4 INVERSION METHODS

4.1 INTRODUCTION

As explained in Chapter 3, in order to obtain the magnitude of the scattering coefficient, we have to solve the system of equations:

$$\begin{array}{l} w_{11}f_1+\cdots+w_{1j}f_j+\cdots+w_{1N}f_N=p_1 \\ \vdots \\ w_{i1}f_1+\cdots+w_{ij}f_j+\cdots+w_{iN}f_N=p_i \\ \vdots \\ w_{M1}f_1+\cdots+w_{1j}f_j+\cdots+w_{MN}f_N=p_M \end{array} \tag{4.1}$$

where N is the number of blocks in which the region has been divided and M is the number of residuals obtained from the seismograms. In our case $N \approx 50000$ and $M \approx 10000$. To solve such a system of equations it is not possible to use conventional matrix theory methods to invert the system. There are several reasons [46]. The most important reason is that these systems of equations are always close to singular. While not exact linear combinations of each other, some of the equations may be so close to be linearly dependent that round off errors render them linearly dependent at some stage in the solution process. This may make the numerical procedure to fail. If it does not fail, round off errors in the solution process can swamp the true solution. This problem particularly emerges if N is a large number (N>10)!

For large values of M and N there are very convenient algebraic iterative methods based on the "method of projections" as first proposed by Kaczmarz [47]. These methods have been successfully used in Computerized Tomographic (CT) imaging for medical applications [48]. The simplest iterative method is the so-called Algebraic Reconstruction Technique (ART) algorithm. Another method based on ART is the Simultaneous Iterative Reconstruction Technique (SIRT). These methods are discussed in sections 4.2 and 4.3.

Algebraic methods are slow although they have some advantages discussed in section 4.4.7. There is a non-iterative algorithm that performs the inversion more efficiently. This algorithm is the Filtered Backprojection (FBP) algorithm. This algorithm uses a completely different approach to carry out the inversion and the solution is readily obtained as a linear combination of the residuals. The theoretical background necessary to grasp the intuitive ideas behind the method are introduced for 2-dimensional reconstruction problems in section 4.4.1. In section 4.5 the algorithm is generalized in order to carry out inversions in three dimensions.

4.2 ART ALGORITHM

In order to solve Eq.(4.1) we may consider the geometrical meaning of a system of equations. We consider a *N*-dimensional space. In this space, each equation represents a hyperplane. When a unique solution exists, the intersection of all these hyperplanes is a single point.

The computational procedure to locate the solution consists of first starting with an initial guess, denoted by $\vec{f}^{(0)} = (f_1^{(0)}, f_2^{(0)}, ..., f_N^{(0)})$. In most cases we simply assign a value of zero to all the f_i's. This initial guess is projected on the hyperplane represented by the first equation in (4.1) giving $\vec{f}^{(1)}$. $\vec{f}^{(1)}$ is then projected on the hyperplane represented by the second equation in (4.1) to yield $\vec{f}^{(2)}$ and so on. This is illustrated in Figure 4-1 for a system of two equations with two unknowns.

When $\vec{f}^{(i-1)}$ is projected on the hyperplane represented by the *i*th equation to yield $\vec{f}^{(i)}$, the process can be mathematically described by:

$$\vec{f}^{(i)} = \vec{f}^{(i-1)} - \frac{\vec{f}^{(i-1)} \cdot \vec{w}_i - p_i}{\vec{w}_i \cdot \vec{w}_i} \vec{w}_i \tag{4.2}$$

where $\vec{w}_i = (w_{i1}, w_{i2}, ..., w_{iN})$. To see from where Eq.(4.2) comes from we first rewrite the first equation of (4.1) as:

$$\vec{w}_1 \cdot \vec{f} = p_1 \tag{4.3}$$

The hyperplane represented by this equation is perpendicular to the vector $\vec{w}_1$ as can be seen in Figure 4-2. The equation (4.3) states that the length of the projection of the vector $\vec{f}$ on the vector $\vec{w}_1$ has a constant length (if $|\vec{w}_1| = 1$ then p_1 is the distance of the plane from the origin).

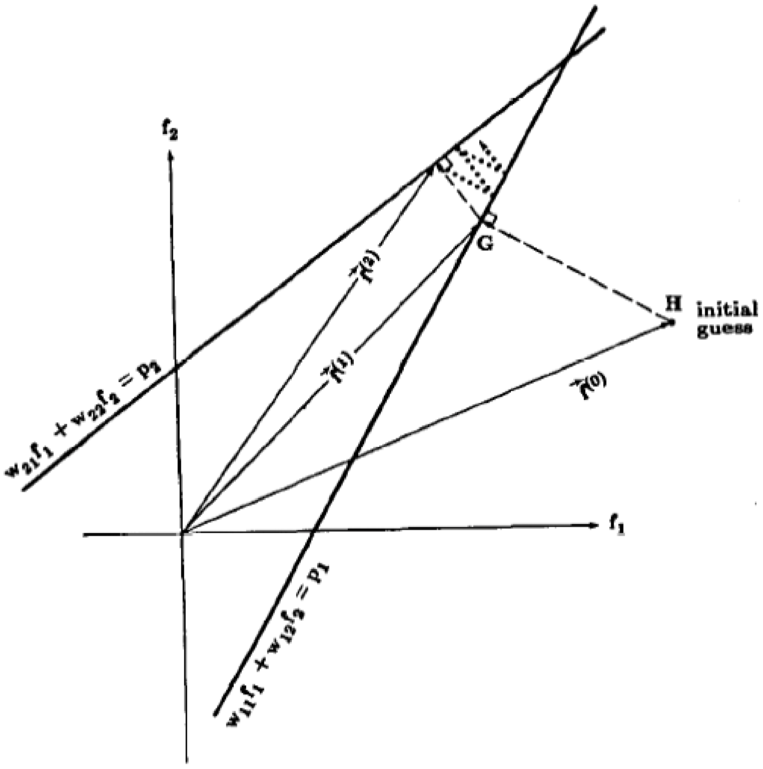


Figure 4-1. The Kaczmarz method of solving algebraic equations is illustrated for the case of two unknowns. One starts with some arbitrary initial guess and then projects onto the line corresponding to the first equation. The resulting point is now projected onto the line representing the second equation. If there are only two equations, this process is continued back and forth, as illustrated by the dots in the figure, until convergence is achieved [48].

