A global optimization algorithm applied to the CRS problem

M. Salvatierra, F. Yano, L.T. Santos, J.M. Martínez, R. Andreani, and M. Tygel

email: *lucio@ime.unicamp.br*
keywords: *Traveltime, CRS*

ABSTRACT

The Common Reflection Surface (CRS) method uses the general hyperbolic moveout, which depends on the classical NMO velocity and some other parameters. The CRS parameters are estimated applying a suitable coherence analysis to multicoverage data. The construction of simulated (stacked) zero offset sections in the 2D situation requires three CRS parameters. This work focuses on the estimation of these three parameters, where the coherence analysis is performed using a global optimization algorithm.

INTRODUCTION

The Common Reflection Surface (CRS) stacking method (see, e.g., Müller et al. (1998)) is a recent technique that is establishing itself as a better alternative to the conventional NMO/DMO stacking. As recently shown in Trappe et al. (2001), the CRS stack is able to provide, in many cases,significantly improved stacked sections that represent simulated zero-offset sections. The CRS stacking method provides, in addition to a better stacking, a set of parameters (called the CRS attributes) that convey more information of the propagating medium than the single NMO-velocity parameter that results from the NMO/DMO stack.

For a horizontal seismic line and a constant near surface velocity, v_0, the CRS parameters are given by the triplet (β, K_N, K_{NIP}) where $-\pi/2 \leq \beta \leq \pi/2$ and $-\infty < K_N, K_{NIP} < \infty$. The parameters K_N and K_{NIP} represent the wavefront curvatures of the normal (N) and normal incident point (NIP) waves (Hubral, 1983). The CRS method uses the general hyperbolic traveltime moveout given by

$$t(x, h; A, B, C)^2 = [t_0 + A(x - x_0)]^2 + B(x - x_0)^2 + Ch^2, \tag{1}$$

where

$$A = \frac{2\sin\beta}{v_0},\ B = \frac{2t_0\cos^2\beta}{v_0}K_N,\ C = \frac{2t_0\cos^2\beta}{v_0}K_{NIP}, \tag{2}$$

for all source-receiver pairs in a appropriate neighborhood of a central point x_0, and x and h are the midpoint and half-offset coordinates of the source receiver pair for which the traveltime is being computed. The use of A, B and C instead of the CRS parameters β, K_N and K_{NIP} simplifies the hyperbolic traveltime calculation.

For a given (x_0, t_0) and for fixed parameters (β, K_N, K_{NIP}), the graph of the function $T(x, h) = t(x, h, \beta, K_N, K_{NIP})$ is a surface within the volume of multicoverage data points (x, t, h). If (x_0, t_0) pertains to a reflection event at the ZO section to be simulated and the CRS triplet (β, K_N, K_{NIP}) provides the correct coefficients of the hyperbolic traveltime representation of the respective event, then, according to ray theory, the graph of T is, up to second order, tangent to the event's reflection traveltime surface. As a consequence, the coherence of the data samples $u(x, h, t)$ along the graph of T, for some suitable vicinity (called aperture) of (x_0, t_0), should be high.

The CRS parameter estimation problem is formulated as follows:

For each midpoint and traveltime (x_0, t_0) at the ZO section to be simulated, find the CRS parameter triplet (β, K_N, K_{NIP}) for which a coherence function attains a minimum for source-receiver pairs within a given spatial aperture around x_0 and for time samples within a time around t_0.

The present work is concerned with the use of a global optimization algorithm to obtain the CRS parameters. First, we obtain the coefficients A, B and C, and then we recover the CRS attributes using the relations given by equation (2). A global minimization strategy for each pair (x_0, t_0) is applied until some predefined time, based on the amount of data and the quality of the solution, is reached.

GLOBAL OPTIMIZATION PROBLEM

For each midpoint x_0 and zero-offset traveltime t_0, in the simulated ZO section to be constructed, the CRS problem consists of finding A, B, C for which the coherence function (semblance) calculated on the traces inside an aperture around $(x = x_0, h = 0)$, is maximum. In mathematical terms, we have to minimize the semblance function

$$S(A,B,C) = \frac{1}{N} \frac{\displaystyle\sum_{|t-t_i|\le\tau} \left[\sum_{i=1}^{N} U_i(t) \right]^2}{\displaystyle\sum_{|t-t_i|\le\tau} \left[\sum_{i=1}^{N} U_i(t)^2 \right]} , \tag{3}$$

where $U_i(t)$ is the interpolated value of trace i at time t,

$$t_i = t_i(A,B,C) = t(x_i, h_i; A, B, C) \tag{4}$$

is the hyperbolic traveltime data given by equation (1), x_i and h_i are, respectively, the midpoint and half-offset of trace number i, τ is a time window around t_i and N is the total number of traces in the multicoverage data. From the solution obtained, we restore (β, K_N, K_{NIP}) using relations (2).

