

Enhanced Method for the Solution of the Potential Difference Integral

Rafael E. Banchs

INTRODUCTION

This report describes an enhanced method for the solution of the potential difference integral in the Time Harmonic Field Electric Logging problem. Under certain circumstances, the original methodology for solving the potential difference integral fails to give an accurate representation of the ΔR 's located relatively far from the current element. Although these inaccuracies do not alter significantly the final tool measurements, the solution of this problem would make the algorithm more robust and reliable. The enhanced method presented here, instead of a replacement for the original one, actually constitutes a modification that improves the overall performance of the algorithm.

THE ORIGINAL METHOD AND ITS LIMITATIONS

First of all, let us present a brief review of the original technique. A more detailed description of it is provided in [1]. The original technique is based in the use of two complementary exponential windows for decomposing the potential difference integral into two independently solvable integrals, denoted as $I_1(z)$ and $I_2(z)$.

Each of these two new integrals are approximated by a discrete convolution as follows:

$$I_1(z) = g_1(z) * f_1(z) \tag{1}$$

where $f_1(z)$ is a known finite impulse response filter and $g_1(z)$ is an interpolated sequence of the cosine transform samples computed by the fast integration algorithm [2].

$$I_2(z) = g_2(z) * f_2(z) + \xi(z) \quad (2)$$

where $f_2(z)$ is another known finite impulse response filter, $g_2(z)$ is an interpolated sequence of the sine transform samples computed by the fast integration algorithm [2] and $\xi(z)$ is a bias term that is computed analytically.

The main limitation of the original methodology for solving the potential difference integral is its failure in giving an accurate representation of those ΔR 's located far from the current element when the experimental scenario leads to strong attenuation of the electromagnetic fields. That usually happens when the mud conductivity is much smaller than the surrounding zones or when the tool is operating at relatively high frequencies.

Figures 1 and 2 show the absolute value of the ΔR 's for both of the scenarios described above, high conductivity contrast (mud conductivity \ll true conductivity) and high frequency of operation respectively.

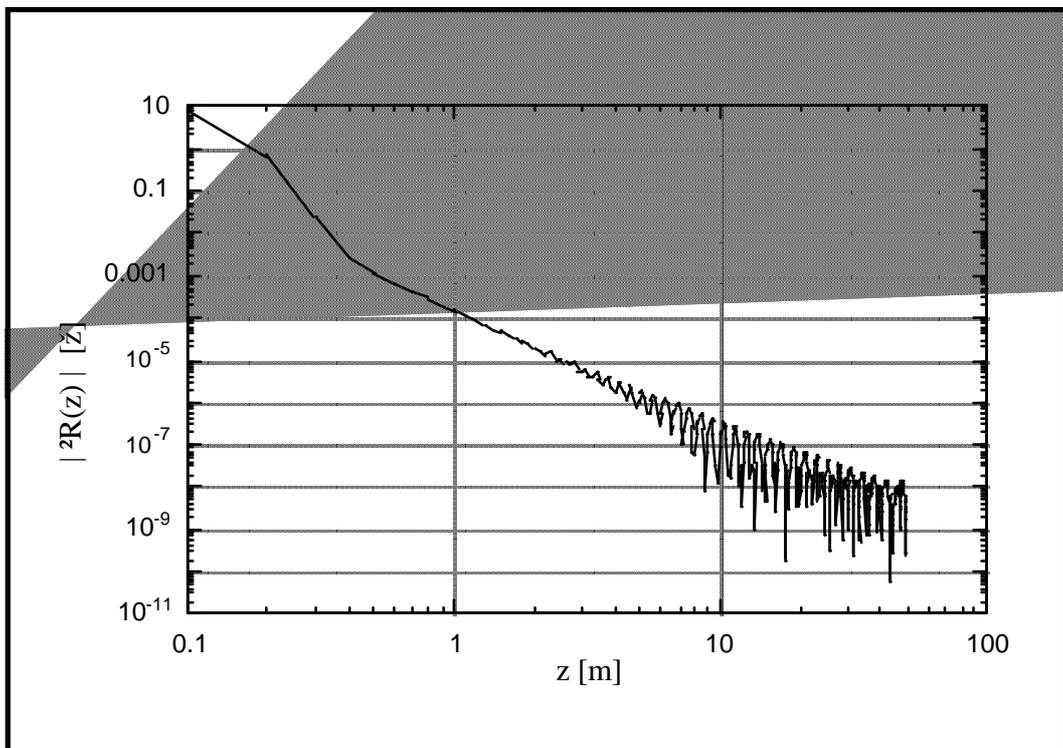


Figure 1: ΔR 's in a high conductivity contrast scenario.

