z}{-z_1} \right| < 1$$

Otherwise, it will diverge; the filter coefficients will become larger and larger.

But $|z/z_1| < 1$ for z_1 outside the unit circle; therefore, the expansion will converge for a minimum-phase inverse filter. It is the modulus of the complex number z/z_1 that governs convergence.

Application of Equation (8.18) to the quantity in brackets in Equation (8.19.a) gives

$$\begin{aligned} \frac{1}{1 - z_1^{-1}z} &= (1 - z_1^{-1}z)^{-1} \\ &= 1 + (-1)(-z_1^{-1}z) + \frac{(-1)(-1-1)}{2!}(-z_1^{-1}z)^2 + \frac{(-1)(-1-1)(-1-2)}{3!}(-z_1^{-1}z)^3 + \dots \\ &= 1z^0 + z_1^{-1}z^1 + \frac{2}{2!}z_1^{-2}z^2 + \frac{6}{3!}z_1^{-3}z^3 + \dots \end{aligned}$$

so that

$$\frac{1}{1 - z_1^{-1}z} = 1z^0 + z_1^{-1}z^1 + z_1^{-2}z^2 + z_1^{-3}z^3 + \dots \quad (8.20)$$

From Equation (8.19.a), each term in Equation (8.20) must be multiplied by $-z_1^{-1}$ before convolution with the couplet so that Equation (8.20) becomes

$$= (-z_1^{-1})z^0 - z_1^{-2}z^1 - z_1^{-3}z^2 - z_1^{-4}z^3 + \dots$$

Observe in Equation (8.20) that, after the expansion,

- Only powers of z^t result for which $0 \leq t < \infty$.
- The time series is of infinite length.

Thus, the time-domain filter coefficients (the inverse z-transform) are defined only for $t \geq 0$, and are

$$-z_1^{-1}, -z_1^{-2}, -z_1^{-3}, -z_1^{-4}, +\dots$$

This series will converge because

$$\frac{-1}{z_1^n}$$

goes to zero as n increases because z_1 is a root outside of the unit circle, by definition. On the unit circle, convergence will not occur, which is just another way of saying that the wavelet $(1, 1)$ never decays with time—a physically impossible situation. One would therefore expect faster convergence as the root is moved further away from the unit circle.

If we arbitrarily define some single point at time t_i on a seismic trace to be the “present” (a point in the middle of the trace, say), then sample points before t_i are in its “past”, and sample points after t_i are in its “future” (with respect to its “present” time t_i). The inverse filter that we have just derived is called a *memory* filter because it requires (uses) only the present and past values of the time series, not values that have not yet been recorded (the future).

Programming of Equation (8.18) is a simple matter in Fortran or *Mathematica*. To get the series expansion, the *Mathematica* statement is

`Series[1/(-z1+z), {z, 0, 11}]`

and the output of this is

$$\begin{aligned} z\text{-transform} &= 0.769231 - 0.591716 z + 0.455166 z^2 - 0.350128 z^3 + 0.269329 z^4 \\ &- 0.207176 z^5 + 0.159366 z^6 - 0.122589 z^7 + 0.0942996 z^8 - 0.0725382 z^9 \\ &+ 0.0557986 z^{10} - 0.042922 z^{11} + \dots \end{aligned}$$

To extract the coefficients, the statement

`Table[SeriesCoefficient[%, n], {n, 0, 11}]`

can be used right after `Series`, which results in

```
Coefficients = 0.769231, -0.591716, 0.455166, -0.350128, 0.269329,
-0.207176, 0.159366, -0.122589, 0.0942996, -0.0725382, 0.0557986,
-0.042922
```

Examples

Several examples follow in Figures 8.1, 8.2, and 8.3 to illustrate the minimum-delay time-domain 2-term wavelet, the inverse filter that will spike out the wavelet, and the result of the convolution of the filter with the wavelet. In general, the wavelet and the filter can each be complex so that a complex convolution would be required.

The first example shown in Figure 8.1 is for a complex binomial $(-z_1, 1)$, where $-z_1 = 1.3e^{i\theta}$ for $\theta = 135^\circ$. In this figure and those to follow θ is converted to radians for actual computations. The filter coefficients in Figure 8.1 (center) are computed from Equation (8.19). After filtering (Figure 8.1, right), the (real) output is "1" at $t = 0$. The small error at the end of the time series (time index 10) is consistent with the theory; i.e., perfect answers are obtained only if the input time series is of infinite length (two non-zero coefficients followed by an infinite number of zeros for the input); because this is not true (the input series and the filter were only 11 points long) some error is to be expected. The magnitude of the error depends upon how fast the Taylor series converges.

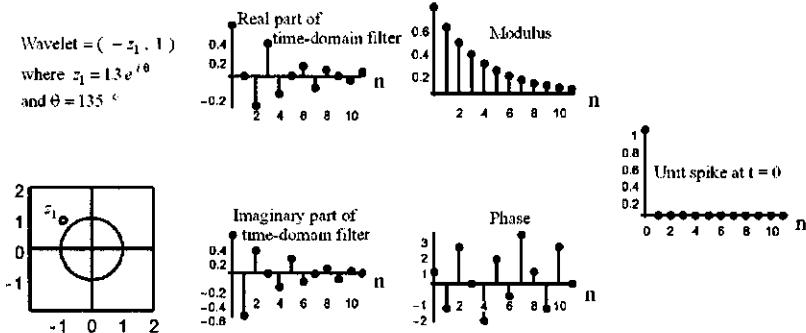


Figure 8.1: Complex time-domain minimum-delay 2-term wavelet (upper left) equal to $-z_1, 1$ where $-z_1 = 1.3 e^{i\theta}$ for $\theta = 135^\circ$. The filter required to spike out this complex wavelet is shown in the center four diagrams. The result of the complex convolution is the real spike at $t = 0$ as shown on the diagram at the right. (diplabc.nb → diplabc.cdr → diplabc.wmf)

In Figure 8.2, the positive-imaginary time-domain minimum-delay couplet is $(-z_1, 1)$ where $-z_1 = 1.3 e^{i\theta}$ for $\theta = 90^\circ$. The filter coefficients are computed from Equation (8.19). After filtering with the filter coefficients shown in Figure 8.2, the (real) output is "1" at $t = 0$.

In Figure 8.3, the (real) couplet is $(-z_1, 1)$ (left column) where $-z_1 = 1.3 e^{i\theta}$ and $\theta = 180^\circ$. The filter coefficients (middle four diagrams in each figure) are computed from Equation (8.19). The real, deconvolved output (right) is "1" at $t = 0$.

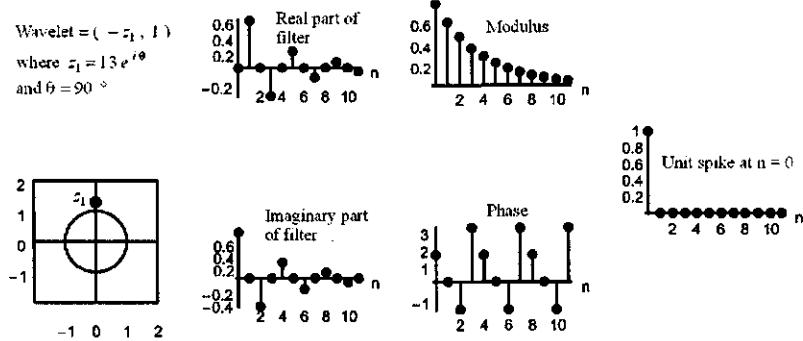


Figure 8.2: Positive-imaginary complex time-domain minimum-delay 2-term wavelet (left column), (complex) filter coefficients required to spike out wavelet (center four diagrams), and deconvolved (real) output (right). (dip2abc.nb → dip2abc.cdr → dip2abc.wmf)

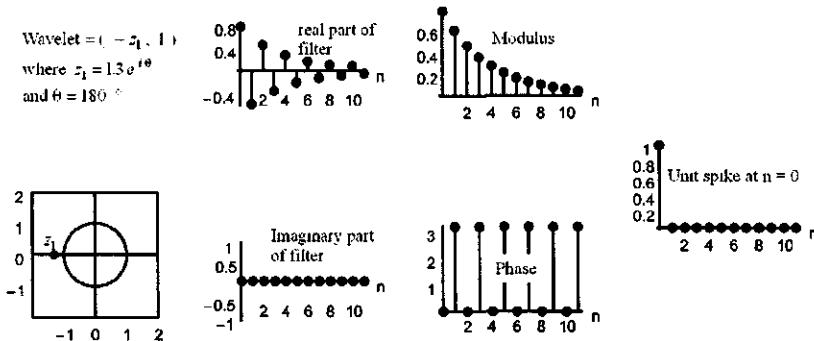


Figure 8.3: Negative real minimum-delay time-domain 2-term wavelet (left), Inverse filter coefficients (center), and deconvolved real output (right). (dip3abc.nb → dip3abc.cdr → dip3abc.wmf)

8.1.2 Inverse filtering of a maximum-delay 2-term wavelet using z -transforms

We wish to determine the filter $f(t)$ that will deconvolve the two-term wavelet

$$(-z_1, 1)$$

into the shorter wavelet "1" at $t = 0$. z_1 is chosen to be a complex number in the z -plane. That is, we wish to determine the inverse z -transform of

$$\frac{1}{-z_1 + z}$$

because

$$1 \iff (-z_1 + z) \frac{1}{-z_1 + z}$$

The inverse filter, when convolved with the couplet

$$(-z_1, 1)$$

will reshape it to a spike (the number "1") at the origin, ($t = 0$). *For all other times, we require the inverse z-transform to be zero.*

Here is a complete *Mathematica* program to illustrate the above. First define the convolution subroutine `fold` as shown below. Then define a 2-term wavelet, which might be either minimum- or maximum-delay. That is,

wavelet = $\{-z_1, 1\}$

If $|z_1| > 1$ then the wavelet is minimum-delay. If $|z_1| < 1$ then the wavelet is maximum-delay. Now find the inverse filter coefficients that when convolved with the wavelet will spike it out at $t = 0$. This can be done with

```

n   = 10;
filter = Series[ $\frac{1}{-z1 + z}$ , {z, 0, n}]
filter = CoefficientList[filter, z]

```

where n is the number of filter coefficients desired. Then convolve the wavelet with the filter:

```
fold[wavelet, filter];
```

The output is in c. The entire program is

```

fold[a_, b_] := (lc = Length[a] + Length[b] - 1; c = Table[0, {lc}];  

Do[Do[k = i + j - 1;  

c[[k]] = c[[k]] + a[[i]]*b[[j]], {j, 1, Length[b]}], {i, 1, Length[a]}]);

```

```

wavelet = {-z1, 1};

filter = Series[ $\frac{1}{-z1 + z}$ , {z, 0, n}];

filter = CoefficientList[filter, z];
          fold[wavelet, filter];
          c

```

For $n = 10$, $n = 20$, and $n = 30$ the output in the vector c for $c/z_1 - > 2$ is

$$n = 10: \quad \{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \frac{-1}{2048}\}$$

As predicted, the longer the filter the smaller the error. We would like that last term to be zero, but that is only possible if the inverse filter is infinitely long.

See page 387 for how to use **Series** if the 2-point wavelet to be spiked out is maximum-delay.

The z -transform of the inverse filter that will spike out a maximum-delay couplet is

$$\frac{1}{-z_1 + z} = \frac{1}{z(1 - z_1/z)} = \frac{1}{z(1 - z_1 z^{-1})}$$

Again use the power series called the “binomial series” (Boas, [19, p.24]):

$$(1 + z)^p = 1 + pz + \frac{p(p-1)}{2!} z^2 + \frac{p(p-1)(p-2)}{3!} z^3 + \dots \text{ for } |z| < 1 \quad (8.18)$$

where p is any real number, positive or negative, and z is a complex number in the z -plane. For this series to converge $|z|$ must be less than 1.

We can compute the power series expansion of a binomial by using Equation (8.18) as long as the binomial is in the form

$$(1 + z)^p$$

before expansion, where $|z| < 1$. Otherwise, the series will not converge.

Given the inverse of the z -transform of a maximum-delay binomial in the form

$$\frac{1}{-z_1 + z} = \frac{1}{z(1 - z_1 z^{-1})} = z^{-1}[1 - z_1 z^{-1}]^{-1} \quad (8.21)$$

the quantity in brackets is now in the form of Equation (8.18), where $p = -1$ and z becomes x . This expansion will converge as long as

$$|-z_1 z^{-1}| < 1$$

Otherwise, it will diverge; the filter coefficients will get larger and larger.

Note that after expansion of the bracketed quantity, in order to reshape the couplet to “1” at $t = 0$, each term in the series expansion of $(1 - z_1 z^{-1})^{-1}$ must be multiplied by z^{-1} .

By our sign convention,

$$z = e^{-i\omega}$$

so that multiplication by z^{-1} of each term in the expansion of the bracketed term in Equation (8.21) means that the series is time-shifted towards negative time by one sample interval. Thus, the filter coefficients are defined only for negative time in the expansion of Equation (8.21).

Using Equation (8.18) with $p = -1$ and with $-z_1 z^{-1}$ substituted for z gives

$$\begin{aligned} \frac{1}{z(1 - z_1 z^{-1})} &= z^{-1}[1 - z_1 z^{-1}]^{-1} \\ &= z^{-1} \left[1 + (-1)(x) + \frac{(-1)(-1-1)}{2!} (-z_1 z^{-1})^2 \right. \\ &\quad \left. + \frac{(-1)(-1-1)(-1-2)}{3!} (-z_1 z^{-1})^3 + \dots \right] \end{aligned}$$

so that

$$\frac{1}{z(1 - z_1 z^{-1})} = 1 z^{-1} + z_1 z^{-2} + z_1^2 z^{-3} + z_1^3 z^{-4} + \dots \quad (8.22)$$

Observe in Equation (8.22) that, after the expansion and multiplication of the Taylor series coefficients by z^{-1} , only powers of z^t result for which

$$-\infty \leq t < 0$$

Again, the time series is of infinite length. This series will converge because

$$|z_1^n|$$

goes to zero as n increases because z_1 is a root inside the unit circle, by definition. Thus, the time-domain inverse filter coefficients are defined only for $t < 0$, and are

$$1, z_1, z_1^2, z_1^3, +\dots \quad (8.23)$$

Such a filter is called an *anticipation* filter because it requires only future values of the recorded time series, not the value presently being recorded, or past values that have already been recorded.

From (8.23), a *Mathematica* statement that can be used to generate the inverse filter coefficients is

```
order=11;
filter=Table[z1^n,{n,order,0,-1}]
```

and the coefficients for $z_1 = 0.8 e^{i\theta}$, $\theta = 135^\circ$ are:

$$\begin{aligned}
\text{Coefficients} = & 0.06074 + 0.06074 i, -9.31323 10^{-17} - 0.107374 i, \\
& 0.0949063 + 0.0949063 i, 0.167772 - 1.16415 10^{-16} i, \\
& -0.148291 + 0.148291 i, 1.36424 10^{-16} + 0.262144 i, \\
& 0.231705 - 0.231705 i, -0.4096 + 1.42109 10^{-16} i, \\
& 0.362039 + 0.362039 i, -1.11022 10^{-16} - 0.64 i, \\
& -0.565685 + 0.565685 i, 1
\end{aligned}$$

The time order of coefficient generation by the Table statement is for the most negative time first, and that the coefficients converge toward zero for increasing negative time (Figure 8.4). All filter coefficients for zero and positive time must be zero.

An alternate approach to getting the inverse filter coefficients using the same *Mathematica Series* function that was used above for the minimum-delay 2-point wavelet is to recognize that an expansion in powers of $1/z = z^{-1}$ is an expansion around infinity. The *Mathematica* statements are therefore

```
Series[1/(-z1+z),{z,Infinity,11}]
CoefficientList[% ,z]
```

which results in the following identical output for $z_1 = 0.8 e^{i\theta}$, $\theta = 135^\circ$:

$$\begin{aligned}
 \text{Series} &= \frac{0.06074 + 0.06074 i}{z^{12}} - \frac{9.85625 \cdot 10^{-17} + 0.107374 i}{z^{11}} \\
 &- \frac{0.0949063 - 0.0949063 i}{z^{10}} + \frac{0.167772 - 1.249 \cdot 10^{-16} i}{z^9} \\
 &- \frac{0.148291 + 0.148291 i}{z^8} + \frac{1.52056 \cdot 10^{-16} + 0.262144 i}{z^7} \\
 &+ \frac{0.231705 - 0.231705 i}{z^6} - \frac{0.4096 - 1.66533 \cdot 10^{-16} i}{z^5} \\
 &+ \frac{0.362039 + 0.362039 i}{z^4} - \frac{1.11022 \cdot 10^{-16} + 0.64 i}{z^3} \\
 &- \frac{0.565685 - 0.565685 i}{z^2} + \frac{1}{z}
 \end{aligned}$$

The coefficients from **Series** are already ordered with the most negative exponents (time) first. No terms in positive powers of z occur in the expansion. The inverse filter is therefore defined only in negative time.

The entire *Mathematica* program set up for a maximum-delay 2-point wavelet $\{ -z_1, 1 \}$ using the same subroutine **fold** (differences in bold-face) is

```

wavelet = {-z1, 1};
filter = Series[ $\frac{1}{-z1 + z}$ , {z,  $\infty$ , n}];
filter = Reverse[CoefficientList[filter,  $\frac{1}{z}$ ]];
fold[wavelet, filter];
c

```

Again, for $n = 10$, $n = 20$, and $n = 30$ the output in the vector c for $c/\cdot z_1 - 0.8$ is

Again, as predicted for the minimum-delay wavelet, the longer the filter the smaller the error. We would like that first term in each list to be zero, but that is only possible if the inverse filter is infinitely long.

Examples

An example to illustrate a maximum-delay 2-term wavelet, the inverse filter that will spike out the wavelet, the folded and shifted filter, and the result of

the convolution of the filter with the wavelet are shown in Figures 8.4 and 8.5. In general, the wavelet and the filter can each be complex so that a complex convolution would be required.

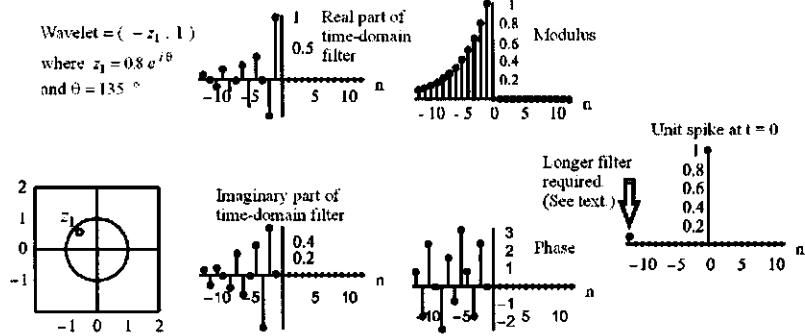


Figure 8.4: Complex maximum-delay 2-term wavelet (upper left) equal to $(-z_1, 1)$ where $-z_1 = 0.8 e^{j\theta}$ for $\theta = 135^\circ$. The filter computed from (8.22) required to spike out this complex wavelet is shown in the center four diagrams. Complex convolution is the real spike at $t = 0$ as shown on the diagram at the right. See Figure 8.5 for explanation of anomalous point at $n = -12$. (dip10abc.nb,cdr → dip10abc.wmf)

An “equi-delay” 2-term wavelet is shown in Figure 8.6. This is not a physically realizable wavelet because it does not decay. It can be reshaped to a spike at $t = 0$, however, but the filtered output will not be zero for all times other than $t = 0$.

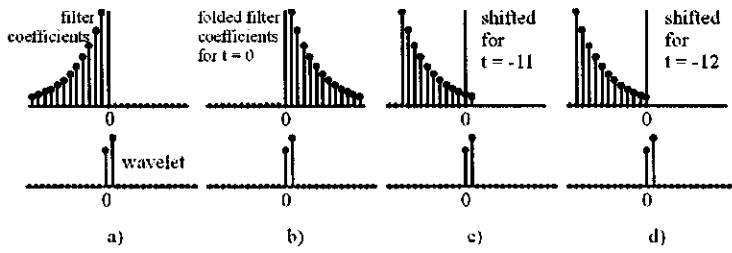


Figure 8.5: (a) Plot of filter and 2-point wavelet. Wavelet is maximum-delay, so filter is defined only for $t < 0$. (b) Filter coefficients are folded about $t = 0$ but not shifted. Filtered output at $t = 0$ is obtained by multiplying wavelet and filter coefficients and summing products. It is clear that the value of the filtered output at $t = 0$ must be zero. (c) Filter coefficients shifted toward $t < 0$ and positioned with respect to wavelet for the output at $t = -11$. Each point of the wavelet will be multiplied by a filter coefficient. Correct output (zero) will therefore be obtained. (d) Filter coefficients shifted one additional point toward $t < 0$ for output at $t = -12$. Now the filtered output would be incorrect because not enough filter coefficients have been computed. One additional filter coefficient would be required to get the correct output (zero) at $t = -12$. To get a correct output for all t the filter would have to be infinitely long for all of these examples. (dip10abcComment.nb,cdr → dip10abcComment.wmf)

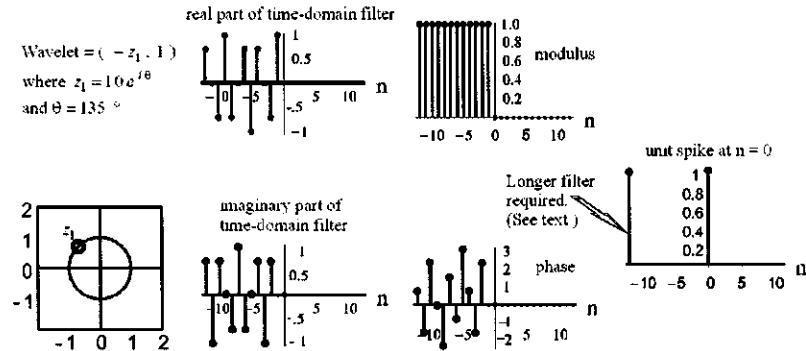


Figure 8.6: Complex time-domain equi-delay 2-term wavelet $(-z_1, 1)$ where $z_1 = 1.0 e^{j\theta}$ for $\theta = 135^\circ$. The filter required to spike out this complex wavelet is shown in the center four diagrams. The filter coefficients are computed from Equation (8.22). The filter coefficients are constant, and neither converge or diverge, because the root z_1 is on the unit circle. After filtering with the filter coefficients shown, the output is “1” at $t = 0$. See Figure 8.5 for explanation of anomalous point at $n = -12$. (dip20abc.nb,cdr → dip20abc.wmf)

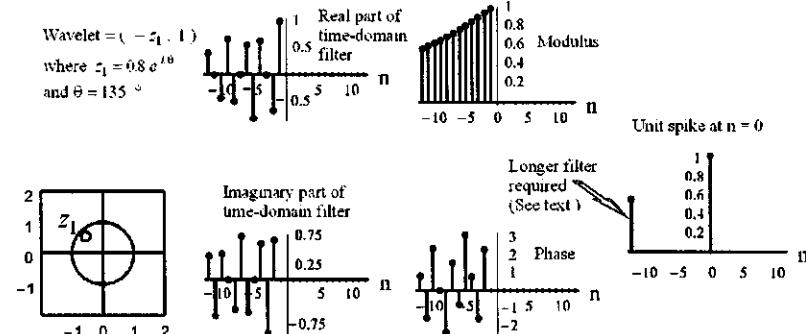


Figure 8.7: Complex minimum-delay time-domain 2-term wavelet with root close to the unit circle. Wavelet is $(-z_1, 1)$ where $z_1 = 0.95 e^{j\theta}$ for $\theta = 135^\circ$. The filter required to spike out this complex wavelet is shown in the center four diagrams. The filter coefficients are computed from Equation (8.22). Note the much slower rate of convergence (from right to left) of the filter coefficients compared with the wavelet of Figure 8.4. This root, z_1 , is closer to the unit circle. After filtering with the complex filter coefficients shown, the output is “1” at $t = 0$. (dip30abc.nb,cdr → dip30abc.wmf)

8.1.3 Inverse filtering of a seismic trace using z -transforms

The objective of inverse filtering (deconvolution) of a seismic trace is to image the subsurface geology. The concepts involved can be illustrated by generating a synthetic seismic trace:

- Define the roots (minimum-, maximum-, or mixed-delay) of the seismic wavelet,
- Generate the seismic wavelet by convolution of the 2-term wavelets formed from the roots,
- Convolve the wavelet with a reflectivity function (commonly approximated by a sequence of random numbers).

Decomposition (deconvolution) of the seismic trace into the reflection coefficients is a simple procedure if the shape of the seismic wavelet is known:

- Determine the roots of the seismic wavelet (assuming that the wavelet can be isolated and identified),
- Compute the Taylor series expansion (the inverse z -transform) of each of the reciprocal two-term z -transform formed from the roots,
- Convolve these expansions to obtain the time-domain inverse filter,
- Convolve this inverse filter with the seismic trace. This yields the original reflectivity function that was used to generate the trace. We can refer to this reflectivity function as the “geology”; it is the time sequence of the acoustic impedance contrasts that image the geometry of the subsurface reflectors.

The procedure is illustrated in Figure 8.8, where a time-invariant minimum-delay wavelet was used. If the wavelet shape is known, however, then, the deconvolution filter can be computed no matter what the delay properties of the wavelet.

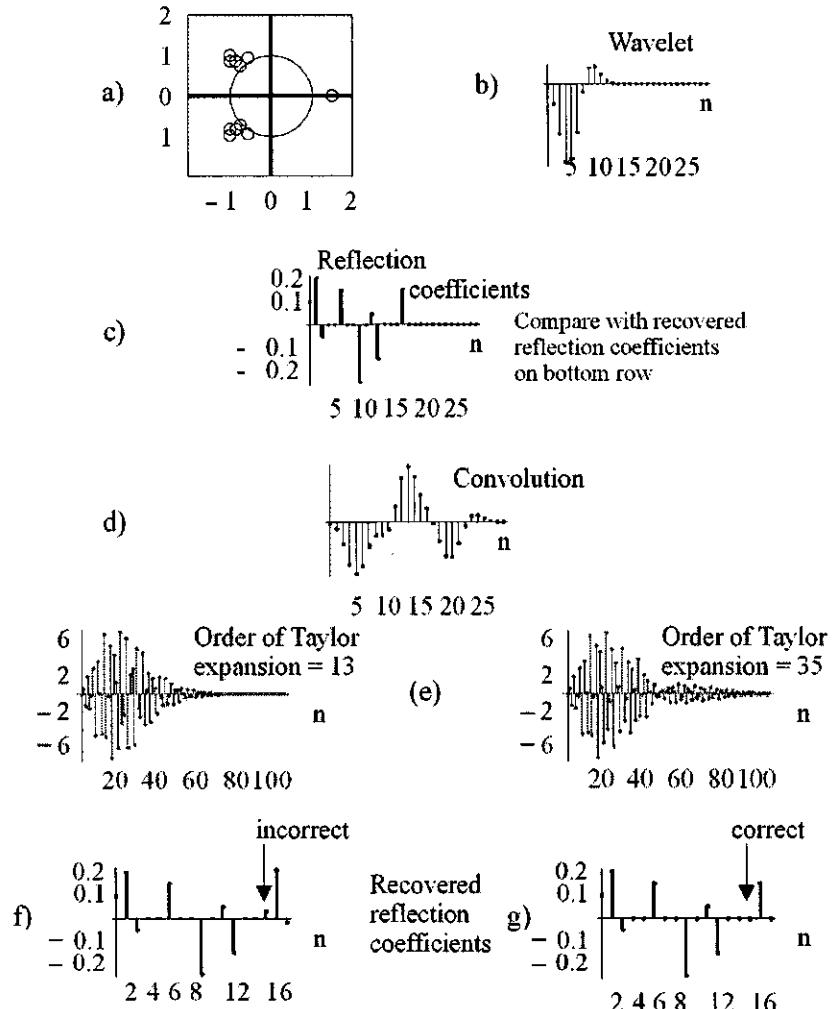


Figure 8.8: (a) (Top) Minimum-delay roots used to generate the wavelet shown to right. (b) Minimum-delay wavelet generated from the roots. (c) Reflection coefficient function. (d) Convolution of reflection coefficients with wavelet. (e) Taylor series filter coefficients that will be used to recover the reflection coefficients; i.e., deconvolve the seismic trace. (f) Recovered reflectivity function after deconvolution using 13 coefficients in the Taylor series expansion of each couplet. The deconvolution is good, but not quite perfect (see "Incorrect" above). (g) Another inverse filter with Taylor series expansion having 35 coefficients. Although the theory requires a Taylor series expansion of infinite length to get perfect results for all times t , it is clear that the expansion (g) is long enough for essentially perfect results. The results would not be good, however, for longer reflection times, and the series expansions would have to be made longer, as required by the theory. lab1f.nb → lab1f.cdr → lab1f.wmf)

8.2 Inverse finite filters, infinite input

The second subdivision of deconvolution and shaping filters listed on page 372 includes those inverse filters of finite length but input of infinite length. Consider a plane, horizontal wave front in a water layer (Backus [8], Kalisvaart [91]). The water layer behaves as a *wave guide* as shown in Figure 8.9.

8.2.1 Exact filters for wave guides

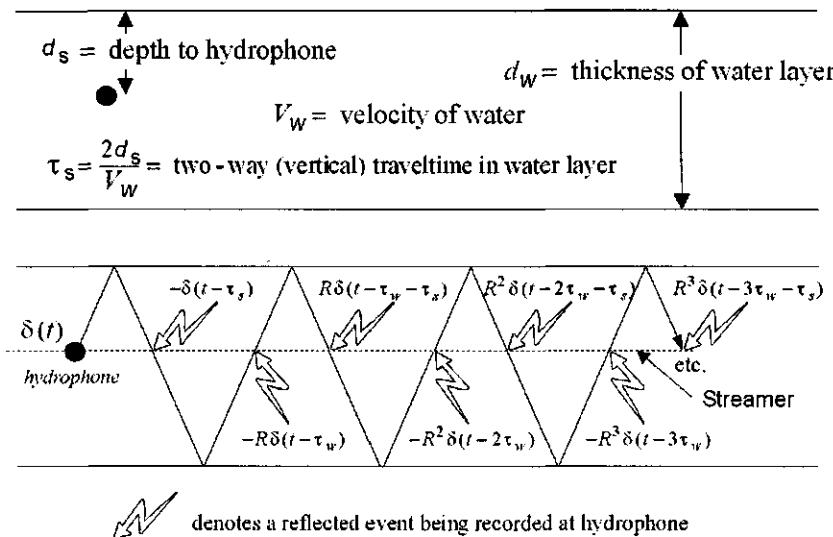


Figure 8.9: A water-layer wave guide. This layer geometry applies just as well to a low-velocity weathered zone. (wlwg.cd → wlwg.wmf)

The energy released within the wave guide is partially trapped within the water layer. We wish to examine the multiple reflections, or reverberations, that arrive at a hydrophone in the water layer because of repeated reflection of the wave front between the water bottom and water surface. The large-amplitude reverberations developed within the wave guide interfere with weaker, later-arriving reflections from the deeper horizons of geological interest. The reverberations theoretically continue forever; this is the input to the filter. We can remove the reverberations with a filter of finite length.

Reverberations in the waveguide also leave the waveguide, travel downward where they are reflected back and then must re-enter the waveguide and undergo further reverberation. The process is equivalent to having an “effective source wavelet” reflected from each subsurface reflector, as shown in Figure 8.10 instead of a reverberation wavelet leaving the waveguide, reflect, and return to the waveguide.

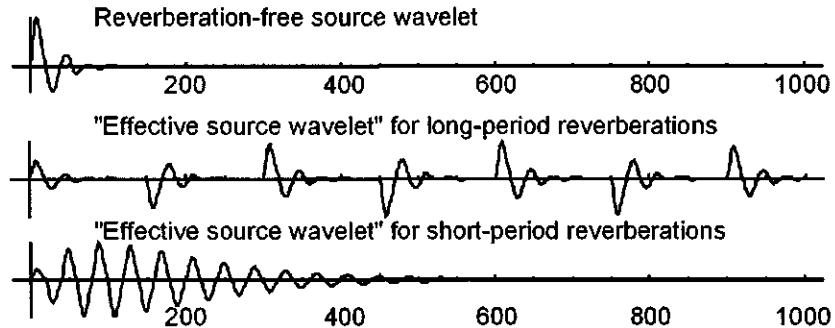


Figure 8.10: Top: The basic non-reverberating wavelet. Middle: The “effective source wavelet” for long-period reverberations. Bottom: For short-period reverberations. The reverberation process can be thought of as a linear one-step process of reflection of the long-period or short-period reverberation source wavelets shown instead of the actual process of having the wavelet leave the waveguide and re-enter after reflection. Different thicknesses of waveguide beneath the source and receiver will require separate treatment to remove the reverberations. BackusFilterExamples.nb,cdr → EffectiveSourceWavelet.wmf

The water layer will have a well-defined frequency response depending on its thickness; it will suppress some frequencies more than others. The output of the hydrophone will depend on this frequency response, and also on the depth of the hydrophone below the surface of the water.

As far as the derivation of the frequency response of the water layer is concerned, we can start measuring the times of successive multiple reflections of the primary wave front with respect to the time t as the front passes the hydrophone either on the way up or on the way down. We choose the starting reference time t as the front passes the hydrophone on the way up.

To examine the nature of the inverse filter required to remove the reverberations, we must know the frequency response of the water layer. By definition, the Fourier transform of the impulse response of the water layer is the frequency response of the water layer (page 109), so we introduce an impulsive source, i.e., a Dirac delta function $\delta(t)$ into the wave guide, deduce the time-response of the wave guide, and then as usual take the Fourier transform of this time-response to obtain the frequency response of the water layer.

Choose as a model a water layer with a thickness of d_w and with a hydrophone at a depth d_s below the surface. The model is shown in Figure 8.9. The two-way travel-time between the hydrophone and the surface of the water is $\tau_s = 2d_s/V_w$, where V_w is the velocity of water. The travel-time between the water surface and the water bottom, and then back to the surface is $\tau_w = 2d_w/V_w$.

The reflection coefficient at the water bottom is r and at the water surface is $r_0 = -1$. Successive multiple reflections are shown in the lower part of the figure. We consider vertical incidence only, at a single hydrophone. For clarity,

the paths associated with each reflection recorded at the hydrophone are not drawn vertically.

The output $h(t)$ of the hydrophone is obtained by summing the primary impulsive wave front $\delta(t)$ and all of the multiple reflections as shown in Figure 8.9 (bottom). Reflection from the top of the wave guide results in a polarity reversal ($r_0 = -1$); reflection from a “hard” bottom does not, but the reflected event is reduced in amplitude by the value of the reflection coefficient r . For a “hard” water bottom ($0 < r < 1$).

When a marine source array is fired at time t the hydrophone would ideally record the impulse $\delta(t)$. The next event recorded at the hydrophone is one reflected and reversed in polarity from the top of the water layer, i.e., $\delta(t - \tau_s)$. Taking into account all polarity reversals caused by reflection from the top of the waveguide, and scaling caused by reflection from the bottom, we obtain for the time sequence $h(t)$ of reordered impulses shown in Figure 8.9.

$$\begin{aligned} h(t) &= \delta(t) - \delta(t - \tau_s) - r\delta(t - \tau_w) + r\delta(t - \tau_w - \tau_s) \\ &\quad + r^2\delta(t - 2\tau_w) - r^2\delta(t - 2\tau_w - \tau_s) - r^3\delta(t - 3\tau_w) + \dots \\ &= \sum_{n=0}^{\infty} (-1)^n r^n \delta(t - n\tau_w) - \sum_{n=0}^{\infty} (-1)^n r^n \delta(t - n\tau_w - \tau_s) \end{aligned}$$

The impulse response of the water layer is thus composed of two parts [8]:

$$\sum_{n=0}^{\infty} (-1)^n r^n \delta(t - n\tau_w) \quad (8.24)$$

a part that is independent of the hydrophone because it only contains τ_w , and

$$\sum_{n=0}^{\infty} (-1)^n r^n \delta(t - n\tau_w - \tau_s) \quad (8.25)$$

a part that depends upon the depth of the hydrophone as well as water layer thickness.

Equation (8.24), which is independent of the hydrophone because the summation does not involve τ_s , is called the *water-layer filter*. The frequency response of the water-layer filter is obtained, as usual, by taking the Fourier transform of the time-response of the water layer:

$$H(\omega) = \int_{-\infty}^{\infty} \left[\sum_{n=0}^{\infty} (-1)^n r^n \delta(t - n\tau_w) \right] e^{-i\omega t} dt \quad (8.26)$$

Now, because

$$\delta(t - n\tau_w) \iff e^{-i\omega n \tau_w}$$

Then

$$H(\omega) = \sum_{n=0}^{\infty} (-1)^n r^n e^{-i n \omega \tau_w}$$

which is the binomial expansion of

$$H(\omega) = \frac{1}{1 + r e^{-i\omega\tau_w}} \quad (8.27)$$

The modulus $|H(\omega)|$ is shown in Figure 8.11 for $r = 0.05, 0.5, 0.7, 1.0$ for a water depth of 11 meters.

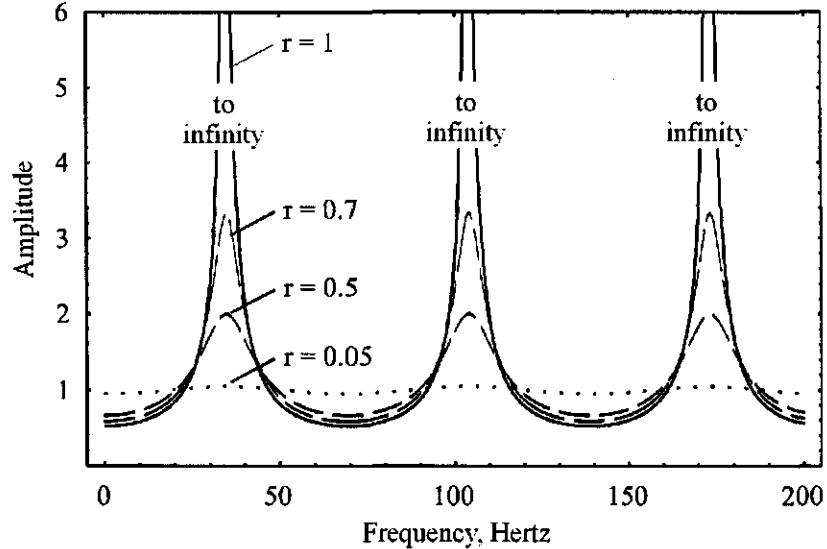


Figure 8.11: Amplitude response of a water-layer wave guide for a water depth of 11 meters for $r = 0.05, 0.5, 0.7, 1$. This is the amplitude response of a water-confined reverberation spike train computed from Equation (8.28). (BackusWaterLayerD30R)

The modulus of $H(\omega)$ is also given by Backus [8] as

$$|H(\omega)| = [1 + r^2 + 2r \cos(\omega\tau_w)]^{-1/2} \quad (8.28)$$

and the phase angles $\phi(\omega)$ by

$$\phi(\omega) = \tan^{-1} \left[\frac{r \sin(\omega\tau_w)}{1 + r \cos(\omega\tau_w)} \right]$$

Most impedance contrasts at a water-sediment interface are probably less than 0.8; however, for the (impossible) limiting case of a perfectly rigid bottom, $r = 1$ and Equation (8.28) reduces to

$$|H(\omega)| = \left| \frac{1}{2} \sec\left(\frac{\omega\tau_w}{2}\right) \right| \quad (8.29)$$

For $r = 1$ Equation (8.28) goes to infinity when

$$2 + 2 \cos(\omega\tau_w) = 0$$

That is, when

$$\cos(\omega\tau_w) = -1$$

which happens when

$$\omega\tau_w = n\pi \quad n = 1, 3, 5, \dots$$

where τ_w is two-way vertical traveltime $= 2d_w/V_w$. Substituting for $\omega\tau_w$,

$$2\pi\nu_n\tau_w = n\pi \quad n = 1, 3, 5, \dots$$

or at frequencies

$$\nu_n = \frac{nV_w}{4d_w} \quad n = 1, 3, 5, \dots$$

Any water or rock layer behaves as a comb filter, peaked at the fundamental frequency ($n = 1$) (Figure 8.12) of 25.4 Hertz, and at the odd harmonics 76.2 Hertz ($n = 3$), 127 Hertz ($n = 5$), etc. If the water layer has a thickness of 11 m then a seismic source with a frequency content of, say, 10-50 Hertz must survive the relatively narrow pass band of the water-layer filter centered at 25 Hertz; the seismic trace will therefore be dominated by frequencies of 25 Hertz.

All layers are wave guides regardless of whether the medium is rock or water. If the reflection coefficients are smaller, i.e., more on the order of what is to be expected at a rock-rock interface instead of a rock-water interface, then the filtering effect of the wave guide is considerably reduced, as shown in Figure 8.11. Notice the almost negligible frequency effect for $r = 0.05$ (dotted line).

Equation (8.28) is plotted in Figure 8.12 for water depths of 15, 30, and 60 meters and for a water-bottom reflection coefficient r of 0.5.

The filtering effect of the depth d_r of a receiver (pressure transducer) in the wave guide given by Equation (8.25) is

$$\sum_{n=0}^{\infty} (-1)^n r^n \delta(t - n\tau_w - \tau_s)$$

The frequency response H_r of the filter for $r = 1$ is [8]

$$\begin{aligned} H_r(\omega) &= \int_{-\infty}^{\infty} [\delta(t) - \delta(t - \tau_s)] e^{-i\omega t} dt \\ &= 1 - e^{-i\omega \tau_s} \end{aligned} \tag{8.30}$$

The modulus and phase are

$$\begin{aligned} |H_r(\omega)| &= \left| 2 \sin\left(\frac{\omega\tau_s}{2}\right) \right| \\ \phi_r(\omega) &= \frac{3}{2\pi} - \frac{\omega\tau_s}{2} \end{aligned} \tag{8.31}$$

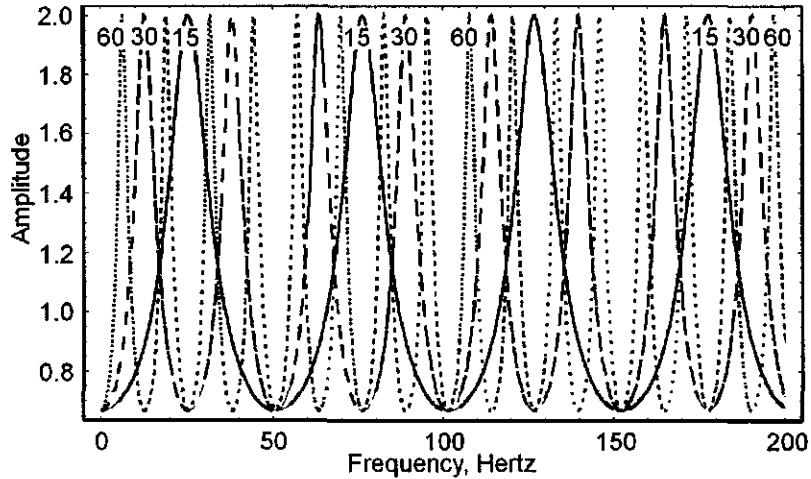


Figure 8.12: Response of a water-layer wave guide for $r = 0.5$ and water layer thicknesses of 15, 30, and 60 meters as shown. These are the amplitude responses of water-confined reverberation spike trains computed from the modulus of Equation (8.27). For just the water layer as the layer thickness increases more and more high frequency energy within a given bandwidth reaches the hydrophones. (D15.cd → D15.wmf)

The effect of hydrophone depth is shown in Figure 8.13. As the hydrophone depth is increased, the peak response shifts towards lower frequencies, and the seismic trace therefore takes on a “character” that is richer in lower frequencies. This is an important effect as shown in Figure 8.13.

From Figure 8.13, the filtering effect of hydrophone depth is not as sharply peaked as the water-layer filter, but the depth of the hydrophones below the water surface greatly affects the “character” (frequency content) of the seismic trace (Figure 8.13). The lower frequencies are enhanced at greater receiver depths. A receiver depth of 10-15 feet is commonly used in marine work because of the frequencies that are of interest in offshore exploration.

If Equations (8.24) and (8.25) are combined, then the frequency response of the combined effect of water layer thickness and hydrophone depth is shown in Figure 8.14. A typical water reverberation record from the Persian Gulf is shown in Figure 8.15. A Fourier analysis of four segments of trace 4 of Figure 8.15 is shown in Figure 8.16. Water depth is about 200 feet, and successive multiples are inverted because of the negative reflection coefficient at the air-water interface and the positive reflection coefficient at the water bottom.

It is apparent that we have a recording situation that requires instruments able to record a large dynamic range, because the signal-to-noise ratio might be small; most of the frequencies in the seismic wavelet will be sharply attenuated by the water-layer guide. If these frequencies are to be restored, then they must

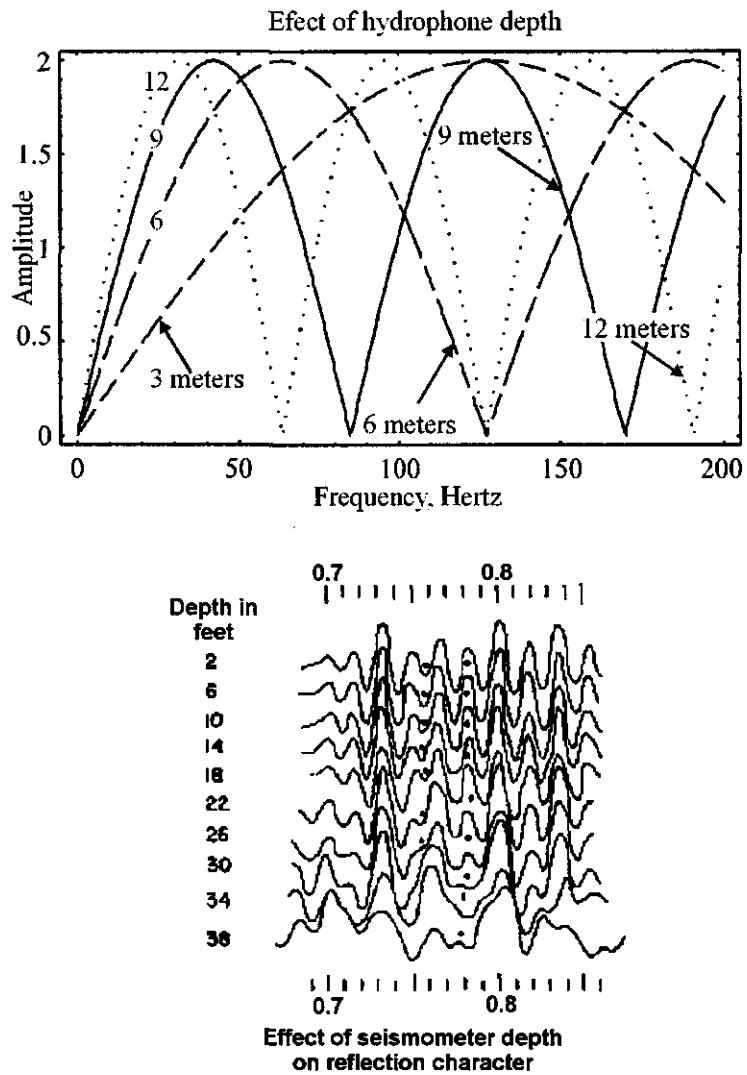


Figure 8.13: Top: Response of wave guide for hydrophone depths of 3, 6, 9, and 12 meters and water-bottom reflection coefficient $r = 1$. From Equation (8.30). As hydrophone depth increases more and more low-frequency energy is recorded. (BackusFilterSeismometerDepthVariesSD48.cd → BackusFilterSeismometerDepthVariesSD48.wmf). Bottom: Effect of seismometer depth on wavelet character. From Backus [8]. (char.bmp)

Combined effect of hydrophone depth and thickness of water layer

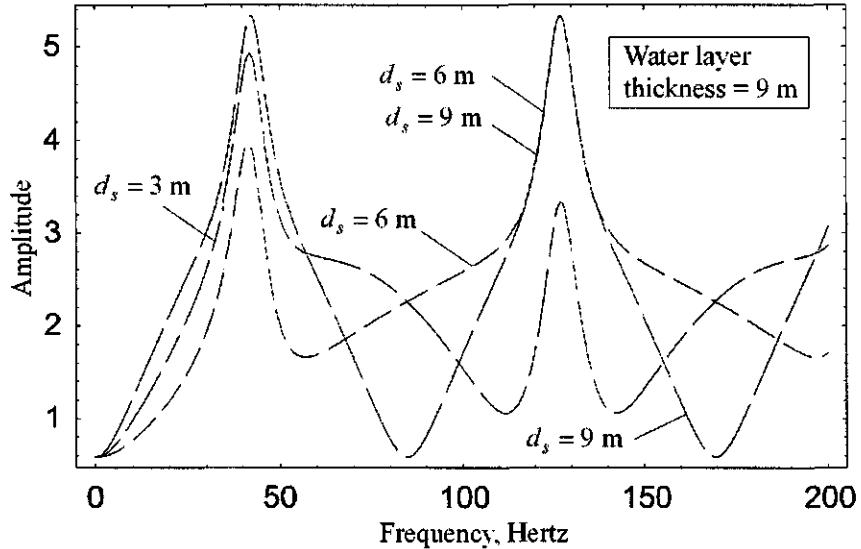


Figure 8.14: Response of 9-meter wave guide for water depths of 3, 6, and 9 meters and a reflection coefficient of $r = 0.5$. This is the amplitude response of a water-confined reverberation spike train computed from Equations (8.30) and (8.32). (BackusTotalResponse.nb,cdr)

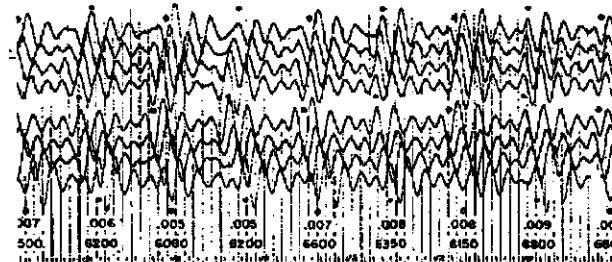


Figure 8.15: Textbook example of water reverberations in reflection data from the Persian Gulf. Water depth is about 200 feet. Note that successive reflections are inverted in polarity due to reflection at the free surface ($r = -1$) and lack of polarity reversal at the water bottom ($r > 0$). Figure from Backus [8]. (PG1.bmp)

be recorded to begin with, even though greatly attenuated with respect to the resonant frequencies. It is important that any automatic volume control (AVC)

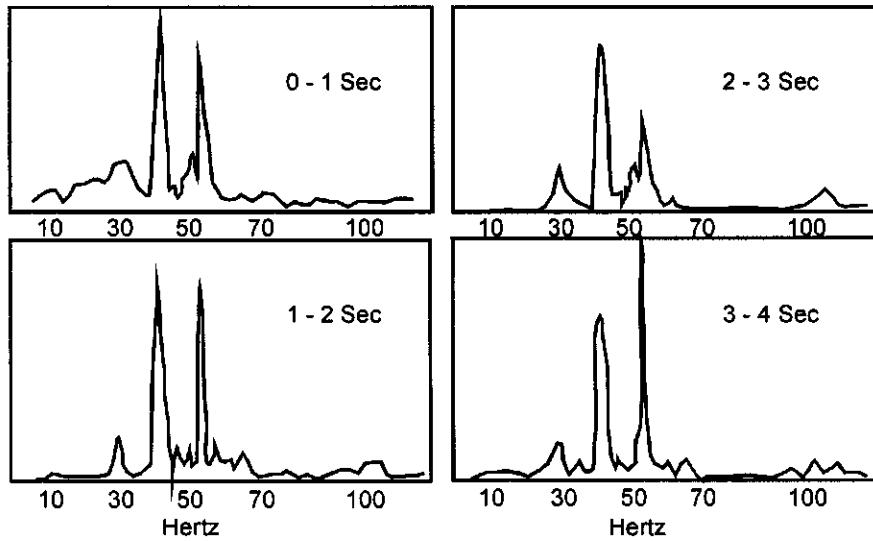


Figure 8.16: Frequency analysis of four segments of a trace shown in Figure 8.15. The sharply peaked responses are at 42 and 54 Hertz and are in excellent agreement with the resonant frequencies predicted for the thickness of the water-layer (about 200 feet) and the seismometer depth (25 feet). Figure reproduced from Backus [8, Figure 13] (PG2.bmp)

is not operative and does not further suppress these frequencies below the noise level of the recording system.

For an incident plane wave at an angle i with the vertical, the effective filtering of the water layer is found by replacing τ_w in Equation (8.27) by $\tau_w \cos i$ [8, p. 237]. If the water-bottom reflection coefficient is negative, the shape of the water-layer filter remains the same, but the frequency scale is shifted $f = V_w/4d_w$ Hertz to the right.

***z*-transform notation**

In *z*-transform notation, we define, as usual,

$$z = e^{-i\omega} \quad \text{and} \quad \tau_w = n\Delta t$$

where n is the number of time units (sample intervals) of two-way travel-time in water, that is, the period between multiples. Then

$$H(z) = \frac{1}{1 + r z^n} \tag{8.32}$$

In Equation (8.32), the expression $H(z) = 1/(1 + r z^n)$ is the *z*-transform of the *water-confined reverberation spike train*. This means that if a source wavelet of

arbitrary shape enters or leaves the water layer, then its frequency spectrum will undergo relative attenuation according to (8.27).

The z -transform of (8.24) is

$$1 - r z^n + r^2 z^{2n} - r^3 z^{3n} + \dots$$

where $\tau_w = n\Delta t$ and n is the number of time units (sample intervals) of two-way travel-time in water. For example, if the water depth is 30 meters and the water velocity V_w is 1,524 m/sec, then the two-way travel-time will be about 40 ms. If the data are sampled at an interval of 4 ms then $n = 10$ time units. The sequences of impulses (the water-confined reverberation spike train) arriving at the hydrophone is

$$1, 0, 0, 0, 0, 0, 0, 0, 0, -r, 0, 0, 0, 0, 0, 0, 0, 0, r^2 + \dots$$

In this infinite series, the first coefficient ("1") can be thought of as the primary source "wavelet". The coefficient $-r$ represents the reverberation that has been reflected from the water surface and from the water bottom $n = 10$ time units (40 ms) later. The coefficient r^2 represents the reverberation that has been reflected twice from the water surface and twice from the water bottom, arriving 20 time units (80 ms) after the primary.

To eliminate these water-confined reverberations we require an inverse filter whose z -transform is

$$1 + r z^n$$

and whose time-domain representation is therefore

$$1, 0, 0, 0, 0, 0, 0, 0, 0, r \quad (8.33)$$

where "1" is defined at $t = 0$ and " r " appears at the time position of $t = 10$ time units. It is apparent that convolution of this filter with the water-confined reverberation spike train converts the spike train into a spike at $t = 0$, that is, at the time corresponding to the arrival of the primary, and so eliminates the reverberations.

The delay characteristics of the wavelet are not important as shown in Figure 8.17 where long-period (the reflections do not overlap) reverberations have been generated and removed. The 2-point filter is designed to remove the reverberations, which are minimum-delay. It is not designed to shorten the wavelet, which does not have to be minimum-delay. Similar results are obtained for short-period multiples where the reflections overlap as shown in Figure 8.18.

We have concerned ourselves so far only with what a hydrophone would record as reverberations of a primary pulse reverberating in a wave guide. We have been able to remove these reverberations by application of the inverse filter whose z -transform is $1 + rz^n$. Some energy, however, leaves the water layer and propagates on down to be reflected back by a deep reflecting horizon. This reflected energy again enters the water layer and the wavelet reflected from

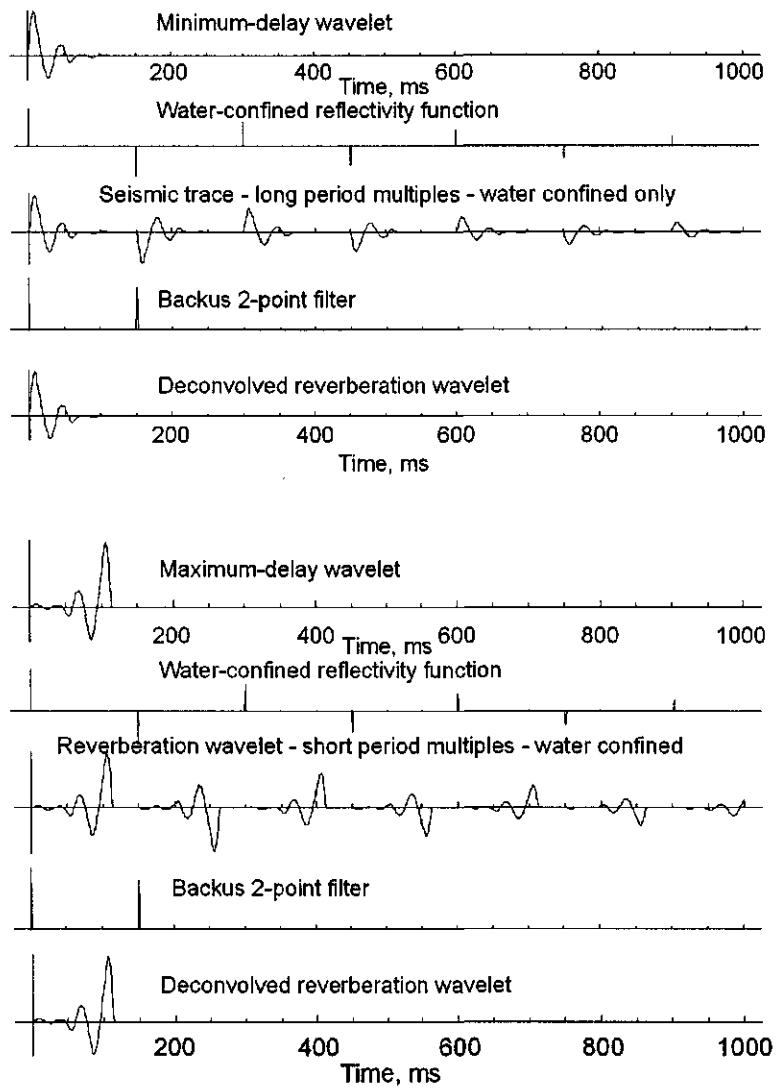


Figure 8.17: Top set: (a) Minimum-delay source wavelet. (b) Water-confined spike train (the reflectivity function) showing polarity reversals. (c) Convolution of wavelet with the spike train (the effective source wavelet). (d) The finite 2-point filter $1 + r z^n$. (e) Deconvolved trace. Bottom set: Same parameters but with a maximum-delay wavelet (the time-reverse of the above minimum-delay wavelet). The deconvolution is equally effective regardless of the delay characteristics of the wavelet. BackusFilterExamples.nb,cdr → LPMultiplesWCRST.wmf

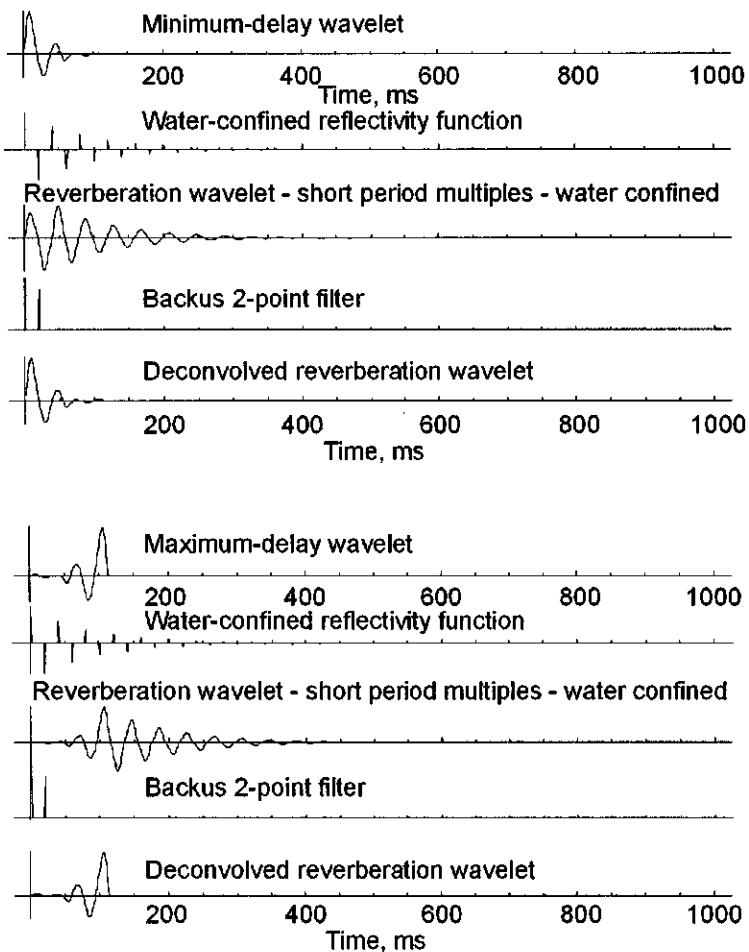


Figure 8.18: Top set: (a) Minimum-delay source wavelet. (b) Water-confined spike train showing polarity reversals. (c) Convolution of wavelet with water-confined spike train (the effective source wavelet). (d) The finite 2-point filter $1 + r z^{-n}$. (e) Deconvolved trace. Bottom set: Same parameters but with a maximum-delay wavelet (the time-reverse of the above minimum-delay wavelet). The deconvolution is equally effective regardless of the delay characteristics of the wavelet. BackusFilterExamples.nb,cdr → SP-MultiplesWCRST.wmf

a deeper interface therefore *also* undergoes reverberations in the wave guide. These must also be removed.