In general, specially for real data, there is a high number of local minimizers of the semblance function and so, local optimization algorithms might not be very effective for finding global minimizers. Moreover, local methods may converge for critical points, not necessarily local minimizers. Therefore, the number of possible solutions that are not global solutions, is enormous and a global search is necessary.

From now on we will consider the problem of finding a global minimizer of a function $f : \mathbb{R}^n \to \mathbb{R}$, twice continuously differentiable, over the set $\Omega = \{x \in \mathbb{R}^n \mid \ell \le x \le u\}$. The CRS problem is such that $n = 3$, $x = (A, B, C)$ and $f(x) = -S(A, B, C)$ (recall that maximizing S is equivalent to minimizing $-S$) and the box-parameters ℓ and u are conveniently chosen.

Our global optimization procedure consists of using a local algorithm for finding a critical point x^* and, then, trying to "escape" to a new starting point y such that $f(y) < f(x^*)$. For the local minimization we have chosen the Box Euclidian Trust Region Algorithm (BETRA) proposed by Andretta et al. (2005). BETRA is a local method with guaranteed convergence to critical points.

The strategy to escape from the critical point is to make a movement along a Lissajous curve that passes through the critical point x^*. A Lissajous curve in $\mathbb{R}^n$ is defined by the parametric equations

$$\gamma(r) = (\cos(\theta_1 r + \varphi_1),\ \cos(\theta_2 r + \varphi_2),\ \ldots, \cos(\theta_n r + \varphi_n)) \ , \quad r \in \mathbb{R}, \tag{5}$$

where $\varphi_j,\ \theta_j \in \mathbb{R}$ $(j = 1, 2, \ldots, n)$ and the θ_j's are linearly independent over $\mathbb{Q}$, i.e., if $d_1, d_2, \ldots, d_n \in \mathbb{Q}$ and $\sum_{j=1}^{n} d_j\ \theta_j = 0$ then $d_1 = \cdots = d_n = 0$. Figure 1 shows some examples of these curves in two and three dimensions, for the case of $\theta_1 = \sqrt{2}$, $\theta_2 = \sqrt{3}$, $\theta_3 = \sqrt{5}$, and $\varphi_1 = \varphi_2 = \varphi_3 = \pi/2$.

Andreani et al. (2004) proved that the image of the curve γ is dense in the box $[-1, 1]^n$. Therefore, through the use of the linear transformation $\Gamma(r) = [\ell+u+(u-\ell)\gamma(r)]/2$, we can construct a dense curve Γ in Ω and choose the φ_j's so that $\Gamma(0) = x^*$. The movement along Γ is made testing the candidates $x(\alpha) = \Gamma(\alpha/(1 - |\alpha|))$ with $\alpha = 1/2, -1/2, 1/3, 2/3, -1/3, -2/3, 1/5, 2/5, 3/5, 4/5, -1/5, -2/5, -3/5, \ldots$, until $f(x(\alpha)) < f(x^*)$.

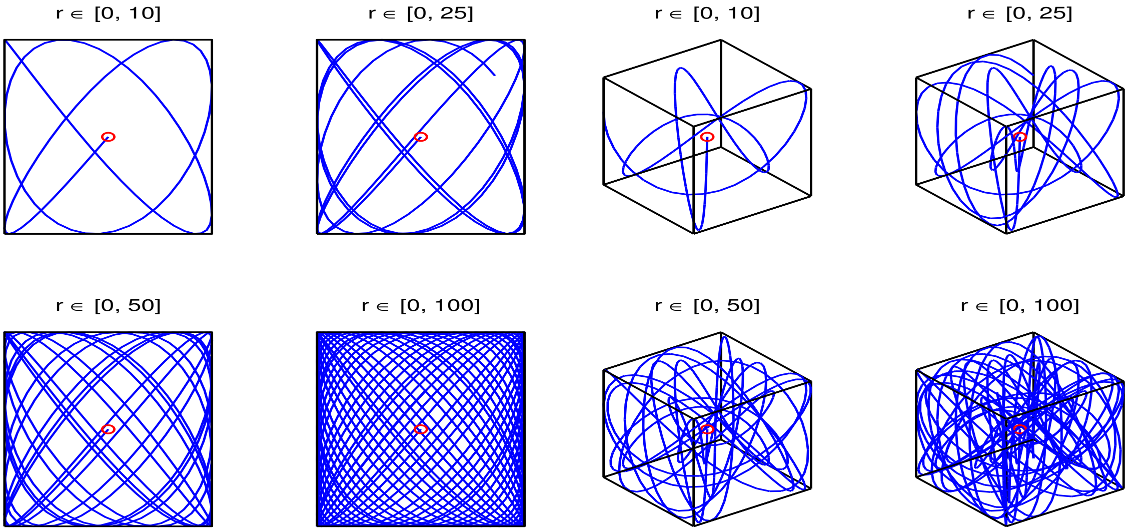

Figure 1: Examples of Lissajous curves in two (left) and three (right) dimensions.

