

Research Article

An Algorithm for Curve Identification in the Presence of Curve Intersections

Giorgio Gnecco 

IMT School for Advanced Studies, Piazza S. Francesco 19, 55100 Lucca, Italy

Correspondence should be addressed to Giorgio Gnecco; giorgio.gnecco@imtlucca.it

Received 9 September 2018; Accepted 26 November 2018; Published 9 December 2018

Academic Editor: Vyacheslav Kalashnikov

Copyright © 2018 Giorgio Gnecco. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the paper, a simple but effective algorithm is proposed to solve the problem of curve identification in the presence of curve intersections. The problem is motivated by the fact that, in several engineering problems, one has to deal with samples of possibly intersecting curves, and a successful identification can significantly improve their visualization through a successive function interpolation step. Numerical results are presented, together with possible areas of application of the algorithm.

1. Introduction

In several problems in engineering, one has to deal with samples of possibly intersecting curves. As an example, when studying dispersive wave-propagation properties of phononic [1] and photonic [2] periodic structures, one simultaneously obtains samples of various curves by evaluating the eigenvalues of a suitable Hermitian matrix (or, more often, the generalized eigenvalues [3] of a pair of suitable Hermitian matrices) depending continuously on a scalar parameter, for various discrete choices of such a parameter. In this case, curve intersections can arise in the presence of eigenvalue degeneracy, i.e., for those values of the parameter for which at least one eigenvalue has multiplicity larger than one as a root of the characteristic polynomial. As a second example, curves may represent the positions of point objects (e.g., of markers detected by cameras in a motion-capture system [4]) with respect to time. In this situation, even assuming that no two objects can occupy the same position at the same time, intersections may still arise either in case of noisy measures or in case of only two components of the three-dimensional position vectors (e.g., the ones belonging to either a horizontal or a vertical plane) being measured.

Attributing the curve samples to the correct curves (where the “correct” choice may be defined, e.g., as the one making the resulting reconstructed curves as smooth as possible) not only is of interest per se, but also represents

an important preliminary step for a successive curve interpolation procedure, which can significantly improve their visualization in case of an initially small number of samples. The latter may be motivated, e.g., by a possibly high cost of generating the single sample or simply by a technical limit in the acquisition process (e.g., a small frame rate). For instance, in the example above related to wave propagation, it often happens that the Hermitian matrix has large dimension, which makes the computational time required to solve the eigenvalue problem large for each choice of the parameter, especially if a high precision in the eigenvalue computations is needed. This is the typical case of eigenvalue problems of interest in engineering applications, as they are often ill-conditioned [5].

In this framework, in the paper we propose and test a simple but effective algorithm for the problem of curve identification in the possible presence of curve intersections. The algorithm requires only, as a very reasonable assumption, that the curves do not intersect for at least one value of the independent variable (for simplicity and without loss of generality, the first one, in the description of the algorithm provided in the next section). Related works are [6, 7] (an extended version of which is [8]). However, [7] presents a multitarget tracking algorithm which requires, among other assumptions, the availability of a model of the “law of motion” for each curve, whereas [6] presents an approach more similar to the one adopted in the present work (by using current possible curve

orientations to estimate next curve points, emulating human perception of intersecting curves), but does not process all the curves simultaneously. Instead, it relies on a preliminary rough curve detection based on an algorithm proposed in [9], followed by a correction of the estimated curve at successive junctions, based on current possible curve orientations for the estimation of next curve points. In more detail,

- (i) the correction of the preliminarily estimated curve is implemented only when such a curve is characterized by large slope changes (i.e., numerical approximations of second derivatives) on a discretization of the horizontal axis;
- (ii) the corrected estimated curve is constrained to have small slope changes on the same discretization.

These two features may lead to an incorrect curve identification, respectively, when

- (i) two intersecting curves are erroneously identified by the algorithm proposed in [9]; in this case, when their second derivatives are small with respect to a given threshold, no correction is implemented;
- (ii) the curves to be identified have large second derivatives in some parts of the domain.

In contrast, the algorithm proposed in this work

- (i) always uses (apart from its first two initialization steps (see Section 2 for their description)) current possible curve orientations for the estimation of next curve points, without needing a preliminary rough curve identification and thresholds on the second derivatives of preliminarily estimated curves;
- (ii) does not require any bounds on the slope changes of each curve, allowing in principle identifying even curves with large slope changes.