The high conductivity contrast scenario considered for figure 1 was a two-zone formation with mud conductivity of 0.1 S/m, true conductivity of 10 S/m and borehole radius of 0.1 m, with a tool's operation frequency of 0 Hz. On the other hand, the high frequency scenario considered for figure 2 was an homogeneous formation of conductivity 1 S/m, and an operation frequency of 1.0 Mrad/s. The same symmetric Lls tool was used for both of the simulations.

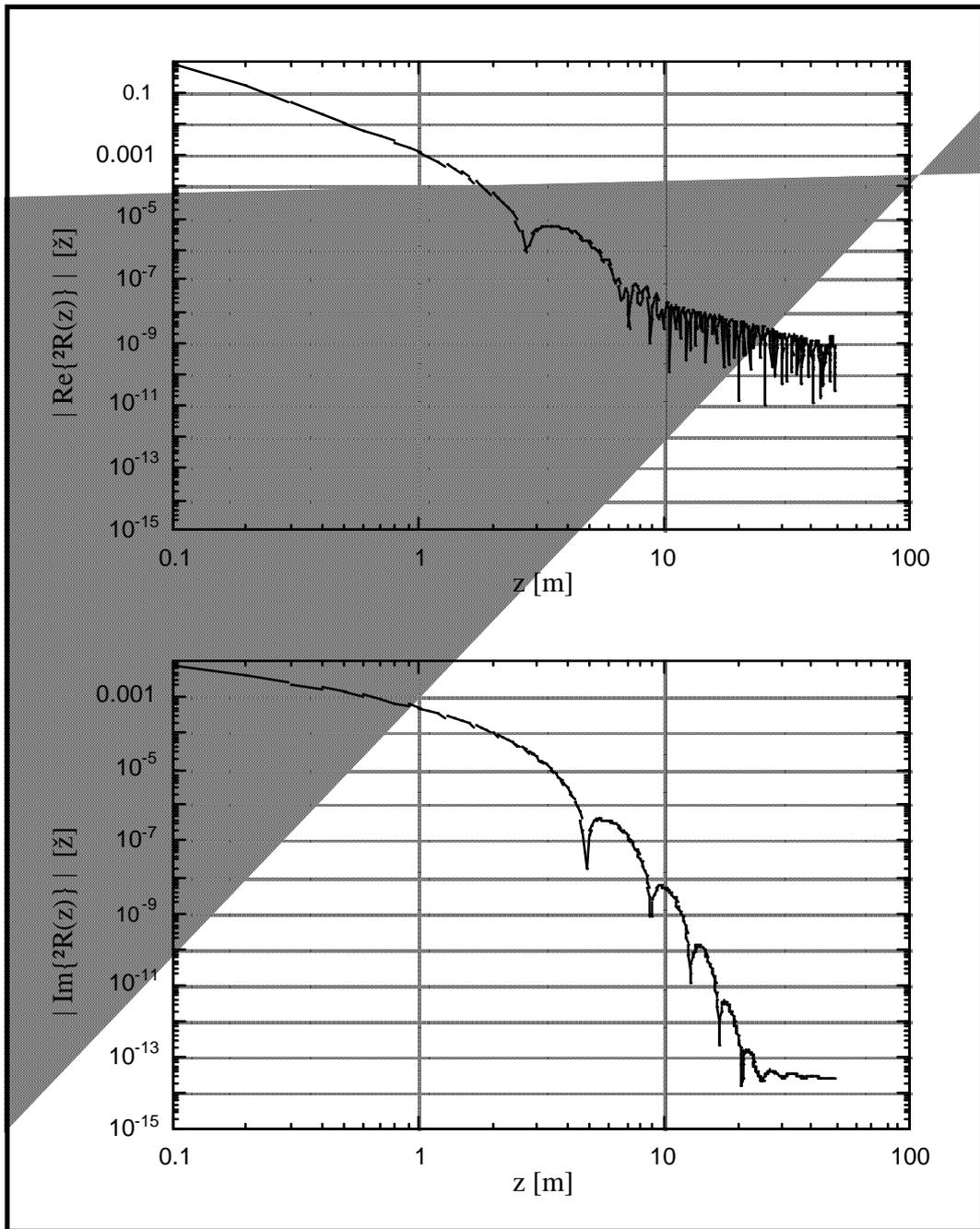


Figure 2: ΔR 's in a high operation frequency scenario.

As it can be seen from figures 1 and 2, only the real part of the ΔR 's is affected by the problem. After a careful and detailed analysis of the intermediate computations along the solution

procedure over a wide variety of experimental scenarios, the problem was detected to be generated by the way $I_2(z)$ is computed. Figure 3 illustrates how it happens.