The unit vector along $\vec{w}_1$ is given by:

$$\vec{u} = \frac{\vec{w}_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} \tag{4.4}$$

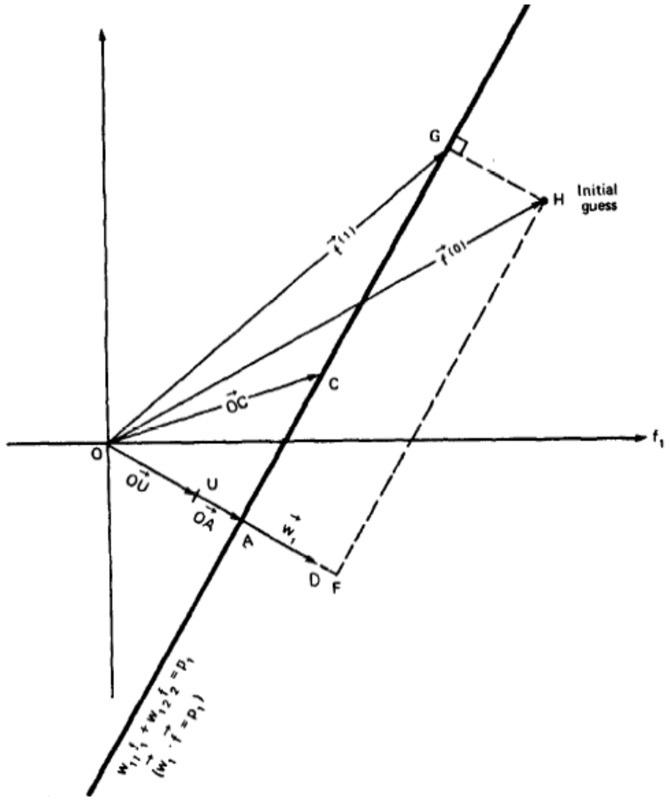


Figure 4-2. The hyperplane $\vec{w}_1 \cdot \vec{f} = p_1$ (represented by a line in this two-dimensional figure) is perpendicular to the vector $\vec{w}_1$ [48].

Then, the perpendicular distance of the hyperplane from the origin, is given by:

$$\vec{u} \cdot \vec{f} = \frac{\vec{w}_1 \cdot \vec{f}}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} = \frac{p_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} \tag{4.5}$$

To get the projection $\vec{f}^{(0)}$ on the hyperplane, we consider the distance of $\vec{f}^{(0)}$ from the plane that may be written as:

$$\vec{u} \cdot \vec{f}^{(0)} - \frac{p_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} = \frac{\vec{w}_1 \cdot \vec{f}^{(0)} - p_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} \tag{4.6}$$

Then, the projection $\vec{f}^{(1)}$ may be written as:

$$\vec{f}^{(1)} = \vec{f}^{(0)} - \frac{\vec{w}_1 \cdot \vec{f}^{(0)} - p_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} \cdot \vec{w}_1 \tag{4.7}$$

For the computer implantation of this method our initial guess at the solution $\vec{f}_0$ is a unity value to all the $f_{0,j}$. The ART iteration process can be mathematically described by the following equation:

$$\Delta f_j^{(i)} = f_j^{(i)} - f_j^{(i-1)} = \frac{\vec{f}^{(i-1)} \cdot \vec{w}_i - p_i}{\vec{w}_i \cdot \vec{w}_i} \tag{4.8}$$

where $\vec{w}_i = (w_{i1}, w_{i2}, \ldots, w_{iN})$, and the new solution $f_j^{(i)}$ is obtained from the last solution $f_j^{(i-1)}$ by the addition of the change $\Delta f_j^{(i)}$. As already stated, every iteration has a geometrical meaning: in each iteration, the solution is projected in the hyperplane represented by each equation. Each projection becomes closer to the solution if it exists.

4.2.1 CONVERGENCE AND CHARACTERISTICS OF THE SOLUTIONS

An important comment about the convergence of the algorithm is in order. If the consecutive hyperplanes have only a very small angle between them, the rate of convergence to the solution might be very slow because only a small increment $\Delta f_j^{(i)}$ is added from one equation to the next one. If equations are arranged in such way that hyperplanes are as much orthogonal as possible, the rate of convergence becomes much faster. But too much orthogonalization will also tend to enhance the effects of the ever present measurement noise in the final solution. The rate of convergence depends also on the choice of the initial guess $\vec{f}_0$.

ART reconstructions usually suffer from “salt and pepper” noise which is caused by the inconsistencies introduced in the set of equations by the approximations commonly used in the calculation of the matrix parameters. It is possible to reduce the effects of this noise by relaxation, in which we update a block by $\alpha \Delta f_j^{(i)}$ where α is less than one. In some cases it is convenient to make the relaxation parameter α a function of the iteration number; that is, it becomes progressively smaller with increase number of iterations. The resulting improvements in the quality of reconstruction are usually at the expense of convergence. Another method based on the ART may be considered in the next section 4.3 in order to minimize still more this kind of noise.

If $M>N$ a unique solution of the set of linear system in Eq. (4.1) does not exist, and, in fact, an infinite number of solutions are possible. In this case ART algorithm converges to a solution $\vec{f}_s'$ such that $\left|\vec{f}^{(0)} - \vec{f}_s'\right|^2$ is minimized. For an over determined problem, $N>M$, no unique solution can be found by ART. A not uncommon situation in image reconstruction is that of an over determined system in the presence of measurement noise. That is, we may have $N>M$ and e_j corrupted by noise. No unique solution exists in this case: the "solution" doesn't converge to a unique point, but will oscillate in the neighbourhood of the intersections of the hyperplanes.

One attractive feature of the iterative approach is that it is possible to incorporate into the solution some types of a priori knowledge about the scattering coefficients. For example, if the coefficients are known to be positive, one may set the negative components equal to zero.

4.3 SIRT ALGORITHM

The Simultaneous Iterative Reconstructive Technique (SIRT) [49] is another algorithm which eliminates the continual and competing block update as each equation is considered. Then, using the SIRT algorithm smoother and better looking reconstructions are usually obtained at the expense of slower convergence [50]. In each iteration of the SIRT algorithm, the change in each block is computed by the use of the same equations as in the ART algorithm (Eq. (4.8)), but before making any changes, all the equations are considered, and then only at the end of each iteration are the block values changed, the change of each cell being the average value of all the computed changes for that block. It is also known that SIRT algorithms perform better in extreme situations [51] such as uneven distribution of data, incomplete data, etc. It is also possible to easily incorporate constrains as positivity, limited spatial support, etc.