The paper is organized as follows. In Section 2, the proposed algorithm is presented. Section 3 provides numerical results related to the application of the algorithm to four different tests. Section 4 applies the algorithm to the problem of identifying smooth eigenvalue curves. Finally, Section 5 concludes the paper and discusses possible extensions.

2. Description of the Algorithm

To describe the proposed algorithm, the following notation is introduced. Let N, l be positive integers and, for $n = 1, \dots, N$, let $\underline{f}^{(n)} : [0, 1] \rightarrow \mathbb{R}^l$ be N smooth functions (i.e., smooth curves parameterized by the independent variable x). The functions $\underline{f}^{(n)}(x)$ are uniformly sampled for a total of $S > 2$ samples each, located at $x = 0, 1/(S-1), 2/(S-1), \dots, (S-2)/(S-1), 1$. Moreover, it is assumed that the samples are distinct for $x = 0$. For $s = 0, \dots, S-1$, the set of samples evaluated at $x = s/(S-1)$ is denoted by $F_s := \{\underline{f}^{(n)}(s/(S-1)) : n = 1, \dots, N\}$. The idea behind the algorithm is the following. For each sampled value $s/(S-1)$ of the independent variable, the algorithm attributes to the n -th

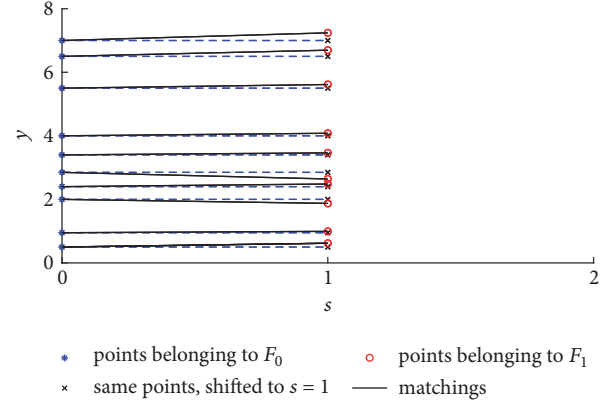


FIGURE 1: An example of the sets F_0 and F_1 (with $n = 10$) and of their matchings obtained using *Step 2* of the proposed algorithm.

reconstructed curve a sample $\underline{y}_s^{(n)} \in F_s$, and (for $s+1 = 2, \dots, S-1$) a feasible increment $\underline{i}_{s+1}^{(n)} \in \mathbb{R}^l$. The two are combined (apart from the cases $s = 0, 1$ and $S-1$) to generate, still for the n -th reconstructed curve, a prediction $\underline{y}_{s+1}^{(n)}$ of the point obtained for the successive sampled value $(s+1)/(S-1)$ of the independent variable. The N predictions are compared with the set of actual samples F_{s+1} , to attribute each of them, in a one-to-one way, to one of the reconstructed curves, generating the points $\underline{y}_{s+1}^{(n)}$ and, apart when $s+1 = S-1$, also the next feasible increments $\underline{i}_{s+2}^{(n)}$. To conclude, each reconstructed (sampled) curve is the collection $C^{(n)} := \{(s/(S-1), \underline{y}_s^{(n)}) : s = 0, \dots, S-1\}$. In more detail, the algorithm works as follows.

Step 1. Each element of F_0 is attributed to one of the N reconstructed curves. For $n = 1, \dots, N$, let $\underline{y}_0^{(n)} \in F_0$ be the sample attributed to the n -th reconstructed curve.

Step 2. The elements of F_1 are attributed (matched) to the elements of F_0 (then, due to *Step 1*, also to the corresponding reconstructed curves) in the following way (possible ties in the next descriptions are solved by coin flipping). A copy \tilde{F}_0 of F_0 and a copy \tilde{F}_1 of F_1 are constructed. Then, one denotes by \underline{y}_1 the nearest element (in terms of the Euclidean distance) of \tilde{F}_1 from \tilde{F}_0 . Such an element is attributed to the element \underline{y}_0 of \tilde{F}_0 that minimizes its own distance from \underline{y}_1 . Then, \underline{y}_0 and \underline{y}_1 are removed from \tilde{F}_0 and \tilde{F}_1 , respectively, and the procedure is repeated until both sets become empty. Finally, for $n = 1, \dots, N$, let $\underline{y}_1^{(n)} \in F_1$ be the sample attributed to the n -th reconstructed curve. Figure 1 shows an example of the sets F_0 and F_1 and of their matchings obtained using *Step 2* of the proposed algorithm.