GLOBAL OPTIMIZATION ALGORITHM

After many trials and errors we define the global algorithm described below. Besides the local minimization and the escaping phases we introduced a multistart procedure for generating different initial points, defining criteria for discarding poor initial points and establishing two upper limits: `TMAX` for the total run of the algorithm and `TESC` for each call of the escaping phase.

Algorithm:

Step 0 [Initialization]

Choose TMAX, TESC ≥ 0. Set $k \leftarrow 1$, $f_{\min} \leftarrow \infty$, $\mathcal{C} \leftarrow \emptyset$, $\mathcal{A} \leftarrow \emptyset$ and $\delta = 0.1 \times \min\limits_{1 \leq i \leq n} \{u_i - \ell_i\}$.

Step 1 [Random Choice]

Choose a random uniformly distributed initial point $x_{I_k} \in \Omega$.
If $k = 1$ update $\mathcal{A} \leftarrow \mathcal{A} \cup \{x_{I_k}\}$ and go to Step 5.

Step 2 [Functional Discarding Test)

Set $f_{\max} \leftarrow \max\{f(x) \mid x \in \mathcal{A}\}$.
Compute the probability $Prob$ of discarding x_{I_k}:
- If $f(x_{I_k}) \leq f_{\min}$, $Prob \leftarrow 0$;
- If $f(x_{I_k}) \geq f_{\max}$, $Prob \leftarrow 0.8$;
- If $f_{\min} < f(x_{I_k}) < f_{\max}$, $Prob \leftarrow 0.8 \times (f(x_{I_k}) - f_{\min})/(f_{max} - f_{\min})$.

Discard x_{I_k} with probability $Prob$. If x_{I_k} was discarded return to Step 1.

Step 3 [Neighborhood Discarding Test]

Set $d_{\min} \leftarrow \min\{\|x - x_{I_k}\|_\infty \mid x \in \mathcal{C}\}$ and update $Prob$:
- If $d_{\min} \leq \delta$, $Prob \leftarrow 0.8$;
- If $d_{\min} > \delta$, $Prob \leftarrow 0$.

Discard x_{I_k} with probability $Prob$. If x_{I_k} was discarded return to Step 1.

Step 4 [10–Iterations Discarding Test)

Perform 10 iterations of the local method obtaining iterate $x_{10,k}$. Set $f_{10} \leftarrow f(x_{10,k})$ and $f_{aux} \leftarrow f(x_{I_k}) - 0.1 \times (f(x_{I_k}) - f_{min})$ and update $Prob$:
- If $f_{10} \leq f_{\min}$, $Prob \leftarrow 0$;

· If $f_{10} \geq f_{aux}$, $Prob \leftarrow 0.8$;
· If $f_{\min} < f_{10} < f_{aux}$, $Prob \leftarrow 0.8 \times (f_{10} - f_{\min})/(f_{aux} - f_{\min})$.
Discard x_{I_k} with probability $Prob$. If x_{I_k} was discarded, return to Step 1. Otherwise, update $\mathcal{A} \leftarrow \mathcal{A} \cup \{x_{I_k}\}$.

Step 5 [(Local Minimization]
Taking x_{I_k} as an initial point, execute the local algorithm (in this work we used BETRA) for obtaining a critical point, $x_{*,k}$. Update the set of critical points $\mathcal{C} \leftarrow \mathcal{C} \cup \{x_{*,k}\}$ and the best functional value, $f_{\min} \leftarrow \min\{f_{\min}, f(x_{*,k})\}$.
If $f_{\min} = f(x_{*,k})$ set $x_{\min} \leftarrow x_{*,k}$. If the CPU time exceeds TMAX, STOP the algorithm.

Step 6: [Escaping Phase]
Using the Lissajous curve that passes through $x_{*,k}$ try to obtain $x(\alpha)$ such that $f(x(\alpha)) < f(x_{*,k})$ in less than TESC units of time.. In case of success, set $k \leftarrow k+1$ and $x_{I_k} \leftarrow x(\alpha)$ and go to Step 5. Otherwise, set $k \leftarrow k+1$ and return to Step 1.

NUMERICAL EXPERIMENTS

To analyse the performance of the global optimization procedure described above, we generate a multicoverage data for the synthetic model depicted in Figure 2. The data was modeled by ray tracing, using the package `Seis88` (Červený and Pšenčik, 1984). We apply the global algorithm for each pair (x_0, t_0), with $x_0 \in [3, 7]$ km and increment $\Delta x_0 = 25$ m, and $t_0 \in [0, 4]$ s with time sample $\Delta t_0 = 0.04$ s. Therefore, to solve the CRS problem we have made $161 \times 101 = 16261$ calls of the algorithm. All the experiments were run on a 2.8 GHz Intel Pentium IV Computer with 2 Gb of RAM in double precision Fortran. The linearly independent parameters that define the Lissajous curves (5) are chosen to be the square roots of the first three prime numbers, i.e., $\theta_1 = \sqrt{2}$, $\theta_2 = \sqrt{3}$ and $\theta_3 = \sqrt{5}$. For the box constraints we used $-2 \leq A \leq 2$, $-4 \leq B \leq 4$ and $-4 \leq C \leq 4$.

