Appendix A

The PREM Model

For many years the most widely used 1-D model of Earth's seismic velocities has been the Preliminary Reference Earth Model (PREM) of Dziewonski and Anderson (1981). This model was designed to fit a variety of different data sets, including free oscillation center frequency measurements, surface wave dispersion observations, travel time data for a number of body wave phases, and basic astronomical data (Earth's radius, mass, and moment of inertia). In addition to profiling the P and S velocities, PREM specifies density and attenuation as functions of depth. Although these parameters are known less precisely than the seismic velocities, including them is important because it makes the model complete and suitable for use as a reference to compute synthetic seismograms without requiring additional assumptions. In order to simultaneously fit Love and Rayleigh wave observations, PREM is transversely isotropic between 80 and 220 km depth in the upper mantle. This is a spherically symmetric form of anisotropy in which SH and SV waves travel at different speeds. For simplicity, the table here lists only values from an isotropic version of PREM. The true PREM model is also specified in terms of polynomials between node points; linear interpolation between the 100 km spacing of values in this table will produce only approximate results. All current Earth models have values that are reasonably close to PREM; the largest differences are in the upper mantle, where, for example, a discontinuity at 220 km is not found in most models.

Table A.1: Preliminary Reference Earth Model (isotropic version)

Depth (km)	Radius (km)	V_p (km/s)	V_s (km/s)	ρ (g/cc)	Q_μ	Q_κ	P (GPa)
0.0	6371.0	1.45	0.00	1.02	0.0	57823.0	0.0
3.0	6368.0	1.45	0.00	1.02	0.0	57823.0	0.0
3.0	6368.0	5.80	3.20	2.60	600.0	57823.0	0.0
15.0	6356.0	5.80	3.20	2.60	600.0	57823.0	0.3
15.0	6356.0	6.80	3.90	2.90	600.0	57823.0	0.3
24.4	6346.6	6.80	3.90	2.90	600.0	57823.0	0.6
24.4	6346.6	8.11	4.49	3.38	600.0	57823.0	0.6
71.0	6300.0	8.08	4.47	3.38	600.0	57823.0	2.2
80.0	6291.0	8.08	4.47	3.37	600.0	57823.0	2.5
80.0	6291.0	8.08	4.47	3.37	80.0	57823.0	2.5
171.0	6200.0	8.02	4.44	3.36	80.0	57823.0	5.5
220.0	6151.0	7.99	4.42	3.36	80.0	57823.0	7.1
220.0	6151.0	8.56	4.64	3.44	143.0	57823.0	7.1
271.0	6100.0	8.66	4.68	3.47	143.0	57823.0	8.9
371.0	6000.0	8.85	4.75	3.53	143.0	57823.0	12.3
400.0	5971.0	8.91	4.77	3.54	143.0	57823.0	13.4
400.0	5971.0	9.13	4.93	3.72	143.0	57823.0	13.4
471.0	5900.0	9.50	5.14	3.81	143.0	57823.0	16.0
571.0	5800.0	10.01	5.43	3.94	143.0	57823.0	19.9
600.0	5771.0	10.16	5.52	3.98	143.0	57823.0	21.0
600.0	5771.0	10.16	5.52	3.98	143.0	57823.0	21.0
670.0	5701.0	10.27	5.57	3.99	143.0	57823.0	23.8
670.0	5701.0	10.75	5.95	4.38	312.0	57823.0	23.8
771.0	5600.0	11.07	6.24	4.44	312.0	57823.0	28.3
871.0	5500.0	11.24	6.31	4.50	312.0	57823.0	32.8
971.0	5400.0	11.42	6.38	4.56	312.0	57823.0	37.3
1071.0	5300.0	11.58	6.44	4.62	312.0	57823.0	41.9
1171.0	5200.0	11.73	6.50	4.68	312.0	57823.0	46.5
1271.0	5100.0	11.88	6.56	4.73	312.0	57823.0	51.2
1371.0	5000.0	12.02	6.62	4.79	312.0	57823.0	55.9
1471.0	4900.0	12.16	6.67	4.84	312.0	57823.0	60.7
1571.0	4800.0	12.29	6.73	4.90	312.0	57823.0	65.5
1671.0	4700.0	12.42	6.78	4.95	312.0	57823.0	70.4
1771.0	4600.0	12.54	6.83	5.00	312.0	57823.0	75.4
1871.0	4500.0	12.67	6.87	5.05	312.0	57823.0	80.4
1971.0	4400.0	12.78	6.92	5.11	312.0	57823.0	85.5
2071.0	4300.0	12.90	6.97	5.16	312.0	57823.0	90.6
2171.0	4200.0	13.02	7.01	5.21	312.0	57823.0	95.8
2271.0	4100.0	13.13	7.06	5.26	312.0	57823.0	101.1
2371.0	4000.0	13.25	7.10	5.31	312.0	57823.0	106.4
2471.0	3900.0	13.36	7.14	5.36	312.0	57823.0	111.9
2571.0	3800.0	13.48	7.19	5.41	312.0	57823.0	117.4
2671.0	3700.0	13.60	7.23	5.46	312.0	57823.0	123.0

Depth (km)	Radius (km)	V_p (km/s)	V_s (km/s)	ρ (g/cc)	Q_μ	Q_κ	P (GPa)
2741.0	3630.0	13.68	7.27	5.49	312.0	57823.0	127.0
2771.0	3600.0	13.69	7.27	5.51	312.0	57823.0	128.8
2871.0	3500.0	13.71	7.26	5.56	312.0	57823.0	134.6
2891.0	3480.0	13.72	7.26	5.57	312.0	57823.0	135.8
2891.0	3480.0	8.06	0.00	9.90	0.0	57823.0	135.8
2971.0	3400.0	8.20	0.00	10.03	0.0	57823.0	144.2
3071.0	3300.0	8.36	0.00	10.18	0.0	57823.0	154.8
3171.0	3200.0	8.51	0.00	10.33	0.0	57823.0	165.2
3271.0	3100.0	8.66	0.00	10.47	0.0	57823.0	175.5
3371.0	3000.0	8.80	0.00	10.60	0.0	57823.0	185.7
3471.0	2900.0	8.93	0.00	10.73	0.0	57823.0	195.8
3571.0	2800.0	9.05	0.00	10.85	0.0	57823.0	205.7
3671.0	2700.0	9.17	0.00	10.97	0.0	57823.0	215.4
3771.0	2600.0	9.28	0.00	11.08	0.0	57823.0	224.9
3871.0	2500.0	9.38	0.00	11.19	0.0	57823.0	234.2
3971.0	2400.0	9.48	0.00	11.29	0.0	57823.0	243.3
4071.0	2300.0	9.58	0.00	11.39	0.0	57823.0	252.2
4171.0	2200.0	9.67	0.00	11.48	0.0	57823.0	260.8
4271.0	2100.0	9.75	0.00	11.57	0.0	57823.0	269.1
4371.0	2000.0	9.84	0.00	11.65	0.0	57823.0	277.1
4471.0	1900.0	9.91	0.00	11.73	0.0	57823.0	284.9
4571.0	1800.0	9.99	0.00	11.81	0.0	57823.0	292.3
4671.0	1700.0	10.06	0.00	11.88	0.0	57823.0	299.5
4771.0	1600.0	10.12	0.00	11.95	0.0	57823.0	306.2
4871.0	1500.0	10.19	0.00	12.01	0.0	57823.0	312.7
4971.0	1400.0	10.25	0.00	12.07	0.0	57823.0	318.9
5071.0	1300.0	10.31	0.00	12.12	0.0	57823.0	324.7
5149.5	1221.5	10.36	0.00	12.17	0.0	57823.0	329.0
5149.5	1221.5	11.03	3.50	12.76	84.6	1327.7	329.0
5171.0	1200.0	11.04	3.51	12.77	84.6	1327.7	330.2
5271.0	1100.0	11.07	3.54	12.82	84.6	1327.7	335.5
5371.0	1000.0	11.11	3.56	12.87	84.6	1327.7	340.4
5471.0	900.0	11.14	3.58	12.91	84.6	1327.7	344.8
5571.0	800.0	11.16	3.60	12.95	84.6	1327.7	348.8
5671.0	700.0	11.19	3.61	12.98	84.6	1327.7	352.3
5771.0	600.0	11.21	3.63	13.01	84.6	1327.7	355.4
5871.0	500.0	11.22	3.64	13.03	84.6	1327.7	358.0
5971.0	400.0	11.24	3.65	13.05	84.6	1327.7	360.2
6071.0	300.0	11.25	3.66	13.07	84.6	1327.7	361.8
6171.0	200.0	11.26	3.66	13.08	84.6	1327.7	363.0
6271.0	100.0	11.26	3.67	13.09	84.6	1327.7	363.7
6371.0	0.0	11.26	3.67	13.09	84.6	1327.7	364.0