Thus, the data of interest, the deep reflection data, pass through two cascaded sections of the “water-layer filter”. The overall filtering effect of the water layer for reflections leaving and deep reflections re-entering the wave guide is given by

$$F(z)F(z) = (1 - rz^n + r^2 z^{2n} - r^3 z^{3n} + \dots)^2 \quad (8.34)$$

or

$$\frac{1}{(1 + r z^n)^2}$$

In the time domain, the *deep-reflection reverberation spike train* is therefore

$$\begin{aligned} 1, 0, 0, 0, 0, 0, 0, 0, 0, -2r, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3r^2, \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, -4r^3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5r^4, 0, 0, 0, \dots \end{aligned} \quad (8.35)$$

Multiplication in the z -transform domain is equivalent to convolution of coefficients in the time-domain; therefore, the above expression is obtained by simply convolving the water-confined reverberation spike train with itself. The event of interest in the above infinite series is the number “1”, which represents the primary reflection from a deep reflecting horizon. The rest of the events at later times are reverberations. In order to eliminate the reverberations, then, we must convolve the deep-reflection reverberation spike train with the inverse filter whose z -transform is given by

$$(1 + r z^n)^2$$

or

$$1 + 2rz^n + r^2 z^{2n} \quad (8.36)$$

The impulse response of the deep-reflection reverberation spike train deconvolution filter for $n = 10$ is

$$1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2r, 0, 0, 0, 0, 0, 0, 0, 0, r^2 \quad (8.37)$$

In the time domain, the impulse response of the deep-reflection reverberation spike train for a water depth of 30 meters is

$$\begin{aligned} 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2r, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3r^2, \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, -4r^3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5r^4, 0, 0, 0, \dots \end{aligned}$$

Deconvolution of this time series with the 3-point inverse filter of (8.37) removes the reverberations.

Throughout this discussion of a wave guide we have thought of the Dirac delta function as a source wavelet. There is therefore no overlap between the

primary and successive multiple reflections. The “wavelet” is already ideally compressed, and we have been able to identify each individual well-separated reverberation. The reverberating spike train, however, is convolved with an arbitrary wavelet that leaves the source and so becomes a much longer “effective source wavelet” as shown in Figure 8.10. The advantage of using an impulse as a source wavelet is that the inverses of the reverberation functions, that is, the inverses of the water-confined reverberation spike train and the deep reflection reverberation spike train are easily and quickly derived; however, the same approach can be used for any wavelet, regardless of its delay characteristics. If we know what the source wavelet looks like, then we can find its inverse by the methods described earlier. That is, instead of requiring the inverse of Equation (8.34), we would require the inverse of the convolution of (8.34) with the “effective source wavelet” wavelet. If the wavelet were truly time-invariant, then the reverberations would be time-invariant, and we could find the exact inverse of the deep-reflection reverberation *wave train*. Convolution of this inverse with the observed seismogram would then yield the reflection coefficient function, and we would have a better look at the geology; however, it is difficult to obtain good approximations to inverses. The determination of inverse filters for general source wavelets or for general reverberation functions is approached through the *autocorrelation function*. We will explore later other methods of determining inverse filters when we discuss *predictive deconvolution* in Section 8.3.2 on page 454, but all the methods depend upon being able to get a reliable estimate of the autocorrelation function of the wavelet, or of the reverberation wave train, in order to obtain the inverse. Predictive deconvolution (See 454) is one approach to the design of an inverse filter in order to increase the resolution of the seismogram.

For a more realistic example convolve a wavelet with the spike series given by (8.35). Then deconvolution of this seismogram with the 3-point inverse filter of (8.37) shortens the reverberation wavelet—the effective source wavelet—to the non-reverberation wavelet but does not shorten the basic wavelet itself. The filter reshapes the reverberation wavelet to the basic wavelet. Convolution of the 3-point inverse filter of Equation (8.36) with the deep-reflection reverberation effective wavelet removes the wave guide reverberations as shown in Figure 8.19.

The impulse response of the deep-reflection reverberation spike train deconvolution filter is

$$1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2r, 0, 0, 0, 0, 0, 0, 0, 0, r^2$$

This filter, when convolved with the deep-reflection reverberation spike train, eliminates the reverberations caused by the primary reverberating inside the wave guide *before* deep reflections arrive, and those caused by the reflected source wavelet reverberating in the wave guide after the wavelet reenters the water layer. It is apparent that we need not have restricted our source wavelet to the shape of the ideally compressed Dirac delta function. Application of the above simple 3-point operator will remove the reverberations of *any* shape wavelet, but it will not affect the shape of the primary wavelet itself. The

primary wavelet can be minimum-, mixed- or maximum-delay as shown in Figures 8.19 and 8.20.

Neidell [121] extended the reverberation problem to include a second layer and showed that a 4-point filter was required to remove reverberations. His formulation for the z -transform of the reverberating spike train associated with two layers is:

$$S(z) = \frac{(r_1 + r_2 z^n) z^m}{1 + r_1 z^m + r_1 r_2 z^{n+m} + r_2 z^{n+m}} \quad (8.38)$$

This formulation does not include the impulsive primary ("1" at $t = 0$). If (8.38) is the z -transform of the reverberating spike train, then clearly multiplication of Equation (8.38) by the 4-point z -transform in the denominator will yield the non-reverberating seismogram (the numerator). Neidell called the 4-point deconvolution filter the "Koehler 4-point operator" [99].

Neidell's two-layer annotated model is shown in Figure 8.21. The reader is referred to his excellent discussion for the derivation of Equation (8.38). A *Mathematica* program to generate the Neidell reverberating synthetics and then remove the reverberations using the 4-point Koehler filter is listed below. Three examples are shown here: the first to reproduce the example given by Neidell [121], which illustrates the severe nature of reverberations when large reflection coefficients are involved, a second with the same parameters but with a wavelet convolved with the spike train, and a third with smaller reflection coefficients to illustrate the nature of reverberations to be expected from ordinary geologic layering. In the section on predictive deconvolution (page 454) we will revisit the Backus 2-and 3-point filters and the Koehler 4-point filter and how they are accommodated in predictive deconvolution.

See the CD-ROM that accompanies this volume for a *Mathematica* program (Neidell1972Model) to compute synthetic seismograms for Neidell's two-layer model using the Koehler 4-point filter.

One application of the concept of filters in parallel as discussed on page 181 is the recording of a reverberating source wavelet coming in at the same time as a primary reflection.

$$\begin{aligned} z\text{-transform of the primary reflection} &= A(z) \\ z\text{-transform of the reverberation} &= G(z) \\ z\text{-transform of the total signal} &= Y(z) \\ Y(z) &= A(z) + G(z) \end{aligned}$$

If reverberations are added to $A(z)$ then these are just time-shifted and scaled (by the reflection coefficient r) versions of the wavelet $A(z)$ and we have

$$\begin{aligned} z\text{-transform of reverberating trace} &= z\text{-transform of wavelet} \\ &\quad + z\text{-transform of time-shifted} \\ &\quad \quad (by n \text{ time units}) \text{ and scaled wavelet} \\ Y(z) &= A(z) + z^n r A(z) \end{aligned}$$

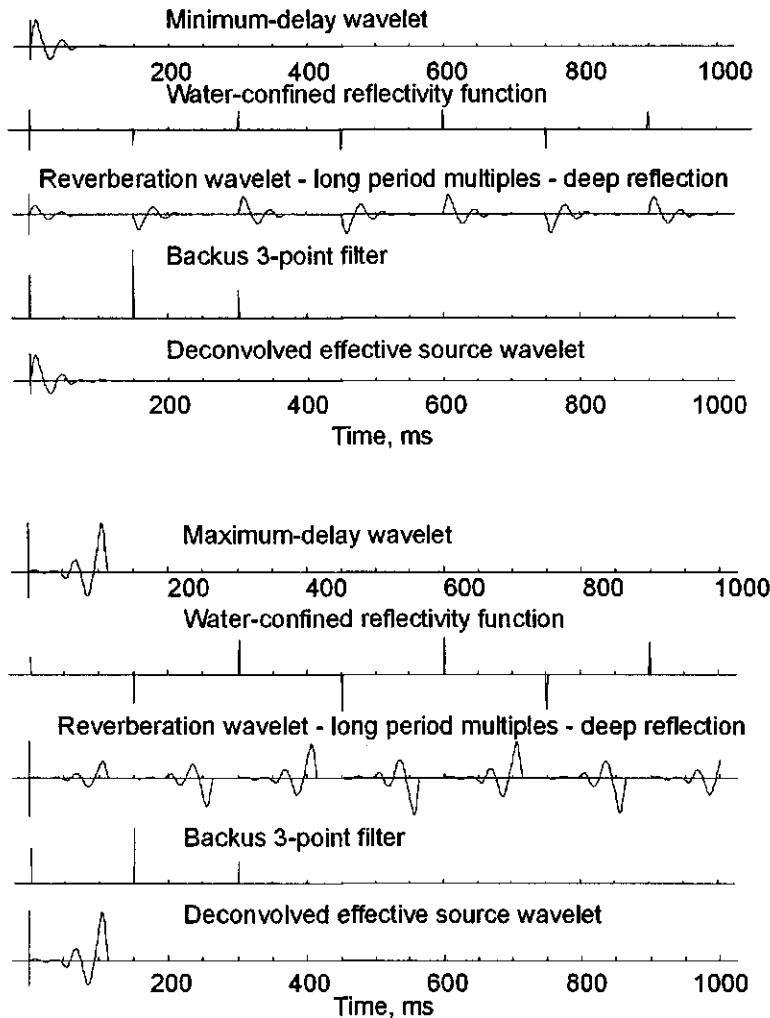


Figure 8.19: Top set: (a) Minimum-delay source wavelet. (b) Deep reflection reverberation spike train showing polarity reversals. (c) Convolution of source wavelet with spike train (the effective source wavelet). (d) The finite 3-point filter $1 + 2r z^n + r^2 z^{2n}$. (e) Deconvolved effective source wavelet. Bottom set: Same parameters but with a maximum-delay wavelet (the time-reverse of the above minimum-delay wavelet). The deconvolution is equally effective regardless of the delay characteristics of the wavelet.
BackusFilterExamples.nb,cdr → LPMultiplesDRRST.wmf

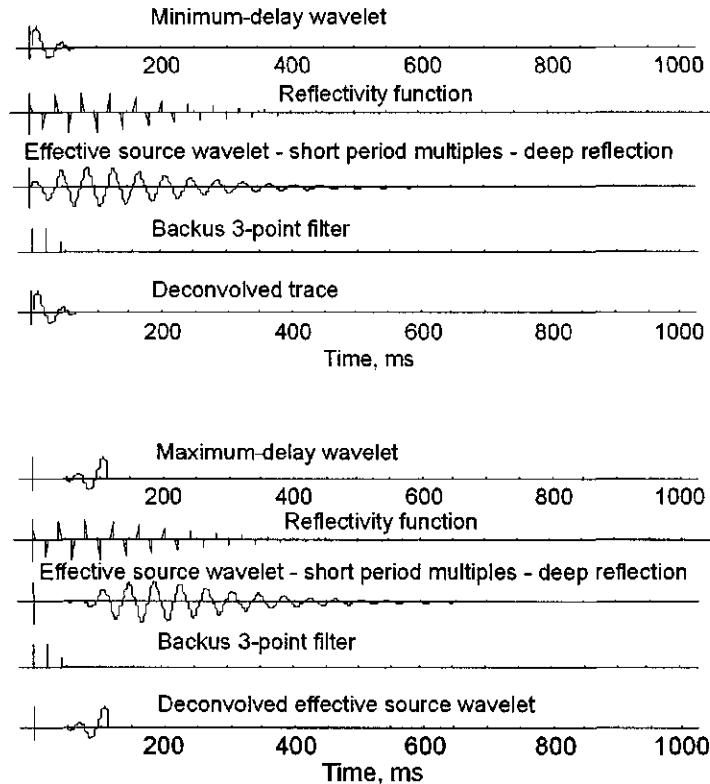


Figure 8.20: Top set: (a) Minimum-delay source wavelet. (b) Deep reflection reverberation spike train showing polarity reversals. (c) Convolution of wavelet with spike train (the effective source wavelet). (d) The finite 3-point filter $1 + 2r z^n + r^2 z^{2n}$. (e) Deconvolved effective source wavelet. Bottom set: Same parameters but with a maximum-delay wavelet (the time-reverse of the above minimum-delay wavelet). The deconvolution is equally effective regardless of the delay characteristics of the wavelet. PD-Multiples.nb → BackusFilterExamples.cdr → SPMultiplesDRRST.wmf

where r is a reflection coefficient and n is the time delay after arrival of reflection.

Is $Y(z)$ minimum phase? Note that $|z^n| = 1$ and $r < 1$ (because r is a reflection coefficient) so that $|z^n r A(z)|$ is less than $|A(z)|$ for all ω . Thus

$$\begin{aligned} Y(z) &= A(z) + z^n r A(z) \\ &= A(z) + G(z) \\ &= A(z) \left[1 + \frac{G(z)}{A(z)} \right] \end{aligned}$$

and $G(z) = z^n r A(z)$ is always $< A(z)$. Therefore, $Y(z)$ is minimum delay. A

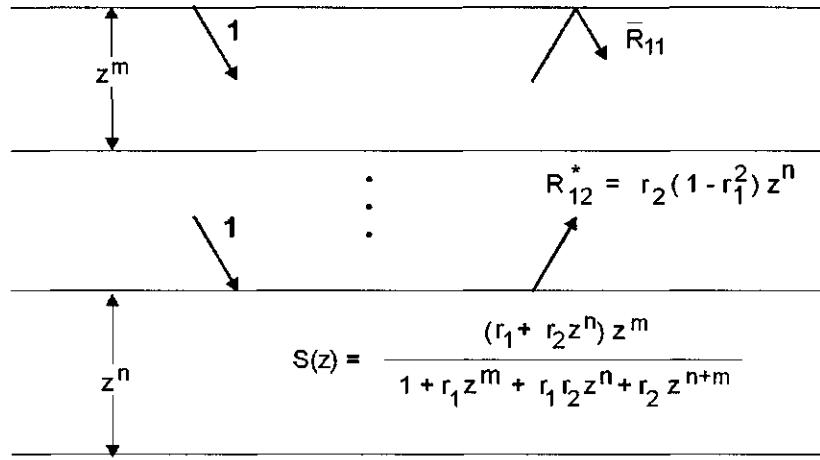


Figure 8.21: The 2-layer reverberation model of Neidell[121]. $S(z)$ is the z -transform of the 2-layer reverberation spike train. R_{12}^* is the “one-bounce” assumption made by Darby and Neidell [55] in the deeper layer. Neidell1972Model.nb,cdr → Neidell1972Model.wmf

minimum-phase reflection along with its echo is minimum phase.

Design of general inverse filters using z -transforms and partial fractions

This section follows directly from Robinson [148]. A continuous-time function $a(t)$ is defined for discrete time as $a_0, a_1, a_2, a_3, \dots, a_m$. Its z -transform $A(z)$ is

$$A(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \dots + a_m z^m = \sum_{m=0}^{\infty} a_m z^m$$

The coefficients a_0, a_1, a_2, \dots may represent the values of a filter impulse response, or the digitized values of a seismic wavelet for example, and may be real or complex constants. The subscript m denotes the time index.

As an example, the sampled function

$$a_t = (1, 3, 4, -2, -3, -1)$$

has the z -transform

$$A(z) = 1 + 3z + 4z^2 - 2z^3 - 3z^4 - z^5$$

Given a second function whose transform is

$$B(z) = b_0 + b_1 z + b_2 z^2 + b_3 z^3 + \dots + b_n z^n$$

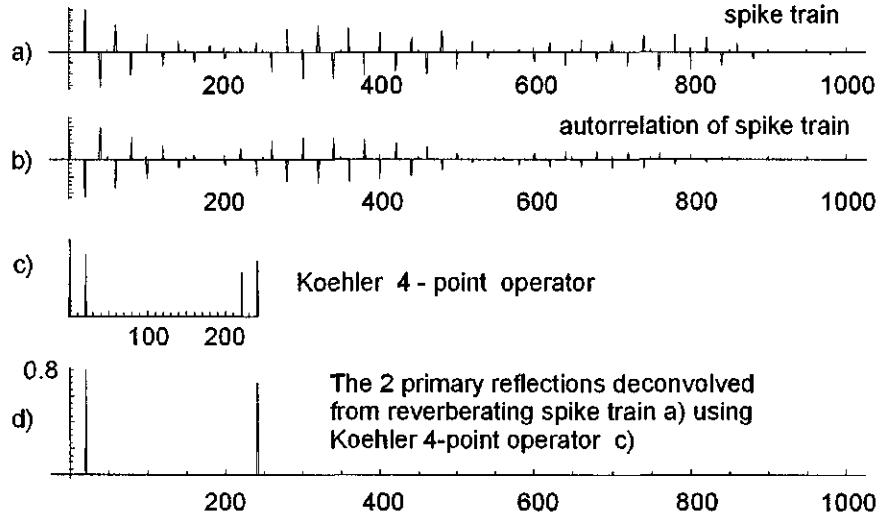


Figure 8.22: The 2-layer reverberation model of Neidell[121] with $m = 20$, $n = 220$, $r_1 = 0.8$, $r_2 = 0.7$. (a) Reverberating spike train. (b) Autocorrelation of spike train. (c) Koehler 4-point operator. (d) The deconvolution of (a). The 4-point Koehler filter removes all reverberations. The seismogram does not include the impulsive primary at $t = 0$. Neidell1972Model.nb,cdr → Neidell1972Model1.wmf

and taking the product of $B(z)$ and $A(z)$, then

$$\begin{aligned} C(z) = A(z)B(z) &= (a_0 + a_1z + \dots + a_mz^m)(b_0 + b_1z + \dots + b_nz^n) \\ &= c_0 + c_1z + c_2z^2 + \dots + c_{m+n}z^{m+n} \end{aligned}$$

where

$$\begin{aligned} c_0 &= a_0b_0 \\ c_1 &= a_0b_1 + a_1b_0 \\ &\vdots \\ c_k &= a_0b_k + a_1b_{k-1} + \dots + a_kb_0 \\ c_{m+n-1} &= a_{m-1}b_n + a_mb_{n-1} \\ c_{m+n} &= a_mb_n \end{aligned}$$

and $a_i = 0$ for $i > m$ and $b_j = 0$ for $j > n$.

It is apparent that multiplication of z -transform polynomials corresponds to convolution of their coefficients, because the convolution of two functions is defined as

$$= 0 \quad \text{for } k < 0$$

$$\begin{aligned}
 &= c_k = a_0 b_k + a_1 b_{k-1} + \cdots + a_k b_0 \quad \text{for } 0 \leq k \leq m+n \\
 &= 0 \quad \text{for } k > m+n
 \end{aligned}$$

and these coefficients, c_k , are exactly the same as those obtained above by polynomial multiplication.

The general objective of the design and use of inverse digital filters is to shorten the length of the seismic wavelet. (This would be important in identifying stratigraphic traps and reef development, for example.) We will identify $w_t = (w_0, w_1, w_2)$ with a seismic wavelet. We wish to find a filter f_t such that

$$W(z)F(z) = 1$$

Assume $W(z) = w_0 + w_1 z + w_2 z^2$. It is apparent that

$$\begin{aligned}
 F(z) &= \frac{1}{W(z)} \\
 &= \frac{1}{w_m(-z_1+z)(-z_2+z)}
 \end{aligned}$$

Now, expand $F(z)$ in terms of partial fractions. Note that

$$F(z)(-z_1+z) = \frac{1}{w_m(-z_2+z)}$$

Evaluate $\frac{1}{w_m(-z_2+z)}$ at $z = z_1$ and call it μ_1 ; i.e.,

$$\mu_1 = \frac{1}{w_m(-z_2+z_1)}$$

Then, similarly,

$$\mu_2 = \frac{1}{w_m(-z_1+z_2)}$$

and the expansion in terms of partial fractions is

$$F(z) = \frac{\mu_1}{(-z_1+z)} + \frac{\mu_2}{(-z_2+z)}$$

Check this by substituting

$$\begin{aligned}
 F(z) &= \frac{1}{w_m(-z_2+z_1)(-z_1+z)} + \frac{1}{w_m(-z_1+z_2)(-z_2+z)} \\
 &= \frac{(-z_1+z_2)(-z_2+z) + (-z_2+z_1)(-z_1+z)}{w_m(-z_2+z_1)(-z_1+z)(-z_1+z_2)(-z_2+z)} \\
 &= \frac{z z_2 - z z_1 - z_2^2 + z_1 z_2 + z z_1 - z z_2 - z_1^2 + z_1 z_2}{w_m(-z_2+z_1)(-z_1+z)(-z_1+z_2)(-z_2+z)} \\
 &= \frac{-z_1^2 + 2 z_1 z_2 - z_2^2}{w_m(-z_2+z_1)(-z_1+z)(-z_1+z_2)(-z_2+z)} \\
 &= \frac{(-z_2+z_1)(-z_1+z_2)}{w_m(-z_2+z_1)(-z_1+z)(-z_1+z_2)(-z_2+z)} \\
 &= \frac{1}{w_m(-z_1+z)(-z_2+z)}
 \end{aligned}$$

In general, then,

$$\begin{aligned}
 F(z) &= \frac{1}{W(z)} \\
 &= \frac{1}{w_m(-z_1+z)(-z_2+z)\cdots(-z_m+z)} \\
 &= \frac{\mu_1}{(-z_1+z)} + \frac{\mu_2}{(-z_2+z)} + \cdots + \frac{\mu_m}{(-z_m+z)} \\
 &= \mu_1(-z_1+z)^{-1} + \mu_2(-z_2+z)^{-1} + \cdots + \mu_m(-z_m+z)^{-1}
 \end{aligned} \tag{8.39}$$

and we have expanded $F(z)$ in terms of partial fractions.

The roots $z_1, z_2, z_3, \dots, z_m$ will define either maximum or minimum delay binomials depending on whether they lie inside or outside, respectively, of the unit circle.

We can now apply the binomial theorem to each of the $(-z_i+z)^{-1}$ terms in Equation (8.39). It is important that the binomial theorem be correctly applied. If it is not, then “instability” results, and the filter “blows up”. Instability can always be avoided by correct application of the binomial theorem.

Let us assume that all partial fractions in Equation (8.39) are associated with roots z_j that lie outside of the unit circle. Apply the binomial theorem to the general term in the partial fraction expansion of $F(z)$ above under the assumption that $|z_j| > 1$; i.e., we want to expand $\mu_j(-z_j+z)^{-1}$. Rearranging so that we can properly apply the binomial theorem we get

$$\begin{aligned}
 \mu_j(-z_j+z)^{-1} &= -\mu_j z_j^{-1} \left(1 - \frac{z}{z_j}\right)^{-1} \\
 &= \mu_j z_j^{-1} \left[1 + (-1) \left(\frac{z}{z_j}\right) + \frac{(-1)(-2)}{2!} \left(\frac{-z}{z_j}\right)^2 \right. \\
 &\quad \left. + \frac{(-1)(-2)(-3)}{3!} (-z_j)^3 + \cdots\right] \\
 &= -\mu_j [z_j^{-1} + z_j^{-2}z + z_j^{-3}z^2 + \cdots]
 \end{aligned}$$

The proper expansion of the binomial has resulted in positive powers of z . For the expansion of the j -th term, then, we have

$$\begin{aligned}
 \mu_j(z-z_j)^{-1} &= -\mu_j z_j^{-1} - \mu_j z_j^{-2}z - \mu_j z_j^{-3}z^2 - \cdots \\
 &= c_{0,j} + c_{1,j}z + c_{2,j}z^2 + \cdots
 \end{aligned}$$

where

$$\begin{aligned}
 c_{0,j} &= -\mu_j z_j^{-1} \\
 c_{1,j} &= -\mu_j z_j^{-2} \\
 c_{2,j} &= -\mu_j z_j^{-3} \\
 &\vdots
 \end{aligned}$$

Since

$$F(z) = \sum_{j=1}^m \frac{\mu_j}{(-z_j + z)}$$

then the coefficients of the inverse filter f_t ($= f_0, f_1, f_2, \dots$) whose z -transform is

$$f_0 + f_1 z + f_2 z^2 + \dots$$

are obtained by summing the coefficients of like powers of z from the expansion of each partial fraction term and then equating coefficients of like powers on each side of the equation:

$$\begin{aligned} f_0 + f_1 z + f_2 z^2 + \dots &= (-\mu_1 z_1^{-1} + \mu_1 z_1^{-1} z - \mu_1 z_1^{-3} z^2 - \dots) \\ &+ (-\mu_2 z_2^{-1} - \mu_2 z_2^{-1} z - \mu_2 z_2^{-3} z^2 - \dots) \\ &+ (-\mu_3 z_3^{-1} - \mu_3 z_3^{-1} z - \mu_3 z_3^{-3} z^2 - \dots) \end{aligned}$$

No negative powers of z occur in the expansion. Filter coefficients f_t will therefore be defined only for zero and positive time. So,

$$\begin{aligned} f_0 &= -\mu_1 z_1^{-1} + \mu_2 z_2^{-1} - \mu_3 z_3^{-1} - \dots \\ &= -\sum_{j=1}^m \mu_j z_j^{-1} \\ f_1 &= -\mu_1 z_1^{-1} + \mu_2 z_2^{-1} - \mu_3 z_3^{-1} - \dots \\ &= -\sum_{j=1}^m \mu_j z_j^{-1} \end{aligned}$$

Or, in general:

$$f_t = -\sum_{j=1}^m \mu_j z_j^{-t-1} \text{ for } t \geq 0$$

The coefficients f_t are the coefficients of an inverse filter which, when convolved with the source wavelet w_t , will reshape the wavelet to a spike at $t = 0$. The inverse filter f_t will only spike out minimum delay source wavelets because each term in the partial fraction expansion of $F(z)$ is associated with a minimum delay binomial.

If $F(z)$ is the z -transform of a *maximum delay* polynomial then proper application of the binomial theorem will again give the correct inverse filter.

For a maximum delay polynomial:

$$\begin{aligned} F(z) &= \frac{1}{W(z)} \\ &= \frac{\mu_1}{(-z_1 + z)} + \frac{\mu_2}{(-z_2 + z)} + \dots + \frac{\mu_m}{(-z_m + z)} \end{aligned}$$

The magnitude of each of the roots is less than one; therefore proper application of the binomial theorem requires that we expand each partial fraction in terms of *negative* powers of z . We can rewrite $F(z)$ as

$$\begin{aligned} F(z) &= \frac{\mu_1}{(-z_1 + z)} + \frac{\mu_2}{(-z_2 + z)} + \frac{\mu_3}{(-z_3 + z)} + \cdots + \frac{\mu_j}{(-z_j + z)} \\ &= \mu_1(-z_1 + z)^{-1} + \mu_2(-z_2 + z)^{-1} + \mu_3(-z_3 + z)^{-1} \\ &\quad + \cdots + \mu_j(-z_j + z)^{-1} \end{aligned}$$

The expansion of each maximum delay polynomial is

$$\begin{aligned} \mu_j (-z_j + z)^{-1} &= \mu_j \left[z^{-1} \left(1 - \frac{z_j}{z} \right)^{-1} \right] \\ &= \mu_j \left[z^{-1} \left(1 + \frac{z_j}{z} + \frac{(-1)(-2)}{2!} \left(\frac{-z_j}{z} \right)^2 + \frac{(-1)(-2)(-3)}{3!} \left(\frac{-z_j}{z} \right)^3 + \cdots \right) \right] \\ &= \mu_j \left(z^{-1} + z_j z^{-2} + z_j^2 z^{-3} + \cdots \right) \\ &= \mu_j z^{-1} + \mu_j z_j z^{-2} + \mu_j z_j^2 z^{-3} + \cdots \end{aligned}$$

We have expanded in negative powers of z ; therefore, the filter coefficients are defined for negative time only. For all times $t > 0$, the coefficients are zero. That is,

$$\begin{aligned} d_{-1,j} &= \mu_j \\ d_{-2,j} &= \mu_j z_j \\ d_{-3,j} &= \mu_j z_j^2 \\ &\vdots \end{aligned}$$

The coefficients of the inverse filter, f_t , are obtained as before by summing coefficients of like powers of z from the expansion of each partial fraction term, and then equating coefficients of like power on each side of the equation.

$$\begin{aligned} f_{-1} z^{-1} + f_{-2} z^{-2} + f_{-3} z^{-3} + \cdots &= (\mu_1 z^{-1} + \mu_1 z_1 z^{-2} + \mu_1 z_1^2 z^{-3} + \cdots) \\ &= +(\mu_2 z^{-1} + \mu_2 z_2 z^{-2} + \mu_2 z_2^2 z^{-3} + \cdots) \\ &= +(\mu_3 z^{-1} + \mu_3 z_3 z^{-2} + \mu_3 z_3^2 z^{-3} + \cdots) \\ &= +\cdots \end{aligned}$$

All coefficients corresponding to zero and positive time must be zero because no terms in positive powers of z exist on the right-hand side of the equation. Thus,

$$\begin{aligned} f_{-1} = \mu_1 + \mu_2 + \mu_3 + \cdots &= \sum_{j=1}^m \mu_j \\ f_{-2} = \mu_1 z_1 + \mu_2 z_2 + \mu_3 z_3 + \cdots &= \sum_{j=1}^m \mu_j \end{aligned}$$

or, in general,

$$f_t = \sum \mu_j z_j^{-t-1} \quad \text{for } t \leq 0$$

If $F(z)$ is the z -transform of a mixed delay polynomial then the partial fraction expression for $F(z)$ will contain some terms that must be expanded in positive powers of z and some that must be expanded in negative powers of z .

In summary (Robinson [148]), given a source wavelet w_t then the inverse operators f_t that will reshape this wavelet to a spike at $t = 0$ are given by

Minimum delay

$$\begin{aligned} f_t &= -\sum_{j=1}^h \mu_j z_j^{-t-1} \quad \text{for } t \geq 0 \\ f_t &= 0 \quad \text{for } t < 0 \end{aligned} \tag{8.40}$$

where the roots z_1, z_2, \dots, z_h have a modulus greater than unity; i.e., they lie outside of the unit circle.

Mixed delay

$$\begin{aligned} f_t &= -\sum_{j=1}^h \mu_j z_j^{-t-1} \quad \text{for } t \geq 0 \\ f_t &= \sum_{j=h+1}^m \mu_j z_j^{-t-1} \quad \text{for } t < 0 \end{aligned} \tag{8.41}$$

where the roots z_1, z_2, z_h have magnitude greater than unity, and the roots $z_{h+1}, z_{h+2}, \dots, z_m$ have magnitude less than unity.

Maximum delay

$$\begin{aligned} f_t &= \sum_{j=h+1}^m \mu_j z_j^{-t-1} \quad \text{for } t < 0 \\ f_t &= 0 \quad \text{for } t \geq 0 \end{aligned} \tag{8.42}$$

In summary, we require an inverse filter that exists only in the present and past for a minimum delay input, in the present, past, and future for a mixed-delay input, and in the future for a maximum delay input. The part of the filter that is defined in negative time is called the “anticipation component”. The part defined for zero and positive time is called the “memory component”.

The coefficients for the general inverse filter for a mixed-delay wavelet can be found by using a subroutine like **INVSMM**, a Fortran version of which has been

included on the CD with this volume. The program computes the minimum-delay and maximum-delay coefficients associated with the factored z -transform polynomial. You must *convolve* the stable memory inverse and the stable anticipation inverse together before applying the filter in the time domain.

8.3 Inverse filters and input each of finite length

The third and last subdivision of deconvolution filters listed on page 372 includes those inverse filters of finite length and input of finite length. Previous discussions have been concerned with

1. Input data (wavelets) of finite length and filters of infinite length
2. Filters of finite length and input data (water reverberations) of infinite length

It was shown that if the desired output is compared with the actual output, then exact results could be obtained; however, either the filter had to be of infinite length, or the input data had to be of infinite length to get exact results. The reader should review those “delay” properties of the input that were required for proper computation of the filter coefficients and therefore convergence of the desired output. That is, should filter coefficients be computed for zero and positive time (minimum-delay), for just negative time (maximum-delay), or for both positive and negative times (mixed-delay—see Section 8.2.1). This is clearly not a practical consideration because neither inputs nor filters are of infinite length. We must therefore consider input time series as well as filters that are each of finite length. Although the present discussion is concerned with finite-length time series (input, filter coefficients, and desired finite-length output), the same requirements for convergence of the filtered output hold and, indeed, as the filter length approaches infinity, we will converge to the same numerical values that were obtained in, for example, Sections 8.1.1, 8.2, and 8.2.1. The criteria for the design of a digital filter of finite length is that it operate on some input of finite length and produce an output of finite length that is “as close as possible” to the desired output. Accomplishing this goal is commonly done by the “method of least squares”.

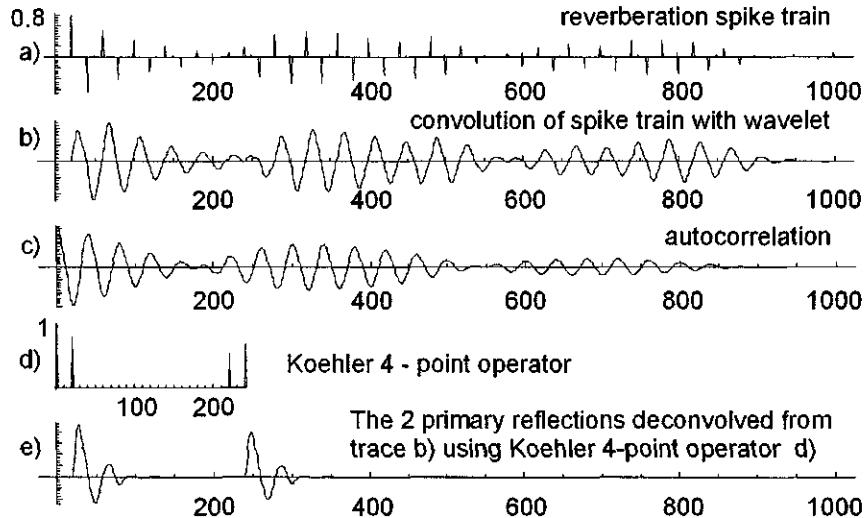


Figure 8.23: Same as Figure 8.22 but spike train convolved with a wavelet. The 4-point Koehler filter removes all reverberations leaving only the two primary reflections. Seismogram does not include the impulsive primary at $t = 0$. Neidell1972Model.nb,cdr → Neidell1972Model2.wmf

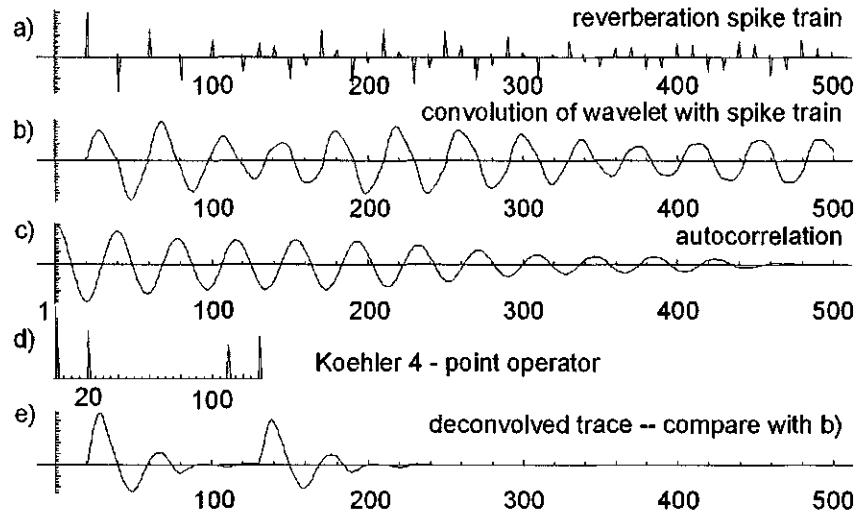


Figure 8.24: Two-layer reverberation model using the example of Neidell [121] with $m = 20$, $n = 110$, $r_1 = 0.8$, $r_2 = 0.7$ and minimum-delay wavelet. Seismogram does not include primary source wavelet at $t = 0$. The 4-point Koehler filter removes all reverberations leaving only the two primary reflections, but does not affect the shape of the primary source wavelet itself. Any wavelet with any delay characteristics can be used with this filter; wavelet does not have to be minimum-delay. Neidell1972Model.nb,cdr → Neidell1972Model3.wmf

8.3.1 General shaping and least-squares method

The name least-squares comes from the design of a filter that will minimize the difference (error) between an input signal and some desired output signal. For example, we might want to

- Shape a seismic wavelet (the input) to an impulse (the desired output),
- Remove reverberations from a seismic trace,

where the desired output is, for these examples, an impulse or a sequence of reflection coefficients (the geology), respectively.

The concepts underlying the method of least squares can be illustrated by fitting a set of n points to a straight line. The equation of a straight line is

$$y = mx + b \quad (8.43)$$

where m and b are constants. Only two points such as (x_1, y_1) and (x_2, y_2) , however, are required to define a straight line. So given the n points

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

it will in general not be possible for the straight line to pass through all of the points (it probably will not pass through any). That is, the general point (x_i, y_i) will not fall on the straight line defined by Equation (8.43). There will be a difference δ_i , which is

$$\delta_i = y_i - (mx_i + b) \neq 0$$

If the difference δ_i is determined for each point of the set

$$(x_1 y_1), (x_2 y_2), \dots, (x_n y_n)$$

then these differences can be squared so that large positive values of δ_i will not cancel with large negative values of δ_i , for unless the differences are squared then an unjustified impression of accuracy would result. So the total error E^2 becomes

$$E^2 = \sum_{i=1}^n \delta_i^2 = (y_1 - b - mx_1)^2 + (y_2 - b - mx_2)^2 + \dots + (y_n - b - mx_n)^2$$

where E^2 is some measure of how well the straight line, which is defined by the constants m and b , fits the set of points. If $E^2 = 0$ then all the points lie on the straight line. The larger the value of E^2 the farther the points lie from the straight line. The least-squares criterion is simply that the constants m and b be chosen such that E^2 is as small as possible, i.e., a minimum. This is accomplished in the usual way by requiring that

$$\frac{\partial E^2}{\partial m} = \frac{\partial E^2}{\partial b} = 0$$

Thus,

$$\begin{aligned}\frac{\partial E^2}{\partial b} &= 2(y_1 - b - mx_1)(-1) + 2(y_2 - b - mx_2)(-1) + \dots \\ &\quad + 2(y_n - b - mx_n)(-1) \\ &= 0\end{aligned}$$

and

$$\begin{aligned}\frac{\partial E^2}{\partial m} &= 2(y_1 - b - mx_1)(-x_1) + 2(y_2 - b - mx_2)(-x_2) + \dots \\ &\quad + 2(y_n - b - mx_n)(-x_n) \\ &= 0\end{aligned}$$

which reduces to

$$nb + m \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \quad (8.44)$$

and

$$b \sum_{i=1}^n x_i + m \sum_{i=1}^n x_i^2 = x_i y_i \quad (8.45)$$

Equations (8.44) and (8.45) constitute a set of linear equations that can be solved for the constants m and b .

The same conceptual approach can be used to design a filter that will convert –in an optimal least-squares sense –an input to a desired output. The filter coefficients, f_i , are determined such that the difference squared between the actual output, y_i , and the desired output, x_i , is a minimum. That is,

$$J = E(d_i - y_i)^2$$

is a minimum. The notation E denotes the average value of whatever quantity is in the braces.

The requirements for convergence of filtered output using a non-least-squares filter are:

- Changing the shape of a minimum-delay wavelet requires a filter of infinite length. The filter coefficients are defined only for zero and positive time.
- Changing the shape of a maximum-delay wavelet requires a filter of infinite length. The filter coefficients are defined only for negative time.
- Changing the shape of a mixed-delay wavelet requires a filter of infinite length. The filter coefficients are defined for negative, zero, and positive time.

Exactly the same conditions for convergence of filtered output are required for a least-squares filter, f_i , of finite length. That is,

- Changing the shape of a minimum-delay wavelet requires a filter whose coefficients are defined only for zero and positive time.
- Changing the shape of a maximum-delay wavelet requires a filter whose coefficients are defined only for negative time.
- Changing the shape of a mixed-delay wavelet requires a filter whose coefficients are defined for negative, zero, and positive time.

The length of the least-squares filter is now the additional parameter that must be considered. A common rule of thumb for many applications is that the filter length (the time duration of the filter) should be about the same length as the input wavelet on which the filter operates. A seismic trace is the convolution of a wavelet with a sequence of reflection coefficients; therefore, to reshape the seismic wavelet the length of the filter should be about the same length as the duration of the wavelet. Because the autocorrelation of a wavelet can only be as long as the wavelet, the filter of finite length can be taken to be about the same as the duration of the autocorrelation function. It should be remembered, however, that theoretically there can be only one correct filter, the one that is of infinite length; therefore, the filter of finite length is only "best" in a "least-squares" sense.

Let the filter, f_t , be defined by $m + 1$ equally-spaced coefficients. Then the actual output, c_t , is the convolution of the input, b_t with the filter:

$$c_t = \sum_{\tau=0}^m f_\tau b_{t-\tau}$$

The difference between the desired output, d_t , and the actual output, c_t , is $d_t - c_t$. The error squared is

$$J = E(d_t - \sum f_\tau b_{t-\tau})^2 \quad (8.46)$$

The quantity, J , is minimized by taking the partial derivative of J with respect to each of the filter coefficients, f_i .

$$\begin{aligned} \frac{\partial J}{\partial f_1} &= E \left\{ 2 \left(d_t - \sum_{\tau=0}^m f_\tau b_{t-\tau} \right) \frac{\partial}{\partial f_1} \left(d_t - \sum_{\tau=0}^m f_\tau b_{t-\tau} \right) \right\} = 0 \\ &= 2 E \left\{ \left(d_t - \sum_{\tau=0}^m f_\tau b_{t-\tau} \right) (-b_{t-1}) \right\} = 0 \end{aligned}$$

$$E = 1 - \sum_{\tau=0}^m f_\tau \phi'_{xx}(\tau) \quad (8.47)$$

Note that

$$0 \leq E \leq 1$$

and is called the normalized mean square error [153].

In practice, all filters are of finite length. We know from earlier discussions that *any* wavelet, regardless of its shape, requires an inverse filter of infinite length to convert it to an impulse $\delta(t)$ at $t = 0$. This is not practical (or necessary), so we design filters that are finite and do the best we can in a least-squares sense. We also know that a reverberating wave train associated with a wave guide is of infinite time duration, but that the inverse filter is a simple one of finite length (the Backus 3-point filter, for example). It is not practical (or necessary) to record an infinitely long reverberating trace to design an exact inverse filter, so we record a seismogram of finite extent and design filters that are finite and do the best we can, in a least-squares sense. In between these extremes of filters of infinite length and data of infinite duration are the real-world filters and data of finite length. To remove reverberations from a seismic trace we should expect to be able to design a more perfect (finite) inverse filter the longer the recorded trace we have to work with. To convert any wavelet to an impulse at $t = 0$ we should expect to design a more perfect inverse filter the longer we allow it to become. The basic goal in the design of finite least-squares filters is to minimize the difference between the actual filter output y_t and the desired filter output d_t . Mathematically this can be stated as

$$J = E \left\{ (d_t - y_t)^2 \right\} = \text{a minimum} \quad (8.48)$$

where

$$E \left\{ (d_t - y_t)^2 \right\} = \sum_{t=0}^n (d_t - y_t)^2 = (d_0 - y_0)^2 + (d_1 - y_1)^2 + \dots$$

The quantity J is often referred to as a quantity proportional to the error energy or power of the wavelet or trace of length n . We want some filter f_t of length $m + 1$ that when convolved with the input x_t , will give the output y_t , which will be as close as possible in a least-squares sense to the desired output d_t . Convolution of the filter with the input data sequence x_t gives

$$y_t = \sum_{\tau=0}^m f_\tau x_{t-\tau} \quad \text{for } t = 0, 1, 2, \dots \quad (8.49)$$

Substituting (8.49) into Equation (8.48) we get

$$J = E \left\{ \left(z_t - \sum_{\tau=0}^m f_\tau x_{t-\tau} \right)^2 \right\} \quad (8.50)$$

The quantity J is minimized by taking the partial derivative of J with respect to each of the filter coefficients f_t equal to zero. That is, for f_1 , for example,

$$\begin{aligned}
 \frac{\partial J}{\partial f_1} &= E \left\{ 2 \left(z_t - \sum_{\tau=0}^m f_\tau x_{t-\tau} \right) \frac{\partial}{\partial f_1} \left(z_t - \sum_{\tau=0}^m f_\tau x_{t-\tau} \right) \right\} \\
 &= 2E \left\{ \left(z_t - \sum_{\tau=0}^m f_\tau x_{t-\tau} \right) (-x_{t-1}) \right\} \\
 &= 2E \left\{ -z_t x_{t-1} + \sum_{\tau=0}^m f_\tau x_{t-\tau} x_{t-1} \right\} \\
 &= 2 \left[-E \{ z_t x_{t-1} \} + \sum_{\tau=0}^m f_\tau E \{ x_{t-\tau} x_{t-1} \} \right] \\
 \frac{\partial J}{\partial f_1} &= 2 \left[-g_1 + \sum_{\tau=0}^m f_\tau r_{1-\tau} \right]
 \end{aligned} \tag{8.51}$$

where g_τ denotes the crosscorrelation of the input with the desired output and r_τ is the autocorrelation of the input. Reference to Figure 8.25 recalls the directions of relative shift between the input and desired output to obtain positive and negative lags. For positive lags, shift the input toward positive time with respect to the desired output.

Setting (8.51) equal to zero gives

$$\sum_{\tau=0}^m f_\tau r_{1-\tau} = g_1$$

Do this for every filter coefficient f_t . That is,

$$\sum_{\tau=0}^m f_\tau r_{j-\tau} = g_j \quad \text{for } j=0,1,2,\dots,m \tag{8.52}$$

and the result is a system of equations called the normal equations whose solution yields the filter coefficients f_t .

As an example of the matrix form of (8.52), we assume a filter of length $m-1$ whose coefficients are defined only for $t \geq 0$. Expanding (8.52) for $m = 6$ and for zero and positive values of t we get the simultaneous linear equations

$$\begin{array}{ccccccccc}
 r_0 f_0 & +r_1 f_1 & +r_2 f_2 & +r_3 f_3 & +r_4 f_4 & = & g_0 \\
 r_{-1} f_0 & +r_0 f_1 & +r_1 f_2 & +r_2 f_3 & +r_3 f_4 & = & g_1 \\
 r_{-2} f_0 & +r_{-1} f_1 & +r_0 f_2 & +r_1 f_3 & +r_2 f_4 & = & g_2 \\
 r_{-3} f_0 & +r_{-2} f_1 & +r_{-1} f_2 & +r_0 f_3 & +r_1 f_4 & = & g_3 \\
 r_{-4} f_0 & +r_{-3} f_1 & +r_{-2} f_2 & +r_{-1} f_3 & +r_0 f_4 & = & g_4
 \end{array}$$

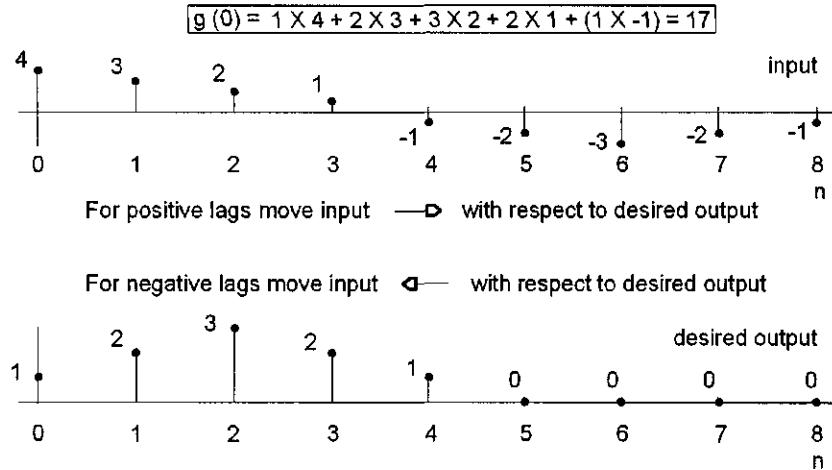


Figure 8.25: For positive-lag crosscorrelation coefficients $g(+\tau)$ shift the input to the right (toward positive time) with respect to the desired output. For negative-lag coefficients $g(-\tau)$ shift the input toward negative time with respect to the desired output. Numerical values used to obtain $g(0)$ shown at top of figure. See Figures 8.26 and 8.27 for coefficient values, for positive $g(+\tau)$, and negative $g(-\tau)$, lags, respectively. (xcorfig.nb → xcorfig.cdr → xcorfig.wmf)

which can be written in matrix form as

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r_4 \\ r_{-1} & r_0 & r_1 & r_2 & r_3 \\ r_{-2} & r_{-1} & r_0 & r_1 & r_2 \\ r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 \\ r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix}$$

and because $r(-\tau) = r(\tau)$ the above matrix equation becomes

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r_4 \\ r_1 & r_0 & r_1 & r_2 & r_3 \\ r_2 & r_1 & r_0 & r_1 & r_2 \\ r_3 & r_2 & r_1 & r_0 & r_1 \\ r_4 & r_3 & r_2 & r_1 & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix}$$

Wavelet shaping

In general, however, filter coefficients will need to be defined for both positive and negative time because of the requirements discussed earlier in this chapter for the convergence of the filter coefficients. These requirements do not change simply because we are designing least-squares filters. Deconvolution (inverse) filters for a minimum-delay input will require coefficients defined only for zero and

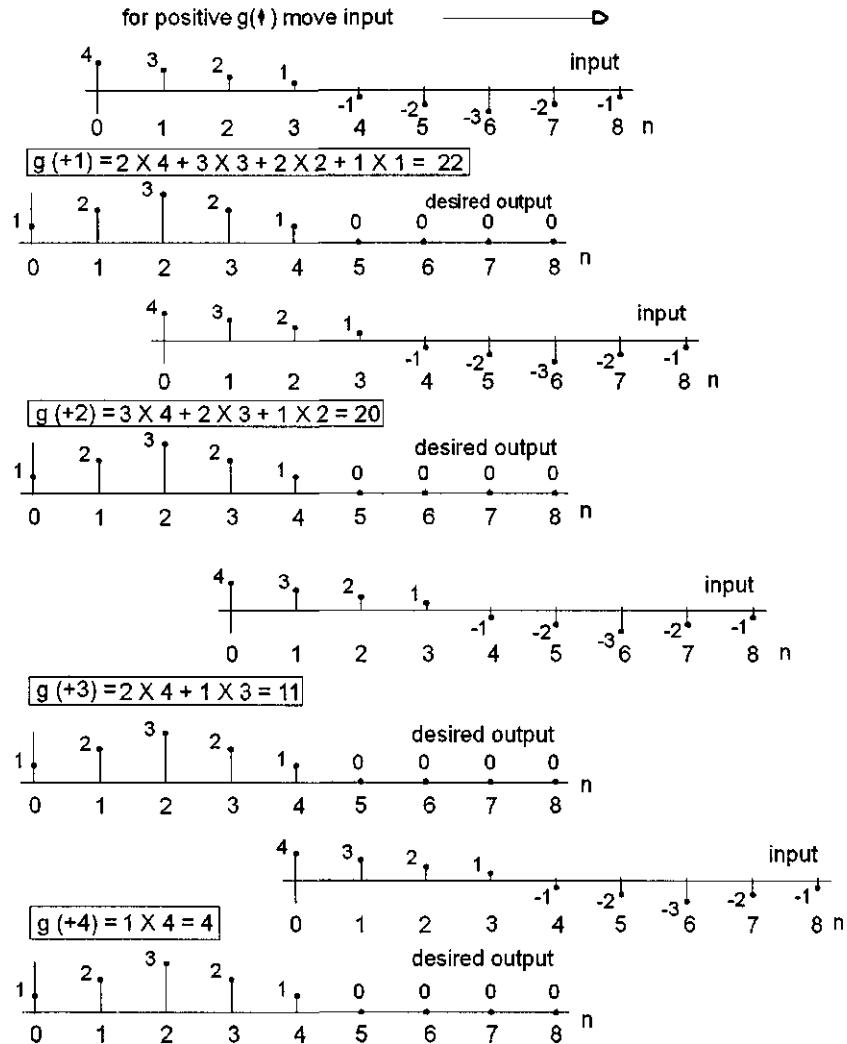


Figure 8.26: Procedure for obtaining positive-lag crosscorrelation coefficients. Relative positions of the two functions to be crosscorrelated are shown for lag $\tau = +1, +2, +3, +4$. Boxed areas show numerical cross-products, sums, and final answer for value of lag, τ . (xcorfig.nb → xcorfig.cdr → xcorfig2.wmf)

positive time. Filters operating on a maximum-delay input require coefficients defined only for negative time. Filters applied to a mixed-delay input require coefficients defined for both negative and positive time (see also Berkhouit [17]). These requirements hold no matter what the delay properties of the desired

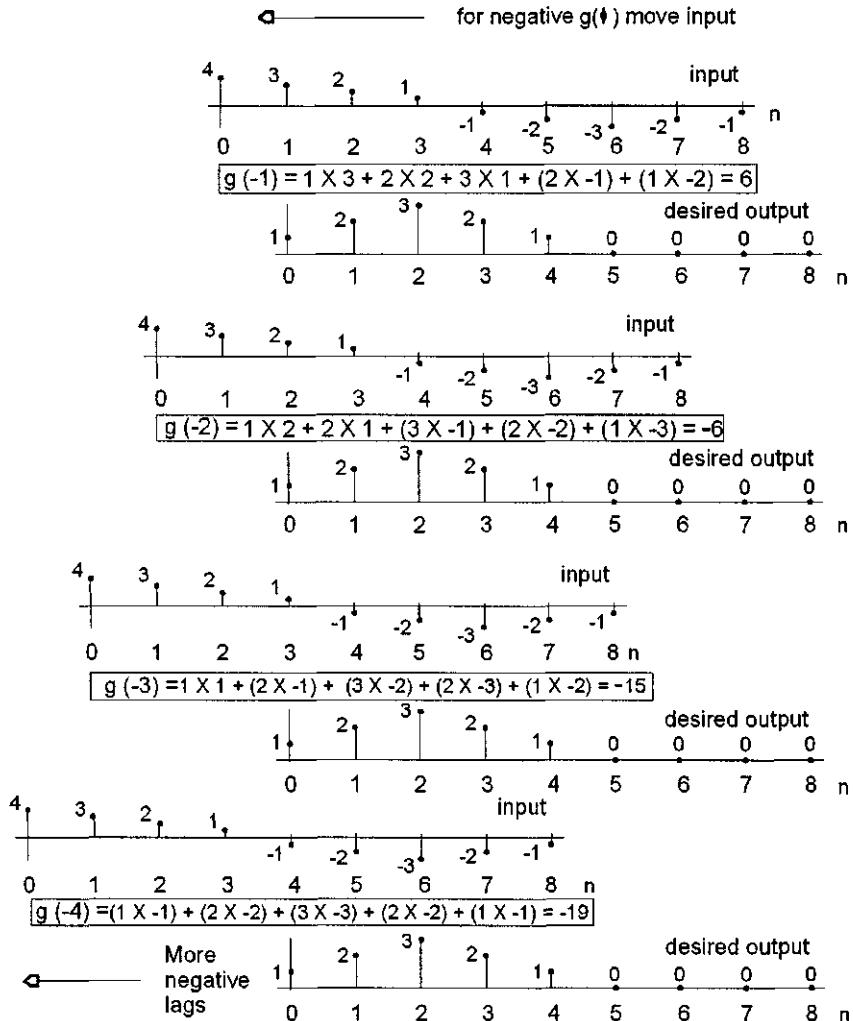


Figure 8.27: Procedure for obtaining negative-lag crosscorrelation coefficients. Relative positions of the two functions to be crosscorrelated are shown for lag $\tau = -1, -2, -3, -4$ only. Continue shifting the input to the left to obtain all of the negative-lag coefficients. Boxed areas show numerical cross-products, sums, and final answer for value of lag τ . (xcorfig.nb → xcorfig.cdr → xcorfig3.wmf)

output; they are independent of the desired output. This can be understood by realizing that shaping a general input to any desired output is, conceptually at least, a 2-step process: 1) convert the input to an impulse at $t = 0$ —this

is where the decision about negative and/or positive filter coefficients must be made—and 2) convolution of the impulse with the desired output, which clearly does not involve any consideration of convergence, no matter what the delay properties of the desired output. Thus, assuming that both the filter and the input are defined for both positive and negative time, Equation (8.52) becomes

$$\sum_{\tau=-m}^{+m} f_\tau r_{t-\tau} = g_j \quad \text{for } t = -n, \dots, -2, -1, 0, 1, 2, \dots, n \quad (8.53)$$

which for, say, $m = 3$, becomes

$$\begin{aligned} f_{-1}r_0f_0 & + r_1f_1 & + r_2f_2 & + r_3f_3 & + r_4f_4 & = g_0 \\ r_{-1}f_0 & + r_0f_1 & + r_1f_2 & + r_2f_3 & + r_3f_4 & = g_1 \\ r_{-2}f_0 & + r_{-1}f_1 & + r_0f_2 & + r_1f_3 & + r_2f_4 & = g_2 \\ r_{-3}f_0 & + r_{-2}f_1 & + r_{-1}f_2 & + r_0f_3 & + r_1f_4 & = g_3 \\ r_{-4}f_0 & + r_{-3}f_1 & + r_{-2}f_2 & + r_{-1}f_3 & + r_0f_4 & = g_4 \end{aligned}$$

For an extension of this single-channel discussion to multi-channel data the reader is referred to Treitel [184].

Examples

Figure 8.28 is redrafted from Robinson and Treitel's [153, Figure 6-2, p. 151] excellent discussion of examples of least-squares filtering.