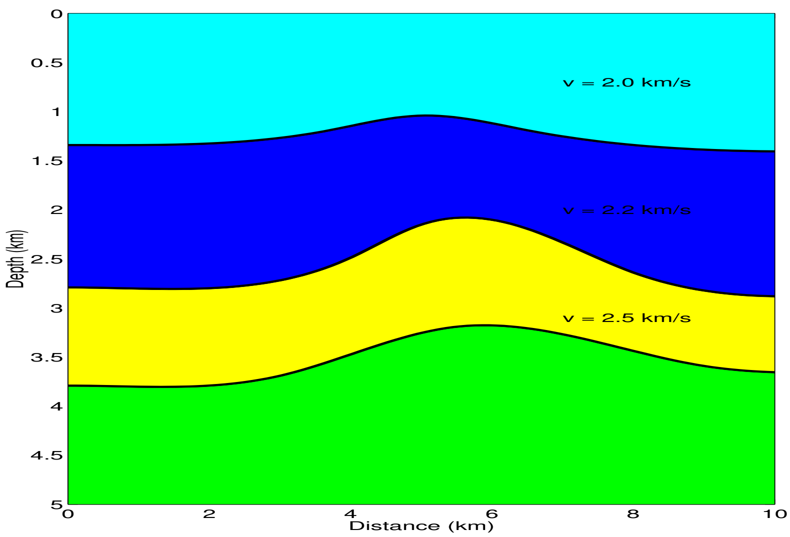

Figure 2: Synthetic model for the numerical experiments.

We tested the global optimization procedure for three increasing values of the maximum time allowed for each run of the algorithm: `TMAX` = 0 s (no escape/global search), 1 s and 2 s. The parameter `TESC` was chosen equal to 0.5 s. The respective total CPU times (all the 16221 runs of the algorithm) are 1h51m20s, 7h15m22s and 11h33m16s. Figure 3 shows the final semblance panels obtained. The improvement with the global search is evident.

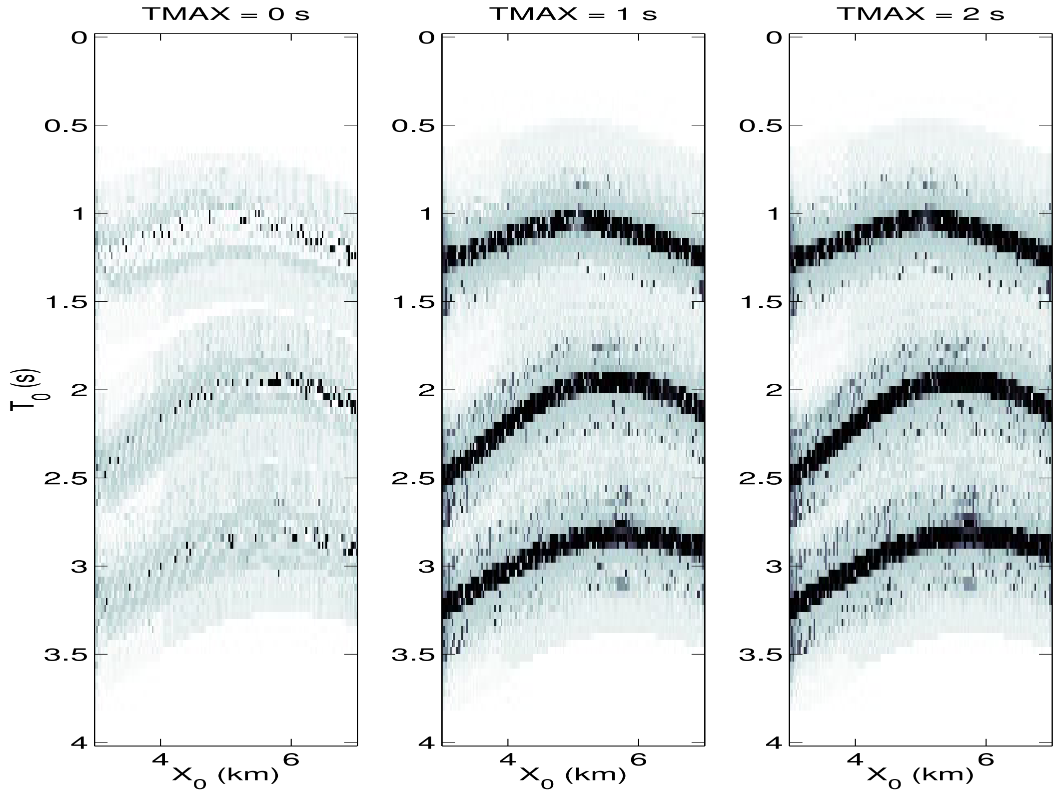

Figure 3: Semblances panels for different values of `TMAX`.