Appendix B

Math Review

This appendix is not intended to teach anyone vector calculus or complex number theory, but simply to list some of the important definitions to assist those whose skills may have become rusty. Many of the equations are expressed in both standard vector notation and the index notation used in this book.

B.1 Vector Calculus

Consider a Cartesian coordinate system with x, y, and z axes. The length, or magnitude, of a vector $\mathbf{u}$ is written as $\|\mathbf{u}\|$. A vector may be expressed in terms of its components as

$$\mathbf{u} = u_x\hat{\mathbf{x}} + u_y\hat{\mathbf{y}} + u_z\hat{\mathbf{z}}, \tag{B.1}$$

where $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ are unit length vectors in the x, y, and z directions. The dot product of two vectors is a scalar (a single number) and is defined as

$$\begin{aligned} \lambda = \mathbf{u}\cdot\mathbf{v} &= \|\mathbf{u}\|\|\mathbf{v}\|\cos\theta & \text{(B.2)}\\ &= u_x v_x + u_y v_y + u_z v_z, & \text{(B.3)} \end{aligned}$$

where θ is the angle between the two vectors. It follows that $\mathbf{u}\cdot\mathbf{v} = 0$ when $\mathbf{u}$ and $\mathbf{v}$ are orthogonal and that for unit vectors $\hat{\mathbf{x}}\cdot\hat{\mathbf{x}} = 1$. Note that for a unit vector $\hat{\mathbf{u}}$, the dot product $\hat{\mathbf{u}}\cdot\mathbf{v}$ gives the length of the orthogonal projection of $\mathbf{v}$ onto $\hat{\mathbf{u}}$.

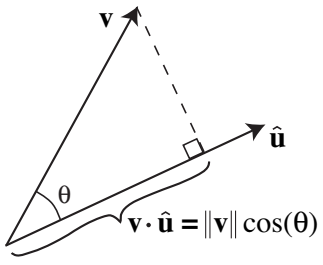


Figure B.1: The dot product of a vector with a unit vector is the length of the projection onto the unit vector.

The cross-product between two vectors is a third vector that points in a direction perpendicular to both (according to the right-hand rule). The cross-product can be expressed in component form as

$$\mathbf{u} \times \mathbf{v} = (u_y v_z - u_z v_y)\hat{\mathbf{x}} + (u_z v_x - u_x v_z)\hat{\mathbf{y}} + (u_x v_y - u_y v_x)\hat{\mathbf{z}}, \tag{B.4}$$

and the length of this vector may be expressed as

$$\|\mathbf{u} \times \mathbf{v}\| = \|\mathbf{u}\|\|\mathbf{v}\| \sin\theta. \tag{B.5}$$

Note that the dot product is commutative, but not the cross-product, that is

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}, \tag{B.6}$$

$$\mathbf{u} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{u}). \tag{B.7}$$

A second-order tensor, $\mathbf{U}$, is a linear operator that produces one vector from another, that is,

$$\begin{aligned} \mathbf{u} &= \mathbf{U}\mathbf{v}, \\ u_i &= U_{ij} v_j \quad (\text{sum over } j = 1,\, 2,\, 3). \end{aligned} \tag{B.8}$$

Here we introduce the use of index notation; i and j are assumed to take on the values 1, 2, and 3 for the x, y, and z components, respectively. Notice that in a Cartesian coordinate system, the second-order tensor $\mathbf{U}$ has the form of a 3×3 matrix. We also begin using the *summation convention*; repeated indices in a product are assumed to be summed over values from 1 to 3.

The projection property of the dot product can be used to express a vector in a different (i.e., rotated) Cartesian coordinate system. If the new coordinate axes are defined by the orthogonal unit vectors $\hat{\mathbf{x}}'$, $\hat{\mathbf{y}}'$, and $\hat{\mathbf{z}}'$ (expressed in the original x,y,z coordinates), then the x' coordinate of a vector $\mathbf{v}$ is given by $\hat{\mathbf{x}}' \cdot \mathbf{v}$. In this way the vector in the new coordinate system is given by

$$\mathbf{v}' = \begin{bmatrix} \hat{x}'_1 & \hat{x}'_2 & \hat{x}'_3 \\ \hat{y}'_1 & \hat{y}'_2 & \hat{y}'_3 \\ \hat{z}'_1 & \hat{z}'_2 & \hat{z}'_3 \end{bmatrix} \mathbf{v} \equiv \mathbf{A}\mathbf{v} \tag{B.9}$$

where $\mathbf{A}$ is the *transformation tensor* with components equal to the cosines of the angles between the primed and unprimed axes. We can express the same equation in index notation as

$$v'_i = A_{ij} v_j. \tag{B.10}$$

Because the rows of $\mathbf{A}$ are orthogonal unit vectors it follows that

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I} \tag{B.11}$$

where $\mathbf{I}$ is the identity matrix.

