Chapter 1: Introduction

The aim of the work package is to develop a flexible software which allows to retrieve the compressional (Vp) and the shear (Vs) velocities as function of depth from dispersion curves. They can be either obtained from array data processing (WP06) or from active source experiments.

Calculating a dispersion curve from the Vp, Vs, density and thickness of all layers is numerically possible and results in only one unique solution (forward problem). Calculating a model from a dispersion curve is much more difficult (inverse problem) as no analytical expression exists and as the same dispersion curve can be explained by various distinct models (non unicity of the solution), especially when dealing with high uncertainties as it is the case for noise measurements. Inverse methods are distributed in two main categories:

1) Direct sampling methods that use random search within the parameter space (Monte Carlo type)

2) Least-square methods that converge to the solution by means of the partial derivatives and a series of iterations for non linear cases.

Very often, the software used for the inversion of dispersion curves is Herrmann's code (St Louis, 1966) based on a least-square method. This classical approach suffers some strong disadvantages:

1) The result is only a single model supposed to be the optimum;

2) The risk for the inversion to be trapped in a local minimum is sometimes significant and depends on the starting model;

3) Manual control of the damping factor prevents from any automatic fit.

On the other hand, random search algorithms can determine a set of equivalent models having the same reproduction of the original dispersion curve. During the last ten years several of them were used in the geophysical domain:

1) Genetic algorithms (Lomax et Snieder, 1994);

2) Simulated Annealing (Sen and Stoffa, 1991);

3) Neighbourhood Algorithm (Sambridge, 1999).

During this project we make use of the Neighbourhood Algorithm and we linked it to a revised version of the forward computation written by Herrmann (1987). If the structure remains the same, the formulation from Dunkin (1965) has been re-written and inside loops have been speeded up resulting in a time consumption divided by 7 to 8. A robust automatic quality control has been added at the end to be sure that the calculated curve does not suffer from any mode jumping supposed to occur in many situations. The developed inversion code has been tested on several synthetic models with 2 or 3 layers where Vp, as well as Vs, could be correctly deduced. For curve with strong standard deviations, as it is the case for noise measurements, the non uniqueness of the model is frequent and direct search algorithms have proved to be an efficient solution.

Chapter 2: The Neighbourhood Algorithm

2.1. Parameter space

There are quite a lot of distinct manners of inverting a layered model which is characterized by a few parameters. Parameters can either be Vs, Vp, Thicknesses, Depths, Density, Poisson's Ratio, ... All selected parameters form a vector of $\boldsymbol{d}$ components. The number of components (d) determines the dimension of the parameter space, which space is a sub-ensemble of $\boldsymbol{R}^n$ containing all admissible vectors of parameters, in other words all possible 1D models. Each parameter is bounded by a minimum and a maximum value defined by the a-priori knowledge of the soil structure or by the physical properties of the materials.

2.2. Misfit function

For each individual vector (or model), the dispersion curve of Rayleigh waves is computed and compared to the observed dispersion, values obtained by noise array measurements or active source experiments. The misfit, which indicates how well the calculated response fits the measurements, is generally defined as the distance between the two responses (simulated and observed). Usually, it shows at least one minimum when the model is very close to the real soil structure. However, local minima may exist, resulting from the non uniqueness of the problem or from the noise level.

2.3. Principles of the Neighbourhood Algorithm

The early direct search methods were based on uniform pseudo-random sampling of a parameter space. The heavy computations required by this type of approach makes it impractical once the dimensionality of the parameter space increases. The solution introduced by Sambridge (1999) is motivated by the question: ”How can a search for new models be best guided by all previous models for which the forward problem has been solved (and hence the data-misfit value estimated)?”. Genetic algorithms and Simulated Annealing make use of previous samples but in rather complex ways. On the other hand, the NA is driven by only 2 tuning parameters, easy to handle.

The NA iterative procedure take the following form:

1) Construct the “approximate misfit surface” from the n_p (total number of models generated so far) previous models for which the forward problem has been solved;

2) Use this approximation (instead of actual forward modelling) together with a chosen search algorithm to generate the next n_s samples.

3) Add n_s to n_p and go back to (1).

The two items to fix are how to construct the approximate misfit surface, and how to generate new samples with it.

2.4. Voronoi cells

This is a unique way of dividing a d-dimensional parameter space into n_p regions (convex polyhedra). Each cell is the nearest neighbour region around the model points. Since the data misfit

value is known at all previous samples, the “Neighbourhood approximation” to the misfit surface is simply defined by setting the misfit to a constant inside each cell.

Sambridge has developed an algorithm for high-dimensional space that avoids the complete computation of all cell geometry, which would be a prohibitive task.

2.5. Generating samples

The key idea is to generate new samples inside chosen Voronoi cells with a local uniform density. A random walk (Gibbs sampler) is realized with a uniform probability density function inside the cell and zero outside. A walk is a sequence of perturbations to a model along all axis. The modifed model is statistically independent of the original model. Asymptotically, the samples produced by this walk will be uniformly distributed inside the cell regardless of its shape.

The cells used to generate new samples are selected by ranking them by their misfit values. The n_s new models are generated only inside the n_r best candidates. The n_r control parameter fixes the behaviour of the search algorithm: it is more exploratory if the sampling is spread over more cells and it optimizes more if the sampling is restricted to the very few best cells.

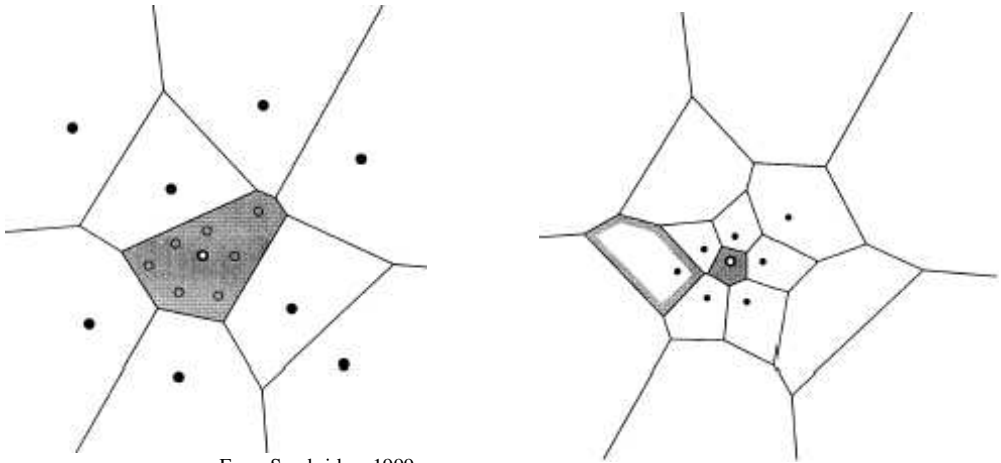

Figure 2.1: an example of Voronoi cells and how the NA can increase, decrease and shift the centre of its sampling density in parameter space. From left to right, seven new models are generated inside the central cell. On the right is the resulting new Voronoi geometry.

2.6. Appraisal of the ensemble of models

The NA technique is aimed at finding the best model to explain the data. In most cases, the information brought by the NA search cannot be summarized with just a single best model. As stated by Sambridge, the ensemble of all models (even the worst) can help to measure the resolution and the trade-off in the model parameters.

Bayesian integrals cannot be applied to the brute generated ensemble, as the density of models near the best regions depends upon the number of iterations and the type of problem. A complete re-sampling of the parameter space is hence necessary using the previously computed misfit approximation.

Though interesting to assess the resolution power of the method, e.g. versus depth, the appraisal of the ensemble has still not been applied within the SESAME project. This is one of the task we have to complete by the end of the project.

Chapter 3: Computation of dispersion curve

The computation of theoretical dispersion curves is based on the eigenvalue problem described by Thomson (1950), Haskell (1953), modified by Knopoff (1964), Dunkin (1965) and Herrmann (1987). We use the Dunkin's notations, here after and inside the source code. The Herrmann's code uses almost the same method as Dunkin.

3.1. Eigenvalue problem for Rayleigh waves

In a vertically heterogeneous, isotropic and elastic medium occupying a half-space, it is possible to define a motion-stress vector for Rayleigh waves that contains the two active displacements and their related stresses. If the displacements are of the form:

$u = r_1 \quad (k, z, \omega) e^{i(kx - \quad \omega t)}$ (radial component)

$v = 0$ (transversal component) (1)

$w = r_2 (k, z, \omega) e^{i(kx - \quad \omega t)}$ (vertical component)

the motion-stress vector is:

$$r(k, z, \omega) = [r_1 \quad (k, z, \omega), r_2(k, z, \omega), \tau_{zz}(k, z, \omega), \tau_{xz}(k, z, \omega)]^T \quad (2)$$

From the equation of motion and (1) the solution must satisfy the following differential equation:

$$\frac{d}{dz}\begin{pmatrix} r_1 \\ r_2 \\ \tau_{zz} \\ \tau_{xz} \end{pmatrix} = \begin{pmatrix} 0 & -ik & 0 & 1/\mu \\ -i\lambda/(\lambda + 2\mu) & 0 & 1/(\lambda + 2\mu) & 0 \\ 0 & -\omega^2 \rho & 0 & -ik \\ 4k^2 \mu (\lambda + \mu)/(\lambda + 2\mu) & 0 & -ik\lambda/(\lambda + \mu) & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \tau_{zz} \\ \tau_{xz} \end{pmatrix} \quad (3)$$

Parameters ρ, λ, μ are functions of the depth. For surface waves, the boundary conditions require that:

$r_1 \quad \to 0, r_2 \to 0$ when $z \to \infty$ (4)

$\tau_{zz} = \tau_{xz} = 0$ at the free surface $z = 0$

Under those conditions and for a given frequency ($\omega / 2\pi$), only a few discrete values are possible for the slownesses of the surface wave ($k_i(\omega)/\omega$), corresponding to the dispersion curves of various existing modes. This equation is solved for a stack of layers by the **propagator matrix** method (Gilbert and Backus, 1966, Aki and Richard 2002), a generalization of the Thomson-Haskell method. Equation 3 can be written in a simpler way:

$\frac{df(z)}{dz} = A(z) f(z)$ (f is a vector of n components and A a n*n matrix, n=4 for Rayleigh)

If A is independent of z, which is valid inside a layer, the solution is given by

$$f(z) = G(z, z_0) f(z_0) \quad (5)$$

where $G(z, z_0) = e^{(z - z_0) A}$ (6)

The following property can be easily deduced from equation 5:

$$f(z_2) = G(z_2, z_1) f(z_1) = G(z_2, z_1) G(z_1, z_0) f(z_0)$$

Hence, for Rayleigh waves, the motion-stress vector at depth z , inside layer n is:

$$r(z) = G(z, z_n) G(z_n, z_{n-1}) \ldots G(z_1, z_0) r(z_0) \quad (7)$$

This links the motion-stress at any depth with its value at the free surface ($z = z_0$). The propagator matrices G are functions of the relative depth inside each layer and the matrix A which depends upon layer's properties.

Equation (6) can be developed to find the elements of matrix G using an eigenvalue decomposition of matrix A (Aki and Richards, 2002). This matrix (called P in Aki and Richards) is strictly equivalent to matrix G defined in Dunkin, 1965.

In the Thomson-Haskell method, there is not constraint on the motion-stress at infinity but only a radiation condition: no upgoing waves in the bottom half-space. For P-SV waves in a homogeneous half space, it is then possible to relate the motion-stress vector at its top surface ($z = z_n$) to the amplitudes of upgoing (P_n' for P waves and S_n' for S waves) and downgoing ($P_n`$ for P waves and $S_n`$ for S waves) waves traveling across the medium.

$$\left(P_n', S_n', P_n`, S_n` \right)^T = T_n^{-1} \left(r_1(z_n), r_2(z_n), \tau_{zz}(z_n), \tau_{xz}(z_n) \right)^T$$

or $$\left(P_n', S_n', P_n`, S_n` \right)^T = T_n^{-1} G(z_n, z_{n-1}) \ldots G(z_1, z_0) r(z_0) \quad (8)$$

where,

$$T_n^{-1} = \frac{-\beta_n^2}{2 \mu_n \hat{h}_n \hat{k}_n \, omegaa^2} \begin{pmatrix} 2i \mu_n k \hat{h}_n \hat{k}_n & \mu_n \hat{l}_n \hat{k}_n & \hat{h}_n \hat{k}_n & ik \hat{k}_n \\ -\mu_n \hat{l}_n \hat{k}_n & 2i \mu_n k \hat{h}_n \hat{k}_n & ik \hat{h}_n & -\hat{h}_n \hat{k}_n \\ 2i \mu_n k \hat{h}_n \hat{k}_n & -\mu_n \hat{l}_n \hat{k}_n & \hat{h}_n \hat{k}_n & -ik \hat{k}_n \\ \mu_n \hat{l}_n \hat{k}_n & 2i \mu_n k \hat{h}_n \hat{k}_n & -i k \hat{h}_n & -\hat{h}_n \hat{k}_n \end{pmatrix} \quad (9)$$

ω is the angular frequency (rad/s);

k/ω is the slowness of Rayleigh waves;

$\hat{h}_n^2 = 2k^2 - \omega^2/\alpha_n^2$;

$1/\alpha_n$ is the slowness of P waves;

$\hat{k}_n^2 = 2k^2 - \omega^2/\beta_n^2$;

$1/\beta_n$ is the slowness of S waves;

$l_n = k^2 + \hat{k}_n^2$;

$\mu_n = \rho_n \, betaa_n^2$ is the rigidity and ρ_n is density.

The subscript n is added to all parameters defined for layer n. Putting together equations (7) and (8) and boundary conditions (4):

$$\begin{pmatrix} 0 \\ 0 \\ \grave{} \\ P_n \\ \grave{} \\ S_n \end{pmatrix} = T_n^{-1} G(z_n, z_{n-1}) \ldots G(z_1, z_0) r(z_0) = R_0 \begin{pmatrix} r_1(z_0) \\ r_2(z_0) \\ 0 \\ 0 \end{pmatrix} \quad (10)$$

This is always true when the sub-determinant ($r_{11} r_{22} - r_{12} r_{21}$ where r_{ij} are the elements of the matrix R_0) vanishes. The problem of finding the dispersion curves for Rayleigh wave is thus reduced to a root search along the slowness axis for a given frequency.