Step 3. This step is repeated for $s = 2, \dots, S-1$. For each such s , the elements of F_s are attributed (matched) to the elements of F_{s-1} (then, due to either *Step 2* or the previous iteration of *Step 3*, also to the corresponding reconstructed curves) in the

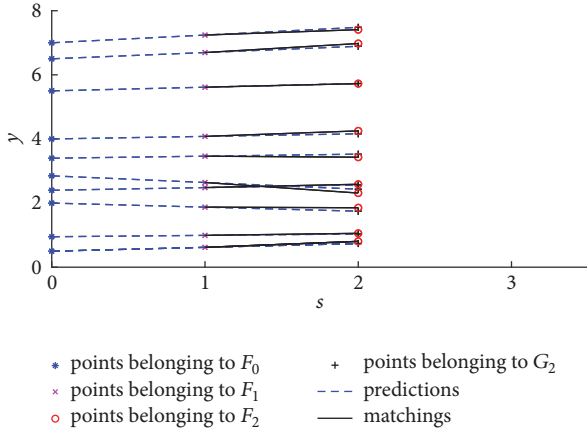


FIGURE 2: An example of the sets F_1 and F_2 (with $n = 10$) and of their matchings obtained using *Step 3* of the proposed algorithm.

following way (again, possible ties in the next descriptions are solved by coin flipping). For each $n = 1, \dots, N$, the feasible direction $\underline{i}_s^{(n)} := \underline{y}_{s-1}^{(n)} - \underline{y}_{s-2}^{(n)}$ is constructed, together with the estimate $\underline{\hat{y}}_s^{(n)} := \underline{y}_{s-1}^{(n)} + \underline{i}_s^{(n)}$ of the next point on the n -th reconstructed curve. The set of such estimates $G_s := \{\underline{\hat{y}}_s^{(n)} : n = 1, \dots, N\}$ is defined. Copies \tilde{F}_s , \tilde{F}_{s-1} , and \tilde{G}_s of F_s , F_{s-1} , and G_s , respectively, are constructed. Then, one denotes by $\underline{\tilde{y}}$ the nearest element (in terms of the Euclidean distance) of \tilde{F}_s from \tilde{G}_s . Such an element is attributed to the element $\underline{\tilde{y}}_{s-1}$ of \tilde{F}_{s-1} whose associated estimate $\underline{\hat{y}}_s \in G_s$ minimizes its own distance from $\underline{\tilde{y}}$. Then, $\underline{\tilde{y}}_{s-1}$, $\underline{\tilde{y}}_s$, and $\underline{\hat{y}}_s$ are removed from \tilde{F}_{s-1} , \tilde{F}_s , and \tilde{G}_s , respectively, and the procedure is repeated until all the three sets become empty. Finally, for $n = 1, \dots, N$, let $\underline{y}_s^{(n)} \in F_s$ be the sample attributed to the n -th reconstructed curve. Figure 2 shows an example of the sets F_1 and F_2 and of their matchings obtained using *Step 3* of the proposed algorithm, for $s = 2$.

The reason for which the algorithm is able to generate smooth reconstructed curves is that it penalizes large slope changes when attributing points to the reconstructed curves (no slopes are used in *Step 2* since they are not yet available at that stage). As an example, for $N = 2$, $l = 1$, and $S = 10$, Figure 3 compares, for a possible situation, (a) the portions of the reconstructed curves ($C^{(1)}$ and $C^{(2)}$) provided by the proposed algorithm up to the stage $s = 4$ and (b) the ones ($\tilde{C}^{(1)}$ and $\tilde{C}^{(2)}$) obtained by swapping the two labels associated with that stage. The figure shows clearly that a much larger smoothness is obtained in the first case.

Remark 1. If the curves do not intersect in $x = 0$ and sampling is sufficiently fast, then Steps 1 and 2 provide a correct identification of the curves at $x = 0, 1/(S-1)$ and also of their initial slopes. In this way, at its first iteration (i.e., for $s = 2$), *Step 3* receives correct estimates of the feasible directions $\underline{i}_s^{(n)} = \underline{y}_{s-1}^{(n)} - \underline{y}_{s-2}^{(n)}$, for $n = 1, \dots, N$.