We often will also want a way to transform a Cartesian tensor to a new coordinate system. This can be obtained by applying the transformation tensor $\mathbf{A}$ to both sides of (B.8)

$$\begin{aligned} \mathbf{u} &= \mathbf{U}\mathbf{v} && \text{(B.12)} \\ \mathbf{A}\mathbf{u} &= \mathbf{A}\mathbf{U}\mathbf{v} && \text{(B.13)} \\ \mathbf{A}\mathbf{u} &= \mathbf{A}\mathbf{U}(\mathbf{A}^T\mathbf{A})\mathbf{v} && \text{(B.14)} \\ \mathbf{A}\mathbf{u} &= \mathbf{A}\mathbf{U}\mathbf{A}^T(\mathbf{A}\mathbf{v}) && \text{(B.15)} \\ \mathbf{u}' &= \mathbf{A}\mathbf{U}\mathbf{A}^T\mathbf{v}' && \text{(B.16)} \end{aligned}$$

and we see that the tensor operator that produces $\mathbf{u}'$ from $\mathbf{v}'$ in the primed coordinate system is given by

$$\mathbf{U}' = \mathbf{A}\mathbf{U}\mathbf{A}^T \qquad \text{(B.17)}$$

which we can use to convert $\mathbf{U}$ to $\mathbf{U}'$. In Chapter 2, we use the eigenvector matrix $\mathbf{N}$ to rotate the stress tensor into its principal axes coordinate system. The definition of $\mathbf{N}$ is similar to $\mathbf{A}$ except that the unit vectors are set to the columns rather than the rows. Thus $\mathbf{N}^T = \mathbf{A}$ and in this case the transformation equation is

$$\mathbf{U}' = \mathbf{N}^T\mathbf{U}\mathbf{N} \qquad \text{(B.18)}$$

Useful matrix identities include

$$\begin{aligned} \mathbf{A}(\mathbf{B}+\mathbf{C}) &= \mathbf{A}\mathbf{B}+\mathbf{A}\mathbf{C} && \text{(B.19)} \\ (\mathbf{A}+\mathbf{B})^T &= \mathbf{A}^T+\mathbf{B}^T && \text{(B.20)} \\ (\mathbf{A}\mathbf{B})^T &= \mathbf{B}^T\mathbf{A}^T && \text{(B.21)} \\ (\mathbf{A}\mathbf{B})^{-1} &= \mathbf{B}^{-1}\mathbf{A}^{-1} && \text{(B.22)} \\ (\mathbf{A}^{-1})^T &= (\mathbf{A}^T)^{-1} && \text{(B.23)} \end{aligned}$$

where for the last two we assume the existence of inverses of $\mathbf{A}$ and $\mathbf{B}$.

Functions that vary with position are termed *fields*; we can have scalar fields, vector fields, and tensor fields. In this case we may define spatial derivatives, such as the gradient, divergence, Laplacian, and curl.

The *gradient* of a scalar field, written $\nabla\lambda$, is a vector field, defined by the partial derivatives of the scalar in x, y, and z directions:

$$\begin{aligned} \mathbf{u} &= \nabla\lambda = \frac{\partial\lambda}{\partial x}\hat{\mathbf{x}} + \frac{\partial\lambda}{\partial y}\hat{\mathbf{y}} + \frac{\partial\lambda}{\partial z}\hat{\mathbf{z}}, && \text{(B.24)} \\ u_i &= \partial_i\lambda, \end{aligned}$$

where ∂_i is shorthand notation for $\partial/\partial x$, $\partial/\partial y$, and $\partial/\partial z$ for $i = 1$, 2, and 3 respectively. The gradient vector, $\nabla\lambda$, is normal to surfaces of constant λ.

The gradient of a vector field is a tensor field:

$$\begin{aligned} \mathbf{U} &= \nabla\mathbf{u}, && \text{(B.25)} \\ U_{ij} &= \partial_i u_j. \end{aligned}$$

The *divergence* of a vector field, written $\nabla \cdot \mathbf{u}$, is a scalar field:

$$\begin{aligned} \lambda &= \nabla \cdot \mathbf{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \\ &= \partial_i u_i \quad (\text{sum over } i = 1,\, 2,\, 3). \end{aligned} \tag{B.26}$$

The divergence of a second-order tensor field is a vector field:

$$\begin{aligned} \mathbf{u} &= \nabla \cdot \mathbf{U}, \\ u_j &= \partial_i U_{ij} \quad (\text{sum over } i = 1,\, 2,\, 3). \end{aligned} \tag{B.27}$$

The *Laplacian* of a scalar field, written $\nabla^2 \lambda$, is a scalar field:

$$\begin{aligned} \phi &= \nabla^2 \lambda = \nabla \cdot \nabla \lambda = \frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial y^2} + \frac{\partial^2 \lambda}{\partial z^2} \\ &= \partial_j \partial_j \lambda \quad (\text{sum over } j = 1,\, 2,\, 3). \end{aligned} \tag{B.28}$$

The Laplacian of a vector field is a vector field:

$$\begin{aligned} \mathbf{u} &= \nabla^2 \mathbf{v} = \nabla \cdot \nabla \mathbf{v}, \\ u_i &= \partial_j \partial_j v_i \quad (\text{sum over } j = 1,\, 2,\, 3). \end{aligned} \tag{B.29}$$

The *curl* of a vector field is a vector field:

$$\begin{aligned} \mathbf{u} = \nabla \times \mathbf{v} &= \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \right) \hat{\mathbf{x}} \\ &\quad + \left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \right) \hat{\mathbf{y}} \\ &\quad + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right) \hat{\mathbf{z}}. \end{aligned} \tag{B.30}$$

The operator ∇ is distributive, that is,

$$\nabla(\lambda + \phi) = \nabla\lambda + \nabla\phi, \tag{B.31}$$

$$\nabla \cdot (\mathbf{u} + \mathbf{v}) = \nabla \cdot \mathbf{u} + \nabla \cdot \mathbf{v}, \tag{B.32}$$

$$\nabla \times (\mathbf{u} + \mathbf{v}) = \nabla \times \mathbf{u} + \nabla \times \mathbf{v}. \tag{B.33}$$

A vector field defined as the gradient of a scalar field is curl free, that is,

$$\nabla \times (\nabla \lambda) = 0. \tag{B.34}$$

A vector field defined as the curl of another vector field is divergence free, that is,

$$\nabla \cdot (\nabla \times \mathbf{u}) = 0. \tag{B.35}$$

The following identities are often useful:

$$\nabla \cdot \lambda \mathbf{u} = \lambda \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla \lambda, \tag{B.36}$$

$$\nabla \times \lambda \mathbf{u} = \lambda \nabla \times \mathbf{u} + \nabla \lambda \times \mathbf{u}, \tag{B.37}$$

$$\nabla \times (\nabla \times \mathbf{u}) = \nabla \nabla \cdot \mathbf{u} - \nabla^2 \mathbf{u}. \tag{B.38}$$

The identity matrix $\mathbf{I}$ can be written in index notation as δ_{ij} where

$$\delta_{ij} = \begin{cases} 1 \text{ for } i = j, \\ 0 \text{ for } i \neq j. \end{cases} \tag{B.39}$$

When δ_{ij} appears as part of a product in equations, it can be used to switch the indices of other terms, that is,

$$\partial_i \delta_{ij} u_k = \partial_j u_k. \tag{B.40}$$

Of great importance in continuum mechanics is *Gauss's theorem*, which equates the volume integral of a vector field to the surface integral of the orthogonal component of the vector field:

$$\int_V \nabla \cdot \mathbf{u} \, dV = \int_S \mathbf{u} \cdot \hat{\mathbf{n}} \, dS, \tag{B.41}$$

where $\hat{\mathbf{n}}$ is the outward normal vector to the surface.