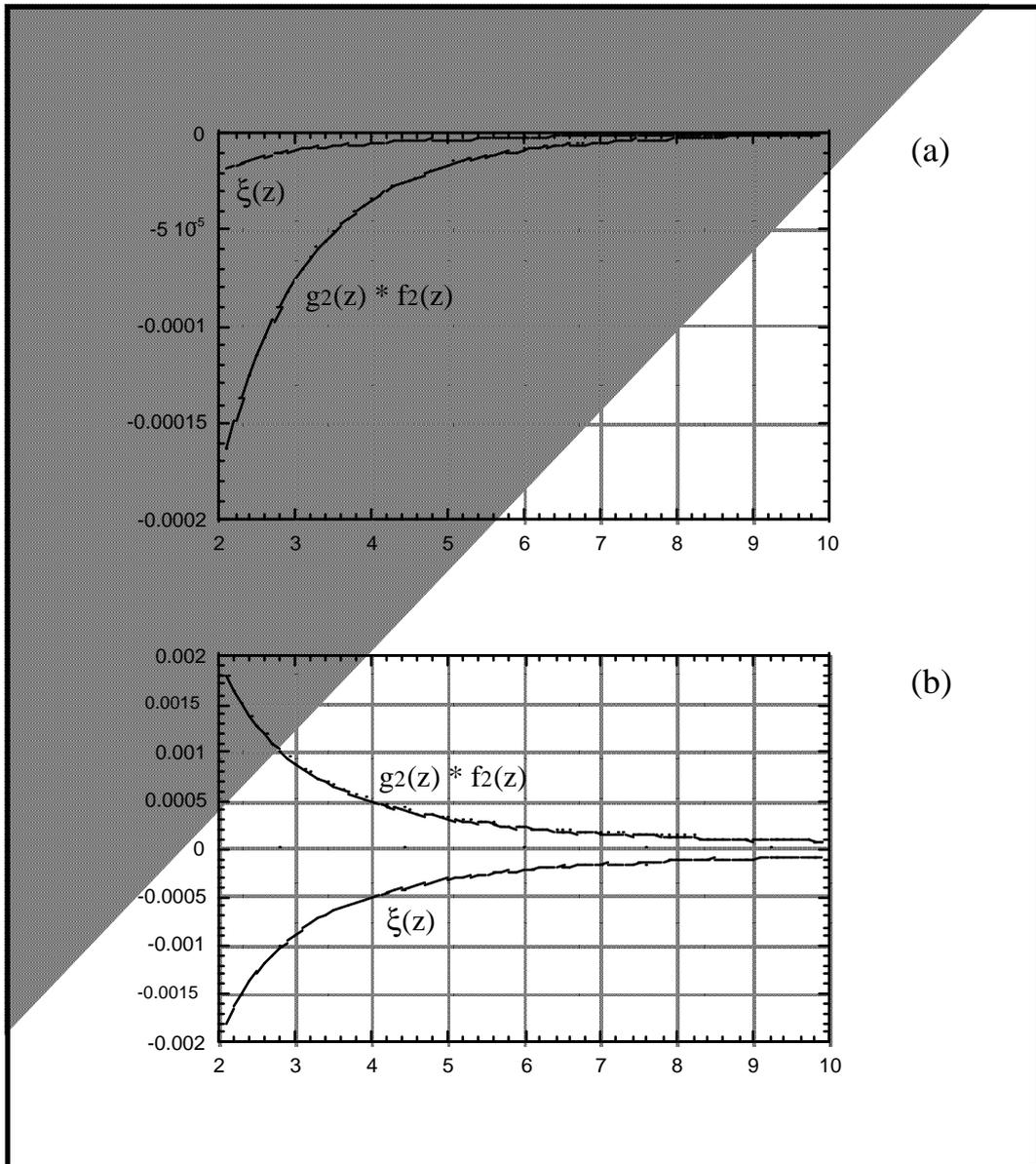


Figure 3: Bias term and convolution term in high conductivity contrast scenarios with:
 (a) mud conductivity \gg true one, and (b) true conductivity \gg mud one.

Notice from figure 3 that the problem is related to the existence of the bias term $\xi(z)$. In fact, because the convolution in (2) numerically approximates the value of an integral, a small numerical error (noise) is always embedded in the result. When strong attenuation must occur, the method relies on cancellation, between the bias term and the convolution term, in order to produce small results. Then, the numerical error becomes more significant in proportion to the final result.

For the case of high operation frequency scenarios the situation happens to be very similar to the one illustrated in figure 3.b.

Because of the reasons exposed above, it is now clear that any alternative procedure attempting to overcome these limitations must compute $I_2(z)$ in the same way the integral $I_1(z)$ is computed, by approximating the integral value without the necessity of any bias term.

A BRIEF ANALYSIS ON THE COMPUTATION OF $I_2(z)$

The potential difference integral, which is presented in [3], is given by:

$$\Delta R(z) = \frac{2}{r_0 h \pi^2} \int_{-\infty}^{\infty} \frac{\beta_1}{\sigma_1} \frac{K_0(\beta_1 r_0) + \Gamma_1 I_0(\beta_1 r_0)}{K'_0(\beta_1 r_0) + \Gamma_1 I'_0(\beta_1 r_0)} \frac{\text{Sin}^3(\lambda h/2)}{\lambda^3} e^{-j\lambda z} d\lambda \quad (3)$$

$$\text{where } \beta_1 = \beta_1(\lambda) = \sqrt{\lambda^2 + j \omega \mu \sigma_1}, \quad (4)$$