Remark 2. *Step 2* of the algorithm could be replaced by the optimal solution to an instance of the minimum cost Euclidean bipartite matching problem [10], since such a solution assigns, in a one-to-one way, each point in F_0 to a point in F_1 in such a way to minimize the sum of the Euclidean distances of the pairs of matched points. However, no difference between such attribution and the one produced by *Step 2* is obtained if, for each point $\underline{\hat{y}}_0$ in F_0 , there is a point $\underline{\hat{y}}_1$ in F_1 (a different $\underline{\hat{y}}_1$ for each $\underline{\hat{y}}_0$) whose distance from $\underline{\hat{y}}_0$ is significantly smaller than the distance from $\underline{\hat{y}}_0$ of any other point in F_1 (this situation occurs, e.g., in the case reported in Figure 1). Similarly, also *Step 3* of the algorithm could be replaced, for $s = 2, \dots, S-1$, by the optimal solution to an instance of the minimum cost Euclidean bipartite matching problem, substituting F_0 and F_1 with G_s and F_s , respectively. However, also in this case, no difference in the attribution is obtained, under a condition analogous to the one reported above for *Step 2* (again, this situation occurs, e.g., in the case reported in Figure 2). The advantage of Steps 2 and 3 is that they are less computationally demanding than solving the corresponding instances of the minimum cost Euclidean bipartite matching problem (indeed, the algorithm proposed in [10] to solve the minimum cost Euclidean bipartite matching problem requires $O(n^{2.5} \log n)$ time (it is worth recalling that the minimum cost Euclidean bipartite matching problem can be formulated as an integer linear programming problem, but also as the associated relaxed continuous linear programming problem, whose domain has the form $\{\underline{x} \in \mathbb{R}^n : A\underline{x} \leq \underline{b} \text{ and } \underline{x} \geq \underline{0}\}$, where A is a suitable constraint matrix; indeed, since A is totally unimodular and \underline{b} has integer components, the vertices of such domain have integer coordinates [11])).

3. Numerical Tests and Results

To evaluate the effectiveness of the algorithm, we consider the four following numerical tests, which have been implemented in MATLAB 8.2 (R2017a) [12].

In the first test, we generate N scalar-valued functions (i.e., $l = 1$) of the form $f^{(n)}(x) = \sum_{k=0}^3 c_k^{(n)} x^k$, where the coefficients $c_k^{(n)}$ are chosen as realizations of independent random variables uniformly distributed on $[-1/2, 1/2]$. In the following, we choose $N = 10$ and $S = 100$. The obtained samples are plotted in Figure 4(a). In the figure, they are plotted using the same color, because the identity of each curve is not provided a priori to the algorithm, but the algorithm has to find it. In Figure 4(b), the curves reconstructed by the algorithm are reported, using a different color for each curve. It is clear from the figure that the reconstructed curves are smooth. Incidentally, in this case, each reconstructed curve coincides with one of the original curves (so, no separate figure is presented to show the ground truth).

In the second test, we consider a similar setting as before. In this case, however, the number of samples per curve is only $S = 10$. The samples provided as inputs to the algorithm are represented as circles in Figure 5(a). As shown by Figure 5(b), also in this case the proposed algorithm is able to produce smooth curves. Moreover, its preliminary identification of smooth curves makes the successive linear interpolation step