B.2 Complex Numbers

We use complex numbers in this book mostly as a shorthand way to keep track of the phase and amplitude of harmonic waves. The imaginary number i is defined as

$$i^2 = -1. \tag{B.42}$$

It follows that

$$\sqrt{-1} = \pm i \qquad \text{and} \qquad 1/i = -i. \tag{B.43}$$

A *complex number* can be written as

$$z = x + iy, \tag{B.44}$$

where $x = \mathrm{Re}(z)$ is the real part of z and $y = \mathrm{Im}(z)$ is the imaginary part of z (note that y itself is a real number). Complex numbers obey the commutative, associative, and distributive rules of arithmetic. The *complex conjugate* of z is defined as

$$z^* = x - iy. \tag{B.45}$$

Complex numbers may be represented as points on the complex plane (see Fig. B.2), either in Cartesian coordinates by x and y, or in polar coordinates by their *phase*, θ, and their *magnitude*, $r = |z|$. These forms are related by

$$z = re^{i\theta} = r(\cos\theta + i \sin\theta) = x + yi. \tag{B.46}$$

The magnitude $|z|$ is also sometimes referred to as the *absolute value* of z. Note that

$$y/x = \tan\theta \tag{B.47}$$

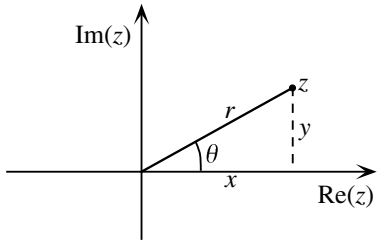


Figure B.2: The complex number z can be represented as a point in the complex plane.

and that

$$zz^* = (x+iy)(x-iy) = x^2 + y^2 = |z|^2. \tag{B.48}$$

Now let us illustrate the convenience of complex numbers for describing wave motion. A harmonic wave of angular frequency ω is defined by its amplitude a and phase delay ϕ (Fig. B.3), that is,

$$f(t) = a\cos(\omega t - \phi). \tag{B.49}$$

Using a trigonometric identity for $\cos(\omega t - \phi)$, this can be rewritten

$$\begin{aligned} f(t) &= a\cos\phi\cos\omega t + a\sin\phi\sin\omega t & \text{(B.50)}\\ &= a_1\cos\omega t + a_2\sin\omega t, & \text{(B.51)} \end{aligned}$$

where $a_1 \equiv a\cos\phi$ and $a_2 \equiv a\sin\phi$. This is a more convenient form because it is a linear function of the coefficients a_1 and a_2. A harmonic wave of arbitrary phase can always be expressed as a weighted sum of a sine and a cosine function. Two waves of the same frequency can be summed by adding their sine and cosine coefficients. Note that a and ϕ may be recovered from the new coefficients using

$$a^2 = a_1^2 + a_2^2 \quad \text{and} \quad \phi = \tan^{-1}(a_2/a_1). \tag{B.52}$$

We can obtain the same relationships using a single complex coefficient A by writing the function $f(t)$ as a complex exponential function

$$f(t) = \mathrm{Re}\left[Ae^{-i\omega t}\right]. \tag{B.53}$$

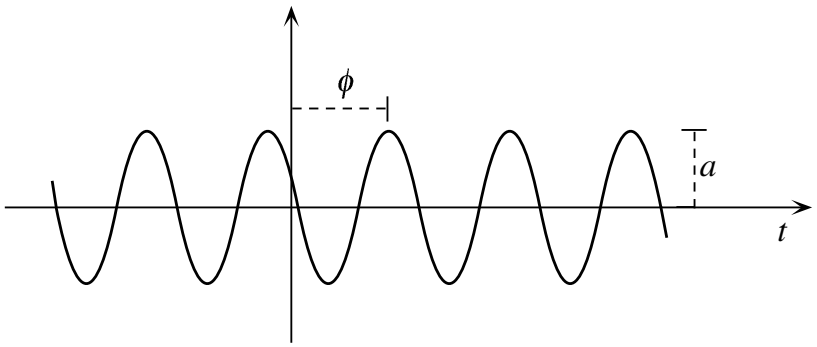


Figure B.3: The amplitude a and phase delay ϕ of a cosine function.

Expanding this, we have

$$\begin{aligned} f(t) &= \mathrm{Re}\left[A\left(\cos(-\omega t) + i\sin(-\omega t)\right)\right] \\ &= \mathrm{Re}\left[A(\cos\omega t - i\sin\omega t)\right]. \end{aligned} \tag{B.54}$$

Now consider the real and imaginary parts of $A = x + iy$:

$$f(t) = \mathrm{Re}\left[(x+iy)(\cos\omega t - i\sin\omega t)\right]. \tag{B.55}$$

The real terms give

$$f(t) = x\cos\omega t + y\sin\omega t. \tag{B.56}$$

This is identical to (B.51) if we assume

$$a_1 = x = \mathrm{Re}(A), \tag{B.57}$$

$$a_2 = y = \mathrm{Im}(A). \tag{B.58}$$

In this way a single complex number can keep track of both the amplitude and phase of harmonic waves. For convenience, equations such as (B.53) usually do not explicitly include the Re function; in these cases the reader should keep in mind that the real part must always be taken before the equation has a physical meaning. This applies, for example, to Equation (3.36) in Chapter 3.

Appendix C

The Eikonal Equation

Consider the propagation of compressional waves in heterogeneous media. From (3.31), we have

$$\nabla^2\phi - \frac{1}{\alpha^2}\frac{\partial^2(\phi)}{\partial t^2} = 0, \tag{C.1}$$

where the scalar potential for compressional waves, ϕ, obeys the relationship $\mathbf{u} = \nabla\phi$ where $\mathbf{u}$ is displacement. The P-wave velocity, α, is a function of position, $\alpha = \alpha(\mathbf{x})$. Now assume a harmonic solution of the form

$$\phi(t) = A(\mathbf{x})e^{-i\omega[t-T(\mathbf{x})]}, \tag{C.2}$$

where T is a phase factor and A is the local amplitude. We can expand the spatial derivatives of ϕ as

$$\nabla\phi = \nabla A\, e^{-i\omega[t-T(\mathbf{x})]} - i\omega A\,\nabla T e^{-i\omega[t-T(\mathbf{x})]}, \tag{C.3}$$

$$\begin{aligned}\nabla^2\phi &= \nabla^2 A\, e^{-i\omega[t-T(\mathbf{x})]} - i\omega\nabla T\cdot\nabla A\, e^{-i\omega[t-T(\mathbf{x})]} \\ &\quad -i\omega\,\nabla A\cdot\nabla T e^{-i\omega[t-T(\mathbf{x})]} - i\omega A\,\nabla^2 T e^{-i\omega[t-T(\mathbf{x})]} - \omega^2 A\nabla T\cdot\nabla T e^{-i\omega[t-T(\mathbf{x})]} \\ &= \left(\nabla^2 A - \omega^2 A|\nabla T|^2 - i[2\omega\nabla A\cdot\nabla T + \omega A\,\nabla^2 T]\right) e^{-i\omega[t-T(\mathbf{x})]}\end{aligned} \tag{C.4}$$

and the time derivatives as

$$\frac{\partial^2(\phi)}{\partial t^2} = -A\omega^2 e^{-i\omega[t-T(\mathbf{x})]}. \tag{C.5}$$