r_0 is the radius of the logging tool, h is the segment length, ω is the angular frequency of operation, μ is the magnetic permeability, σ_1 is the electric conductivity of zone 1, Γ_1 is the reflection coefficient of zone 1 (which is computed by the recursive procedure described in [3]), and I_0 and K_0 are the zero order Modified Bessel functions of first and second kind.

And the integral $I_2(z)$ is initially defined as (see [1]):

$$I_2(z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\frac{\beta_1}{\lambda \sigma_1} \frac{K_0(\beta_1 r_0) + \Gamma_1 I_0(\beta_1 r_0)}{K_0'(\beta_1 r_0) + \Gamma_1 I_0'(\beta_1 r_0)} w_2(\lambda) \right] \left[\frac{\text{Sin}^3(\lambda h/2)}{\lambda^2} \right] e^{-j\lambda z} d\lambda \quad (5)$$

where $w_2(\lambda)$ is the following exponential window:

$$w_2(\lambda) = 1 - \text{Exp}\left(\frac{-\lambda^2}{\omega \mu \sigma_1}\right) \quad (6)$$

Notice that according to the product property of the fourier transform, (5) can be computed in the z domain by convolving the inverse fourier transforms of the factors in brackets which are denoted $G_2(\lambda)$ and $F_2(\lambda)$ respectively. Let us denote their inverse fourier transforms as $g_2(z)$ and $f_2(z)$. While $f_2(z)$ can be computed analytically, $g_2(z)$ must be computed by using numerical integration. However, as it is described in [1], there is a problem involved in the computation of $g_2(z)$. That is the fact that $G_2(\lambda)$, instead of going to zero, tends to a constant value when λ approaches infinity, making the numerical computation of $I_2(z)$ very inaccurate.