4.3.1 THE WEIGHT COEFFICIENTS w_{ij}

In applications like ours, requiring a large number of equations the difficulty of using Eq. (4.2) can be in the calculation, storage, and fast retrieval of the weight coefficients w_{ij}. The number of coefficients is in our case of the order of 10^9. This problem is somewhat eased by making approximations, such as considering w_{ij} to be only a function of the perpendicular distance of the ith spheroidal shell and the centre of the jth cell.

In many ART [52] and SIRT implementations to find the distribution of scattering coefficients the w_{ij}'s are simply replaced by 1's and 0's depending upon whether the centre of the jth block is within the ith spheroidal shell. As consequence extra salt and pepper noise is introduced in the reconstruction. In our calculations the width of the shell is smaller than the width of the blocks. Then it is important to calculate the fraction of volume V_{ij} of each block lying inside the ith spheroidal shell. Then, instead of using Eq. (3.8) we will use the following expression for the coefficients w_{ij}:

$$w_{ij} = \frac{V_{ij}}{\sum_j \frac{V_{ij}}{\left(r_{1,j} r_{2,j}\right)^2} \cdot \left(r_{1,j} r_{2,j}\right)^2} \tag{4.9}$$

Also, it is important to use a relaxation parameter (λ, a factor smaller than unity multiplying the increment $\Delta f_j^{(i)}$) which is commonly determined by trial and error. If incorrectly selected, will either cause premature termination and incorrect result or, if number of iterations or λ too small, will result in a reconstruction lacking high-frequency details. By trial and error we chose $\lambda = 0.01$ for about 120 iterations.

4.4 BACKPROJECTION ALGORITHM

We will start the derivation of the Backprojection algorithm [48,53] under the set of conditions as simplest as possible. These conditions will be clearly different from our problem's: two-dimensional distribution, ray emitters and transducers located over arrays around the two-dimensional area. This initial development will be derived in section 4.4.1. Nevertheless, the results that we will obtain with this derivation will be easily extended by intuitive reasoning to the geometry of our case in section 4.5. Then, using this algorithm, scattering coefficients will become a weighted average value of the residues that correspond to a certain block. This will make this algorithm to be much faster than any other iterative method. Computations times will be about 100 times smaller than the ones for ART or SIRT and no relaxation parameter will have to be chosen.

4.4.1 BACKPROJECTION ALGORITHM IN TWO DIMENSIONS WITH LINEAR ARRAYS OF TRANDUCERS. GEOMETRY AND DEFINITIONS

We start by assuming the geometry outlined in Figure 4-3 [51]. Notice that we will use two set of axis: the main set (*x*,*y*) is the original set of coordinates used to describe the object function *g*(*x*,*y*). We also have a second frame (*x*',*y*') that takes into account the direction of the beams. We consider then *g*(*x*,*y*) (from now on it will be noted as object function) to be traversed by a set of parallel beams. There is a set of transducers located on a line recording a "*parallel projection*" of *g*(*x*,*y*) on a line parallel to the *x*' axis at an angle θ from main reference frame. The coordinate systems will allow us to describe *line integrals* and *projections* in a simple fashion. Let us define both concepts rigorously.

A *line integral* will represent the integral of the function *g*(*x*,*y*) along a line. This may correspond to the total attenuation suffered by a ray as it travels in a straight line through the object function. Each line integral may be represented by a set of two parameters (θ, x') because the equation of the lines describing the beams in the figure is:

$$\mathbf{x} \cdot \mathbf{n} = x\cos\theta + y\sin\theta = x' \tag{4.10}$$

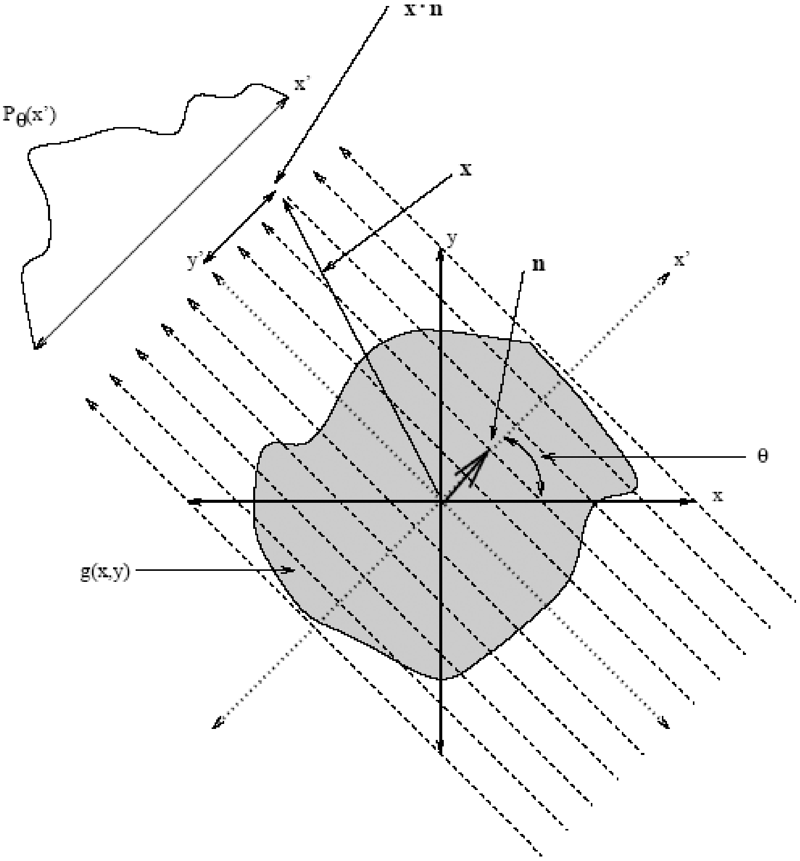


Figure 4-3. Illustration of the geometry of a set of parallel projection beams [53].

And we will use this equation to write the line integral $P_\theta(x')$ as:

$$P_\theta(x') = \int_{-\infty}^{\infty} g(x', y')\,\mathrm{d}\,y' \tag{4.11}$$

Using a delta function this can be rewritten as :

$$P_\theta(x') = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\delta(x\cos\theta + y\sin\theta - x')\,\mathrm{d}\,x\,\mathrm{d}\,y \tag{4.12}$$

The function $P_\theta(x')$ is the so-called Radon Transform of the object function *g*(*x*,*y*) (the graphic representation of the radon transform is called sinogram). A projection is formed by combining a set of line integrals. The simplest projection is a collection of parallel ray integrals as is given by $P_\theta(x')$ for a constant θ. This is known as a parallel projection.