As stated by Dunkin (1965), the terms of the sub-determinant can become large. Subtracting two large numbers results in a loss of significant digits, or even reduced to zero. He proposed an alternative way of propagating motion-stress vector by the mean of the following theorem:

if $P = A_1 A_2 \ldots A_n$ then

$$p\begin{pmatrix} i & j \\ k & l \end{pmatrix} = a^1\begin{pmatrix} i & j \\ m & n \end{pmatrix} a^2\begin{pmatrix} m & n \\ o & p \end{pmatrix} \ldots a^{n-1}\begin{pmatrix} s & t \\ u & v \end{pmatrix} a^n\begin{pmatrix} u & u \\ k & l \end{pmatrix} \quad (11)$$

where,

the summed pairs of indices are to be only distinct pairs of distinct indices (by convention, m<n, o<p, ..., s<t, u<v);

$p\begin{pmatrix} i & j \\ k & l \end{pmatrix} = p_{ik}\, p_{jl} - p_{il}\, p_{jk}$, the second order sub-determinant of P.

It follows from (11) that

$$r_{11}^0 r_{22}^0 - r_{12}^0 r_{21}^0 = r^0 \left\langle \begin{matrix} 1 & 2 \\ 1 & 2 \end{matrix} \right\rangle = t_n^{-1} \left\langle \begin{matrix} 1 & 2 \\ a & b \end{matrix} \right\rangle g_n \left\langle \begin{matrix} a & b \\ c & d \end{matrix} \right\rangle \ldots g_1 \left\langle \begin{matrix} e & f \\ 1 & 2 \end{matrix} \right\rangle \quad (12)$$

With the condition on indices, the factor $t_n^{-1} \left\langle \begin{matrix} 1 & 2 \\ a & b \end{matrix} \right\rangle$ has 6 components (12, 13, 14, 23, 24 and 34). On the other hand, $g_n \left\langle \begin{matrix} a & b \\ c & d \end{matrix} \right\rangle$ has 6*6 components. The vector reduces to 5 and the matrix to 5*5 components when developing the analytical expressions of the sub-determinants of G as a function of the layer parameters. To speedup the computation, we slightly modified the Dunkin's original formulation to reduce the total number of operations, preferring subtractions and additions to multiplications and even longer divisions. To avoid overflow when computing hyperbolic sinus for large numbers, it is better to get rid of positive exponential terms in the following manner:

$$\sinh(x) = \frac{e^x + e^{-x}}{2} = e^x \frac{1 + e^{-2x}}{2}$$

We can drop a constant like e^x for every layer without affecting the overall determination of the roots of (12). We define the following adimensional real quantities:

$h_n = \hat{h}_n / k$, $k_n = \hat{k}_n / k$

$SH = 0.5 \frac{1}{h_n} (1 - e^{-2 d_n \hat{h}_n})$ and $CH = 0.5 (1 + e^{-2 d_n \hat{h}_n})$ if h_n is real

or $SH = \frac{1}{h_n} \sin(-i d_n \hat{h}_n)$ and $CH = \cos(-i d_n \hat{h}_n)$ if h_n is imaginary

$SK = 0.5 \frac{1}{k_n} (1 - e^{-2 d_n \hat{k}_n})$ and $CK = 0.5 (1 + e^{-2 d_n \hat{k}_n})$ if k_n is real

or $SK = \frac{1}{k_n} \sin(-i d_n \hat{k}_n)$ and $CK = \cos(-i d_n \hat{k}_n)$ if k_n is imaginary

where d_n is the thickness of the layer n.

$\gamma_n = 2k^2 / (\omega / \alpha_n)^2$, $a1 = \gamma_n^2 - 2\gamma_n + 1$, $a2 = h_n^2 k_n^2$, $a3 = \gamma_n^2 + a1$,
$a4 = 1 - \gamma_n$, $a5 = \gamma_n^2 a2$, $expCorr = e^{-\hat{h}_n d_n - \hat{k}_n d_n}$

And the following 2 dimensional quantities:

$c_1 = rhoô_n omegaa^2 / k$, $c_2 = 1 / c_1$

The sub-determinants of G are detailed here below ($G_{ijkl} = g_n \left\langle \begin{matrix} i & j \\ k & l \end{matrix} \right\rangle$, i before G means that this component is imaginary) :

$$G_{1212} = a3\,CH\,CK - (a1 + a5)\,SH\,SK - (a3 - 1)\,expCorr \tag{13}$$
$$G_{1213} = c_2\,(CH\,SK - h_n^2\,SH\,CK)$$
$$iG_{1214} = i\,c_2\,((a1 - \gamma_n^2)(expCorr - CH\,CK) + (a4 - \gamma_n\,a2)\,SH\,SK)$$
$$iG_{1223} = iG_{1414}$$
$$G_{1224} = c_2\,(k_n^2\,CH\,SK - SH\,CK)$$
$$G_{1234} = c_2^2\,(CH\,CK + CH\,CK + (1 + a2)\,SH\,SK)$$
$$G_{1312} = c_1\,(\gamma_n^2\,k_n^2\,CH\,SK - a1\,SH\,CK)$$
$$G_{1313} = CH\,CK$$
$$iG_{1314} = i\,(a4\,SH\,CK + \gamma_n\,k_n^2\,CH\,SK)$$
$$iG_{1323} = iG_{1314}$$
$$G_{1324} = k_n^2\,SH\,SK$$
$$G_{1334} = G_{1224}$$
$$iG_{1412} = i\,c_1\,((a1 - a4)(a4 - \gamma_n)(expCorr - CH\,CK) + (a4\,a1 - \gamma_n\,a5)\,SH\,SK)$$
$$iG_{1413} = i\,(\gamma_n\,h_n^2\,SH\,CH + a4\,CH\,SK)$$
$$G_{1414} = expCorr + G_{1423}$$
$$G_{1423} = CH\,CK - G_{1212}$$
$$iG_{1424} = iG_{1314}$$
$$iG_{1434} = iG_{1214}$$
$$iG_{2312} = iG_{1412}$$
$$iG_{2313} = iG_{1413}$$
$$G_{2314} = G_{1423}$$
$$G_{2323} = G_{1414}$$
$$iG_{2324} = iG_{1314}$$
$$iG_{2334} = iG_{1214}$$
$$G_{2412} = c_1\,(a1\,CH\,SK - \gamma_n\,\gamma_n\,h_n^2\,SH\,CK)$$
$$G_{2413} = h_n^2\,SH\,SK$$
$$iG_{2414} = iG_{1314}$$
$$iG_{2423} = iG_{1413}$$
$$G_{2424} = G_{1313}$$
$$G_{2434} = G_{1213}$$
$$G_{3412} = c_1^2\,(\gamma_n^2\,a1\,(CH\,CK + CH\,CK) + (a1^2 + \gamma_n^2\,a5)\,SH\,SK)$$
$$G_{3413} = G_{2412}$$
$$iG_{3414} = iG_{1412}$$
$$iG_{3423} = iG_{1412}$$
$$G_{3424} = G_{1312}$$
$$G_{3434} = G_{1212}$$

If $T_{ijkl} = t_n^{-1}\left\langle \begin{matrix} i & j \\ k & l \end{matrix} \right\rangle$, from (7), then T_{1214} and T_{1223} are equal and imaginary. Using equations (12) and definition of G_{ijkl} (13), it follows:

(14)

$$R_{1212}^{n-1} = T_{1212}\,G_{1212} + (T_{1213}\,G_{1312} - 2\,T_{1214}\,iG_{1412} + T_{1224}\,G_{2412} - T_{1234}\,G_{3412})/\omega^2$$

$$R_{1213}^{n-1} = \omega^2\,T_{1212}\,G_{1213} + T_{1213}\,CH\,CK - 2\,T_{1214}\,iG_{1413} - T_{1224}\,G_{2413} + T_{1234}\,G_{2412}$$

$$R_{1214}^{n-1} = \omega^2\,T_{1212}\,iG_{1214} + T_{1213}\,iG_{1314} + T_{1214}\,(2\,G_{1423} + expCorr) - T_{1224}\,iG_{1413} + T_{1234}\,iG_{1412}$$

$$R_{1223}^{n-1} = \omega^2\,T_{1212}\,iG_{1214} + T_{1213}\,iG_{1314} + T_{1214}\,(2\,G_{1423} + expCorr) - T_{1224}\,iG_{1413} + T_{1234}\,iG_{1412}$$

$$R_{1224}^{n-1} = \omega^2\,T_{1212}\,G_{1224} + T_{1213}\,G_{1324} - 2\,T_{1214}\,iG_{1314} + T_{1224}\,CH\,CK + T_{1234}\,G_{1312}$$

$$R_{1234}^{n-1} = T_{1213}\,G_{1224} - \omega^2\,T_{1212}\,G_{1234} - 2\,T_{1214}\,iG_{1214} + T_{1224}\,G_{1213} + T_{1234}\,G_{1212}$$

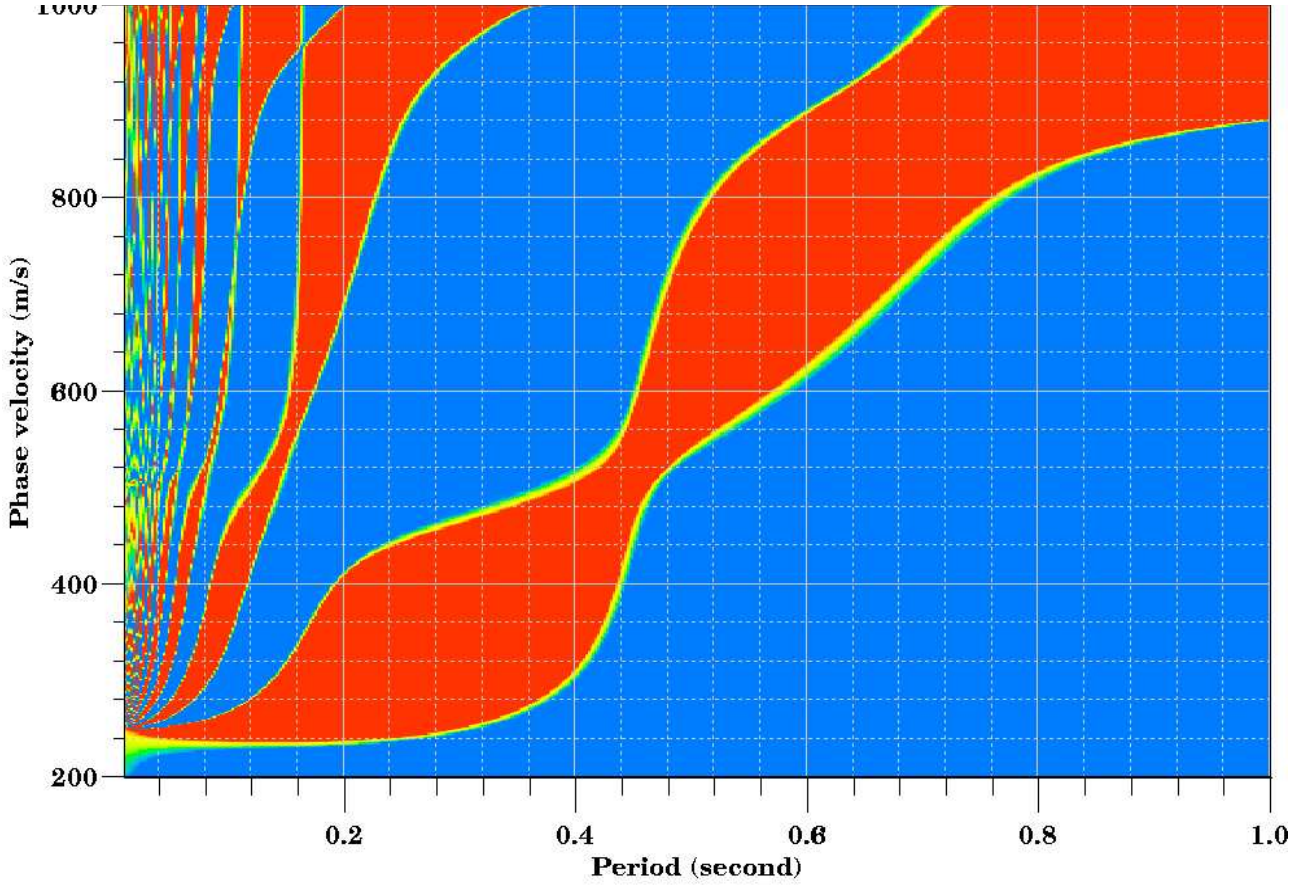

Figure 3.1: Values taken by R_{1212}^{0} at different couples $(\omega\ , k)$ for a 2 layer model (500 and 250 m/s for Vp, Vs of first layer, 50 m thick, 2500 and 1000 m/s for half-space). All values are normalized between -1 and 1. The color palette is linear from -1 to 1. We can identify all modes at the roots of the function (green-yellow).

As T_{1214} and T_{1223} are equal (from equation 9), it is obvious that $R_{1223}^{n-1} = R_{1214}^{n-1}$. From bottom to top the 2 components are always equal (equation 8) and we can reduce the number of

components to 5 rather than 6. R^{n-1}_{1214} like T_{1214} is the only imaginary component and it is preserved across the layered medium. A frequency factor has been introduced in T_{1212} to avoid unscaled vector at low frequencies.

At each step, the terms R^{n-1}_{12ab} are normalized to avoid large numbers, by multiplying by a constant that does not affect the root searching.

Compared to Herrmann's formulation in the same conditions (not in the original Fortran code but translated in C++), this implementation reduces by 25% the time consumption.

3.2. A quick root search

For a given ω , the problem is to find the roots of $R^0_{1212}(k)$ in a correct order to clearly identify the modes. All real roots are less than or equal to the maximum Vs of the model (k_{min}). We define the minimum velocity by the Rayleigh fundamental velocity of an homogeneous half-space having the Vs and Vp of the layer with the minimum Vs (k_{max}). This latter one is found by solving the following equation with a few Newton-Raphson iterations:

$$\left(\frac{\omega^2}{\beta^2} - 2k^2\right)^2 = 4k^2\sqrt{k^2 - \frac{\omega^2}{\alpha^2}}\sqrt{k^2 - \frac{\omega^2}{\beta^2}} \quad (15)$$

A first task is to identify and to bracket the first root at the lowest velocity (or the highest k). For the fundamental mode, we start from k_{max} , decreasing it by a constant step until finding a root that always exists. The initial step is defined by the half of the difference between k_{max} and $\omega / V_{s,min}$. If necessary, this step is divided by a factor 10 (see section 3.3). For higher modes, the limits of the velocity interval are reduced to k_{min} and the wave numbers of the last mode.

Once bracketed, there are several classical ways of refining a root of a non-linear or non-analytical function. Among them, the most robust is the bisection (Numerical Recipes). It always gives the correct answer if the root was correctly bracketed and if the function is continuous. In most of the situations, it is not the quickest way . As in Herrmann's code, we then implemented an algorithm that mixes the bisection and a Lagrange polynomial fit. The Lagrange polynomial is best constructed using the iterative Neville's algorithm (Numerical Recipes, chapter 3).