Because of the reason exposed above, $G_2(\lambda)$ had to be redefined by adding a step function such that the new $G_2(\lambda)$ would tend to zero as λ approaches infinity. Although that modification allowed a more accurate computation of $g_2(z)$, it also causes the necessity of a bias term whose roll is to compensate the effect of adding the step function. In fact, the bias term is computed as the convolution between $f_2(z)$ and minus the step function [1].

ALTERNATIVE PROCEDURE FOR COMPUTING $I_2(z)$

After considering the facts discussed in the previous section, it seems that in order to achieve a successful alternative procedure, (5) has to be modified in some way (different than the addition of a step function of course) such that the factor inside the first brackets goes to zero when λ approaches infinity. The most obvious and natural way of doing that is by rewriting it as follows:

$$I_2(z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\frac{\beta_1}{\lambda^2 \sigma_1} \frac{K_0(\beta_1 r_0) + \Gamma_1 I_0(\beta_1 r_0)}{K_0'(\beta_1 r_0) + \Gamma_1 I_0'(\beta_1 r_0)} w_2(\lambda) \right] \left[\frac{\text{Sin}^3(\lambda h/2)}{\lambda} \right] e^{-j\lambda z} d\lambda \quad (7)$$

Again, let us denote as $\tilde{G}_2(\lambda)$ and $\tilde{F}_2(\lambda)$ the new factors inside brackets, and as $\tilde{g}_2(z)$ and $\tilde{f}_2(z)$ their inverse fourier transforms respectively. Then, by using the product property of the fourier transform, the integral in (7) can be written in the z domain as the following convolution:

$$I_2(z) = \tilde{g}_2(z) * \tilde{f}_2(z) \quad (8)$$

where the function $\tilde{f}_2(z)$ can be computed analytically and is given by:

$$\begin{aligned} \tilde{f}_2(z) = & -1/16 & \text{for } & z = -3h/2 \\ & -1/8 & \text{for } & -3h/2 < z < -h/2 \\ & +1/16 & \text{for } & z = -h/2 \\ & +1/4 & \text{for } & -h/2 < z < +h/2 \\ & +1/16 & \text{for } & z = +h/2 \\ & -1/8 & \text{for } & +h/2 < z < +3h/2 \\ & -1/16 & \text{for } & z = +3h/2 \\ & 0 & \text{otherwise} & \end{aligned} \quad (9)$$

On the other hand, $\tilde{g}_2(z)$ has to be computed numerically. Notice from (7), that because $\tilde{G}_2(\lambda)$ exhibits even symmetry with respect to $\lambda = 0$, the computation of $\tilde{g}_2(z)$ can be performed by an inverse cosine transform as follows:

$$\tilde{g}_2(z) = \frac{1}{\pi} \int_0^{\infty} \tilde{G}_2(\lambda) \text{Cos}(\lambda z) d\lambda \quad (10)$$

where the addition of a step function is not required anymore.

In this way, $I_2(z)$ can be directly computed as described in (8) without the necessity of any kind of bias function. However, this new procedure, which overcomes the computational limitations of the original methodology, presents some important limitations and cannot directly substitute

the first procedure. A detailed discussion about those limitations is presented in the following section.

LIMITATIONS OF THE NEW PROCEDURE

Now, let us analyze in more detail the implications of the modification introduced in (7). There are basically two important limitation associated with the methodology developed in the previous section. Although one of these limitations can be solved and does not represent an actual problem; the other one will finally determine that the ultimate algorithm must be a combination of the original and the new one.

The first and resolvable limitation is related to the fact that the function $\tilde{f}_2(z)$ in (8) is actually a differentiator of second order. It can be easily verified from (7) that its fourier transform tends to λ^2 when λ approaches 0. Nevertheless, the real cause of the problem is the use of a natural cubic spline for interpolating the output of the fast integration technique used to compute (10).