Substituting into (C.1) and dividing out the constant $e^{-i\omega[t-T(\mathbf{x})]}$ factor, we obtain

$$\nabla^2 A - \omega^2 A|\nabla T|^2 - i[2\omega\nabla A\cdot\nabla T + \omega A\nabla^2 T] = -\frac{A\omega^2}{\alpha^2}. \tag{C.6}$$

From the real part of this equation we have

$$\nabla^2 A - \omega^2 A|\nabla T|^2 = -\frac{A\omega^2}{\alpha^2} \tag{C.7}$$

and from the imaginary part we have

$$2\omega\nabla A\cdot\nabla T + \omega A\nabla^2 T = 0$$

or

$$2\nabla A \cdot \nabla T + A\nabla^2 T = 0. \tag{C.8}$$

Dividing (C.7) by $A\omega^2$ and rearranging, we obtain

$$|\nabla T|^2 - \frac{1}{\alpha^2} = \frac{\nabla^2 A}{A\omega^2}. \tag{C.9}$$

We now make the high-frequency approximation that ω is sufficiently large that the $1/\omega^2$ term can be ignored. We thus have

$$|\nabla T|^2 = \frac{1}{\alpha^2}. \tag{C.10}$$

A similar equation can be derived for S-waves. Thus a more general form for this equation is

$$|\nabla T|^2 = \frac{1}{c^2}, \tag{C.11}$$

where c is either the local P-wave speed, α, or the local S-wave speed, β. This is the standard form for the *eikonal equation* (e.g., Equation 4.41 in Aki and Richards, 2002). This equation can also be expressed as

$$|\nabla T|^2 = u^2, \tag{C.12}$$

where $u = 1/c$ is called the *slowness*. Since the velocity, c, typically appears in the denominator in ray tracing equations, we will find that it is usually more convenient to use the slowness. The phase factor, T, is also sometimes called the travel time function. We can write (C.12) in expanded form as

$$|\nabla T|^2 = (\partial_x T)^2 + (\partial_y T)^2 + (\partial_z T)^2 = u^2. \tag{C.13}$$

Note that the phase factor T has a gradient whose amplitude is equal to the local slowness. The function $T(\mathbf{x}) = constant$ defines surfaces called wavefronts. Lines perpendicular to $T(\mathbf{x})$ or parallel to $\nabla T(\mathbf{x})$ are termed rays. The ray direction is defined by the gradient of T,

$$\nabla T = u\hat{\mathbf{k}} = \mathbf{s}, \tag{C.14}$$

where $\hat{\mathbf{k}}$ is the unit vector in the local ray direction and $\mathbf{s}$ is the slowness vector. The function $T(\mathbf{x})$ has units of time and, because the wavefronts propagate with the local slowness in a direction parallel to the rays, it is simply the time required for a wavefront to reach $\mathbf{x}$.

The eikonal equation forms the basis for ray theoretical approaches to modeling seismic wave propagation, which are discussed in Chapter 4. It is an approximate solution, valid at high frequencies so that the terms in the wave equation that involve spatial velocity gradients in the Lamé parameters (see Equation 3.18) and the wave amplitude (C.9) can be neglected. Thus it is valid only at seismic wavelengths short compared to the distances in the medium over which velocity and amplitude change significantly. However, because this is often the case in the Earth, ray theoretical methods based on the eikonal equation have proven to be extremely useful.

Now recall Equation (C.8), the imaginary part of (C.6):

$$2\nabla A \cdot \nabla T + A\nabla^2 T = 0, \tag{C.15}$$

where A is the wave amplitude and T is the phase factor or travel time for the wavefront. Remembering that $\nabla T = u\hat{\mathbf{k}}$ where u is the local wave slowness and $\hat{\mathbf{k}}$ is the unit vector in the ray direction, we have

$$2u\hat{\mathbf{k}} \cdot \nabla A = -\nabla \cdot (u\hat{\mathbf{k}})A, \tag{C.16}$$

or

$$A = \frac{-2u\nabla A \cdot \hat{\mathbf{k}}}{\nabla \cdot (u\hat{\mathbf{k}})}. \tag{C.17}$$

Integrating along the ray path in the direction $\hat{\mathbf{k}}$, a solution to (C.17) is provided by

$$A = \exp\left(-\frac{1}{2}\int \frac{\nabla \cdot (u\hat{\mathbf{k}})}{u}\, ds\right). \tag{C.18}$$

Substituting this expression into (C.2) for the compressional wave potential we can write

$$\phi(\omega) = Ae^{-i\omega T(\mathbf{r})} = \exp\left(-\frac{1}{2}\int_{\text{path}} \frac{\nabla \cdot (u_\alpha\hat{\mathbf{k}})}{u_\alpha}\, ds\right) \exp\left(-i\omega \int_{\text{path}} u_\alpha\, ds\right). \tag{C.19}$$

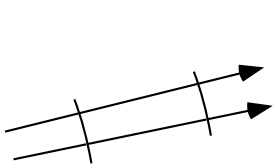
Here $u_\alpha\, ds$ is the travel time along the path and represents the usual oscillatory wave motion encountered previously. The exponent in the first exponential, however, is negative and real. Thus it represents a decay in amplitude along the ray path. This exponent can be further manipulated:

$$\begin{aligned}
-\frac{1}{2}\int_{\text{path}} \frac{\nabla \cdot (u_\alpha\hat{\mathbf{k}})}{u_\alpha}\, ds &= -\frac{1}{2}\int_{\text{path}} \left(\frac{\hat{\mathbf{k}} \cdot \nabla u_\alpha}{u_\alpha} + \nabla \cdot \hat{\mathbf{k}}\right) ds \\
&= -\frac{1}{2}\int_{\text{path}} \left(\frac{1}{u_\alpha}\frac{du_\alpha}{ds} + \nabla \cdot \hat{\mathbf{k}}\right) ds \\
&= -\frac{1}{2}\int_{\text{path}} \frac{du_\alpha}{u_\alpha} - \frac{1}{2}\int_{\text{path}} \nabla \cdot \hat{\mathbf{k}}\, ds \\
&= -\frac{1}{2}\ln u\Big|_{u_0}^{u_\alpha} - \frac{1}{2}\int_{\text{path}} \nabla \cdot \hat{\mathbf{k}}\, ds \\
&= -\frac{1}{2}\ln\left(\frac{u_\alpha}{u_0}\right) - \frac{1}{2}\int_{\text{path}} \nabla \cdot \hat{\mathbf{k}}\, ds.
\end{aligned} \tag{C.20}$$

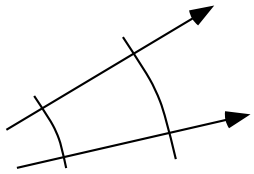
Here u_0 is the slowness at the source where the radiation first began. If we substitute (C.20) into (C.19) we find:

$$\phi(\omega) = (u_0/u_\alpha)^{1/2} e^{-\frac{1}{2}\int_{\text{path}} \nabla \cdot \hat{\mathbf{k}}\, ds} e^{-i\omega \int_{\text{path}} u_\alpha\, ds}. \tag{C.21}$$

This equation describes the effect of geometrical spreading on wave amplitudes. $\nabla \cdot \hat{\mathbf{k}}$, the divergence of the unit vector parallel to the ray, represents the curvature of the wavefront and, when large, leads to a large amplitude reduction.