Figure 6-2(a) from Robinson and Treitel

In Figure 8.28a, the input is a simple 2-point minimum-delay wavelet. The desired output is a unit impulse at $t = 0$. We know from section 8.2.1 that the filter length required to accomplish this must be infinitely long. If a filter length of only two points is chosen then we cannot expect a correct result. We wish to observe the improvement in the actual output as the filter length is increased. A *Mathematica* program to reproduce all of the examples of least-squares filtering shown in Figure 8.28 is given on page 438. The autocorrelation coefficients, r_τ , are

$$r_\tau = 5, 2 \text{ for } \tau = 0, 1 \text{ respectively}$$

The desired output d_t is

$$d_t = 1, 0 \text{ for } t = 0, 1$$

In matrix notation,

$$r = \begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix}$$

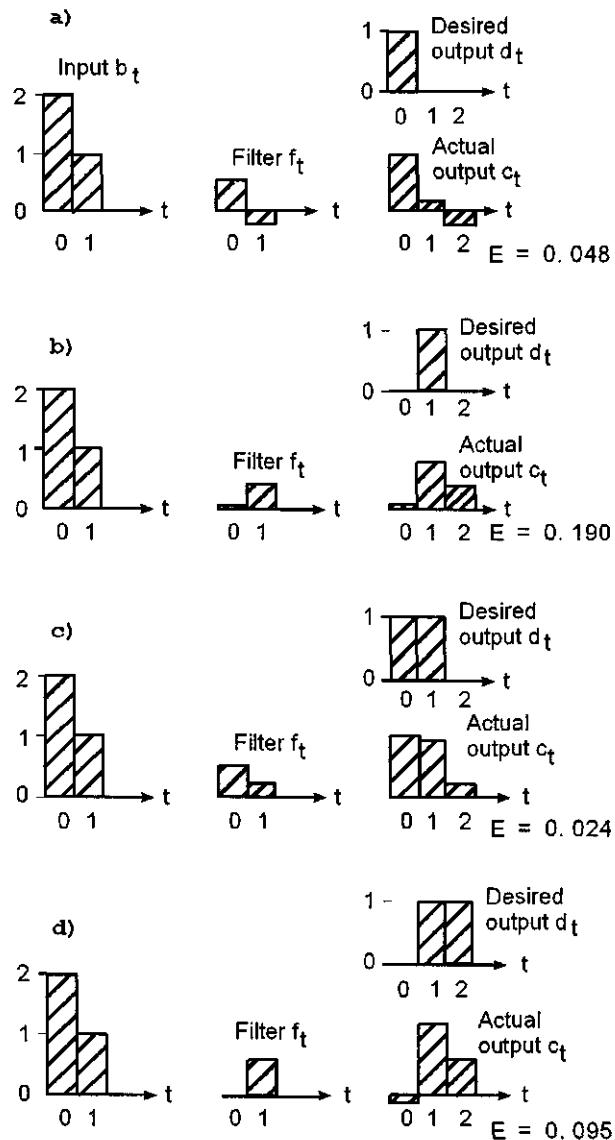


Figure 8.28: Figure 6-2 reproduced from Robinson and Treitel [153]. Left column: Input wavelet. Middle column: Computed filter coefficients using Equation (8.52). Right column: Desired and actual output. The quantity E is the minimum error energy given by Equation (8.47). (Figure6-2a1RobinsonAndTreitel.nb and Figure62FromRobinsonAndTreitel.cdr,wmf)

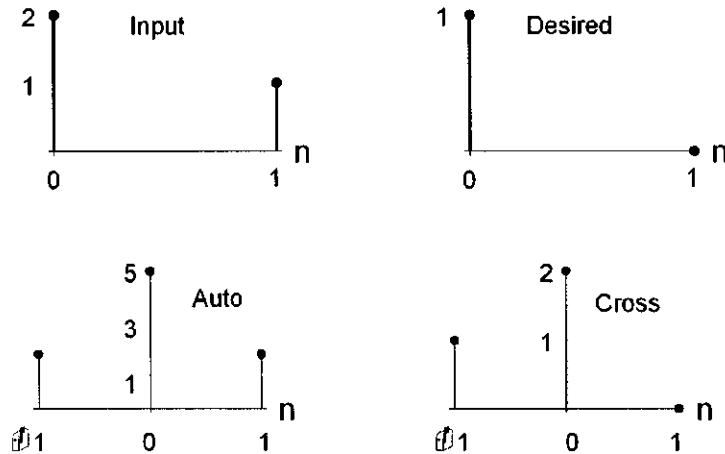


Figure 8.29: Autocorrelation of input and crosscorrelation of desired output with input for example (a) of Figure 8.28 See page 438 for the *Mathematica* program.
(RT-Fig6-2.nb,cdr → RT-Fig6-2a.wmf)

$$g = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

and the matrix equation is

$$\begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad (8.54)$$

Solving for the vector of filter coefficients $f[n]$ we obtain

$$f[n] = \begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \frac{10}{21}, \frac{-4}{21} \text{ for } n = 0, 1 \text{ respectively}$$

Convolution of the filter $f[n]$ with the wavelet $b[n]$ gives

$$f[n] * b[n] = \sum_{m=-\infty}^{m=+\infty} f[m] b[n-m] = (0.952381, 0.0952381, -0.190476) \text{ for } n = 0, 1, 2$$

If we want a more accurate result then it will be necessary to increase the length of the filter. For a desired output of

$$d[n] = 1, 0, 0, 0, 0, 0$$

the autocorrelation matrix, r , takes the form

$$r = \begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 0 & 2 & 5 \end{pmatrix}$$

the crosscorrelation vector $g[n]$ becomes

$$g = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and the matrix equation is

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 0 & 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Solving for $f[n]$ we obtain

$$\begin{aligned} f[n] &= \begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 0 & 2 & 5 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ &= (0.499634, -0.249084, 0.123077, -0.0586081, 0.0234432) \end{aligned}$$

and convolution of this longer filter with the wavelet results in

$$\begin{aligned} f[n] * b[n] &= \sum_{m=-\infty}^{m=+\infty} f[m] b[n-m] \\ &= (0.9992, 0.0014, -0.0029, 0.0058, -0.0117, 0.0234) \\ &\text{for } n = 0, 1, 2, 3, 4, 5 \end{aligned}$$

which, as expected, is a better desired output. The error energy is now

$$E = 1 - \sum_{\tau=0}^5 f_\tau \phi'_{db}(\tau) = 0.000183$$

Increasing the filter length to just 15 points results in the matrix equation

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (8.55)$$

with an error energy E now equal to

$$E = 1 - \sum_{\tau=0}^{14} f_\tau \phi'_{db}(\tau) = 6.9810^{-10}$$

and an actual output c of

$$c = (1., 1.3969810^{-9}, -2.7939710^{-9}, 5.5879410^{-9}, -1.1175910^{-8}, 2.2351710^{-8}, -4.4703510^{-8}, 8.940710^{-8}, -1.7881410^{-7}, 3.5762810^{-7}, -7.1525610^{-7}, 1.4305110^{-6}, -2.8610210^{-6}, 5.7220510^{-6}, -0.0000114441, 0.0000228882)$$

Although the filter is only 15 points long, it is interesting to compare the numerical values of the least-squares filter coefficients with those we know to be exactly correct. The least-squares filter values are

$$f[n] = (0.5, -0.25, 0.125, -0.0625, 0.03125, -0.015625, 0.00781247, -0.00390619, 0.00195301, -0.000976324, 0.000487804, -0.000243187, 0.000120163, -0.0000572205, 0.0000228882)$$

The exact filter coefficients can be obtained using z -transform theory as described in Section 8.2.1. We want filter coefficients, $f[n]$, such that

$$f[n] * b[n] = \sum_{m=-\infty}^{m=+\infty} f[m] b[n-m] = \delta[t] \text{ for } n = 0, 1, 2$$

where $\delta[t] = 1$ for $t = 0$ and $= 0$ for $t \neq 0$

In z -transform notation,

$$F(z) B(z) = 1$$

Thus, the z -transform of the filter is given by

$$F(z) = \frac{1}{B(z)}$$

For the 2-point wavelet $(2, 1)$, its z -transform is

$$B(z) = 2z^0 + 1z^1 = 2 + z$$

and

$$F(z) = \frac{1}{2+z} = (2+z)^{-1}$$

The Mathematica function *Series* can be used to find the power series expansion:

```
Series[(2+z)^(-1),{z,0,15}]//N
```

and the output is the z -transform of the filter, which is seen to be

$$\begin{aligned} & 0.5 - 0.25z + 0.125z^2 - 0.0625z^3 + 0.03125z^4 - 0.015625z^5 \\ & + 0.0078125z^6 - 0.0039062z^7 + 0.0019531z^8 - 0.00097656z^9 \\ & + 0.00048828z^{10} - 0.00024414z^{11} + 0.00012207z^{12} - 0.000061035z^{13} \\ & + 0.000030518z^{14} - 0.000015259z^{15} + \dots \end{aligned}$$

and the first 15 exact filter coefficients are therefore (a box has been placed around the coefficient corresponding to the time $t = 0$)

$$\begin{aligned} f[n] = (& \boxed{0.5}, -0.25, +0.125, -0.0625, +0.03125, -0.015625, \\ & +0.0078125, -0.0039062, +0.0019531, -0.00097656, \\ & +0.00048828, -0.00024414, +0.00012207, -0.000061035, \\ & +0.000030518) \end{aligned}$$

which are to be compared with the least-squares filter coefficients obtained using Equation 8.55:

$$\begin{aligned} f[n] = (& \boxed{0.5}, -0.25, +0.125, -0.0625, +0.03125, -0.015625, \\ & +0.00781247, -0.00390619, +0.00195301, -0.000976324, \\ & +0.000487804, -0.000243187, +0.000120163, -0.0000572205, \\ & +0.0000228882) \end{aligned}$$

and the comparison is excellent.

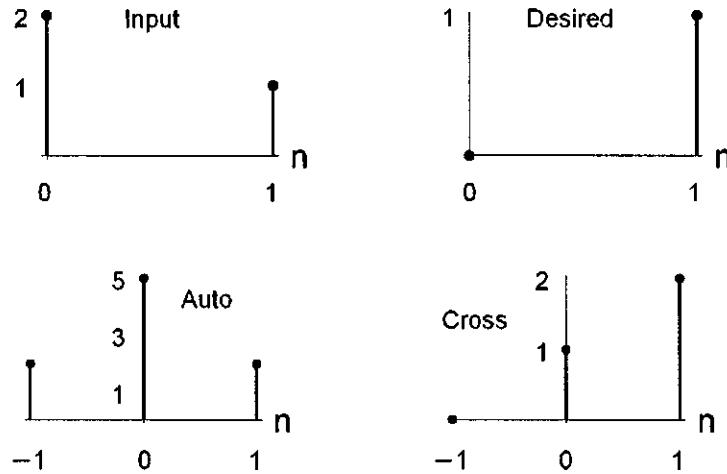


Figure 8.30: Autocorrelation of input and crosscorrelation of desired output with input for example (b) of Figure 8.28. See page 438 for *Mathematica* program.
(RT-Fig6-2.nb,cdr → RT-Fig6-2b.wmf)

Figure 6-2(b) from Robinson and Treitel

Refer to Figure 8.28b. The input is the same two-point wavelet. The desired output is a unit impulse time-shifted one time unit to the right. The autocorrelation of the input and the crosscorrelation of the desired output with the input are shown in Figure 8.30. Only filter coefficients corresponding to $t = 0$ and $t > 0$ are required because the input is still minimum-delay. The matrix equation is

$$\begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Solving for the filter coefficients, we get

$$f_0, f_1 = (0.047619, 0.380952)$$

and convolution with the input wavelet results in

$$\begin{aligned} f[n] * b[n] &= \sum_{m=0}^{m=1} f[m] b[n-m] \\ &= (0.0952381, 0.809524, 0.380952) \\ &\text{for } n = 0, 1, 2 \end{aligned}$$

with an error energy, E , given by

$$E = 1 - \sum_{\tau=0}^1 f_\tau \phi'_{db}(\tau) = 0.19$$

Increasing the length of the filter to, say, 15 points results in the matrix equation

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

now with an error energy, E , given by

$$E = 1 - \sum_{\tau=0}^1 f_\tau \phi'_{db}(\tau) = 2.7910^{-9}$$

and an actual output, $c[n]$, of

$$c[n] = f[n] * b[n] = \sum_{m=0}^{m=14} f[m] b[n-m]$$

$$= (\boxed{1.39698 \times 10^{-9}}, 1., 5.58794 \times 10^{-9}, -1.11759 \times 10^{-8}, 2.23517 \times 10^{-8}, \\ -4.47035 \times 10^{-8}, 8.9407 \times 10^{-8}, -1.78814 \times 10^{-7}, 3.57628 \times 10^{-7}, \\ -7.15256 \times 10^{-7}, 1.43051 \times 10^{-6}, -2.86102 \times 10^{-6}, 5.72205 \times 10^{-6}, \\ -0.0000114441, 0.0000228882, -0.0000457764) \\ \text{for } n = 0, 1, 2, \dots, 14$$

Figure 6-2(c) from Robinson and Treitel

The auto- and crosscorrelation coefficients for the example of Figure 8.28c are shown in Figure 8.31. See page 438 for *Mathematica* program. The input is the same minimum-delay wavelet; the desired output is two unit impulses at $n = 0$ and $n = 1$. The matrix equation is

$$\begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

The filter coefficients are

$$f[n] = (0.523809, 0.190476) \text{ for } n = 0, 1$$

and the convolution $c[n]$ with the filter coefficients is

$$c[n] = (1.04762, 0.90476, 0.190476) \text{ for } n = 0, 1, 2$$

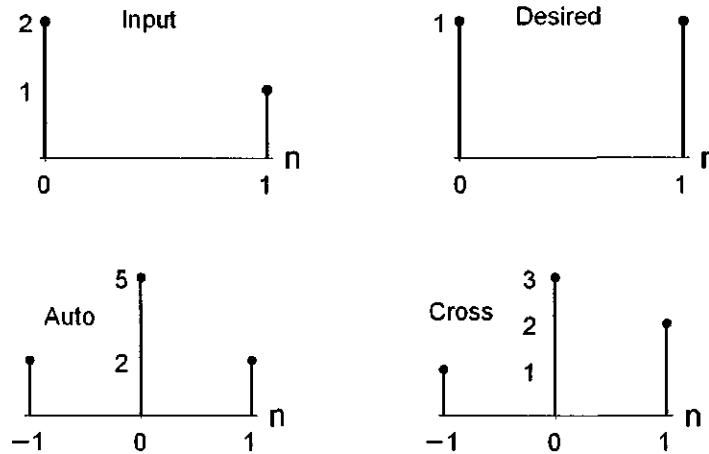


Figure 8.31: Autocorrelation of input and crosscorrelation of desired output with input for example (c) of Figure 8.28. See page 438 for *Mathematica* program.
(RT-Fig6-2.nb,cdr → RT-Fig6-2c.wmf)

with a normalized error energy, E , equal to

$$E = 0.0238$$

As expected, increasing the filter length to 15 points improves the results. The matrix equation becomes

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and the resulting convolution $c[n]$ is

$$c[n] = ([1.], 1., 2.79397 \times 10^{-9}, -5.58794 \times 10^{-9}, 1.11759 \times 10^{-8}, -2.23517 \times 10^{-8}, 4.47035 \times 10^{-8}, -8.9407 \times 10^{-8}, 1.78814 \times 10^{-7}, -3.57628 \times 10^{-7}, 7.15256 \times 10^{-7}, -1.43051 \times 10^{-6}, 2.86102 \times 10^{-6}, -5.72205 \times 10^{-6}, 0.0000114441, -0.0000228882)$$

now with an error energy of only

$$E = 3.49 \times 10^{-10}$$

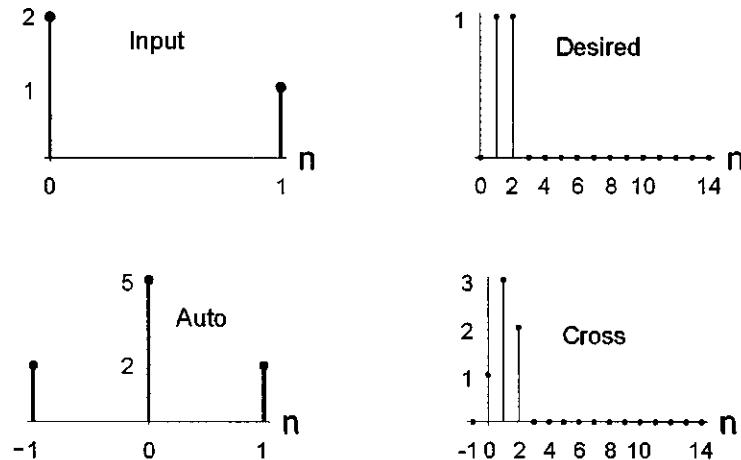


Figure 8.32: Autocorrelation of input and crosscorrelation of desired output with input for example (d) of Figure 8.28. See page 438 for *Mathematica* program. (RT-Fig6-2.nb,cdr → RT-Fig6-2d.wmf)

Figure 6-2(d) from Robinson and Treitel

The final example in Figure 8.28 is to reshape the same minimum-delay wavelet into a 2-point pulse time-shifted one unit to the right, as shown in Figure 8.32. See page 438 for *Mathematica* program. This time we go directly to a filter length of 15 points. Again, only filter coefficients that correspond to zero and positive times can be used, although the complete non - zero auto- and cross-correlation functions are shown in Figure 8.32. The matrix equation is

$$\left(\begin{array}{cccccccccccccc} 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 \end{array} \right) = \left(\begin{array}{c} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{array} \right)$$

See the CD-ROM that accompanies this volume for the *Mathematica* program Figure6-2a.RobinsonAndTreitel.nb for the least-squares filtering example of Figure 8.28a. This program contains segments for autocorrelation, crosscorrelation,

and convolution.

Maximum-delay examples

Figure 8.28a and d are shown below in Figure 8.33 with the input wavelets changed to maximum-delay. The desired outputs are the same as shown in Figure 8.28a and d, respectively. Complete non-zero auto- and crosscorrelation

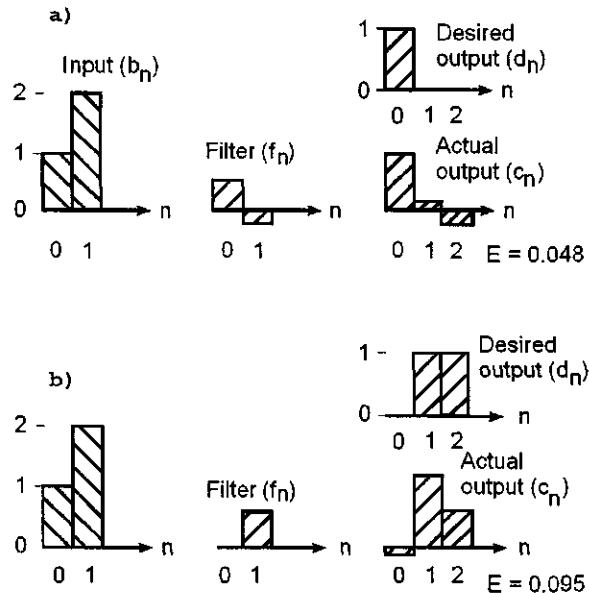


Figure 8.33: Left column: Input maximum-delay wavelet. Middle column: Computed filter coefficients. Right column: Desired and actual output. The quantity E is the minimum error energy given by Equation (8.47). (Figure6-2MaxFromRobinsonAndTreitel.cdr,wmf)

coefficients are shown in Figure 8.34. The input wavelet must be reshaped by using only crosscorrelation coefficients corresponding to negative time. For a 2-point filter, the matrix equation is

$$\begin{bmatrix} r_0 & r_1 \\ r_{-1} & r_0 \end{bmatrix} \begin{bmatrix} f_{-2} \\ f_{-1} \end{bmatrix} = \begin{bmatrix} g_{-2} \\ g_{-1} \end{bmatrix}$$

$$\begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} f_{-2} \\ f_{-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad (8.56)$$

Note that no crosscorrelation coefficients can correspond to zero or positive time lags. The filter coefficients are therefore defined only for the corresponding

negative times. Solving for the filter vector we get

$$f[n] = (-0.190476, 0.47619) \text{ for } n = -2, -1 \text{ respectively}$$

and convolution with the maximum-delay input gives

$$c[n] = (-0.190476, 0.0952381, 0.952381) \text{ for } n = -2, -1, 0$$

Increasing the filter length improves the result. For a 5-point filter

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 2 & 5 \end{pmatrix} \begin{pmatrix} f_{-5} \\ f_{-4} \\ f_{-3} \\ f_{-2} \\ f_{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

gives filter coefficients

$$f[n] = (0.0234432, -0.0586081, 0.123077, -0.249084, 0.499634)$$

for $n = -5, -4, -3, -2, -1$ respectively

and after convolution with the maximum-delay input the actual output is

$$c[n] = (0.0234432, -0.0117216, 0.00586081, -0.0029304, 0.0014652, \boxed{0.999267})$$

for $n = -5, -4, -3, -2, -1, 0$, respectively.

with an error energy of

$$E = 0.0007326$$

For a 15-point filter with only an anticipation component , Equation 8.55 becomes

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 \end{pmatrix} \begin{pmatrix} f_{-15} \\ f_{-14} \\ f_{-13} \\ f_{-12} \\ f_{-11} \\ f_{-10} \\ f_{-9} \\ f_{-8} \\ f_{-7} \\ f_{-6} \\ f_{-5} \\ f_{-4} \\ f_{-3} \\ f_{-2} \\ f_{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

The filter coefficients are

$$f[n] = (0.0000228882, -0.0000572205, 0.000120163, -0.000243187, 0.000487804, -0.000976324, 0.00195301, -0.00390619, 0.00781247, -0.015625, 0.03125, -0.0625, 0.125, -0.25, 0.5) \\ \text{for } n = -15, -14, \dots, -2, -1$$

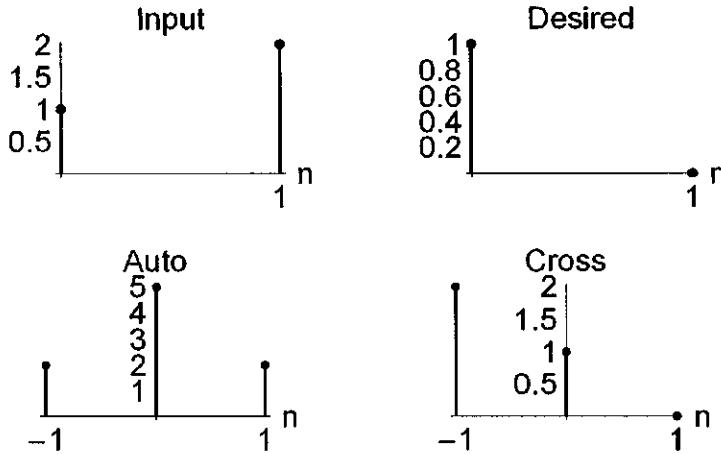


Figure 8.34: Autocorrelation of input and crosscorrelation of desired output with input for example (a) of Figure 8.33. (RT-Fig6-2aMax.nb → Figure6-2Max.cdr → RT-Fig6-2aMax.wmf)

and the resulting convolution is

$$c[n] = (0.0000228882, -0.0000114441, 5.72205 \times 10^{-6}, -2.86102 \times 10^{-6}, \\ 1.43051 \times 10^{-6}, -7.15256 \times 10^{-7}, 3.57628 \times 10^{-7}, -1.78814 \times 10^{-7}, \\ 8.9407 \times 10^{-8}, -4.47035 \times 10^{-8}, 2.23517 \times 10^{-8}, -1.11759 \times 10^{-8}, \\ 5.58794 \times 10^{-9}, -2.79397 \times 10^{-9}, 1.39698 \times 10^{-9}, \boxed{1.}, \\ \text{for } n = -15, -14, \dots, -2, -1, 0)$$

The exact filter coefficients required to convert the 2-point wavelet $(2, 1)$ to the wavelet $(1, 0)$ are (Equation 8.56) and are found by expanding the inverse of the z -transform of the wavelet $(2, 1)$. The inverse z -transform is

$$0.5 z^0 - 0.25 z^1 + 0.125 z^2 - 0.0625 z^3 + 0.03125 z^4 - 0.015625 z^5 + 0.0078125 z^6 - \\ 0.0039062 z^7 + 0.0019531 z^8 - 0.00097656 z^9 + 0.00048828 z^{10} - \\ 0.00024414 z^{11} + 0.00012207 z^{12} - 0.000061035 z^{13} + 0.000030518 z^{14} - \\ 0.000015259 z^{15} + \dots \quad (8.57)$$

The exact filter coefficients required to convert the 2-point wavelet $(1, 2)$ to the wavelet $(1, 0)$ are found by expanding the inverse of the z -transform of the wavelet $(1, 2)$. The z -transform of the inverse is

$$0.5 z^1 - 0.25 z^2 + 0.125 z^3 - 0.0625 z^4 + 0.03125 z^5 - 0.015625 z^6 + \\ 0.0078125 z^7 - 0.0039062 z^8 + 0.0019531 z^9 - 0.00097656 z^{10} + \\ 0.00048828 z^{11} - 0.00024414 z^{12} + 0.00012207 z^{13} - 0.000061035 z^{14} + \\ 0.000030518 z^{15} + \dots \quad (8.58)$$

Inspection of Equations (8.57) and (8.58) shows that no coefficients corresponding to $t = 0$ are found in Equation (8.58) because there is no term z^0 in Equation (8.58). Otherwise the numerical values of the inverse filter coefficients are

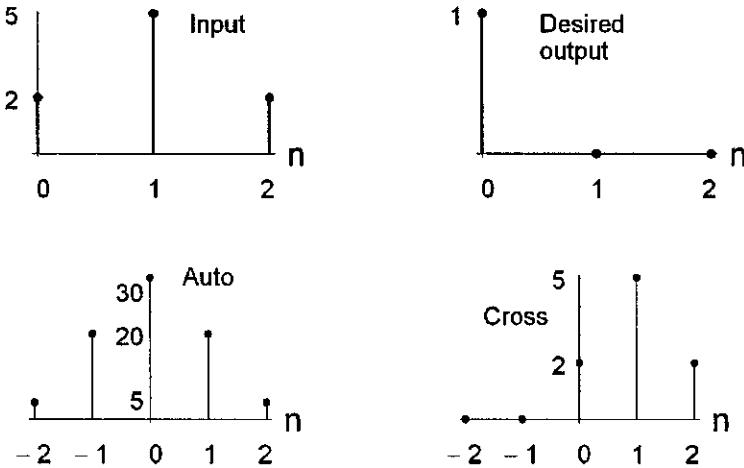


Figure 8.35: Autocorrelation of input and crosscorrelation of desired output with input. (RT-Fig6-2aMixed.nb → RT-Fig6-2Mixed.cdr → RT-Fig6-2Mixed.wmf)

identical. This is, of course, just confirmation that we have properly expanded each power series in order to achieve convergence of the filter coefficients.

Shaping mixed-delay wavelets

Convolution of minimum- and maximum-delay wavelets results in a mixed-delay wavelet. In spite of its simplicity, a simple example to illustrate all of the principles involved will be to convolve the two 2-point wavelets $(2,1)$ and $(1,2)$ and then design a least-squares filter to reshape the mixed-delay result. The convolution is

$$(2, 1) * (1, 2) = (2, 5, 2)$$

We want to design a least-squares filter that will convert this wavelet to an impulse at $t = 0$. The mixed-delay wavelet, desired output, and correlations are shown in Figure 8.35. The matrix equation is now

$$\begin{bmatrix} r_0 & r_1 & r_2 & r_3 & r_4 \\ r_{-1} & r_0 & r_1 & r_2 & r_3 \\ r_{-2} & r_{-1} & r_0 & r_1 & r_2 \\ r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 \\ r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 \end{bmatrix} \begin{bmatrix} f_{-2} \\ f_{-1} \\ f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} g_{-2} \\ g_{-1} \\ g_0 \\ g_1 \\ g_2 \end{bmatrix}$$

which, for our example, becomes

$$\begin{bmatrix} 33 & 20 & 4 & 0 & 0 \\ 20 & 33 & 20 & 4 & 0 \\ 4 & 20 & 33 & 20 & 4 \\ 0 & 4 & 20 & 33 & 20 \\ 0 & 0 & 4 & 20 & 33 \end{bmatrix} \begin{bmatrix} f_{-2} \\ f_{-1} \\ f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 5 \\ 2 \end{bmatrix} \quad (8.59)$$

Solving for the inverse filter coefficients, $f[n]$, using, for example, the Mathematica function *LinearSolve*,

```
filter=LinearSolve[r,g]/N
```

where r and g are the auto- and crosscorrelation coefficients, respectively, as shown in Equation (8.59). The result is

$$f[n] = (-0.0172989, 0.0540635, -0.127602, 0.27848, -0.0927028) \text{ for } n = -2, -1, 0, +1, +2$$

and convolution of the filter with the mixed-delay wavelet gives

$$c[n] = (-0.0345977, 0.0216327, -0.019484, 0.0270773, \boxed{0.951791}, 0.093446, -0.185406)$$

where the boxed value corresponds to $n = 0$.

If we arbitrarily take the same number of points, say 5, for the anticipation and memory components of the inverse filter, the matrix equation is

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 \\ r_{-1} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 \\ r_{-2} & r_{-1} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 \\ r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \\ r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 \\ r_{-5} & r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 & r_2 & r_3 & r_4 \\ r_{-6} & r_{-5} & r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 & r_2 & r_3 \\ r_{-7} & r_{-6} & r_{-5} & r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 & r_2 \\ r_{-8} & r_{-7} & r_{-6} & r_{-5} & r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 & r_1 \\ r_{-9} & r_{-8} & r_{-7} & r_{-6} & r_{-5} & r_{-4} & r_{-3} & r_{-2} & r_{-1} & r_0 \end{pmatrix} \begin{pmatrix} f_{-5} \\ f_{-4} \\ f_{-3} \\ f_{-2} \\ f_{-1} \\ f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} g_{-5} \\ g_{-4} \\ g_{-3} \\ g_{-2} \\ g_{-1} \\ g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} \quad (8.60)$$

Solving for the filter coefficients $f[n]$,

$$f[n] = 0.000730455, -0.00219163, 0.00493191, -0.0102309, 0.0206937, -0.041532, 0.0831592, -0.166395, 0.332876, -0.165888, 0.0820284, -0.0395493, 0.0175774, -0.00585913$$

and convolution with the mixed-delay wavelet yields

$$c[n] = (-0.000730321, 0.000365429, -0.000183251, 0.0000926982, -0.0000484949, 0.0000285389, -0.0000223524, 0.0000285921, -0.0000486279, 0.0000929777, \boxed{0.999816}, 0.000366563, -0.000732591, 0.00146492, -0.0029297, 0.00585933, -0.0117186)$$

with a box around the value of the filtered output corresponding to $t = 0$.

Figure 6-3 from Robinson and Treitel

Robinson and Treitel [153] provided another useful example of least-squares filtering to reshape a general mixed-delay wavelet (Figure 8.36 and Figure 8.37). The point of their example is to show that introducing a harmless delay into the desired output produces better results. We now explain this result in terms of crosscorelation coefficients defined for negative, zero, and positive lags. In Figure 8.36, note that only filter coefficients defined for zero and positive lags were used (crosscorrelation of the input and desired output was done for zero and positive lags only). The input and desired output were digitized from Figure 8.36 and are replotted in Figure 8.38a and b, respectively. The complete auto- and crosscorrelation functions for all positive and negative lags are shown in Figure 8.38c and d, respectively. We know from the Section on Shaping Mixed Delay Wavelets on page 441 that a mixed-delay input must be shifted to the left with respect to the desired output as well as to the right so that the general least-squares matrix equation becomes

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & \dots & r_{k-1} \\ r_1 & r_0 & r_1 & r_2 & \dots & r_{k-2} \\ r_2 & r_1 & r_0 & r_1 & \dots & r_{k-3} \\ r_3 & r_2 & r_1 & r_0 & \dots & r_{k-4} \\ \vdots & & & & & \\ r_{k-1} & r_{k-2} & r_{k-3} & r_{k-4} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_{m-1} \\ \vdots \\ f_{-1} \\ f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_{m-1} \\ \vdots \\ g_{-1} \\ g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix} \quad (8.61)$$

g_τ is now defined for negative, zero, and positive lags. This results in a two-sided filter f_t with an anticipation component as well as a memory component. The actual output from this two-sided filter is shown in Figure 8.38f. The filtered result is a good approximation to the desired output and is properly positioned in time. A better approximation to the desired output could be obtained by simply making the inverse filter defined by Equation (8.61) longer. Try this with the *Mathematica* program RT-Fig6-3TwoSidedFilter.nb on the CD-ROM that accompanies this volume.

The point of Figure 8.36, however, is to show that if the desired output is delayed in time, then the actual output will be a better approximation to the desired output *even if inverse filter coefficients are defined only for zero and positive time*. To see why this is so we delay the desired output and examine the resulting auto-and crosscorrelation coefficients, which are shown in Figure 8.38. Equation (8.61) now becomes

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_3 & \dots & r_{k-1} \\ r_1 & r_0 & r_1 & r_2 & \dots & r_{k-2} \\ r_2 & r_1 & r_0 & r_1 & \dots & r_{k-3} \\ r_3 & r_2 & r_1 & r_0 & \dots & r_{k-4} \\ \vdots & & & & & \\ r_{k-1} & r_{k-2} & r_{k-3} & r_{k-4} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_{n-1} \end{pmatrix} \quad (8.62)$$

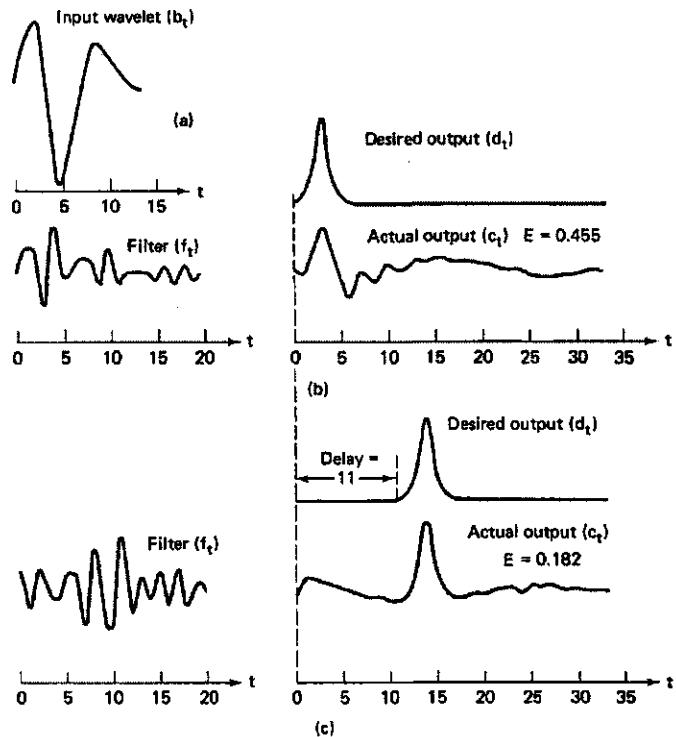


Figure 8.36: (a) Figure 6-3 reproduced from Robinson and Teritel [153]. Input mixed-delay wavelet. (b) Desired and actual output if the desired output starts at $t = 0$. (c) Desired output and actual output if the desired output is time-delayed. A better match is obtained because the introduction of a delay into the desired output is equivalent to the introduction of filter coefficients that are now defined for negative lags for the crosscorrelation coefficient function as well as for positive lags. If the inverse filter has been design by using crosscorrelation coefficients corresponding to both negative, zero, and positive lags then the same result is obtained without the arbitrary delay. See Figure 8.38. (RT-Fig6-2aMax.nb → Figure6-2MaxFromRobinsonAndTreitel.cdr → RT-Fig6-3.bmp)

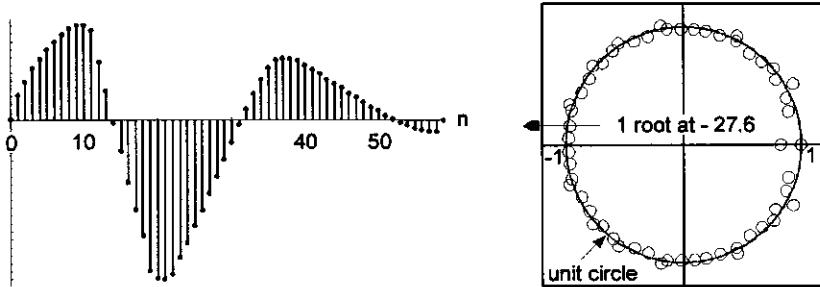


Figure 8.37: Left: Digitized mixed-delay input wavelet of Figure 8.36. Right: Roots of wavelet shown with plot of unit-circle. One root at -27.6 not shown on plot. (RootsOfRTFig63Wavelet.nb → RT-Fig6-3.cdr → RT-Fig6-3WaveletRoots.wmf)

where only zero and positive lags are now used to solve for the inverse filter. The results shown in Figure 8.36 demonstrate that a better approximation to the desired output is obtained if the desired output is delayed in time. The best way to visualize what is happening is to plot the crosscorrelation function obtained when the desired output is time-delayed. The result is shown in Figure 8.39b where the desired output has been delayed in time by an amount equal to the duration of the input. Positive lags are obtained by shifting the input to the right with respect to the desired output. Inspection of Figure 8.39 shows that non-zero values of $g(\tau)$ will be obtained for $\tau \geq 0$. On the other hand, $g(\tau) = 0$ for all negative values of τ if the input is shifted to the left with respect to the desired output. This is entirely due to the delay imposed on the desired output. Note, however, that the numerical values of the crosscorrelation function are identical and are independent of any time-shift of the desired output. The same numerical values of g are used in matrix Equation (8.61) and Figure 8.38 as are used in (8.62) and Figure 8.39. The only difference is a time shift in $g(t)$ and therefore a corresponding time shift in $f(t)$ because $f(0)$ is now at a different location in the solution vector of Equation (8.61). The reader can experiment with *Mathematica* program RT-Fig6-3OneSidedFilter.nb to see the degradation of the desired output as the input is delayed less. Change the statement `DesiredOutput=Join[Table[0,{Length[wavelet]}],DesiredOutput];` to, for example, `DesiredOutput=Join[Table[0,{11}],DesiredOutput];`.

Conclusions

- Least-squares filters are approximations to their corresponding exact theoretical counterparts. There is only one mathematical solution for an inverse filter—the exact filter of infinite length defined by Equation (8.57) or (8.58), as appropriate.
- The longer the least-squares inverse filter, the closer in value the filter

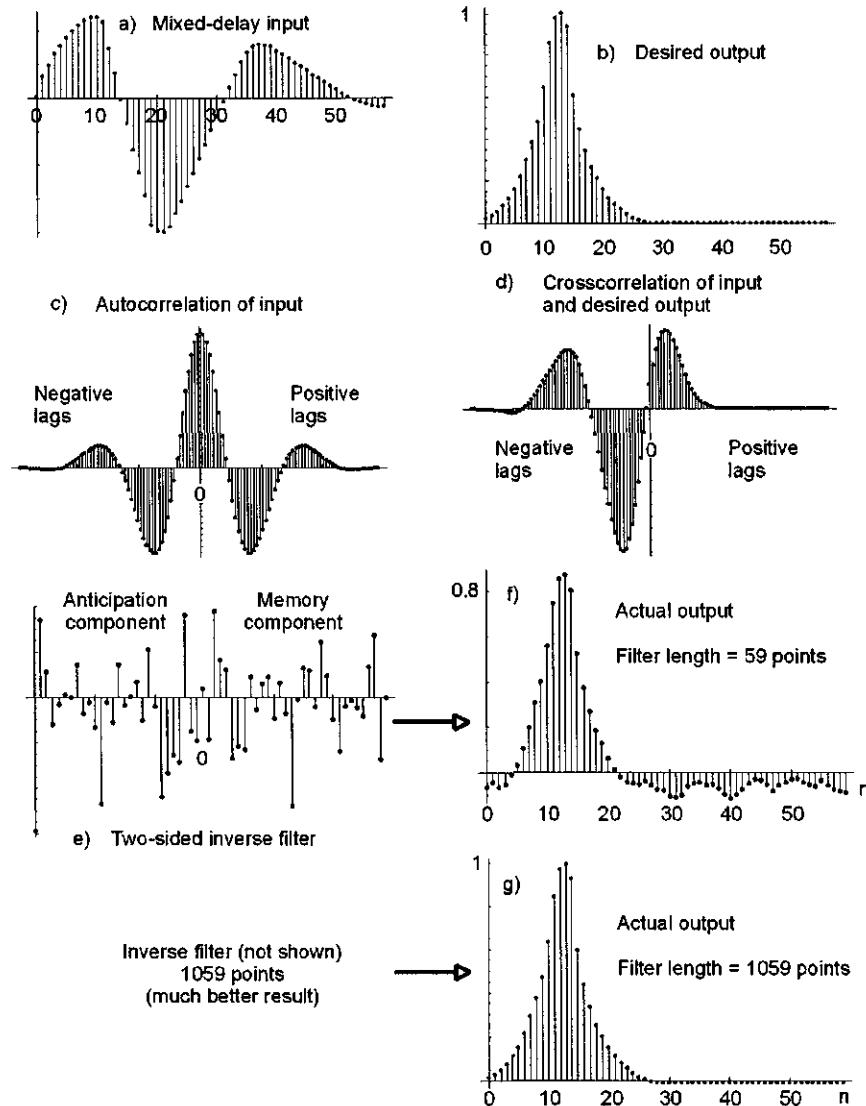


Figure 8.38: Same as Figure 8.36 but with autocorrelations and crosscorrelations added and assuming no shift of desired output. (a) Input mixed-delay wavelet. (b) Desired output starting at $t = 0$, as shown in Figure 8.36. (c) Autocorrelation of input. (d) Crosscorrelation of input and desired output. (e) Inverse filter of 59 points. (f) Actual output from this filter. A good result is obtained because the entire crosscorrelation function was used; i.e., the inverse filter has both a memory and an anticipation component. (g) Actual output is excellent after padding the input wavelet with 1000 zeros, which increases the length of the two-sided inverse filter f_t defined by Equation (8.61). See Figures 8.36 and 8.39 for results that use an inverse filter with only a memory component. (RT-Fig6-3TwoSidedFilter.nb → RT-Fig6-3.cdr → RT-Fig6-3TwoSidedFilter.wmf)

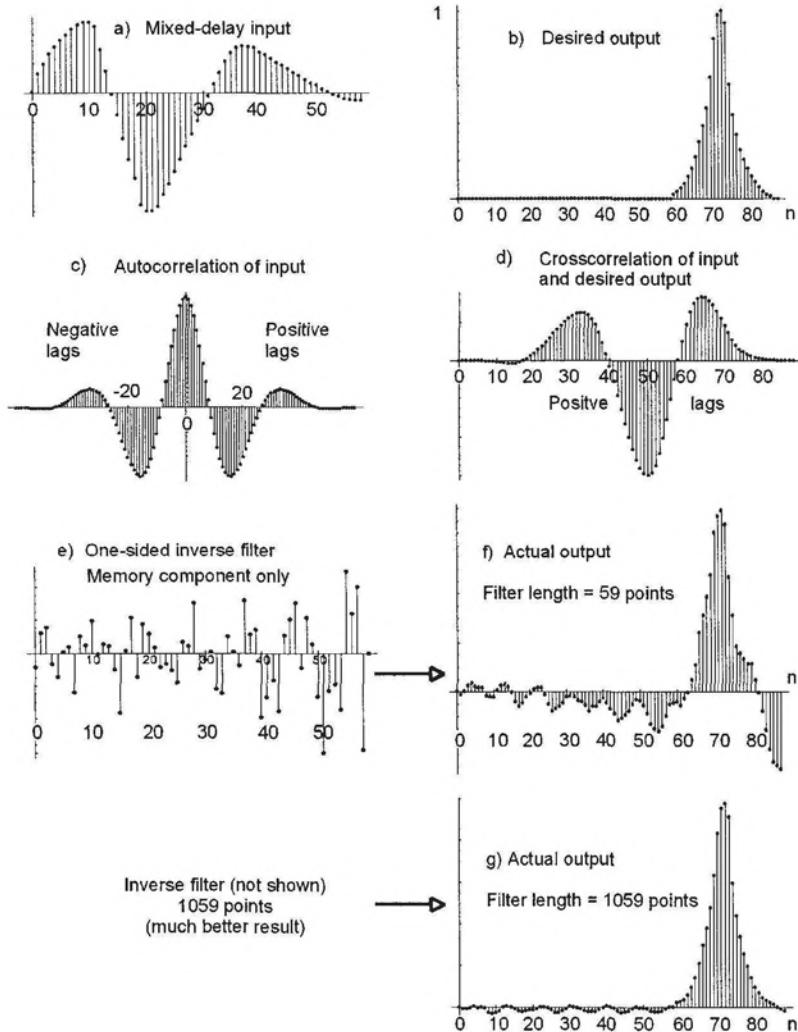


Figure 8.39: Same as Figure 8.36 but with autocorrelations and crosscorrelations added. (a) Input mixed-delay wavelet. (b) Desired output is now delayed, as shown in Figure 8.36. (c) Autocorrelation of input. (d) Crosscorrelation of input and desired output. Because of the amount of delay chosen for the desired output, crosscorrelation coefficients for $\tau < 0$ are zero and are not shown. (e) Inverse filter of 59 points. (f) Actual output from this filter. A good result is obtained because the entire crosscorrelation function was used; i.e., the inverse filter has only a memory component. (g) Actual output is excellent after padding the input wavelet with 1000 zeros, which increases the length of the memory component of the inverse filter f_t defined by (8.62). See Figure 8.38 for results that use an inverse filter with both a memory component and an anticipation component. (RT-Fig6-3OneSidedFilter.nb,wmf and RT-Fig6-3.cdr)

coefficients will approach their exact theoretical values.

- If the input wavelet is neither minimum- nor maximum-delay (it is therefore mixed-delay), then a filter applied to the input in order to convert its shape to an impulse at $t = 0$ or to any other shape must include crosscorrelation coefficients that correspond to negative, zero, and positive lags, as indicated in Equation (8.60) or (8.61). Theoretically at least, the longer the filter (the larger the number of simultaneous linear equations), the better the results.

What about Fourier theory?

Can the same result be obtained from Fourier theory without considering the delay properties of the input wavelet? Yes. To show this, assume the following Fourier transform pairs:

$$\begin{aligned} b(t) &\iff B(\nu) \\ f(t) &\iff F(\nu) \\ c(t) &\iff C(\nu) \end{aligned}$$

We require an inverse filter (for convenience we use notation for continuous functions) such that

$$c(t) = \int_{-\infty}^{+\infty} f(t - \tau) b(\tau) d\tau = \delta(t) \quad (8.63)$$

where $\delta(t) = 1$ for $t = 0$ and $\delta(t) = 0$ for $t \neq 0$. Convolution in the time domain is equivalent to multiplication in the frequency domain, so Equation (8.63) can be written

$$C(\nu) = F(\nu) B(\nu)$$

If the desired output is $\delta(t)$, then the filter $f(t)$ is the inverse Fourier transform of

$$F(\nu) = \frac{1}{B(\nu)}$$

That is, the inverse filter coefficients $f(t)$ are given by

$$f(t) = \int_{-\infty}^{+\infty} \frac{1}{B(\nu)} e^{-i2\pi\nu t} d\nu \quad (8.64)$$

Recall that there is only one correct solution for the numerical values of the inverse filter coefficients such as those obtained from Equation (8.56) for the minimum-delay wavelet (2,1). These values were

$$\begin{aligned} f[n] = (&0.5, -0.25, +0.125, -0.0625, +0.03125, -0.015625, \\ &+0.0078125, -0.0039062, +0.0019531, -0.00097656, \\ &+0.00048828, -0.00024414, +0.00012207, -0.000061035, \\ &+0.000030518) \end{aligned} \quad (8.56)$$

The result using Equation (8.64) is

$$f[n] = (\boxed{0.5}, -0.25, 0.125, -0.0625, 0.03125, -0.015625, \\ 0.0078125, -0.00390625, 0.00195312, -0.000976563, \\ 0.000488281, -0.000244141, 0.00012207, -0.000061035, \\ 0.0000305176)$$

The numerical values are identical to those shown in Equation (8.56).

A simple *Mathematica* program illustrates the procedure and the results. *Mathematica*'s version of a Fourier transform requires that the input and the inverse Fourier transform be scaled by the constant $1/\sqrt{n}$ where n is the number of points in the wavelet. The *Mathematica* program for the minimum-delay wavelet (2,1) is

```
wavelet=Join[{2,1},Table[0,{1500}]];
constant=1/Length[wavelet];
f=constant/Fourier[wavelet];
Take[Chop[InverseFourier[f]],15]
```

Note in the *Mathematica* program that the wavelet (2,1) has been padded ("Join") with a lot (1500) of zeros in order to open the sample window from two points to something approaching "infinity", the latter being the window width associated with the results from z-transform theory (Equation 8.56). The function "Take" prints the first 15 coefficients beginning with $f[0]$. "Chop" simply removes small complex values that result from numerical computation rounding error.

The *Mathematica* program for the maximum-delay wavelet (1,2) is

```
wavelet=Join[{1,2},Table[0,{1500}]];
constant=1/Length[wavelet];
f=constant/Fourier[wavelet];
Take[Chop[InverseFourier[f]],-15]
```

The structure of the Fourier transform requires us to know where the coefficients corresponding to negative time are located in the sequence $f[n]$. The last 15 coefficients in the vector f and are seen to be identical to those given in Equation (8.58). Running the above *Mathematica* program produces the coefficients

$$f[n] = (0.0000305176, -0.0000610352, 0.00012207, -0.000244141, \\ 0.000488281, -0.000976563, 0.00195312, -0.00390625, 0.0078125, \\ -0.015625, 0.03125, -0.0625, 0.125, -0.25, 0.5)$$

which correspond to values of n of
 $n = -15, -14, \dots, -3, , 2, , -1$

The *Mathematica* program for the mixed-delay wavelet (2,5,2) is

```
wavelet=Join[{2,5,2},Table[0,{1500}]];
constant=1/Length[wavelet];
f=constant/Fourier[wavelet];
AnticipationComponent=Take[Chop[InverseFourier[f]],-15];
MemoryComponent=Take[Chop[InverseFourier[f]],+15];
CompleteFilter=Join[AnticipationComponent,MemoryComponent]
```

The anticipation component of the filter is located at $f[m], f[m-1], f[m-2], \dots$ where m is the length of the wavelet padded with zeros. The memory component is located at positions $f[0], f[1], f[2], \dots$ where $f[0]$ is the first element in the vector $f[n]$. "Join" joins the two filter components into the complete filter. Running the above *Mathematica* program produces the filter coefficients

$$f[n] = (0.0000203451, -0.0000406901, 0.0000813802, -0.00016276, \\ 0.000325521, -0.000651042, 0.00130208, -0.00260417, \\ 0.00520833, -0.0104167, 0.0208333, -0.0416667, \\ 0.0833333, -0.166667, 0.333333, \\ \boxed{-0.166667}, \\ 0.0833333, -0.0416667, 0.0208333, \\ -0.0104167, 0.00520833, -0.00260417, \\ 0.00130208, -0.000651042, 0.000325521, -0.00016276, \\ 0.0000813802, -0.0000406901, 0.0000203451, -0.0000101725)$$

Convolution of the two-sided inverse filter $f[n]$ with the mixed-delay input wavelet (2,5,2) yields

$$c[n] = \begin{aligned} & (0.0000406901, 0.0000203451, 2.15756 \times 10^{-17}, -6.23416 \times 10^{-18}, \\ & -5.20417 \times 10^{-18}, 8.58688 \times 10^{-17}, 9.84456 \times 10^{-17}, 1.03216 \times 10^{-16}, \\ & 9.71445 \times 10^{-17}, 1.45717 \times 10^{-16}, 9.02056 \times 10^{-17}, 1.38778 \times 10^{-17}, 0., \\ & -5.55112 \times 10^{-17}, -4.44089 \times 10^{-16}, \\ & \boxed{1.}, \\ & -1.94289 \times 10^{-16}, -2.77556 \times 10^{-17}, -2.77556 \times 10^{-17}, 2.77556 \times 10^{-17}, \\ & -1.73472 \times 10^{-17}, -1.73472 \times 10^{-17}, 4.81386 \times 10^{-17}, 8.32667 \times 10^{-17}, \\ & 1.38453 \times 10^{-16}, 1.0777 \times 10^{-16}, 1.00587 \times 10^{-16}, 7.96618 \times 10^{-17}, \\ & 3.14283 \times 10^{-17}, 1.19723 \times 10^{-16}) \end{aligned}$$

with a box (\square) around the value corresponding to $n = 0$. A simple *Mathematica* program for general inverse filtering and shaping a wavelet with arbitrary delay characteristics is

```

wavelet=Join[{2,5,2},Table[0,{1500}]];
DesiredOutput=Join[{1,1},Table[0,{1501}]];
constant=1/Sqrt[Length[wavelet]];
f=constant*Fourier[DesiredOutput]/Fourier[wavelet];
AnticipationComponent=Take[Chop[InverseFourier[f]],-15];
MemoryComponent=Take[Chop[InverseFourier[f]],+15];
TwoSidedFilter=Join[AnticipationComponent,MemoryComponent];
c=DiscreteConvolution[SampledData[TwoSidedFilter,0,n],
  SampledData[wavelet,0,n],n];
Chop[Take[Part[c,1],{3,30}]]

```

Running the above program yields the convolution, $c[n]$, of the inverse filter, $f[n]$, with the mixed-delay input wavelet $(2, 5, 2)$ for $n = 0, 1, 2$.

```
for n = -13, -12, -11 ... - 2, -1, 0, +1 + 2, ... + 13, +14
```

The impulse actually occurs at $n = 0$ as requested; however, a harmless time shift is introduced by the length of the anticipation component of the filter, and so the output is shifted toward positive time as shown. Any mixed-delay wavelet substituted for $(2, 5, 2)$ will give the same desired output. Any desired output can be used. As in least-squares filtering, the delay characteristics (i.e., the shape) of the desired output are not important. In fact, the above *Mathematica* program is completely general because the Fourier transform always implicitly takes into account inverse filter coefficients that correspond to negative, zero, and positive time. If the input wavelet is minimum-delay, then the filter coefficients for $n < 0$ will automatically be zero. If the wavelet is maximum-delay, then the filter coefficients for $n \geq 0$ will be zero. If the wavelet is mixed-delay, then the requisite filter coefficients for $n < 0$ and $n \geq 0$ will be automatically determined. The user must be aware that the lengths of the "padded wavelet" and the "padded desired output" must always be equal; this can be accomplished by simply padding with zeros as shown in the program.

Inverse filtering of a vibroseis wavelet

We use a slight modification of the preceding "Fourier-based" program to convert a vibroseis wavelet to an impulse. The equation for a chirp signal is

$$b(t) = \sin \left[2\pi \left(\nu_1 + \frac{\nu_2 - \nu_1}{2L} t \right) t \right]$$

where ν_1 and ν_2 are the starting and ending frequencies, respectively, in Hertz, and L is the sweep length in seconds. The chirp is plotted in Figure 8.40a. Next the Fourier transform of the chirp is multiplied by the complex conjugate of its transform and the inverse Fourier transform of the product is the Klauder [76] wavelet shown in Figure 8.40b. For more on the history of the Klauder wavelet, see Chapter 7. This is the wavelet we wish to convert to an impulse. For illustrative purposes in this section, we design the inverse filter in the frequency domain and then convolve its inverse Fourier transform $f(t)$ with the desired output $d(t)$. That is,

$$f(t) = \int_{-\infty}^{+\infty} \frac{1}{F(\nu) F^*(\nu)} d\nu. \quad (8.65)$$

If the desired output, $d(t)$, is not an impulse at $n = 0$, and because $d(t) \iff D(\nu)$, then Equation (8.65) simply becomes

$$f(t) = \int_{-\infty}^{+\infty} \frac{D(\nu)}{F(\nu) F^*(\nu)} d\nu. \quad (8.66)$$

Because of the delay properties of the Klauder [76] wavelet, a two-sided filter is required. In addition, the analysis window must be large in order to approach the exact filter coefficients. For this reason we pad the Klauder wavelet with

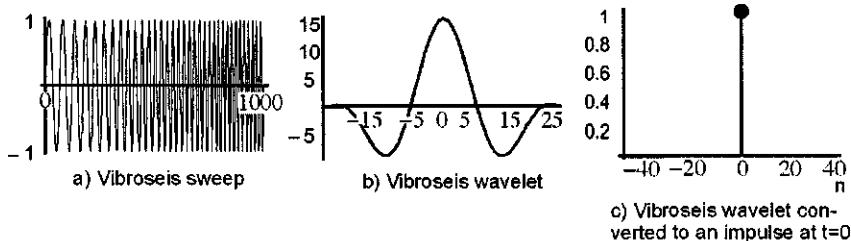


Figure 8.40: From left to right: (a) Vibrator chirp signal, b) Klauder wavelet with $\nu_1 = 14$ and $\nu_2 = 56$ Hertz; duration of sweep is 1 sec. This is the “vibroseis wavelet”, the inverse Fourier transform of the chirp spectrum multiplied by its complex conjugate. (c) Shaping of Klauder wavelet to an impulse. (FourierInverseFilteringVibroseis.nb,cdr → FourierInverseFilteringVibroseis.wmf)

zeros as shown in the *Mathematica* program. The more zeros that are used, the closer the filter coefficients approach their theoretically exact values. In the example of Figure 8.40, the wavelet was arbitrarily padded with 20000 zeros. The reader is encouraged to run the program with different analysis window widths to see the effect on the actual output. As usual, a perfect analysis window is one of infinite width.

The following example of shaping a Klauder wavelet uses a sweep (chirp) length of only one sec (period=1) in order to keep the program execution time short. In practice, our Virginia Tech seismic crew used a 2 ms sampling interval ($dt=0.002$) and a sweep length of 24 sec ($L=24$). For deep crustal research, the starting and ending sweep frequencies were the same, with a 1-sec taper on each end. This two-octave, tapered, 24-sec sweep generally produced excellent results using a single vibrator. See the CD-ROM that accompanies this volume for the *Mathematica* program that generated Figure 8.40. For comparison with Figure 8.28 as well as an example of a more general shaping filter, we choose as a desired output $d(t)$, a Klauder wavelet with a higher resolving power. This requires a few changes to the *Mathematica* program. The result is shown in Figure 8.41.

```
DesiredOutput=Join[{1},Table[0,{lw-1}]]; (* Change to next line *)
DesiredOutput=Join[NewKlauder,Table[0,{lw-Length[NewKlauder]}]];
```

See the CD-ROM that accompanies this volume for the *Mathematica* program that generated Figure 8.41. The reader should examine the effects of zeroing out the anticipation or memory component of the inverse filter before the final convolution. For the above program, this can be done by adding the statement

```
AnticipationComponent=Table[0,{i,1,Length[AnticipationComponent]}];
after the statement
AnticipationComponent=Take[Chop[InverseFourier[f]],-Length[wavelet]];
```

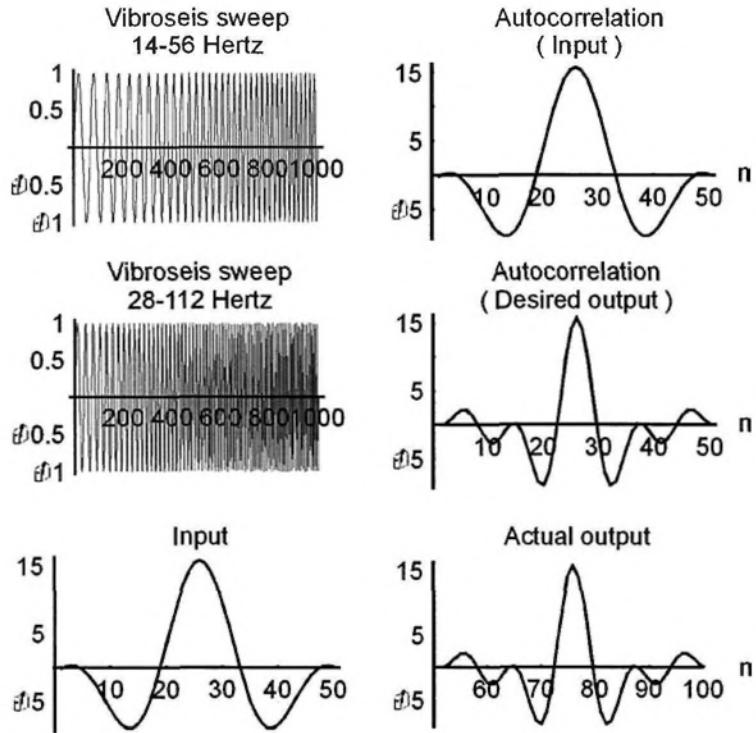


Figure 8.41: (a) Top left: vibrator chirp signal with $\nu_1 = 14$ and $\nu_2 = 56$ Hertz; duration of sweep is 1 sec. Right: Autocorrelation of chirp. (b) Left: vibroseis chirp signal whose autocorrelation is to the right. We wish to design a shaping filter that will convert (a)-right to (b)-right. (c) Left: Input. Right: Higher-resolution desired output. (FourierInverseFilteringVibroseis2.nb → FourierInverseFilteringVibroseis.cdr → FourierInverseFilteringVibroseis2.wmf)

Summary

The preceding discussion and the *Mathematica* programs listed above were designed to show how the same numerical values of spiking or shaping filter coefficients can be determined in either the time domain or the frequency domain. Design of the filter in the time domain is simpler because neither the exact nor least-squares method requires any consideration about the widths of Fourier analysis windows. On the other hand, the design of inverse filters in the frequency domain requires no insight into the “stability” of digital filters, that is, whether a memory component, an anticipation component, or both are required, and how these need to be computed. The proper design of filters in the time domain always requires an understanding of the delay-properties of the

input signal; however, sometimes this is simply assumed. It is clear that either approach, in ideal situations at least, could be used to reshape an input with arbitrary delay properties to any output with arbitrary delay properties.

One of the most common steps in the routine processing of reflection seismic data is called “predictive deconvolution”. The objective of predictive deconvolution is to reshape a seismic source wavelet. This disturbance, however, is commonly more than just a simple minimum- or mixed-delay 30-40 ms wavelet as it leaves the sourcepoint. Rather, it might become much longer because of accompanying reverberations caused by, for example, water-layer wave guides and interbed multiples. Commonly, the individual reverberating wavelets arrive at the receiver so closely in time that they overlap to form an extended source wavelet. It is this longer “effective source wavelet” that must be shortened by a type of inverse filtering known as predictive deconvolution, which is just another name for inverse filtering. Without the insight gained from this chapter, however, it would be difficult to understand the proper design of the predictive deconvolution filter. For example, the reverberations that result in an increase in the time duration of the source wavelet are always minimum-delay, so the reverberations themselves can be removed by a filter with only a memory component. This will shorten the length of the “effective source wavelet”, thereby increasing seismic resolution. If the source wavelet is mixed-delay, however, then the predictive deconvolution filter cannot be designed to shorten the source wavelet too much if the filter has only a memory component. It is hoped that the importance placed on the roles of the memory and anticipation components of an inverse filter has been clarified by the examples that use Fourier theory to arrive at the same results.