As it can be seen in [2], the fast integration technique provides output values with logarithmically spaced abscissas. Interpolation is then performed to estimate the uniformly spaced samples required for the numerical evaluation of (8). As a natural cubic spline is used for interpolating, the subsequent application of a differentiator of second order leads to an approximation of $I_2(z)$ with discontinuous first order derivative, and consequently, to an inaccurate representation of the ΔR 's. Figure 4 illustrates this problem for the same experimental setting used for figure 2. Notice however that a substantial improvement over the original procedure have been achieved (compare figure 4 to figure 2).

The solution of this first limitation, which mainly affects the high operation frequency scenarios (where the values of the ΔR 's exhibit oscillatory responses), can be easily achieved by replacing the natural cubic spline interpolating function by a spline of a higher degree. The disadvantages

of using higher degree splines as interpolators are the increment of the computational complexity of the interpolation subroutines and the tendency to produce strong oscillations. However, in this particular case, there is more to win than to loose by interpolating with a spline of a degree higher than three. In this way, a natural quartic spline algorithm was specifically designed to be included in the new procedure for computing $I_2(z)$. A detailed description of the algorithm is presented in [4]. Figure 5 shows how the results are again improved when the natural quartic spline is used (compare figure 5 to figure 4).

The second and worse limitation appears in the computation of $\tilde{g}_2(z)$ for values of z approaching zero. In fact, when $z = 0$, the integral transform in (10) diverges. This is because the function being transformed decays as $1/\lambda$ as λ approaches infinity; so convergence of (10) is totally due to the oscillations of the transformation kernel (the cosine function); the higher is the value of z the faster is the convergence of the integral.

As it can be seen now, this problem not only makes impossible the numerical computation of $\tilde{g}_2(z)$ at $z = 0$; it also produces relatively large errors when numerically evaluating the integral at small values of z . Nevertheless, for large values of z (where the integral values are mainly determined by the behavior of the functions around $\lambda = 0$), (10) can be computed with a great degree of accuracy.