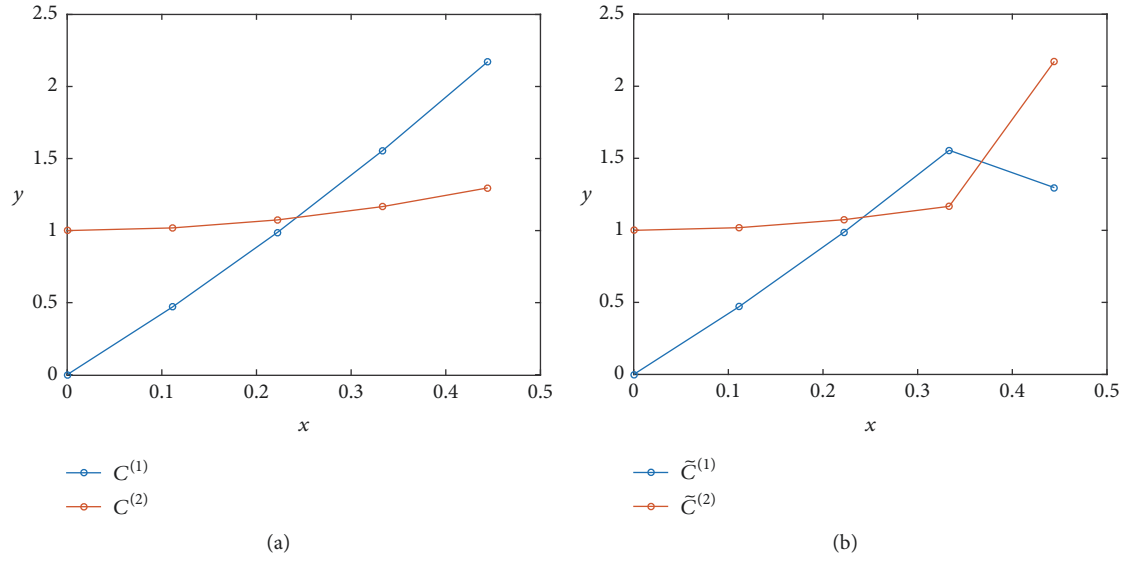


FIGURE 3: (a) An example of portions of reconstructed curves ($C^{(1)}$ and $C^{(2)}$) provided by the proposed algorithm. (b) Portions of the reconstructed curves ($\tilde{C}^{(1)}$ and $\tilde{C}^{(2)}$) obtained by swapping the fifth labels in the portions of reconstructed curves shown in (a).

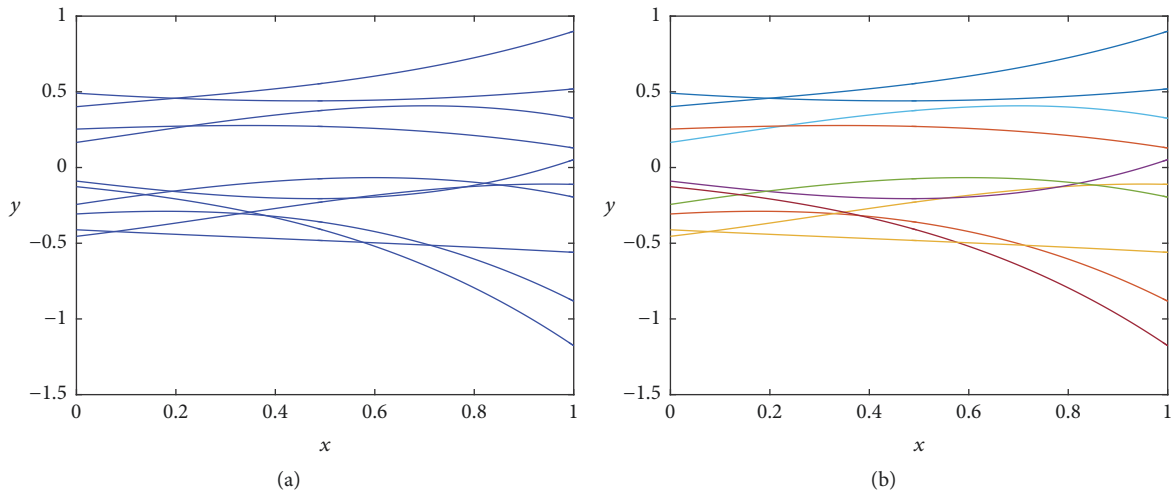


FIGURE 4: (a) Samples used for the first numerical test. (b) Curves reconstructed by the proposed algorithm in the first numerical test.

(also presented in Figure 5(b)) particularly effective for an improved visualization of the curves reconstructed by the algorithm.

The third test refers to the case $l = 2$, i.e., to curves in \mathbb{R}^3 . For this test, $N = 7$ curves have been considered, with $S = 100$ samples for each curve. The two components $f_1^{(n)}$ and $f_2^{(n)}$ of the vector-valued functions $\vec{f}^{(n)}$ have been generated in a similar way to the functions $f^{(n)}$ in the first numerical test. For the third test, Figure 6 shows (a) the samples used, and (b) the curves reconstructed by the algorithm (a different color is used for each reconstructed curve). Again, also in this case, each reconstructed curve coincides with one of the original curves.