We will show now, that if we know $P_\theta(x')$ for all possible values of θ it is possible to estimate $g(x,y)$ by simply performing a two-dimensional inverse Fourier-Transform. This is done by means of the Fourier-Slice Theorem.

4.4.2 THE FOURIER SLICE THEOREM

The Fourier slice theorem is derived by taking the one-dimensional Fourier Transform of a parallel projection and noting that it is equal to a slice of the two-dimensional Fourier transform of the original function $g(x,y)$.

We start by defining the two-dimensional Fourier transform of the object function as:

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\exp\left(-i2\pi\mathbf{x}\cdot\mathbf{u}\right)\mathrm{d}x\,\mathrm{d}y \tag{4.13}$$

where $\mathbf{u} = (u,v)$. Likewise we define the Fourier Transform of $P_\theta(x')$ as:

$$S_\theta(w) = \int_{-\infty}^{\infty} P_\theta(x')\exp\left(-i2\pi wx'\right)\mathrm{d}x' \tag{4.14}$$

Using Eq. (4.11) and Eq.(4.10) in Eq. (4.14) we obtain:

$$S_\theta(w) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x',y')\exp\left(-i2\pi wx'\right)\mathrm{d}x'\,\mathrm{d}y' \tag{4.15}$$

This two-dimensional integral may be rewritten in terms of the coordinates (x,y) as:

$$S_\theta(w) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)\exp\left(-i2\pi w\mathbf{x}\cdot\mathbf{n}\right)\mathrm{d}x\,\mathrm{d}y \tag{4.16}$$

Note that Eq. (4.16) and Eq. (4.13) are very similar. To relate them it is convenient to consider polar coordinates in Eq. (4.13):

$$\begin{aligned} u &= w\cos\theta \\ v &= w\sin\theta \end{aligned} \tag{4.17}$$

Then it is easy to write the following equality:

$$S_\theta(w) = G(w,\theta) \tag{4.18}$$

This equality has a fundamental meaning: the Fourier Transform of the projection is identical to the spectrum of the original object function on a slice normal to the direction of the projection beam. This has been illustrated in Figure 4-4:

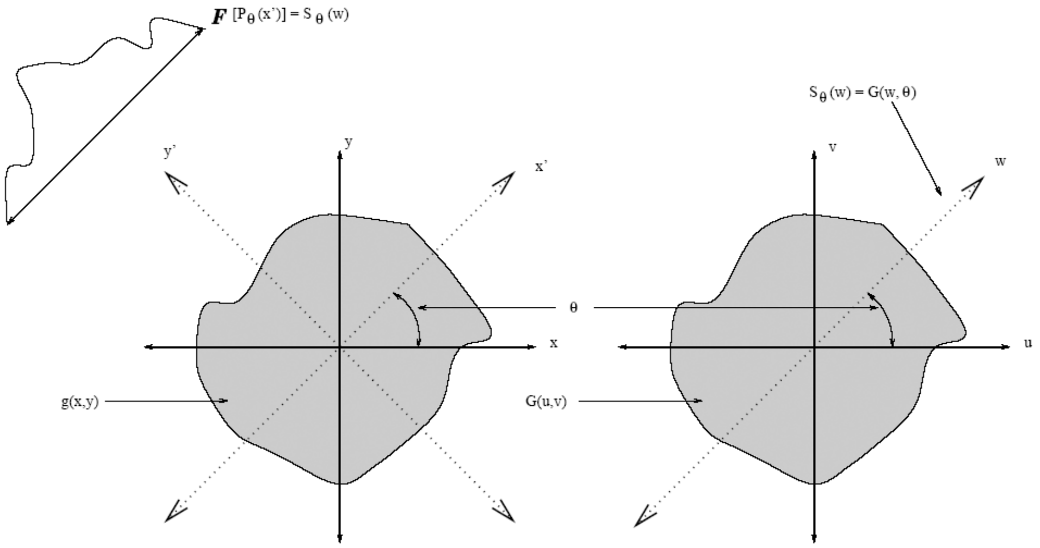

$\boldsymbol{F}\,[P_\theta(x')] = S_\theta(w)$
y'
y
x'
θ
x
g(x,y)
G(u,v)
$S_\theta(w) = G(w,\theta)$
v
w
u

Figure 4-4 Illustration of the Fourier Slice Theorem [53].

We can see then that it might be possible to recover the object function as a function of the projections.

4.4.3 RECONSTRUCTION ALGORITHM FOR PARALLEL PROJECTIONS

From Eq. (4.18) we wish now to write the object $g(x,y)$ as a function of $G(w,\theta)$. This can be done considering the inverse Fourier transform of $G(u,v)$ in Eq. (4.13) written in polar coordinates:

$$g(x,y) = \frac{1}{4\pi^2}\int_0^{2\pi}\int_0^{\infty} G(w,\theta)\exp\left(i2\pi w\{x\cos\theta + y\sin\theta\}\right)w\,\mathrm{d}\,w\,\mathrm{d}\,\theta \tag{4.19}$$

This integral can be split into two by considering:

$$\begin{aligned} g(x,y) &= \frac{1}{4\pi^2}\int_0^{\pi}\int_0^{\infty} G(w,\theta)\exp\left(+i2\pi w\left\{x\cos\theta + y\sin\theta\right\}\right) w \,\mathrm{d}w\,\mathrm{d}\theta \\ &+ \frac{1}{4\pi^2}\int_0^{\pi}\int_0^{\infty} G(w,\theta+\pi)\exp\left(+i2\pi w\left\{x\cos(\theta+\pi) + y\sin(\theta+\pi)\right\}\right) w \,\mathrm{d}w\,\mathrm{d}\theta \end{aligned} \tag{4.20}$$

Using the following property:

$$G(w,\theta+\pi) = G(-w,\theta) \tag{4.21}$$

and Eq. (4.10) we obtain:

$$g(x,y) = \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} G(w,\theta)\exp\left(+i2\pi w x'\right)|w|\,\mathrm{d}w\,\mathrm{d}\theta \tag{4.22}$$

This expression is now ready to include the information from the projections given in Eq. (4.18). Using this equation we rewrite the integral in Eq. (4.22) as:

$$g(x,y) = \frac{1}{4\pi^2}\int_0^{\pi}\left[\int_{-\infty}^{\infty} S_\theta(w)\exp\left(+i2\pi w x'\right)|w|\,\mathrm{d}w\right]\mathrm{d}\theta \tag{4.23}$$

This integral may be expressed as:

$$g(x,y) = \frac{1}{4\pi^2}\int_0^{\pi} Q_\theta\left(x\cos\theta + y\sin\theta\right)\mathrm{d}\theta \tag{4.24}$$

where:

$$Q_\theta(x') = \int_{-\infty}^{\infty} S_\theta(w)\exp\left(i2\pi w x'\right)|w|\,\mathrm{d}w \tag{4.25}$$

Eqs. (4.24) and (4.25) are the key result of this development. We will now explain the meaning of these expressions.