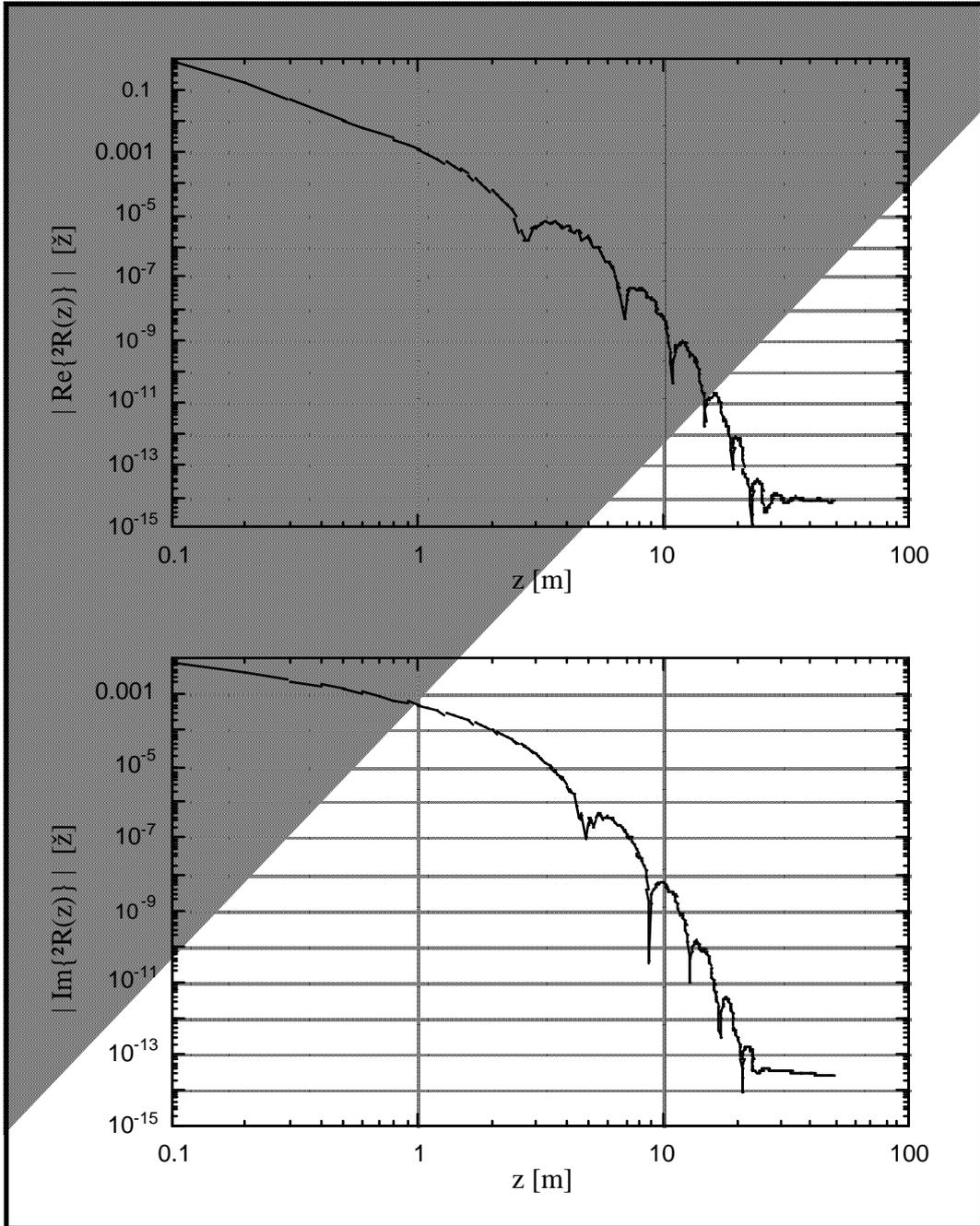


Figure 4: Performance of the new method in a high operation frequency scenario.