8.3.2 Predictive deconvolution

Now we come to predictive deconvolution, which is a commonly used procedure to attenuate reverberations and to shorten the duration of the primary source wavelet. The method was developed by Enders Robinson [145, 146, 149] and his colleagues based on work by Norbert Wiener[196] and was originally called “predictive decomposition” [150]. The objective is to

1. Shorten the time duration of the primary seismic source wavelet, or
2. Shorten the duration of the reverberation “effective source wavelet” as defined on page 406; i.e., remove water-layer wave guide reverberations, multiples, ghosts, etc.

These objectives are similar to those for other kinds of “inverse filtering”—that is, to increase seismic resolution by reshaping the source wavelet to something different, usually something shorter. Only the approach is different. It will be shown that for predictive deconvolution the increase in seismic resolution is obtained by “truncating” wavelets.

Recall that any inverse filtering procedure used to change the shape of a non-reverberating primary source wavelet is conceptually, *but not computationally*, a two-step process:

1. Convert the input wavelet to an impulse at the first point of the wavelet. This is the step where the decision about using negative- and/or positive-time inverse filter coefficients must be made in order to obtain convergence of the filter coefficients.
2. Convolve the impulse with the desired output. This step does not involve any concern about filter convergence no matter what the delay properties of the desired output might be. It is always Step 1 that requires the inverse filter of infinite length; therefore, the longer the least-squares filter, the closer the actual output will approach the desired output.

There are two basic approaches to wavelet shaping: (1) deterministic, and (2) statistical. So far, all of our discussions of inverse filtering have assumed that the wavelet shape was known; we used a deterministic approach to accomplish deconvolution, or inverse filtering. We investigated how a filter could be designed that would reshape an individual wavelet of known delay characteristics to a wavelet of *any* other shape. For reflection seismic data, however, the wavelets overlap too much to reveal the shape of a single reflected wavelet. We must design a filter to operate on a wavelet whose shape we don't know. The statistical approach involves random elements. These are the reflection coefficients, and they are assumed to be an uncorrelated white sequence. Predictive deconvolution is a statistical, rather than a deterministic, approach to inverse filtering, because the design of the inverse filter depends upon the statistical properties of the reflection coefficients.

The general idea behind predictive deconvolution is simple. We wish to design a filter that will predict some of the amplitude values of the primary source wavelet itself, as illustrated in (Figure 8.42). If we can do this, then we can subtract those values from the wavelet. For example, the desired filter will predict wavelet amplitudes starting at some arbitrary time α (referred to as the "prediction distance", "lag", or "gap") after the first point of the wavelet, and then subtract these predicted values from the wavelet. The result will shorten the duration of the wavelet and therefore increase resolution. The time α shown in Figure 8.42 is the prediction time to the first wavelet value to be predicted as measured from the first value. The value at time α and at later values will then be subsequently removed by subtraction. Predictive deconvolution therefore truncates the wavelet. The wavelet shape is assumed to be minimum-delay, or some wavelet-shaping filter is applied before deconvolution to make it minimum-delay. The reason for the minimum-delay assumption is a direct consequence of requiring prediction filter coefficients that are defined only for zero and positive time. As a truncation process, it is clear from Figure 8.42 that *after* predictive deconvolution, whatever remains of the truncated wavelet may be minimum-, maximum-, or mixed-delay.

For tutorial reasons, we divide the discussion of predictive deconvolution into two parts:

1. Deconvolution of a single known wavelet, and

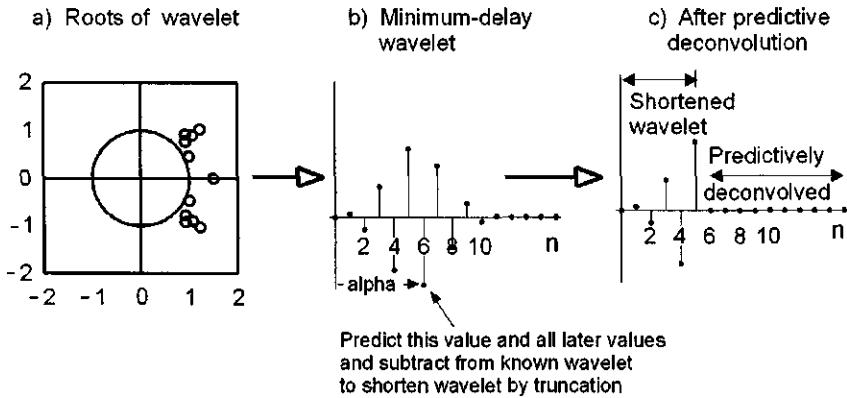


Figure 8.42: General diagram illustrating predictive deconvolution of a minimum-delay seismic wavelet. After deconvolution, the wavelet might be minimum-, maximum-, or mixed-delay depending on the choice of α . Same result is obtained when deconvolving a trace instead of a wavelet. (WaveletMakerMinDelay.nb → PredictiveDeconvolutionWavelets.cdr → PDgeneral.wmf)

2. Deconvolution of a seismic trace consisting of many overlapping wavelets whose shapes are unknown. The wavelet is assumed to be statistically time-invariant. It does not change shape along the path of propagation.

With one exception, the background and theoretical development of predictive deconvolution here follows the pioneering work of Robinson [145, 146, 150, 148, 151] (the latter is a volume in this Handbook of Geophysical Exploration Series), Robinson and Treitel [153], and Peacock and Treitel [134]. The exception is that in this volume we start with examples of the predictive deconvolution of a single, known wavelet, rather than with the usual convolutional model (a wavelet convolved with reflection coefficients) of the seismic trace. For either a known wavelet or one making up a seismic trace, the design of the deconvolution filter is entirely dependent upon the determination of autocorrelation coefficients, and we now explore this assumption in some detail.

If the reflectivity function (the reflection coefficients) is random then the shape of the autocorrelation of the seismic trace is directly proportional to the shape of the autocorrelation of the wavelet (Robinson and Treitel [153, Appendix 6-1, pages 162-163]). To the extent that this is a valid assumption, then starting a discussion of predictive deconvolution with the wavelet instead of with the seismic trace calls for some mathematical justification, which we now provide. Assume that the seismic trace is a stationary time series u_t given by the convolution

$$u_t = \sum_{s=0}^{\infty} b_s \epsilon_{t-s} \quad (8.67)$$

where b_t is the seismic wavelet and ϵ_t is white noise (it is assumed to be the sequence of reflection coefficients) of power P and zero mean. That is,

$$\sum \epsilon_t = 0 \quad \text{and} \quad \sum \epsilon_s \epsilon_t = \begin{cases} P & \text{if } s = t \\ 0 & \text{if } s \neq t \end{cases} \quad (8.68)$$

This convolutional model can be simulated by generating a sequence of random numbers, ϵ_t (the reflection coefficients), and convolving the sequence with a wavelet, b_t .

The autocorrelation $\phi(\tau)$ of the stationary time series u_t is

$$\phi(\tau) = \sum_{t=0}^{\infty} u_t u_{t+\tau}$$

Our objective now is to show that the autocorrelation of the wavelet b_t and the autocorrelation of the seismic trace u_t are directly proportional to one another.

Letting the reflection coefficients $\epsilon_t = (\epsilon_0, \epsilon_1, \epsilon_2)$ and the wavelet $b_t = (b_0, b_1, b_2)$, the autocorrelation of the stationary time series of Equation (8.67) yields the list of *Mathematica* "elements" (the autocorrelation coefficients $\phi(\tau)$ of the trace, separated in the list below by commas):

$$\begin{aligned} \phi(\tau) = & \{2 b_1 b_2 \epsilon_1 (\epsilon_0 + \epsilon_2) + \\ & b_0^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_1^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_2^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + \\ & 2 b_0 (b_2 \epsilon_0 \epsilon_2 + b_1 \epsilon_1 (\epsilon_0 + \epsilon_2)), \\ & b_0^2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_1^2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_2^2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_1 b_2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_0 \epsilon_2 + \epsilon_2^2) + \\ & b_0 (b_2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_1 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_0 \epsilon_2 + \epsilon_2^2)), \\ & b_0^2 \epsilon_0 \epsilon_2 + b_1^2 \epsilon_0 \epsilon_2 + b_2^2 \epsilon_0 \epsilon_2 + b_1 b_2 \epsilon_1 (\epsilon_0 + \epsilon_2) + \\ & b_0 (b_1 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2)), \\ & b_1 b_2 \epsilon_0 \epsilon_2 + b_0 (b_1 \epsilon_0 \epsilon_2 + b_2 \epsilon_1 (\epsilon_0 + \epsilon_2)), \\ & b_0 b_2 \epsilon_0 \epsilon_2\} \end{aligned}$$

where, using Equation (8.68) and choosing successive elements from the list,

$$\begin{aligned} \phi(0) &= 2 b_1 b_2 \epsilon_1 (\epsilon_0 + \epsilon_2) + \\ & b_0^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_1^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_2^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + \\ & 2 b_0 (b_2 \epsilon_0 \epsilon_2 + b_1 \epsilon_1 (\epsilon_0 + \epsilon_2)) \\ &= b_0^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_1^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_2^2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) \\ &= b_0^2 \sum_{t=0}^{t=2} \epsilon_t^2 + b_1^2 \sum_{t=0}^{t=2} \epsilon_t^2 + b_2^2 \sum_{t=0}^{t=2} \epsilon_t^2 \\ \phi(0) &= P(b_0^2 + b_1^2 + b_2^2) \\ \phi(1) &= b_0^2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_1^2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_2^2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_1 b_2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_0 \epsilon_2 + \epsilon_2^2) + \\ & b_0 (b_2 \epsilon_1 (\epsilon_0 + \epsilon_2) + b_1 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_0 \epsilon_2 + \epsilon_2^2)) \\ &= b_0 b_1 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) + b_1 b_2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) \\ &= b_0 b_1 \sum_{t=0}^{t=2} \epsilon_t^2 + b_1 b_2 \sum_{t=0}^{t=2} \epsilon_t^2 \end{aligned}$$

$$\begin{aligned}
\phi(1) &= P(b_0 b_1 + b_1 b_2) \\
\phi(2) &= b_0 b_2 \epsilon_0^2 + b_0 b_1 \epsilon_0 \epsilon_1 + b_1 b_2 \epsilon_0 \epsilon_1 + b_0 b_2 \epsilon_1^2 + b_0^2 \epsilon_0 \epsilon_2 + b_1^2 \epsilon_0 \epsilon_2 + b_2^2 \epsilon_0 \epsilon_2 + \\
&\quad b_0 b_1 \epsilon_1 \epsilon_2 + b_1 b_2 \epsilon_1 \epsilon_2 + b_0 b_2 \epsilon_2^2 \\
&= b_0 b_2 (\epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2) \\
&= b_0 b_2 \sum_{t=0}^{t=2} \epsilon_t^2 \\
\phi(2) &= P(b_0 b_2) \\
\phi(3) &= b_0 b_2 \epsilon_0 \epsilon_1 + b_0 b_1 \epsilon_0 \epsilon_2 + b_1 b_2 \epsilon_0 \epsilon_2 + b_0 b_2 \epsilon_1 \epsilon_2 \\
&= 0
\end{aligned}$$

The above example using the two short time series $b_t = (b_0, b_1, b_2)$ and $\epsilon_t = (\epsilon_0, \epsilon_1, \epsilon_2)$ is enough to generalize $\phi(\tau)$ to [153]

$$\begin{aligned}
\phi(\tau) &= b_0 b_\tau \sum_t \epsilon_t^2 + b_1 b_{\tau+1} \sum_t \epsilon_{t-1}^2 + \dots \\
&= P(b_0 b_\tau + b_1 b_{\tau+1} + \dots) \\
\phi(\tau) &= P \sum_{t=0}^{\infty} b_t b_{t+\tau}
\end{aligned} \tag{8.69}$$

The conclusion is therefore

$$\text{Autocorrelation of trace} = P \times \text{Autocorrelation of wavelet}$$

where the quantity P is equal to a simple constant, the “energy” of the sequence of reflection coefficients. The autocorrelations of the trace and the wavelet are therefore directly proportional to one another, if the time series is stationary. A *Mathematica* computer program to keep track of the algebra shown in the above derivation is listed on page 461. The above discussion is summarized graphically in Figure 8.43.

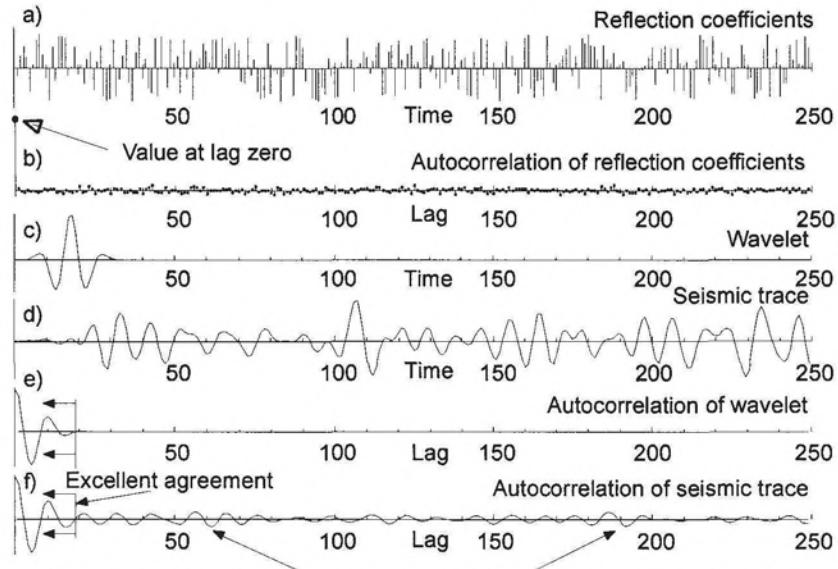
Noteworthy is the excellent confirmation of the assumption

$$\sum \epsilon_t = 0 \quad \text{and} \quad \sum \epsilon_s \epsilon_t = \begin{cases} P & \text{if } s = t \\ 0 & \text{if } s \neq t \end{cases}$$

as shown in Figure 8.43b. Comparison of autocorrelations in Figures 8.43e and f, confirms that

$$\phi(\tau) = P \sum_{t=0}^{\infty} b_t b_{t+\tau}$$

and that agreement is excellent up to the last lag possible for the autocorrelation of the finite-length wavelet (Figures 8.43e and 8.44). After this lag time the autocorrelation of the trace (Figure 8.43f) is seen to contain ripples with the same dominant frequency content as the wavelet. The ripples are a consequence of multiplication in the time-domain, which is equivalent to convolution in the frequency domain. That is, selection of only a portion of the trace is the same as



Convolution of wavelet spectrum with reflectivity spectrum results in these peaks and troughs because reflectivity function not perfectly random.
 These peaks and troughs have dominant frequency content of wavelet.

Figure 8.43: Figure summarizes mathematical discussion leading to Equation (8.69).
 (AutocorrelationOfASeismicTrace.nb → RandT1980Appendix6-1.cdr → RandT1980Appendix61.wmf)

multiplying the trace by a rectangular window. The trace and the window each have a Fourier spectrum. Therefore, the wavelet spectrum is superimposed on (convolved with) the spectrum of the reflectivity function. The spectrum of the reflectivity function, however, is not perfectly flat because it is not a perfectly random sequence. Some Fourier components “stick up” more than others and this results in a superposition of the dominant wavelet spectrum on the reflectivity spectrum; the convolution of the wavelet spectrum with the spectrum of the reflectivity function results in the observed ripples.

The relationship between the autocorrelation of the wavelet and that of the seismic trace having been examined, we continue our two-part discussion of predictive deconvolution:

1. Deconvolution of a single known wavelet,
2. Deconvolution of a seismic trace consisting of a time-invariant wavelet of unknown shape reflected from each subsurface reflector. The recorded reflections overlap and therefore conceal the shape of the individual wavelet.

In order to emphasize the “truncation” nature of predictive deconvolution, we start with a single wavelet. For a single, isolated, wavelet there are no random

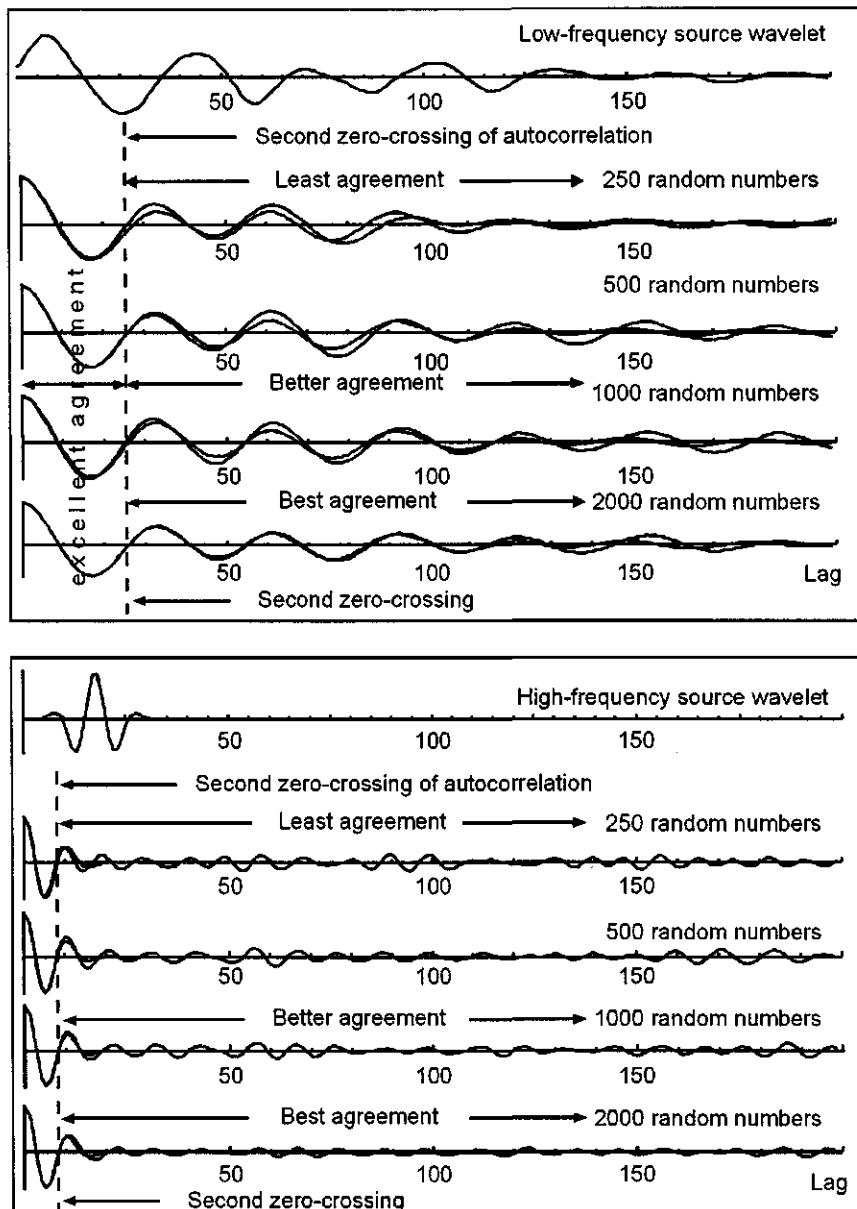


Figure 8.44: Top box: "Low-frequency" wavelet. Same reflectivity as Figure 8.43. Enlargement of portion of Figure 8.43e and f, but trace and wavelet acf's superimposed. By Equation (8.68), normalized acf's should overlay. Trace lengths used to obtain acf shown by number of reflection coefficients. Note better superposition of acf's with Equation (8.68) as trace length increases. Excellent for lags up to 2nd zero-crossing. Bottom box: "High-frequency" wavelet. Same notation. (AutocorrelationOfASeismicTrace.nb → RandT1980Appendix6-1.cdr → AutoDetail.wmf)

elements (reflection coefficients) involved. Our following examples of wavelet deconvolution are therefore “deterministic”, even though we will use the standard equations of predictive deconvolution. The exact autocorrelation is computed from a known wavelet instead of from the seismic trace itself, but the equations for predictive deconvolution are used in the normal way. This defines predictive deconvolution as a deterministic process rather than as a statistical one. Later, we revisit predictive deconvolution from the more conventional statistical viewpoint.

Throughout the discussions of deconvolution, we offer numerical examples to illustrate the required connection between inverse filters derived from predictive deconvolution, from Fourier theory, and from z -transforms because, for any given model, there can be only one numerical solution. One way to illustrate this is to determine inverse filter coefficients using different approaches to filter design and then show that the coefficients have the same numerical values. Hopefully, this approach will complement a purely theoretical development and provide some additional insight into filter design.

See also the CD-ROM that accompanies this volume for the *Mathematica* program Appendix6-1.nb used to implement Appendix 6-1, Robinson and Treitel, 1980.

```
(* Program Appendix6-1.nb *)
(* Program to implement Appendix 6-1, Robinson and Treitel, 1980 *)
(* Autocorrelation of wavelet proportional to autocorrelation
   of seismic trace *)
fold[a_,b_]:=Do[Do[k=i+j-1; c[[k]]=c[[k]]+a[[i]]*b[[j]],{j,1,Length[b]}, {i,1,Length[a]}];
cross[a_,b_]:=Do[c=Take[a,{k,Length[a]}];sum=0;
Do[sum=sum+c[[j]]*b[[j]],{j,1,Min[Length[c],Length[b]]}]; g[[k]]=sum,{k,1,Min[Length[a],Length[b]]}];

b={b0,b1,b2}; e={e0,e1,e2}; a=b;la=Length[a];b=e;lb=Length[b];
lc=la+lb-1; c=Table[0,{lc}]; fold[a,b];ut=c

Output={b0 e0,b1 e0+b_0 e1,b2 e0+b1 e1+b_0 e2,b2 e1+b1 e2,b2 e2}

ut={b0*e0,b1*e0+b0*e1,b2*e0+b1*e1+b0*e2,b2*e1+b1*e2,b2*e2};
a=ut;la=Length[a];b=ut;lb=Length[b];
lg=Min[la,lb];g=Table[0,{lg}]; cross[a,b]; Simplify[Expand[g]]
(* Run program to get output, which is phi(tau) *)
```

Design of the predictive deconvolution filter

Any linear digital filtering process is defined by the convolution

$$y_t = \sum_{\tau} x_{\tau} a_{t-\tau} \quad (8.70)$$

where x_t is the input, a_t is some kind of a filter, and y_t is the output, the convolution of x_t with a_t . (The sampling interval is $\Delta t = 1$ unit of time.)

First we will design a digital least-squares filter that, when convolved with the input, will return the entire input itself. We already know that such a filter in continuous-time is simply the Dirac delta function $\delta(t)$ so that (using continuous notation for convenience)

$$y(t) = \int_{-\infty}^{\infty} \delta(\tau) x(t - \tau) d\tau = x(t) \quad (8.71)$$

For discrete data, in order to obtain a filter that does the same thing as (8.71) we solve the following system of linear equations for the filter f_n :

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix}$$

where r_n and g_n are, as usual, the autocorrelation of the input and the crosscorrelation of the input and the desired output, respectively. But now the desired output is the input itself. So the above becomes

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{n-1} \end{pmatrix} \quad (8.72)$$

because the crosscorrelation of the input and the desired output is exactly equal to the autocorrelation of the input. Therefore, $g_n \equiv r_n$. As a numerical example we fabricate a minimum-delay wavelet by convolving together

$$(2, 1) * (2, 1) * (-4, 3) * (5, 4)$$

to obtain the minimum-delay wavelet

$$-80, -84, 24, 47, 12$$

Using (8.72) we obtain

$$\begin{pmatrix} 16385 & 6396 & -5580 & -4768 & -960 \\ 6396 & 16385 & 6396 & -5580 & -4768 \\ -5580 & 6396 & 16385 & 6396 & -5580 \\ -4768 & -5580 & 6396 & 16385 & 6396 \\ -960 & -4768 & -5580 & 6396 & 16385 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} 16385 \\ 6396 \\ -5580 \\ -4768 \\ -960 \end{pmatrix} \quad (8.73)$$

Using the *Mathematica* module `LinearSolve[r,g]` to solve this system for f_n we obtain (predictably)

$$f_n = \{1., 0, 0, 0, 0\} \quad n=0,1,2,3,4 \quad (8.74)$$

where, for comparison with continuous time we note the same result:

$$f_0 = \int_{-\infty}^{+\infty} \delta(t) dt = 1$$

The list of f_n coefficients (8.74) can be thought of as a prediction filter because when convolved with the wavelet the output of the filter will predict the entire wavelet. Now, suppose we want a filter that will not predict the entire wavelet but instead the entire wavelet except for the first point. We can then subtract the filter output from the entire wavelet and leave only the first point. This result is called “spiking deconvolution”.

Again using (8.72), this time we will predict all wavelet values except the first point of the wavelet. Before obtaining the value g_1 , the input is shifted one sample point to the right with respect to the desired output. We do not shift the input to the left, which would yield g_{-1} , because we are dealing with a minimum-delay input and negative lags correspond to filter coefficients defined for negative times, and for convergence reasons these are not allowed if the input is minimum-delay; the filter would not converge. In any case, what would be the point of shifting the input to the left as well as to the right with respect to the desired output (the input)? We would simply obtain a crosscorrelation function in Equation (8.72) that would yield a solution vector that is a trivial variation of Equation (8.73); i.e., a time-shifted Dirac delta function, which when convolved with the input would simply shift the entire input.

So we want to predict all wavelet values except, say, the first point of the wavelet. Matrix equation (8.73) becomes

$$\begin{pmatrix} 16385 & 6396 & -5580 & -4768 & -960 \\ 6396 & 16385 & 6396 & -5580 & -4768 \\ -5580 & 6396 & 16385 & 6396 & -5580 \\ -4768 & -5580 & 6396 & 16385 & 6396 \\ -960 & -4768 & -5580 & 6396 & 16385 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} 6396 \\ -5580 \\ -4768 \\ -960 \\ 0 \end{pmatrix} \quad (8.75)$$

Using **LinearSolve** to solve this system for f_n , $n = 0, 1, 2, \dots, 4$, we obtain

$$f_n = 0.928424, -1.10826, 0.720678, -0.517112, 0.179185$$

Convolution of this with the input gives (the desired values corresponding to filter times, f_n , of f_0, f_1, f_2, f_3 are boxed)

$$d_n = \boxed{-74.2739, 10.6729, 57.7215, -2.13029}, 5.45186, -6.88944, -11.3557, 2.21637, 2.15022$$

and these values are clearly not close to the desired prediction $-84, 24, 47, 12$. But this is to be expected because theoretically the required filter must be infinitely long. Recall Step 1 on page 454. We can therefore improve on this result by padding the input wavelet with, say, 10 zeros to make the inverse filter longer and try again. This time the filter coefficients are found to be

$$f_n = 1.04843, -1.39875, 1.19329, -1.20334, 0.953269, -0.865619, 0.661293, -0.566812, 0.419795, -0.341001, 0.239179, -0.177113, 0.108273, -0.0621803, 0.0236195$$

Convolution of the longer f_n with the input wavelet gives (the desired wavelet values we are trying to predict are boxed)

$$d_n = \boxed{-83.8744, 23.8815, 47.1941, 11.7374, 0.297854, -0.40574, 0.449547, -0.614901, \\ 0.654781, -0.892863, 0.880158, -1.17743, 0.966464, -1.22176, 0.477986, -0.512905, \\ -1.05634, 0.363951, 0.283434, 0, 0, 0, 0, 0, 0, 0, 0, 0}$$

These predicted values are closer but we can do better. Padding the wavelet with even more zeros (20) results in filter coefficients $f_0, f_1, f_2, \dots, f_{n-1}$:

$$f_n = 1.05, -1.4025, 1.20012, -1.214, 0.968273, -0.88619, 0.687771, -0.601256, \\ 0.462253, -0.394604, 0.302936, -0.255076, 0.196216, -0.163761, 0.126392, -0.104819, \\ 0.0811928, -0.066997, 0.052071, -0.0427806, 0.0333412, -0.0272765, 0.0212921, \\ -0.017332, 0.0135198, -0.0109209, 0.00846982, -0.0067403, 0.00513621, -0.00395229, \\ 0.00287416, -0.00203268, 0.00130317, -0.000712832, 0.000298954$$

Now convolution of f_n with the input gives (the desired wavelet values are boxed)

which is an excellent result even though the filter f_n is nowhere near one of infinite length. We have successfully predicted the 2nd, 3rd, 4th, and 5th points. All the rest of the predicted values should be close to zero, which they are. The 2nd, 3rd, 4th, and 5th values can now be subtracted from the wavelet leaving only the first point. We have converted the wavelet to an impulse at $t = 0$. Note that the prediction filter coefficients f_n , $n = 0, 1, 2, \dots, n - 1$ always start at time $t = 0$ and that the convolution of f_n with the input wavelet also always starts at time $t = 0$ so that before these predicted values can be subtracted from the input wavelet, they must first be properly lined up with the wavelet, a minor but important step. We now generalize.

Let a_t be a *prediction filter* that when convolved with the input will predict just part of the input, say the part \geq the time α , where α is completely arbitrary. That is, a_t is to be a “prediction filter” so that, given the input *wavelet* x_t , then a_t will predict the value of the wavelet at the time $x_{t+\alpha}$ and at later times. Thus, the desired output y_t is

$$y_t = \sum_{\tau} x_{\tau} a_{t-\tau} = \hat{x}_{t+\alpha} \quad (8.76)$$

where $\hat{x}_{t+\alpha}$ is an estimate of $x_{t+\alpha}$. It is an “estimate” because the filter to be designed will be a least-squares filter, not because of any statistical properties associated with the wavelet. In Equation (8.76), we use the symbol \hat{x} instead of x not for the statistical reasons normally associated with predictive deconvolution,

but because we *know* from our earliest discussions of wavelet shaping (page 425) that to predict $x_{t+\alpha}$ *exactly* would require a filter of infinite length because, repeating for emphasis, to shape a general input to any desired output is a two-step process:

1. Convert the input to an impulse at $t = 0$. This is where the decision about negative and/or positive filter coefficients must be made, and
2. Convolve the unit impulse with the desired output, which clearly does not involve any consideration of filter convergence no matter what the delay properties of the desired output might be. In our present example the desired output is a fragment of the input. This changes nothing. It is always Step 1 that requires the filter of infinite length; the longer the least-squares filter, the closer $\hat{x}_{t+\alpha}$ will approach $x_{t+\alpha}$.

We want to predict the tail end of a wavelet from time $t = \alpha$ to the end of the wavelet so that we can subtract it from the entire wavelet, leaving only the front end. Thus, we shorten the seismic wavelet by truncation. The value of α can be any positive value, so there is flexibility; however, we must acknowledge that a careless choice of α will lead to “hanging wavelets”, that is, to termination of a wavelet at one of its maxima or minima rather than at a zero-crossing. The consequences of this are discussed in this section.

The n -length general least-squares filter results in the matrix formulation

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix} \quad (8.77)$$

where r_t is the autocorrelation of the input, g_t the crosscorrelation of the desired output and the input, and f_t are the filter coefficients.

The filter coefficient f_0 corresponds to the crosscorrelation coefficient g_0 , f_1 to g_1 , etc. In our present example (using a wavelet) the desired output is a piece of the wavelet itself—its tail end. So the value of g_0 is obtained by first positioning the input to the right of the desired output and then doing the usual cross-multiplication and summation to get g_0 . The filter is to be defined only for times $t \geq 0$ so that after convolution of the filter with the input the output will be the tail end of the wavelet, and it will start at $t = 0$, a time *before* the *same wavelet amplitude value* that corresponds to the input. Thus, the output of the filter is commonly referred to as a “time-advanced version” of the input. The “version” of the input wavelet is simply its tail end. That is, the desired output is the back part of the wavelet (not the entire wavelet) shifted toward negative time by some user-specified amount, α .

In order to solve Equations (8.72) we must know the autocorrelation r_t of the input wavelet x_t and the positive-lag coefficients g_t of the crosscorrelation of the desired output and the input.

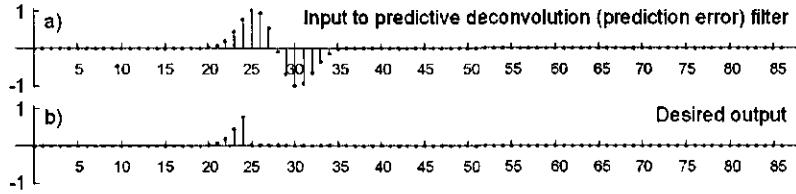


Figure 8.45: (a) Minimum-delay input wavelet. (b) Desired output is the front part of the wavelet. See Figure 8.46 for illustration of the intermediate steps taken to arrive at the desired output. (Prediction.cdr → PDInputAndDesiredOutput.wmf)

The input and desired output of the predictive deconvolution filter are shown in Figure 8.45. The intermediate steps required to accomplish the truncation of the wavelet are shown in Figure 8.46.

A minimum-delay input wavelet is shown in Figure 8.46a. The autocorrelation r_τ of the input wavelet x_t is

$$r_\tau = \sum_t x_t x_{t-\tau}$$

The crosscorrelation (see Figure 8.46) of the desired output and the input is the crosscorrelation of the tail end of the wavelet and the input. The zero-lag correlation coefficient corresponds to the relative position of the input and a copy of itself shifted to the left (time-advanced) by a user-defined amount α . Mathematically the crosscorrelation coefficients g_τ for all lags $\tau \geq 0$ are therefore

$$g_\tau = \sum_t x_{t+\alpha} x_{t-\tau} \quad (8.78)$$

which becomes

$$g_\tau = \sum_t x_{t+\alpha} x_{t-\tau} = \sum_t x_t x_{t-(\tau+\alpha)} = r_{\tau+\alpha}$$

Thus, the crosscorrelation of the desired output and the input is seen to be equal to the autocorrelation of the input for lags $\geq \alpha$ [134]. The desired output is shown in Figure 8.46d. The crosscorrelation function is shown in Figure 8.46c. The normal equations of (8.72) become

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} r_\alpha \\ r_{\alpha+1} \\ \vdots \\ r_{\alpha+n-1} \end{pmatrix} \quad (8.79)$$

We rename f_t to a_t to emphasize that a_t is a “prediction” filter that predicts the tail end of the seismic wavelet from time α to the end of the wavelet. Of course,

in order to do this the filter must be at least as long as the wavelet. The output of the prediction filter is, therefore, that part of the wavelet corresponding to times $\geq \alpha$, as shown in Figure 8.46d. This is the tail end of the wavelet and is what we want to subtract from the input wavelet leaving only the front part. In order to subtract the tail end from the entire wavelet, we multiply the function shown in Figure 8.46d by -1 to yield Figure 8.46e, then shift it back toward positive time by an amount α as shown in Figure 8.46f, and finally subtract this fragment from the original wavelet shown in Figure 8.46a. The result is the deconvolved wavelet shown in Figure 8.46g. Small non-zero values can be seen for times $\geq \alpha$ the reason for which is discussed below. The steps shown in Figure 8.46 are valid no matter what the arrival time of the wavelet shown in Figure 8.46a. More important, this is the same process that takes place when we deal with a normal seismic trace for which we have no knowledge of the shape of the individual wavelet, before or after deconvolution, and which to a first approximation we assume to be time-invariant.

In Figure 8.46a, several numerical values of the input wavelet are labeled on the figure (0.774288, 1.0, 0.939282). The goal of the prediction filter is to predict these numerical values of the wavelet. How well it does this is shown in Figure 8.46d where the value 1.0 is predicted to be 0.97495, a value within about 2.5% of the correct value. We know, however, that there can be only one filter (it is of infinite length) that will produce the correct value and that to improve on the prediction we simply have to make the prediction filter longer. As usual an exact result is possible only if the prediction filter defined by Equation (8.79) is made infinitely long. See the CD-ROM that accompanies this volume for the *Mathematica* program *PredictionPlusMinusXcor* used to generate Figure 8.46.

Figure 8.47 is included to show that the arrival time of the wavelet changes none of the steps in Figure 8.46. The wavelet in Figure 8.47 now starts at $t = 0$ with the portion to be deconvolved shown by a heavy line (Figure 8.47a). The desired output from the prediction filter is now those values of the input wavelet defined for times $t \geq \alpha$ (Figure 8.47b). The negative of (b) is shown in (c). Figure 8.47d is the predicted part of the wavelet delayed by α time units. The final deconvolution is shown in Figure 8.47e, which is the algebraic sum of the input wavelet and the negative of the delayed output of the prediction filter. The final deconvolution is the truncation of the wavelet from time α to the end of the wavelet.

See the CD-ROM that accompanies this volume for the *Mathematica* program *Prediction.nb* used to generate Figure 8.47.

In Figure 8.47a, the value of the input wavelet at $\alpha = 12$ is 0.649. The goal of the prediction filter is to predict these numerical values of the wavelet. The prediction filter predicts a value of 0.636 at α , a value within about 2% of the correct value. A longer prediction filter would predict a value closer to the correct value. An exact result is possible only if the prediction filter defined by Equation 8.79 is made infinitely long. Figure 8.47 was generated by the *Mathematica* program *Prediction.nb*.

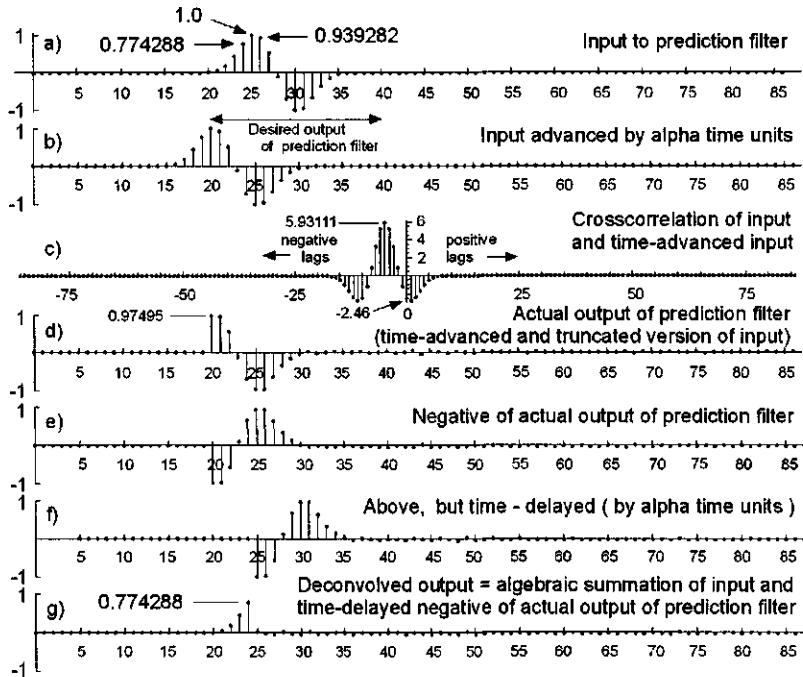


Figure 8.46: Predictive deconvolution broken down into its component steps. Top:
 (a) Input minimum-delay wavelet. (b) Input wavelet time-advanced by $\alpha = 5$ time units and positioned with respect to previous figure for computation of crosscorrelation coefficient g_0 . Note labeled desired output of prediction filter (not prediction error filter). For g_1 move the input one sample point to the right. (c) Full crosscorrelation coefficient function of the wavelet (the input) and the time-advanced input. The crosscorrelation function is the same as the autocorrelation function. Only positive lags $\tau \geq 0$ can be used, however, to obtain g_τ , required for the “prediction filter” because the input is minimum-delay. (d) actual output of prediction filter (a least-squares approximation to the desired output). (e) Negative of above. (f) Above shifted toward positive time (time-delayed) by $\alpha = 5$ time units. (hg) The simple algebraic summation of the entire wavelet in (a) and the delayed, negative part (e) of the output of the prediction filter. The result of the predictive deconvolution is thus a truncated wavelet, as shown. The steps shown in the figure portray the convolution of the input with the predictive deconvolution (also called the prediction error filter) filter $1, \dots, \alpha - 1$ zeros, $\dots, -a_0, -a_1, -a_2, \dots, -a_{n-1}$ where a_i are the coefficients of the prediction filter. This is the same process that takes place for a seismic trace for which we have no knowledge of the shape of the source wavelet. See text for significance of numerical values of wavelet amplitudes. See also Figure 8.47. (PredictionPlusMinusXcor.nb → Prediction.cdr → Prediction2.wmf)

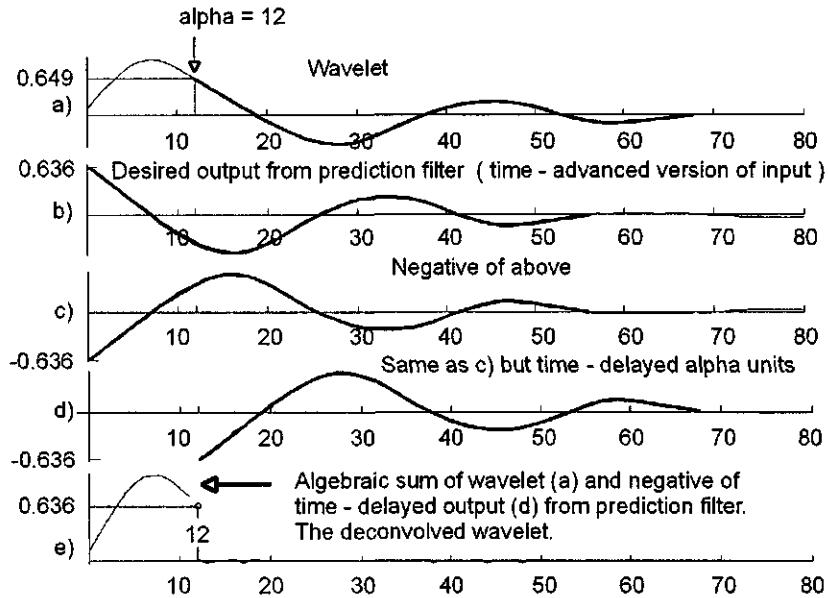


Figure 8.47: Similar to Figure 8.46 but with input starting at $t = 0$. Top:
 (a) Minimum-delay wavelet with portion to be deconvolved highlighted by heavy line. (b) Output of prediction filter is the part of the wavelet for times $t \geq \alpha$ that has been shifted toward negative time (advanced) by an amount of time, $t = \alpha$. Thus, the output of the prediction filter is a time-advanced “version” of the input, with zero amplitudes for times $t < \alpha$ simply because those amplitudes were not predicted in the first place (filter coefficients f_i computed for $t < 0$ would not converge because the input is minimum-delay). The “version” is that part of the wavelet (heavy line) at and after the time, α . (c) The negative of (b). (d) The negative of the predicted part of the wavelet (c) shifted back toward positive time (delayed) by the amount of time, α . (e) The simple algebraic summation of the entire wavelet, (a), and the delayed, negative part (c) of the output of the prediction filter. The result of the predictive deconvolution is thus a truncated wavelet. Process shown in figure is the graphical equivalent of the mathematical convolution of the predictive deconvolution filter $\{1, \dots, \alpha - 1 \text{ zeros} \dots -f_0, -f_1, -f_3, \dots, f_{n-1}\}$ with the wavelet, where f_i are the coefficients of the prediction filter. See text for significance of numerical values of wavelet amplitudes. See Figure 8.46 for a similar graphical interpretation of predictive deconvolution as a truncation process. Figure generated by *Mathematica* program Prediction.nb. (Prediction.nb → Prediction.cdr → Prediction.wmf)

This graphic introduction to predictive deconvolution can be summarized mathematically as follows:

1. Decide on the desired value of the “prediction distance” α . More on that later. See especially Section 8.3.3 on page 491.
2. Solve Equation (8.79) to obtain the coefficients of the “prediction” filter”.
3. Convolve the prediction filter with the input. The output of this filter is a time-advanced part of the input, which is the tail of the input wavelet for times $t \geq \alpha$.
4. Multiply the prediction filter coefficients by -1 .
5. Time-delay the output of the prediction filter by α time units.
6. Subtract this result from the input to obtain the final predictive deconvolution.
7. Accomplish the above predictive deconvolution in a single step by forming a filter called the “prediction error operator” given by

$$1, \underbrace{0, 0, \dots, 0}_{\alpha-1 \text{ zeros}}, -a_0, -a_1, \dots, -a_{n-1} \quad (8.80)$$

and then simply convolve this composite prediction error filter with the input to obtain the predictively deconvolved (truncated) wavelet.

Peacock and Treitel [134, 153] showed that the prediction error operator defined by Equation (8.80) can be formulated as a “spiking deconvolution” filter by augmenting the prediction operator matrix of Equation (8.72) with α set equal to unity resulting in

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \quad (8.81)$$

This system can be written as n simultaneous linear equations,

$$\begin{aligned} r_0 a_0 + r_1 a_1 + \dots + r_{n-1} a_{n-1} &= r_1 \\ r_1 a_0 + r_0 a_1 + \dots + r_{n-2} a_{n-1} &= r_2 \\ &\vdots && \vdots \\ r_{n-1} a_0 + r_{n-2} a_1 + \dots + r_0 a_{n-1} &= r_n \end{aligned}$$

Now modify this system of equations by subtracting the coefficient r_i from each side of the equation. This changes nothing, of course. The original system is

enclosed in a box:

$$\begin{array}{l} -r_1 + \boxed{r_0a_0 + r_1a_1 + \cdots + r_{n-1}a_{n-1} = r_1} \quad -r_1 \\ -r_2 + \boxed{r_1a_0 + r_0a_1 + \cdots + r_{n-2}a_{n-1} = r_2} \quad -r_2 \\ \vdots \qquad \qquad \vdots \qquad \qquad \vdots \\ -r_n + \boxed{r_{n-1}a_0 + r_{n-2}a_1 + \cdots + r_0a_{n-1} = r_n} \quad -r_n \end{array}$$

Next augment this system in the following form, where β will be defined below.

$$\begin{array}{l} -r_0 + r_1a_0 + r_2a_1 + \cdots + r_na_{n-1} = -\beta \\ -r_1 + \boxed{r_0a_0 + r_1a_1 + \cdots + r_{n-1}a_{n-1} = r_1} \quad -r_1 \\ -r_2 + \boxed{r_1a_0 + r_0a_1 + \cdots + r_{n-2}a_{n-1} = r_2} \quad -r_2 \\ \vdots \qquad \qquad \vdots \qquad \qquad \vdots \\ -r_n + \boxed{r_{n-1}a_0 + r_{n-2}a_1 + \cdots + r_0a_{n-1} = r_n} \quad -r_n \end{array}$$

Multiply each equation above by -1 and get the system

$$\begin{array}{l} r_0 -r_1a_0 -r_2a_1 -\cdots -r_na_{n-1} = \beta \\ r_1 -r_0a_0 -r_1a_1 -\cdots -r_{n-1}a_{n-1} = 0 \\ \vdots \qquad \qquad \vdots \\ r_n -r_{n-1}a_0 -r_{n-2}a_1 -\cdots -r_0a_{n-1} = 0 \end{array}$$

for which the associated matrix equation is

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_n \\ r_1 & r_0 & \cdots & r_{n-1} \\ \vdots & & & \\ r_n & r_{n-1} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ -a_0 \\ \vdots \\ -a_{n-1} \end{pmatrix} = \begin{pmatrix} \beta \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (8.82)$$

where

$$\begin{aligned} \beta &= \sum_{i=0}^n b_i r_i \\ \text{and} \\ b_i &= -a_{i-1} \quad (i = 1, \dots, n) \end{aligned}$$

The “prediction error operator” defined by the filter in Equation (8.82) is the same as Equation (8.80),

$$1, \underbrace{0, 0, \dots, 0}_{\alpha-1 \text{ zeros}}, -a_0, -a_1, \dots, -a_{n-1}$$

that was obtained using a more conceptual approach and which, with $\alpha = 1$, becomes the prediction error operator of Equation (8.82).

In like manner, the general Equation (8.79)

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_{n-1} \\ r_1 & r_0 & \cdots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} r_\alpha \\ r_{\alpha+1} \\ \vdots \\ r_{\alpha+n-1} \end{pmatrix}$$

can be written

$$\begin{aligned} r_0 a_0 + r_1 a_1 + \cdots + r_{n-1} a_{n-1} &= r_\alpha \\ r_1 a_0 + r_0 a_1 + \cdots + r_{n-2} a_{n-1} &= r_{\alpha+1} \\ &\vdots \\ r_{n-1} a_0 + r_{n-2} a_1 + \cdots + r_0 a_{n-1} &= r_{\alpha+n-1} \end{aligned} \quad (8.83)$$

and this system of equations can be augmented in a manner similar to that used for $\alpha = 1$, as developed in references [134, 153]. Only the result is given here because the procedure is similar to that used to obtain Equation (8.82). After augmenting (8.83), the associated matrix equation is

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_{\alpha+n-1} \\ r_1 & r_0 & \cdots & r_{\alpha+n-2} \\ \vdots & & & \\ r_{\alpha-1} & r_{\alpha-2} & \cdots & r_n \\ r_\alpha & r_{\alpha-1} & \cdots & r_{n-1} \\ \vdots & & & \\ r_{\alpha+n-1} & r_{\alpha+n-2} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -a_0 \\ \vdots \\ -a_{n-1} \end{pmatrix} = \begin{pmatrix} \rho_0 \\ \rho_1 \\ \vdots \\ \rho_{\alpha-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (8.84)$$

where [134, 153]

$$\begin{aligned} \rho_0 &= r_0 - (r_\alpha a_0 + r_{\alpha+1} a_1 + \cdots + r_{\alpha+n-1} a_{n-1}) \\ \rho_1 &= r_1 - (r_{\alpha-1} a_0 + r_\alpha a_1 + \cdots + r_{\alpha+n-2} a_{n-1}) \\ &\vdots \\ \rho_{\alpha-1} &= r_{\alpha-1} - (r_1 a_0 + r_2 a_1 + \cdots + r_n a_{n-1}) \end{aligned} \quad (8.85)$$

Numerical values of a_i in (8.84) would be the same as those found using Equation (8.79).

We thus have two formulations for the prediction error filter. The first is:

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_{n-1} \\ r_1 & r_0 & \cdots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} r_\alpha \\ r_{\alpha+1} \\ \vdots \\ r_{\alpha+n-1} \end{pmatrix} \quad (8.79)$$

from which we form

$$1, \underbrace{0, 0, \dots, 0}_{\alpha-1 \text{ zeros}}, -a_0, -a_1, \dots, -a_{n-1} \quad (8.80)$$

and the second is

$$\begin{pmatrix} r_0 & r_1 & \cdots & r_{\alpha+n-1} \\ r_1 & r_0 & \cdots & r_{\alpha+n-2} \\ \vdots & & & \\ r_{\alpha-1} & r_{\alpha-2} & \cdots & r_n \\ r_\alpha & r_{\alpha-1} & \cdots & r_{n-1} \\ \vdots & & & \\ r_{\alpha+n-1} & r_{\alpha+n-2} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -a_0 \\ \vdots \\ -a_{n-1} \end{pmatrix} = \begin{pmatrix} \rho_0 \\ \rho_1 \\ \vdots \\ \rho_{\alpha-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (8.82)$$

which is a system of equations that could be solved directly for the prediction error operator by using (8.85). The column vector of crosscorrelation coefficients in (8.84), however, requires the prediction filter a_i so clearly it is more computationally efficient to use (8.79) and then form (8.80). Numerical values of the prediction error coefficients from the two systems of equations are now compared.

Equations (8.80) and (8.84) are two formulations for predictive deconvolution and should produce the same results. Comparisons are shown in Figure 8.48 along with the de-reverberation Backus filters that will be discussed later.

The best way to compare the two formulations is to examine the numerical values of the filter coefficients themselves. Numerical results for the prediction error filter using Equation (8.80) are

$$\begin{aligned} 1, 0, 0, 0, 1.26539, 0, 0, 0, 0.323901, 0, 0, 0, -0.040131, \\ 0, 0, 0, 0.0051624, 0, 0, 0, -0.00219923, 0, 0, 0, 0.0128707, \\ 0, 0, 0, -0.103337, 0, 0, 0, -0.0949912 \end{aligned}$$

and the same result but using Equations (8.84) and (8.85) is

$$\begin{aligned} 1., 0., 0., 0., 1.26539, 0., 0., 0., 0.323901, 0., 0., 0., -0.040131, \\ 0., 0., 0., 0.0051624, 0., 0., 0., -0.00219923, 0., 0., 0., 0.0128707, \\ 0., 0., 0., -0.103337, 0., 0., 0., -0.0949912, 0. \end{aligned}$$

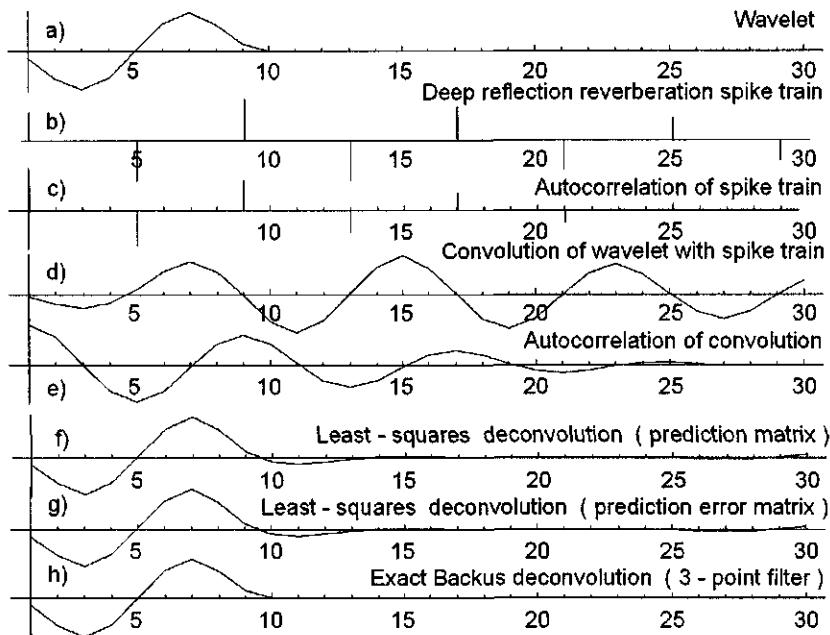


Figure 8.48: Comparison of three deconvolution filters. (a) through (e) as labeled. (f) Using Equation 8.80, (g) Using Equation 8.84, (h) Using exact 3-point Backus filter. Excellent agreement between (f) and (g). Excellent agreement with the exact deconvolution shown in (h) could be achieved for (f) and (g) by increasing length of the least-squares filter. Wavelet is minimum-delay, but this is not important. (PDTester.nb → PDTester.cdr → PDCompare.wmf)

The above filter output was produced by the *Mathematica* program “PDTester”, which also includes a short section for the generation of minimum-, mixed-, or maximum-delay wavelets by choosing roots outside, both outside and inside, or just inside the unit circle, respectively. The program is included on the CD-ROM that accompanies this volume. The user is advised to run it as is for the first time, then remove the pairs of characters (**) that comment out some of the intermediate steps. See the CD-ROM that accompanies this volume for the *Mathematica* Program PDTester.nb used to generate the above filter coefficients. The number of equations given by Equation (8.84) and solved by *Mathematica* module **LinearSolve** (or any other computer program) clearly contains fewer filter coefficients than the same size system of Equation (8.79) for determining the prediction filter. In order to solve for the same number of non-zero filter coefficients f_n the system of Equation (8.79) must simply be increased by the prediction distance α using the statement

```
PDFilterLength=LengthOfEffectiveSourceWavelet+alpha;
```

For reverberations, the “length of the effective source wavelet” is defined as the

approximate length of the reverberation spike train. Much insight into inverse filtering can be obtained by examining the numerical values of the predictive deconvolution filter and comparing these values with those obtained from other approaches that are described earlier in this chapter. One of the examples we now discuss is the excellent tutorial figure already shown in Figure 8.28 from Robinson and Treitel [153] for wavelet shaping and repeated here for convenience (Figure 8.49). Those authors used crosscorrelation coefficients to shape the input wavelet to a unit impulse; here we do the same thing with predictive deconvolution. Predictive deconvolution must arrive at the same result *because there is only one theoretically correct inverse filter for a given application*. Predictive deconvolution, however, shapes wavelets by truncation, which we can do in one step (Step 1 on page 454). If the desired output is something other than the numerical value of the first point in the wavelet, then Step 2 (page 454) is also required. Thus, exploring with predictive deconvolution the examples shown in Figure 8.49 nicely separates the step associated with convergence of the filter coefficients from that required for subsequent shaping of the impulse obtained from the first step.

The first example is the predictive deconvolution of the wavelet $(2, 1)$ in Figure 8.49 to “spike out” the wavelet and convert it to an impulse at $n = 0$. This only requires Step 1.

The general least-squares matrix equation is

$$\begin{pmatrix} r_0 & r_1 \\ r_{-1} & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \end{pmatrix}$$

The equation for the prediction filter f is

$$\begin{pmatrix} r_0 & r_1 \\ r_{-1} & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} r_\alpha \\ r_{\alpha+1} \end{pmatrix} \quad (8.86)$$

where r contains the autocorrelation coefficients and α is the prediction distance. For the first example in Figure 8.49, Equation (8.86) becomes

$$\begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

where $\alpha = 1$ and therefore $g_0 = r_1 = 2$. The prediction error filter coefficients are found to be

$$1, -0.47619, 0.190476$$

and the convolution of the filter with the wavelet gives

$$2, 0.047619, -0.0952381, 0.190476$$

The desired value at $t = 0$ is “1”, however, not “2”. A unit impulse is easily obtained by scaling the output by 0.5 (Step 2). The values after $t = 0$ should

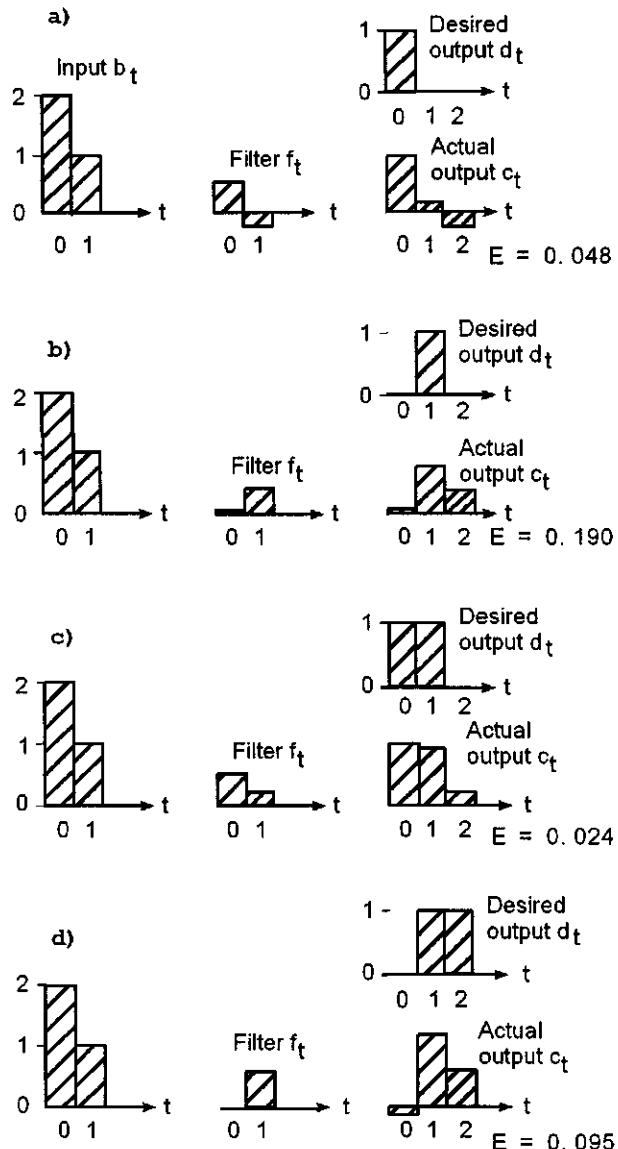


Figure 8.49: Same as Figure 8.28, repeated here to show how predictive deconvolution would have to be used to obtain the same results as least-squares wavelet shaping. Step 1 would require conversion of the input (left column) to an impulse (not necessarily a unit impulse) at $t = 0$ by a spiking deconvolution. The desired output (right column) is obtained by convolving the impulse obtained at $t = 0$ with the appropriate function (see text) that will simply scale and time-shift the impulse. This explicitly demonstrates the conceptual two-step approach defined on page 465. The quantity E is the minimum error energy given by Equation (8.47). (Figure6-2a1RobinsonAndTreitel.nb, Figure62FromRobinsonAndTreitel.cdr,wmf)

be zero, so a better result can be reached by making the filter longer. Padding the input wavelet with ten zeros gives for Equation (8.86)

$$\begin{pmatrix} 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 2 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (8.87)$$

Solving for f_n gives (the boxed value corresponds to $t = 0$):

$$f_n = \{ \boxed{0.5}, -0.25, 0.125, -0.0625, 0.03125, -0.015624, 0.0078106, -0.0039024, 0.0019455, -0.0009613, 0.00045776, -0.00018311 \}$$

which are the same as the theoretically exact coefficients obtained using z -transform theory and shown in Equation (8.56). For comparison, they are reproduced here:

$$f[n] = (\boxed{0.5}, -0.25, 0.125, -0.0625, 0.03125, -0.015625, 0.0078125, -0.0039062, 0.0019531, -0.00097656, 0.00048828, -0.00024414)$$

A better result can be obtained by padding the wavelet with, say, 20 zeros, obtaining prediction filter coefficients of

$$\begin{aligned} 0.5, -0.25, 0.125, -0.0625, 0.03125, -0.015625, 0.0078125, -0.00390625, \\ 0.00195312, -0.000976562, 0.000488281, -0.000244141, 0.00012207, \\ -0.0000610340, 0.0000305171, -0.0000152579, 7.62753 \times 10^{-6}, \\ -3.81097 \times 10^{-6}, 1.8999 \times 10^{-6}, -9.38773 \times 10^{-7}, \\ 4.47035 \times 10^{-7}, -1.78814 \times 10^{-7} \end{aligned}$$

and the prediction *error* coefficients (the above multiplied by -1) are seen to be nearly the same except for a small error introduced by numerical computation. Convolution of the prediction error filter with the wavelet gives essentially

$$2, 0, 0, \dots, 0$$

This completes Step 1. The desired output at $t = 0$ is "1", not "2". Step 2 (page 454) is a simple scaling of the filtered output by the constant 0.5 to obtain the desired output shown in Figure 8.49.