Practically, from the two initial bracketing values ($[k_1, R(k_1)]$ and $[k_2, R(k_2)]$), a $[k_3, R(k_3)]$ is calculated by bisection. If $R(k_3)$ is located between $R(k_1)$ and $R(k_2)$, the function is bijective and hence invertible. It is possible to construct the polynomial $P(R(k))$ whose value $k = P(0)$ is a good approximation to the root we are looking for. The brackets are modified at each iteration according to the signs of $R(k_i)$. As the brackets tend, on both sides, to the root, the degree of the polynomial increases, fitting all previously calculated points. If $R(k_1)$ and $R(k_2)$ differs from a factor 10 or more, the Neville's algorithm may fail and it is better to return to the bisection until reducing the ratio. To avoid a quick return to bisection, the relative place of the newly generated point from the true root has to be chosen with care. In our code, we consider that the calculated value is the true root, and we shift it by a tenth of the current bracket interval towards the boundary having the highest value for the function. In most cases, we

prevent returning to bisection due to a large difference between values of the function at the limits of the bracket interval.

If $R(k_3)$ is exactly null, we loose our inferior and superior brackets, and it may ruin the root search for next modes. In this case, the computation of the function is redone at a slightly distinct value (minus one tenth of the current bracketed interval).

With this method, only 4 to 6 iterations are generally necessary to obtain a good precision. The precision is initially set to 1e-7. It is divided by 10 each time the search step is reduced (see section 3.3). One iteration corresponds to one evaluation of the numerical function. In the original code written by Herrmann, the degree of the polynomial never increases over 2 or 3, quickly returning to bisection. More than 10 iterations were necessary to achieve the same precision. Our code rarely return to bisection, increasing the degree of the polynomial at each iteration. Together with the removal of all file I/O, it has been possible to drop the time consumption by a factor 5 to 6.

3.3. Mode jumping control

If the forward computation fails in finding the correct mode, the misfit value will be wrong. If the considered model is close to the solution, the error will probably disorient the inversion process. An internal check of the validity of the dispersion computation is thus necessary.

A common error is the mode jumping. While looking for the fundamental mode, the step size may be greater than the distance between the fundamental mode and the 2nd higher mode. It is often the case when computing the Love modes or higher Rayleigh modes at high frequencies. As two sign changes occur, the root is considered as bracketed and the refinement may lead to the fundamental or the second higher mode in an unpredictable manner.

Unlike Herrmann's code, the root search is always preserving the upper and lower limits of the roots. In this way, there is absolutely no risk to jump from a higher mode to previously calculated modes.

When the curves is completely calculated, we add a refined search at the lowest frequency of the frequency range, starting from the end point of the curve toward higher values of the slowness. If any root is encountered below the preceding mode or the maximum slowness, it means that some mode is missing. The computation is re-started for this mode using a smaller step as described in section 3.2.

3.4. Ellipticity computation

The ellipticity is defined by the amplitude ratio $\frac{r_1(z_0)}{r_2(z_0)}$ where $r_i(z_0)$ is defined as in equation (1). For elastic waves in a layered model, this ratio is an imaginary number either positive (prograde) or negative (retrograde). The terms come from the analogy between a rolling ball and the particle motion.

With equation (10), it easy to deduce the ellipticity from elements of matrix R.

$$\frac{r_1(z_0)}{r_2(z_0)} = -\frac{r_{12}^0}{r_{11}^0} \quad (16)$$

After the dispersion curve computation, we cannot re-construct the whole R matrix. However, it is possible to calculate the ratio (16) from sub-determinants R^0_{12ab} as shown here below.

From the computation of the dispersion curve we know that $R^0_{1212} \approx 0$. We drop the approximation and assume that the problem is perfectly solved.

$$r^0_{11}\, r^0_{22} = r^0_{12}\, r^0_{21} \quad (17)$$

$$R^0_{1213} = r^0_{11}\, r^0_{23} - r^0_{13}\, r^0_{21} \quad (18)$$

$$i\, R^0_{1214} = r^0_{12}\, r^0_{23} - r^0_{13}\, r^0_{22} \quad (19)$$

It is useful to mention that R^0_{1214} is imaginary as demonstrated by equation (16). We multiplied R^0_{1214} by i as it is internally computed as a real value. If we replace in (19), r^0_{23} with its expression calculated by (18), we obtain:

$$\frac{r^0_{12}}{r^0_{11}}\left(R^0_{1213} + r^0_{13}\, r^0_{21}\right) - r^0_{13}\, r^0_{22} = i\, R^0_{1214} \quad (20)$$

which simplifies into $\dfrac{r^0_{12}}{r^0_{11}} = i\,\dfrac{R^0_{1214}}{R^0_{1213}}$ using equation (17).

Finally, $\dfrac{r_1(z_0)}{r_2(z_0)} = i\,\dfrac{R^0_{1214}}{R^0_{1213}}$ (19)

For a half space, using equations (7) and (8), we obtain the classical formula (Tokimatsu,1997):

$$\frac{r_1(z_0)}{r_2(z_0)} = i\,\frac{k\,(l_n - 2\,k_n\,h_n)}{h_n\,(l_n - 2\,k^2)} = -\,i\,\frac{2k_n\,k}{l_n} = -\,2\,i\,\sqrt{1-\left(V_r/V_s\right)^2}\,/\left(2-\left(V_r/V_s\right)^2\right) \quad (20)$$

It is always a negative imaginary number and hence $r_1(z_0)$ and $r_2(z_0)$ are out of phase by 90 degrees with each other. The particle motion at the surface is then retrograde elliptical.

On figure 3.3 is depicted the ellipticity curve computed from the same model as for figure 3.1. If we make a zoom around the peak frequency (0.72 s., figure 3.4), the shape of the curve is very singular. This surprising behaviour probably disappears when introducing the anelasticity in the computation. This has to be tested in order to use the ellipticity in the inversion. Preliminary tests made for the Sesame project have shown that the inversion of the true ellipticity amplitude lead to incorrect models. Another scheme is under study using only the frequency of the peak but the singularities of this curve, yet present for the simplest model, render the task not straight forward.

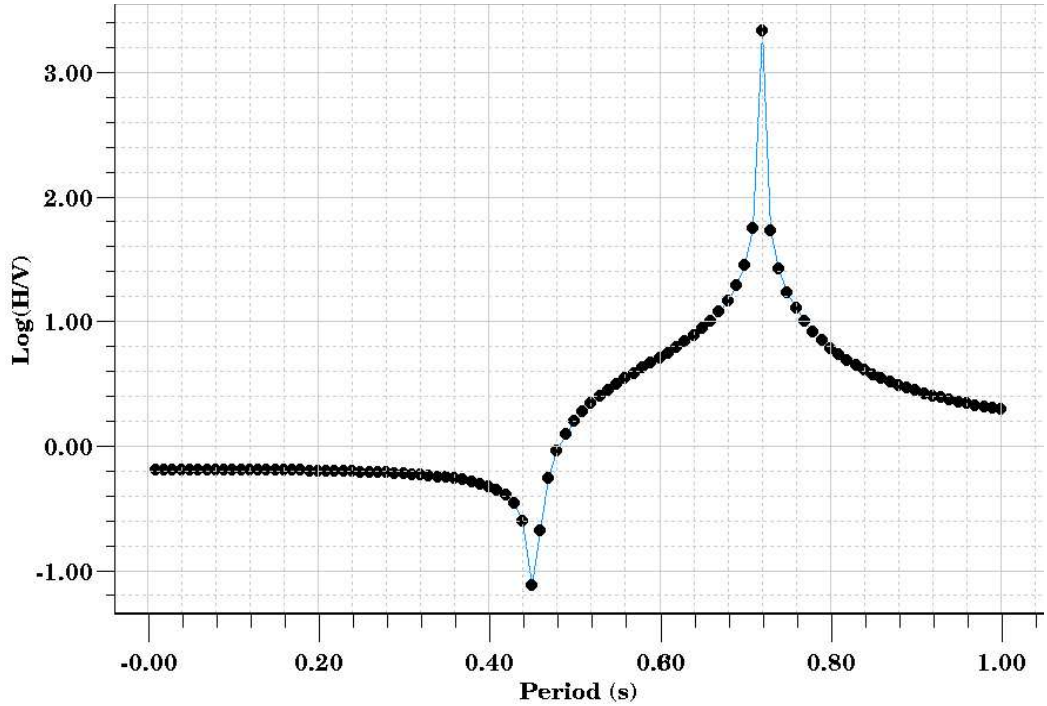

Figure 3.3: fundamental Rayleigh ellipticity calculated for the same model as figure 3.1.

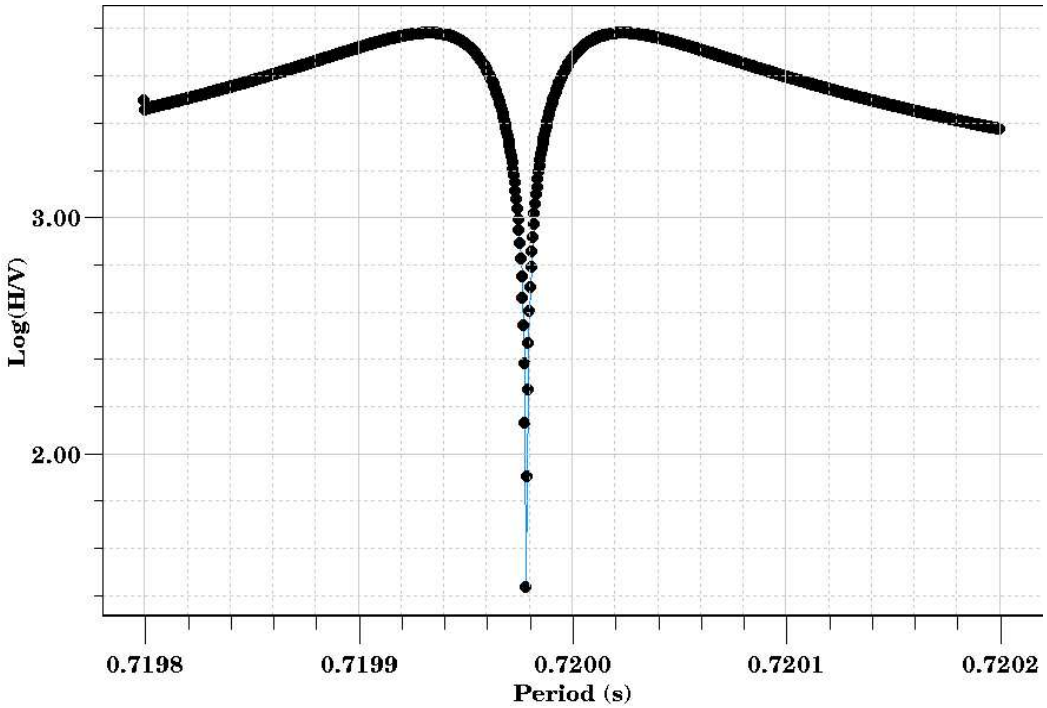

Figure 3.4: fundamental Rayleigh ellipticity calculated for the same model as figure 3.1, refined around the peak frequency.

3.5. Sensitivity of the dispersion against layer parameters

Before starting the inversion itself, we estimate the influence of all 4 parameters: thickness, Vp, Vs and density. Some parameters have a variable influence whose amplitude depends upon the value of the parameter itself and/or the value of the others.

For a single layer overlaying a half-space (figures A.1 to A.9 and A.24), the density of both layer has almost no influence on the dispersion and on the ellipticity curve (figure A.2 and A.6). However, a very small variation is observed a low frequencies for both densities with no prevailing effect of the half-space or the superficial layer. The Vp value of the half-space (figure A.7) has also a very small impact on the computed curves. Vs of the half-space (figure A.8) has a larger influence on the dispersion curve at low frequencies only. The frequency of the ellipticity peak frequency is also slightly affected. Other parameters have a strong effect on the dispersion curve, on the ellipticity amplitude and peak frequency. The thickness (figure A.1) has a frequency shift influence on the dispersion: increasing depth move the dispersion towards lower frequencies, with only small changes in the curve's shape. The velocities at very high and very low frequencies are not modified by the deepening of the sediment layer. On the other hand, slight modifications of the Vs (figure A.4) change the Rayleigh velocity at high frequency but change also the shape of the curve in the same way as the thickness. The velocity at low frequency is not affected by the velocity of the sediment layer.

For 2 layers overlaying a half-space (figures A.10 to A.23), the same conclusions as above are also valid. One outstanding difference is the influence of the density of the first layer (figure A.11). The densities of the half-space and of the 2nd layer have no influence as for 2 layer case.

When it tends to 0.5, the Poisson's Ratio reduces the effect of Vp of the superficial layers. For a 2 layer model, it is obvious when comparing figures A.3 and A.24 (Poisson's Ratio close to 0.5). The phenomenon is also observed for more complex structures. When inverting both Vs and Vp, the obtained Vp will be generally only a minimum of the possible values. The invertible parameters are thus the thickness of superficial layers, as well as their Vs and Vp values. Inversions with a more constrained Vp is usually helpful.

3.6. Misfit definition

In the inversion program, the misfit is calculated using two different ways according to the type of data used in the inversion: data with or without uncertainty. In both case the values are relative and never absolute. To save disk space storing computed dispersion curves, all modes share the same set of frequency samples. But for certain reasons, either physical (e.g. higher modes do not exist at all frequencies) or user wishes (limited frequency range, distinct ranges for modes, ...), not the whole set of samples is used to compare the data with the calculated curve. This number is called the “data count” or n_D . On the other hand, when calculating the curve, the valid frequency range may be slightly different than for the true data (e.g. when deepening the model the frequency cut-off for the higher modes is decreasing toward lower frequencies) . The number of common points is called “real count” or n_R .

A factor is added in front of the square root (equations 20 and 21) to include, as a misfit criteria, the number of points where the comparison was possible. Without it, a very good misfit can be achieved by fitting only one point of the data curve and ignoring the others if the validity ranges of calculated and measured curves are very different, with only one point in common. If n_D equals to n_R , expressions (20) and (21) reduce to a classical Root Mean Square formula.

If the data are given with an uncertainty, the following formula is used:

$$misfit = (1 + n_D - n_R)\sqrt{\sum_{i=0}^{n_D} \frac{(x_{di} - x_{ci})^2}{\sigma_i^2 n_R}} \quad (20)$$

If the data are given without uncertainty, the following formula is used:

$$misfit = (1 + n_D - n_R)\sqrt{\sum_{i=0}^{n_D} \frac{(x_{di} - x_{ci})^2}{x_{di}^2 n_R}} \quad (21)$$

Where,

x_{di} is the measured value for frequency ω_i

x_{ci} is the calculated value for frequency ω_i

σ_i is the standard deviation for frequency ω_i

The terms where x_{di} or x_{ci} are not valid are removed from the sums. n_R is always less than n_D .