In the last test, we illustrate a particular situation in which the algorithm adopted in this work has better performance

than the one proposed in [6], considering a simple case with $l = 1$ and only two intersecting curves, and $S = 100$ samples for each curve. We also assume their wrong preliminary identification by the algorithm proposed in [9] (consistently with an analogous case of erroneous identification reported in [8, Figure 2.3]). More precisely, we assume that, for each value of $x = s/(S - 1)$, $s = 0, \dots, S - 1$, that algorithm associates the first curve with the smallest element of F_s , and the second curve with the largest value of F_s . Since the two intersecting curves have very small slope changes, no correction step is performed by the algorithm proposed in [6] (see also Section 1). So, in this particular case, the two algorithms from [6, 9] produce the same output (more extensive numerical comparisons in which the correction step of [6] is actually performed are among the possible subjects of further research, together with real-world applications of

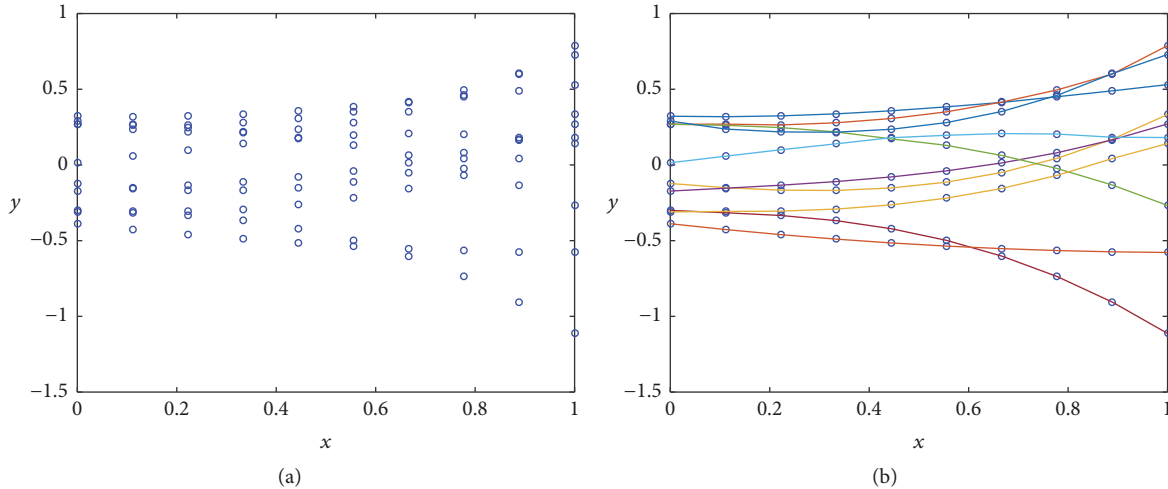


FIGURE 5: (a) Samples used for the second numerical test. (b) Curves reconstructed by the proposed algorithm in the second numerical test, then interpolated linearly.

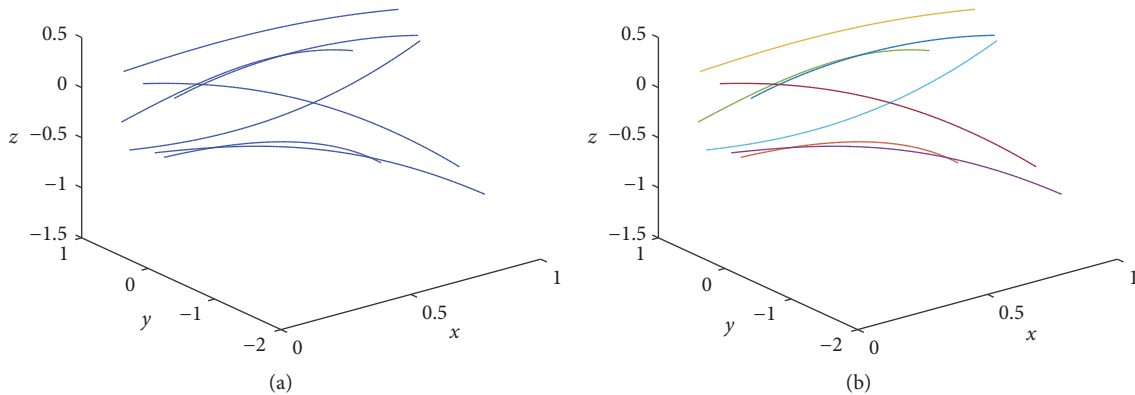


FIGURE 6: (a) Samples used for the third numerical test. (b) Curves reconstructed by the proposed algorithm in the third numerical test.

the proposed algorithm). Figure 7 shows that, in contrast, the proposed algorithm is able to identify correctly the two intersecting curves.