The second example in Figure 8.49 can be set up the same way:

$$\begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

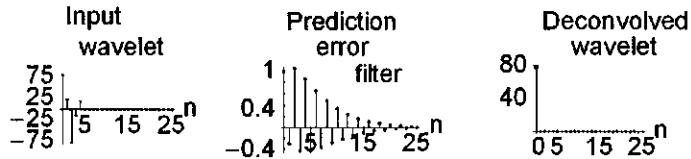


Figure 8.50: Spiking deconvolution of a minimum-delay wavelet. Prediction distance $\alpha = 1$. (PredictiveDeconvolution.nb → PredictiveDeconvolution-Wavelets.cdr → PDmin2.wmf)

from which, after padding the wavelet with ten zeros, we also obtain

$$2, 0, 0, \dots, 0 \quad (8.88)$$

but the desired output is

$$(0, 1, 0)$$

so Step 2 can be accomplished by a simple convolution of (8.88) with $(0, 0.5, 0)$ to obtain $(0, 1, 0)$ as shown in Figure 8.49b.

Figure 8.49c is obtained by convolving (8.88) with $(0.5, 0.5)$ to obtain the desired output $(1, 1)$. Figure 8.49d is obtained by convolving (8.88) with $(0.0, 0.5, 0.5)$ to obtain the desired output $(0, 1, 1)$. These simple exercises are included to help visualize the relationship between generalized least-squares filters whose output can be an arbitrary shape, and predictive deconvolution, whose output is a truncated shape. See the CD-ROM that accompanies this volume for the *Mathematica* Program Figure6-2RandTbyPD.nb used to generate the above four examples.

The above examples included “spiking deconvolution” for which the filter was designed to spike out the wavelet. In practice, this is not desireable because of high (as well as low) frequencies necessarily amplified in order to arrive at the shorter “spike” wavelet, which contains all frequencies (recall that $\delta(t) \iff 1$). All seismic data have some noise and the noise as well as the signal frequencies will be enhanced as the wavelet is shortened in time. Seismic data are also band-limited and efforts to produce an impulse will enhance noise. A less-demanding deconvolution is simply to make the wavelet shorter but not go to the extreme of reducing it to a single point, as shown in the following example.

Levinson solution to Toeplitz matrix

```
(* a is the one-sided autocorrelation function of the wavelet
f is the solution vector of filter coefficients that
spikes out the wavelet *)
-----Begin subroutine-----
Levinson[a_] := (f = Table[0, {Length[a]}]; g = f;
f[[1]] = 1; g[[1]] = 1; f[[2]] = -a[[2]]/a[[1]];
```

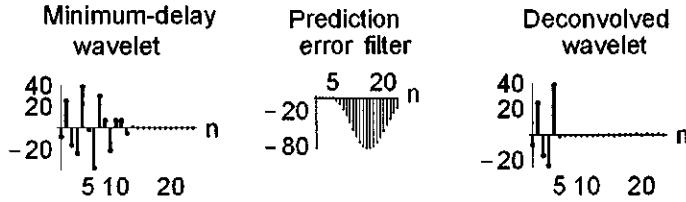


Figure 8.51: Deconvolution of a longer minimum-delay wavelet. Prediction distance $\alpha = 6$. From $n = 6$ and after, the wavelet has been zeroed by the prediction error filter (center). (PredictiveDeconvolutionWavelets.nb → PredictiveDeconvolutionWavelets.cdr → PDminlong.wmf)

```
v = a[[1]] - a[[2]]^2/a[[1]];
Do[e = 0; Do[e = e + a[[j - i + 1]]*f[[i]], {i, 1, j - 1}];
c = e/v; v = v - c*e;
Do[g[[i]] = f[[i]] - c*f[[j - i + 1]], {i, 2, j - 1}];
g[[j]] = -c;
Do[f[[i]] = g[[i]], {i, 1, Length[a]}], {j, 3, Length[a]}];
filter = f;);
```

Non-spiking deconvolution

As a numerical example, we convolve several minimum-delay, 2-point wavelets to obtain the input wavelet

$$\begin{aligned} b[n] = & (-8.35006, 25.1414, -16.4604, -24.1863, 38.9151, -1.66489, \\ & -37.1444, 29.8559, 6.22231, -21.1912, 6.72102, 5.91688, -4.78095, 1) \end{aligned}$$

The deconvolution is shown in Figure 8.51.

The autocorrelation coefficients are computed from the wavelet itself, which we do not normally see. No random elements are involved so our approach can be called deterministic. Results for predictive deconvolution of a more realistic wavelet are shown in Figure 8.52. The coefficients for this minimum-delay wavelet, which is used in many of the illustrations that follow, are

```
wavelet={0.132984,0.342512,0.548545,0.744799,0.907561,1.04204,1.12834,1.16132,1.14894,
1.07934,0.981866,0.870631,0.753987,0.641438,0.534492,0.422551,0.300063,0.196714,0.099406,
-0.0225922,-0.125436,-0.215479,-0.306348,-0.384165,-0.45418,-0.516001,-0.557161,-0.584517,
-0.589772,-0.579596,-0.54334,-0.492505,-0.425367,-0.349645,-0.265985,-0.181953,-0.100738,
-0.0363371,0.0281819,0.0917981,0.151973,0.195556,0.232337,0.257646,0.271141,0.277681,
0.272208,0.267031,0.250813,0.217928,0.171922,0.119909,0.0633082,0.0171123,-0.03006,
-0.0776557,-0.12405,-0.153802,-0.168138,-0.165966,-0.155284,-0.136698,-0.113179,-0.0910421,
-0.0715249,-0.0540389,-0.0337813,-0.0130044,-0.000170491,0.010792,0.0176374,0.0211389,
0.0197672,0.0179044,0.0211134,0.0205213,0.0152477,0.0159675,0.0136121,0.0123929,
0.00978807,0.00622238,-0.00373036,-0.00802724,-0.0172832,-0.0264725,-0.0327029,-0.033411,
-0.0401914,-0.0381475,-0.0384762,-0.0348436,-0.0293412,-0.0217437,-0.0167644,-0.00920205,
-0.00332008,0.00399458,0.00993271,0.0162817,0.0222198,0.0267815,0.0290488,0.0287685,
0.0299113,0.0301194,0.0297948,0.0272967,0.0234607,0.018213,0.0125574,0.00656805,0.000936034};
ListPlot[wavelet, PlotJoined->True, PlotRange->All];
```

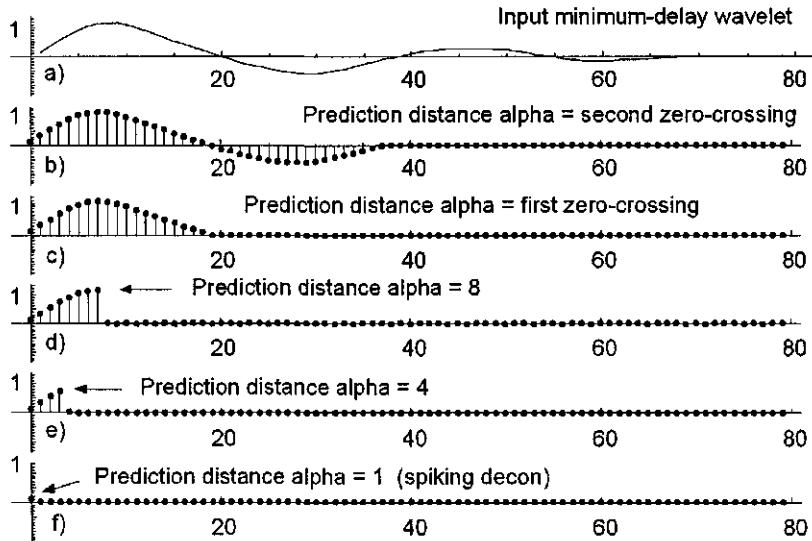


Figure 8.52: Predictive deconvolution of minimum-delay wavelet. Top: (a) Input wavelet. (b) Wavelet deconvolved to second zero-crossing of wavelet ($\alpha = 39 = 78$ ms). (c) Wavelet deconvolved to first zero crossing ($\alpha = 19 = 38$ ms). (d) Gap = time to maximum amplitude of wavelet ($\alpha = 16 = 32$ ms). (e) $\alpha = 8 = 16$ ms. (f) Spiking deconvolution ($\alpha = 1 = 2$ ms). All figures to same relative scale. Note preference for $1 < \alpha < 8$ for better S/N ratio. (PredictiveDeconvolution2msWavelet.nb → PDdt2msGaps.cdr → PDdt2msGaps.wmf)

8.3.3 What does predictive deconvolution do?

From Equation (8.86), predictive deconvolution predicts the values, after a specified prediction distance, of each point in the seismic wavelet and then subtracts these values from the wavelet. Predictive deconvolution therefore simply shortens the length of the wavelet by truncation; it does not take all of the energy of the wavelet and concentrate it toward the front. It is a simple truncation. Any prediction distance can be used. If the prediction distance is unity ($\alpha = 1$), then this is known as a “spiking deconvolution”, and the filtered output will simply be a wavelet that has been shortened to an impulse (see Figure 8.52). In the ideal case, the impulse will be directly proportional to the reflection coefficient (a nice goal, but not too practical!). If the prediction distance is any other value, then the wavelet will (ideally, in a noise-free environment) be *truncated* after the prediction distance (for example, see Figure 8.51); that is, wavelet shortening is accomplished by wavelet truncation. For values of $\alpha > 1$ the deconvolution is called a “gapped deconvolution”. If the prediction distance can be chosen such that it coincides with a zero-crossing of the (generally unknown) wavelet, then the amplitude values of the wavelet will increase from an initial

value of something close to zero and terminate near a value of zero, perhaps, say, near the second zero crossing of the wavelet. Note, however, that re-shaping of the wavelet by predictive deconvolution is done by a simple truncation of the wavelet, not by repositioning the zero crossings of the wavelet. It is clear that the process of predictive deconvolution does not redistribute the total energy in the wavelet toward the front of the wavelet leaving the total energy of the wavelet unchanged, because what we do is to successfully predict an actual numerical value of any point and discard everything after that, so wavelet energy is being discarded. From another point of view, in order to arrive at a shortened wavelet, we are essentially spiking out the wavelet and then convolving that impulse with a desired output that is the front part of the wavelet (although we don't explicitly perform such a separate, second convolution). But to spike out the input (any input of any delay characteristics) requires a filter of infinite length (there can only be one theoretically correct inverse filter that converts the input to a "1" at $t = 0$). A consequence of this observation is that the longer the predictive deconvolution filter is, the more perfect the desired prediction will be. Filters of infinite length, however, are not practical. Therefore we don't get a perfect deconvolved output, only one that is "perfect" in a least-squares sense. Examining the predictive deconvolution of wavelets with long filters does, however, provide insight into what the mathematics is attempting to do, and one conclusion we can make is that wavelet energy is not being redistributed. If the prediction is not perfect then it is because we can do nothing else but choose a filter of finite length.

What are the consequences of predictive deconvolution? We can get some insight by examining what happens in a noise-free (synthetic) or low-noise (high S/N ratio) environment. Synthetic data have always been useful for this purpose. A review of the preceding mathematical development suggests the following:

1. Predictive deconvolution is a wavelet truncation process.
2. If the prediction distance is not chosen carefully, then the result of predictive deconvolution will be "hanging wavelets" where the wavelet amplitude suddenly drops to zero. This is easily seen when dealing with noise-free data (Figure 8.52), but will not be as obvious in the usual situation of hundreds of overlapping wavelets.
3. If the prediction distance is chosen to be unity ($\alpha = 1$), then predictive deconvolution discards most of the energy of the wavelet (everything after the first point) and what remains is simply the first value of the wavelet, which is always of low amplitude anyway. Therefore, a "spiking decon" can drastically lower the S/N ratio of the wavelet (or of the many wavelets that are embedded in the seismic trace). What, therefore, is the point of this common practice of "spiking-decon"? Although our objective can be clearly stated, i.e., to convert wavelets to unit impulses ("spikes") thereby improving seismic resolution, this is an ambitious goal. To repeat, we know from Chapter 3.5.4 that a unit impulse contains all frequencies from

zero to infinity, each frequency with the same relative amplitude. We would be required to design a filter that would not only amplify (in a relative sense) the high-frequency components of the wavelet spectrum, but the low-frequency ones as well; however, the relatively low and high frequencies were not recorded in the field in the first place! Any noise present in the wavelet (or trace) will be amplified along with the same frequencies that are present in the signal. The result is a noisy seismic trace. Predictive deconvolution can therefore drastically lower the S/N ratio. It is common practice to observe a noisy trace after a spiking deconvolution. The next generally accepted step is to use a "cosmetic" bandpass filter to limit the frequency content to the original bandwidth of the wavelet and to eliminate the high-frequency noise caused by the deconvolution; however, the S/N ratio of the data might already have been severely compromised. The dynamic range of the seismograph is necessarily limited by the number of bits in the A/D converter, the seismometer frequency response, etc. Said another way, if you want high-resolution seismic data, then you must record the low-frequencies as well as the high *at the same time*. We do the best we can, and this turns out to be quite good indeed.

4. After predictive deconvolution, the truncated wavelet may be minimum-, mixed-, or maximum-delay. Although deconvolution results in a wavelet of shorter duration, the seismic trace still does not yield up an isolated wavelet whose delay characteristics can be directly examined. One can autocorrelate the seismic trace and use Hilbert transforms to recover a wavelet, but the recovered wavelet will always be minimum-delay (see the discussion in Section 8.5.1), and this might not be even close to the actual wavelet shape that left the sourcepoint.

From a mathematical standpoint, the choice of the prediction distance α is arbitrary, but it is useful to examine what happens in the ideal noise-free case if α is intentionally chosen to coincide with the maximum amplitude of the wavelet. A minimum-delay wavelet is shown in Figure 8.53 with its zeros shown in Figure 8.53a. The numerical values of the wavelet used in the figure are

$$(2.10452, 2.38015, -10.8688, -12.3648, 23.3776, 26.7594, -26.8054, \\ -30.8804, 17.2808, 20.0417, -5.93884, -6.936, 0.85, 1)$$

and the input wavelet is shown as a "stick-o-gram" in Figure 8.53b, and as a more conventional continuous plot in Figure 8.53c. The prediction distance was chosen to be $\alpha = 6$.

The prediction distance α is also referred to as the "gap" by some software packages and the filtering process described above is known as "gapped deconvolution". A commonly accepted value of α can be obtained by examining the autocorrelation function and choosing the "second zero crossing". Truncation of the wavelet after the second crossing would leave one positive and one negative lobe, more or less, and this result is at least an approximation to a

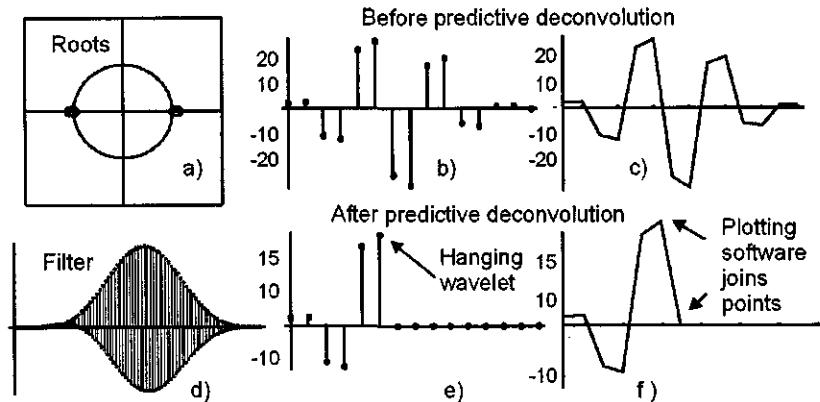


Figure 8.53: (a) Zeros of wavelet shown in (b) as a line plot and in (c) as a continuous plot. $\alpha = 6$ chosen to coincide with top of the wavelet. From $n = 6$ on, wavelet is zeroed out by filter in (d). (e) and (f) Output from predictive deconvolution. Truncation of wavelet is clear in (e), but not in (f) because conventional plotting software draws a continuous line from the top of the wavelet to the zero axis. Wavelet after predictive deconvolution is not minimum-delay. HangingWavelet.nb → PredictiveDeconvolution-Wavelets.cdr → HangingWavelet.wmf

real, physically realizable wavelet. In any case, resolution would be considerably increased after predictive deconvolution. Some software packages have an option to have the second zero-crossing chosen automatically by a search algorithm. In practice, an autocorrelogram of unstacked shot traces is generated and the interpreter picks a few zero crossings and lets the computer do a linear interpolation for the others. In the examples of Figures 8.44, 8.54, and 8.55, however, it is clear that there is no near-zero value of the wavelet near the second zero-crossing of the autocorrelation. These examples are included to emphasize that we do not know what is happening to the shape of the wavelet in a seismic trace. The zero-crossings of the autocorrelation do not coincide with the zero-crossings of the wavelet. The second zero-crossing is, however, commonly chosen because it produces a deconvolved wavelet that resembles a physically realizable minimum-delay wavelet. We suggest an alternative choice of α on page 491.

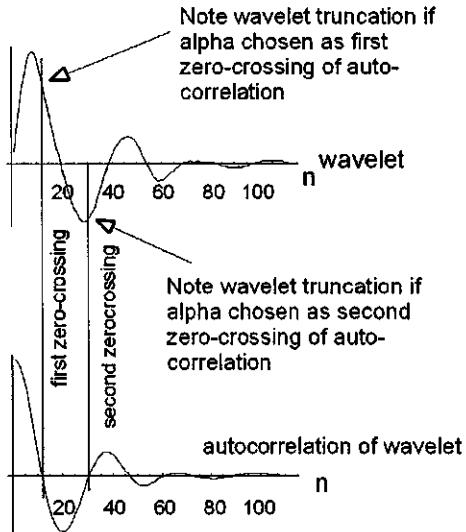


Figure 8.54: Top: The same wavelet of Figure 8.52. Bottom: Autocorrelation of this wavelet. Only the positive lags are plotted. Note that the zero-crossings of the autocorrelation do not coincide with the zero-crossings of the wavelet (arrows). The second zero-crossing is, however, commonly chosen and is generally more desirable than choosing the “first zero-crossing” if the desired result is a wavelet shape that resembles a physically realizable minimum-delay wavelet. SecondZeroCrossing.nb → SecondZeroCrossing.cdr → SecondZeroCrossing.wmf

Predictive deconvolution and mixed-delay wavelets

The matrix equation for predictive deconvolution, Equation (8.79), is

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} r_\alpha \\ r_{\alpha+1} \\ \vdots \\ r_{\alpha+n-1} \end{pmatrix} \quad (8.89)$$

where the input is shifted to the right with respect to the desired output in order to compute the crosscorrelation coefficients. Predictive deconvolution applied to wavelets is therefore effective only for minimum-delay wavelets. In Figure 8.56 a minimum-delay wavelet and a mixed-delay wavelet (a Ricker wavelet) are shown along with their deconvolved counterparts. It is clear that the deconvolution is successful only for the minimum-delay wavelet.

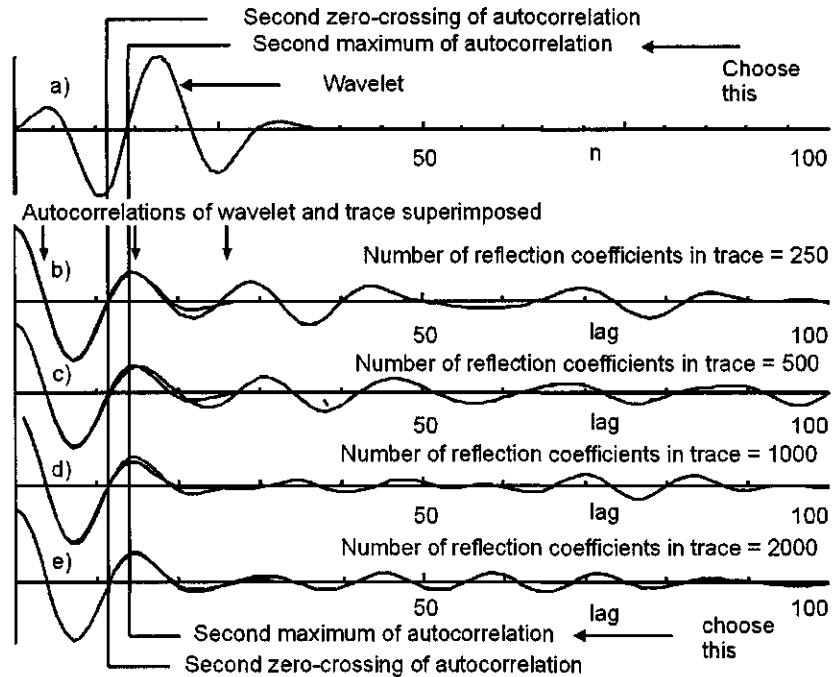


Figure 8.55: (a) Wavelet. (b-e) Choosing the second zero-crossing of the trace autocorrelation results in wavelet truncation, as shown in (a). Choosing second peak of trace autocorrelation minimizes wavelet truncation, as shown in (a). (AutocorrelationOfASeismicTrace.nb → RandT1980Appendix6-1.cdr → ChooseAuto.wmf)

Deconvolution of a seismic trace

Each interface in a layered medium gives rise to reflected and transmitted wavelets. We showed earlier that, for an interface reflection coefficient as high as ± 0.3 then 0.92 % of the amplitude of the incident wave will be recovered after two-way transmission through the interface. For our purposes here it will therefore be sufficient to neglect transmission coefficients and just consider reflection coefficients associated with each interface.

The approximation of the reflectivity function (the series of reflection coefficients) by a sequence of random numbers has been shown to be a highly successful part of the convolutional model of reflection seismology [164]. That is, if the reflection coefficients contained within different time gates are examined, they are uncorrelated. Thus, convolution of a minimum-delay source wavelet with a sequence of random numbers is a useful tutorial model to examine predictive deconvolution. Such a wavelet and a set of reflection coefficients are shown in Figure 8.57.

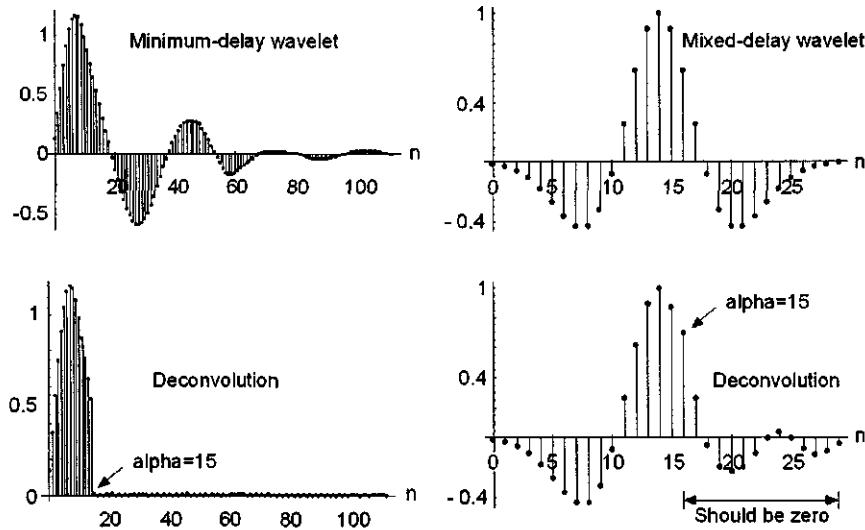


Figure 8.56: Predictive deconvolution (PD) applied to non-minimum-delay wavelets. Top left: Minimum-delay wavelet. Top right: mixed-delay wavelet (Ricker wavelet). Prediction distance $\alpha = 15$ for each. Bottom left: Deconvolution successful for minimum-delay wavelet. Bottom right: Deconvolution not successful. Similar comparisons are obtained for any value of α . Delaying the input in order to obtain a better approximation to the desired output (Figure 8.38) does no good in the case of PD because the entire crosscorrelation function is not used in (8.89) whereas it is used in (8.61). `PDAndNonMinimumDelay.nb` → `PDAndNonMinimumDelay.cdr` → `PDAndNonMinimumDelay.wmf`

For Figure 8.57, the autocorrelation coefficients required for the predictive deconvolution were determined from the wavelet itself, which, for purposes of discussion, was minimum-delay and assumed to be known. The sequence of reflection coefficients is shown in Figure 8.57c. It is important to note that the two-dimensional space-time sequence of reflection coefficients is in fact the representation of the geologic layering that we seek. How the reflection coefficients line up in space and time is an image of the subsurface geometry. If the wavelet left the energy source as a unit impulse, and remained so during transmission through the earth, then we could record a sequence of reflection coefficients, but this is not possible because of absorption near the energy source as well as along the path of transmission. The time duration of the wavelet must therefore be shortened *after* it has been recorded.

For $\alpha = 1$, a “spiking decon”, the deconvolved trace is shown in Figure 8.57d. The deconvolution has discarded everything except the first point in the wavelet and has therefore recovered the shape of the reflectivity function itself. Compare (d) with (c). The additional deconvolved traces shown in Figures 8.57e and f are

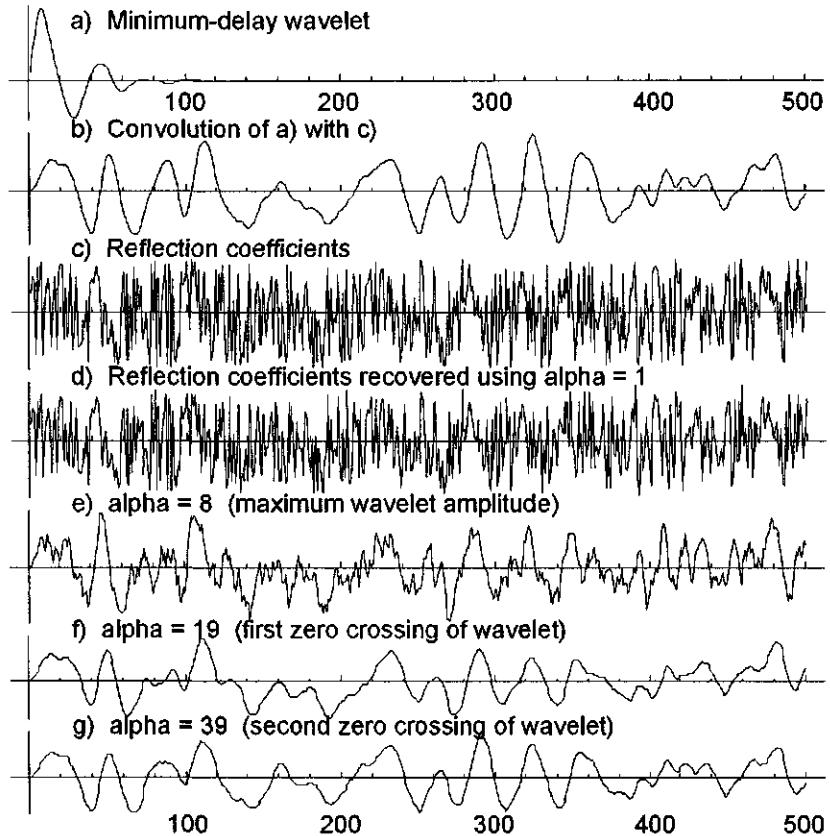


Figure 8.57: For this figure, the autocorrelation coefficients were determined from the wavelet, not from the autocorrelation of the seismic trace. (a) The same minimum-wavelet of Figure 8.53b and (c). (b) The seismic trace, i.e., the convolution of (a) with (c), the sequence of reflection coefficients. (d) The seismic trace of (b) after “spiking decon”, i.e., $\alpha = 1$. The reflectivity function is recovered (these are synthetic data, so this is possible). Actually, the recovered reflectivity function has been scaled by an unimportant constant and normalized for this plot. See Equation (8.69). (e) The seismic trace after choosing α to be equal to the first zero-crossing of the wavelet. (f) The seismic trace after choosing α to be equal to the second zero-crossing of the wavelet. SyntheticSeismogramForPD.nb → PredictiveDeconvolutionWavelets.cdr → SyntheticSeismogram1ForPD.wmf

deconvolutions used greater “gaps”, or values of α —in this case chosen to be equal to the first- and second zero-crossings of the wavelet shown in Figure 8.57a. The result of any of these deconvolutions is to shorten the length of the source wavelet by discarding wavelet energy via wavelet truncation.

Removal of trace energy after predictive deconvolution

The decrease in total trace energy after deconvolution is certainly one measure of the success of a deconvolution—energy has been removed. That is, the wavelet has been shortened. The energy change after deconvolution has been determined for various values of α and is shown in Figure 8.58 as the ratio of the total deconvolved trace (d_i^2) energy to the trace energy before (b_i^2) deconvolution using the definition

$$\text{Energy ratio} = \frac{\text{energy after deconvolution}}{\text{energy before deconvolution}} = \frac{\frac{1}{n} \sum_i^n d_i^2}{\frac{1}{n} \sum_i^n b_i^2}$$

From Figure 8.58 the energy of the deconvolved trace decreases rapidly as the value of α decreases. Thus, predictive deconvolution using shorter prediction distances may be viewed as a process that discards energy, which is to be expected in removing reverberations; however, in the absence of reverberations it will discard energy associated with the primary source wavelet itself.

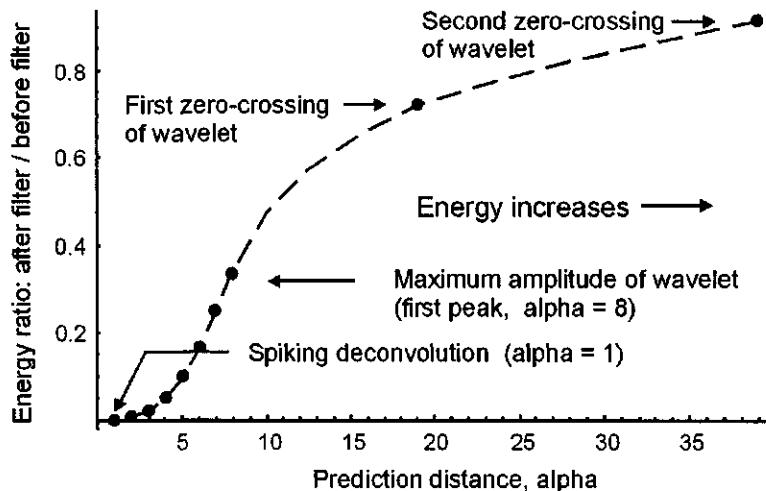


Figure 8.58: Vertical axis: ratio of total trace energy after deconvolution to that before. Horizontal axis: Prediction distance α . The wavelet referenced is shown in Figure 8.52. Removal of trace energy increases rapidly with decreasing values of α . SyntheticSeismogramForPD.nb → PredictiveDeconvolutionWavelets.cdr → EnergyVersusAlpha.wmf

Effect of Design Window on Deconvolution

To show the effect of the width of the window used to design the prediction operator (not the prediction error operator) we can generate a synthetic seismogram of, say, 500 points and then attempt to recover the reflection coefficients by using

design windows of different widths. This is an extreme test because we will be reshaping a minimum-delay wavelet to a unit impulse. Recall that any inverse filtering procedure used to change the shape of a non-reverberating primary source wavelet is conceptually, *but not computationally*, a two-step process:

1. Convert the entire wavelet to an impulse at its first value, and
2. Convolve the impulse with the desired output. This step does not involve any consideration of filter convergence, no matter what the delay properties of the desired output might be. It is always Step 1 that requires the filter of infinite length; the longer the least-squares filter, the closer the actual output will approach the desired output.

The effect of the width of the design window on predictive deconvolution is shown in Figure 8.59. The reflection coefficients (500) were generated using *Mathematica*'s random number generator and then convolved with the minimum-delay wavelet shown in the figure. The prediction filter with $\alpha = 1$ was designed from segments of the seismic trace starting at the beginning of the trace. Window widths used for filter design were 100, 200, 300, 400, and 500 points. As can be seen from the figure, and as anticipated from theory, recovery of the reflection coefficients improves with increasing window width (increasing length of the prediction filter as per Item 1 above).

Minimum number of points in the design window for a non-reverberation trace

The above results suggest that the minimum width (number of points) in the design window (minimum length of prediction filter) is approximately 300 points. Within the design window, recovery is excellent for the larger window widths showing that, to the extent that the reflectivity function is random,

$$\text{Autocorrelation of trace} = P \times \text{Autocorrelation of wavelet}$$

where P is a simple proportionality constant. Application of the prediction error operator to that portion of the trace outside of where the filter was designed gives less satisfactory results, as shown. As a reference standard for what the optimum deconvolution should be, a deconvolution using a prediction error filter designed from the autocorrelation coefficients of the wavelet itself are shown in Figure 8.59h. See the CD-ROM that accompanies this volume for the *Mathematica* program PDacfComparisons.nb used to generate Figure 8.59.

What are the observable effects of wavelet truncation?

“Spiking deconvolution” generally introduces high-frequency noise because of the attempt to shorten a primary source wavelet to a unit impulse. A different source of noise, however, is the high-frequency effect of wavelet truncation in cases for which the value of α is not chosen to coincide with one of the zero-crossings of the wavelet. This effect is easily seen if the trace-plotting software

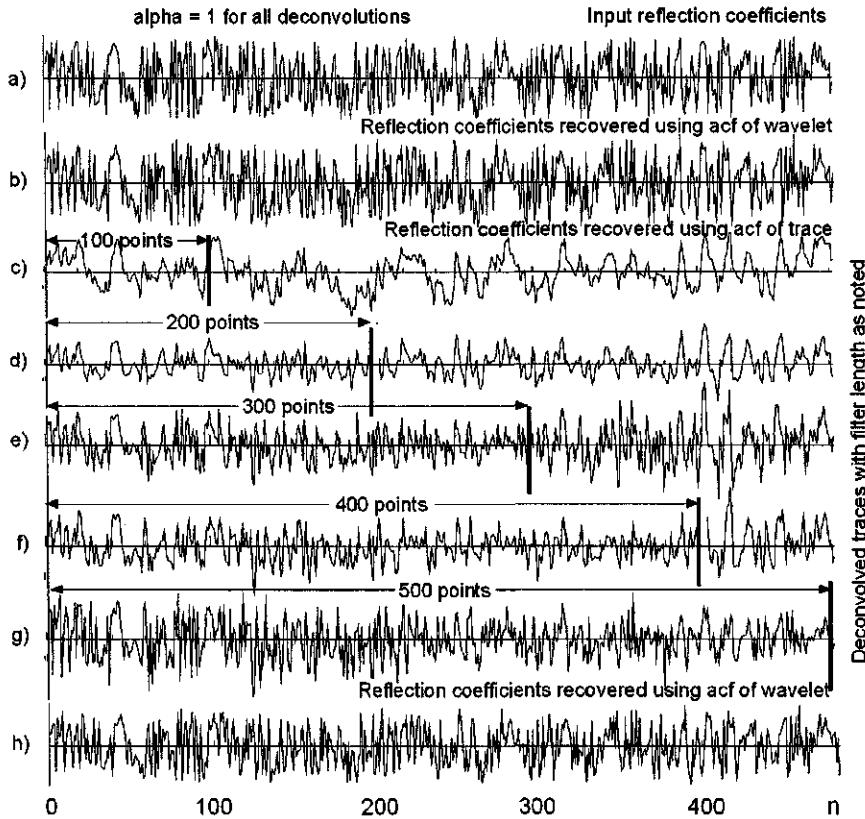


Figure 8.59: Top (a) Input reflection coefficients. (b) Input reflection coefficients recovered using autorrelation of wavelet. All deconvolutions for $\alpha = 1$. (c) through (g) Deconvolutions of trace using autocorrelation of trace with analysis window widths increasing from 100 to 500 points. (h) Same as (b). Input reflection coefficients recovered using autorrelation of wavelet. Successful recovery of the reflection coefficients improves with increasing width of the design window, and is not as good if the filter is designed in one window but applied outside of the design window, because the reflectivity function is not perfectly random. Figure generated using *Mathematica* program PDacfComparisons.nb on page 490. PDacfComparisons.nb → PDacfComparisons.cdr → PDacfComparisons.wmf

of commercial processing packages can be altered in such a way as to prevent the joining of successive ordinates of the sampled data, and instead plot the trace as a series of unconnected “spikes”, as shown in Figure 8.60c and d. Here, values of $\alpha = 10$ and 39 were chosen to represent severe and essentially no truncation, respectively. One can use this type of plot to

1. Provide an observable but qualitative measure of the undesirable high-frequency effects of wavelet truncation, in order to avoid the necessity for the application of a post-processing “cosmetic” low-pass filter and further distort what remains of the primary source wavelet after truncation, and
2. Minimize truncation effects by examining plots such as those shown in Figure 8.60c and d versus α . This is not possible by examining the trace energy subtracted by successive applications of predictive deconvolution using different values of α because, according to Figure 8.58 at least, the increase in trace energy appears to be a smooth monotonic function of increasing α .

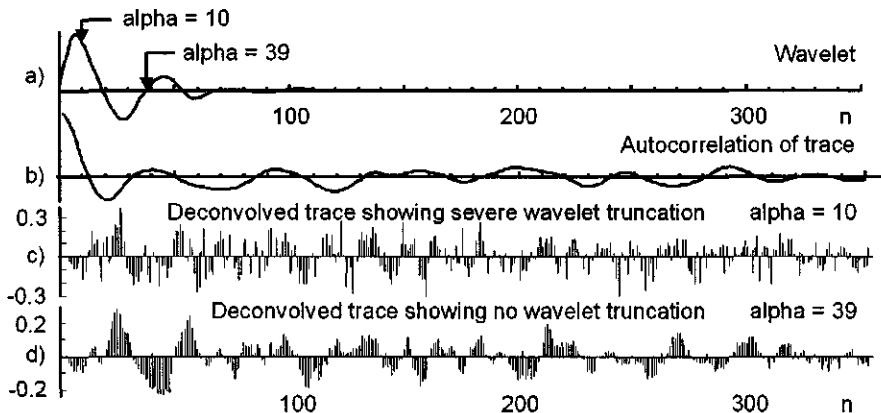


Figure 8.60: Top: (a) Minimum-delay wavelet. (b) Autocorrelation of seismic trace (trace before deconvolution not shown). (c) Deconvolved trace, differentiated, then plotted without joining ordinates. Differentiation emphasizes wavelet truncations. This method of plotting clearly shows noisier deconvolution caused by wavelet truncation for $\alpha = 10$. (d) Differentiated trace plotted without joining points and with $\alpha = 39$ (second zero crossing of wavelet). Higher frequency content evident in (c) because of terminations from wavelet truncation. Qualitative observations confirmed by higher numerical values of (c) as compared with (d). EdgeDetection.nb → EdgeDetection.cdr → EdgeDetection.wmf

See the CD-ROM that accompanies this volume for the *Mathematica* program EdgeDetection.nb used to generate Figure 8.60.

An alternative to spiking deconvolution?

The total energy removed from the seismic trace by predictive deconvolution as a function of α is shown in Figure 8.58. Trace energy that remains after deconvolution increases rapidly with increasing values of α . For a spiking decon ($\alpha = 1$), attempts to whiten the spectrum introduce noise because the

highest (as well as the lowest) frequencies were not even recorded in the field. Of course, this is not a problem with synthetic data. If a less demanding value of α is used, however, say $\alpha = 4$ points, then a reasonably good approximation to the reflectivity function (the geology) can be obtained. This is shown in Figure 8.61 where a good comparison between the deconvolved trace and a smoothed reflectivity function has been obtained by

1. Use of a gap of $\alpha = 4$ points, and
2. Application of a 3-point moving-average filter to the recorded seismic trace *before* deconvolution. Compare Figure 8.61d and e, the latter representing the reflection coefficients after a 3-point moving average filter is applied. Figure 8.61d is the seismic trace after predictive deconvolution using $\alpha = 4$.

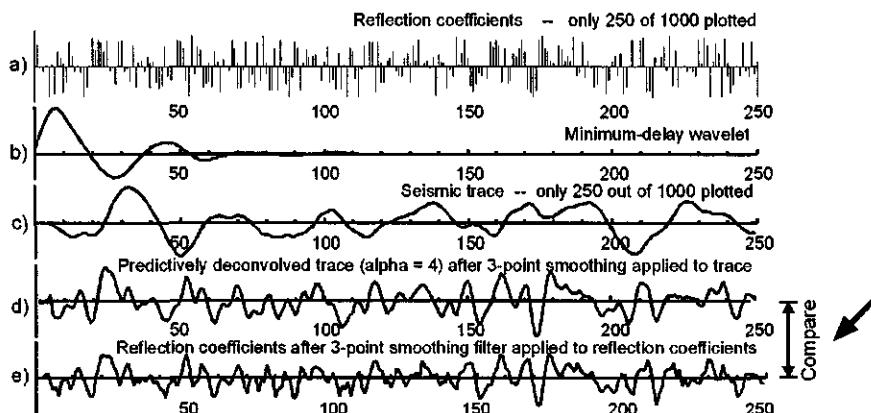


Figure 8.61: Top: a) Reflection coefficients—1000 used to generate synthetic trace, only 250 plotted. b) Minimum-delay wavelet. c) The seismic trace—convolution of a) and b). No smoothing. d) Deconvolved trace with $\alpha = 4$. Seismic trace has been smoothed before deconvolution with a 3-point moving average filter. e) The reflection coefficients shown in a) after a 3-point moving average filter has been applied to c). Compare d) and e). The deconvolution is a close approximation to the “geology”. The message of the figure is to suggest that a gap of $\alpha = 4$ points gives better results than either a spiking decon, which tends to result in a noisy trace, or a second zero-crossing decon, which results in less resolution. DoesTruncationMakeADifference.nb → DoesTruncationMakeADifference.cdr → AlphaEquals4.wmf

This is suggested as an alternative to spiking deconvolution where the objective is to shorten the length of the non-reverberating primary source wavelet. See the CD-ROM that accompanies this volume for the *Mathematica* program DoesTruncationMakeADifference.nb used to generate Figure 8.61.

Thin beds

A sequence of thin beds is shown in Figure 8.62 with bed thickness increasing horizontally (time thickness in sample points in italics) and prediction distance increasing vertically downward from $\alpha = 1$ to $\alpha = 8$, the latter corresponding to the maximum wavelet amplitude of the first half-cycle of the same wavelet shown in Figure 8.52. These results suggest that an optimum value of α is $\alpha = 4$ sample points because this prediction distance preserves the derivative waveform that is characteristic of reflection from a thin bed. Neither the first nor the second zero-crossing of the wavelet performs as well, and smaller values of α would promote a lower S/N ratio (see Figure 8.58) for the trace and do not produce a derivative waveform.

Effect of noise on predictive deconvolution

The effect of adding random noise to the seismic trace is shown in Figure 8.63. Again, the deconvolution filter was computed from the autocorrelation of the wavelet rather than from the trace. The usual approach taken for the addition of white noise to a trace before predictive deconvolution is to increase the value of the zero-lag autocorrelation coefficient r_0 by something from zero to about 5%. Here, we simply add random noise to the trace before autocorrelation and observe the increase in the value of r_0 . The percentage of noise added in these examples is that percentage of the standard deviation of the trace that results in an increase in the value of r_0 such that r_0 increases by about 5% for the noisy trace when α is set equal to the second zero-crossing of the wavelet. The results show that, for the same noise level but different values of α , random noise has less effect for greater prediction distances. Retention of just the first two wavelet points ($\alpha = 2$) in the presence of random noise (Figure 8.63) results in an unacceptable deconvolution. If the first eight wavelet points are retained then deconvolution of a noisy trace is acceptable, but only after smoothing the deconvolved trace with a moving-average filter, say 11 points (Figure 8.63). Recall that the deconvolved “wavelets” shown in the figure are fragments of *truncated* wavelets, a normal consequence of predictive deconvolution. No comparisons of noisy and noise-free traces for $\alpha > 8$ are shown because the deconvolutions of the noisy traces for values of α greater than 8 are excellent. The reader can run the *Mathematica* program *SyntheticSeismogramForPDWithFold* for other values of α , especially for first-and second zero-crossings of the wavelet.

All results shown in Figure 8.63 were obtained with a program that uses a translation of a convolution subroutine called FOLD from Robinson [148] to a *Mathematica* subroutine named *fold[a_,b_]*. See the CD-ROM that accompanies this volume for the *Mathematica* program *SyntheticSeismogramForPDWithFold.nb* used to generate Figure 8.63.

Reverberations

We reviewed a deterministic model for the removal of reverberations generated by a water layer (Section 8.2 on page 394). The reverberations were removed by

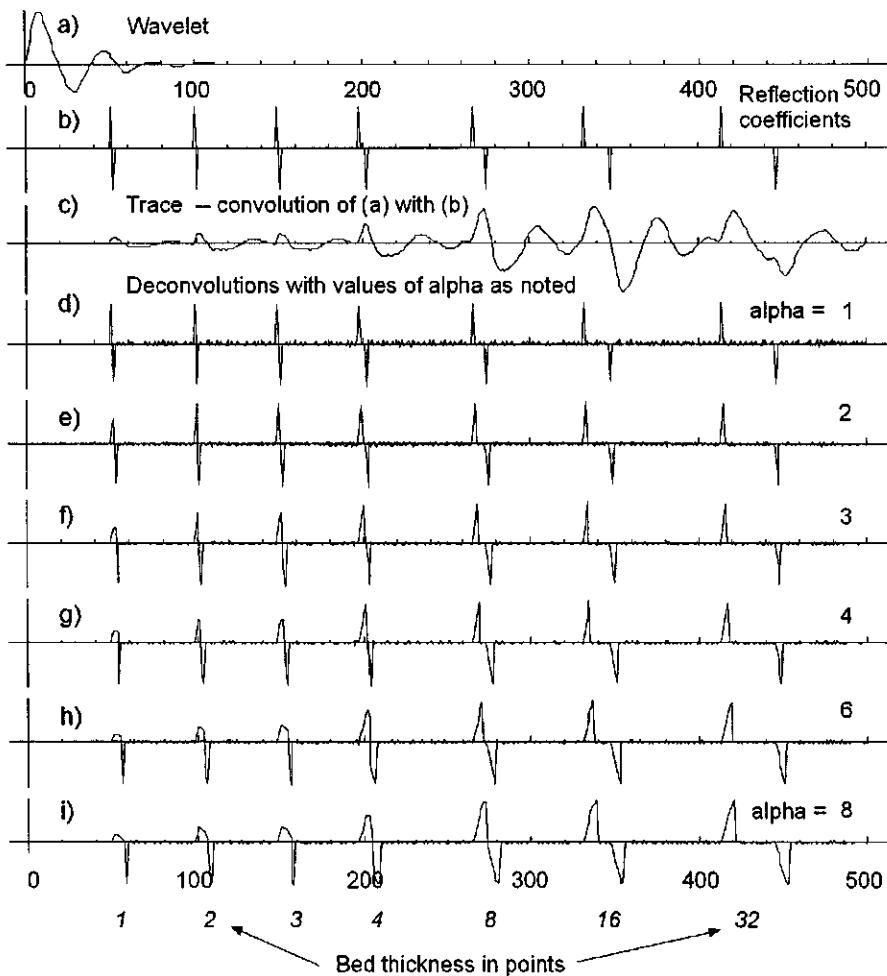


Figure 8.62: Top (a) minimum-delay wavelet. (b) the reflection coefficients. (c) seismic trace, the convolution of (a) with the reflection coefficients. (d) through (i) are synthetic seismograms showing the effect of various values of α (increasing downward) versus bed thickness in sample points (increasing thickness from left to right). The characteristic derivative waveform of a reflection from a thin bed is preserved for all values of $1 \leq \alpha \leq 8$ sample points and for thin-bed thicknesses ≥ 4 two-way time sample points. All deconvolutions for thin-bed thicknesses ≥ 4 and for any value of α are clearly an improvement over the recorded trace (c). SyntheticSeismogramForPDThinBed.nb,cdr
 \rightarrow SyntheticSeismogramForPDThinBed.wmf

simply knowing the water depth α and the water-bottom reflection coefficient, r_c . As usual, the wavelet reverberating in the water layer does not have to be minimum-delay. Possibly it is a Klauder wavelet from a vibrator source over a thick near-surface low-velocity layer, or it might be a water gun signature or an event from an explosive source. The only requirement was that the reverberations themselves be minimum-delay, which they are because the reflection coefficient at the bottom of the water layer is less than unity in magnitude. Each successive reverberation of the wavelet, therefore, produced a reflection of lower amplitude than the one before. Hence, the concentration of energy is located at the beginning of the reverberating wave train. Although the deterministic model described in Section 8.2 provides general insight into the problem of inverse filtering, the variations that we encounter in earth structure, earth layering, thickness of a water layer, and near-surface conditions make it impractical to estimate the water depth and water-bottom reflection coefficient. We need a more general approach to the removal of multiples. Predictive deconvolution is one of the most useful approaches.

Choosing the autocorrelation window for removal of reverberations by predictive deconvolution

We record a reflection arriving at some time t and want to shorten this event in order to increase seismic resolution. Whatever the source of the event, it started out as a seismic wavelet of some finite duration that left the source point and encountered layering of some sort within which reverberations were produced. This resulted in a longer source wavelet—an “effective source wavelet” so to speak. These effective source wavelets from the many subsurface reflecting interfaces overlap in time and degrade the seismic resolution. Our discussions so far have been concerned with the pros and cons of shortening the primary source wavelet itself by predictive deconvolution. Now we concern ourselves with removing scaled down “copies” (multiples) of the primary source wavelet, thus eliminating the reverberations.

Long-period multiples

Long-period multiples are defined as reverberations of the primary source wavelet that arrive sufficiently separated in time so that they do not overlap with each other. This is illustrated in Figure 8.64 for a “water-confined reverberation spike train”. These are the multiples generated in the “Backus water-layer filter” (Section 8.2), which was discussed earlier from a z -transform point of view.

In z -transform notation (Section 8.2), the equation for the “water-confined reverberation spike train” is

$$\frac{1}{(1 + r z^\alpha)}$$

where r is the water-bottom reflection coefficient and α is the time in sample intervals between successive scaled down “copies” (reverberations, arrivals,

multiples) of the primary source wavelet. Expansion of Equation (8.34b) using *Mathematica*'s function *Series* is shown in Figure 8.64c. Convolution of this with the primary source wavelet is shown in (d). The reader is encouraged to run the program **PDMultiples**, which was used to generate all of the examples in this section.

The assumption that the autocorrelation of the seismic trace is strictly proportional to the autocorrelation of the source wavelet is not correct unless the reflectivity function is perfectly random. Comparisons of deconvolutions using wavelet autocorrelations versus trace autocorrelations are shown in Figure 8.59. Of course, for synthetic data, perfect recovery can be expected if there are no random components in the deconvolution model. Using the autocorrelation coefficients computed directly from the reverberating source wavelet instead of from the trace satisfies this condition, as shown in Figure 8.64.

Excellent results were obtained earlier from predictive deconvolution by using the autocorrelation coefficients derived from the source wavelet; however, if the source "wavelet" is now considered to be a reverberating train of multiples of the primary source wavelet then we should expect comparable results if the coefficients are computed from the reverberation wave train, as shown in Figure 8.64c. There is no statistical component involved in this model even though predictive deconvolution is used.

As shown in Figures 8.64 and 8.65, the prediction error filter has successfully *truncated* the reverberating "effective source wavelet", leaving only the primary source wavelet.

By definition, long-period reverberations have individual multiple reflections separated from one other. Autocorrelation functions computed from such reverberation wave trains therefore lose no phase information associated with each reverberation event, and predictive deconvolution can effectively remove the reverberations regardless of the delay characteristics of the primary source wavelet. By definition, short-period reverberations interfere with each other and are recorded as a single long, "effective source wavelet" as shown in Figure 8.66 (top). Phase information about individual interfering reverberations is lost after autocorrelation. Predictive deconvolution can be applied to the reverberation to shorten its duration but the result will be a truncation of the short-period reverberation instead of recovery of the primary source wavelet. Recovery of the primary source wavelet is not possible, but the short-period reverberation can be effectively shortened, as shown in Figure 8.66.

It is instructive to look at the numerical values of the prediction error filter (Equation 8.80). We choose short-period multiples ($n = 20$), a prediction distance of $\alpha = 20$, and compare the exact numerical values of the inverse filter known from *z*-transform theory with those obtained from the least-squares approach of predictive deconvolution. The *z*-transform of the finite Backus 3-point inverse filter is $1 + rz^n + 2rz^{2n}$. These exact coefficient values can only be obtained from predictive deconvolution in the limit as the length of the reverberation deep-reflection spike train, and therefore its autocorrelation, becomes infinitely long. This is the mathematical symmetry between the inverse filter of infinite length required to spike out a wavelet of finite duration, and the filter of

finite length required to spike out the reverberating “wavelet” of infinite length. Values of filter coefficients for several reverberating “wavelet” lengths are shown in this figure. The longer the reverberation wave train, the closer the non-zero values returned by the system of simultaneous equations of predictive deconvolution will be to the exact values so easily determined from z -transform theory. Of course, the smaller the value of the water-bottom reflection coefficient, the shorter the filter needs to be because the autocorrelation function converges faster toward zero. The model results are shown in Figure 8.67.

The reader is encouraged to use the *Mathematica* program PDMultiples to investigate the convergence of the filter coefficients determined from predictive deconvolution toward the exact values known from z -transform theory for the water-confined reverberation spike train. Choosing a larger reflection coefficient will result in a slower decay of the spike train and therefore will require a longer seismic trace, which in turn will require a longer autocorrelation and filter length, *even though most of the filter coefficients are almost zero*. There is only one correct solution for the deconvolution filter regardless of how it is obtained. See the CD-ROM that accompanies this volume for the *Mathematica* program PDMultiplesDRRST used to generate Figure 8.65.

8.3.4 Summary guidelines for predictive deconvolution

1. On the basis of the theory underlying the design of the predictive deconvolution filter it is difficult to recommend a spiking predictive deconvolution. This choice of α discards most of the energy of the wavelet by wavelet truncation. Furthermore, the point that is not discarded is the first point in the wavelet(s) and therefore the one(s) just emerging from the noise with a poor S/N ratio with respect to the rest of the wavelet. A “spiking deconvolution” attempts to restore frequencies that were never recorded in the field in the first place. If a spiking deconvolution is desired one may compromise by choosing a “gap” (prediction distance) of a few points instead of just one, as discussed on page 491. This uses the first few points in the onset of the wavelet with a correspondingly better S/N ratio.
2. Predictive deconvolution is effective in removing long-period reverberations and recovering the primary source wavelet itself. The designation of “short” and “long” is a general one and refers to end members between which there exists a continuum of periods between multiples. Short-period multiples are simply overlapping multiples (see Section 8.2) that result in “ringing” on seismic traces; i.e., a lengthening of the primary source wavelet to what was called an “effective source wavelet” in Section 8.2. When the multiples become more widely separated in time they are referred to as “long-period”. It is often possible to observe the polarity changes between long-period successive multiples, as shown in Figures 8.64 and 8.65. The time, or index α is the period of time between the arrival of successive multiples. Both short and long-period reverberations are minimum-delay. Long-period reverberations can be successfully re-

moved by predictive deconvolution because they are separated in time. It makes no difference what the delay characteristics are of the wavelet leaving the source, because all we do with predictive deconvolution is to remove the reverberations, and this can be done very effectively, leaving the de-reverberated wavelet, which can be minimum-, mixed-, or maximum-delay. Predictive deconvolution can be used to shorten the wavelet further, *as long as the wavelet is minimum-delay*. If it is not minimum-delay, then filter coefficients existing for negative as well as positive time must be used. That is, a two-sided filter would be required with both an anticipation component and a memory component. For short period reverberations, predictive deconvolution is effective in shortening the duration of the reverberation wave train associated with short-period (small values of the prediction distance α) multiples but it cannot recover the primary source wavelet itself because phase information is lost during the autocorrelation of the overlapping multiples. If the filter is not made long enough then only the central part of the reverberation wave train will be suppressed as shown on Figure 8.68 after Robinson and Treitel [153, page 281].

3. The theoretically best results are obtained by computing autocorrelation coefficients that are derived from the entire seismic trace when dealing with water-layer reverberations. In order for the PD system of equations to arrive at the exact values as determined by a 2- or 3-point Backus filter, the entire reverberating wave train must be recorded. For smaller values of the water-bottom reflection coefficient, this constraint can be relaxed because the reverberating spike train converges to small values quickly; however, windowing the trace serves no useful purpose except to save on computer time and is therefore not recommended. If the trace is contaminated by, for example, a surface wave noise train, then a good dereverberation might not be possible.
4. Predictive deconvolution can be used to remove water-layer reverberations regardless of the delay characteristics of the primary source wavelet as long as the reverberations do not overlap significantly (i.e., they are long-period). If the reverberations overlap, as they do for short-period reverberations, then the primary source wavelet cannot be recovered.
5. In many of the examples shown in this section, the length of the prediction error filter for first-order ringing (a “water-confined-reflection reverberation spike train”) was taken to be equal to the length of the positive lags of the autocorrelation of the trace; however, after a length of $\alpha + n$, the filter coefficients derived from the complete autocorrelation of the reverberation trace are close to zero anyway, as predicted by the 3-point Backus filter. The length of the prediction error filter therefore just has to span one reverberation. That is, given that τ_1 is the two-way vertical traveltime through the water layer divided by the sample interval, then the inequality $\alpha + n \geq \tau_1$ must hold. It is only necessary that the *prediction* filter length of n points be such that the inequality $\alpha + n > \tau_1$ holds [153].

6. The length of the prediction error filter for second-order ringing ("deep-reflection reverberation spike train") can be shortened from the length of the positive lags of the autocorrelation filter to a length of just $\alpha + 2n$. The inequality $\alpha + n \geq 2\tau_1$ must hold. The length of the prediction error filter just has to span one period of the 2nd order reverberation to be removed—a single pair of reverberations of opposite polarity.

8.3.5 Predictive deconvolution—Conclusion

Although statistical predictive deconvolution is a common step in data processing, it should be noted that it is a mathematical procedure that is indifferent to the geologic model or the wavelet characteristics assumed by the interpreter. Part of the basis for a successful deconvolution depends on the assumed proportionality between the autocorrelation of the trace and the autocorrelation of the unknown wavelet. A common practice in attempting to shorten the length of an unseen wavelet is to choose a "gap" and then examine the autocorrelation of the trace to see if the autocorrelation function is attenuated after the gap. After the trace has been deconvolved, however, one can look at the autocorrelation of the deconvolved trace and it will always be attenuated over the interval defined by the length of the prediction filter (not over the interval defined by the length of the prediction error filter) because that's what the mathematics of the least-squares filter design is supposed to do. It does not signal a successful deconvolution or a shortened wavelet. It simply says that there are no bugs in the computer program that designed the least-squares prediction filter. The prediction error filter is *designed* to ignore the first part of the wavelet (by defining the "gap") and by attenuating its tail. It will even work if there is no wavelet at all because the mathematics is valid regardless of the statistical model the interpreter cares to assume. Of course, if the assumptions underlying the assumed model are correct, then predictive deconvolution works very well. It is hoped that the discussion in this volume further clarifies what predictive deconvolution does and what the penalty might be for requesting a "spiking decon" (usually followed by a cosmetic filter to remove the resulting noise).

Robinson [151, p. 293] in Volume 4A of this Handbook of Geophysical Exploration Series lists the following conditions that should hold for a successful application of predictive deconvolution:

1. The seismic wavelet is well-defined.
2. The seismic wavelet is a minimum delay time function (i.e., a stable causal time function with minimum-phase spectrum).
3. The additive noise is absent or small.
4. The spectrum of the reflectivity function (i.e., reflection coefficient series) is white.

5. The field trace is stationary, i.e., the wavelet does not change with time over a given time gate. A statistical test of the hypothesis that a time series is stationary has been discussed by Bryan [28].
6. The field trace is broad band.