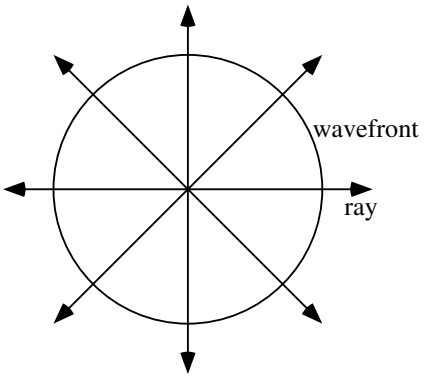


small amplitude reduction



large amplitude reduction

To illustrate the effect of the geometrical spreading term, consider a spherical wave diverging in a homogeneous whole space. In this case, wavefronts are spheres while rays are radii ($\hat{\mathbf{k}} = \hat{\mathbf{r}}$). Recalling the expression for the divergence in spherical coordinates we can write:



$$\nabla \cdot \hat{\mathbf{k}} = \frac{1}{r^2}\partial_r(r^2) = \frac{2}{r}. \qquad \text{(C.22)}$$

Substituting for the first exponential term in (C.21), we obtain

$$-\frac{1}{2}\int_{path} \nabla \cdot \hat{\mathbf{k}}\, ds = -\frac{1}{2}\int_{path} \frac{2ds}{r} = -\int_{r_0}^{r} \frac{dr}{r} = \ln\left(\frac{r_0}{r}\right) \qquad \text{(C.23)}$$

and thus from (C.21), and because $u_0 = u_\alpha$ from homogeneity, we obtain

$$\phi(\omega) = \left(\frac{r_0}{r}\right) e^{i\omega \int_{\text{path}} u_\alpha\, ds} = \left(\frac{r_0}{r}\right) e^{-i\omega u_\alpha r}. \qquad \text{(C.24)}$$

Thus, in a homogeneous medium (a whole space), the amplitude decays as r^{-1}. This result also follows from energy considerations (see Chapter 6) since the area of the wavefront grows as r^2.

Appendix D

Fortran Subroutines

The following FORTRAN77 subroutines are required for some of the exercises.

```
! LAYERXT calculates dx and dt for a ray in a layer with a linear
!   velocity gradient.  This is a highly modified version of a
!   subroutine in Chris Chapman's WKBJ program.
!
! Inputs:   p     =  horizontal slowness
!           h     =  layer thickness
!           utop  =  slowness at top of layer
!           ubot  =  slowness at bottom of layer
! Returns:  dx    =  range offset
!           dt    =  travel time
!           irtr  =  return code
!                 = -1,  zero thickness layer
!                 =  0,  ray turned above layer
!                 =  1,  ray passed through layer
!                 =  2,  ray turned in layer, 1 leg counted in dx,dt
!
      subroutine LAYERXT(p,h,utop,ubot,dx,dt,irtr)

      if (p.ge.utop) then      !ray turned above layer
         dx=0.
         dt=0.
         irtr=0
         return
      else if (h.eq.0.) then        !zero thickness layer
         dx=0.
         dt=0.
         irtr=-1
         return
      end if

      u1=utop
      u2=ubot
      v1=1./u1
      v2=1./u2
```

```
      b=(v2-v1)/h             !slope of velocity gradient

      eta1=sqrt(u1**2-p**2)

      if (b.eq.0.) then       !constant velocity layer
         dx=h*p/eta1
         dt=h*u1**2/eta1
         irtr=1
         return
      end if

      x1=eta1/(u1*b*p)
      tau1=(alog((u1+eta1)/p)-eta1/u1)/b

      if (p.ge.ubot) then     !ray turned within layer,
         dx=x1                !no contribution to integral
         dtau=tau1            !from bottom point
         dt=dtau+p*dx
         irtr=2
         return
      end if

      irtr=1

      eta2=sqrt(u2**2-p**2)
      x2=eta2/(u2*b*p)
      tau2=(alog((u2+eta2)/p)-eta2/u2)/b

      dx=x1-x2
      dtau=tau1-tau2

      dt=dtau+p*dx

      return
      end
```

```
! RTCOEF calculates P/SV reflection/transmission coefficients
!   for an interface between two solid layers, based on the
!   equations on p. 149-150 of Aki and Richards.  This version
!   is modified from an older routine provided by Tom Sereno.
!
!  Inputs:    vp1     =  P-wave velocity of layer 1 (top layer)
!  (real)     vs1     =  S-wave velocity of layer 1
!             den1    =  density of layer 1
!             vp2     =  P-wave velocity of layer 2 (bottom layer)
!             vs2     =  S-wave velocity of layer 2
!             den2    =  density of layer 2
!             hslow   =  horizontal slowness (ray parameter)
```

```
!  Returns:   rt(1)   =  down P to P up     (refl)
!  (complex)  rt(2)   =  down P to S up     (refl)
!             rt(3)   =  down P to P down   (tran)
!             rt(4)   =  down P to S down   (tran)
!             rt(5)   =  down S to P up     (refl)
!             rt(6)   =  down S to S up     (refl)
!             rt(7)   =  down S to P down   (tran)
!             rt(8)   =  down S to S down   (tran)
!             rt(9)   =    up P to P up     (tran)
!             rt(10)  =    up P to S up     (tran)
!             rt(11)  =    up P to P down   (refl)
!             rt(12)  =    up P to S down   (refl)
!             rt(13)  =    up S to P up     (tran)
!             rt(14)  =    up S to S up     (tran)
!             rt(15)  =    up S to P down   (refl)
!             rt(16)  =    up S to S down   (refl)
!
! NOTE:  All input variables are real.
!        All output variables are complex!
!        Coefficients are not energy normalized.
!
      SUBROUTINE RTCOEF(vp1,vs1,den1,vp2,vs2,den2,hslow,rt)
      implicit complex (a-h,o-z)
      complex rt(16)
      real vp1,vs1,den1,vp2,vs2,den2,hslow

      alpha1=cmplx(vp1,0.)
      beta1=cmplx(vs1,0.)
      rho1=cmplx(den1,0.)
      alpha2=cmplx(vp2,0.)
      beta2=cmplx(vs2,0.)
      rho2=cmplx(den2,0.)
      p=cmplx(hslow,0.)

      cone=cmplx(1.,0.)
      ctwo=cmplx(2.,0.)

      term1=(cone-ctwo*beta1**2*p**2)
      term2=(cone-ctwo*beta2**2*p**2)
      a=rho2*term2-rho1*term1
      b=rho2*term2+ctwo*rho1*beta1**2*p**2
      c=rho1*term1+ctwo*rho2*beta2**2*p**2
      d=ctwo*(rho2*beta2**2-rho1*beta1**2)

! compute signs and cosines, allowing for complex incidence angles
      si1=alpha1*p
      si2=alpha2*p
      sj1=beta1*p
      sj2=beta2*p
      ci1=csqrt(cone-si1**2)
      ci2=csqrt(cone-si2**2)
```

```
cj1=csqrt(cone-sj1**2)
cj2=csqrt(cone-sj2**2)

E=b*ci1/alpha1+c*ci2/alpha2
F=b*cj1/beta1+c*cj2/beta2
G=a-d*ci1*cj2/(alpha1*beta2)
H=a-d*ci2*cj1/(alpha2*beta1)
DEN=E*F+G*H*p**2

trm1=b*ci1/alpha1-c*ci2/alpha2
trm2=a+d*ci1*cj2/(alpha1*beta2)
rt(1)=(trm1*F-trm2*H*p**2)/DEN          !refl down P to P up

trm1=a*b+c*d*ci2*cj2/(alpha2*beta2)
rt(2)=(-ctwo*ci1*trm1*p)/(beta1*DEN)    !refl down P to S up

rt(3)=ctwo*rho1*ci1*F/(alpha2*DEN)      !trans down P to P down

rt(4)=ctwo*rho1*ci1*H*p/(beta2*DEN)     !trans down P to S down

trm1=a*b+c*d*ci2*cj2/(alpha2*beta2)
rt(5)=(-ctwo*cj1*trm1*p)/(alpha1*DEN)   !refl down S to P up

trm1=b*cj1/beta1-c*cj2/beta2
trm2=a+d*ci2*cj1/(alpha2*beta1)
rt(6)=-(trm1*E-trm2*G*p**2)/DEN         !refl down S to S up

rt(7)=-ctwo*rho1*cj1*G*p/(alpha2*DEN)   !trans down S to P down

rt(8)=ctwo*rho1*cj1*E/(beta2*DEN)       !trans down S to S down

rt(9)=ctwo*rho2*ci2*F/(alpha1*DEN)      !trans up P to P up

rt(10)=-ctwo*rho2*ci2*G*p/(beta1*DEN)   !trans up P to S up

trm1=b*ci1/alpha1-c*ci2/alpha2
trm2=a+d*ci2*cj1/(alpha2*beta1)
rt(11)=-(trm1*F+trm2*G*p**2)/DEN        !refl up P to P down

trm1=a*c+b*d*ci1*cj1/(alpha1*beta1)
rt(12)=(ctwo*ci2*trm1*p)/(beta2*DEN)    !refl up P to S down

rt(13)=ctwo*rho2*cj2*H*p/(alpha1*DEN)   !trans up S to P up

rt(14)=ctwo*rho2*cj2*E/(beta1*DEN)      !trans up S to S up

trm1=a*c+b*d*ci1*cj1/(alpha1*beta1)
rt(15)=(ctwo*cj2*trm1*p)/(alpha2*DEN)   !refl up S to P down

trm1=b*cj1/beta1-c*cj2/beta2
trm2=a+d*ci1*cj2/(alpha1*beta2)
```

```
      rt(16)=(trm1*E+trm2*H*p**2)/DEN        !refl up S to S down

      return
      end



! GETAUX returns auxiliary fault plane, given strike,dip,rake
! of main fault plane.
!
! Inputs:  strike1, dip1, rake1 (degrees, primary fault plane)
! Returns: strike2, dip2, rake2 (degrees, auxiliary fault plane)
!
      subroutine GETAUX(strike1,dip1,rake1,strike2,dip2,rake2)
      degrad=180./3.1415927
      s1=strike1/degrad
      d1=dip1/degrad
      r1=rake1/degrad

      d2=acos(sin(r1)*sin(d1))

      sr2=cos(d1)/sin(d2)
      cr2=-sin(d1)*cos(r1)/sin(d2)
      r2=atan2(sr2,cr2)

      s12=cos(r1)/sin(d2)
      c12=-1./(tan(d1)*tan(d2))
      s2=s1-atan2(s12,c12)

      strike2=s2*degrad
      dip2=d2*degrad
      rake2=r2*degrad

      if (dip2.gt.90.) then
         strike2=strike2+180.
         dip2=180.-dip2
         rake2=360.-rake2
      end if
      if (strike2.gt.360.) strike2=strike2-360.

      return
      end
```