4. Application to the Identification of Smooth Eigenvalue Curves

In the following, we illustrate how the algorithm proposed in this paper can be applied to the identification of smooth eigenvalue curves, considering a last numerical test, whose description is detailed as follows. At first, we generate three real matrices $A, B, C \in \mathbb{R}^{N \times N}$ (whose elements are obtained, for simplicity, as realizations of independent random variables, uniformly distributed on $[0, 1]$). Then, for $x \in [0, 1]$, we construct the real matrix $D(x) = A + Bx + Cx^2$ and the symmetric real matrix $H(x) = D^T(x)D(x)$, where $D^T(x)$ is the transpose of $D(x)$. For each x , $f^{(n)}(x)$ is defined to be one of the (real) eigenvalues of the matrix $H(x)$. The functions $f^{(n)}$ are unknown to the algorithm, which has at its disposal only the sets F_s (i.e., the collections of eigenvalues, for each

$x = s/(S - 1)$, $s = 0, \dots, S - 1$). Nevertheless, the algorithm attributes the eigenvalues to the functions not sequentially according to their increasing values, but in such a way to make the resulting functions smooth. It is worth noting that, since by construction the matrix $H(x)$ is symmetric and depends analytically on the parameter x , its eigenvalues are analytic functions of x , due to [13, Theorem 6.1 in Chapter II]. Interestingly, this holds for both simple and multiple eigenvalues. Finally, the applicability of the algorithm (more precisely, of its Steps 1 and 2) derives from the fact that, with probability 1 with respect to the random construction of the matrix A , the matrix $H(0) = A^T A$ has distinct eigenvalues (which is assumed in the following).

Figure 8 shows (a) the samples obtained according to the construction above (with $N = 9$ curves (the simulation has been actually performed with $N = 10$ curves; however, since by running the simulation several times, one curve is typically well-separated from the other ones, in order to improve the visualization, Figure 8 refers only to the other 9 curves) and $S = 20$ sets of samples F_s), and (b) the curves reconstructed by the proposed algorithm (a different color

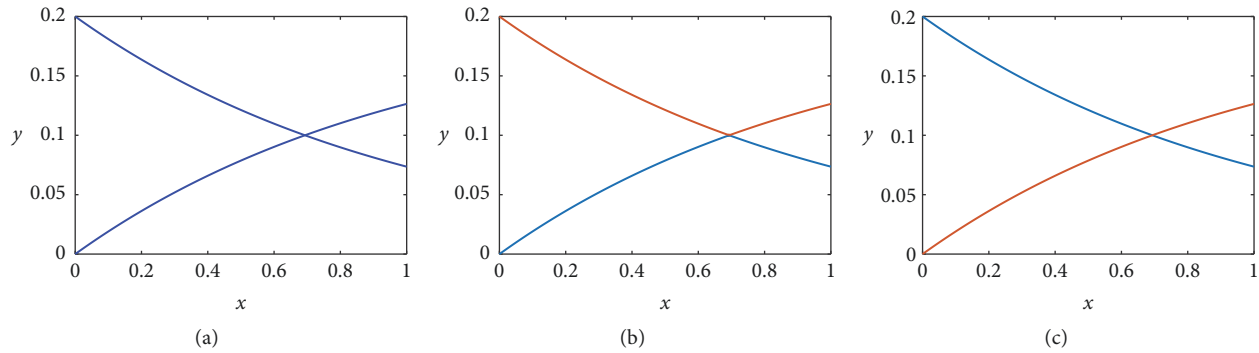


FIGURE 7: (a) Samples used for the fourth numerical test. (b) Curves reconstructed by the algorithm proposed in [6] in the fourth numerical test, assuming a sufficiently large threshold for slope changes. (c) Curves reconstructed by the proposed algorithm in the fourth numerical test.