In reality, continues Robinson [151], the following conditions probably exist:

1. The seismic wavelet is not well defined.
2. The wavelet is stable and causal, but is mixed phase.
3. The additive noise may be great.
4. The spectrum of the reflectivity is not white.
5. The field trace is time-variant over the given time gate.
6. The field trace is narrow band.

Schneider [155] noted

The work horse of statistical wavelet deconvolution for the past one and one half decades has been the predictive decomposition approach, which assumes the reflectivity function is statistically white and the convolutional wavelet is minimum-phase. To say that this has not been an effective tool is to condemn hundreds of thousands of miles of seismic processing and to deny untold millions of barrels of oil discovered from these data.

More recently [33, Cary, p. 20]

Despite isolated arguments in favor of deterministic approaches to deconvolution, statistical deconvolution still prevails. A consequence is that the phase of our data will continue to be somewhat uncertain without the ability to match it to synthetic seismograms derived from well logs.

The reader is referred to Volume 4A in this Handbook of Geophysical Exploration for a detailed treatment of seismic inversion and deconvolution [151].

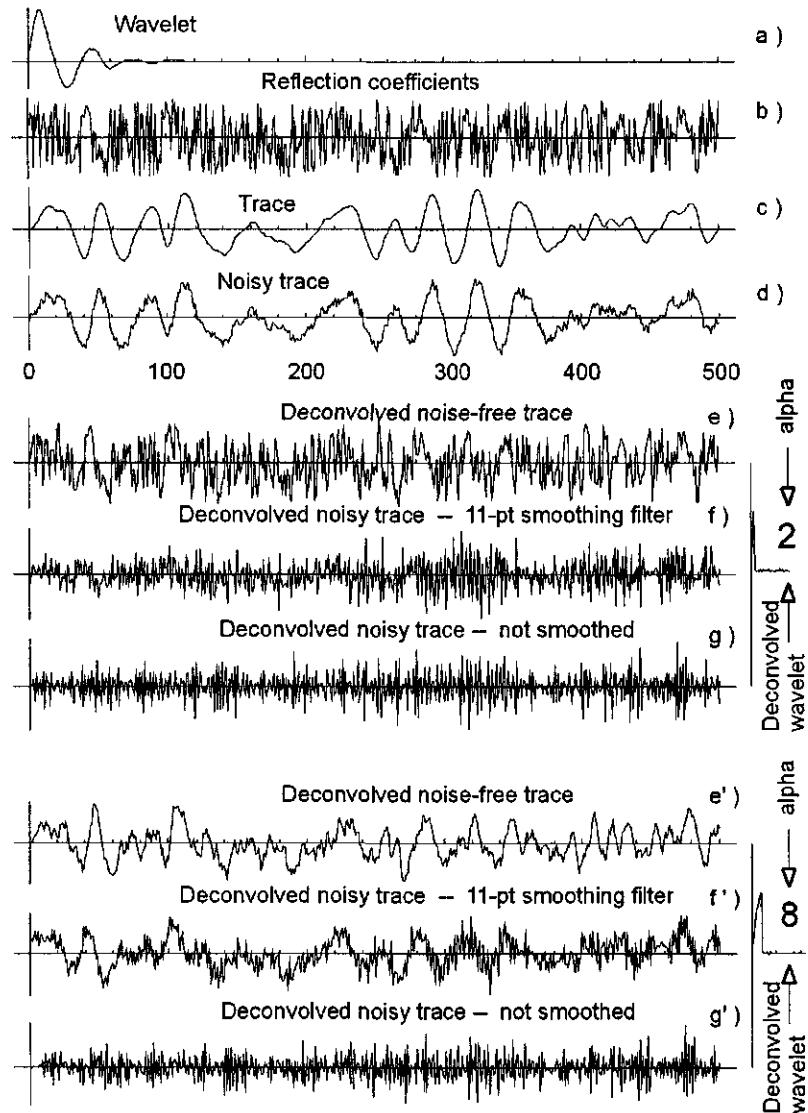


Figure 8.63: Top: (a) Minimum-delay wavelet. (b) reflection coefficients. (c) seismic trace—convolution of (a) with (b). (d) trace with additive random noise. (e) deconvolved noise-free trace $\alpha = 2$. (f) deconvolved noisy trace $\alpha = 2$. (g) deconvolved noisy trace, not smoothed. (e'), (f'), and (g') same except $\alpha = 8$. Note that deconvolved wavelets are truncated versions of the source wavelet shown in (a). SyntheticSeismogramForPDWithFold.nb → PDwithNoise.cdr → PDwithNoise.wmf

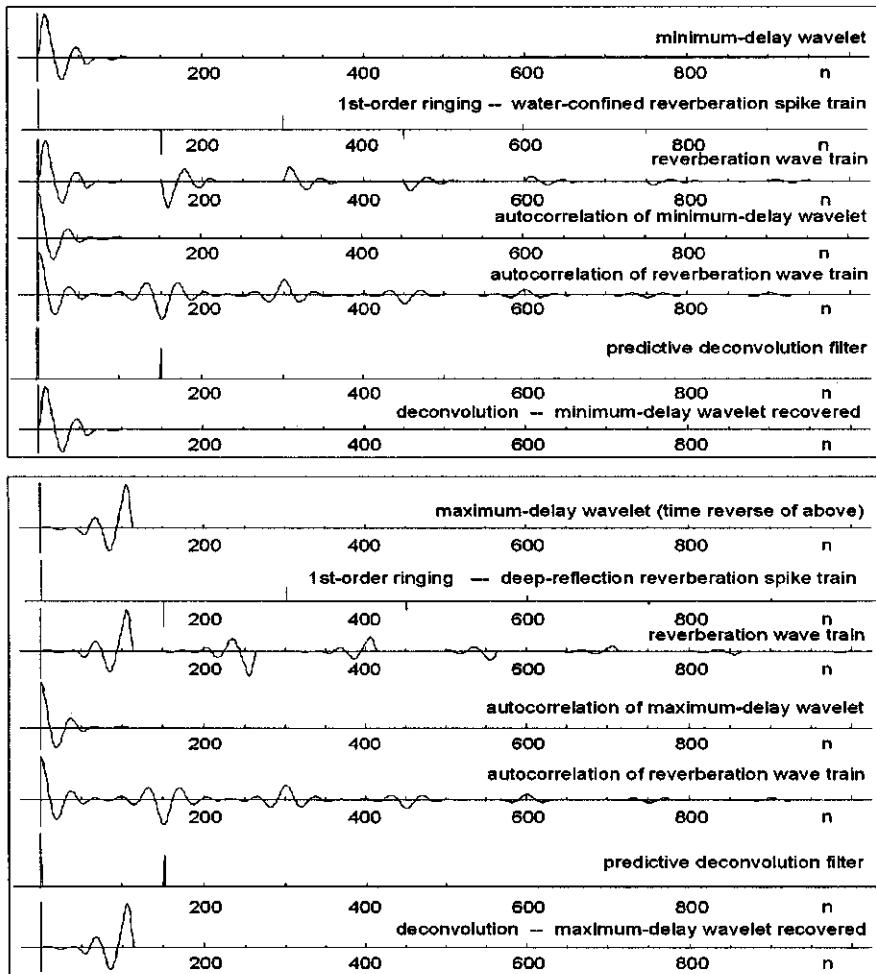


Figure 8.64: Removal of 1st order (water-confined reverberation spike train) long-period multiples by predictive deconvolution using (8.80). Top: minimum-delay wavelet. Autocorrelation of entire trace used to design filter. Same result if autocorrelation of wavelet were used. Bottom: maximum-delay wavelet (time reverse of wavelet used in top section). Same PD filter can be used in spite of the difference in the delay properties of the wavelets because primary wavelet is not shortened, only the length of the *minimum-delay* reverberation wave train. PDMultiplesWCRST.nb → PDMultiples.cdr → PDMultiplesWCRST.wmf

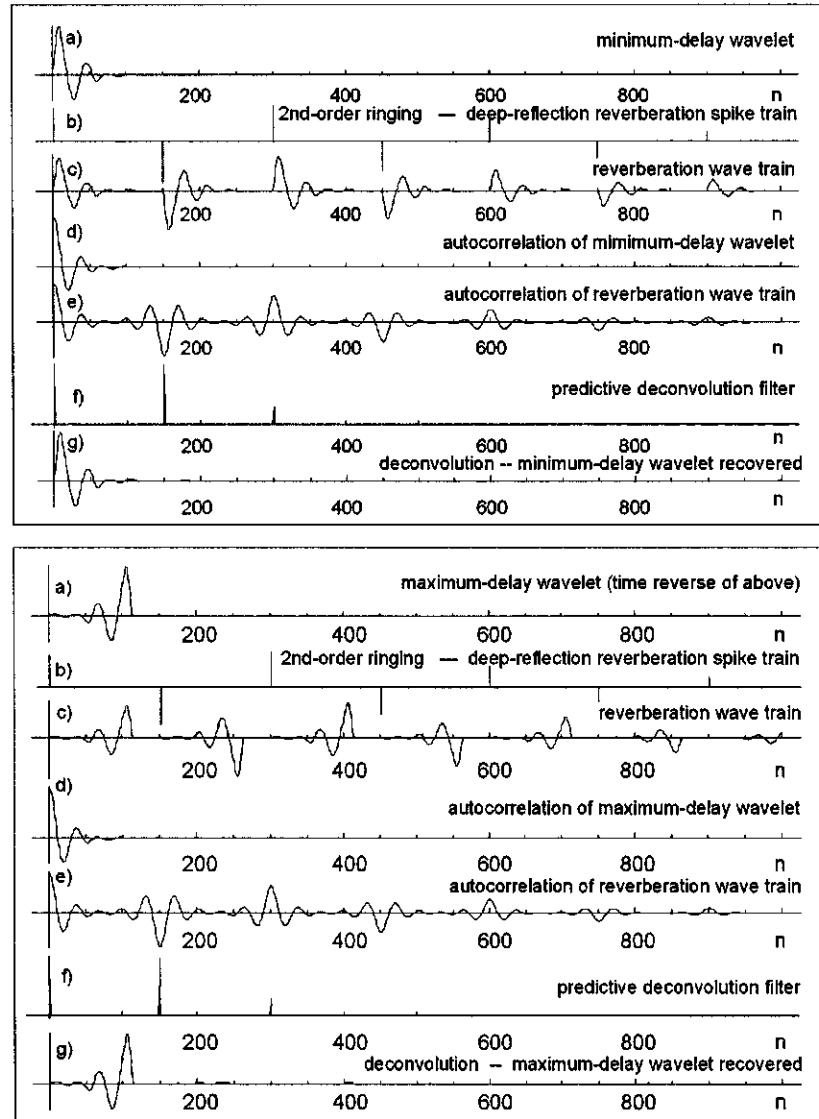


Figure 8.65: Removal of 2nd order (deep-reflection reverberation spike train) long-period multiples. Top: minimum-delay wavelet. Multiples removed by predictive deconvolution (PD) using (8.80). Autocorrelation of entire trace used to design filter. Same result if autocorrelation of wavelet used. Bottom: maximum-delay wavelet (time reverse of wavelet in top section). Same PD filter can be used in spite of difference in delay properties of wavelets because we do not shorten the primary wavelet itself, but just the *minimum-delay* reverberation wave train. PDMultiples.nb → PDMultiples.cdr → PDMultiplesDRRST.wmf

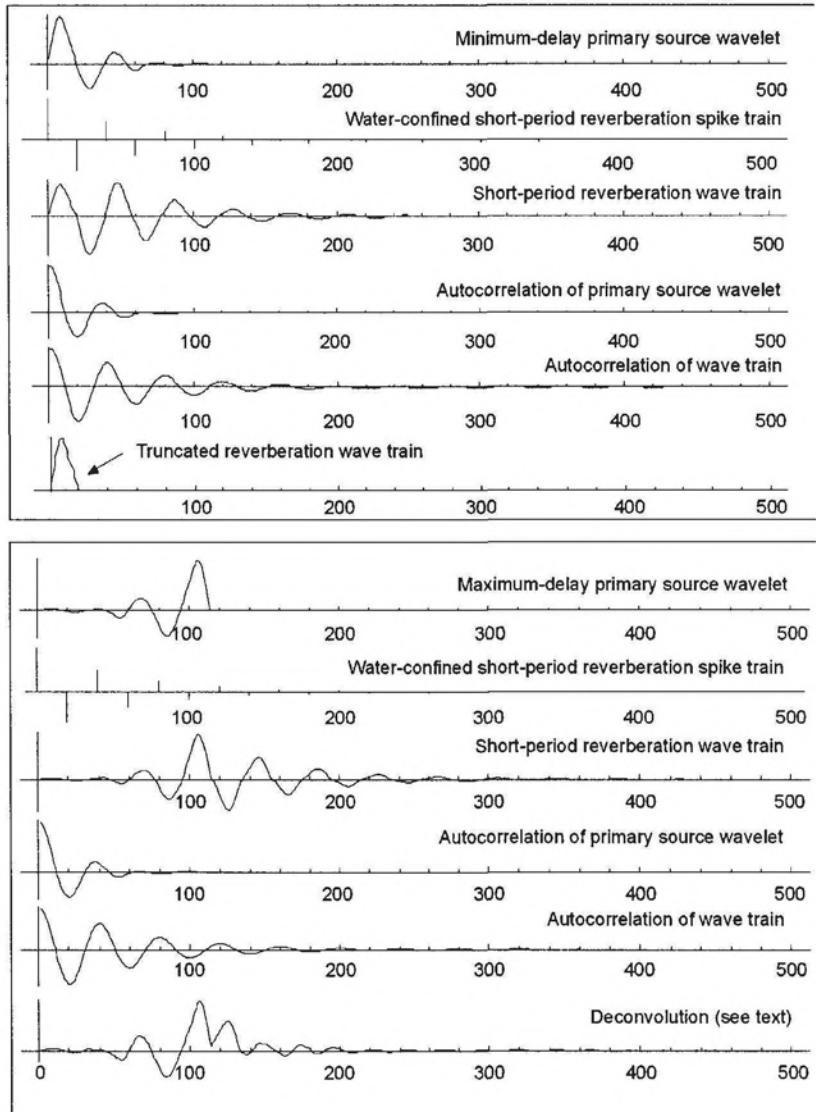


Figure 8.66: Removal of 1st order short-period multiples. Top box: minimum-delay wavelet. Minimum-delay reverberations successfully shortened by predictive deconvolution (PD) using Equation (8.80) but wavelet not recovered because reverberations overlap and phase information is lost. Autocorrelation of entire reverberation used to design filter. Bottom box: maximum-delay source wavelet (time reverse of wavelet used in top section). PD cannot recover the maximum-delay wavelet because reverberations overlap and filter coefficients for $t < 0$ are not used.
PDMultiplesWCRST.nb → PDMultiples.cdr → PDMultiplesShortPeriodWCRST.wmf

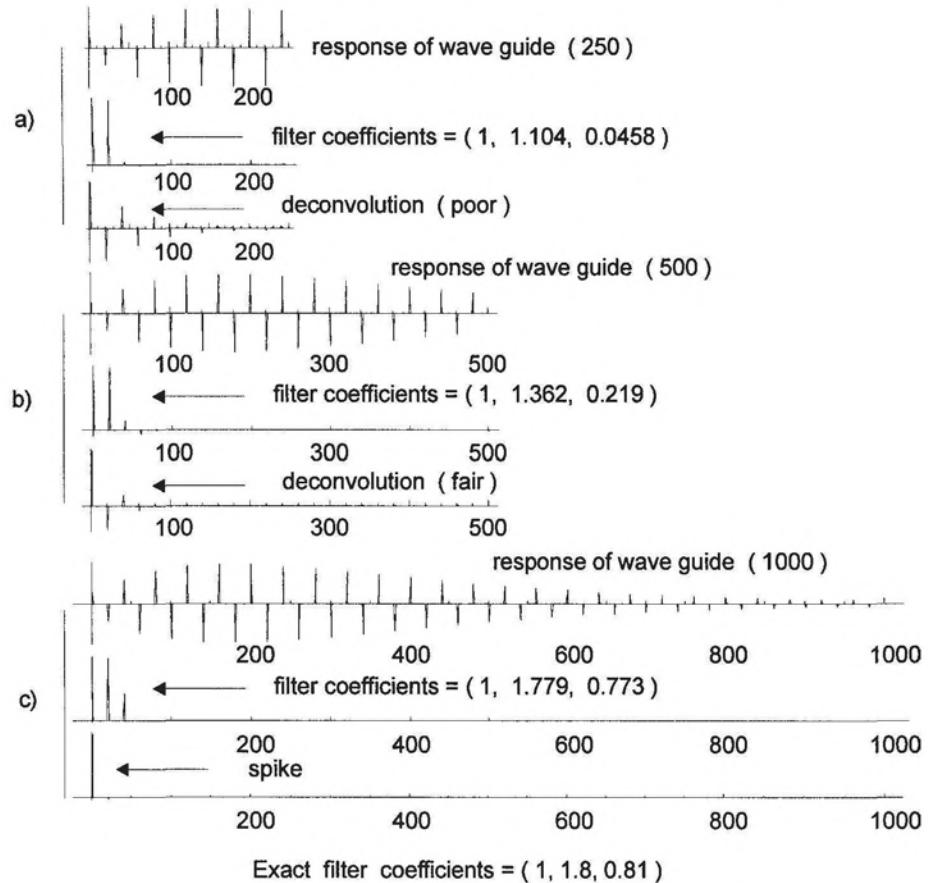


Figure 8.67: The z -transform of the finite Backus 3-point inverse filter is $1 + rz^n + 2rz^{2n}$, which for $r = 0.9$ is $(1, 1.8, 0.81)$. These exact coefficient values can be obtained from predictive deconvolution only in the limit as the length of the reverberating deep-reflection spike train—and therefore the autocorrelation of the spike train—becomes infinitely long, or at least long enough so the reverberation wave train is not severely truncated, as it is for 250 and 500 points. This is the mathematical symmetry between the inverse filter of infinite length required to spike out *any* wavelet of finite duration, and the filter of finite length required to spike out the reverberating “wavelet” of infinite length. Values of filter coefficients for three reverberation wave trains of 250, 500, and 1000 points shown. The longer the reverberation wave train, the closer the values returned by the system of simultaneous equations of predictive deconvolution will be to the exact values known from z -transform theory. For 250 points the deconvolution is poor. For 1000 points the deconvolution is essentially perfect. PDMultiples.nb → PDversusBackus.nb → PDversusBackus.cdr → PDversusBackus.wmf

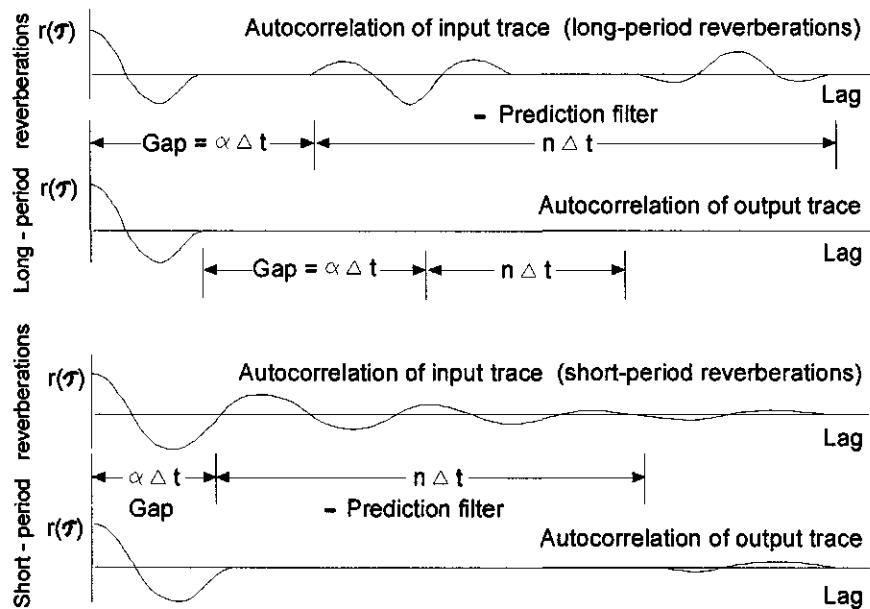


Figure 8.68: Top: Autocorrelation of long-period reverberations. Polarity clearly evident. Individual reverberations do not overlap. Predictive deconvolution (PD) can therefore remove reverberations no matter what the delay characteristics of the primary source wavelet might be. Bottom: Autocorrelation of short-period reverberations. Individual reverberations overlap. Autocorrelation of the trace discards phase information about individual overlapping events. PD can therefore not remove reverberations, but can truncate reverberating wave train after time $\alpha\Delta t$. If length $n \Delta t$ of prediction filter does not span autocorrelation of short-period reverberation, then the reverberation is not completely removed, as shown. Figures from Robinson and Treitel [153]. RandT12-5and3.cdr → RandT12-5and3.wmf

8.4 Spectral whitening

In addition to the conventional deconvolution methods, there are also a number of unconventional spectral whitening procedures. The basic principles of stretched automatic amplitude adjustment [36] and automatic amplitude adjustment before vibroseis correlation [37] are discussed here as a means of spectral whitening of vibroseis, dynamite, or marine seismic data. The main aspect of these methods is to separate the spectral components of the data in time by stretching the data and then to balance the amplitude of the separated spectral components (spectral whitening) before the data are compressed back to their original form. These methods spectrally whiten data in a time-variant fashion and are called stretched automatic amplitude adjustment or stretched automatic gain control (SAGC).

Stretched automatic amplitude adjustment may be viewed as a three-step process:

1. Separation of amplitudes of frequency components of the wavelet in the time domain by convolution with a stretching signal,
2. Adjustment of the amplitudes of the stretched data by application of an amplitude balancing procedure such as automatic gain control and,
3. Compression of the data back to the original length by crosscorrelation with the stretching signal.

The stretching signal $s(t)$ is similar to the vibroseis sweep signal [53, 187] where the frequency is time variant:

$$s(t) = a(t) \sin \left[2\pi \left(f_s + \frac{f_e - f_s}{T_s} \right) t \right] \quad (8.90)$$

In this equation $a(t)$ is the time varying amplitude function, f_s and f_e are the starting and ending frequencies of the stretching signal, and T_s is its length. If $x(t)$ is the seismic trace then the stretched trace $y(t)$ is

$$y(t) = \int_{-\infty}^{\infty} x(t - \tau) s(\tau) d\tau = x(t) * s(t) \quad (8.91)$$

where $*$ denotes convolution.

If the length of the data is T_x then the length of the stretched trace is $T_y = T_x + T_s$. When the length of the stretching signal is chosen large enough with respect to the length of the wavelet, $y(t)$ will be in a form such that the amplitudes of the different frequency components are separated in time. After this separation, differences in spectral amplitudes in time become apparent and the amplitudes of different frequencies can then be adjusted.

After stretching, time-domain automatic amplitude adjustment is used to balance the amplitudes of the different frequency components. The balancing is given by

$$y'(t) = y(t) G(t) \quad (8.92)$$

$G(t)$ is the time-variant automatic amplitude adjustment function defined as

$$G(t) = \frac{L}{\Sigma|y(\tau)|} \quad (8.93)$$

where L is the length of the window used to compute the amplitude factor at the center of the window at time t .

After amplitude adjustment, the stretched trace $y'(t)$ is compressed by cross-correlation using the same stretching signal

$$z(t) = y'(t) * s^*(t) \quad (8.94)$$

where $s^*(t)$ is the complex conjugate of the stretching signal and $z(t)$ is the compressed data. The length of the compressed trace T_z after crosscorrelation is $T_z = T_x = T_y - T_s$.

The combined effect of stretching as a separation filter and compression may be easily shown when there is no intermediate amplitude adjustment. The compressed trace is

$$z(t) = y(t) * s^*(t),$$

or

$$z(t) = x(t) * \phi_{ss}(t) \quad (8.95)$$

where $\phi_{ss}(t)$ is the autocorrelation of the stretching signal.

From Equation (8.95) it is clear that the effect of spectral separation and compression in time is equivalent to a filtering process where the filtering operator is the autocorrelation of the stretching signal, which makes critical the frequency bandwidth of the stretching signal.

Model data composed of three Ricker wavelets are used to examine the SAGC processing scheme where the data and its amplitude spectrum are shown in Figures 8.69a and d. A tapered stretching signal of 5 s with a frequency bandwidth of 10-100 Hz was used for spectral separation. The stretched model data and its amplitude spectrum are shown in Figures 8.69b and d. There is no loss of spectral content because of the wider frequency bandwidth of the stretching signal. The wavelets are no longer recognizable in time (Figure 8.69b) and the data show increasing frequency with time because of spectral separation. In order to emphasize this separation, the segments of data (Figure 8.69b) from 0.5 to 1 s and from 4.5 to 5 s, and their amplitude spectra are shown in Figures 8.70a and b and Figures 8.70c and d, respectively, where the amplitude spectra indicate that each time window has a different frequency band separated by the stretching. Therefore the changes of amplitude with time after stretching (Figures 8.69b and 8.70) are the changes in amplitude of the spectral components, and any amplitude adjustment in time on the stretched data will result in spectral whitening. It is obvious that the spectral separation will be greater if the ratio of the length of stretching signal to the length of the trace is increased.

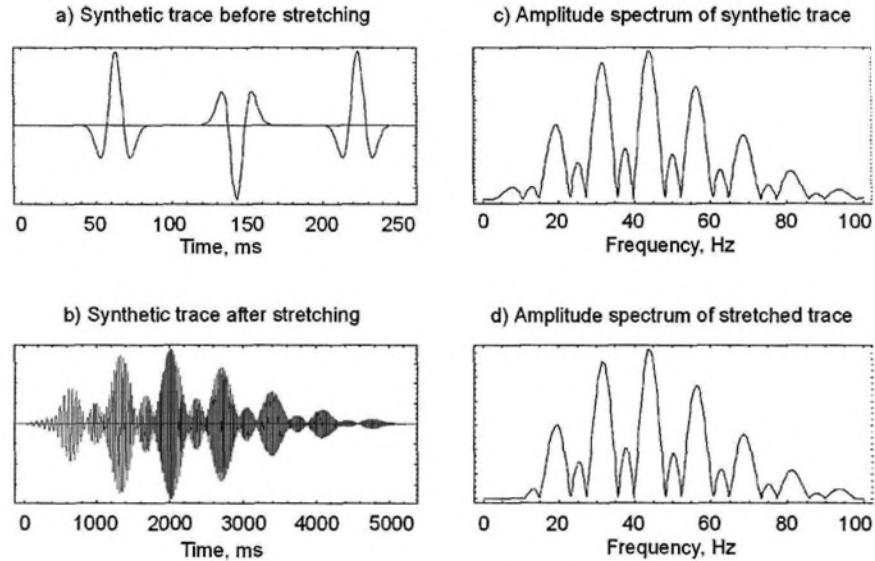


Figure 8.69: Model data consisting of three Ricker wavelets and their amplitude spectra (a) before and (b) after spectral separation by stretching.
SAGCFig1.nb \Rightarrow StretchAGC.cdr \Rightarrow sagc1.wmf

Stretched automatic gain control

To show the combined effect of stretching and compression a 10-100 Hz linear stretching signal of length 20 s with a tapering of 0.25 s is used to stretch and compress the model data without and with intermediate amplitude adjustment in the time-domain. The results and corresponding amplitude spectra are shown in Figure 8.71 where the wavelets are recovered with their relative amplitudes. No side effects in terms of oscillations due to correlation are apparent when there is no amplitude balancing. After SAGC, however, the data exhibit side-lobe effects because of the intermediate amplitude balancing in the time-domain. These oscillations are low-amplitude side-lobes as a result of the correlation that compresses the stretched data. The side lobes will not be visible in real data; they would be in the background because of their low amplitudes.

An example with real vibroseis data acquired by Virginia Tech on the Atlantic Coastal Plain in Virginia is shown in Figure 8.72. The data are 1.0 s long with a 2 ms sample interval and were subjected to SAGC processing with a 10 s long stretching signal with a 250 ms taper. The linear sweep was generated with a frequency band of 20-100 Hz. SAGC was applied with a 100 ms long automatic amplitude balancing (AGC) window.

The data before SAGC (Figure 8.72a) are contaminated by a noise wave train that dominates the seismogram with a dominant frequency band from 25 to 35 Hz (Figure 8.72b), even though the original vibroseis recording was carried

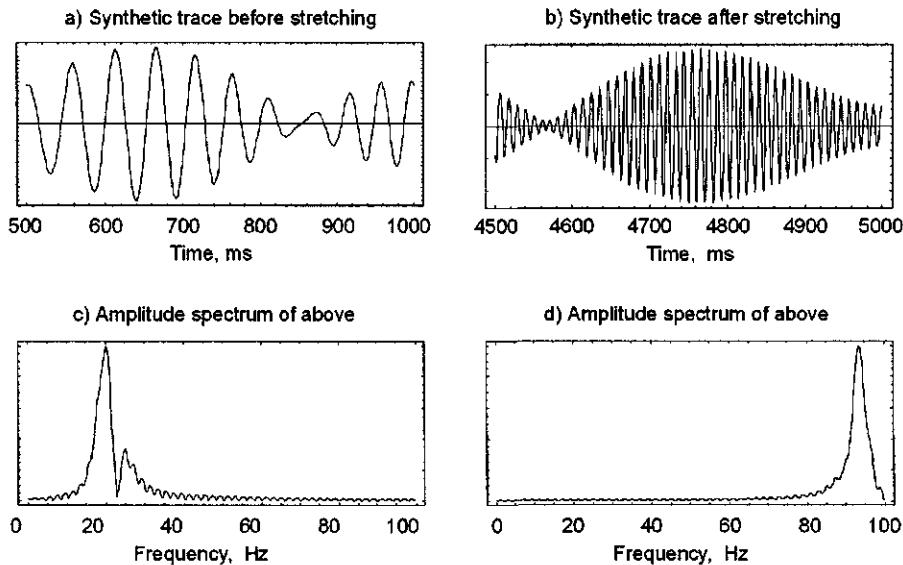


Figure 8.70: Portions of model data after spectral separation by stretching in time and their normalized amplitude spectra. Stretched data from the windows of (a) 0.5-1 s and (b) 4.5-5 s. Amplitude spectrum of (a) and (b) are shown in (c) and (d), respectively. The spectra indicate dominant frequencies of about 20 and 94 Hz for respective windows because of the spectral separation in time that results from the stretching process. SAGCFig2.nb \Rightarrow StretchAGC.cdr \Rightarrow sagc2.wmf

out with a sweep of 20-100 Hz. Because of the dominant noise, the amplitudes of higher frequency components are considerably less and no distinct reflection event can be identified. These data are considered to be a good example for the application of SAGC. To emphasize the effect of SAGC processing, a conventional AGC with a window length of 0.1 s was applied after stretching for comparison. The results are shown in Figures 8.72c and 8.72d, respectively. The data of Figure 8.72c exhibit a higher frequency response with distinct reflection events as a result of spectral whitening. The effect of the spectral whitening is obvious in the amplitude spectra (d) where the spectrum after SAGC shows a whitened spectrum between frequencies 20 and 100 Hz in spite of the domination of frequencies between 25 and 35 Hz in the data before SAGC.

The SAGC procedure can be easily incorporated with vibroseis correlation without the stretching step because vibroseis data before correlation is already in a stretched form. A shot gather that was originally correlated without amplitude manipulation is shown in Figure 8.73a. This data set was recorded on the Atlantic Coastal Plain in Virginia with a receiver interval of 5 m. No array was used to attenuate surface waves that are dominant and mask reflections,

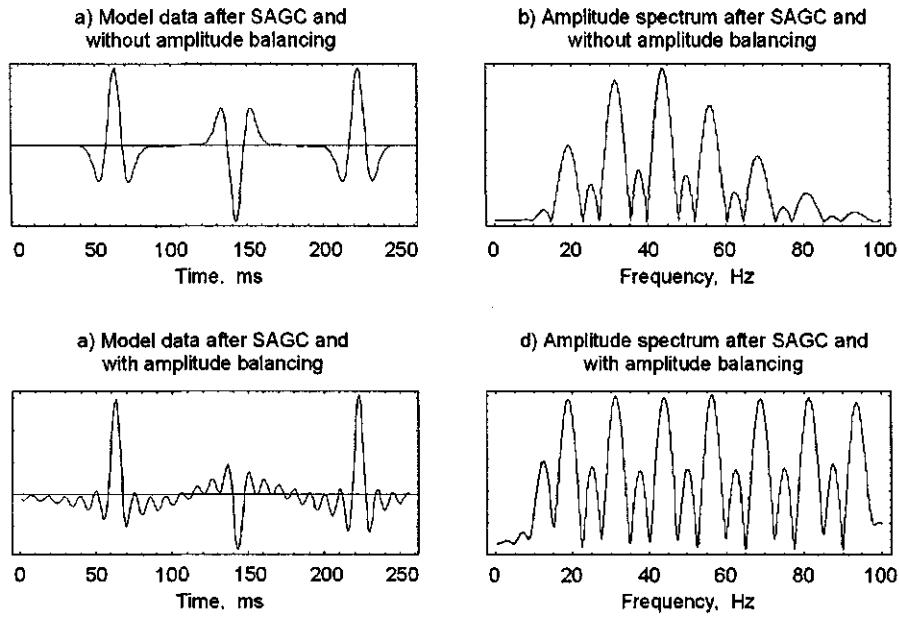


Figure 8.71: The model data after SAGC and with and without amplitude balancing in time-domain are shown in (a) and (c), respectively. The corresponding spectra are shown in (b) and (d), respectively. When there is no amplitude adjustment between stretching and compression there is no noticeable difference either in data or spectra before and after the stretching and compression process. There are, however, correlation side-lobes when an amplitude adjustment is applied. These oscillations will be in the background and will therefore not be that visible in real data. SAGCFig3.nb
⇒ StretchAGC.cdr ⇒ sage3.wmf

including the one from basement labeled "B" in Figure 8.73 at 0.21-0.25 s between traces 20 and 48, and one at 0.2 s between traces 15 and 25. On the near traces no reflection is recognizable because of the masking effect of the large amplitude surface waves.

SAGC using a 5-100 Hz stretching signal of length 8 s with a 1 s taper was applied to the shot gather data shown in Figure 8.73a. The result is shown in Figure 8.73b where the increased continuity of reflections (the event "B" and shallower events) and resolution in time are evident and confirm the effectiveness of the method. The window length for the automatic amplitude adjustment between the stretching and compression was 0.1 s. In spite of the residual noise in Figure 8.73b, an appreciable improvement in the signal-to-noise ratio is evident; compare Figure 8.73a and b. As a specific example, the data shown in Figure 8.72 is channel 35 from this shot gather. Trace 35 and its amplitude spectrum after application of SAGC in Figure 8.72c and 8.72d show a whitened

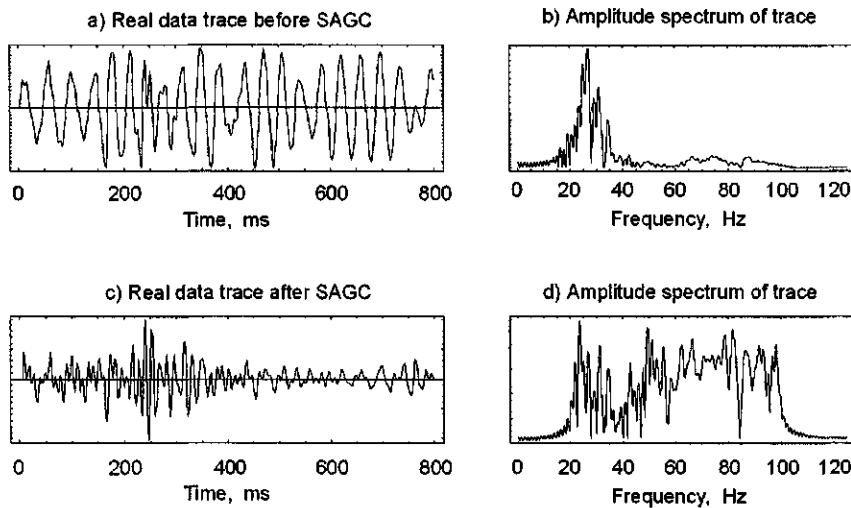


Figure 8.72: Real data (a and c) and amplitude spectra (b, d) before and after SAGC, respectively. Low frequency (25-35 Hz) dominant spectra (b) is enhanced by SAGC and shows a balanced spectrum (d) in the frequency band (20-100 Hz). SAGCFig4.nb \Rightarrow StretchAGC.cdr \Rightarrow sags4.wmf

spectrum that results in increased resolution in time. After SAGC processing, the largest amplitude in time (Figure 8.73) is associated with the reflection from basement.

Discussion

The length of the stretching signal, its frequency content, and the AGC window length are important parameters in this spectral whitening process. A stretching signal with a frequency band that is narrower than the frequency band of the data will function as a band-reject filter because of the convolution and correlation operations inherent in SAGC processing. The longer the stretching signal and the shorter the length of the automatic amplitude adjustment window, the better the expected whitening; however, because of the computation time involved, some experimentation is required for adequate whitening.

The SAGC spectral whitening scheme can be summarized as follows:

- A frequency-modulated signal is used to stretch the data and thereby separate spectral components by convolution and then to compress it by cross-correlation,
- A conventional automatic gain control is used after stretching to balance separated spectral amplitudes in time resulting in spectral whitening,

- Energy balancing instead of amplitude could be used in the balancing process. The amplitude of the stretching signal can be made time-variant to allow frequencies in specified windows to be enhanced with respect to others.
- The method is simple but requires long computation times because of the amplitude adjustment in the time domain of the stretched data, and because of the intermediate steps of stretching and compression.

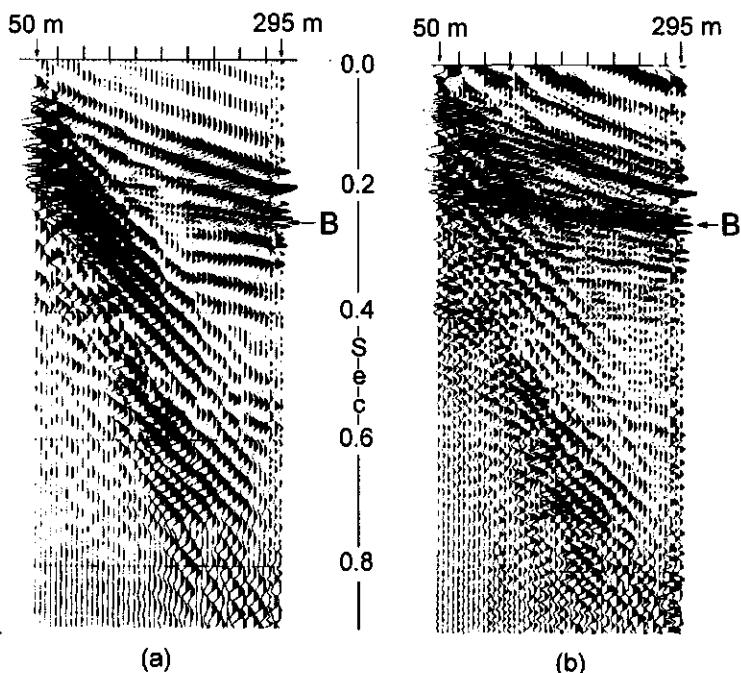


Figure 8.73: (a) Virginia Tech vibroseis data from the Atlantic Coastal Plain after conventional vibroseis correlation. (b) After application of SAGC using an automatic amplitude adjustment (AGC) in the time domain. Data stretched using an 8 s stretching signal with a frequency bandwidth of 5-100 Hz and a 1 s taper. AGC then applied to the stretched signal in the time domain followed by compression by crosscorrelation . The same result could have been accomplished by inserting an automatic amplitude adjustment (AGC) step before vibroseis correlation. PhotoShop
⇒ sagc5.bmp

8.5 Further Applications of Hilbert Transforms

The basic theory of Hilbert transforms was discussed in Section 3.5.11. In the previous chapters emphasis was placed on assumptions about the delay characteristics of the seismic wavelet. We saw that although we cannot determine directly both the amplitude and phase of the unseen seismic wavelet, we could determine its amplitude spectrum. This would be possible under the Robinson assumptions that 1) the autocorrelation of the seismic trace is directly proportional to the autocorrelation of the unknown seismic wavelet, and 2) the wavelet is minimum delay.

8.5.1 Relationship between the amplitude and phase spectrum of a causal function

The complex amplitude spectrum $A(\omega)$ of a wavelet (or of a unit-impulse response) can be written

$$A(\omega) = |A(\omega)| e^{i\phi(\omega)} \quad (8.96)$$

where $|A(\omega)|$ is the modulus of the spectrum, $A(\omega)$ is the complex spectrum, and $\phi(\omega)$ is the phase angle spectrum. We assume that somehow we know the amplitude spectrum but not the phase spectrum. Without both we cannot see the shape of the wavelet.

In theory at least an estimate of the wavelet amplitude spectrum $|A(\omega)|$ can be obtained by taking the square root of the power spectrum which we know to be the Fourier transform of the autocorrelation of the wavelet (or the impulse response). The autocorrelation $\phi_{11}(\tau)$ of $h(t)$ is

$$\phi_{11}(\tau) = \int_{-\infty}^{\infty} h(t) h(t + \tau) dt \quad (8.97)$$

and the Fourier transform of $\phi_{11}(\tau)$ is the power spectrum $\Phi(\omega)$. Thus,

$$\Phi(\omega) = \int_{-\infty}^{\infty} \phi_{11}(\tau) e^{-i\omega\tau} d\tau = |A(\omega)|^2 \quad (8.98)$$

and (8.117) becomes

$$A(\omega) = \sqrt{\Phi(\omega)} e^{i\phi(\omega)} \quad (8.99)$$

Now taking the log of both sides of (8.99) we get

$$\begin{aligned} \log A(\omega) &= \log \sqrt{\Phi(\omega)} + i\phi(\omega) \\ &= \frac{1}{2} \log \Phi(\omega) + i\phi(\omega) \\ &= H_e(\omega) + iH_o(\omega) \end{aligned}$$

where $H_e(\omega) = 1/2 \log \Phi(\omega)$ is even and $H_o(\omega) = \phi(\omega)$ is odd. We can now obtain the phase spectrum of the wavelet (or of an impulse response) by taking the Hilbert transform of $1/2 \log P(\omega) = H_e(\omega)$. That is,

$$\begin{aligned}\log A(\omega) &= \frac{1}{2} \log P(\omega) + i \mathcal{H} \left[\frac{1}{2} \log P(\omega) \right] \\ &= H_e + i H_o\end{aligned}$$

where \mathcal{H} denotes Hilbert transformation, and then

$$A(\omega) = e^{H_e + i H_o} \quad (8.100)$$

Having now both the amplitude and phase spectrum we can take the inverse Fourier transform of $A(\omega)$ to obtain the causal function $h(t)$:

$$\begin{aligned}h(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} A(\omega) e^{+i\omega t} d\omega \\ h(t) &\iff H(\omega)\end{aligned}$$

The mathematical relationship (the Hilbert transform) between the real and imaginary parts of the Fourier spectrum guarantees that $h(t)$ will be causal; it does not guarantee that $h(t)$ will be minimum- maximum- or mixed-delay. More on this below.

A *Mathematica* program to recover the (minimum-delay) wavelet from its power spectrum is shown below.

Minimum-, mixed-, or maximum-delay wavelets can have the same autocorrelation. This has important implications for predictive deconvolution. The simplest example is to examine results for the two 2-point wavelets $(2, 1)$ and $(1, 2)$. The former is minimum-delay; the latter maximum-delay. Autocorrelation of either 2-point wavelet, however, gives the same result.

$$\phi(\tau) = (2, 5, 2) \text{ for } \tau = -1, 0, +1$$

Using Hilbert transforms to recover the wavelet from the autocorrelation will therefore give the same result as shown by the following *Mathematica* examples. In each case h is the input wavelet. Several print statements have been added for clarification and can be safely deleted.

```
h={2,1}; (* Pad with a few trailing zeros for better results *)
Print["Input wavelet is ",h];
acf=DiscreteConvolution[h,Reverse[h]];
Print["Autocorrelation of wavelet is",acf];
acf=RotateLeft[acf,Floor[Length[acf]/2]];
n=Length[acf];
PowerSpectrumOfh=Chop[Fourier[acf]Sqrt[n]];
AmplitudeSpectrumOfh=Sqrt[PowerSpectrumOfh];
He=0.5*Log[PowerSpectrumOfh];
He=RotateLeft[He,Ceiling[Length[He]/2]];
hev=Table[0,{n}];dt=0.004; period=n*dt;dfreq=1/period;
Do[c=0;Do[c=c+He[[f]]*Cos[2 Pi*
(f-Ceiling[n/2])*dfreq*(t-Ceiling[n/2])*dt],
{f,1,n }],
```

```

he[[t]] = c, {t, i, n}]; Ho=Table[0,{n}];
Do[s=0;Do[ s=s+he[[t]]*
Sin[2 Pi (f-Ceiling[n/2])*dfreq*Abs[t-Ceiling[n/2]]*dt],
{t,Ceiling[n/2],n}];
Ho[[f]]=-2s/n ,{f,Ceiling[n/2],n}];
HoPos=Take[Ho,-Floor[Length[Ho]/2]];
HoNeg=-Reverse[HoPos];Ho=Join[HoNeg,{0},HoPos];
Ho=RotateLeft[Ho,Floor[Length[Ho]/2]];
He=RotateLeft[He,Floor[Length[He]/2]];
ComplexSpectrumOfh=
AmplitudeSpectrumOfh*Cos[Ho]-I*
AmplitudeSpectrumOfh*Sin[Ho];
Print["Wavelet recovered from Hilbert transform:",
NumberForm[
Chop[InverseFourier[ComplexSpectrumOfh]]/Sqrt[n],4]];
----- Output -----
Input wavelet is {2,1}
Autocorrelation of wavelet is {2,5,2}
Wavelet recovered from Hilbert transform: {2,1}
-----
```

Using the maximum-delay 2-point wavelet (1, 2) the autocorrelation is the same but the wavelet recovered by Hilbert transforms is again the minimum-delay wavelet (2, 1). This must be so because the operation of taking the Hilbert transform of the log of $\sqrt{P(\omega)}$ requires that the phase angles lie between 0 and 360° on the principal Riemann sheet. The phase angle spectrum associated with the maximum-delay wavelet (1, 2) cannot be recovered; only the ‘analytic’ (computable) minimum-phase spectrum is possible. The following numerical example shows the results for the maximum-delay wavelet (1, 2).

```

h={1,2}; (* Pad with a few trailing zeros for better results *)
Print["Input wavelet is ",h];
.
.
.
----- Output -----
Input wavelet is {1,2}
Autocorrelation of wavelet is {2,5,2}
Wavelet recovered from Hilbert transform: {2,1}
-----
```

Convolution of (2, 1) with (1, 2) results in the mixed-delay wavelet (2, 5, 2) with autocorrelation $\phi(\tau)$ equal to

$$\phi(\tau) = (4, 20, 33, 20, 4) \text{ for } \tau = -2, -1, 0, +1, +2$$

Attempting to recover the mixed-delay wavelet using Hilbert transforms results in the minimum-delay wavelet 4, 4, 1 which has the same autocorrelation as the wavelet (2, 5, 2). The *Mathematica* example is:

```

h={2,5,2}; (* Pad with a few trailing zeros for better results *)
Print["Input wavelet is ",h];
.
.
.
----- Output -----
Input wavelet is {2,5,2}
Autocorrelation of wavelet is {4,20,33,20,4}
Wavelet recovered from Hilbert transform: {4,4,1}
-----
```

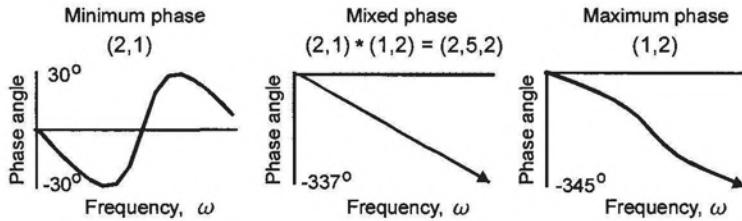


Figure 8.74: Phase spectra of minimum-, mixed-, and maximum-delay wavelets compared. Minimum-delay wavelet is (2,1). Maximum-delay wavelet is (1,2). Mixed-delay wavelet is convolution of (2,1) and (1,2) = (2,5,2). Phase angles for the minimum-delay wavelet are closest to the 0°-degree value. This is therefore a minimum-phase spectrum. All phase spectra are unwrapped. Phase angles for mixed- and maximum-delay wavelets increase without bound as ω increases. Phase spectrum for minimum-delay wavelet is periodic and does not increase without bound. WaveletPhase.cdr \Rightarrow WaveletPhase.wmf

The phase spectra of the “wavelets” (2, 1), (4, 4, 1), and (1, 2) are shown in Figure 8.74. Values of phase angles (unwrapped) in degrees shown for the Fourier transforms of each wavelet on the figure. The phase angle spectrum that stays closest to the horizontal axis corresponding to 0° is that for the minimum-delay wavelet (2,1). Next closest is the spectrum for the mixed-delay wavelet (2,5,2), which is the convolution of (2,1) and (1,2). Plotting furthest from the horizontal axis is the phase angle spectrum corresponding to the maximum-delay wavelet (1,2). The *Mathematica* program used to generate Figure 8.74 is WaveletPhase below.

```
-----WaveletPhase.nb-----
unwrapphase[data_?VectorQ, tol_:Pi, inc_:2Pi]:=data+inc FoldList[Plus, 0,
  Sign[Chop[Apply[Subtract, Partition[data, 2, 1], {1}], tol]]];
-----  

h = {1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  

H = Fourier[h, FourierParameters -> {1, -1}];  

PhaseAngles = ArcTan[Re[H], Im[H]];  

data1 = unrapphase[PhaseAngles]*(180/Pi);  

ListPlot[data1, PlotRange -> All, PlotJoined -> True];
```

These results from simple 2-point wavelets are valid for any length wavelet. Recall from the previous sections that wavelet shaping uses only the autocorrelation r_i and the crosscorrelation g_i of the input with the desired output to change the shape of the input wavelet. That is, the design of the shaping filter f_i . That is, the equations

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{n-1} \\ r_1 & r_0 & \dots & r_{n-2} \\ \vdots & & & \\ r_{n-1} & r_{n-2} & \dots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix}$$

depend upon a *good* estimate of the auto-correlation r_s of the unknown wavelet, which we hope to get from the seismic trace itself. We can obtain an estimate of the wavelet shape by using Hilbert transforms, but the shape will be constrained to be minimum-delay and might not be the shape of the actual wavelet.

Alas, the recovery of the Fourier amplitude and phase spectrum of a time-variant seismic wavelet is not as simple as outlined above. It is even more difficult in the presence of absorption, which is the conversion of the mechanical energy of the wavelet to heat. But the wavelet shape contains much of the history of its journey from the source to the receiver and so remains a continuing focus of effort in reflection seismology. Lindsey [112] for a z -transform approach concluded

The ultimate limit of performance of the method is established by the quality of the estimate of the wavelet autocorrelation. This is fundamental to all data-dependent wavelet extraction methods, including (and especially) deconvolution. This is only minimized by the astute selection of data to be autocorrelated.

8.5.2 Q

This section on Q was written by John Costain drawing on the unpublished Ph.D Dissertaion of Ecevitoglu [66] and the publication of Ecevitoglu and Costain [68]. The *Mathematica* programs were written by Costain who remains responsible for errors of omission or content.

Q stands for *quality factor*. Formulas that define the dimensionless number Q will be given below, but in simple terms the numerical value of Q is just a direct measure of how much mechanical energy is converted to heat as a seismic disturbance propagates away from the its source. Generally speaking, except for unconsolidated, water-saturated sediments, Q has been commonly thought to be essentially independent of frequency; Q is a constant of the medium in which a disturbance propagates. This kind of *absorption* of energy is also called *intrinsic damping*. The energy lost is dissipated and cannot be recovered. There are many reasons why a determination of Q is important. For example, it has been experimentally shown that Q rather than velocity is strongly correlated with permeability (Klimentos and McCann [96]). Thus it should be possible to relate Q anisotropy with anisotropic permeability in partially or completely saturated rocks [74, 75].

The concepts of absorption and body wave dispersion are introduced in this section. Body wave dispersion means that the different frequency components that make up the seismic wavelet travel at different speeds. Higher frequency components travel faster than lower frequency components. Absorption means that the amplitude of the seismic wavelet decreases along the path of propagation as adjacent grains in a rock matrix rub against each other and mechanical energy is continuously converted to heat. You cannot have absorption without dispersion. The reason for this is simple. Absorption causes the seismic wavelet to spread. It spreads to the right toward increasing time and it spreads to the

left toward decreasing time. But it cannot spread into times before the theoretical arrival time of the wavelet. In order to prevent this from happening the different frequencies must travel at different speeds. The lower frequencies travel more slowly thereby keeping the wavelet causal.

The loss in energy has two affects on the seismic wavelet:

1. A gradual decrease in wavelet amplitude along the path of propagation even in a homogeneous, isotropic medium, and
2. A change in wavelet shape—that is, the time between two successive peaks or troughs increases. Also, the relative amplitudes of the peaks and troughs change. For higher- Q rocks the time changes are not much, maybe only a fraction of a millisecond, but the change in wavelet shape is impressive even for higher- Q rocks.

The absorption is different for different frequencies so that the wavelet undergoes a change in shape as well as a change in amplitude. A change in shape means that the “time-shifting theorem” (page 93) is not obeyed. The phenomenon is clearly evident even on a single shot record where it becomes obvious that wavelets recorded at long distances from the sourcepoint are more attenuated and spread out more than those recorded at closer distances. These changes in wavelet shape and amplitude are a result of absorption accompanied by body wave dispersion.

The fact that you cannot have absorption without dispersion means that the phenomenon can be described mathematically by an “absorption-dispersion pair”. There has been much discussion in the literature as to what the correct formulation for this “pair” is. Early choices of a pair by Futterman [72], Azimi [7], Strick [172] and [173], White [192], Kjartansson [94], and Wuenschel [200] led to spirited discussions regarding “the” equation for body wave velocity versus frequency and the absorption-dispersion pair. Later, Ecevitoglu [67, 66, 68] described a different approach to the theoretical formulation of body wave velocity versus frequency and also reconciled the differences between the above authors. It is Ecevitoglu’s [66] approach that is described here.

Introduction

The change in amplitude by body wave dispersion is always associated with a change in shape, so that we are not simply scaling the wavelet amplitude—it does not retain the same shape. High- Q rocks ($Q = 250$, say) are associated with a much smaller energy loss than low- Q rocks ($Q = 30$, say). “High-quality” rocks distort the wavelet less than “low-quality” rocks. Ecevitoglu [66], using methods of singular value decomposition, found a value of $Q = 250$ for the rocks of the indurated but relatively unmetamorphosed Paleozoic shelf sequence in southwestern North Carolina. Demirbag [56, 57, 58, 59], using a generalized linear inversion of the Zoeppritz Equations, found a value of $Q = 50$ for the relatively unconsolidated sediments of the Atlantic Coastal Plain. (In this context one might say that granites are higher quality rocks than shales.) In this section we are concerned only with the absorption of energy due to the mechanical

energy of particle motion that is converted to heat when adjacent grains in a rock matrix rub against each other as the disturbance passes.

The higher frequencies in the wavelet are attenuated more than the lower frequencies but this is not the only reason for the change in wavelet shape. The different frequency components travel at different velocities. Those wavelets that have traveled to more distant receivers are attenuated and changed in shape more than those recorded at near-offset receivers. In other words, the phase shift is not linear with frequency. Clearly then, comparison of the shapes and amplitudes of wavelets recorded at different receiver offsets should contain information about the rock properties traversed by the wavelets. The universally used common-mid-point (CMP) method of acquiring seismic data and then summing traces in a "gather" always sums traces with different source-receiver offsets, and therefore wavelets of different amplitude and shape are summed together even though they have been reflected from the same reflector. This is undesirable but unavoidable; however, a correction for the change in shape and amplitude can be made before summing if the value of Q is known. Intrinsic damping, like velocity or density, is a measurable rock physical property and if Q can be correlated, like velocity and density, with a particular geologic unit, then using this additional information brings us one step closer to a better geologic interpretation.

There are, of course, other factors that cause changes in wavelet amplitude and shape, for example geometrical spreading of the wavefront, destructive interference with another wavelet, reflection or refraction at an interface just to name a few, and we know from this volume that reflection from a thin bed changes wavelet shape as well as amplitude. Attenuation of energy from the seismic wavelet can also result from layering by a combination of transmission losses and intrabed multiples (O'Doherty and Anstey [127]). Schoenberger and Levin [156] reported that attenuation due to layering accounted for 30 – 50 per cent of the total frequency dependent attenuation estimated from field seismograms. Recent interpretations of reflections from the base of the crust indicate layering near the M-discontinuity; this might affect determinations of values of intrinsic damping at these deeper depths. In many recording situations, it would be necessary to correct for all of these other effects, or to demonstrate that they are not important, in order to obtain a true value of intrinsic damping.

In our discussion of Q we do not concern ourselves with the physical mechanisms that cause the absorption of energy because the definition of Q does not depend on the details by which the energy is dissipated [187]. We assume here that intrinsic damping can be isolated in the presence of any other amplitude-changing mechanisms, but this is not always an easy task. The extremely clean waveforms associated with many vertical seismic profiles (VSPs), however, clearly show gradual changes in wavelet shape and should allow quite accurate determinations of Q by the mechanism of intrinsic damping. The widespread use of VSPs in itself justifies a closer look at how to measure intrinsic damping. Dispersion affects traveltimes and is known to be a cause of the discrepancy between integrated sonic times and seismic times [32]. This section is devoted to an understanding of body wave dispersion.

Definitions

The most commonly used measure of absorption by intrinsic damping in rocks is the *quality factor* or *specific dissipation constant* Q . Other commonly used measures of absorption in rocks due to intrinsic damping are the *attenuation coefficient* $\alpha(\nu)$, the internal friction or dissipation factor $1/Q$, and the logarithmic decrement Δ . The lower the value of Q the more the wavelet is attenuated by intrinsic damping and the more its shape as well as its amplitude changes as the travel distance increases. Again, this is an important effect because in a single gather wavelets from the same reflector recorded at short offsets will have a different shape than those recorded at long offsets. Relatively unconsolidated sediments have low values of Q . Granites have much higher values.

As noted, except for unconsolidated, water-saturated sediments, the quality factor Q is believed to be essentially independent of frequency ν and is related to the rate at which the mechanical energy of vibration is converted irreversibly into heat. The above quantities are related [82] as follows:

$$\frac{1}{Q} = \frac{\alpha(\nu) V}{\pi \nu} \quad (8.101)$$

$$\alpha(\nu) = \frac{\pi \nu}{V Q} \quad (8.102)$$

where V is the non-dispersive constant rock velocity at $\nu = \infty$. From (8.102) linear-with-frequency absorption implies a constant Q . The attenuation coefficient α can be expressed in nepers/unit length (or simply inverse length) or in dB per unit length. The relationship between the two is given by α (dB per unit length) = 8.686 α (nepers per unit length). Also, α (dB/ λ) = 8.686 π/Q .

Dispersion

As the higher frequencies are attenuated relative to the lower frequencies, the wavelet changes shape. The change in shape must be such that the wavelet remains causal. In order for this to happen each Fourier frequency component ν must travel at a different phase velocity $p(\nu)$. This dependence of body wave velocity on frequency is called body wave dispersion. From available data it is reasonable to assume that for most rocks over the seismic frequency range of interest α is a linear function of frequency ν [97, 183] as defined by (8.102).

The amplitude $H_2(\nu)$ of a plane seismic wave after having travelled a distance a is

$$H_2(\nu) = H_1(\nu) e^{-\alpha(\nu)a} \quad (8.103)$$

where the amplitude spectra $H_1(\nu)$ and $H_2(\nu)$ are measured at points separated by the travel distance a as shown in Figure 8.75.

Taking the log of each side of (8.103) gives

$$\alpha(\nu) = -\frac{1}{a} \ln \left[\frac{A_2(\nu)}{A_1(\nu)} \right]$$

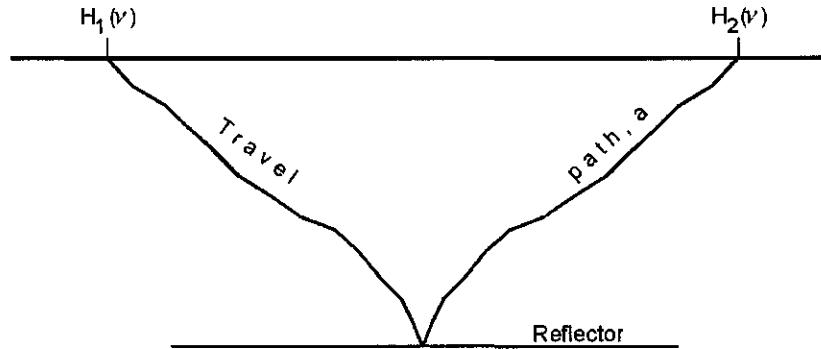


Figure 8.75: Spectra $H_1(\nu)$ and $H_2(\nu)$ are measured at points separated by some travel path distance a . For the discussion of body wave dispersion the reflection coefficient at the reflector is ignored. For a plane wave reflection coefficient (the Zoeppritz equations) it will not affect the shape of a reflected wavelet unless a critical angle is exceeded. Q.cdr → A1A2.wmf

$$\alpha(\nu)a = -\ln \left[\frac{A_2(\nu)}{A_1(\nu)} \right] \quad (8.104)$$

An example of results from ultrasonic measurements is shown in Figure 8.76.