Appendix E

Time Series and Fourier Transforms

The following is a summary of the time series concepts that are used in this book. For more details, the reader should consult Bracewell (1978) or other texts on time series analysis.

E.1 Convolution

Consider two time series $u(t)$ and $v(t)$. The *convolution* of these functions is defined as

$$u(t) * v(t) = \int_{-\infty}^{\infty} u(\tau)v(t-\tau)\,d\tau. \tag{E.1}$$

Convolution with simple pulses generally results in a smoothing of the original time series. For example, convolution with a boxcar function will produce the same result as averaging the adjacent points (Fig. E.1).

Convolution is commutative and associative, that is,

$$u(t) * v(t) = v(t) * u(t), \tag{E.2}$$

$$u(t) * [v(t) * w(t)] = [u(t) * v(t)] * w(t). \tag{E.3}$$

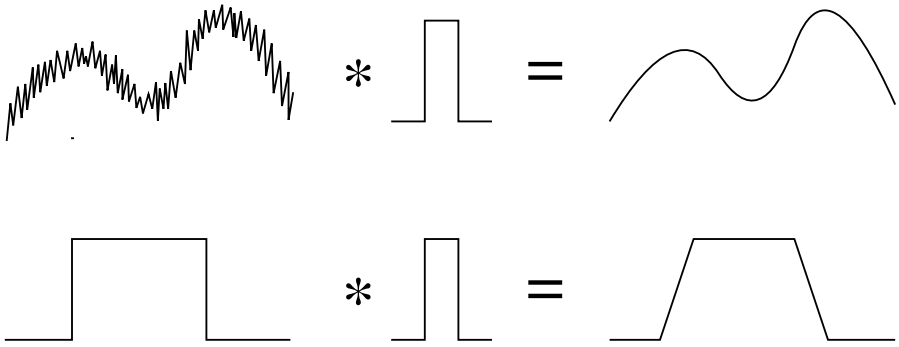


Figure E.1: Examples of convolution with a boxcar function.

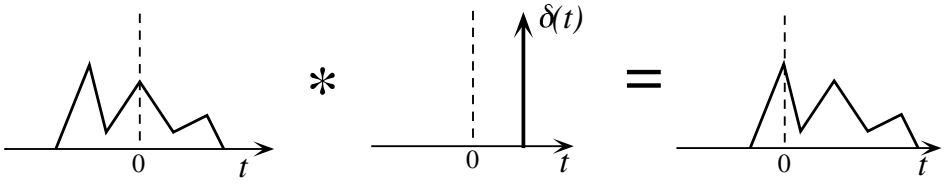


Figure E.2: Convolution with a delta function does not change the shape of a function, but shifts it in time to align with the position of the delta function.

It also follows from (E.1) that

$$\frac{\partial}{\partial t}\left[u(t) * v(t)\right] = \left[u(t) * v(t)\right]' = u'(t) * v(t) = u(t) * v'(t). \tag{E.4}$$

Convolution provides a convenient way to describe the construction of synthetic seismograms (the predicted ground motion at a particular site as a function of the seismic source and a specified Earth model). For example, the seismogram could be written

$$u(t) = s(t) * G(t) * a(t) * r(t), \tag{E.5}$$

where $s(t)$ is the effective source-time function, $G(t)$ is the elastic Green's function that connects the source and receiver (the hard part to compute!), $a(t)$ is an attenuation operator that approximates the effect of Q along the ray path, and $r(t)$ is the response of the receiver.