These formulas are used to compare the dispersion curves, the ellipticity amplitudes and the autocorrelation curves. For higher modes and the for the former ones, the number of common points is of real importance. For multi-modal fitting, e.g. inverting the fundamental mode with the 1st higher mode together, the total misfit has the same form as the preceding formulas (without uncertainty, replace σ_{mi} by x_{dmi}):

$$misfit = (1 + n_D - n_R)\sqrt{\frac{1}{n_R}\sum_{m=0}^{n_M}\sum_{i=0}^{n_D} \frac{(x_{dmi} - x_{cmi})^2}{\sigma_{mi}^2}}$$

where,

n_R is the total number of valid points (both for calculated values and data) across all modes;

n_M is the number of modes considered;

n_d is the total number of valid points for data across all modes.

To perform a joint inversion of ellipticity amplitude and dispersion the misfit is the weighted sum of the two misfits calculated separately:

$$misfit_{total} = (1 - \eta)\, misft_{dispersion} + etaa\ misfit_{ellipticity}$$

Where η is defined by the user (usually 0.5).

This rule also applies to sum a misfit calculated with a dispersion curve and the misfit obtained from comparing the calculated Vp profile versus depth with an a-priori Vp profile.

Chapter 4: 1D Synthetic data tests

In this chapter, the inversion code is tested using synthetic data, which are dispersion curves, ellipticity or the autocorrelation curves calculated with the forward code implemented in the inversion program. We make then a clear difference between synthetic data and synthetic signals which are full time series computed from numerical modeling techniques.

4.1. One layer over a half space

We propose to invert the model depicted on figure B.1 composed of 2 layers: a sediment layer overlaying an infinite half-space. The parameters to invert are the thickness of the sediments, their Vp and Vs, and the Vp and Vs of the bed-rock (or the bottom half-space). The fundamental and 1st higher mode dispersion curves, as well as the ellipticity curves are plotted on the same figure.

4.1.a. Fundamental mode

In order to simulate a real dispersion curve, it is necessary to reduce the frequency range. Due to the high pass filtering effect of the soil structure on the vertical component of the motion (Scherbaum et al. 2003), it is very rare to measure the phase velocity at frequencies below the frequency of the H/V peak or the ellipticity peak. The high frequency limit is linked to the wave length of surface waves and the depth of interest (depth of the interface). We decided to cut the dispersion curve around the point where no more dispersion is observed (constant curve, figure B.1). Thus, the fundamental curve to fit ranges from 0.1 to 0.5 seconds.

The a-priori intervals for the chosen parameters are given in the following table:

Parameter	*Minimum*	*Maximum*
Thickness0	1 m	50 m
Vs0	20 m/s	707 m/s
Vp0	200 m/s	1000 m/s
Vs1	21 m/s	3535 m/s
Vp0	210 m/s	5000 m/s

For all layers except the first one, it is possible to force Vp to increase with depth from one layer to the next. Vp is said incremental and the defined parameter is not Vp itself but the difference in velocity between a layer and the layer just above.

Choosing Vs and Vp randomly and individually leads to a number of non physical models with a Poisson's Ratio over 0.5. In the inversion program, the Poisson's ratio must stand between 0 and 0.5 which corresponds to a ratio Vs/Vp between 0 and 0.706. To take this physical relation between the 2 velocities into account, it is better to define the parameters as Vp and Vs/Vp. Vs/Vp is better than Vp/Vs because the latter is not limited to a particular value, as it ranges from 1.4 to infinity (in case of water). Hence, the list of parameters and their a-priori interval is transformed into:

Parameter	*Minimum*	*Maximum*
Thickness0	1 m	50 m
Vp0	200 m/s	1000 m/s
Vs0/Vp0	0.1	0.707
Vp1-Vp0	10 m/s	4000 m/s
Vs1/Vp1	0.1	0.707

As the Neighbourhood Algorithm is based on a random search across the parameter space, a random generator must be initialized by a seed. A seed is an integer value that will give a unique series of pseudo-random numbers different for every seed. The series of numbers have a uniform probability law. In order to check the stability of the inversion, it is needed to start at least 2 processes with distinct seeds. In this case, 3 were launched at the same time. The results are depicted on figures B.2 to B.5.

On figure B.2, the best models perfectly match the theoretical Vs profile. Vp value within the sediment is correctly retrieved as well. According to observations made in chapter 3, Vp, and to a least extent, Vs in the basement are not found with a good resolution. This bad resolution is due to the simulated high pass filtering effect of the soil structure. On the contrary, a very good resolution is obtained when using the complete dispersion curve, down to 1 Hz in this case (not shown here).

The figure B.5 is a good example of an inversion trapped in a local minimum. It is one of the 3 processes merged in figures B.2 to B.4 (blue line). In a complex and ill posed problem, it necessary to explore the parameter space using a series of distinct processes to not miss some good solutions.

Rather than using the thickness of each layer, it is possible to set the depth of each interface. In the case of two layers it is not worth doing it as the thickness is numerically the same as the depth. This option is preferably tested on a 3 layer model.

4.1.b. First higher mode alone

For the 1st higher mode, we suppose that it is possible to measure it between 4 and 20 Hz, cutting it just before the velocity rise. The parameter is the same as for the fundamental mode inversion. 3 runs are simultaneously launched and their results are summarized on figure B.6. For Vs and Vp of the sediment layer, inverting the fundamental of the 1st higher mode seems to be equivalent. The content of information is almost the same in both cases, located at distinct frequency ranges.

4.1.c. Fundamental and 1st higher modes together

The curves of the two modes have been used together to invert both modes. The results are shown on figure B.7. Even if the constraint seems to be better on Vp, comparing figures B.2 and B.7. No quantitative conclusion can be drawn as the statistical dispersion deduced from plots depends upon arbitrary parameters: the cut-off misfit for models to plot and the color scale. As the way the misfit are calculated differs in paragraphs a,b and c, it is not easy to define a common reference. The final a-posteriori covariance matrix would be the only objective way of comparing the results of the various approaches. However, its computation is still not working at the present time.

4.1.d. Case of a low Vs/Vp ratio

As mentioned in paragraph 3.5, when the Poisson's ratio tends to 0.5, Vp looses its influence on the dispersion curve. It is thus interesting to measure the final effect on the inversion. The theoretical dispersion curves for the fundamental and the 1st higher modes are depicted on figure B.8.

The a-priori interval for parameters must be revised as follow:

Parameter	*Minimum*	*Maximum*
Thickness0	1 m	50 m
Vp0	200 m/s	2000 m/s
Vs0/Vp0	0.1	0.707
Vp1-Vp0	10 m/s	4000 m/s
Vs1/Vp1	0.1	0.707

At the frequency of the peak of the ellipticity, the phase velocity is very close to the Vs of the basement. Hence, cutting at the peak frequency induce a better constraint on the Vs profile than in the preceding cases. On figure B.9, it is clear that the Vs profile is perfectly retrieved along the whole soil column. Unlike former cases, the Vp profile is not correct even for the superficial layer, corroborating conclusions of paragraph 3.5.

4.2. Two layers over a half space

We propose to invert the model depicted on figure B.10 composed of 3 layers: 2 sediments layers with almost the same properties overlaying an infinite half-space. A small difference between the properties of the 2 layers was chosen rather than a high impedance contrast, to test the performance of the implemented algorithm. The parameters to invert are the thickness of the sediments, their Vp and Vs, and the Vp and Vs of the bed-rock (or the bottom half-space).

4.2.a. Fundamental mode

The dispersion curve is cut at 0.45 seconds, corresponding to the peak of the ellipticity.

The list of parameters and their a-priori interval is described in the following table:

Parameter	*Minimum*	*Maximum*
Thickness0	1 m	20 m
Vp0	200 m/s	500 m/s
Vs0/Vp0	0.1	0.707
Thickness1	1 m	30 m
Vp1-Vp0	1 m/s	500 m/s
Vs1/Vp1	0.1	0.707
Vp2-Vp1	10 m/s	4000 m/s

Parameter	*Minimum*	*Maximum*
Vs1/Vp1	0.1	0.707

Three different runs were used to construct the figure B.11. If the Vs values for the 2 superficial layers are correctly re-calculated, the depth of the 2 contacts are not well constrained. Only Vp at the surface is reasonably retrieved.

In the preceding parametrization, it is difficult to add a-priori information on the depth of the sedimentary basin which is sometimes reached by boreholes. Thus, we apply the depth parametrization rather than the thickness used so far. Parameters thickness0 and 1 are replaced by depth0 and 1:

Parameter	*Minimum*	*Maximum*
Depth0	5 m	20 m
Depth1	20 m	50 m

The results are depicted on figure B.12. The tuning parameters of the Neighbourhood Algorithm have been changed to behave more like an explorer (Nr=50 rather than 25 before). Comparing it to figure B.11, the thickness or depth parametrization is equivalent. The only remarkable change is to way the parameter space is sampled. On figure B.11, some solutions are deeply (generating a lot of model close to the solution) and quickly investigated whereas some parts are clearly avoided. On figure B.12, depths varying from 28 to 34 m are almost equally sampled but it would have taken a longer time to achieve the same misfit as in B.11. If the problem is ill-posed or badly constrained, it is clear that it is better to explore the parameter space than optimizing directly one solution.

In earthquake engineering, one application of this method is the computation of the theoretical response of the site. As we observe a non negligible uncertainty in the inversion results, it is interesting to estimate the resulting uncertainties on the 1D SH site response. Figure 4.1 compares the response of the true model and the models plotted on figure B.12.

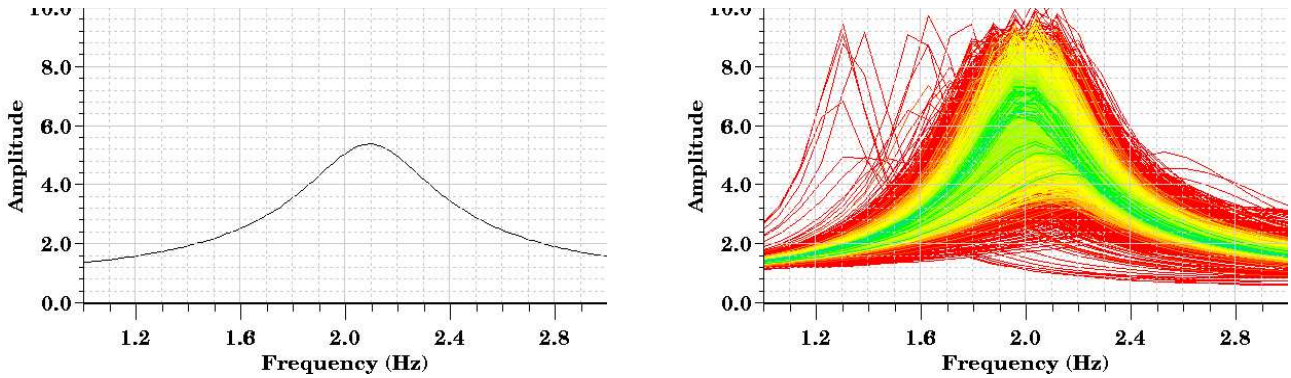

Figure 4.1: comparison of theoretical SH response for a vertical incidence for the true model and the inverted models of figure B.12. Qs is 25 in the soft layers and Qp is 100. The color scale ranges from 0.004 to 0.05 (misfit value) with a geometrical step.

For the best models, the peak frequency is almost retrieved but not exactly, between 2 and 2.2. A great statistical dispersion is observed on the amplitude ranging from 4 to 8 (green models).

4.2.b. Fundamental mode – a-priori depth of the basement

As concluded in the last paragraph, the dispersion curve itself is not sufficient to correctly lead the inversion to a unique and well constrained solution. In the framework of the depth parametrization, it possible to fix or to reduce the interval of the depth of whatever layer. In practical cases, the depth of the basement can be known from borehole data, for example.

The ensemble of models with the bedrock depth fixed are plotted on figure B.13. The whole Vs profile is well calculated even for the basement layer (a bad resolution is achieved without the fixed depth, section 4.2.a). Though a very small impedance contrast, the contact at 10 m is also retrieved with a very good precision (about 50 cm of error, 5%). The SH response is presented on figure 4.2.

Figure 4.1 and 4.2 can be compared as the color scales are identical and the way the misfit is computed is also the same. All green models are located at the same distance from the true dispersion curve in both cases. If the green category (misfit less than 1%) ranges from 4 to 8 in figure 4.1, it reduces to [4.2,6.4] for figure 4.2. The peak frequency is also better located at 2.1 for the best models (dark blue). Including a-priori information when available can then strongly improve the inversion of the dispersion curve. It is useless to introduce doubtful information as a good fitting set of models will very often be found, but probably presenting a systematic difference with the reality.

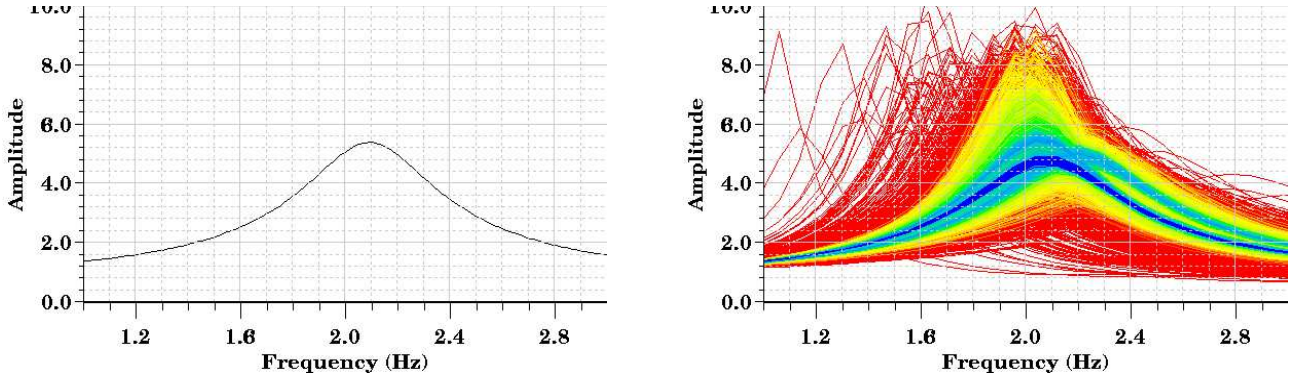

Figure 4.2: comparison of theoretical SH response for a vertical incidence for the true model and the inverted models of figure B.13. Qs is 25 in the soft layers and Qp is 100. The color scale is close to B.13 and the same as figure 4.1.