For most rocks a plot of $\alpha(\nu)$ versus ν appears to be linear with frequency [97, 183]. If a least-squares straight line is fitted to $\alpha(\nu)$ versus ν then

$$\alpha(\nu) \text{ is proportional to } \nu$$

and if a is the (constant) distance between observation points, then

$$\alpha(\nu)a = b\nu \quad (8.105)$$

where b is a constant equal to

$$b = \frac{\pi a}{VQ} \quad (8.106)$$

The relationship between $\alpha(\nu)$ and ν is shown in Figure 8.77.

Having defined some well-known relationships we now proceed to derive the expression for the phase velocity p at any frequency ν for sampled data. The earth is a filter and it has a frequency response defined by an amplitude term and a phase term. We recall the relationship between the amplitude $|H(\nu)|$ and phase $\phi(\nu)$ of any function. A complex amplitude spectrum $H(\nu)$ can be written

$$H(\nu) = |H(\nu)| e^{i\phi(\nu)} \quad (8.107)$$

where $H(\nu)$ is the complex Fourier spectrum, $|H(\nu)|$ is the modulus of the spectrum, and $\phi(\nu)$ is the phase angle spectrum. This is simply a definition

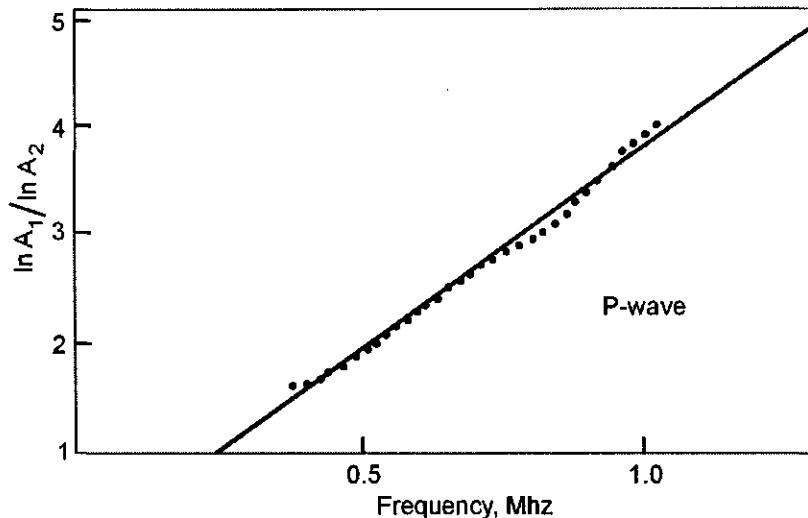


Figure 8.76: Experimental data underlying the assumption of linear-with-frequency attenuation. Figure after Toksoz, Johnston and Timur [183, Figure 2].
Q.cdr → Toksoz.wmf

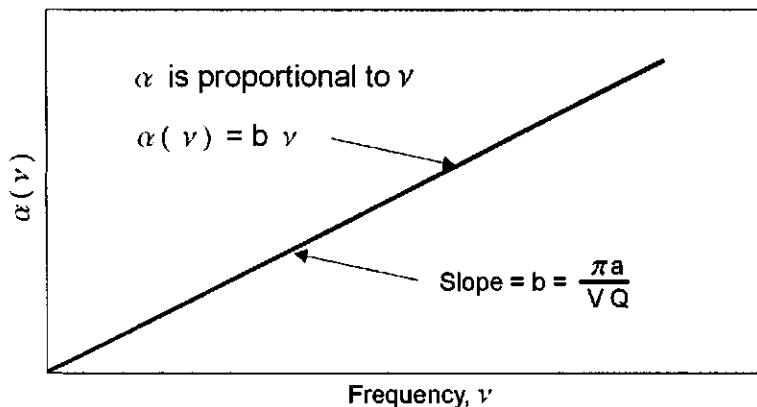


Figure 8.77: For most rocks a plot of $\alpha(\nu)$ versus ν appears to be linear with frequency [187, p. 32]. Q.cdr → AlphaVSnu.wmf

for any function, causal or not. With $h(t) \iff H(\nu)$ if $h(t)$ happens to be some wavelet then an estimate of the wavelet amplitude spectrum $|H(\nu)|$ can be obtained (page 514) by taking the square root of the power spectrum, which we know to be the Fourier transform of the autocorrelation of the wavelet $h(t)$.

The autocorrelation $\phi_{11}(\tau)$ of $h(t)$ is

$$\phi_{11}(\tau) = \int_{-\infty}^{\infty} h(t) h(t + \tau) dt$$

and the Fourier transform of $\phi_{11}(\tau)$ is the power spectrum $\Phi(\nu)$. Thus,

$$\Phi(\nu) = \int_{-\infty}^{\infty} \phi_{11}(\tau) e^{-i2\pi\nu\tau} d\tau = |H(\nu)|^2 \quad (8.108)$$

and (8.107) becomes

$$H(\nu) = \sqrt{\Phi(\nu)} e^{i\phi(\nu)} \quad (8.109)$$

Now, taking the log of both sides of (8.109) we get

$$\begin{aligned} \log H(\nu) &= \log \sqrt{\Phi(\nu)} + i\phi(\nu) \\ &= \frac{1}{2} \log \Phi(\nu) + i\phi(\nu) \\ &= H_e(\nu) + iH_o(\nu) \end{aligned}$$

where $H_e(\nu) = 1/2 \log \Phi(\nu)$ is even and $H_o(\nu) = \phi(\nu)$ is odd. We can obtain the phase spectrum of the wavelet by taking the Hilbert transform of $1/2 \log \Phi(\nu) = H_e(\nu)$. That is,

$$\begin{aligned} \log H(\nu) &= \frac{1}{2} \log P(\nu) + i\mathcal{H}\left[\frac{1}{2} \log P(\nu)\right] \\ &= H_e + iH_o \\ &= H_e + i\mathcal{H}(H_e) \end{aligned}$$

where $\mathcal{H}(\)$ denotes Hilbert transformation of the argument in parentheses. Having now both the amplitude and phase spectrum we could take the inverse Fourier transform of $H(\nu)$ to obtain the causal wavelet $h(t)$. The mathematical relationship (Hilbert transformation) between the real and imaginary parts of (8.110) guarantees that the wavelet $h(t)$ will be causal.

Now suppose that $h(t)$ is instead the impulse response of the earth and that $h(t) \iff H(\nu)$. We don't have access to $h(t)$ but we do know from experiments and observation how the amplitude $H(\nu)$ of any single Fourier component of frequency ν falls off with distance a (Figure 8.75). An impulse $\delta(t)$ leaving the sourcepoint contains all frequencies each with a Fourier amplitude of "1". After a sinusoid of amplitude "1" travels a distance a then from (8.103) its amplitude is given by

$$(1) e^{-\alpha(\nu)a} \quad (8.110)$$

This is the amplitude part of the complex absorption response $H(\nu)$ of the earth. Following the notation of (8.109) we can write the general expression for the frequency response $H(\nu)$ of the earth as

$$H(\nu) = (1) e^{-\alpha(\nu)a} e^{i\phi(\nu)} \quad (8.111)$$

Now take the log of (8.111) as we did before and get

$$\log H(\nu) = -\alpha(\nu) a + i\phi(\nu)$$

But from (8.105)

$$-\alpha(\nu) a = -b\nu$$

So

$$\begin{aligned} \log H(\nu) &= -\alpha(\nu) a + i\phi(\nu) \\ &= -b\nu + i\phi(\nu) \end{aligned} \quad (8.112)$$

So, $-b\nu$ is the part of the earth absorption filter that is the analog of $1/2 \log \Phi(\nu)$. When we took the Hilbert transform of $\log |H(\nu)|$ we got a phase spectrum $\phi(\nu)$ with very special properties—the inverse Fourier transform of $H(\nu)$ will be causal. So (8.112) can be written

$$\begin{aligned} \log H(\nu) &= [-\alpha(\nu) a] + i\phi(\nu) \\ &= -b\nu + i\phi(\nu) \\ &= -b\nu + i\mathcal{H}(-b\nu) \end{aligned} \quad (8.113)$$

This says that the phase spectrum $\phi(\nu)$ of the causal earth absorption filter can be obtained by taking the Hilbert transform of $-b\nu$. This phase spectrum is special because of the causal nature of the Hilbert transformation so we assign to it a different symbol $B(\nu)$:

$$B(\nu) = \mathcal{H}(-b\nu)$$

Because b is a constant the quantity $B(\nu)$ can be written in several equivalent ways:

$$\begin{aligned} B(\nu) &= \mathcal{H}(-b\nu) \\ &= -b\mathcal{H}(\nu) \\ &= b\mathcal{H}(-\nu) \end{aligned}$$

We will arbitrarily adopt the latter form of the expression for $B(\nu)$ because it looks nice in the final form of the derivation of body wave dispersion (Equation (8.127) on page 528). Thus,

$$\log H(\nu) = -b\nu + iB(\nu) \quad (8.114)$$

$$= H_e(\nu) + iH_o(\nu) \quad (8.115)$$

What if the phase term $B(\nu)$ is ignored and only the amplitude part defined by Equation (8.110) is used? This won't work because the attenuation of frequency components in $H_1(\nu)$ results in spreading of the wavelet, as shown in Figure 8.78. The wavelet spreads to the right toward increasing time as well as to the left

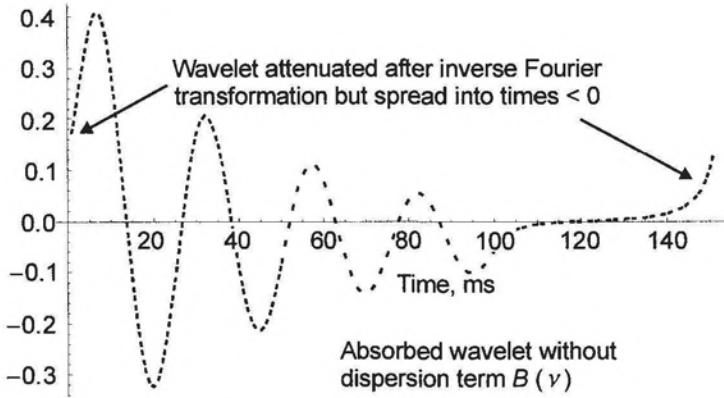


Figure 8.78: Absorption without dispersion. The attenuated wavelet but without using the $B(\nu)$ term spreads into negative as well as positive time after inverse Fourier transformation and therefore violates causality. This is similar to the unabsoed wavelet shown in Figure 7.16. `Absorption-Spreading.nb → Q.cdr → WithoutB.wmf`

toward decreasing time. But it cannot spread into times before the wavelet is supposed to arrive based on the known rock velocity and source-receiver distance. In order to prevent this from happening the different frequencies ν that make up the spectrum of the wavelet must travel at different speeds. This means that the phase spectrum $B(\nu)$ cannot be zero.

We can determine H_o from H_e in (8.115) for continuous data by using Equation (3.125) on page 161, which is repeated here for convenience:

$$\begin{aligned} H_o(\omega) &= -\frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega |t| dt \int_{-\infty}^{\infty} H_e(u) \cos u t du \\ H_o(\nu) &= - \int_{-\infty}^{\infty} \sin 2\pi\nu |t| dt \int_{-\infty}^{\infty} H_e(u) \cos 2\pi u t du \end{aligned} \quad (8.116)$$

Once we have $H_o(\nu)$ we can construct $H(\nu)$ by

$$\begin{aligned} H(\nu) &= e^{H_e(\nu) + i H_o(\nu)} \\ &= e^{-b\nu + i\mathcal{H}(-b\nu)} \end{aligned}$$

and then take the inverse Fourier transform to get $h(t)$. That is,

$$h(t) = \int_{-\infty}^{\infty} [e^{-b\nu + i\mathcal{H}(-b\nu)}] e^{+i2\pi\nu t} d\nu$$

The function $h(t)$ is the response of the Earth to an impulsive input. It is the absorption response of the Earth. The impulse response is causal because of the Hilbert transformation step used to compute the phase spectrum.

Equation (8.116) becomes

$$B(\nu) = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega |t| dt \int_{-\infty}^{\infty} H_e(\nu) \cos ut du \quad (8.117)$$

From (8.112) we substitute $H_e(\nu) = -b\nu$ in (8.117), which then becomes

$$B(\nu) = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega |t| dt \int_{-\infty}^{\infty} (-b\nu) \cos ut du \quad (8.118)$$

In summary, the Hilbert transform of $-b\nu$ gives the phase $B(\nu)$ associated with linear-with-frequency absorption (the causal absorption) of the "Earth filter". Equation (8.118) is a Hilbert transformation. In (8.118), $-b\nu$ is an even function of ν and $B(\nu)$ is odd. If we take the inverse Fourier transform of $H(\nu)$ (which we will display) we obtain the absorption response $h(t)$ of the Earth to an impulse.

So far we have just looked at the part $B(\nu)$ of the total phase spectrum $\theta(\nu)$ associated with causal absorption. There is also a part $2\pi\nu\tau$ associated with non-dispersive traveltime. The total phase angle $\theta(\nu)$ including the non-dispersive part and the dispersive part of the total phase angle $\theta(\nu)$ is

$$\theta(\nu) = 2\pi\nu\tau + B(\nu) \quad (8.119)$$

where τ is the non-dispersive (pure) traveltime—the linear-with-frequency part from the "time-shifting theorem"—as the frequency component ν travels the distance a in the time τ .

Thus [66, Equation 14, p. 17] from Equation (8.119)

$$\theta(\nu) = 2\pi\nu t = 2\pi\nu \frac{a}{p(\nu)} \quad (8.120)$$

where t is the total time equal to the linear-with-frequency time plus the dispersive time. From (8.119) and (8.120) we have

$$2\pi\nu \frac{a}{p(\nu)} = 2\pi\nu\tau + B(\nu) \quad (8.121)$$

Solving (8.121) for $p(\nu)$ we get

$$p(\nu) = \frac{2\pi\nu a}{B(\nu) + 2\pi\nu\tau} \quad (8.122)$$

In the denominator of (8.122) we substitute

$$B(\nu) = b\mathcal{H}(-\nu) \quad (8.123)$$

We make one other substitution. The quantity τ in (8.122) is a non-dispersive traveltime. If we choose an upper frequency limit ν_N above which there is assumed to be no dispersion then we can write

$$\tau = \frac{a}{p[\nu_N]} \quad (8.124)$$

The upper frequency limit can be conveniently taken to be the Nyquist frequency. It may be, as with Wuenschel's [200] experimental data for Plexiglas, that this will be a real data point on an actual dispersion curve, as discussed on page 544. Note that we have switched from continuous notation for $p(\nu)$ to discrete notation $p[\nu_N]$ because we will be taking the discrete Hilbert transform.

Substituting (8.123) and (8.124) in (8.122) we get

$$\begin{aligned} p[\nu] &= \frac{2\pi\nu a}{B[\nu] + 2\pi\nu\tau} \\ &= \frac{2\pi\nu a}{b\mathcal{H}[-\nu] + 2\pi\nu \frac{a}{p[\nu_N]}} \end{aligned} \quad (8.125)$$

The quantity b in (8.125) is

$$b = \text{the slope of } \alpha(\nu) \text{ versus } \nu = \frac{\pi a}{Q p[\nu_N]} \quad (8.126)$$

Making this substitution in (8.125) we get

$$\begin{aligned} p[\nu] &= \frac{2\pi\nu a}{\frac{\pi a \mathcal{H}[-\nu]}{Q p[\nu_N]} + 2\pi\nu \frac{a}{p[\nu_N]}} \\ &= \frac{2\pi\nu}{\frac{\pi \mathcal{H}[-\nu]}{Q p[\nu_N]} + \frac{2\pi\nu}{p[\nu_N]}} \\ &= \frac{2\pi\nu p[\nu_N]}{\frac{\pi \mathcal{H}[-\nu]}{Q} + 2\pi\nu} \\ &= \frac{p[\nu_N] 2\pi\nu}{\left(\frac{2\nu}{2\nu}\right) \left(\frac{\pi \mathcal{H}[-\nu]}{Q}\right) + 2\pi\nu} \\ p[\nu] &= \frac{p[\nu_N]}{1 + \frac{1}{2Q} \frac{\mathcal{H}[-\nu]}{\nu}} \end{aligned} \quad (8.127)$$

Clearly, the phase velocity $p[\nu]$ cannot depend on a sampling interval, which is implied by the value of the velocity $p[\nu_N]$ at the Nyquist frequency ν_N . What the velocity $p[\nu_N]$ does is to scale the dispersion curve without changing its shape; that is, $p[\nu_N]$ scales the quantity

$$\frac{1}{1 + \frac{1}{2Q} \frac{\mathcal{H}[-\nu]}{\nu}}$$

as defined by (8.127). As already noted, $p[\nu_N]$ at ν_N might be an actual observed coordinate on a dispersion curve as is the case for Wuenschel's Plexiglas data (see page 544).

Equation (8.127) is Ecevitoglu's equation for the computation of the exact shape of the body wave phase velocity $p[\nu]$ versus frequency ν in a dispersive medium for any value of Q [66, 68, Equation 21, p. 20]. Every value of dispersive velocity can be computed using (8.127) up to $\nu = \nu_N$; there are no arbitrary constant(s) to be chosen. Although the notation $\mathcal{H}[-\nu]$ looks like one is taking the Hilbert transform of the variable ν it can be interpreted as taking the discrete Hilbert transform of a straight line defined by $m\nu + b$ where the slope $m = -1$ and the intercept $b = 0$.

If Q and one other phase velocity $p[\nu]$ are known then $p[\nu_N]$ can be computed from Equation (8.127). For example, Wuenschel's [200] experiment with the Pierre Shale yielded a coordinate {150 Hertz, 6835 ft/sec}. For $Q = 30$ using this pair in (8.127) and using **Dispersion.nb** to compute the value $\mathcal{H}(-\nu) = \mathcal{H}(-150)$ gives

$$\begin{aligned} p[\nu_N] &= p[\nu] \left[1 + \frac{1}{2Q} \frac{\mathcal{H}[-\nu]}{\nu} \right] \\ &= p[150] \left[1 + \frac{1}{2Q} \frac{\mathcal{H}[-150]}{150} \right] \\ &= 6835 \left[1 + \frac{1}{2 \times 30} \frac{165.008}{150} \right] \\ &= 6960.31 \end{aligned}$$

This value can also be obtained by trial-and-error by running the program **Dispersion.nb**, guessing at any value of $p[\nu_N]$ above 150 Hertz, and changing it until $p[150] = 6835$. The result will be $p[\nu_N] = 6960$. This is so because $p[\nu_N]$ is just a scale factor and does not influence the shape of (8.127).

If Q is unknown but $p[\nu_N]$ is known as well as any one coordinate $\{\nu, p[\nu]\}$ then Q can be computed from (8.127). Wuenschel's [200] experiment with the Pierre Shale yielded observed coordinates {150 Hertz, 6835 ft/sec} and {500, 6960}. Using these in (8.127) and computing the value $\mathcal{H}(-\nu) = \mathcal{H}(-150)$ gives

$$\begin{aligned} Q &= - \frac{p[\nu] \mathcal{H}[-\nu]}{2\nu (p[\nu] - p[\nu_N])} \\ &= - \frac{(6835)(165.008)}{2(150)(6835 - 6960)} \\ &= 30 \end{aligned}$$

It will be useful to display graphically the intermediate and final steps in the derivation of (8.127). A plot of $-\nu$ versus ν is shown in Figure 8.79a, then $\mathcal{H}[-\nu]$ (Figure 8.79b), and finally $p[\nu]$ (Figure 8.79c). First, Equation (3.125), which we repeat here, is reviewed in the context of body wave dispersion:

$$H_o(\omega) = -\frac{1}{2\pi} \int_{-\infty}^{\infty} \sin \omega |t| dt \int_{-\infty}^{\infty} H_e(u) \cos u t du$$

$$\begin{aligned} H_o(\nu) &= - \int_{-\infty}^{\infty} \sin 2\pi\nu|t| dt \int_{-\infty}^{\infty} H_e(u) \cos 2\pi u t du \\ B(\nu) &= - \int_{-\infty}^{\infty} \sin 2\pi\nu|t| dt \int_{-\infty}^{\infty} (-u) \cos 2\pi u t du \quad (8.128) \end{aligned}$$

The Hilbert transform of $-\nu$ is obtained by first taking the inverse Fourier transform of $-\nu$, which is the outer integration in (8.128):

$$\int_{-\infty}^{\infty} (-u) \cos 2\pi u t du \quad \text{will be an even function in the time domain}$$

This outer integration will be an even function in the time domain because $-\nu$ (or the dummy variable “ $-u$ ”) is even in the frequency domain. We can conveniently use the *Mathematica* function `InverseFourier` to do the inverse transform defined by the outer integral of (8.128). That is, in order to evaluate the outer integral of (8.128) by using *Mathematica*’s subroutine `InverseFourier` an even function of $-\nu$ was simply *constructed* as illustrated at the top of Figure 8.79. We are operating on an even frequency-domain function so the inverse operation is really just an inverse cosine transform, as required by (8.128). Now we are in the time domain. With reference to Figure 3.47 on page 130 we know that to obtain the causal function associated with $-\nu$ (the real part of the Fourier transform of the unknown causal function) we must suppress the negative times of this even time-domain function (whatever it is, we don’t know or care yet) and multiply the result by “2”. This is exactly what $|t|$ and multiplication by “2” do in (8.128). Now we have a causal function (that we have not yet inspected). Take the Fourier transform of this causal function. Equation (8.128) says to take the sine transform but using `Fourier` gives us that if we just look at the odd part of the transform of the causal function obtained by use of $|t|$ and “2”. The odd part of the resulting Fourier transform is shown in Figure 8.79b. From (8.128) the odd part of the Fourier transform is the Hilbert transform of the even part.

Figures 8.79a and b constitute a Hilbert transform pair. The Hilbert transformation is done in the `hilbertF` *Mathematica* subroutine in programs *Absorption.nb* and *Dispersion.nb*. The transformation is the discrete version of Equation (8.116). The program is

```
(* Program hilbertF *)
hilbertF[RealPart_] := (lh = Length[RealPart];
Print["Length of input to Hilbert = ", lh];
If[OddQ[lh], lh2 = Ceiling[lh/2], lh2 = lh/2];
(* Go to the time domain *)
fd = Re[InverseFourier[RealPart, FourierParameters -> {1, -1}]];
(* The result will be a pure real even function *)
(* Now impose symmetry to get a causal time domain function *)
Do[fd[[i]] = 2*fd[[i]], {i, 1, lh2}];
Do[fd[[i]] = 0, {i, lh2 + 1, lh}];
```

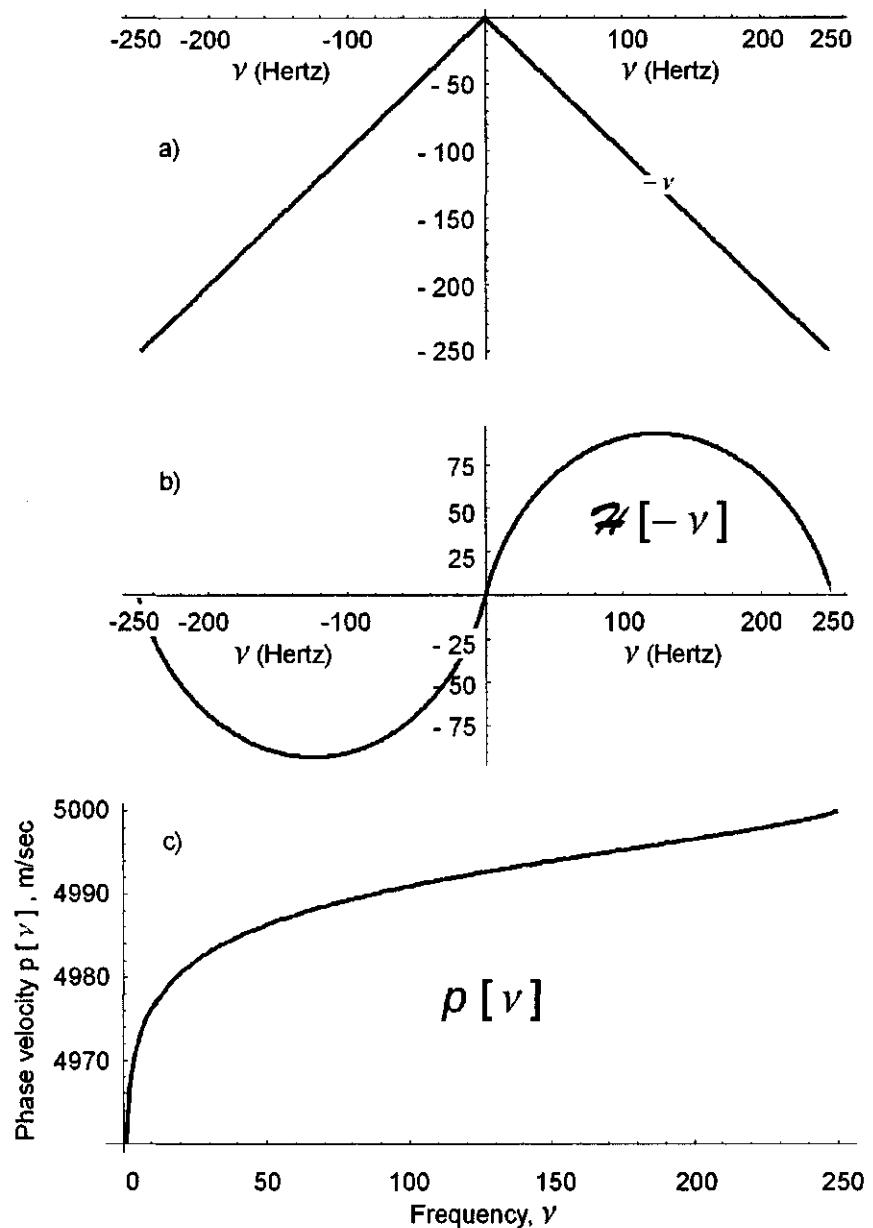


Figure 8.79: (a) Plot of $-\nu$ before Hilbert transformation, which is an even function by construction because we wish to use *Mathematica*'s InverseFourier subroutine for computations. (b) $\mathcal{H}[-\nu]$, the Hilbert transform of $-\nu$, which gives an odd function. (c) Dispersive velocity $p[\nu]$ as determined from Equation (8.127) for $Q = 250$ and $p[\nu_N] = 5000$ m/sec. Only velocity values for positive ν need to be displayed. Dispersion.nb → Q.cdr → Hplots.wmf

```
(* Now take the Fourier transform of fd to get Ho *);
HilbertTransform =
    -Chop[Im[Fourier[fd, FourierParameters -> {1, -1}]]];;
```

How do the above results for the computation of body wave dispersive velocities compare with those of others [7, 72, 94, 172, 173] who approached the problem differently? In a classic paper Futterman [72] gave an approximation to the dispersive phase velocity $c(\nu)$ versus ν as

$$c(\nu) = \frac{c_0}{1 - \frac{1}{2\pi Q} \ln \left| \frac{\nu^2}{\nu_0^2} - 1 \right|} \quad (8.129)$$

where c_0 is the velocity at ν_0 , an arbitrary low-frequency cutoff. His Equation (8.129) requires the choice of two constants c_0 and ν_0 whereas (8.127) requires none. Following Futterman, ν_0 is chosen below the frequency range of interest. Because the choice is arbitrary we can choose $\nu_0 = 1$ Hertz. Having chosen ν_0 , the constant c_0 can be chosen such that the theoretical dispersion curve defined by (8.129) passes through some known coordinate $\{\nu, c(\nu)\}$ that has been determined from observed data. As an example, Wuenschel [200, p. 544] chose a value of c_0 such that the dispersion curve defined by (8.129) passed through the point whose coordinates are {150 Hertz, 6835 ft/sec}, the latter values determined from observed data from the Pierre Shale. Equation (8.129) can then be used to determine the entire dispersion curve as shown by the upper thin solid line in Figure 8.80. Equation (8.127) on the other hand requires no arbitrary constants. It describes an exact shape. The value of $p[\nu_N]$ is actually the dispersive velocity value that would be obtained experimentally for a rock of a given value of Q at the frequency ν_N . The fact that ν_N happens to be designated as the “Nyquist frequency” here is immaterial and is simply computationally convenient.

The physical process of absorption between a source and a receiver is complete when the wavelet is recorded at the receiver. The absorption process is independent of any sampling interval and therefore of any value of ν_N . In order to implement (8.127), however, we must assign a value of ν_N and $p[\nu_N]$.

Equation (8.127) requires no arbitrary constants (actually, not even a value for $p[\nu_N]$) to give the correct *shape* of the dispersion curve. It describes an exact shape. A value of a coordinate $\{\nu_0, c(\nu_0)\}$ is required to compute Futterman’s dispersion curve but the shape of the curve is approximate. Equation (8.129) can therefore be used to approximate velocities over a limited frequency range but not the shape of the entire curve unless many values of c_0 are used and it is determined piecewise. Futterman’s approach acknowledges the observation by Aki and Richards [3] that it is not possible to take the Hilbert transform of a straight-line (the linear-with-frequency assumption for attenuation) that goes off to infinity and therefore does not converge. Futterman approached this problem by deriving an approximation that does converge but that predicts velocity $c(\nu)$ versus frequency ν only over a limited frequency interval. Ecevitoglu

and Costain [68] approached the problem by assuming that the linear-with-frequency variation was periodic over the interval $\nu = 0$ to $\nu = \nu_N$ and that velocity values $p[\nu]$ could be computed over the entire range of frequencies to give the correct shape of the dispersion curve. (Review the Dirichlet conditions on page 64 for the conditions under which a Fourier series—read “Hilbert transform”—converges to the function wherever it is continuous.) The assumption of periodicity results in a phase that is periodic and therefore bounded versus Aki and Richard’s phase that is unbounded and increases without limit. The theoretical result of (8.127) is nicely supported by the Plexiglas dispersion experiment of Wuenschel [200] as discussed on page 544. Although $p[\nu_N]$ in (8.127) can be thought of as a scale factor that plays the same role that c_0 does in Equation (8.129), it is more than that; in the case of (8.127) it is a real, physical velocity $p[\nu_N]$ at the frequency ν_N associated with a particular value of Q . We compare numerical values from each method below. See Table 8.1 on page 539 for a summary of differences between the two approaches to body wave dispersion as well as those of others.

Comparison of dispersive phase velocity values of Ecevitoglu and Costain [68] with Futterman [72]

The theoretical dispersion curves of Futterman [72] and Ecevitoglu and Costain [68] both require prior knowledge of a single {velocity, frequency} coordinate through which, in the case of (8.127), the exact shape of the dispersion curve will pass. As an example, $p[\nu_N]$ can be chosen by trial and error such that $p[150]$ passes through Wuenschel’s observed coordinate of {150 Hertz, 6835 ft/sec}, as shown in Figure 8.80. This value of $p[\nu_N]$, once chosen, is the actual dispersive rock velocity that would be obtained if we were to make a velocity measurement at that frequency. Once $p[\nu_N]$ has been determined the entire exact body wave dispersion curve can be computed from (8.127) instead of from (8.129).

Using Wuenschel’s results for the Pierre Shale and a value of $Q = 30$ and selecting a value of $\nu_N = 500$ Hertz (clearly $\nu_N = 500$ from his figure) then we can use the *Mathematica* program *Dispersion.nb* to compute the entire theoretical dispersion curve. In this program, the value of $p[\nu_N]$ is changed by trial and error until it is found to be equal to 6960 ft/sec when $\nu = 150$ Hertz and $p[150] = 6835$ ft/sec. Now, with $p[\nu_N]$ set equal to 6960 ft/sec in (8.127) we have the dispersion curve for all frequencies from $\nu = 0$ to $\nu_N = 500$. Wuenschel set the constants $\{\nu_0, c_0\}$ in (8.129) equal to {1 Hertz, 6522 ft/sec} so that the theoretical curve from (8.129) can approximate the observed data. From (8.127) we compute the pair {1, 6477} in agreement with Wuenschel to within about 0.6 %.

As a comparison with (8.129) take several values for $Q = 30$ obtained from (8.127) and compare them with those obtained from (8.129). The values of {frequency, velocity} obtained using (8.127) with $p[\nu_N] = 6960$ are:

{10., 6645.54}, {20., 6692.48}, {30., 6720.3}, {40., 6740.21},
 {50., 6755.77}, {60., 6768.56}, {70., 6779.44}, {80., 6788.93},
 {90., 6797.35}, {100., 6804.93}, {110., 6811.83}, {120., 6818.18},
 {150., 6834.69}, {200., 6856.74}, {300., 6890.78}, {400., 6920.47}

These values can be compared with Futterman's equation after choosing $\nu_0 = 1$ Hertz and $c_0 = 6522$ ft/sec, the latter value having been chosen as described above. Futterman's equation is repeated here for convenience as

$$c(\nu) = \frac{c_0}{1 - \frac{1}{2\pi Q} \ln \left| \frac{\nu^2}{\nu_0^2} - 1 \right|} = \frac{6522}{1 - \frac{1}{2\pi Q} \ln \left| \frac{\nu^2}{1^2} - 1 \right|} \quad (8.130)$$

The corresponding values from (8.130) for the same frequencies are

{10., 6684.89}, {20., 6735.95}, {30., 6766.06}, {40., 6787.57},
 {50., 6804.35}, {60., 6818.11}, {70., 6829.79}, {80., 6839.94},
 {90., 6848.92}, {100., 6856.97}, {110., 6864.27}, {120., 6870.95},
 {150., 6888.13}, {200., 6856.74}, {300., 6942.05}, {400., 6964.68}

The percent differences between (8.127) and (8.130) for the various { frequency, velocity } values are

{10., -0.59}, {20., -0.65}, {30., -0.68}, {40., -0.70},
 {50., -0.72}, {60., -0.73}, {70., -0.74}, {80., -0.75},
 {90., -0.76}, {100., -0.76}, {110., -0.77}, {120., -0.77},
 {150., -0.78}, {200., -0.78}, {300., -0.74}, {400., -0.64}

and all are in agreement with (8.127) to within a fraction of one per cent. The measured value of the phase velocity at 400 Hertz is 6860 ft/sec [200, p. 543, Fig. 4]. This differs from (8.130) by -1.5 % but differs from (8.127) by only -0.88 %, suggesting that the value given by the discrete Hilbert transform used by (8.127) might be a better estimate. The above numerical values were obtained using the *Mathematica* program *Dispersion\-\Futterman\-\Comparison.nb* on the CD-ROM.

Additional comparisons with Futterman are shown graphically in Figure 8.82 from Ecevitoglu and Costain [68] where Futterman's [72] velocity dispersion expression is superimposed in a piecewise manner upon the curve defined by Equation (8.127). The excellent agreement using constants c_0 and ν_0 for Futterman obtained from (8.127) is apparent in the figure. Note that it is not possible to superimpose Futterman's entire results with a single selection of his constants. On the other hand, there are no arbitrary constants that govern the shape of the velocity dispersion curves computed from discrete Hilbert transformation.

This leads us to Figure 8.81 and generalized dispersion curves. To obtain numerical values from the curves you must multiply the vertical axis by $p[\nu_N]$

and the horizontal axis by ν_N . Said another way, you must know the rock velocity $p[\nu_N]$ at the frequency ν_N . But it is not necessary that the value of $p[\nu_N]$ at ν_N for each curve be known directly from observational data; however, a control point $\{\nu, p[\nu]\}$ from observed data must be used to estimate $p[\nu_N]$, as was done for the above example for the Pierre Shale. For an example where this is unnecessary and the $p[\nu_N]$ and ν_N pair are known directly from observational data see page 544 and Wuenschel's [200] Plexiglas experiment.

As noted earlier several attempts have been made to formulate analytical expressions for absorption and the body wave dispersion that must accompany absorption. It is beyond the scope of this volume to delve into the details of this history. An excellent review can be found in [66]. Aki and Richards [3, pages 172–177] summarized difficulties with analytic expressions that involve the total phase (linear-with-frequency phase plus dispersive phase) and frequency limits of integration that extend to infinity. Ecevitoglu and Costain [68] suggested that the problems of a divergent integral as described by them [3, Equation 5.74] can be overcome by invoking discrete numerical Hilbert transformation and by recognizing that the total phase spectrum can be split up into a linear-with-frequency nondispersing phase defined by the traveltime of a reflected event plus a pure dispersive phase spectrum that is associated with the body wave dispersion required by causal absorption. A summary of the phase definitions of Futterman [72], Strick [172], Kjartansson [94] and the later results of Ecevitoglu and Costain [68] that have been discussed here is shown in Table 8.1. For additional details the reader is referred to Ecevitoglu [66].

This completes our discussion of body wave dispersion, which is one member of the “absorption-dispersion pair”. We now turn our attention to absorption, the other member of the pair, and its effect on the shape of the wavelet.

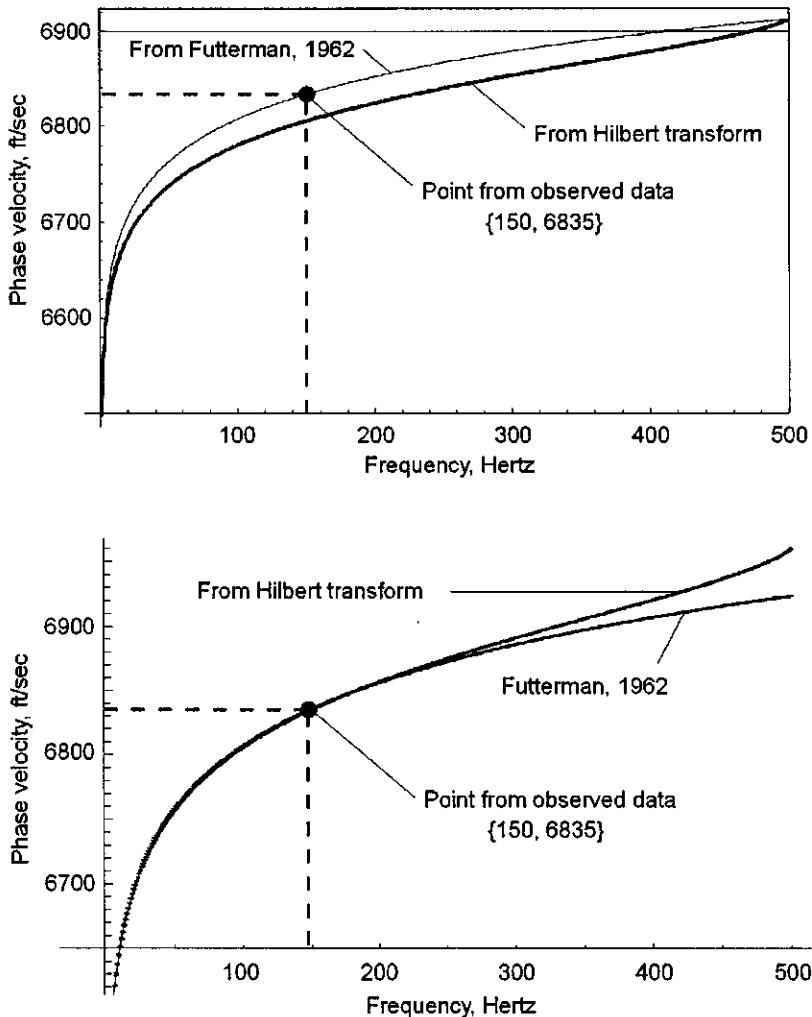


Figure 8.80: Top diagram: Upper light line is the dispersion curve in Pierre Shale (dashed line) from Wuenschel [200] computed using Equation (8.129) and $Q = 30$. Lower heavy line is the exact dispersion curve from Equation (8.127) using the discrete Hilbert transform. The curves are not identical in shape because Futterman's Equation (8.129) is an approximation. The solid line could just as well have been scaled by the value $p[\nu_N]$ so that it passed through the dot that represents the coordinate of the observed data, as shown in the bottom diagram. But then the solid line would not have agreed with (8.129) elsewhere. A visual inspection of Figure 5 of Wuenschel [200] suggests that the dispersion curve computed by (8.127) might fit the observed data better than that determined from Futterman's Equation (8.129). Dispersion.nb → Dispersion.cdr → Wuenschel.wmf

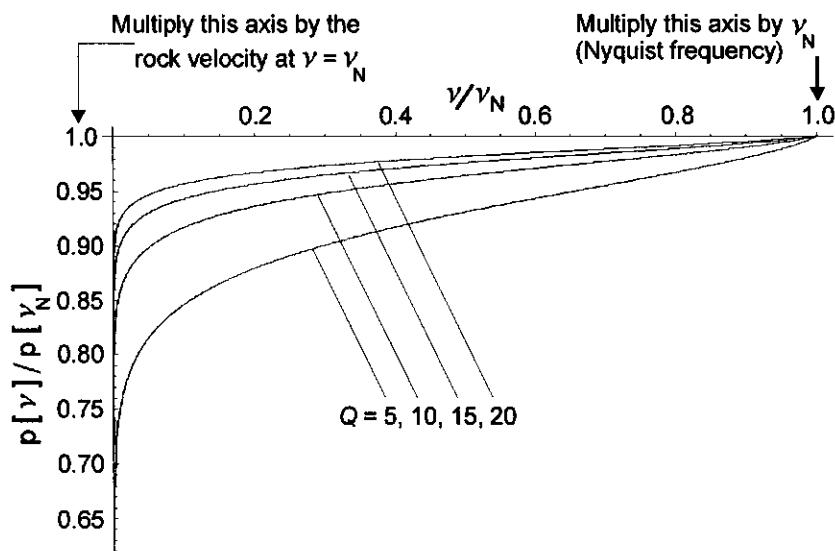
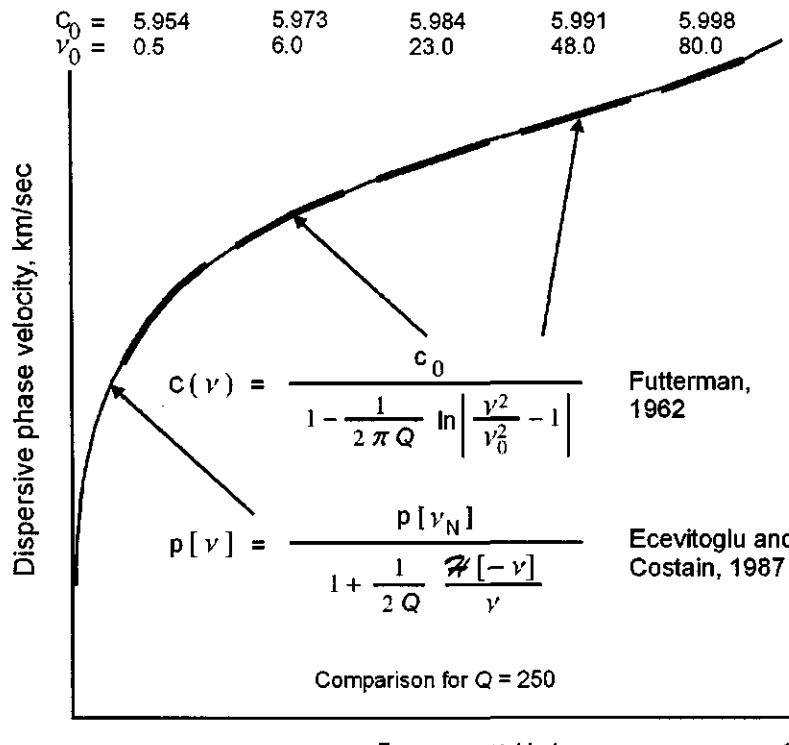


Figure 8.81: Generalized body wave dispersion curves $p[\nu]$ for $Q = 5, 10, 15, 20$. $p[\nu]/p[\nu_N]$ determined from Equation (8.127). Frequency ν normalized by division by $1/(2\Delta t) = \nu_N$ where Δt is the sampling interval. The quantity $p[\nu_N]$ is the rock velocity at the Nyquist frequency, not at $\nu = \infty$. If Q is known it is not necessary to know $p[\nu_N]$. In general a control point $\{\nu, p[\nu]\}$ will be used to determine $p[\nu_N]$ as per the example given for the Pierre Shale. GeneralizedQ.nb → Generalized-BodyWaveDispersion.cdr → NormalizedQ.wmf

Frequency v , Hertz

125

Figure 8.82: Solid, thinner continuous line is the result of using Equation (8.127) to compute phase velocity $p[v]$ versus v . Short and thicker discontinuous line segments superimposed on the continuous curve are computed from Futterman's [72] Equation (8.129) for phase velocity, which is shown on the figure. The constants c_0 and v_0 used in Futterman's equation are taken from Equation (8.127). The agreement is excellent but the use of Futterman's equation requires a choice of arbitrary constants c_0 and v_0 whereas the solid line resulting from the use of discrete Hilbert transformation $\mathcal{H}[-v]$ is exact and requires no arbitrary constants. Comparison made for $Q = 250$. Figure from Ecevitoglu and Costain [68]. Dispersion.nb → Q.cdr → DiscreteVsContinuous.wmf

Table 8.1: Summary of equations for absorption-dispersion pairs

Absorption coefficient	Phase velocity	Total phase
Futterman, 1962 $\alpha(\nu) = b \nu $	Futterman, 1962 $c(\nu) = \frac{c_0}{1 - \frac{1}{2\pi Q} \ln \nu^2/\nu_0^2 - 1 }$	Futterman, 1962 $\theta(\nu) = 2\pi\nu\tau - \frac{\tau}{Q} \nu \ln \left \frac{\nu^2}{\nu_0^2} - 1 \right $
Strick, 1970 $\alpha(\nu) = K \cos\left(\frac{s\pi}{2}\right) 2\pi\nu ^s$	Strick, 1970 $c(\nu) = \frac{c_\infty}{1 + c_\infty K \sin\left(\frac{s\pi}{2}\right) 2\pi\nu ^{s-1}}$	Strick, 1970 $\theta(\nu) = 2\pi\nu\tau + K \sin\left(\frac{s\pi}{2}\right) 2\pi\nu ^s$
Kjartansson, 1979 $\alpha(\nu) = \frac{2\pi\nu_0}{c_0 \tan\left(\frac{s\pi}{2}\right)} \left \frac{\nu}{\nu_0} \right ^s$	Kjartansson, 1979 $c(\nu) = c_0 \left \frac{\nu}{\nu_0} \right ^{1-s}$	Kjartansson, 1979 $\theta_K(\nu) = 2\pi\nu\tau \left \frac{\nu}{\nu_0} \right ^{s-1}$
Ecevitoglu and Costain, 1988 $\alpha[\nu] = \frac{\pi\nu}{Q p[\nu_N]}$	Ecevitoglu and Costain, 1988 $p[\nu] = \frac{p[\nu_N]}{1 + \frac{1}{2Q} \frac{\mathcal{H}[-\nu]}{\nu}}$	Ecevitoglu and Costain, 1988 $\theta[\nu] = 2\pi\nu\tau + \frac{\pi\tau}{Q} \mathcal{H}[-\nu]$

Absorption

Wuenschel [200] noted that

For purposes of seismogram synthesis, a few percent variation in phase velocity can produce a significant change in pulse waveform for long propagation distances. The pulse waveform can be thought of as a vernier to magnify small differences in phase velocity.

If only absorption is taken into account without the $B[\nu]$ causal phase term then the initial wavelet spreads into negative as well as positive time as shown earlier in Figure 8.78, and therefore violates causality. With the addition of the $B[\nu]$ term the lower frequency sinusoids are required to travel more slowly in just the right amount to keep the wavelet causal.

The plots in Figure 8.79a and b constitute a Hilbert transform pair. It is important to note that the shape of Figure 8.79b is actually the shape found by Ecevitoglu [66, Figure 27, p. 95] from real conventional CDP reflection data analyzed by singular value decomposition. His results might be the first to demonstrate causal absorption from conventional CDP reflection data. It is also apparent that if absorption is *not* linear with frequency then this will show up in the analysis because the shapes of Figures 8.79a and b are Hilbert transforms of each other. For example there might be a “hump” or a local depression on the linear-with-frequency curve that will also affect the causal phase. One possibility of the resulting Hilbert transform pair from Ecevitoglu [66, p. 148] is shown in Figure 8.83. According to White [191] “...there is ample evidence from laboratory and field studies to show that Q is not constant over large frequency intervals and that different mechanisms can be called into play.”

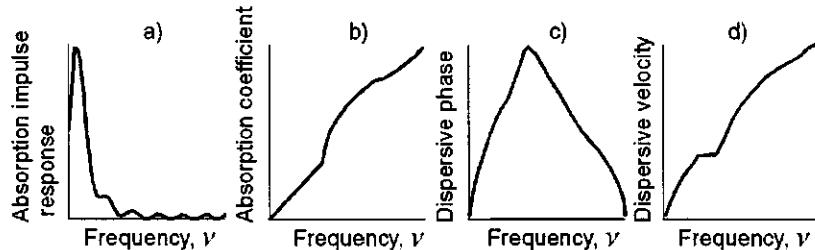


Figure 8.83: The method [66, 68] described in this section is not restricted to a linear-with-frequency (constant Q) function. For example (a) is the Earth's absorption impulse response for a not-linear-with-frequency absorption coefficient. (b) The function $\alpha(\nu)$ versus ν , not a linear function of frequency. (c) What the dispersive phase would look like for this form of $\alpha(\nu)$ versus ν . (d) The corresponding phase velocity $p[\nu]$. Figure from Ecevitoglu [66, p. 148]. Q.cdr → hump.wmf

In our absorption Earth model (absorption only, no spherical spreading for example) the source emits an impulse $\delta(t)$ that contains all frequencies ν and all

of equal amplitude. The amplitudes of frequency components from this impulsive source are gradually attenuated by absorption along the travel path a . This results in a lengthening of the impulse $\delta(t)$ until after a distance $a = 1500$ m, for example, in an absorptive medium with a quality factor $Q = 20$. The Dirac delta function has lengthened and now looks like that shown in Figure 8.84. As discussed above, this absorption impulse response $h(t)$ of the Earth to an impulsive plane wave is

$$h(t) = \int_{-\nu_n}^{+\nu_N} (1) e^{-b\nu + iB(\nu)} e^{i2\pi\nu t} d\nu \quad (8.131)$$

Here $B(\nu)$ is the Hilbert transform of $-b\nu$ and from (8.106)

$$\begin{aligned} b &= \frac{\alpha(\nu) a}{\nu} \\ &= \frac{\pi}{VQ} \end{aligned}$$

We could convolve $h(t)$ with a seismic wavelet to get the new causal shape at the distance a . Alternatively we can multiply the Fourier transform of $h(t)$ by the spectrum of the source wavelet at $a = 0$. The latter choice was made in *Mathematica* program *Absorption.nb* to obtain the attenuated wavelets shown in Figures 8.87 and 8.88.

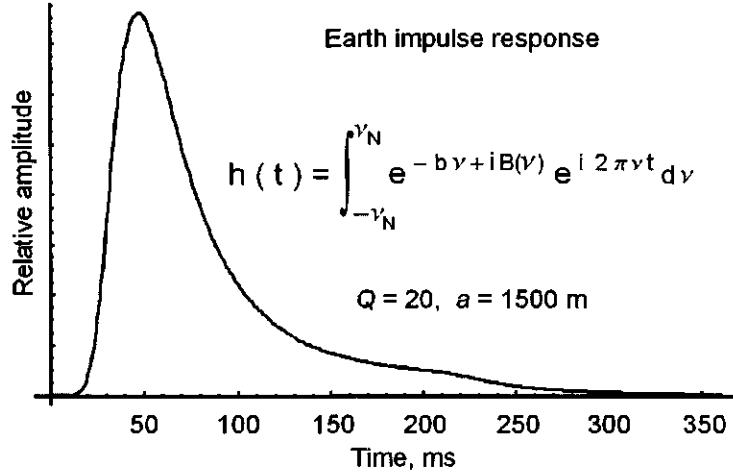


Figure 8.84: (a) The absorption impulse response of the Earth for $Q = 20$ and $a = 1500$ m. This is a plot of Equation (8.131). *Absorption.nb* → *Q.cdr* → *AbsorptionImpulseResponse.wmf*

Gladwin and Stacey [78] used the theories of Futterman [72], Strick [171, 172] and Azimi et al. [7] to calculate the shapes of impulse responses after

propagation in a medium of $Q = 50$ to a distance of $a = 5$ m. Their results for Futterman's theory are shown in Figure 8.85 along with the shape computed from Equation (8.127). The comparison is excellent. The different arrival times shown in their Figure 4 for Futterman [72], Strick [171, 172], and Azimi et al. [7] are due to the "hidden" phases summarized in Table 8.1 and discussed in detail by Ecevitoglu [66]. As noted by Gladwin and Stacey [78] the use of Futterman's theory by them resulted in a waveform that arrives too early. Ecevitoglu [66, p. 7 and Figure 2] explained this phenomenon by showing that a second term (a "hidden" phase term resulting from the approximation in Futterman's theory) in the phase angle spectrum subtracts from the linear-with-frequency term and causes the early arrival.

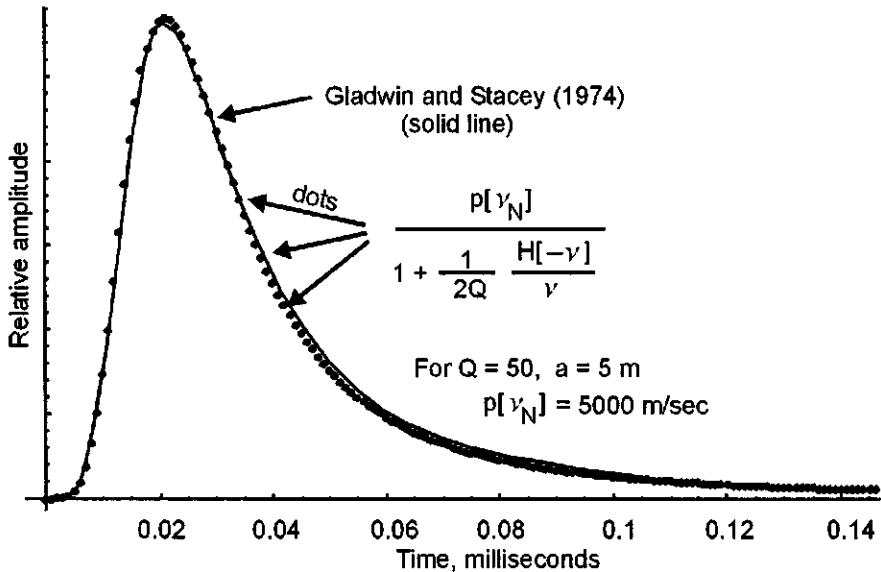


Figure 8.85: Solid line from Gladwin and Stacey [78, Figure 4] is their calculated shape of the absorptive impulse response using the theory of Futterman [72] after propagation in a medium of $Q = 50$ and velocity 5,000 m/sec to a distance of $a = 5$ meters. Superimposed dots are from using the discrete Hilbert transform and Equation (8.127) with $p[\nu_N] = 5000$ m/sec, $a = 5$ meters, and $Q = 50$. The agreement is excellent. Sample interval used in (8.127) is 1 μ sec. Nyquist frequency = 500,000 Hertz. The impulse response arrives "on time" if (8.127) is used. GladwinAndStacey.nb,cdr → GladwinAndStacey.wmf

Normalized dispersion D

The normalized amount D of body wave dispersion, hereafter referred to as just "dispersion", over the bandwidth from $\nu = 0$ to $\nu = \nu_N$ can be defined as [66,

p. 21]

$$D = \frac{p[\nu_N] - p[0]}{p[\nu_N]} \quad (8.132)$$

The Nyquist frequency component ν_N and the frequency component at $\nu = 0$ both travel the same distance a (Figure 8.75) at their respective phase velocities. So

$$p[\nu_N] \tau = p[0] [\tau + \Delta W] = a$$

where “ p ” denotes phase velocity and ΔW is a dispersive time delay due to body wave dispersion. Solving for ΔW

$$\Delta W = \frac{p[\nu_N] - p[0]}{p[0]} \tau \quad (8.133)$$

From (8.127) and taking into account limiting values [66, p. 21] of $\mathcal{H}(-\nu)/\nu$ for $\nu = 0$ and $\nu = \nu_N$ the phase velocity at $\nu = 0$ is

$$p[0] = \frac{p[\nu_N]}{1 + \frac{2}{\pi Q}} \quad (8.134)$$

Repeating (8.132) and substituting (8.134) gives

$$\begin{aligned} D &= \frac{p[\nu_N] - p[0]}{p[\nu_N]} \\ &= 1 - \frac{1}{1 + \frac{2}{\pi Q}} \\ D &= \frac{1}{1 + \frac{\pi}{2} Q} \end{aligned} \quad (8.135)$$

which says that the maximum amount of velocity dispersion for a given value of Q is independent of any frequency ν , even ν_N . For example, if $Q = 20$ then

$$D = \frac{1}{1 + \frac{\pi}{2} Q} = \frac{1}{1 + \frac{\pi}{2} 20} = 0.031$$

which says that the maximum amount of body wave dispersion between $\nu = 0$ and $\nu = \infty$ for $Q = 20$ is 3.1 %. Values of D over a more restricted range of frequencies are easily determined by measurements in the frequency domain as discussed below.

Comparison of dispersion D values with real data

It is possible to estimate the amount of dispersion over any bandwidth between two phase velocity values p at different frequencies ν_1 and ν_2 using

$$D = \frac{p[\nu_2] - p[\nu_1]}{p[\nu_2]} \quad (8.136)$$

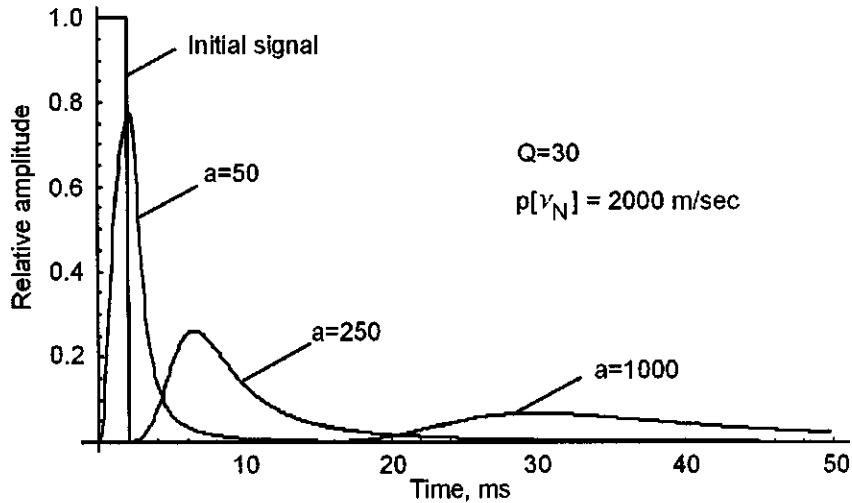


Figure 8.86: Wavelet spreading for $Q = 30$ for various distances a . For this example the “wavelet” is a rectangular function of width 2 ms. Wavelet broadening increases for increasing values of the offset distance a but all wavelets remain causal. `AbsorptionSpreading.nb` → `RelativeSpreading.cdr` → `RelativeSpreading2.wmf`

This formula can be applied to several excellent data sets such as the pulse-propagation experiment in the Pierre Shale by McDonal et al. [118]. Wuenschel [200] reported an observed variation in phase velocity of 2.6 % between 25 and 400 Hertz for the Pierre shale, to which he assigned a value of $Q \approx 30$. According to (8.136) this value of Q corresponds to $D = 2.1\%$. Waters [187, p. 33] assigned an average value of $Q = 17$ to samples of the Pierre shale for phase velocity determinations over the frequency range 75–555 Hertz. According to (8.135) a value of $Q = 17$ corresponds to $D = 3.6\%$. Considering the apparent range of values of Q for the Pierre shale the values of total dispersion from (8.135) appear to be in excellent agreement with published values over more restricted frequency ranges.