For a better analysis of the impact of the global procedure we plot in Figure 4 the semblance functions at $x_0 = 4$ km, $x_0 = 5$ km, $x_0 = 6$ km. As can be observed, as the time allowed for the global search increases, the semblance function also increases. Moreover, only with the global search it is possible to reach large values for the semblance function.

To compare the quality of the CRS parameters, we show in Figures 5, 6 and 7 the recovered attributes β, K_N and K_{NIP}, respectively, for all the three interfaces. The results obtained agree with the previous statement about the semblance: only with the global strategy it is possible to recuperate the majority of the exact values. Indeed, `TMAX` = 1 s is sufficient for a good estimation in all cases.

CONCLUSIONS

The CRS problem in the 2-D situation requires the maximization of the semblance function, depending on three parameters. This work focuses on the estimation of these parameters, where the coherence analysis is performed using a global optimization algorithm.

The main contributions of this work are:

- The search for the "general" parameters A, B and C, instead of the "original" parameters β, K_N and K_{NIP}. This simplification avoids the computation of trigonometric functions, reducing the time for the evaluation of the semblance function. Moreover, v_0 can be given a posteriori, and then it is possible to propose different strategies for recovering the original parameters.
- A global optimization strategy that can be applied in association with any local optimization method.
- A new search procedure for escaping from local solutions, based on Lissajous curves.

From the numerical results, we conclude that the global optimization strategy introduced in this work has the potential to be a powerful tool, not only for solving the CRS problem but also for any geophysical problem that requires global solutions.

Further investigation is being carried to analyse the behavior of the global optimization algorithm when applied to the CRS problem with noisy data and, of course, real data.

ACKNOWLEDGEMENTS

This work has been partially supported by the *National Council of Scientific and Technological Development (CNPq)*, Brazil, the *Research Foundation of the State of São Paulo (FAPESP)*, Brazil, PRONEX, Brazil, and the sponsors of the *Wave Inversion Technology (WIT) Consortium*, Germany.

REFERENCES

Andreani, R., Martínez, J. M., Salvatierra, M., and Yano, F. (2004). Global order-value optimization by means of a multistart harmonic oscillator tunneling strategy. Technical Report MCDO 16/7/04, Department of Applied Mathematics, IMECC – UNICAMP.

Andretta, M., Birgin, E. G., and Martínez, J. M. (2005). Practical active-set euclidian trust-region method with spectral projected gradients for bound-constrained minimization. To appear in *Optimization*.

Hubral, P. (1983). Computing true amplitude refelctions in a laterally inhomogeneous earth. *Geophysics*, 48:1051–1062.

Müller, T., Jäger, R., and Höcht, G. (1998). Common reflection surface stacking method - imaging with an unknown velocity model. *68th EAGE Conference & Exhibition, Expanded Abstracts*, pages 1764–1767.

Trappe, H., Gierse, G., and Pruessmann, J. (2001). Case studies show the potential of common reflection surface stack — structural resolution in the time domain beyond the conventional NMO/DMO stack. *First Break*, 19:625–633.

Červený, V. and Pšenčik, I. (1984). Seis83 – numerical modeling of seismic wave fields in 2-D laterally varying structures by the ray method. *Report SE–35*, 48:36–40.