4.2.c. 1st higher mode

As shown by figure B.10, the small contrast of Vs values has a greater influence on the 1st higher mode curve than on the fundamental curve. We would like to compare the inversion of the 1st higher alone, without depth constraint to the one of the fundamental mode, particularly at finding the depth of the impedance contrast.

The results of 3 runs are displayed on figure B.14. Refined view of the contrast at 10 m is shown on figure 4.3. Though color and misfit scales are different, it is clear that the inversion of the 1st higher mode better constrains the problem.

4.2.d. Fundamental and 1st higher modes together

The dispersion curve the fundamental mode and the one for the 1st higher mode are inverted together. They were re-sampled to set as many common points as possible inside the common range (from 0.05 to 0.25 seconds) and to limit the number of points on the fundamental mode to a maximum of 30 (exactly 27 after cutting it below 0.05 second).

The results are depicted on figure B.15 and include the models generated by 6 distinct runs. Combining both modes leads to better Vs and Vp profiles than single mode inversions.

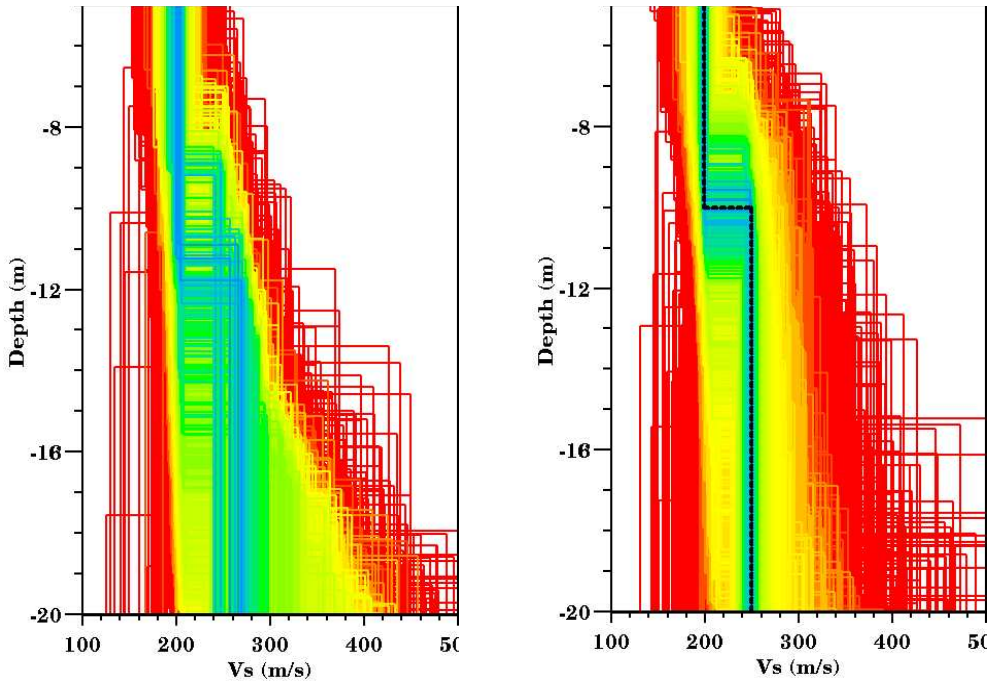

Figure 4.3: precision obtained for the contact at 10 m for the fundamental mode inversion (left) and the 1st higher mode (right). Color scales are different but misfit computation also.

4.3. Power law variation inside a layer

4.3.a. Introduction

In the preceding cases, the layers were supposed to have an homogeneous velocity (both Vp and Vs). We introduce a power law variation across each layer (except for the bottom half-space), as proposed by Scherbaum et al. (2003) for sediments of the Cologne area (Germany). It was based on an earlier regional study and is linked to the compaction of those sediments. The velocity inside the layer is defined by the velocity at the top of the layer and an exponent α, using the following formula:

$$V = V_0\left((z+1)^{\alpha} - (z_0+1)^{\alpha} + 1\right) \quad (1)$$

Where,

z_0 is the depth of the top of the layer,

V_0 is the velocity at the top of the layer,

z is whatever depth inside the layer, $z > z_0$,

α is ranging between 0 and 1.

The second term is designed to force the velocity to be equal to V_0 when $z = z_0$ even if z_0 is greater than 0. If only Vs is fixed, Vs is calculated with the power law equation and Vp is deduced from Vs using the Vs/Vp value. In all other cases, Vp is calculated with the power law equation and Vs is deduced from Vp using the Vs/Vp value. The exponent α is a sensitive parameter because, in most cases, it generates a number of models with unusual velocities (over 5000 m/s) and induces a lot of velocity inversions in the Vp or Vs profile that slow down the inversion process. It is best to choose the velocity variation across the layer as a free parameter. We call it “Vp change” (or “Vs change”) in the program interface, and dV here. If z_n is the depth of the bottom of the layer, the velocity at z_n is $V_n = V_0 + dV$.

V_n is related to α with :

$$V_n = V_0\left((z_n + 1)^\alpha - (z_0 + 1)^\alpha + 1\right)$$

It can be solved using bisection with α between 0 and 1 in the following function:

$$f(\alpha) = (z_n + 1)^\alpha - (z_0 + 1)^\alpha - \frac{V_n - V_0}{V_0}$$

It is monotonously increasing and have only 1 root between 0 and 1. Other iterative methods are not appropriate (for instance, Newton-Raphson even with $\ln(f)$).

The dispersion computation is based of the Thomson-Haskell propagator methods. No arbitrary function of the velocity with depth can be introduced without discretizing it along the depth axis. Each layer with a power law variation is thus internally transformed into a stack of uniform velocity layers, reproducing the functional variation. The user can choose the number of sub-layers used to represent the function (usually 5 to 10). The time needed for computation is increasing with the total number of sub-layers in the parametrized model.

If the thicknesses of the sub-layers are constant, the power law variation is badly sampled, resulting in a very high velocity jump for the first sub-layers. Thus, it is better to impose a constant velocity jump from one sub-layer to the next, equal to dV/n . The depth of the top and bottom of each sub-layer is then easily calculated: $z_0, \ldots, \left(i\frac{dV}{V_0} + (z_0 + 1)^\alpha\right)^{1/\alpha} - 1, \ldots, z_n$ (2)

Inside each sub-layer, for the sake of simplicity, we set the velocity of the sub-layer to the value of the analytical power law function at the middle of the sub-layer. Hence,

$$V_i = V_0\left[\left(\frac{z_{i-1} + z_i}{2} + 1\right)^\alpha - (z_0 + 1)^\alpha + 1\right], i = 1, \ldots, n \quad (3)$$

To summarize, from the thickness of the layer and dV, we are able to define in a unique way the the individual thicknesses of each sub-layer (equation 2) and their velocities (Vp or Vs according to the nature of V_0, equation 3). An intermediate computation is necessary to obtain the value of the exponent α. As only the resulting profile is stored, to display the model parameters once the inversion is finished, the value of α has to be recalculated for each model from the thicknesses and the velocities of the 2 first sub-layers. An equation of the same type as (2) has to be solved.

$$g(\alpha) = (z_2+1)^\alpha - \frac{V_2}{V_1}(z_1+1)^\alpha + (\frac{V_2}{V_1}-1)(z_0+1)^\alpha = \frac{V_2}{V_1} - 1 \quad (4)$$

$g(\alpha) - (\frac{V_2}{V_1}-1)$ is also monotonously increasing and have only 1 root between 0 and 1. It is solved by bisection.

The parameters for a layer with gradient are Vp (V_0), Vs/Vp, dV, and the thickness (or depth of the bottom of the layer). The number of sub-layers is just a tuning parameter.

4.3.b. Synthetic data

The model chosen for the test is depicted on figure B.16. It is composed of 2 layers overlaying a half-space. Inside the superficial layers, the velocity Vs and Vp vary as a power law with depth. Each layer is divided in 20 equally thick sub-layers. Vs at the top of the 1st layer is 250 m/s, 1500 m/s at the top of the 2nd layer and 2500 m/s for the bottom layer. The exponent for layer 1 is 0.3 and 0.15 for layer 2. The ratio Vs/Vp and the density increase with depth from 0.4 (Vs/Vp) and 2 (density) in the 1st layer, 0.45 in the 2nd layer and 0.5 in the basement. To measure the effect of the discretization, the dispersion and the ellipticity curves are computed with thin sub-layers (0.5 m, figure B.17). At high frequencies, the dispersion curve is clearly affected as the refined curve is strongly changed near the surface. A more interesting effect can be evidenced on the ellipticity curve for which the peak frequencies are apparently shifted. If we compare the theoretical SH response for the 2 discretizations, we can observe that the main features are present in both cases but the amplitudes are the same (figure 4.4). We probably measure here the effect of the badly sampled superficial velocities.

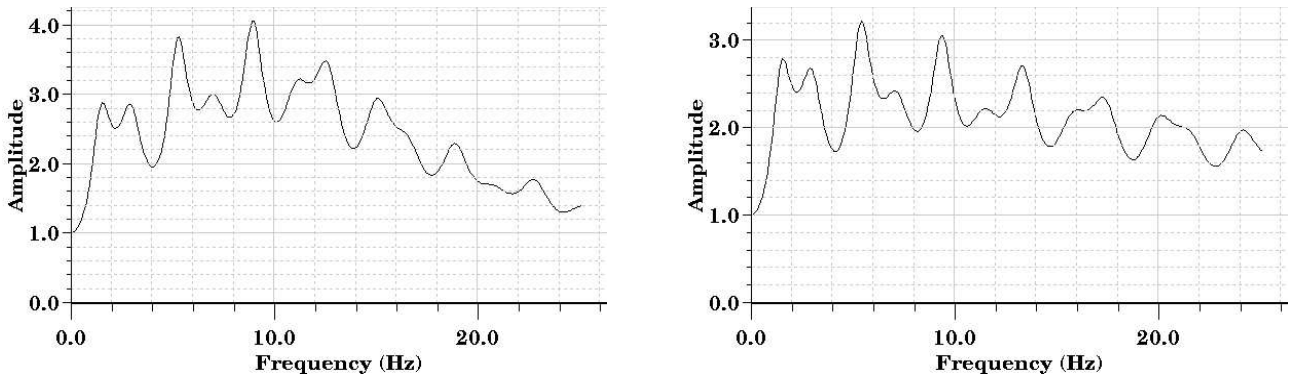

Figure 4.4: comparison of theoretical SH response for a vertical incidence for the two discretized models (41 sub-layers on the left or figure B.16, 601 sub-layers on the right or figure B.17). Qs is 25 in the soft layers and Qp is 100.

We propose to invert the fundamental mode of both cases using the proposed gradient parameterization and fixing the depth of the basement as if it was measured in a borehole to 300 m. The dispersion curve is cut at the ellipticity main peak frequency, i.e. 0.45 seconds.

The list of parameters and their a-priori interval is described in the following table. The number of sub-layers is set to 5 in both layers.

Parameter	***Minimum***	***Maximum***
Depth0	5 m	150 m
Vp0	200 m/s	1000 m/s
Vs0/Vp0	0.1	0.707
Vp0 change	0.01 m/s	3000 m/s
Vp1-Vp0	1 m/s	1500 m/s
Vs1/Vp1	0.1	0.707
Vp1 change	0.01 m/s	2000 m/s
Vp2-Vp1	10 m/s	2000 m/s
Vs1/Vp1	0.1	0.707

The results are summarized on figures B.18 where 5 different runs were plotted together. They are in a relatively good agreement with the original true profile for Vs and a bit less for Vp. The main contrast at 100 m deep is roughly retrieved. The value for the exponent of the 1st layer is found to be 0.4 (rather than 0.3). This is due to the distinct discretization of the Vp curve along the depth for the true and the inverted models. However, looking on figure B.18, the two profiles are very close to each other. For the 2nd layer the best exponent is around 0.15 like the true value.

Though its complexity, we observe on figure 4.5 that the recovery of the amplification spectrum for vertically incident SH wave is relatively good. The main features at low frequencies are correctly inverted. As expected, the amplitudes at high frequencies are not well matched.

As the model depicted on figure B.16 seems to be affected by several biases due to the rough sampling, we tested the inversion on the refined model shown on figure B.17. The parameters are the same as for the last inversion. The number of sub-layers is also 5. Five runs are gathered on figure B.19. As a first remark, the achieved misfit are 4 times lower than for the first model. The velocity contrast at 100 m deep is better matched. The uncertainty below 100 m is still high and compares very well with the rough case. The number of iteration in this former case is limited to 200 (300 in the first case).

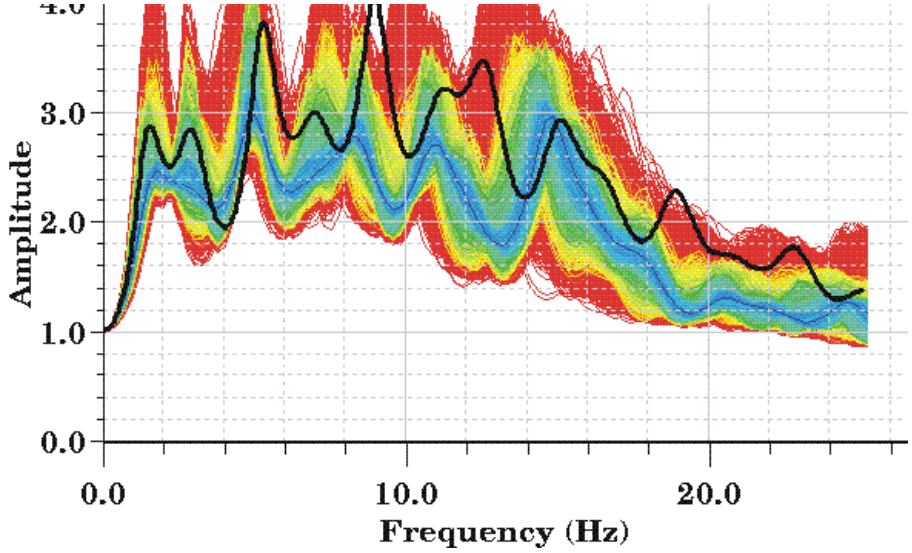

Figure 4.5: comparison of theoretical SH response for a vertical incidence for the true model (figure B.16) and the inverted models of figure B.18. Qs is 25 in the soft layers and Qp is 100. The color scale represent the misfit and ranges from 0.034 to 0.040.