Eq. (4.25) represents a filtering operation over a certain projection $P_\theta(x')$. Notice that the Fourier Transform of $P_\theta(x')$ is $S_\theta(w)$ and that we are performing the inverse Fourier transform of $S_\theta(w)$ times a ramp function $|w|$. Thus, Eq. (4.25) represents filtering the projection set $P_\theta(x')$ with a filter with a frequency response given by $|w|$.

Therefore $Q_\theta\left(x'\right)$ is called “Filtered Projection”. It is very important now to understand what is $Q_\theta\left(x'\right)$ in the real space. Notice that a filtered projection $Q_\theta\left(x'\right)$, for a certain value of *x*’, assigns the same contribution to all points (*x*,*y*) lying along the projection (all points on the line $x\cos\theta + y\sin\theta = x'$). Then we say that each function $Q_\theta\left(x'\right)$ is *backprojecting* a *filtered projection.* In Eq. (4.24) the resulting projections for different angles θ are then *added* to form the estimate of *g*(*x*,*y*). We say then that Eq. (4.24) calls for each filtered projection $Q_\theta\left(x'\right)$ to be *backprojected.* Now, the name of the algorithm becomes evident.

4.4.4 IMPLEMENTING A FILTERED-BACKPROJECTION ALGORITHM FOR PARALLEL DATA.

Several problems arise when trying to implement Eq. (4.23) in a real case. First, it is only possible to obtain a finite number of projections. If the total number of projections *N* is large enough and the projections are distributed over 180° then Eq. (4.23) may be approximated as:

$$g(x,y) \simeq \frac{\pi}{N}\sum_{i=1}^{N} Q_{\theta_i}\left(x\cos\theta_i + y\sin\theta_i\right) \tag{4.26}$$

This equation calls for the filtered projections to be backprojected over the (*x*,*y*) plane. Each filtered projection makes an equal contribution to each image point (*x*,*y*), lying along a parallel projection. However in backprojecting Q_{θ_i} to a point (*x*,*y*) we need to know it for $x' = x\cos\theta_i + y\sin\theta_i$. However this value of *x*’ may not correspond to a known value of Q_{θ_i} due to the projections being discretely sampled. It is possible to find a value of Q_{θ_i} that corresponds to the image point (*x*,*y*) by interpolation. Linear interpolation is often sufficient.

Another problem arises from the filtering of the projection. Notice that the ramp filter in Eq. (4.25) enhances high frequencies making this filtering process extremely sensitive to noise. Therefore, it is necessary to use a different filter to take this into account, usually a band-pass filter. There is a wide variety of choices. Two important examples are the Hamming window:

$$H(w) = w\left(0.54 + 0.46\cos\left(\frac{\pi w}{w_c}\right)\right) \tag{4.27}$$

and the Butterworth filter:

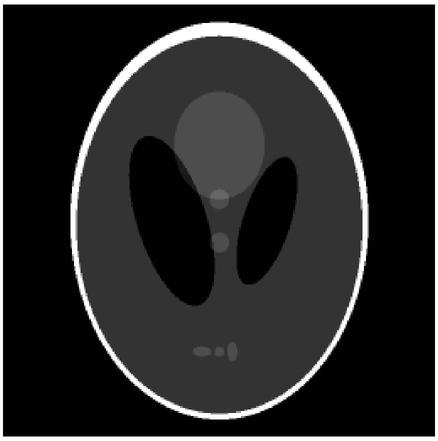
$$H(w) = w\sqrt{\frac{1}{1+\left(\frac{w}{w_c}\right)^{2n}}} \tag{4.28}$$

In the Hamming window w_c is equal to the maximum frequency the transducers can measure. In the Butterworth filter w_c is adjusted to filter the noise and allow the information of the object function to be recovered.

Notice then that the filtered backprojection algorithm is sensitive to noise. This sensitivity is due to the fact that the Radon transform is a smoothing transformation, so taking its inverse will have the effect of amplifying noise.

4.4.5 EXAMPLES

A typical example of object function $g(x,y)$ is the Shepp-Logan Phantom [54]. This phantom appears everywhere through the literature as a standard test for different reconstruction methods. It can be seen in Figure 4-5. The Radon Transform of this Phantom is represented in the corresponding sinogram in Figure 4-6. The horizontal direction corresponds to θ and the vertical direction corresponds to x'. Then in Figure 4-7, we show the reconstruction process as a function of the number of projections used using a ramp filter. We can see that for a low number of projections artefacts appear in the reconstruction image, we say that there is aliasing due to insufficient angular sampling. Those artefacts tend to disappear as the number of projections increases.



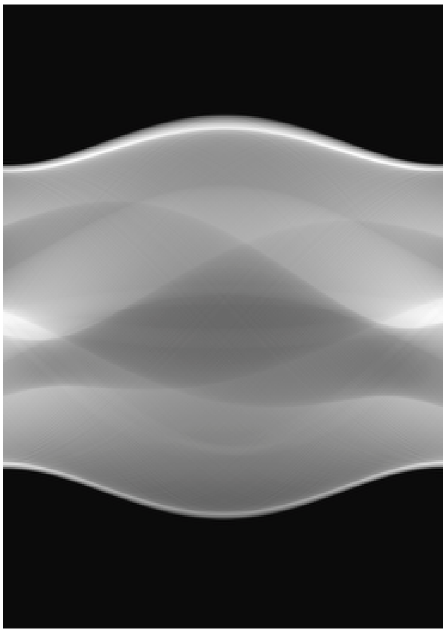


Figure 4-5. Shepp-Logan Phantom [55].

Figure 4-6. Sinogram of the Shep-Logan Phantom [55].