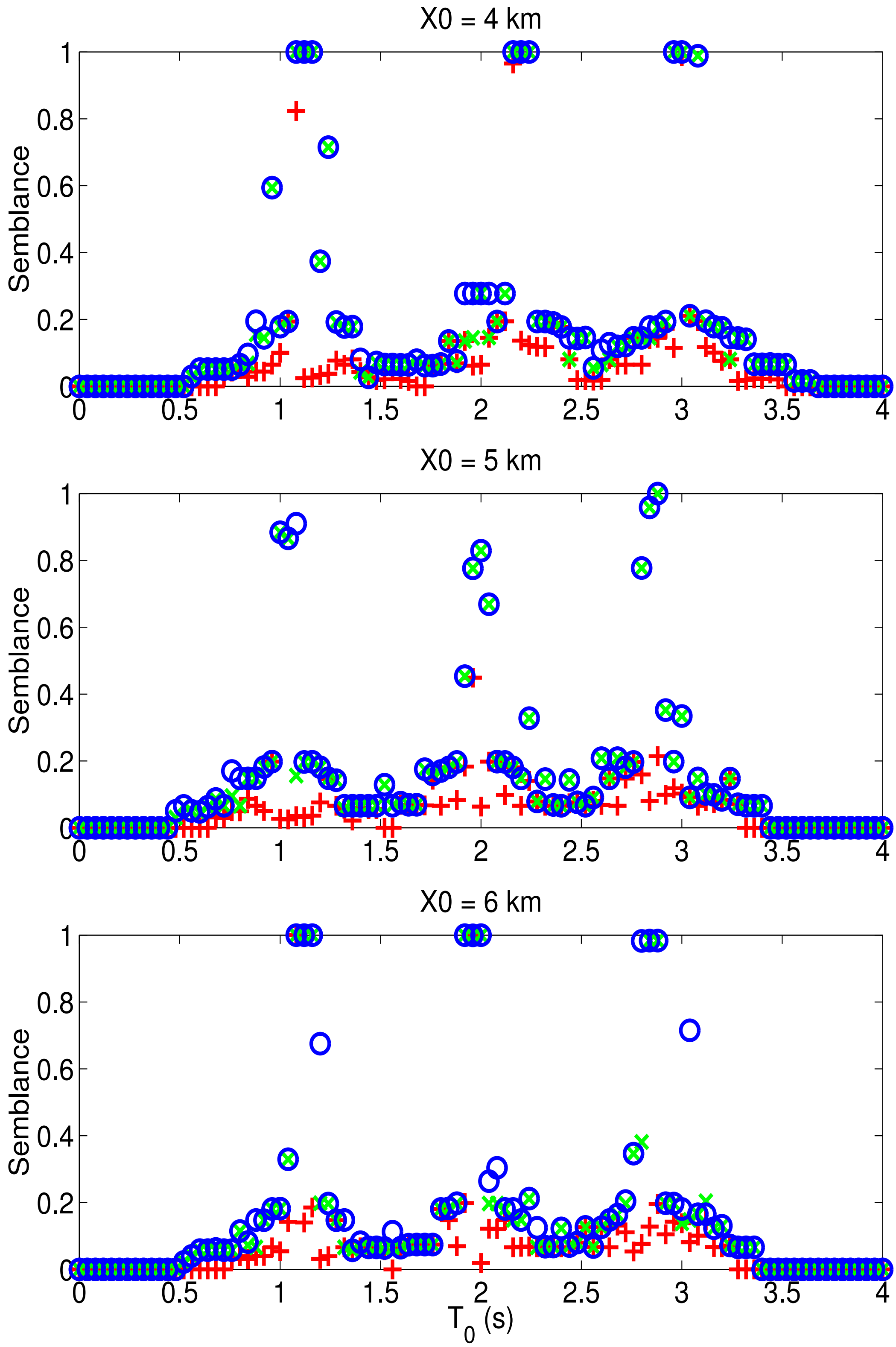

Figure 4: Semblance function for three different values of x_0: (+) TMAX = 0 s, ($\times$) TMAX = 1 s, and ($\circ$) TMAX = 2 s.