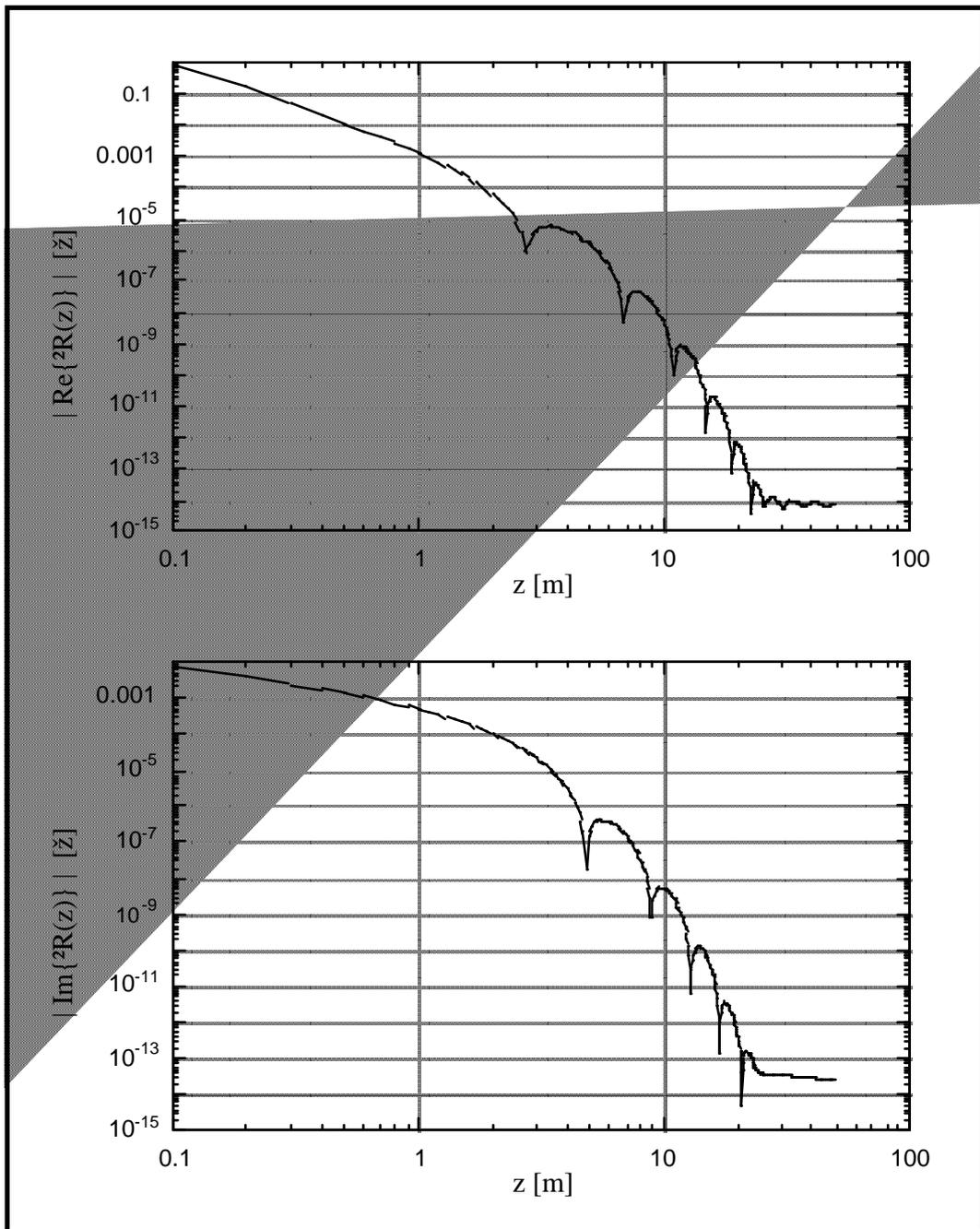


Figure 5: New method with natural quartic spline in a high operation frequency scenario.

THE FINAL ENHANCED METHOD

As it could be seen from the previous section, although the new methodology does actually solve the problems exhibited by the original procedure, it presents its own disadvantages and limitations. For this reason the ultimate enhanced method has to be necessarily a combination of the original algorithm and the new technique developed above.

While the problems presented by the new method are confined to those values of z relatively close to the origin; fortunately, as it can be verified from figures 1 and 2, the original methodology happens to fail at intermediate and relatively large values of z . Then, we can still count on the original algorithm for computing the ΔR 's near the current element, and use the new algorithm for those ΔR 's far from the current element.

The most important task in here is the selection of the value of z at which the switching between algorithms must occur. After some experimentation, the following empirical criterion happened to provide very good results:

$$z_{\text{limit}} = \{ z \in \mathbb{R} : [g_2(z) * f_2(z) + \xi(z)] = 0.01 \xi(z) \} \quad (11)$$

where $g_2(z)$, $f_2(z)$ and $\xi(z)$ are the functions used for computing $I_2(z)$ in the original algorithm. See (2) and [1].

In summary, the final enhanced methodology can be described as follows:

- 1.- Computation of $I_1(z)$ as defined in (1).
- 2.- Computation of $I_2(z)$:
 - .- for $z < z_{\text{limit}}$, compute $I_2(z)$ as defined in (2).
 - .- for $z \geq z_{\text{limit}}$, compute $I_2(z)$ as defined in (8) and use quartic spline.

3.- Computation of $\Delta R(z)$, which is given by $4/(\pi r_0 h) [I_1(z) + I_2(z)]$.

CONCLUSIONS

Although the new method developed cannot be used in replacement of the original algorithm, the adequate combination of both of them provides a more robust and reliable methodology for the computation of the potential difference integral. However, the complexity and the computational time requirements of this new combined algorithm are greater than those of the original one.

There are still some other restrictions in the methodology, but the main inconvenience related to the original procedure for the solution of the Time Harmonic Field Electric Logging problem has been certainly removed by the modifications introduced in the present work. In fact the implementation of the new algorithm guaranties an accurate solution for almost any conceivable experimental scenario. In this way, the limitations of the technique are now more closely related to the physical and mathematical assumptions behind the model than to the numerical procedures involved in the evaluation of the potential difference integral.

REFERENCES

- [1] Update Report #5: Solution of the Potential Difference Integral by Using Exponential Windowing.
- [2] Update Report #2: The Anderson's Integration Technique.
- [3] Bostick, F.; Smith, H. (1994), Propagation Effects in Electric Logging. University of Texas at Austin.

[4] Update Report #7: Natural Quartic Spline.