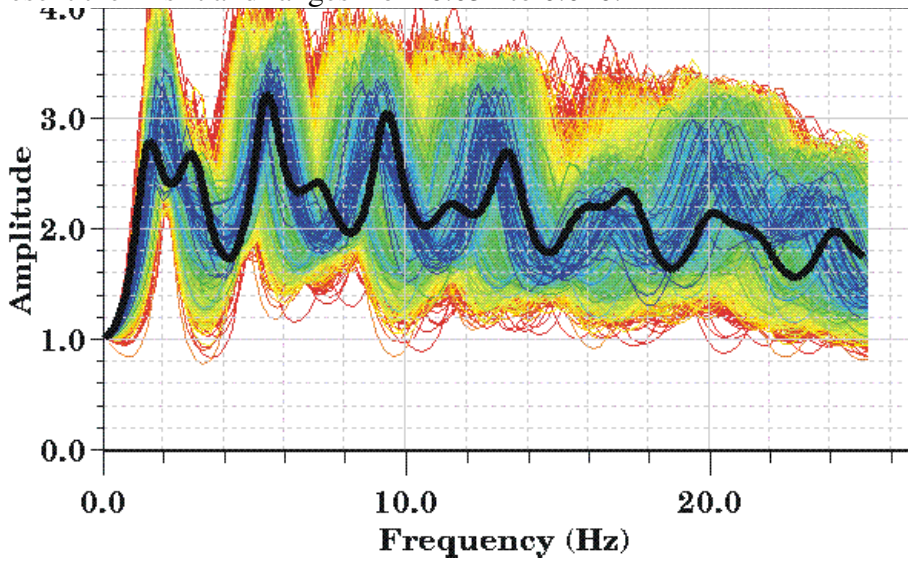

Figure 4.6: comparison of theoretical SH response for a vertical incidence for the true model (figure B.17) and the inverted models of figure B.19. Qs is 25 in the soft layers and Qp is 100. The color scale represent the misfit and ranges from 0.010 to 0.040.

4.4. Velocity inversions

It has been shown so far that the developed inversion code can invert whatever model with an increasing velocity profile with depth. In some cases, it would be interesting to detect velocity inversions inside the soil column. The forward computation has always a correct answer for the fundamental and higher modes when the velocity increases with depth. If this is not true any more, the modes can merge each other or can become very complex, especially at high frequencies. The basic strategy of root search and refinement does not work any more on the whole frequency range. However below a certain frequency, the modes are again well separated like in classical cases. This particular point is difficult to assess and depends upon the model parameters.

Using the same forward algorithm, we allow velocity inversions in the parametrized models and launch the dispersion computation without any pre-checking. If the situation is too complicated, the

automatic detection of mode jumping will force the step search to be smaller. It will be reduced up to 4 times (each time divided by 10). After the 4th reduction, the algorithm will end with an error code. The inversion driver will then remove the velocity inversion as if it were not allowed and will restart the dispersion computation. A penalty (function of the magnitude of the velocity inversion) will multiply the obtained misfit to force the inversion to generate other models. The number of models ending with an error is stored and displayed at the end of the log file.

The basic synthetic test proposed here is shown on figure B.20. The density of the first layer is higher than the rest of the soil structure. It is supposed to be some anthropic deposit. We propose to invert the dispersion curve of the fundamental mode with the list of parameters and their a-priori interval described in the following table. As the density of the first layer can be estimated at low cost, we decided to fix it to its true value to avoid some side effect (see section 3.5).

Parameter	***Minimum***	***Maximum***
Depth0	0.2 m	3 m
Vp0	400 m/s	2000 m/s
Vs0/Vp0	0.1	0.707
Vp1	200 m/s	100 m/s
Vs1/Vp1	0.1	0.707
Vp2-Vp1	10 m/s	2000 m/s
Vs1/Vp1	0.1	0.707

The results are summarized on figures B.21 where 3 different runs were plotted together. They are in a very good agreement with the original true profile for Vs. On the contrary, Vp of the first layer is not matched correctly.

Chapter 5: Conclusions and perspectives

The inversion program developed under the SESAME project is ready to use by other partners. Moreover, input file formats are entirely compatible with the noise recording processing tools.

Synthetic data show that valuable information can be retrieved from the dispersion curves only. The power of the direct search algorithm is also to get quick and reliable information on the resolution even if the non-uniqueness is high.

In some cases, other prospecting methods can add precious data to decrease the non uniqueness of the problem, for instance the depth of the basement is sufficient get an almost perfect Vs profile even with a truncated dispersion curve.

Though not presented in this report, the algorithm has already proved to be efficient on real data as well.

However, we are still developing and testing several improvements of the method.

1) For high frequencies, the higher mode contributions are often not negligible, as demonstrated by some of the field experiments carried out during the SESAME project. In accordance with other authors, synthetic tests proved that a misinterpretation leads to incorrect velocity profiles. For a single source it is well known that its depth is a significant parameter. One the other hand, for ambient vibration, there is still no clear understanding of the mode distribution or still no way of testing it for a particular site. This is why we proposed an automatic scheme to identify the most probable mode at a particular frequency. It has been successfully tested on synthetic data. We are waiting for noise synthetic signals to check its applicability to real cases.

2) As studied by Scherbaum et al. (2003), an interesting issue would be to include the fundamental resonance frequency as a supplementary constraint. It is measured by the H/V technique and it can be computed for every 1D models from the fundamental Rayleigh ellipticity. The fit of the amplitudes can not be realized at this stage without formulating some strong assumptions about the noise nature. Moreover, synthetic tests showed us that matching a wrong amplitude generally lead to inconsistent results. In the frame work of a robust method, it became obvious that only a frequency fit is possible. Due to the complexity of the ellipticity curve, the frequency fit is still under development. In the actual release, only an ellipticity amplitude fit is available. And we strongly recommend to use it when the H/V curve represents only the Rayleigh fundamental mode.

3) For the SPAC method, two inversions are classically performed: the first one to obtain the dispersion curve and a second one to get the velocity profile with depth. In the first one, various types of assumptions have to be made in order to reduce the number of free parameters, among them, the smoothness of the curve. In some case it leads to a non physical shape for the dispersion, altering the information. We extended the computation of dispersion curves to the spacial autocorrelation curves. On synthetic data, the inversion of the autocorrelation curves is as good as the inversion of the dispersion curves if the distances of station pairs are correctly distributed, covering the whole wavelength (or frequency) range.

Chapter 6: Installation and tutorial

In this chapter you will follow all steps needed to install the current release of the software (version 1.1.3). A short example will show you how to generate synthetic dispersion and ellipticity curves with **os_forward** and how to retrieve the original velocity profile using the Neighbourhood algorithm (**os_na** and **na_viewer**).

6.1. Installation

6.1.a. Linux

The package has been built using the multi-platform library Qt available under a GPL license for free at http://www.trolltech.com. Most of the Linux distributions include the Qt libraries by default (used also by well known KDE interface). But not all of them used the Qt version 3.1.1, this is the minimum requirement for any installation.

You can test your Qt version by typing in any terminal:

[user@machine userhome]$ qmake -v
Qmake version: 1.04a (Qt 3.1.1)
Qmake is free software from Trolltech AS.

If you get another message or if the version is older than 3.1.1, you must install the Qt libraries. Even if you have not the privileges of administrator on your computer, you can make it in your home directory without any risk of library confusion. If the last test was successful, you can jump directly to paragraph ii).

i) Installation of Qt libraries

Copy the file **qt-x11-free-3.1.2.tar.gz** from the CD or download the latest release from http://www.trolltech.com/download/qt/x11.html.

In a terminal type the following commands:

[user@machine userhome]$ tar xvfz qt-x11-free-3.1.2.tar.gz
[...]
[user@machine userhome]$ mv qt-x11-free-3.1.2 qt

You can inspect the file INSTALL for full informations on how to install the Qt libraries. However the basic commands are summarized here below.

Set some environment variables in the file .profile (.login, .bashrc, or .bash_profile,... ,depending on your favorite shell) in your home directory. Create the file if it is not already there.

QTDIR	*- the directory in which you're building Qt*
PATH	*- to locate the moc program and other Qt tools*
MANPATH	*- to access the Qt man pages*
LD_LIBRARY_PATH	*- for the shared Qt library*

This is done like this:

In .profile (if your shell is bash, ksh, zsh or sh), add the following lines:

QTDIR=/home/your_user_name/qt
PATH=$QTDIR/bin:$PATH
MANPATH=$QTDIR/doc/man:$MANPATH
LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
export QTDIR PATH MANPATH LD_LIBRARY_PATH

In .login (in case your shell is csh or tcsh), add the following lines:

setenv QTDIR /usr/local/qt
setenv PATH $QTDIR/bin:$PATH
setenv MANPATH $QTDIR/doc/man:$MANPATH
setenv LD_LIBRARY_PATH $QTDIR/lib:$LD_LIBRARY_PATH

After you have done this, you will need to login again, or re-source the profile before continuing, so that at least $QTDIR and $PATH are set. Without these, the installation will halt with an error message.

As a final step, go inside the qt directory and type the following commands (depending on the power of your computer these task may take some time, up to 1 hour on old laptops):

[user@machine qt]$ ./configure

Answer 'yes' to the license question

[user@machine qt]$ make

If your Linux distribution does not contain particular development standard libraries (e.g. libx11-devel.so), the make will quickly end with an error message. Install the needed library and restart by typing make again.

ii) Installation of sesame_array package

Copy the file **sesame_array_1.1.3-devel.tgz** from the CD and uncompress it. Type the following commands:

[user@machine userhome]$ tar xvfz sesame_array_1.1.3-devel.tgz
[...]
[user@machine userhome]$ mv sesame_array_1.1.3 sesame_array
[user@machine userhome]$ cd sesame_array
[user@machine sesame_array]$ ./install

This last command may take some time and should end without any warning nor error. As for Qt libraries you must modify the PATH and LD_LIBRARY_PATH to be able to launch the software. Add the following line in your .profile (replace by setenv for csh or tcsh, see above):

PATH=$HOME/sesame_array/bin:$PATH
LD_LIBRARY_PATH=$HOME/sesame_array/lib:$LD_LIBRARY_PATH
export PATH LD_LIBRARY_PATH

Re-source the profile or login again before continuing. In the bin/ directory you will find all the sesame_array commands now available from any terminal. For all of them, you can type its name followed by “-h” to obtain information about the command line options.

6.1.b. Windows

The software has been tested using 1-month trial license of the Qt libraries. Some small changes due to compiler specifications have been brought to source code. All interfaces remain the same as for the Linux version.

The installation is reduced to uncompressing the .zip file and copying the executable and their attached libraries to a proper location (the directory of your choice, e.g. D:\Program File\Sesame Array\ ). For your convenience you can add shortcuts in the start menu using the Microsoft Explorator tools.

This version will be available only if special needs in this sense are formulated. Unlike Linux, the Qt libraries are not free open sources. A Qt Enterprise Edition License must be purchased (http://www.trolltech.com).

6.1.c. Other platforms

As the source code is using only standard C or C++ libraries and the portable library Qt, it is virtually possible to compile it under Mac OS or any Unix system. No test has currently been carried out but any extension might be possible in the future without code rewriting (only small changes to comply with compiler specifications).

Recently, the Qt libraries were made available under a GPL license (for free) for Mac OS as for Linux or Unix systems.

6.2. Computing the synthetic curves

In this paragraph, you will compute the dispersion and the ellipticity curves for a simple 1D 3 layer model. They will be saved in **report** file. Report file are binary files used by all programs (inside the package) to store the huge amount of data generated by an inversion. At the beginning of those files, we save the information on the parameters chosen for the inversion and the curves or the object that the user wanted to fit (and a log comment on how the curves were obtained). In this way, it is always possible to look back at previous inversions without missing essential information. All report files can be viewed inside the program **na_viewer** (various reports at the same time with the same curves to fit and the same parameters) or **report_viewer** (various reports at the same time with, eventually, distinct parameters).

Create a new working directory called here after “wd” to save all file generated during this tutorial.

6.2.a. Construct a 1D model

Create a text file called “reference.model” with the following content:

```
3
20 0 500 0 250 0 2 0
30 0 1000 0 500 0 2 0
0 0 2500 0 1000 0 2.5 0
```

The first line contains the number of layers. The following lines contain respectively, the thickness (m), Vp (m/s), Vs (m/s), density. Each physical value is followed by its uncertainty (0 in this case). The thickness of the last layers is always ignored as it is the infinite half space.

6.2.b. Call os_forward to compute dispersion (and ellipticity)

Type the following command to get information about the os_foward options:

```
[mwathele@Canucks tutorial]$ os_forward -h
Usage:
os_forward [-L\R -p\f\l -k -d -t] -F modelfile -m nModes -s min -e max -n nsamp
```

-o outputReport [-r rmsReport] [-b batchFile]

-L Love modes are calculated
-R Rayleigh modes are calculated (default)
-p Regular sampling in period (default)
-f Regular sampling in frequency
-l Regular sampling in log(period)
-k In Monte Carlo analysis, keep Poisson's ratios constant
-d In Monte Carlo analysis, let underlying depths moving
-t Allow velocity inversion
when modifying a thickness
-F File containing the models (several models are accepted)
If any of model parameter has a stddev>0, a Montecarlo analysis is performed
-m Number of modes to calculate
-s Minimum value for period of frequency range
-e Maximum value for period of frequency range
-n Number of samples to use between min and max
-o Output report for the resulting curves
-r Report use to calculate the RMS (color of curves)
-b Batch file to execute on starting of this program
-h This help message

To start computation:

[mwathele@Canucks tutorial]$ os_forward -R -p -F reference.model -s 0.05 -e 0.5 -n 100 -m 2 -o reference.report

By the standard output you will get the calculated values, you can re-direct them to an ASCII file for other purposes. The window depicted on figure 6.1 will appear.