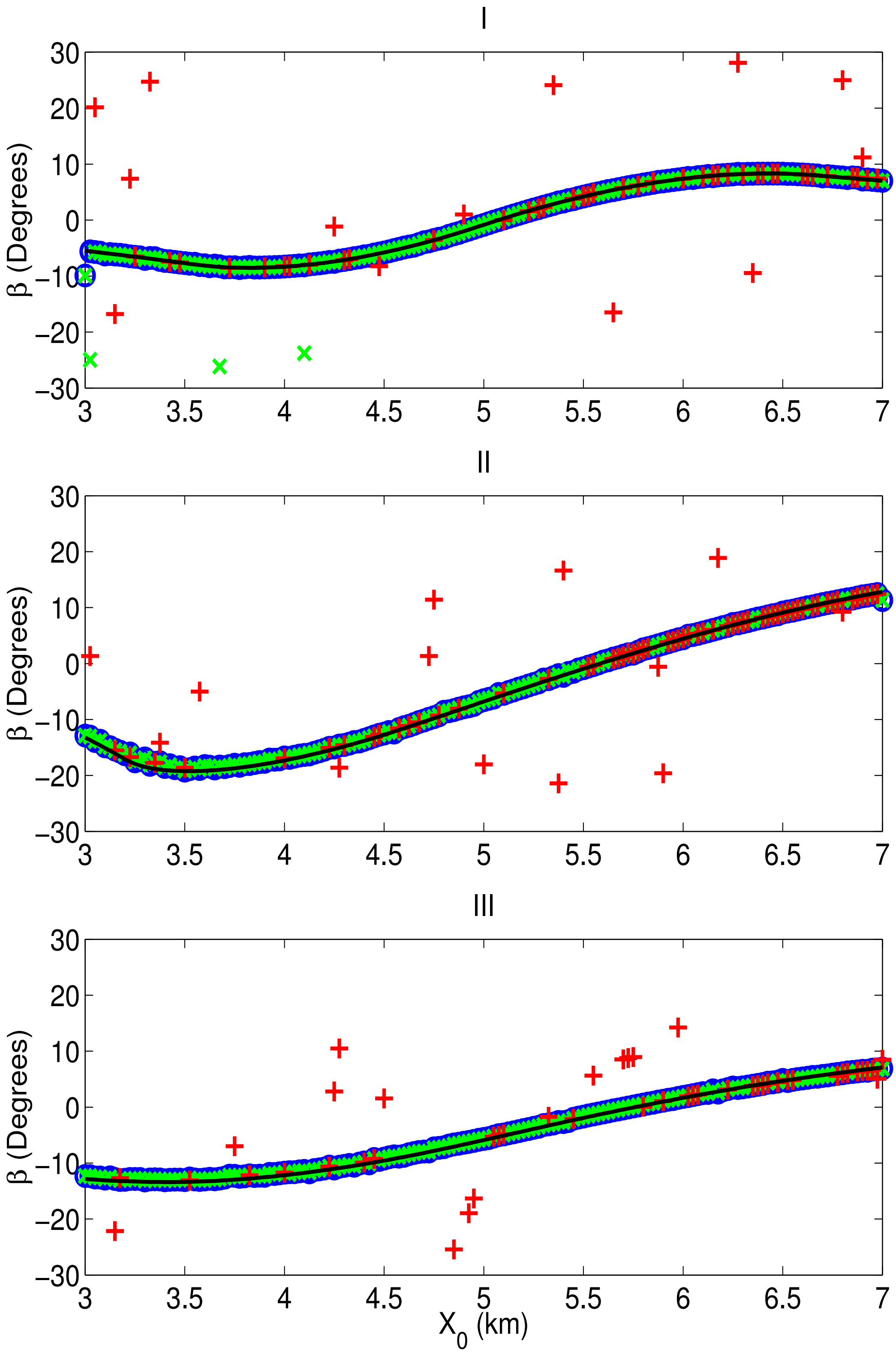

Figure 5: Recovered β for the three interfaces: (—) Exact, (+) `TMAX` = 0 s, (×) `TMAX` = 1 s, and (○) `TMAX` = 2 s.