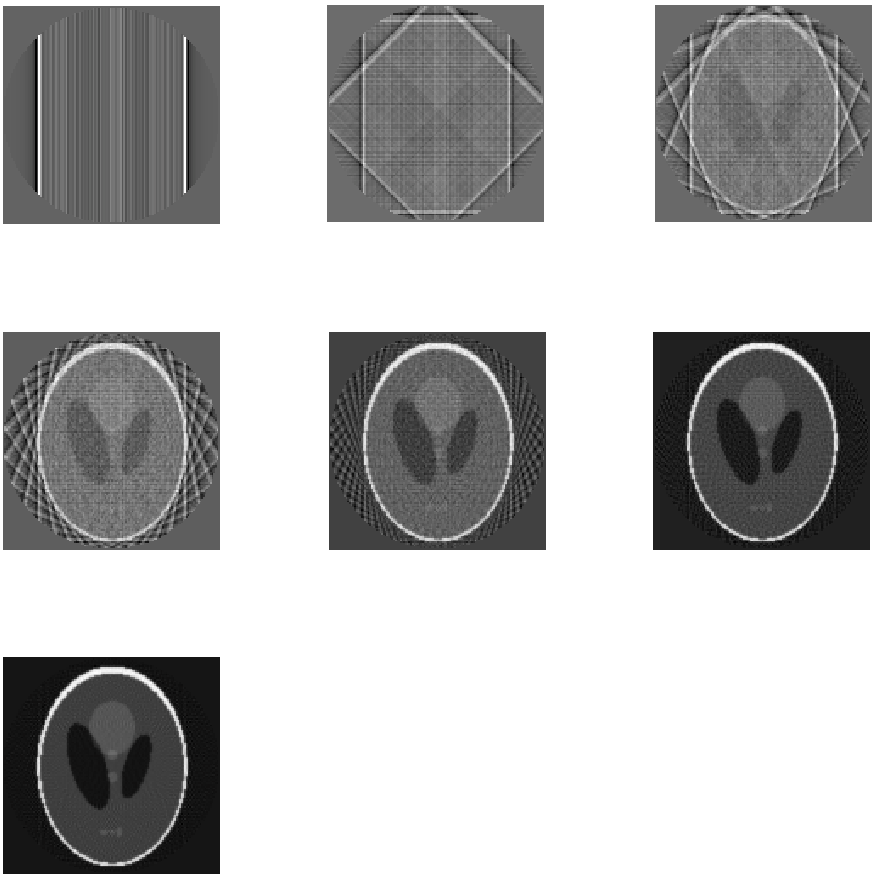


Figure 4-7. Reconstructions for (left to right up to down) *N*=1, 4, 8, 16, 32, 64 and 128 projections [55].

If the object function contains a sharp change between two regions one with important values and another with low values the aliasing effects might be even visible for a reconstruction with a large number of projections. This is shown in Figure 4-8 for *N*=256 projections. The characteristics of the reconstructed phantoms we may be considered to be analogous to reconstructions in the three-dimensional case.

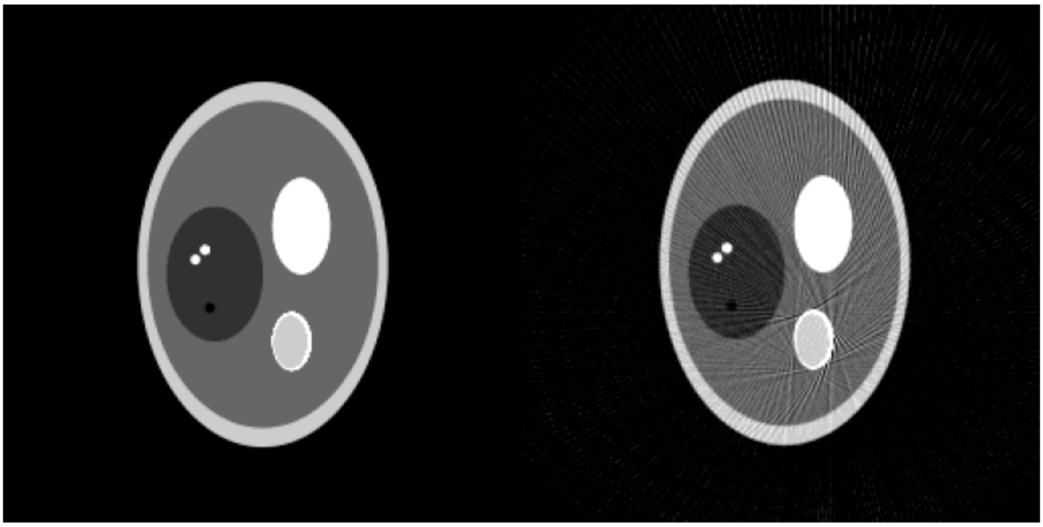


Figure 4-8. Aliasing effects [55].

4.4.6 CONNECTION BETWEEN ART AND BACKPROJECTION. A SIMPLE EXAMPLE.

ART and Filtered Backprojection algorithms seem to be very different and unconnected methods. We will show with a simple example [56] that some similarities exist between both methods. To do this let us consider a very simple example of use of the ART algorithm. Let's assume we use a scanner whose beam scans the sample only along two perpendicular axes. The two-dimensional sample area is divided into a grid of four by four pixels. The embedded object we are interested in absorbs stronger than the matrix and is placed at location [1,1] (numbering the pixels from zero to three). The figure shows the true object. In the real experiment, this is what we would ideally want to measure

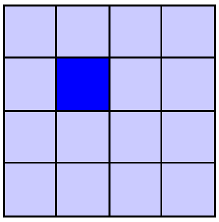


Figure 4-9. Object to be measured [56]

In the experiment, the sample is scanned horizontally and vertically at each row and column of pixels, respectively. The measured value (photon count) is lower in the row(s) and column(s) containing the absorbing object.

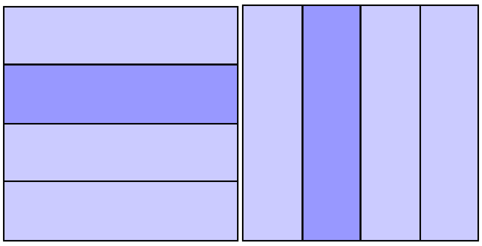


Figure 4-10. Projection process [56].

To start the reconstruction, each pixel is initialized with the average intensity collected over the whole sample area.

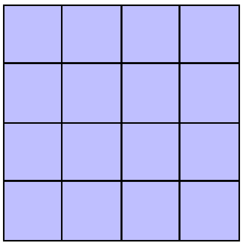


Figure 4-11. Initial Solution [56]

The first iteration takes into account the horizontal readings only. Using Eq.(4.7) the absorptivity value of each pixel is corrected by the path-length weighted absorption of its row (this is the meaning of $\vec{w}$ in this case). Note that the correction is the same one for all the pixels of the row. We may say then that the correction is being "backprojected" to the entire row.