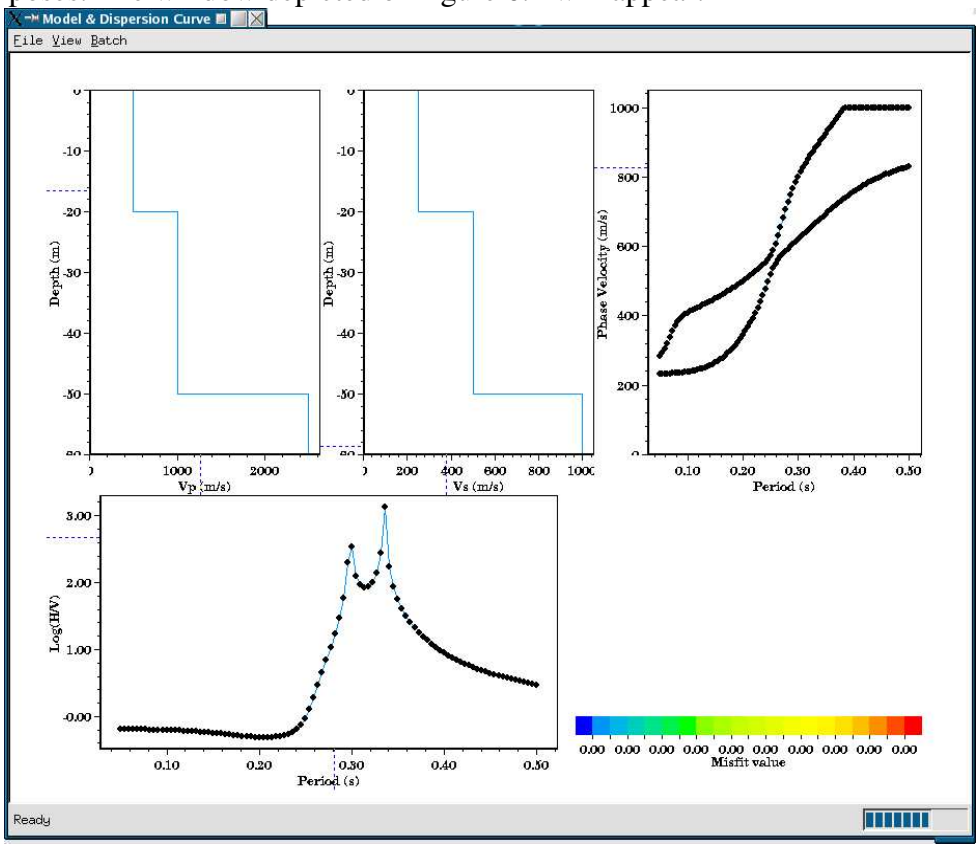

Figure 6.1: main window of os_forward presenting the results of the dispersion curves and ellipticity computation.

Use the right button of the mouse on the graphic contents or on axis to change their appearance and properties (double clicking will directly open the properties dialog box). The appearance of graphs can be saved in “layout” file (save and restore menus for each particular plot with a right click or for the whole graphic sheet with the top menu bar). To get the same display as above select “restore layout” from “file” menu. In the file open dialog box, go to sesame_array directory. In the directory “app_files” you will find basic layout files like “dispersion-ellipticity_hp1220.layout” used in figure XXX. You can manually edit these files, e.g. removing the min, max, tick and label fields to define standard canvas for printing (not destroying a particular frequency range when restoring).

To zoom in, choose “zoom in” in the context menu of graphic contents (right click) and draw the zoom rectangle by pushing down the left button and by dragging the mouse.

At this stage, the file reference.report will be processed by na_viewer to define the curve to fit during an inversion, as if it was the measured curve from real signals.

Close the os_forward window and type the following command:

[mwathele@Canucks tutorial]$ na_viewer

To get ready for the inversion and to launch an inversion we will explore the “Inversion” menu. Choose “Set dispersion to fit” from “Inversion” menu. The dialog box depicted on figure 6.2 will appear.

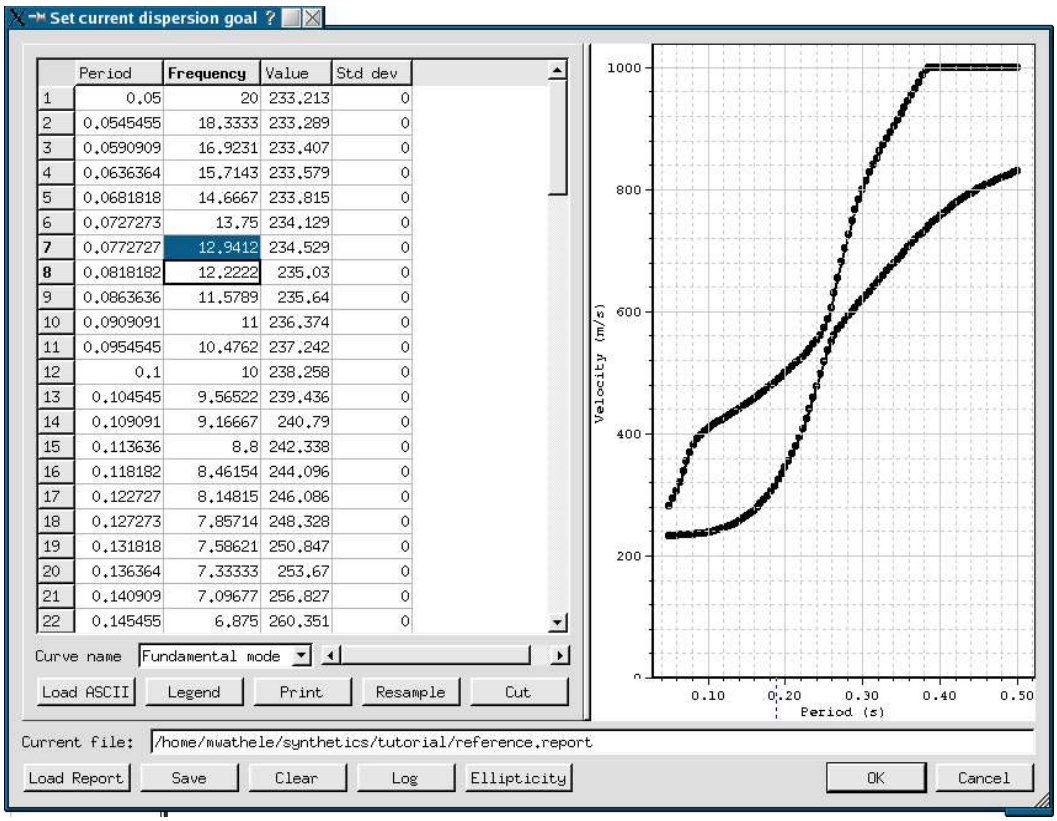

Figure 6.2: Dialog box that allows you to modify the dispersion curves to fit in na_viewer

When first opening the dialog box the plot should be blank. Click on “Load report” to look a the generated report. You can edit the values in the table for all modes, re-sample or cut the curves. If we want to invert only the fundamental mode we must remove the 1st higher mode from the report.

Switch to the table of the 1st higher mode by sliding the horizontal scroll bar to the right. In the standard curve name list choose "### remove ###". The 1st higher mode will disappear from the plot. You can set whatever name for the curves, but only curves named "Fundamental mode" and "Higher mode n°xxx" will be saved when clicking on "save" button, others will be ignored.

Re-sample the curve with constant period step from 0.1 s. to 0.4 s. with 30 samples (click on "Resample" button and set appropriate options). Click on "Save" button to create the "curve_to_fit.report" file containing only the fundamental mode from 2.5 Hz to 10 Hz. We limit the frequency range to 0.4 s. to take into account the high pass filtering effect of the soil structure; it is rare to obtain a good definition of the dispersion above the peak frequency.

You can click on the "Log" button to show all the comments automatically generated so far. You can add your particular information but do not forget to save the report again to keep it. Keep in mind that a new average and merge will be performed; if you do not want to modify the saved curve but only the log comment, keep only the red curve on the plot and remove all others. To end this part click on "OK". When opening the dialog box again, the selected report will be automatically opened. This report will be used by the inversion processes as objective curves (reference curves to compute the misfit).

IMPORTANT: in the present version, the joint inversion of ellipticity is based on the comparison between the calculated amplitudes and the amplitude of the given curve. For real H/V curves, in most situations, this scheme leads to incorrect velocity profiles. This option is still under development and will be adapted in future releases.

6.3. Starting the inversion

Inside the same menu ("Inversion"), choose "Run". The dialog box of figure 6.3 will appear:

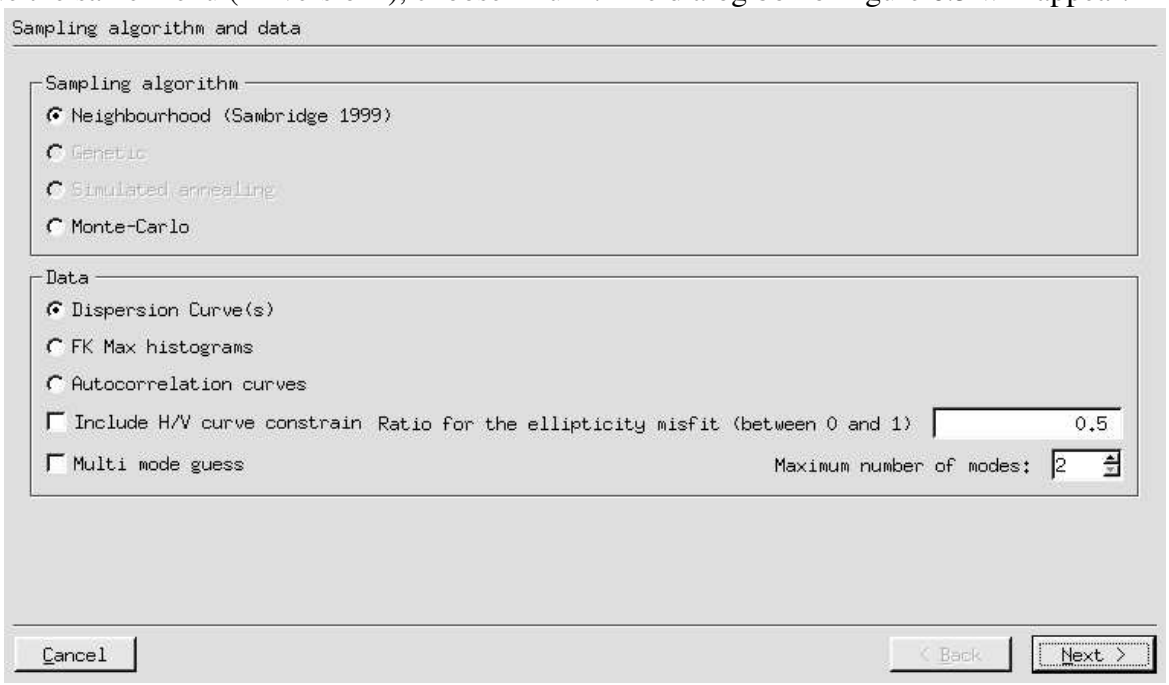

Figure 6.3: in na_viewer, first page of wizard used to construct the input files of the inversion processes (na.in and param.in).

In a few steps you will construct the necessary file for the inversion program os_na: "**na.in**" for the Neighbourhood options and "**param.in**" for dispersion and ellipticity computations.

i) Page "Sampling algorithm and data"

Make sure that Neighbourhood option is selected. Select the dispersion curve(s) option and make sure that all others are not checked.

ii) Page "Neighbourhood Algorithm properties"

Fix the maximum number of iteration to 200, the sample size for first iteration and for others to 50, and the number of cell to re-sample to 25. The number of model that will be generated is therefor 50+200*50=10050. The number of cells to re-sample set the behaviour of the search algorithm: more like an optimizer or an explorer. Its value cannot exceed either the initial sample size or the number of models generated at each iteration.

iii) Page "Model parameters"

You can set the number of layers of the generated models (not necessary the same as the reference model). For each layer you have to define which parameter will be held constant or not. If it is a free parameter you must define its a-priori range (uniform probability over this range). The second table summarize the parameters for all layers as shown on figure 6.4.

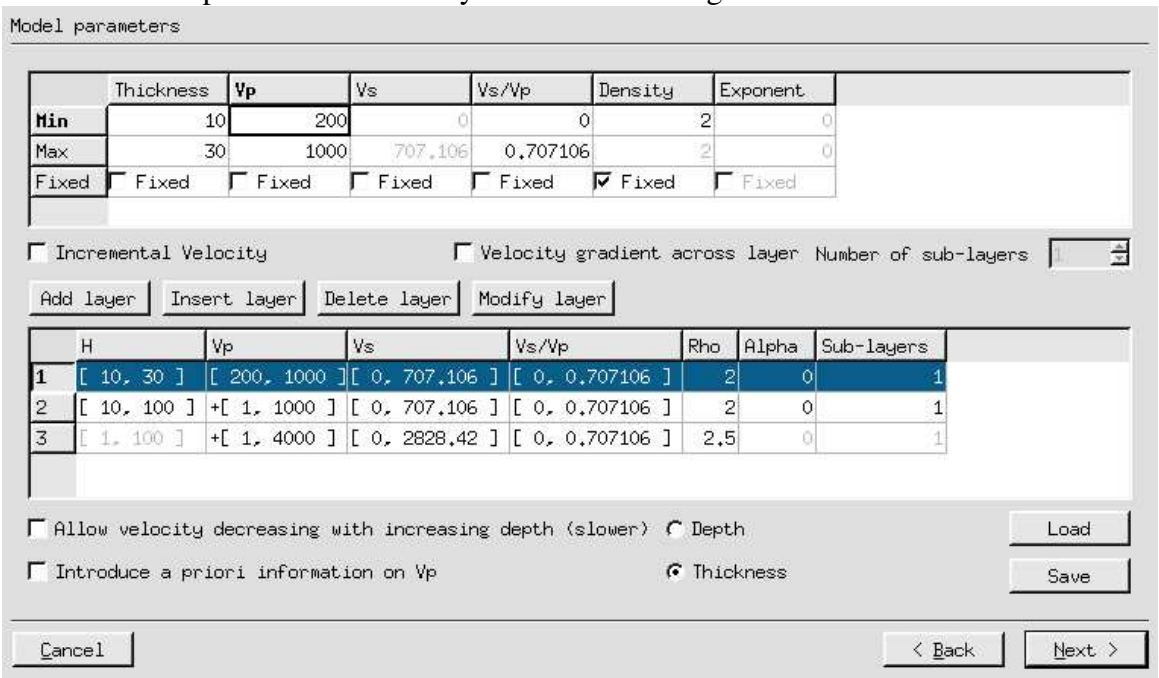

Figure 6.4: interactive definition of the model parameters.

If you want to set a power law variation of the velocity across a layer (not possible for the last one, as it is an infinite half space), check the option "Velocity gradient", and set the number of sub-layers to represent the variation (e.g. between 5 and 10). The density is usually not constrained by the dispersion and must be kept fixed. The "Load" button allows you to open previously ran inversions, to get the same parameters ("param.in" file), or to open previously saved parameters (*.param) with the "Save" button.

If you select the “Depth” instead of the “Thickness”, it is possible to constraint the absolute layer's depth. It can be useful if the top of the bed-rock has been clearly identified by boreholes or other methods.