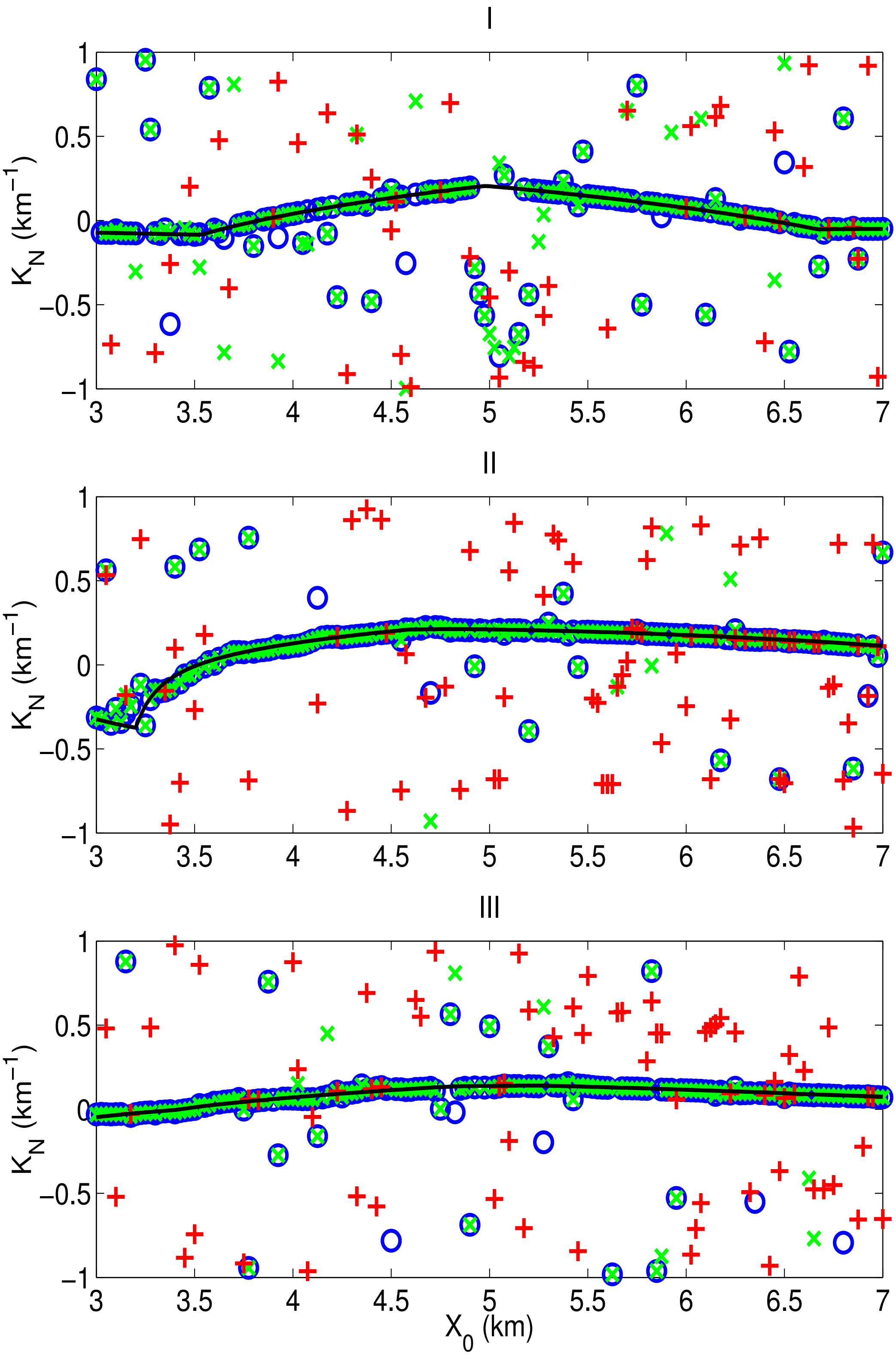

Figure 6: Recovered K_N for the three interfaces: (—) Exact, (+) TMAX = 0 s, ($\times$) TMAX = 1 s, and ($\circ$) TMAX = 2 s.

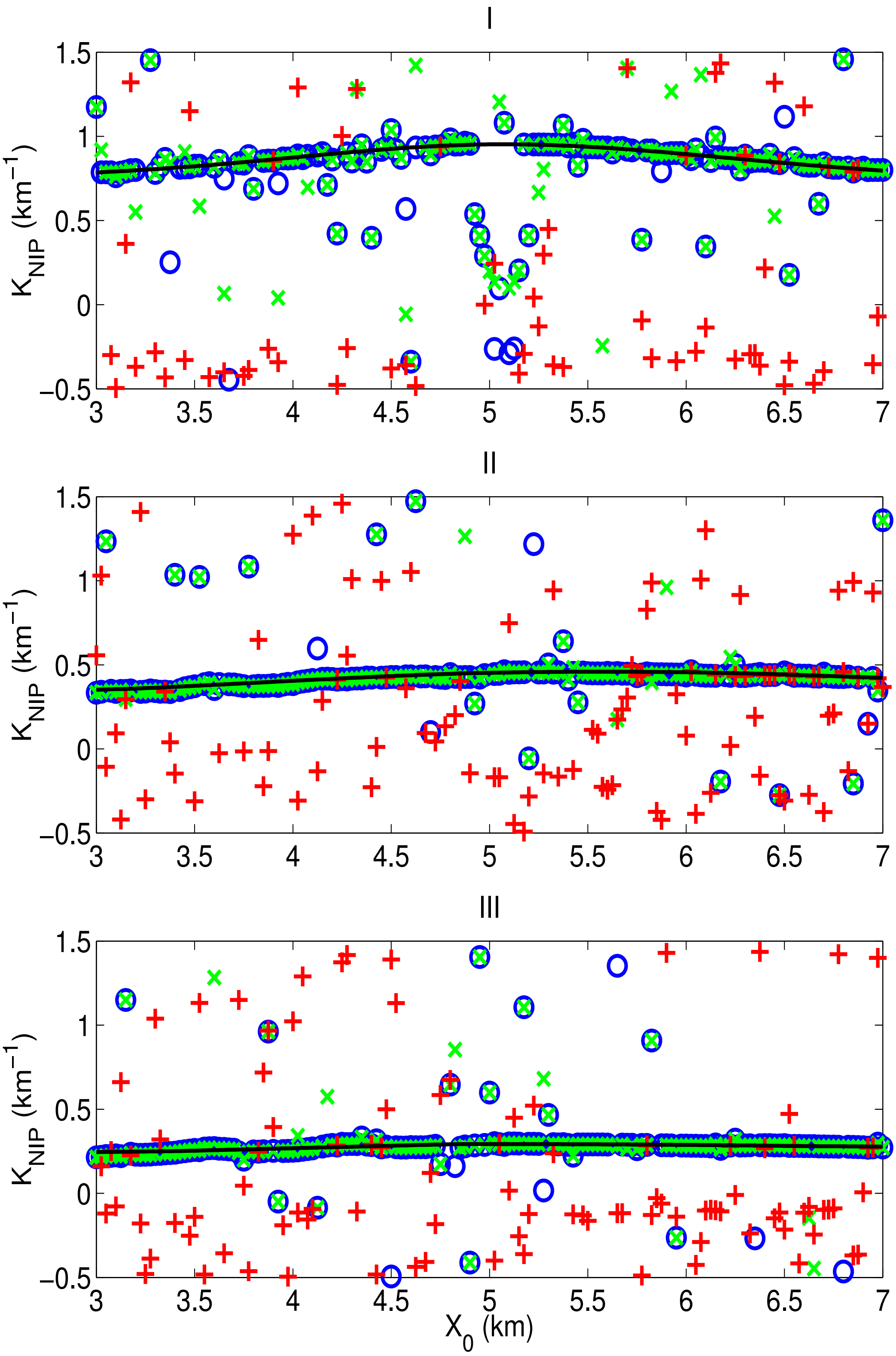

Figure 7: Recovered K_{NIP} for the three interfaces: (—) Exact; (+) `TMAX` = 0 s, (×) `TMAX` = 1 s, and (○) `TMAX` = 2 s