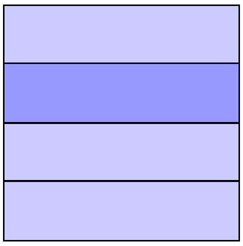


Figure 4-12 Result of the first iteration [56]**.**

The second iteration improves the image by taking into account the vertical readings. The absorptivity value of each pixel is corrected by the path-length weighted absorption of its column. Note that, as before, the correction is the same for each pixel along the beam path. Because the absorptivity of the pixels before this iteration *was different*, the corrected absorptivity is also different as can be deduced from Eq. (4.8) that takes into account the previous correction. This is a important difference between ART and FBP. In the Filtered Backprojection Algorithm each projection is considered to be an independent measurement and it is independently backprojected. In ART each backprojected "projection" depends on the others in order to try to solve the system of equations.

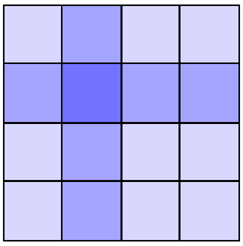


Figure 4-13 Result of the second iteration [56].

Compare the original sample in Figure 4-9 with the image obtained in Figure 4-13. Obviously, the algorithm underestimates the absorptivity of the object and also of the matrix in general, but those pixels in the matrix which are located in the same row or column as the object are coming out too dark. In effect, the object smears out along the beam paths used. The image doesn't change if more iterations are applied because the two sets of beam paths are orthogonal. What's needed is information on the sample from a variety of angles. Then the image can be updated with the information obtained from each beam. The smearing-out effects, which persist for any single beam, cancel out because the beams are no longer orthogonal. This is true for both ART and FBP.

We are going to generalize now the results we have derived for two dimensional reconstructions to 3 dimensions.

4.4.7 BACKPROJECTION ALGORITHM VERSUS ART AND SIRT

Backprojection algorithm is a computationally efficient method and it is used in most commercial medical scanners in CT applications and has proved to be extremely accurate and amenable to fast implementation. Backprojection algorithm is then much faster than ART or SIRT and it is able to provide high quality reconstructions. When the three methods are compared for CT applications, ART and SIRT have a better noise tolerance, and needs fewer projections and perform better when handling for non-uniformly distributed data sets under the condition that angles between projections are not larger than about 20 degrees. Although ART and SIRT may provide sometimes better reconstructions, they are extremely slow and then they are only used for academic purposes or to enhance the quality of a certain reconstruction if an extra amount of time is available.

When the algorithms are compared in other applications, as imaging by transmission measurements using a thermal neutrons to determine water content, cracks and homogeneity in concrete samples the conclusions are different [57]. In this case the Filtered Backprojection produces images of higher contrast, more smoothing and slightly better resolution than those obtained using iterative algebraic methods.

4.5 BACKPROJECTION ALGORITHM TO FIND 3D DISTRIBUTIONS OF SCATTERING COEFFICIENTS

Let us now write the results from section 4.4.1 in a general way. In Figure 4-3 there is an outline of a measurement process. The measured data set has been called projection. Each projection is a set of numbers. Each number corresponds to an integral of a certain property over a certain region of a certain object function. From the data of all the projections it is possible to reconstruct the original object function. To do this we only have to perform the following process. Firstly, to filter the data corresponding to a certain projection with a ramp filter or some other filter if there is a non-negligible amount of noise in the projection data. Secondly, to backproject the projections one by one. This means to assign each measured number of a certain projection to the region it was coming from. All the data corresponding to a certain pixel or block is then averaged in order to obtain the reconstruction value. From this conclusion, that has been written in a way as general as possible we may derive now an inversion process from the residuals obtained from seismograms.

Let us analyze the meaning of the set of residuals obtained from a certain seismogram. We will show now that a set of residuals from a certain seismogram is analogous to a filtered projection. Each residual corresponds to an average value of the scattering coefficient in a certain region of the space (a thin volume between two spheroidal shells). The residuals come from seismograms that have been previously filtered by a band-pass Butterworth filter. Then the set of residuals from a certain seismogram are completely analogous to a filtered projection.

To obtain the scattering distribution we will just have to backproject each residual to the corresponding spheroidal region. Finally, the scattering coefficient in a block is obtained by averaging all the residuals corresponding to that block. Block by block the full distribution is obtained. We notice here that the average will become a *weighted* average due to the fact that the geometry of our inversion problem is different. The exact expression to calculate the inversion from the residuals will be now derived.

We start by considering a set of $k = 1, 2, ..., K$ events that have happened in a certain region. The region is divided in N blocks (identified by a subindex j) and L seismometers (identified by the subindex l). The correspondence between the coefficients and the residuals are established following several steps:

i) For each earthquake k, the travelling time of the signal from the source to the jth block plus the travelling time from the jth block to each seismograph l is computed. This time will be noted as T_{jkl}. With this data we define the corresponding spheroidal surface that we note as S_{jkl}. The centre of the kth block lies on S_{jkl} and the location of the seismograph and the hypocenter are the corresponding focus of the spheroid. Note that every block defines a different spheroidal surface for the same seismograph and hypocenter.

ii) The corresponding magnitude of the residuals for each earthquake k and each seismograph l at the time T_{jkl} is computed by simple linear interpolation (we have the magnitudes of the residues only for certain times, as already stated in section 3.3). We note this magnitude as $R(T_{jkl})$.

iii) The contribution of each block is proportional to $1/\left(r_{1,j}r_{2,j}\right)^2$. This geometrical factor indicates whether the contribution of a certain block is more important or less important than the contribution of other blocks on the spheroidal surface S_{jkl}. Therefore we have to consider a normalized weighted linear combination of residues in order to compute the corresponding scattering coefficient for each *j*th block. Thus the magnitude of the scattering coefficient should be written as:

$$f_j = \frac{\sum_k \sum_l w_{jkl} R(T_{jkl})}{\sum_k \sum_l w_{jkl}} \tag{4.29}$$

The problem now is to find a suitable definition for the weights w_{jkl}. An important fact that has to be taken into account is that each weight in Eq. (4.29) corresponds to a *different spheroidal surface*. The importance of the contribution depends on the magnitude of $1/\left(r_{1,j} r_{2,j}\right)^2$ on each spheroidal surface. In order to normalize the importance of the weights for each spheroidal surface we consider that a good definition for the weights w_{jkl} in Eq. (4.29) would be:

$$w_{jkl} = \frac{\left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2}{\left\langle \left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2 \right\rangle_{S_{jkl}}} \tag{4.30}$$

where $\left\langle \left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2 \right\rangle_{S_{jkl}}$ is the average value of $\left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2$ on the surface S_{jkl}. Note that this definition (that may be considered as inspired in Eq. (3.8)) is very convenient since an analytical expression will be written for the average value. Note also that Eq. (4.30) makes the weights only depend on the time T_{jkl} and the position of the jth block, the *l*th seismograph and the *k*th hypocenter. This is a quite important point in order to perform a very fast calculation. We will now derive an explicit analytical expression for Eq. (4.30).