If you are running this tutorial for the first time, we advice you to enter the same parameters as displayed on figure 6.4.

iv) Page “Processes”

Set the process name for example as “my_first_inversion_with_3_layers”. Click 3 times on button “Add”. Set the maximum cost value to 0.1, only models having a misfit value below 0.1 will be stored in the report file. You are going to start three independent inversion processes initiated by the random seed specified in the table. You can modify it, but it is generally randomly chosen when clicking on “Add”. According to the non uniqueness of the problem, it is helpful to start more than one inversion to check the stability of the final result. For unresolved parameters, the whole a-priori range will hold equally “good” solutions.

v) Page “Host and working directory”

Set the working directory to “/home/username/...../wd” using the button “Browse”. Check that the options “Execute all processes now” and “Periodically add new models” are correctly checked. An interval of 10 seconds is generally good.

End this stage by clicking on “Finish”. In the chosen directory (“/home/username/...../wd”), you will find three sub-directories with the name of the processes preceded by “wd_”, and three reports whose size should normally increase as the inversion is running. Inside the sub-directories, you have the input files **na.in** and **param.in**, a log file (reporting eventually error during inversion) and “start_inversion”. If you want to restart the inversion without making all steps of the wizard, you can go inside the sub-directory and launch the script “start_inversion” from a terminal. The corresponding report file will be erased and its size will begin to increase again. You can visualize the generated models while inverting as if it was launched from na_viewer by opening the report (menu “File”, and periodically select the “Add models”).

6.4. Viewing the results while inverting

Once you are done with the previous step, the inversion processes are launched and their report will be progressively opened in the na_viewer (when they contain at least one model).

i) Nothing happen

If no model appears, you will probably have to seek for some error in the input parameters. Quit na_viewer and kill eventually running processes:

```
[mwathele@Canucks tutorial]$ ps -a | grep os_na
10539 pts/1    00:00:08 os_na
10540 pts/1    00:00:05 os_na
10541 pts/1    00:00:03 os_na
[mwathele@Canucks tutorial]$ kill -9 10539
[mwathele@Canucks tutorial]$ kill -9 10540
[mwathele@Canucks tutorial]$ kill -9 10541
```

You can, in a first approach increase the cost value threshold to admit more models in the report. Edit the param.in files of each process (using vi, kwrite, xemacs,...) and change the "max_cost_value" to a higher value. Re-start the inversions as described in the preceding paragraph.

Look in the log file to see if the process stopped in a unusual way (not ending with the summary from the Neighbourhood Algorithm). If you really do not understand where is the problem, there is probably a bug to correct in the program itself or in the documentation. Please do not hesitate to send a bug report (just an email) to the developer (marc.wathelet@ujf-grenoble.fr).

ii) Plots are created in the main window

The message bar (at the bottom) tells you which inversion process is starting and how many models are stored in the reports corresponding to running processes. In the "Inversion" menu, select "Status" to get information about real advance of each individual process (minimum misfit reached, decrease history, NA parameters and the number of model generated). The number of generated models is always superior or equal to number of stored models.

To plot the best model at front and others at the back, you must sort the models (menu "Sort" in "View"). You can change the X, Y and color scales to automatically fit all points by selecting menu "Set Limits" in "View". At this point, the parameter space should look like figure 6.5, it is never exactly the same as the random seeds are different. You can see various projections on 2D plots of the complex parameter space (dimension 8). For badly resolved parameters there is not a clear minimum.

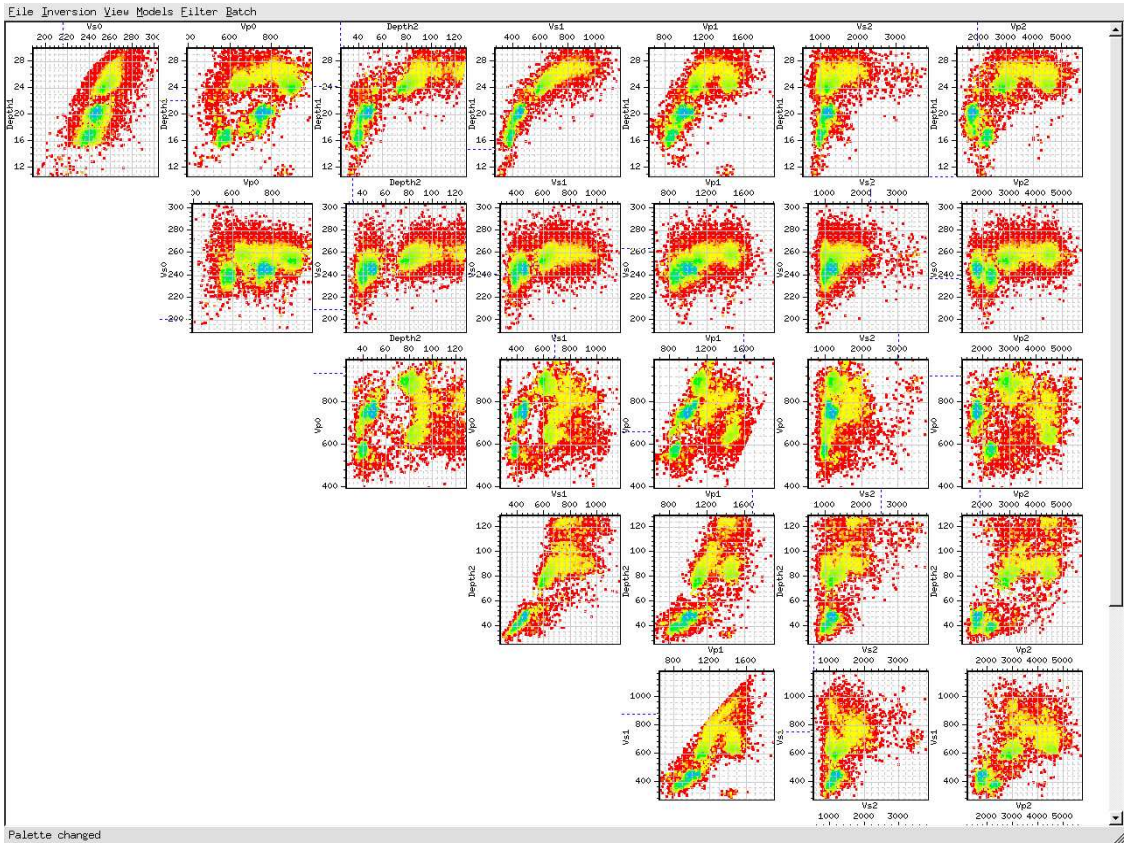

Figure 6.5: parameter space, 2D marginals as presented by na_viewer.

You can change manually the color palette by selecting “Cost Palette” in menu “View”. In the following figure the palette range from 0 to 0.04 with linear intervals and 4 digits for the precision.

It is possible to view the generated models in another way. Select “By threshold” in menu “Models” to view the Vp and Vs profiles with depth, and their dispersion curves. Various ways exist to select the models to plots: models that fit between axis limits, picking them in a list, all models from a report, models with an original misfit less than a threshold and models with a recalculated misfit less than a threshold (computation can be performed on distinct frequency range).

The black line indicates the reference model. You can add it by selecting “Add reference model” in menu “View”.

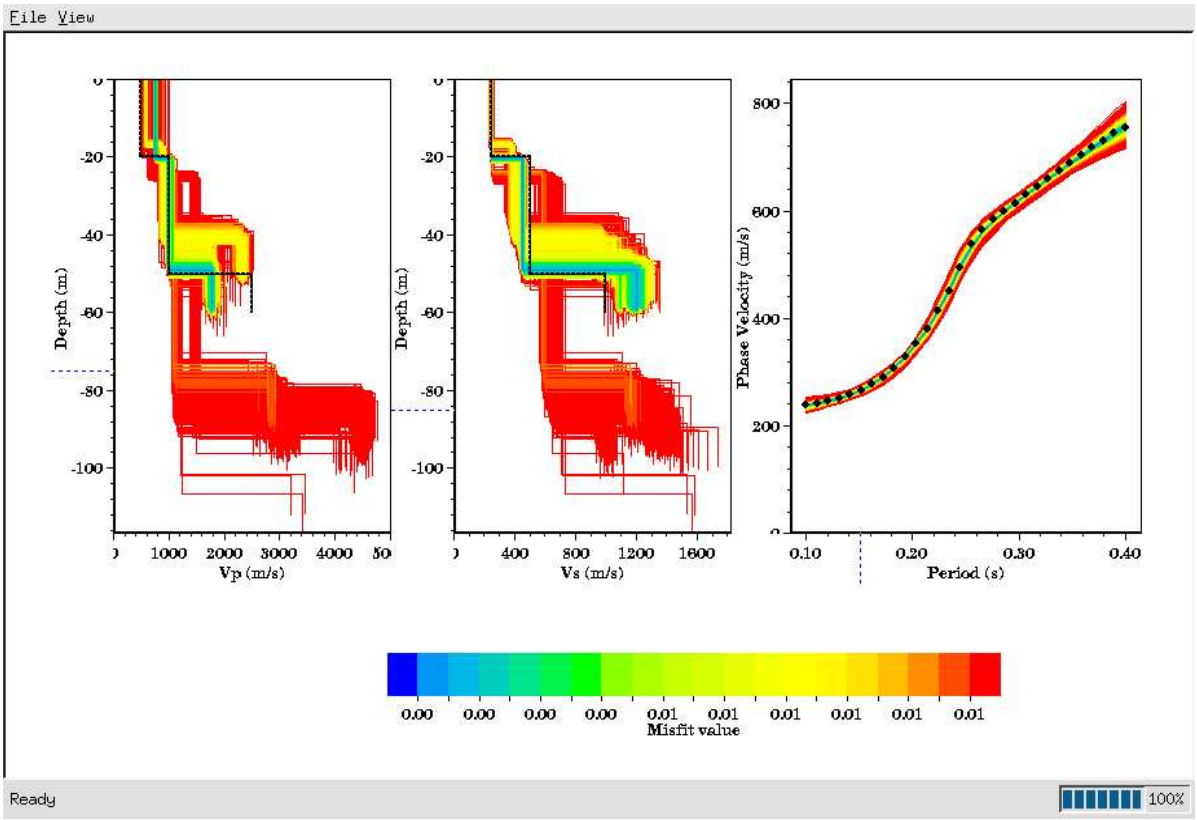

Figure 6.6: a view of the parameter space using the Vs and Vp profiles. On the right side is the dispersion curves for all models presented on the 2 other plots on the left and in the middle.

References

Aki K. (1957) – Space and time spectra of stationary stochastic waves, with special reference to microtremors. Bull. Earthq. Res. Inst., 35, 415-456

Aki K. and Richards P. G. (1980) – Quantitative seismology. Theory and methods. Vol I, W. H. Freeman and Company, ISBN 0-935702-96-2

Bettig B., Bard P.-Y., Scherbaum F., Riepl J., Cotton F., Cornou C. and Hatzfeld D. - Analysis of dense array noise measurements using the modified spatial auto-correlation method (SPAC). Application to the Grenoble area. Submitted August 2001

Capon J. (1969) – High-resolution frequency-wavenumber spectrum analysis. Prc. IEEE 57, 1408-1418

Dunkin J. W. (1965) – Computation of modal solutions in layered, elastic media at high frequencies. Bull. Seism. Soc. Am., 55, 335-358

Fäh D., Kind F. and Giardini D. (2001) – A theoretical investigation of average H/V ratios. Geophys. J. Int., 145, 535-549

Fäh D., Kind F. and Giardini D. (in press) – Inversion of local S-wave velocity structures from average H/V ratios, and their use for the estimation of site-effects.

Gilbert, F. and G. Backus (1966) – Propagator matrices in elastic wave and vibration problems, Geophysics, 31, 326-332

Haskell, N. A. (1953) – The dispersion of surface waves from point sources in a multi-layered medium, Bull. Seism. Soc. Am., 54, 377-393

Jianghai X., Richard D. M., Choon B. P. and Gang T. (2003) – Inversion of high frequency surface waves with fundamental and higher modes. Journal of Applied Geophysics, 52, 45-57

Jongmans D. (1990) – Les phénomènes d'amplification d'ondes sismiques dus à des structures géologiques. Annales de la Société Géologique de Belgique, 112, 369-379

Knopoff L. (1964) – A matrix method for elastic wave problems. Bull. Sesim. Soc. Am., 54, 431-438

Lacoss R. T., Kelly E. J. and Toksöz M. N. (1969) – Estimation of seismic noise structure using arrays. Geophysics, 34, 21-38

Lomax, A. J. and Snieder, R. (1994) – Finding sets of acceptable solutions with a genetic algorithm with application to surface wave group dispersion in Europe. Geophysical Research Letter, 21, 2617-2620

Herrmann, R.-B. (1987) – Computer programs in seismology vol. 4, St Louis University

Numerical Recipes

Ohrnberger M. (2001) – Continious automatic classification of seismic signals of volcanic origin at Mt Merapi, Java, Indonesia. Dissertation University of Potsdam, pp168

Sambridge, M (1999a) – Geophysical inversion with a neighbourhood algorithm I. Searching a parameter space. J. Geophys. Res., 103, 4839-4878

Sambridge, M (1999b) – Geophysical inversion with a neighbourhood algorithm I. Searching a parameter space. J. Geophys. Res., 103, 4839-4878

Scherbaum F., Hinzen K.-G. and Ohrnberger M. (2003) – Determination of shallow shear wave velocity profiles in the Cologne/Germany area using ambient vibrations. Geophys. J. Int., 152, pp 597-612

Sen, M. K. and Stoffa, P. L. (1991) – Nonlinear one-dimensional seismic waveform inversion using simulated annealing. Geophysics, 56, 1624-1638

Shuang X. Z. and Lung S. C. (2003) – Possible effects of misidentifed mode number on Rayleigh wave inversion. Journal of Applied Geophysics, 53, 17-29

Tokimatsu, K. (1997) – Geotechnical site characterization using surface waves. Earthquake Geotechnical Engineering, Ishihara (ed.), Balkema, Rotterdam, ISBN 90 5410 578 X

Thomson, W. T. (1950) – Transmission of elastic waves through a stratified solid medium, J. Appl. Phys., 21, 89-93

Acknowledgements

This project (Project No. EVG1-CT-2000-00026 SESAME) is supported by the European Commission Research General Directorate. The software is currently used by other SESAME partners in Potsdam, Zurich and Grenoble. We would like to thank them for reporting bugs allowing a constant improvement of the reliability and the flexibility of the program. We also appreciate the constructive discussions we had with P.-Y. Bard, S. Bonnefoy, F. Cotton, C. Cornou, D. Fäh, M. Orhnberger and F. Scherbaum.

List of Appendices :

Figures A.1 to A.24	Parameter sensitivity
Figures B.1 to B.21	Synthetic data inversions