The *delta function*, $\delta(t)$, is often useful and is defined as

$$\begin{aligned} \delta(t) &= 0 \quad \text{for } t \neq 0, \\ \int_{-\infty}^{\infty} \delta(t)\, dt &= 1. \end{aligned} \tag{E.6}$$

Convolution with a delta function leaves the original function unchanged, that is,

$$u(t) * \delta(t) = u(t). \tag{E.7}$$

The delta function may act to produce a time shift in the original time series (Fig. E.2):

$$u(t) * \delta(t - t_0) = u(t - t_0). \tag{E.8}$$

E.2 Fourier Transform

There are many definitions of the Fourier transform. Here we assume the transform is defined by

$$\mathcal{F}[u(t)] = u(\omega) = \int_{-\infty}^{\infty} u(t) e^{i\omega t}\, dt \tag{E.9}$$

and the inverse Fourier transform is

$$\mathcal{F}^{-1}[u(\omega)] = u(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} u(\omega) e^{-i\omega t}\, d\omega. \tag{E.10}$$

The function $u(t)$ is said to be in the *time domain*; the corresponding function $u(\omega)$ is in the *frequency domain*. From these definitions, various useful relationships may be derived. The scale rule is

$$\mathcal{F}[u(t/a)] = |a|u(a\omega), \tag{E.11}$$

$$u(t/a) = |a|\mathcal{F}^{-1}[u(a\omega)]. \tag{E.12}$$

The differentiation rule is

$$\mathcal{F}[\dot{u}(t)] = -i\omega u(\omega). \tag{E.13}$$

The shift theorem is

$$\mathcal{F}\left[u(t+a)\right] = u(\omega)e^{-i\omega a}. \tag{E.14}$$

Finally, the convolution rule is

$$\mathcal{F}[u(t) * v(t)] = u(\omega)v(\omega), \tag{E.15}$$

$$\mathcal{F}^{-1}[u(\omega)v(\omega)] = u(t) * v(t). \tag{E.16}$$

In other words, the convolution of two functions in the time domain is equivalent to the product of the corresponding functions in the frequency domain.

E.3 Hilbert Transform

A phase shift of π in the frequency domain is equivalent to a polarity reversal in the time domain (multiplying the time series by -1). In this case the pulse shapes are not changed. However, a frequency-independent phase shift that is not equal to a multiple of π will cause pulse distortion. An example that occurs frequently in seismology is the Hilbert transform, which results from a $\pi/2$ phase advance.

A forward and inverse Fourier transform will reproduce the original time series

$$\begin{aligned} u(t) &= \mathcal{F}^{-1}\left(\mathcal{F}[u(t)]\right) \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} u(\xi)e^{i\omega\xi}\,d\xi\right]e^{-i\omega t}\,d\omega. \end{aligned} \tag{E.17}$$

We may define the Hilbert transform of $u(t)$ as $\bar{u}(t)$, by inserting a $-\pi/2$ phase shift[2] in the outer integral:

$$\bar{u}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} u(\xi)e^{i\omega\xi}\,d\xi\right]\frac{\mathrm{sgn}(\omega)}{i}e^{-i\omega t}\,d\omega, \tag{E.18}$$

where the $\mathrm{sgn}(\omega)$ keeps the time series real. Rearranging and evaluating the ω integral, one can show

$$\bar{u}(t) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{u(\xi)}{\xi - t}\,d\xi, \tag{E.19}$$

[2]The sign of the phase shift depends upon the sign of $i\omega$ in the Fourier transform; here we assume the sign convention of (E.9) and (E.10).

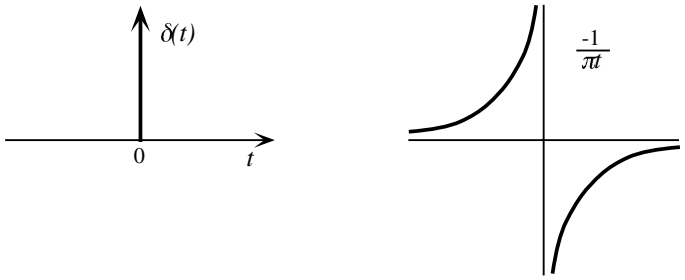


Figure E.3: The Hilbert transform of a delta function.

in which the singularity at $\xi = t$ is handled by taking the Cauchy principle value of the integral. The Hilbert transform of a delta function is

$$\bar{\delta}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\delta(\xi)}{\xi - t}\, d\xi = -\frac{1}{\pi t} \tag{E.20}$$

and is illustrated in Figure E.3).

Thus the Hilbert transform may also be expressed as a convolution

$$\bar{u}(t) = u(t) * \left(-\frac{1}{\pi t}\right). \tag{E.21}$$

Note also that

$$\overline{f * g} = f * \bar{g} = \bar{f} * g. \tag{E.22}$$

Equations (E.18), (E.19), and (E.21) are all equivalent definitions of the Hilbert transform of $u(t)$. In practice, Hilbert transforms are most easily calulated with a computer by using (E.18) and a Digital Fourier Transform.

Any frequency-independent phase shift may be expressed as a linear combination of a function and its Hilbert transform (an extension to all frequency components of the fact that a phase shift for a harmonic wave can be expressed as a linear combination of a sine and cosine wave; the Hilbert transform of a sine wave is a cosine wave). Applying the Hilbert transform twice produces a polarity reversal in the time series (a π phase shift); applying the Hilbert transform four times reproduces the original time series. The Hilbert transform does not change the amplitude spectrum of a time series; only the phase of the different frequency components is affected.

The Hilbert transform of a delta function is acausal in the sense that a precursory tail extends to $-\infty$ in time. In seismology, of course, energy cannot arrive prior to the time of source initiation (and in most cases cannot arrive before the time of the direct P-wave arrival). This is explained by the fact that the Hilbert transformed pulses occur on secondary arriving branches (such as PmP, PP, etc.) and that the Hilbert transform is predicted only as a high-frequency approximation to the true pulse shape. In practice, with bandlimited data, the Hilbert transform does not produce notably acausal pulse shapes (Fig. E.4).

Another useful function is the *analytic time series*, defined as:

$$U(t) = u(t) + i\bar{u}(t). \tag{E.23}$$

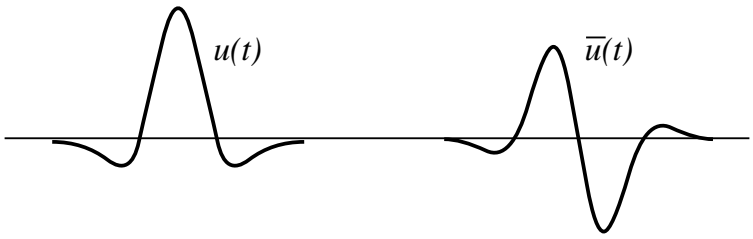


Figure E.4: The Hilbert transform of a typical seismic pulse.

In a sense, $U(t)$ bears the same resemblance to $u(t)$ as does $e^{i\omega t}$ to $\cos(\omega t)$. We also have the *envelope time function*, defined as:

$$E(t) = \left[u^2(t) + \bar{u}^2(t)\right]^{1/2}. \tag{E.24}$$

The envelope time function is useful as a local estimate of the amplitude that is not sensitive to individual zero crossings in the seismogram.