We start by writing the average value as a two-dimensional integral:

$$\left\langle \left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2 \right\rangle_{S_{jkl}} = \frac{1}{A_{jkl}} \int_{S_{jkl}} \left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2 dS \tag{4.31}$$

where A_{jkl} is the area of S_{jkl}. This integral is analogous to the one solved in section 2.4.1. Using the same spheroidal coordinates and taking into account that *r* is the distance between the two centres (hypocenter and station) and that we must use T_{jkl} instead of t in Eq. (2.10) we may rewrite the integral in Eq. (4.31) as:

$$\left\langle \left(1/r_{i,j}\right)^2 + \left(1/r_{i,j}\right)^2 \right\rangle_{S_{jkl}} = \frac{1}{A_{jkl}} \frac{2}{r^2} \int_0^{2\pi} \int_{-1}^{1} \frac{1}{\left(\xi_1^2 - \xi_2^2\right)} \mathrm{d}\,\xi_2 \,\mathrm{d}\,\xi_3 \tag{4.32}$$

Solving it we obtain:

$$\int_{S_{jkl}} \left(1/r_{1,j}\right)^2 \left(1/r_{2,j}\right)^2 dS = \frac{4\pi}{rvT_{jkl}} \cdot \ln\left(\frac{vT_{jkl} + r}{vT_{jkl} - r}\right) \tag{4.33}$$

We need now to write a expression for the area of the spheroidal shell S_{jkl}. There are two types of spheroidal shells, prolate spheroidal shells and oblate spheroidal shells, as can be seen in Figure 4-14. The corresponding equation for both types of spheroids is the same one:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{x^2}{b^2} = 1 \tag{4.34}$$

where a and b are the length of the semi-axis. For the prolate spheroid ($a > b$) and looks like a rugby ball and for the oblate spheroid ($a < b$) and can resemble a disk. Although the equation describing both spheroids is the same one, the expression for the area of the corresponding surface is different and care should be taken choosing the right one. We are interested in the area of the prolate spheroidal shell. It can be written as:

$$A = 2\pi a^2 + \frac{2\pi ab}{\varepsilon} \arcsin \varepsilon \tag{4.35}$$

where a and b are the major and minor semiaxes and ε is the eccentricity of the spheroidal shell and can be written as:

$$\varepsilon = \frac{\sqrt{b^2 - a^2}}{b} \tag{4.36}$$

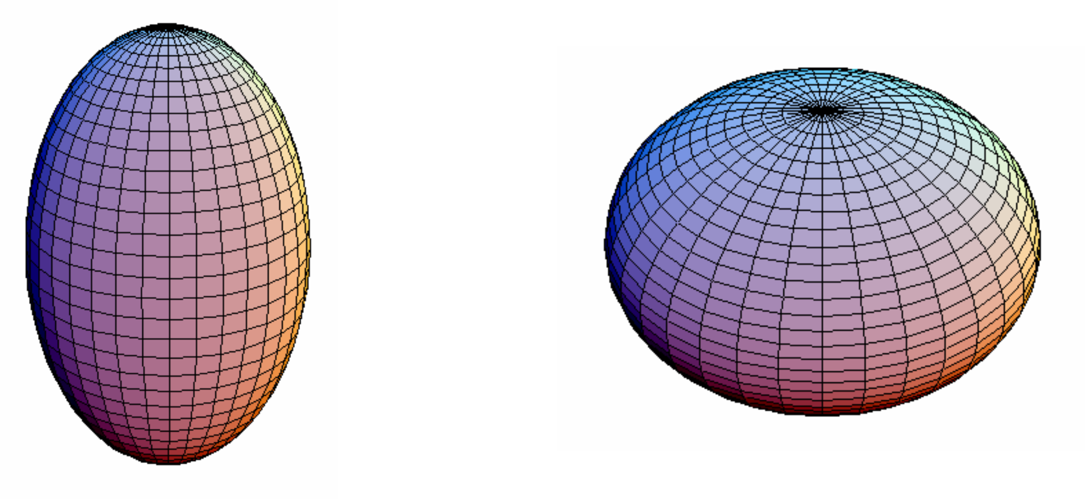


Figure 4-14. Prolate spheroid (left) and oblate spheroid (right) [58].

Taking into account the following identities:

$$a = \frac{vT_{jkl}}{2}$$
$$b = \left(\frac{vT_{jkl}}{2}\right)^2 - \left(\frac{r}{2}\right)^2 \tag{4.37}$$

and using them in Eq.(4.36) we obtain:

$$\varepsilon = \frac{r}{vT_{jkl}} \tag{4.38}$$

Combining Eqs. (4.37) and (4.38) in Eq. (4.35) we may then write:

$$A_{jkl} = \frac{\pi}{2}\left(v^2 T_{jkl}^2 - r^2 + v^2 T_{jkl}^2 \frac{\sqrt{1-\left(r/vT_{jkl}\right)^2}}{r/vT_{jkl}} \arcsin\left(r/vT_{jkl}\right) \right) \tag{4.39}$$

And by using Eq. (4.39) in Eq.(4.30), we finally obtain the expression we were looking for:

$$w_{jkl} = \frac{rvT_{jkl}}{8\left(r_{1,j}r_{2,j}\right)^2} \frac{\left(v^2T_{jkl}^2 - r^2 + v^2T_{jkl}^2 \frac{\sqrt{1-\left(r/vT_{jkl}\right)^2}}{r/vT_{jkl}} \arcsin\left(r/vT_{jkl}\right) \right)}{\ln\left(\frac{vT_{jkl}+r}{vT_{jkl}-r}\right)} \tag{4.40}$$

Note that this expression depends only on v, T_{jkl}, r (distance between hypocenter and seismometer) and $r_{1,j}$ and $r_{2,j}$.