Wuenschel [200] published a composite dispersion curve for Plexiglas. From his experimental data he observed that for Plexiglas the phase velocities vary by 2.3 % over a five-octave range from 4 to 128 kilocycles per sec. Using (8.127) it is possible to estimate a value for D over this same frequency range by setting $p[v_N]$ equal to 7720 ft/sec, which is the observed value of velocity that can be visually estimated from his Figure 12. Wuenschel gives an estimate of 55 for Q . Running the *Mathematica* program `Dispersion.nb` with $Q = 55$ and $p[v_N] = 7720$ gives values for $p[4]$ and $p[128]$ of 7541.19 and 7707.08 ft/sec, respectively. Inserting these computed values into the expression for D in Equa-

tion (8.136) gives

$$D = \frac{p[128] - p[4]}{p[128]} = \frac{7707.08 - 7541.19}{7707.08} = 2.15 \times 10^{-2} \approx 2.2 \%$$

in close agreement with Wuenschel's observed value of 2.3 %. In addition, the theoretical shape of the dispersion curve and the numerical values computed from (8.127) are in excellent agreement with his results for Plexiglas. The experimental data set of Wuenschel [200] appears to be the best available data set to date that supports prediction of the exact shape as well as the numerical values of the dispersion curve as computed by the discrete Hilbert transform and Equation (8.127). In order to compare the observed dispersion curve with the theoretical one defined by (8.127) we required no arbitrary constants and simply set $p[\nu_N]$ equal to the velocity (7720 ft/sec), a velocity that was actually measured at 140 kilocycles/sec as shown in Wuenschel's Figure 12.

A time-domain method for determination of Q

Body wave dispersion and absorption result in a change of wavelet shape. Gladwin and Stacey [78] and Kjartansson [94] showed that wavelet broadening due to absorption and body wave dispersion is proportional to the wavelet traveltime and is related to Q by the empirical equation

$$W_2 = \frac{C}{Q} (\tau_2 - \tau_1) + W_1 \quad (8.137)$$

where τ_1 is the wavelet travelttime on one trace and τ_2 is the travelttime to, say, an adjacent trace. W_2 is the pulse width at travelttime τ_2 , W_1 is the pulse width at time τ_1 , and C is an arbitrary constant to be determined empirically from some other means of measuring Q such as spectral ratio techniques. Ecevitoglu [66, p. 23] pointed out that there is no consensus in the literature about the exact value of C . Kjartansson [94] used $C = 0.485$ and Gladwin and Stacey [78] and Badri and Mooney [9] used $C = 0.5$. Ecevitoglu [66] derived an exact value of C as

$$C = \frac{2}{\pi}$$

Ecevitoglu's [66] exact equation corresponds to the empirical Equation (8.137) of Gladwin and Stacey [78] and Kjartansson [94]. From (8.133) and (8.135) we have

$$\begin{aligned} \Delta W &= \frac{p[\nu_N] - p[0]}{p[0]} \tau \\ &= \frac{p[\nu_N] - p[0]}{p[\nu_N]} \frac{p[\nu_N]}{p[0]} \tau \\ &= D \frac{p[\nu_N]}{p[0]} \tau \end{aligned}$$

$$\begin{aligned}
 &= \frac{1 + \frac{2}{\pi Q}}{1 + \frac{\pi Q}{2}} \tau \\
 &= \frac{2}{\pi Q} \tau
 \end{aligned}$$

or

$$Q = \frac{2\tau}{\pi \Delta W} \quad (8.138)$$

This is Ecevitoglu and Costain's [66, 68, Equation 28, p. 23] result for the determination of Q in the time domain. See Figures 8.87 and 8.88 for an application. The quantity τ is the total travelt ime during which body wave dispersion in the amount of $\Delta W = W_2 - W_1$ has taken place. For our model example W_1 is the pulse width of the wavelet at the source at $a = 0$. W_2 is the pulse width after the wavelet has traveled for the time τ . For values of W_1 and W_2 one measures the same peak to peak, trough to trough, or peak to trough time duration even though the wavelet is now distorted by causal absorption. Values of τ and W_i will normally have to be in fractions of a millisecond. Estimates of fractional values can be obtained from the *Mathematica* graphics output of program *Absorption.nb* by holding down the "Ctrl" key while moving the cursor over the graphics output to the desired locations shown in Figure 8.87. A better approach might be to use

$$f(t) = a_0 + 2 \sum_{n=1}^{\infty} a_n \cos\left(2\pi \frac{n}{T} t\right) - 2 \sum_{n=1}^{\infty} b_n \sin\left(2\pi \frac{n}{T} t\right) \quad (8.139)$$

for an interpolation to a finer sampling interval and then use (8.138).

In (8.138) the time τ is to be compared with the difference between traveltimes τ_1 and τ_2 in Gladwin and Stacey's Equation (8.137). That is, for Gladwin and Stacey, dispersion has taken place during the time $\tau_2 - \tau_1$ whereas in (8.138) dispersion has taken place during the entire time τ . That is, in Equation (8.137) τ_1 is zero and τ_2 in (8.137) is then the same as τ in (8.138). So (8.137) and (8.138) are the same except for the constant C in their equation, which was found by Ecevitoglu to be

$$C = \frac{2}{\pi}$$

The delay of the absorbed wavelet of about 10 ms shown in Figure 8.87 is an artifact of the processing and does not affect the results. The delay is due entirely to the choice of the model sampling interval of 0.1 ms in order to facilitate accurate picking of the values of W_1 and W_2 . The small sampling interval of the modeled data implies high frequencies that are not present in the wavelet so these higher frequencies do arrive before the lower frequencies, as they should, but they have essentially zero amplitude. The result is the 10 ms "delay". If actual field data were to be recorded at a sample interval of 0.1 ms (same as the modeled data) then the attenuated wavelet would be shifted to the

left and the “delay” would disappear. If the wavelets modeled in Figure 8.87 were instead generated using a sampling interval of 4 ms then the “delay” would disappear but it would not be possible to pick accurate times for the peaks or troughs. On the other hand, the resolution could be increased to 0.1 ms by an interpolation using Equation (8.139).

If a field study is designed to examine body wave dispersion in low- Q rocks then a small sampling interval, say 0.1 ms, should be used. If the data are already recorded at, say 4 ms, then (8.139) can be used to resample at 0.1 ms with equivalent results to search for body wave dispersion.

It is clear that Ecevitoglu’s time-domain approach to the measurement of Q has several advantages. For example, it is not necessary to see the entire wavelet. Under ideal conditions only a single peak or trough and another adjacent to it are necessary. So overlapping wavelets and windowing effects become less important. The redundancy of reflection seismic data means that the occasional noisy trace can be tolerated. Q can be determined using single CDP gathers and changes monitored horizontally and vertically. Ecevitoglu used singular value decomposition along with a great multiplicity of data to estimate Q in a crustal volume using (8.138) as well as to produce what might be the first measured observation of the phase angle spectrum associated with body wave dispersion. From this, the Hilbert transform yielded the attenuation coefficient as a function of frequency.

Summary

The results described in this section for the computation of body wave dispersion are in agreement with those of earlier workers but the approach is more general because no arbitrary constants are required to determine the shape of the body wave dispersion curve at any frequency, thus defining “the” pair of absorption-dispersion equations as discussed on page 519 and summarized in Table 8.1. Although a linear-with-frequency attenuation is used here as an example the numerical approach described by Ecevitoglu [66] is appropriate for *any* behavior of $\alpha(\nu)$ versus ν . This has not been emphasized in the above discussion except for Figure 8.83 but it means that the inverse problem, that of determining $\alpha(\nu)$ versus ν from Hilbert transformation of the phase spectrum, the phase spectrum being derived from real data, will reveal the nonlinear dependence of $\alpha(\nu)$ versus ν if it is present in the data. This should have important implications for detecting frequency ranges over which values of Q are different but constant over limited segments.

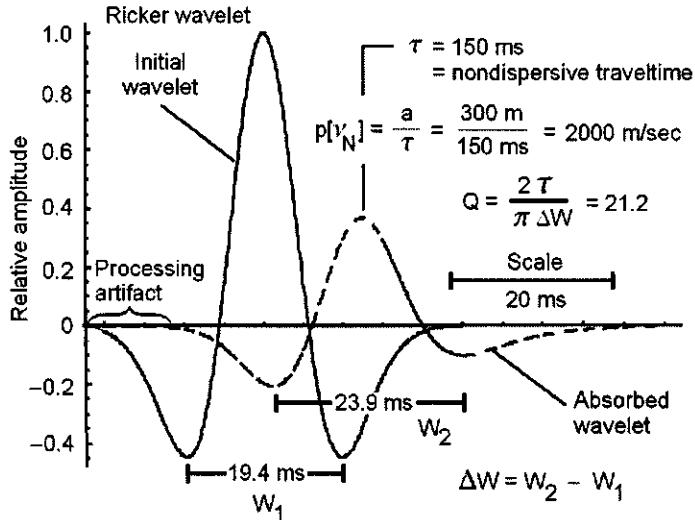


Figure 8.87: Ricker wavelet shape change for $Q = 20$, $a = 300$ m, $p[v_N] = 2000$ m/sec. Sampling interval $\Delta t = 0.1$ ms. Times shown are in ms. Solid line is initial wavelet. Dashed line is attenuated and dispersed Ricker wavelet at a source-receiver distance of 300 m. Value of Q recovered from the analysis using Equation (8.138) is 21.2 versus model value of 20. Absorption.nb → QFromTimeDomain.cdr → QRicker.wmf

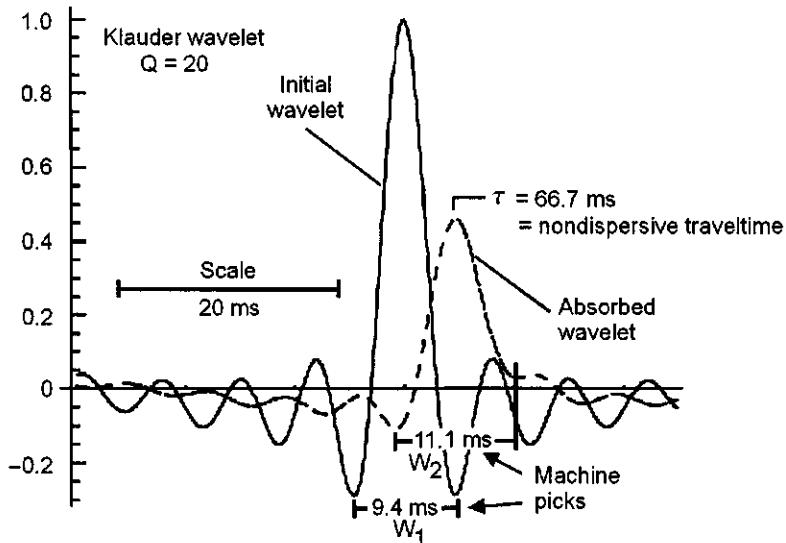


Figure 8.88: Klauder wavelet shape change for $Q = 20$, $p[v_N] = 1500$ m/sec at a distance of $a = 100$ meters. $\Delta t = 0.1$ ms. Solid line is initial Klauder wavelet. Dotted line is attenuated and dispersed Klauder wavelet. The value of Q recovered from the analysis using Equation (8.138) and using a search algorithm for troughs is 25 versus the model value of 20. Decreasing W_1 by 0.1 ms and increasing W_2 by 0.2 ms results in a value of $Q = 20.2$. Absorption.nb → QFromTimeDomain.cdr → QKlauder.wmf

References

- [1] Abriel, W.L. and Wright, R.M. The shapes of gulf coast salt intrusions related to seismic imaging. *The Leading Edge*, August, pages 868–872, 1994.
- [2] Ryuzo Adachi. On a proof of fundamental formula concerning refraction method of geophysical prospecting and some remarks. *Kumamoto Journal of Science*, 2:18–23, 1954.
- [3] Aki, K. and P. G. Richards. *Quantitive Seismology - Theory and Methods*, volume 1. W.H. Freeman and Company, New York, 1980. 557 pp.
- [4] D.F. Aldridge. Mathematics of linear sweeps. *Canadian Journal of Exploration Geophysics*, 28(1):62–68, 1992.
- [5] F. (Editor) Aminzadeh. Section 1. seismic exploration. In Klaus Helbig and Sven Treitel, editors, *Pattern Recognition and Image Processing*, volume 20 of *Handbook of Geophysical Exploration*, page 568. Geophysical Press, 1987.
- [6] N.A. Anstey. *Vibroseis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [7] Azimi, Sh. A., A.Y. Kalinin, V.V. Kalinin and B.L. Pivovasov. Impulse and transient characteristics of media with linear and quadratic absorption laws. *Izv. Earth Physics*, 2:88–93, 1968.
- [8] Milo M. Backus. Water reverberations—their nature and elimation. *Geophysics*, 24:233–261, 1959.
- [9] Badri, M. and Mooney, H.M. q measurements from compressional seismic waves in unconsolidated sediments. *Geophysics*, 52:772–784, 1987.
- [10] Bahorich, M. and Çoruh, C, Robinson, E.S., and Costain, J.K. Static corrections on the southeastern piedmont of the united states. *Geophysics*, 47:1540–1549, 1982.
- [11] Bahorich, M. and Farmer, S. The coherence cube. *The Leading Edge*, pages 1053–1058, 1995.

- [12] N.F. Barber. *Experimental Correlograms and Fourier Transforms*. The MacMillan Company, New York, 1961. 136 pages.
- [13] K.M. Barry. Delay time and its application to refraction profile interpretation. In A.W. Musgrave, editor, *Seismic Refraction Prospecting*, pages 348–361. Society of Exploration Geophysics, 1967.
- [14] Bay, J.A. and Stokoe, II, K.H. Development of a rolling dynamic deflectometer for continuous deflection testing of pavements. *Center for Transportation Research, University of Texas, Austin*, page 237 pp, 1998.
- [15] Belcher, S.W., Pratt, T.L., Costain, J.K. and Çoruh, C. Alternative processing techniques and data improvement provided by single-sweep recording. *Geophysics*, 51:1736–1742, 1986.
- [16] Ninos Benyamin. Key elements of total seismic field design using mathematica—a tutorial. *Geophysics*, 67:1020–1027, 2002.
- [17] A.J. Berkhouit. Least squares inverse filtering and wavelet deconvolution. *Geophysics*, 38:1369–1383, 1977.
- [18] A.J. Berkhouit. Section 1. seismic exploration. In *Seismic Resolution, Handbook of Geophysical Exploration*. Geophysical Press, 1984. 228 pp.
- [19] Mary L. Boas. *Mathematical Methods in the Physical Sciences*. John Wiley and Sons, 1983. 793 pp.
- [20] B. Boashash. Estimating and interpreting the instantaneous frequench—part 1: Fundamentals. In *Proceedings of the IEEE*, volume 80, pages 520–538. IEEE, 1992.
- [21] B. Boashash. Estimating and interpreting the instantaneous frequench—part 2: Algorithms and applications. In *Proceedings of the IEEE*, volume 80, pages 540–568. IEEE, 1992.
- [22] Bobyarchick, A.R. and Edelman, S.H. and Horton, Jr., J.W. The role of dextral strike-slip in the displacement history of the brevard zone. In D.T. Secor, Jr., editor, *Southeastern Geological Excursions*, pages 53–154. South Carolina Geological Survey, Columbia SC, 1988.
- [23] H.W. Bode. *Network Analysis and Feedback Amplifier Design*. Van Nostrand, Princeton, New Jersey, 1945.
- [24] R.N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Book Company, New York, 1965. 381 pp.
- [25] R.N. Bracewell. *The Hartley Transform*. Oxford University Press, 1986. 176 pp.
- [26] R.N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Higher Education, 1999. 3rd ed., 640 pp.

- [27] E.O. Brigham. *The Fast Fourier Transform*. Prentice-Hall, Englewood Cliffs, N.J., 1974.
- [28] J.G. Bryan. Statistical test of the hypothesis that a time series is stationary. *Geophysics*, 32:499–511, 1967.
- [29] R.A. Bryan. Thin-bed resolution from cepstrum analysis. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1985.
- [30] Burrus, C.S. and Eschenbacher, P.W. A statistically optimized deconvolution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29:806, 1981.
- [31] S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7:536–541, 1930.
- [32] Peter Cary. On r.e. white's short note "the accuracy of estimating q from seismic data". *The Leading Edge*, April:256–258, 1993.
- [33] Peter Cary. Seismic signal processing –a new millenium perspective. *Geophysics*, 66(1):18–20, 2001.
- [34] Castagna, J.P. and Backus, M.M., editors. *Offset-Dependent Reflectivity—Theory and Practice of AVO Analysis*, volume 8 of *Investigations in Geophysics*. Society of Exploration Geophysicists, 1993. 348 pp.
- [35] Lines, Larry, Castagna, J.P., and Treitel, Sven. Geophysics in the new millennium. *Geophysics*, 66:14, 2001.
- [36] C. Çoruh. Stretched automatic amplitude adjustment of seismic data. *Geophysics*, 50:252–256, 1985.
- [37] C. Çoruh and J.K. Costain. Noise attenuation by vibroseis whitening(vsw) processing. *Geophysics*, 48:543–554, 1983.
- [38] R.V. Churchill. *Fourier Series and Boundary Value Problems*. McGraw-Hill, 1963. 248 pp.
- [39] Jon F. Claerbout. *Imaging the Earth's Interior*. Blackwell Scientific Publications, Palo Alto, CA 94303, 1985. 398 pp.
- [40] Jon F. Claerbout. *Earth Soundings Analysis – Processing Versus Inversion*. Blackwell Scientific Publications, Cambridge, MA 02142, 1992. 304 pp.
- [41] Clark,H.B. and Costain, J.K. and Glover, III, L. Structural and seismic reflection studies of the brevard ductile deformation zone near rosman, north carolina. *Amer. Jour. Sci*, 278:419–441, 1978.

- [42] Cooley, J. W. and Tukey, J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [43] C.T. Copson. *Introduction to the Theory of Functions of a Complex Variable*. Oxford University Press, 1935.
- [44] Cormier, V.F. and Mellen, M.H. Application of asymptotic ray theory to synthetic vertical seismic profiling. In Helbig, K. and Treitel, S., editors, *Vertical Seismic Profiling*, volume 14B, pages 28–44. Geophysical Press, London, 1984.
- [45] Çoruh, C. and Bollinger, G.A., and Costain, J.K. Seismogenic structures in the central virginia seismic zone. *Geology*, 16:748–751, 1988.
- [46] Çoruh, C., Costain, J.K. and Stephenson, D.E. Composite refraction-reflection stack sections: Tracing faults in the atlantic coastal plain sediments. *Society of Exploration Geophysicists, Extended Abstracts of the Technical Program*, 2:1160–1163, 1993.
- [47] Çoruh, C., Domoracki, W.J., Costain, J.K., Selvi, O. and Stephenson, D.E. Composite refraction-reflection stack sections: Imaging shallow subsurface structures. *Proceedings of the Symposium on the Application of Geophysics to Engineering and Environmental Problems*, pages 937–946. SAGEEP, 1995.
- [48] Çoruh, C., Sen, A.K., Domoracki, W.J., Costain, J.K., Stephenson, D.E., Stieve, A. and Stephenson, D.E. Faulting in the coastal plain sediments: Seismic images from composite reflection and refraction stack sections. *Geological Society of America Abstracts With Programs*, 24:9–10, 1992.
- [49] Costain, J.K. and Çoruh, C. Tectonic settings of triassic half-grabens in the appalachians: Seismic data acquisitin, processing, and results. In Tankard, A. J. and Balkwill, H. R., editors, *Extensional Tectonics and Stratigraphy of the North Atlantic Margins, AAPG Memoir 46*, AAPG Memoir 46. American Association of Petroleum Geologists - Canadian Geological Foundation, 1989. pages 155–174.
- [50] Costain, J.K. and Çoruh, C. and Domoracki, W.J. Continuity of deep crustal reflection seismic signatures in the central appalachian orogen. *Geological Society of America, Abstracts With Programs*, 27:37, 1995.
- [51] Costain, J.K. and Hatcher, R.D., Çoruh, C., Pratt, T.L., Taylor, S.R., Litehiser, J.J., and Zietz, I. Geophysical characteristics of the appalachian crust. In Hatcher, R.D, Jr. and Thomas, W.A., and Viele, G., editors, *The Appalachian-Quachita Orogen in the United States, The Geology of North America*. Geological Society of America, Boulder, Colorado, 1989. pages 385–416.

- [52] Costain, J.K. and Robinson, E.S. Some seismic measurements on the virginia coastal plain. *Virginia Polytechnic Institute Water Resources Research Center Bulletin 56*, page 69, 1972.
- [53] Crawford, J.M., Doty, W.E.N. and Lee, M.R. Continuous signal seismograph. *Geophysics*, 25:95–105, 1960.
- [54] K.S. Cressman. How velocity layering and steep dip affect cdp. *Geophysics*, 33:399–411, 1968.
- [55] Darby, E.K. and Neidell,N.S. Application of dynamic programming to the problem of plane wave propagation in a layered medium. *Geophysics*, 31(6):1037–1048, 1966.
- [56] E. Demirbag. *Estimation of Seismic Parameters from Multifold Reflection Seismic Data by Generalized Linear Inversion of Zoepritz Equations*. Ph.d. dissertation in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1990.
- [57] Demirbag, E. and Çoruh, C. Inversion of zoepritz equations and their approximations. *Soc. Expl. Geophys., Expanded Abstracts, 58th Ann. Internat. Mtg.*, pages 1199–1203, 1988.
- [58] Demirbag, E. and Çoruh, C. Inversion of multilayer amplitude-versus-offset data. *Soc. Expl. Geophys., Expanded Abstracts, 58th Ann. Internat. Mtg.*, pages 709–712, 1989.
- [59] Demirbag, E., Çoruh, C. and Costain, J.K. Inversion of avo. In John P. Castagna and Milo M. Backus, editors, *Offset-Dependent Reflectivity – Theory and Practice of AVO Analysis*, volume 8 of *Investigations in Geophysics Series*. Society of Exploration Geophysicists, 1993. pages 287–302.
- [60] Dey, A.K. and Lines, L.R. Seismic source wavelet estimation and the random reflectivity assumption. *CREWES Research Report*, 10:21–1–21–28, 1998.
- [61] C. Hewitt Dix. Seismic velocities from surface measurements. *Geophysics*, 20:68–86, 1955.
- [62] C.H. Dix. *Seismic Prospecting for Oil*. Harper and Brothers, 1952.
- [63] W.J. Domoracki. *A Geophysical Investigation of Geologic Structure and Regional Tectonic Setting at the Savannah River Site, South Carolina*. Ph.d. dissertation in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1995. 236 pp.
- [64] Domoracki, W.J., Çoruh, C. and Costain, J.K. Faulting in atlantic coastal plain sediments at the savannah river site, sc: Interpretations from seismic reflection time maps. *Geological Society of America Abstracts with Programs*, 26, 1994. p. 11.

- [65] Domoracki, W.J., Costain, J.K. and Çoruh, C. Processing and interpretation of seismic reflection data near the bane dome in bland county, virginia. Technical report, U.S. Nuclear Regulatory Commission, NUREG/CR-5417, 1989. 24 pp.
- [66] Berkan G. Ecevitoglu. *Velocity and Q from multifold seismic data using SVD*. Ph.d. dissertation in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1987.
- [67] Galip Berkan Ecevitoglu. Use of wavelet distortion from supercritical reflections to detect lateral velocity variations. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1984.
- [68] Ecevitoglu, B.G. and Costain, J.K. A new look at body wave dispersion. *Society of Exploration Geophysicists Extended Abstracts*, 2:1043–1045, 1988.
- [69] R.W. Edsall. A seismic reflection study over the bane anticline. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1974. 109 pp.
- [70] B.J. Evans. *A Handbook for Seismic Data Acquisition in Exploration*. Society of Exploration Geophysicists, Tulsa, OK 74137, 1997. 305 pp.
- [71] Richard P. Feynman. *The Feynman Lectures on Physics*. Addison-Wesley Publishing Co., 1963. Commemorative Issue, 1989, California Institute of Technology.
- [72] W.I. Futterman. Dispersive body waves. *Journal of Geophysical Research*, 67:5279–5291, 1962.
- [73] Gardner, G.H.F., Gardner, L.W. and Gregory, A.R. Formation velocity and density: The diagnostic basis for stratigraphic traps. *Geophysics*, 39:770–780, 1974.
- [74] Gelinsky, S. and Shapiro, S.A. Poroelastic velocity and attenuation in media with anisotropic permeability. *Society of Exploration Geophysicists, Extended Abstracts, 64th Annual International SEG Meeting*, pages 818–821, 1994.
- [75] Gelinsky, S. and Shapiro, S.A. Sh-wave velocity and attenuation in poroelastic transverse isotropic media. *56th Mtg., EAEG, Vienna, Austria*, page P114, 1994.
- [76] R.E. Geyer. Catalog of klauder wavelets. Seismograph Service Corporation, Tulsa, OK, 1972.
- [77] Dennis C. Ghiglia and Pritt, Mark D. *Two-Dimensional Phase Unwrapping Theory, Algorithms, and Software*. John Wiley and Sons, Inc, 1998. 493 pp.

- [78] Gladwin, M.T. and Stacey, F.D. Anelastic degradation of acoustic pulses in rock. *Physics of the Earth and Planetary Interiors*, 8:332–336, 1974.
- [79] Godfrey, L.M. and Stewart, J.D. and Schweiger, F. Application of dinoseis in canada. *Geophysics*, 33:65–77, 1968.
- [80] M.J. Gresko. *Analysis and Interpretation of Compressional (P-wave) and Shear (SH-wave) Reflection Seismic and Geologic Data over the Bane Dome, Giles County, Va.* Ph.d. dissertation in geophysics, Virginia Tech, 1985.
- [81] Gresko, M.J. and Costain, J.K. Compressional (p-wave) and shear (sh-wave) reflection seismic case history over the bane dome, giles county, virginia. *Society of Exploration Geophysicists, Extended Abstracts, 55th Annual International SEG Meeting*, pages 339–402, 1985.
- [82] B. Gutenberg. *Physics of the Earth's Interior*. Academic Press, 1959.
- [83] Hale, D., Hill, R. and Stefani, J. Imaging salt with turning seismic waves. *Geophysics*, 57:1453–1462, 1992.
- [84] N. Haskell. The dispersion of surface waves in multilayered media. *Bulletin of the Seismological Society of America*, 43:17–34, 1953.
- [85] Klaus Helbig. Fifty years of amplitude control. *Geophysics*, 63:750–762, 1998.
- [86] Herkenhoff and Ostrander. Effect of elliptical velocity anisotropy on surface velocity measurements. *Geophysics*, 1973.
- [87] Hubbard, S.S., Çoruh C. and Costain, J.K. Paleozoic and grenvillian structures in the southern appalachians: Extended interpretation from seismic reflection data. *Tectonics*, 10:141–170, 1991.
- [88] Personal TeX, Inc. *PCTeX for Windows*. Personal TeX, Inc., 2004. Version 5.0.
- [89] Jenkins, G.M. and Watts, D.G. *Spectral Analysis and its Applications*. Holden-Day, 1968. 525 pp.
- [90] Kalachand, S. and Kaila, K.L. Inversion of wide-angle seismic reflection times with damped least-squares. *Geophysics*, 59:1735–1744, 1994.
- [91] F. Kalisvaart and A.J. Sheriff. Correction to the paper, “water reverberations—their nature and elimination” by m.m. backus. *Geophysics*, 26:242, 1961.
- [92] Kallweit,R.S. and Wood, L.C. The limits of resolution of zero-phase wavelets. *Geophysics*, 47:1035–1046, 1982.

- [93] John H. Karl. *An Introduction to Digital Signal Processing*. Academic Press, Inc., 1989. 341 pp.
- [94] E. Kjartansson. Constant q wave propagation and attenuation. *Journal of Geophysical Research*, 84:4737–4748, 1979.
- [95] Klauder, J.R., Price, A.C., Darlington, S. and Albersheim, W.J. The theory and design of chirp radars. Technical report, Bell System Tech. J. 39, 1960. 745–808.
- [96] Klimentos, T. and McCann, C. Relationships among compressional wave attenuation, porosity, clay content, and permeability in sandstones. *Geophysics*, 55:998–1014, 1990.
- [97] L. Knopoff. Q. *Reviews of Geophysics*, 2:625–660, 1964.
- [98] Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1994. 509 pp.
- [99] F. Koehler. Deterministic deringing of seismic traces. Seismic Computing Corporation, Internal Report, 1964.
- [100] T.M. Kolich. Seismic reflection and refraction studies in the folded valley and ridge province at price mountain, montgomery county, virginia. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1974. 141 pp.
- [101] L.H. Koopmans. *The SDpectral Analysis of Time Series*. Academic Press, 1974. 366 pp.
- [102] Kootsookos, P.J., Lovell, P.C. and Boashash, B. A unified approach to the stft, tfds and instantaneous frequency. *IEEE Transactions on Signal Processing*, 40:1971–1982, 1992.
- [103] Leslie Lamport. *LATeX, A Document Preparation System, User's Guide and Reference Manual*. Addison-Wesley, 1994. 272 pp.
- [104] Lampshire, L.D., Çoruh, C. and Costain, J.K. Crustal structures and the eastern extent of the lower paleozoic shelf strata within the central appalachians: A seismic reflection interpretation. *Geological Society of America Bulletin*, 106:1–18, 1994.
- [105] R. W. Lankston. The seismic refraction method: A viable tool for mapping shallow targets into the 1990s. *Geophysics*, 54:1535–1542, 1989.
- [106] K. J. Laughlin. Interpretation of refraction and reflection stack data over the brevard fault zone in south carolina. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1988. 42 pp.
- [107] Lee Lawyer. From the other side. *The Leading Edge*, page 820, 2002.

- [108] Y.W. Lee. *Statistical Theory of Communication*. John Wiley and Sons, 1960. 509 pp.
- [109] F.K. Levin. Apparent velocity from dipping interface reflections. *Geophysics*, 36:510–516, 1971.
- [110] F.K. Levin. The unreasonable success of reflection seismology. *The Leading Edge*, 3:16–18, 1989.
- [111] N. Levinson. The wiener rms (root mean square) error criterion in filter design and prediction. *J. Math. Phys.*, 25:261–278, 1946.
- [112] J.P. Lindsey. Measuring wavelet phase from seismic data. *The Leading Edge*, 7:10–16, 1988.
- [113] J.P. Lindsey. Crooked lines and taboo places: What are the rules that govern good line layout? *The Leading Edge*, pages 74–77, 1991.
- [114] J.P. Lindsey. Seismic sources i have known. *The Leading Edge*, 10:47–48, 1991.
- [115] Lovell, B.C., Kootsookos, P.J. and Williamson, R.C. The circular nature of discrete-time frequency estimates. volume 3 of *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 3369–3372. Toronto, 1991.
- [116] Marangakis, A., Costain, J.K. and Çoruh, C. Use of integrated energy spectra for thin-layer recognition. *Geophysics*, 50:495–500, 1985.
- [117] D. Marsden. Static corrections—a review. parts i, ii, and iii. *The Leading Edge*, pages 43–49, 115–120, 210–216, 1993.
- [118] McDonal, F.J., Angona, F.A., Mills, R.L., Sengbush, R.L., Van Nostrand, R.G. and White, J.E. Attenuation of shear and compressional waves in pierre shale. *Geophysics*, 23:421–439, 1958.
- [119] Mitra, S.K. and Kaiser, J.F., editors. *Handbook for Digital Signal Processing*. John Wiley, New York, NY, 1993.
- [120] H.M. Mooney. Pole-and-zero design of digital filters. *Geophysics*, pages 354–360, 1968.
- [121] N.S. Neidell. Deterministic deconvolution operators –3 point or 4 point? *Geophysics*, 37:1039–1042, 1972.
- [122] N.S. Neidell. Could the processed seismic wavelet be simpler than we think? *Geophysics*, 56:681–690, 1991.
- [123] Neidell, N.S. and Taner, M.T. Semblance and other coherency measures for multichannel data. *Geophysics*, 36:482–497, 1971.

- [124] L.L. Nettleton. *Geophysical Prospecting for Oil*. McGraw-Hill Book Company, Inc., New York, 1940. 444 pages.
- [125] B.J. Newman. The vibroseis exploration method, a processor's perspective. *The Leading Edge*, 13:664–668, 1994.
- [126] P.N.S. O'Brien. A note on the reflection of seismic pulses with application to second event refraction shooting. *Geophysical Prospecting Eur. Assn. Geosci. Eng.*, 9, 1963.
- [127] O'Doherty, R.F. and Anstey, N.A. Reflections on amplitudes. *Geophysical Prospecting*, 19:430–458, 1971.
- [128] Oppenheim, A.V. and Schaffer, R.W. *Digital Signal Processing*. Prentice Hall, 1975. 585 pages.
- [129] Oppenheim, A.V. and Schaffer, R.W. *Discrete-Time Signal Processing*. Prentice Hall, 1989. 879 pp.
- [130] W.J. Ostrander. Plane-wave reflection coefficients for gas sands at nonnormal angles of incidence. In *52nd Annual International Meeting Expanded Abstracts*, pages 216–220. Society of Exploration Geophysicists, 9999.
- [131] D. Palmer. Production seismology. In Helbig, K. and Treitel, S., editors, *Refraction Seismics: the lateral resolution of structure and seismic velocity*, page 269 pp. Geophysical Press, London, 1986.
- [132] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill, New York, 1962.
- [133] Parks, T.W. and McClellan, J.H. A program for the design of linear phase finite impulse response filters. *IEEE Trans. Audio Electroacoust*, AU-20(3):195–199, 1972.
- [134] Peacock, K.L. and Treitel, S. Predictive deconvolution: theory and practice. *Geophysics*, 34(2):155–165, 1969.
- [135] Sam Peavy. *An Integrated Geophysical Study of the Central Appalachians of Western Virginia and Eastern West Virginia*. Ph.d. dissertation in geophysics, Virginia Tech, 1997.
- [136] Peterson, R.A. and Fillippone, W.R. and Coker, F.B. The synthesis of seismograms from well log data. *Geophysics*, 20:516–538, 1955.
- [137] T.C. Poulter. The poulter seismic method. *Geophysics*, 15:181–197, 1950.
- [138] T.L. Pratt. *A Geophysical Study of the Crust in Central Virginia with Implications for the Lower Crust and Appalachian Structures*. Ph.d. dissertation in geophysics, Virginia Tech, 1986.

- [139] Pratt, T.L. and Çoruh, C., Costain, J.K., Glover, III, L. A geophysical study of the earth's crust in central virginia: Implications for appalachian crustal structure. *Journal of Geophysical Research*, 93:6649–6667, 1988.
- [140] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. *Numerical Recipes in C*. Cambridge University Press, 1988. 735 pp.
- [141] Press, William H. and Teukolsky, W.T., Vetterling, W.T., Flannery, B.P. *Numerical Recipes in C: the Art of Scientific Computing*. The Press Syndicate of the University of Cambridge, 1992. 994 pp.
- [142] W.C. Pritchard. Why waste money with linear sweeps? *The Leading Edge*, 13:943–948, 1994.
- [143] Reilly, J.M. and Hobbs, R. Evaluation of source signature estimation technology. ExxonMobil Exploration Company and Bullard Labs, Cambridge University, 2001.
- [144] N. Ricker. The form and laws of propagation of seismic wavelets. *Geophysics*, 18:10–40, 1953.
- [145] Enders A. Robinson. *Predictive Decomposition of Time Series with Applications to Seismic Exploration*. Ph.d. thesis, MIT, Cambridge, MA, 1954.
- [146] Enders A. Robinson. Predictive decomposition of seismic traces. *Geophysics*, 22:767–778, 1957.
- [147] Enders A. Robinson. Multi-channel z-transforms and minimum-delay. *Geophysics*, 31:482–500, 1966.
- [148] Enders A. Robinson. *Multichannel Time Series Analysis With Digital Computer Programs*. Holden-Day, San Francisco, 1967. 298 pp.
- [149] Enders A. Robinson. Predictive decomposition of time series with applications to seismic exploration. *Geophysics*, 32:418–484, 1967.
- [150] Enders A. Robinson. *Statistical Communication and Detection with special reference to Digital Data Processing of Radar and Seismic Signals*. Hafner Publishing Co., New York, 362 pp, 1967.
- [151] Enders A. Robinson. Section 1. seismic exploration. In *Seismic Inversion and Deconvolution. Volume 4A. Classical Methods*. Handbook of Geophysical Exploration, Geophysical Press, London – Amsterdam, 1984. 349 pp.
- [152] Robinson, Enders A. and Osman, Osman M., editors. *Deconvolution 2*. Number 17 in GeophysicsReprint Series. Society of Exploration Geophysicists, 1996. Series Editor Daniel A. Ebrom.
- [153] Robinson, Enders A. and Treitel, Sven. *Geophysical Signal Analysis*. Prentice-Hall, 1980. 466 pp.

- [154] Saha, J., Rao, N. and Agrawal, M. On: "complex seismic trace analysis" by m.t. taner, f. koehler, and r.e. sheriff (geophysics, june 1979, p. 1041-1063). *Geophysics*, 1980.
- [155] W.A. Schneider. *Wavelet Processing: in The Stationary Convolutional Model of the Reflection Seismogram*. Society of Exploration Geophysicists, Tulsa, OK, 1977.
- [156] Schoenberger, M. and Levin, F. Apparent attenuation due to intrabed multiples. *Geophysics*, 39:278–291, 1974.
- [157] S.M. Scott. The reprocessing and extended interpretation of seismic reflection data recorded over the hayesville-fries thrust sheet in southwestern north carolina. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1987. 90 pp.
- [158] O. Selvi. The generalized reciprocal method (grm) - a new approach without first break picking. *European Association of Exploration Geophysicists (EAEG), 55th Meeting and Technical Session, June 7-11, Stavanger, Norway*, 1993.
- [159] A.K. Sen. Removing near-surface effects in seismic data: Application for determination of faults in the coastal plain sediments. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1991. 98 pp.
- [160] R.L. Sengbush. *Seismic Exploration Methods*. International Human Resources Development Corporation (IHRDC), 1983. 296 pp.
- [161] Sengbush, R.L., Lawrence, P.L. and McDonal, F.J. Interpretation of synthetic seismograms. *Geophysics*, 26:138–157, 1961.
- [162] Shah, P.M. and Levin, F.K. Gross properties of time-distance curves. *Geophysics*, 38:643–656, 1973.
- [163] Sheriff, R.E. and Geldart, L.P. Data-processing and interpretation. In *Data-processing and interpretation*, volume 2 of *Exploration Seismology*, page 221. Cambridge University Press, 1983.
- [164] Silvia, M.T. and Robinson, E.A. *Deconvolution of Geophysical Time Series in the Exploration for Oil and Natural Gas*. Elsevier, 1979. 251 pp.
- [165] Singh, P. and Thompson, W. Exploring the complex plane: Green's functions, hilbert transforms, and analytic continuation. *Computers in Physics*, 7(4):388–392, 1993.
- [166] D. Slepian. On bandwidth. *Proc. IEEE*, 64(3):292–300, 1976.
- [167] M.M. Slotnick. *Lessons in Seismic Computing*. Society of Exploration Geophysics, 1986.

- [168] K. Steiglitz. *A Digital Signal Processing Primer*. Addison-Wesley, 1996. 314 pp.
- [169] Steiglitz, K., Parks, T.W. and Kaiser, J.F. Meteor: A constraint-based fir filter design program. *IEEE Trans. Signal Processing*, 40(8):1901–1909, 1992.
- [170] Stolt, R.H. and Benson, A.K. Section 1. seismic exploration. In Klaus Helbig and Sven Treitel, editors, *Seismic Migration—Theory and Practice*, volume 5 of *Handbook of Geophysical Exploration*, page 382. Geophysical Press, 1986.
- [171] E. Strick. The determination of q, dynamic viscosity, and transient creep curves from wave propagation measurements. *Geophys. J.R. Astr. Soc.*, 13:197–218, 1967.
- [172] E. Strick. A predicted pedestal effect for pulse propagation in constant-q solids. *Geophysics*, 35:387–403, 1970.
- [173] E. Strick. Discussion on “proposed attenuation-dispersion pair for seismic waves” by j.e. white and d.j. walsh. *Geophysics*, pages 423–429, 1973.
- [174] A.K. Susskind. *Notes on Analog-Digital Conversion Techniques*. Chapman and Hall, 1957.
- [175] Taner, M.T. and Koehler, F. Velocity spectra -digital computer derivation and applications of velocity functions. *Geophysics*, 34:859–881, 1969.
- [176] Taner, M.T. and Koehler, F. Velocity spectra -digital computer derivation and applications of velocity functions: Erratum. *Geophysics*, 36:787, 1971.
- [177] Taner, M.T. and Koehler, F. Surface consistent corrections. *Geophysics*, 46:17–22, 1981.
- [178] Taner, M.T. and Sheriff, R.E. Application of amplitude, frequency, and other attributes to stratigraphic and hydrocarbon determination. In C. E. Payton, editor, *Seismic Stratigraphy—Applications to Hydrocarbon Exploration*, AAPG Memoir 26, pages 301–328. American Association of Petroleum Geologists, Tulsa, 1977.
- [179] Taner, M.T., Koehler, F. and Sheriff, R.E. Complex seismic trace analysis. *Geophysics*, 44:1041–1063, 1979.
- [180] Taner, M.T., Matsuoka, T., Baysal, E., Lu, L. and Yilmaz, O. Imaging with refractive seismic waves. *Society of Exploration Geophysics 62nd International Annual Meeting, Expanded Abstracts*, pages 1132–1135, 1992.
- [181] Telford, W.M., Geldart, L.P. and Sheriff, R.E. *Applied Geophysics*. Cambridge University Press, second edition, 1990. 770 pp.

- [182] W.T. Thompson. Transmission of elastic waves through a stratified solid. *Journal of Applied Physics*, 21:89–93, 1950.
- [183] Toksös, M.N., Johnston, D.H. and Timur, A. Attenuation of seismic waves in dry and saturated rocks: I. laboratory measurements. *Geophysics*, 44:681–690, 1979.
- [184] Sven Treitel. Principles of multichannel filtering. *Geophysics*, 35:785–811, 1970.
- [185] D.A. Vossler. *Anisotropic Media and the Determination of Subsurface Velocity by the Use of Surface Reflection Data*. Ph.d. dissertation in geophysics, Virginia Tech, 1971.
- [186] Waddell, M.G. and Domoracki, W.J., and Temples, T.J. Use of seismic reflection amplitude versus offset (avo) techniques to image dense nonaqueous phase liquids (dnapl). *Symposium on the Application of Geophysics to Engineering and Environmental Problems, SAGEEP 2001*, 2001.
- [187] K.H. Waters. *Reflection Seismology: A Tool for Energy Resource Exploration*. John Wiley and Sons, third edition, 1987. 538 pp.
- [188] K. Weisenburger. Reflection seismic data acquisition and processing for enhanced interpretation of high-resolution objectives. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1985. 69 pp.
- [189] E.W. Weisstein. Discontinuity *Mathematica* notebook, <http://mathworld.wolfram.com/discontinuity.html>. CRC Press LLC and Wolfram Research, 2003.
- [190] P.D. Wells. A seismic reflection and refraction study near the blue ridge thrust in virginia. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1975.
- [191] R.E. White. Reply to Peter Cary's comment on R.E. White's "the accuracy of estimating Q from seismic data". *The Leading Edge*, April:256–258, 1993.
- [192] White, J.E. and D.J. Walsh. Proposed attenuation-dispersion pair for seismic waves. *Geophysics*, 37(3):456–461, 1972.
- [193] White, R.E. and P.N.S. O'Brien. Estimation of the primary seismic pulse. *Geophysical Prospecting*, 22:627–651, 1974.
- [194] M.B. Widess. How thin is a thin bed? *Geophysics*, 38:1176–1180, 1973.
- [195] M.B. Widess. Quantifying resolving power of seismic systems. *Geophysics*, 47:1160–1173, 1982.

- [196] N. Wiener. *The Interpolation, Extrapolation, and Smoothing of Stationary Time Series*. Massachusetts Institute of Technology, reprinted in 1949 by John Wiley, New York, 1942.
- [197] R.C. Williamson. The statistical performance of some instantaneous frequency estimators. *IEEE Transactions on Signal Processing*, 40(7):1708–1723, 1992.
- [198] Stephen Wolfram. *The Mathematica Book*. Wolfram Media Corporation, 2003. 1488 pp.
- [199] D.W. Worthington. Some ultrasonic wave model studies. M.s. thesis in geophysics, Virginia Polytechnic Institute and State University, Blacksburg, 24061, 1969.
- [200] P.C. Wuenschel. Dispersive body waves - an experimental study. *Geophysics*, 30:539–551, 1965.
- [201] C.R. Wylie. *Advanced Engineering Mathematics*. McGraw-Hill Book Company, 1951. 640 pp.
- [202] Yantis, B.R. and Costain, J.K., and Ackermann, H.D. A reflection seismic study near charleston, sc. In G. Gohn, editor, *Professional Paper 1313, Studies Related to the Charleston, South Carolina, Earthquake of 1886 - Tectonics and Seismicity*. U.S. Geological Survey, 1983. pages G1–G20.
- [203] O. Yilmaz. Seismic data analysis. In Stephen M. Doherty, editor, *Processing, Inversion, and Interpretation of Seismic Data*, volume I and II of *Investigations in Geophysics*, page 2000. Society of Exploration Geophysicists, 1987.
- [204] O. Yilmaz. *Seismic Data Processing*. Society of Exploration Geophysicists, 1987. 526 pp.
- [205] Xianhuai Zhu. Survey parameters for imaging salt domes. *The Leading Edge*, pages 888–892, 1994. August.
- [206] K. Zoeppritz. ber reflexion und durchgang seismischer wellen durch unstetigkeitsflchen. *Nachrichten der Kniglichen Gesellschaft der Wissenschaften zu Gttingen*, pages 66–84, 1919.

Index

- absorption, 358, 518
 - must have dispersion too, 518
- absorption-dispersion pair, 519, 539
- acoustic impedance, 195, 216
- Adachi, R., 317
- aliasing, 86, 189
 - frequency domain, 189
 - how to determine if present, 193
 - numerical example
 - frequency domain, 190
 - principal alias, 192
 - time domain, 189
- amplitude
 - Fourier component, 186
- amplitude envelope
 - of a seismic trace, 243
- amplitude spectrum, 49
- analysis window, 78
- analytic function, 34
- analytic signal, 34, 142, 145, 243
- analytic trace, 146
- anisotropy, 218
- anticipation component of a filter, 378, 417, 439, 443
- aperiodic, 48
- attributes
 - amplitude envelope, 243
 - instantaneous frequency, 243
 - instantaneous phase, 243
- autocorrelation, 120
 - and the design of inverse filters, 407
- automatic volume control, 401
- averaging filter, 178
- AVO
 - amplitude variation with offset, 208
- and the Zoeppritz equations, 208
- movie, 208
- Backus water-layer wave guide, 394, 495
- Backus, M., 473, 474, 495, 496, 498, 505
- Bahorich, M., 121
- Belcher, S.W., 343
- binomial series, 380
- binomial theorem, 267
- blind zone
 - refraction, 317
- body wave dispersion, 159, 518
 - and Q
 - from time-domain, 545
- body wave dispersive velocity, 519
- Bryan, R.A., 223
- Butterworth filter, 181
- Cauchy principal value
 - not required, 158
- Cauchy-Riemann equations, 34
- causal function, 129
 - amplitude and phase, 514
- character, 399
- chirp signal, 451
- Claerbout, J., 293
- CMP gather, 209
- coherence cube, 121
- common midpoint gather, 328
- common refractor surface, 328
- complex number
 - polar form, 29
- complex numbers
 - branch, 33
 - principal angle, 33

- complex plane, 24
- complex seismic trace, 243
 - analytic signal, 142, 243
- computer programs
 - autocorrelation, 437
 - convolution, 438
 - crosscorrelation, 437
 - EUREKA, 38
 - FOLD, 174, 364, 493
 - Mathematica*, 365
 - fold, 116, 121, 385, 493
 - FT, 47, 88, 92, 375
 - FT Subroutine versus FFT, 92
- general inverse filter using Fourier transforms, 450
- Hilbert transform
 - frequency domain, 530
- Hilbert transform and minimum-delay spectrum, 515
- inverse
 - z*-transform, 387
- inverse filter, 382
- INVSMN, 417
- Neidell 2-layer reverberation model, 408
- NLOGN, 377
- quadrature seismic trace, 246
- reflection traveltime curves from a multilayer model, 262
- refraction travelttime curves, 303
- reshaping a vibroseis wavelet
 - using Fourier theory, 452
- rms velocities for n -layer model, 270
- spike out 3-point mixed-delay wavelet, 449
- spiking a vibroseis wavelet
 - by Fourier theory, 452
- unwrap phase, 246, 517
- wavelet
 - spike out, 385
- wavelet acf proportional to trace acf, 461
- constant-velocity panel, 293
- continuous reflection
 - how it changes wavelet shape, 225
- convergence
 - as a function of distance from unit circle, 382
- filter
 - root inside unit circle, 387
 - root on unit circle, 391
- convolution, 29, 111
 - frequency domain, 123
 - time domain, 111
- convolutional model, 373, 456, 485
- coordinate systems
 - Cartesian, 24
 - polar, 27
- cosine transform, 65, 87, 376
- couplet, 166
- critical angle, 294
- crosscorrelation, 120, 343
- cutoff frequency, 183
- deconvolution, 370
 - predictive, 454, 515
 - gap, 482
 - how to get a good, 497
 - second zero-crossing, 482
 - spiking
 - why?, 481
 - wavelet, 338
- deep-reflection reverberation spike train, 406
- delay time, 296, 317, 333
- Demirbag, E., 209
- DeMoivre's theorem, 30
- dependent variable, 41
- determinant, 35
 - difference between matrix, 37
 - element of, 36
 - Laplace evaluation, 36
 - order of, 36
 - square, 36
 - value of, 36
- DFS, 90
- differencing filter, 178
- differentiation
 - amplifies higher frequencies, 97

- digital signal, 45
- dip
 - effect on reflection traveltime, 286
 - effect on refraction traveltime, 305
- Dirac delta function, 106
- Dirichlet conditions, 64
- dispersion, 521
 - determination in the time-domain, 545
 - must have absorption too, 518
- Dix equation, 269, 272
- Domoracki, W.J., 351
- dynamic range, 47, 482
 - why important, 399
- Ecevitoglu, B., 518, 519, 533, 534, 538
 - dispersive phase velocity, 529
- equi-delay 2-term wavelet, 364
- error
 - normalized mean square, 423
- Euler coefficients, 55
- Euler formulas, 29
- even versus odd number of points, 92
- exponential functions
 - relations between trigonometric functions, 29
- Fast Fourier Transform, 89
 - advantage, 116
- Fermat's principle, 254
- Feynman, Richard, 28
- FFT, 90
 - even number of points, 91
 - odd number of points, 91
 - versus Subroutine FT, 92
- filter
 - anticipation component, 387, 442
 - averaging, 175
 - bandpass
 - design, 180
 - Butterworth, 181
 - differencing, 175
- general least-squares, 420
- instability, 379
- introduction, 175
- inverse
 - maximum-delay, 384
 - minimum-delay, 379
 - only one coefficient series for a given $f(t)$, 380
- low-pass, 182
- memory component, 382, 442
- notch
 - design, 178
- parallel filters, 181
- stopband, 182
- two-term, 176
- water layer, 396
 - frequency response, 396
- folding frequency, 189
- Fourier spectrum, true
 - cannot obtain, 125
- Fourier transform, 42, 85
 - direct, 85
 - inverse, 85, 87
 - pair, 85
 - relationship between periodic and aperiodic, 49, 84, 87
 - sign convention, 53, 163
- FT Subroutine versus FFT, 92
- Futterman, W.I., 519, 532–534
 - dispersive phase velocity, 532
- gapped deconvolution, 480, 482, 499
- gather, 325
- geometrical spreading, 520
- Gibbs phenomena, 64
- Gresko, M.J., 351
- ground velocity, 41
- hanging wavelets, 465
- harmonic series, 72
 - fundamental frequency, 72
- Hartley transform, 377
- head waves, 293, 305
- Helbig, K., 45
- high-resolution, 482
- Hilbert transform, 130

- analytic derivation
 frequency domain, 156
- by swapping real and imaginary
 parts of Fourier transform,
 152
- computational approach preferred
 over Equation (3.90), 148
- computer program, 153
- continuous data
 graphic display, 154
- continuous frequency
 even from odd, 157, 158
 odd from even, 159
- continuous time
 odd from even, 144
- definition, 129
- discrete data
 graphic display, 155
- discrete time
 even from odd, 137
 odd from even, 530
- fastest way to compute, 152
- frequency domain, 144
of a sinusoid, 139
principal value, 156
shared point, 137
- Hubbard, S.S., 352
- hyperbola, 270
- impulse response, 109
- independent variable, 41
- instantaneous frequency, 141, 223,
 243
- instantaneous phase, 141, 243
- intercept time, 298
- intrinsic damping, 95, 96, 358, 518
- inverse filter, 381
 Mathematica program for, 450
 anticipation component, 450
 memory component, 450
 vibroseis wavelet, 451
- inverse filtering, 370
- inverse Fourier transform, 55
- inverse problem
 simple example, 117
- Klauder wavelet, 343, 355
- Koehler, F., 268
- Kolich, T., 312, 314, 339
- lag
 autocorrelation, 121
 crosscorrelation, 120
 prediction distance, 455
- Lagrange multipliers, 264
- Lampshire, L.D., 18
- law of cosines, 286
- Lawrence, P.L., 217
- leakage, 125, 186, 350
 side lobes, 187
 window, 78
- least-squares method, 420
- least-time path, 254
- Lee, Y.W., 156
- Levin's dip correction, 288
- Levin, F., 215
- Levinson, N., 38
- Toeplitz matrix, 478
- linear system, 24
 definition, 109
 impulse response, 109
- Maclaurin series, 28, 380
- Marangakis, A., 223
- matrix, 37
 difference between determinant,
 37
 square, 37
 symmetric, 37
- maximum-delay, 166, 363
- McDonald, F.J., 217
- memory component of a filter, 378,
 382, 417, 443
- minimum phase, 166
- minimum-delay, 156, 166, 169, 171,
 363
 first coefficient not necessarily
 the largest, 169, 172, 364
 functions with more than two
 terms, 167
- minimum-phase, 166, 169, 171
- minimum-phase-lag, 367

- mixed-delay, 167, 364
- mixed-phase, 364
- Mooney, H.M., 181
- multiples
 - long-period, 495, 497
 - short-period, 497
- Neidell, N., 408
- non-periodic, 48
- normal equations, 38, 424
- normal moveout, 258, 287
 - application of correction for, 290
 - wavelet stretching, 290
- normal moveout correction, 269
- numbers, 24
 - complex, 24
 - absolute value, 26
 - argument, 27
 - complex plane, 24
 - conjugate, 26
 - graphical representation, 24
 - imaginary part, 24
 - magnitude, 26
 - modulus, 27
 - phase of, 27, 34
 - polar form, 27
 - real part, 24
 - rules for manipulation, 25
- imaginary, 25
- real, 24, 25
 - logarithm of negative real number, 33
- Nyquist frequency, 189
- odd versus even number of points, 92
- Oppenheim, A.V., 45
- order
 - of Butterworth filter, 183
- p, 266
- parametric travel-time equations, 266
- partial fractions, 414
- Peacock, K.L., 470
- Peavy, S., 17
- periodic, 48
- Peterson model, 217
- phase, 34
 - from amplitude spectrum, 515
- phase angle, 27
- phase lag versus phase, 97
- phase spectrum, 49
- phase versus phase lag, 97
- phasor, 25, 27, 34, 132
 - elements of, 34
- pilot sweep, 121, 342
- plane wave reflection coefficient, 198
- polar coordinates, 27
- polar form of a complex number, 29
- Pratt, T.L., 18
- prediction error filter, 468, 470
- prediction filter, 464
- predictive decomposition, 454
- predictive deconvolution, 38, 454
 - and mixed-delay wavelets, 484
 - consequences, 481
 - gap, 455
 - general idea behind, 455
 - is a wavelet truncation process, 480
 - lag, 455
 - of a seismic trace, 485
 - of a wavelet, 475
 - prediction distance, 455
 - prediction error operator, 470
 - spiking, 463
- principal root, 30
- principle of reciprocity, 311
- Q, quality factor, 518
 - and dispersive velocities from discrete Hilbert transform, 529
 - and dispersive velocities from Futterman's approximation, 532
 - and impulse response from Gladwin and Stacey, 541
 - time-domain determination of, 545
- quadrature filter, 151
- quality factor, 518

- ray parameter, 259, 260, 266, 275
 measurement of, 259
- reciprocity, 311
- record section, 15, 325
- record sections, 253
- reflection coefficient
 complex
 physical significance, 207
 high value, 199
 plane wave, 198
 table of typical values, 201
 typical value, 199
 water-bottom, 495
- reflectivity function, 216, 218, 219,
 485
 random, 496
- refraction
 Adachi, 317
 blind zone, 317
 delay time, 298
 arbitrary dips, 317
 intercept time, 297
 principle of reciprocity, 311
 traveltime curves, 293
 traveltime equations, 319
 arbitrary dips, 317
- refraction moveout, 328
- refraction stacks, 325
- refraction traveltimes, 293
- resolution
 ultimate, 338
- reversed refraction profile, 307, 308
- Ricker wavelet, 232, 355
- ringing, 497
- rms velocity, 262
- Robinson, E.A., 23, 47, 121, 454,
 456, 461, 475, 493, 498, 499
- root
 inside unit circle, 172
 of a polynomial, 167
 on unit circle, 368
 outside unit circle, 170
- SAGC, 507
- sampling theorem, 189
- Schafer, R.W, 45
- seismic attributes, 141, 151
 analytic signal, 243
 quadrature trace, 142, 243
- seismic record section, 325
- seismic trace
 analytic signal, 243
 complex, 243
 generate synthetic, 392
 reflection strength, 244
- semblance, 121
- Sengbush, R.L., 202, 217
- Shannon's theorem, 189
- side lobes, 83, 344
- sign convention, 52, 140, 244
- signals, 41
 analog, 42
 band-limited, 189
 continuous, 42
 discrete, 41
 ground displacement, 41
- sine transform, 65, 87, 376
- slowness, 266
- smoothing, 175
- Snell's law, 254
 and LaGrangian multipliers, 266
 physical significance, 254
- sonic log, 202, 218
- spectrum
 amplitude, 49
 phase, 49
- spiking deconvolution, 480
- stacking velocity, 272, 293
- statics, 337
- stationary time series, 456
- Stretched automatic amplitude ad-
justment of seismic data,
 507
- synthetic seismogram, 214, 217
- Taner, M.T., 268
- Taylor series, 380
- temperature log
 advantages of a precision, 104
- There can be only one, 380, 448,
 461, 467

- z*-transform versus predictive deconvolution, 497
- predictive deconvolution, 475
- spiking out a wavelet, 481
- thin bed, 231, 520
 - integrated energy spectra, 223
- Thumper energy source, 352
- time domain, 41
- time-advanced
 - meaning in predictive deconvolution, 465
- time-convolution theorem, 111
- time-shifting theorem, 93, 527
- Toeplitz matrix, 38
- transmission coefficient
 - two-way, 200
- traveltime curve, 253
 - slope of, 259
- Treitel, S., 23, 456, 461, 470, 475, 498
- trigonometric functions
 - relations between exponential functions, 29
- tuning, 231
- turning wave, 325
- unit circle
 - root on, 368
 - meaning of, 382
- unwrapped phase angles, 517
- velocity
 - apparent, 257
 - average, 263
 - horizontal, 253
 - rms, 266, 269
 - n-layer model, 270
 - stacking, 272, 293
- velocity function
 - of Sengbush et al. [161], 219
- velocity log, 218
- velocity spectra, 272
 - coherence, 272
- vibroseis
 - crosscorrelation, 343
- data degraded by Karst topography, 346
- method, 342
- pilot sweep, 121
- single versus multiple vibrators, 343
- sweep, 342
- tapered sweep, 346
- wavelet, 451
- Virginia Tech seismic crew
 - data acquisition parameters, 111, 452
- Vossler, D., 217
- water layer
 - filter, 396
- water-confined reverberation spike train, 399, 402, 495
- wave guide, 394
 - all layers are wave guides, 398
- wavelet, 338
 - 2-term, 364, 378
 - constraint on spectrum for no change in shape, 95
 - couplet, 378
 - distortion
 - by critical angle, 208
 - by Q , 519
 - by differentiation (thin bed), 221
 - by integration, 225
 - effective source, 454
 - equi-delay
 - one that never decays, 389
 - hanging, 465
 - maximum-delay, 363
 - minimum-delay, 358, 363
 - coefficients, 479
 - minimum-delay from Hilbert transform, 515
 - mixed-delay, 364
 - one that never decays, 382
 - partial energy, 369
 - physical requirements for real, 359
 - physically realizable, 368

roots, 360
shaping, 418
spike out 2-point
 max-delay, expand around ∞ ,
 387
 min-delay, expand around 0,
 385
 zero mean, 338
Weisenburger, K., 20
wide-angle shooting, 305
wiggly trace, 15, 40
window leakage, 78
Wuenschel, P., 532, 540, 544, 545

z-plane, 164, 169
 definition, 53
 sign convention for z , 53
z-transform, 53, 163, 367
 definition, 164
 sign convention, 163
zero of a polynomial, 167
Zoeppritz equations, 35, 204
 CREWES Zoeppritz Explorer,
 206
 in the time-offset domain, 208

This page is intentionally left blank