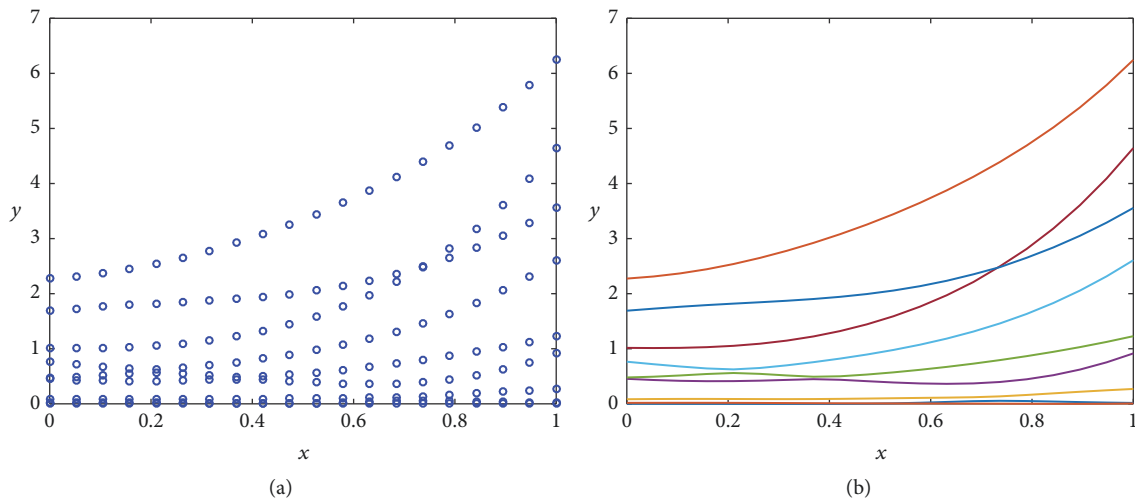


FIGURE 8: (a) Samples used for the identification of smooth eigenvalue curves. (b) Curves reconstructed by the proposed algorithm.

is used for each reconstructed curve). The figure shows that, indeed, the algorithm is able to identify quite smooth curves.

5. Conclusions

An algorithm has been proposed for the identification of possibly intersecting curves. Despite its simplicity, the algorithm has been demonstrated to be very effective and applicable to problems such as the automatic identification of eigenvalue curves. Although artificially generated eigenvalue curves have been considered in the numerical example presented in the paper, the algorithm could be applied also to dispersion curves arising in wave-propagation problems, since these are also eigenvalue (or, more generally, generalized eigenvalue) curves [1]. As remarked in the Introduction, the algorithm is also potentially applicable to the analysis of motion-capture data [4]. Other possible applications are in medical image analysis [14] and in handwritten character recognition [15].

The numerical results presented in the paper refer to curves embedded in the plane and in the space. However, similar results can be obtained for curves embedded in higher-dimensional spaces. As a possible future research

direction, the algorithm could be extended to the case of possibly intersecting surfaces, which would be potentially applicable to the identification of dispersion surfaces (instead of curves), still in wave-propagation problems [16]. Another research direction concerns the formulation of a suitable optimization problem modeling curve identification, for which the proposed algorithm (or its suitable variation) would provide a good approximation of the optimal solution value. Finally, in the paper, the proposed algorithm has been applied to noiseless data. This is justified by the fact that, in several applications, the data are not corrupted by noise, but still one has only a few such data (because their generation could be computationally expensive, as in the case of data coming from dispersion curves associated with complex physical-mathematical models). However, the algorithm could be also applied (possibly with some variations, such as the use of regularization techniques typical of machine learning [17–21]) in the case of noisy data and compared, on such problems, with existing multitarget tracking algorithms [7, 22]. In this particular case, the final interpolation step, implemented in the second test presented in Section 3, could be replaced by a more appropriate approximation step.

Data Availability

The MATLAB code and the data used for the numerical tests presented in the paper are included within the Supplementary Materials.

Disclosure

The author is a member of GNAMPA-INdAM (Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni - Istituto Nazionale di Alta Matematica).

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

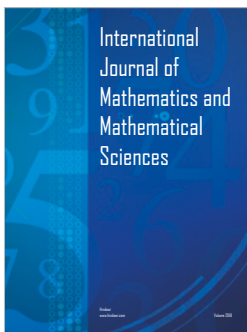
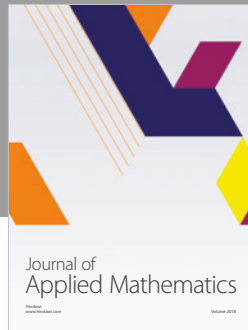
The author was partially supported by a FFABR (Fondo per il Finanziamento delle Attività Base di Ricerca) research fund from the Italian Ministry of Education, University and Research (MIUR).

Supplementary Materials

test1.m: MATLAB code for the generation of Figures 4 a) and b); it uses workspace1.mat. test2.m: MATLAB code for the generation of Figures 5 a) and b); it uses workspace2.mat. test3.m: MATLAB code for the generation of Figures 6 a) and b); it uses workspace3.mat. test4.m: MATLAB code for the generation of Figures 7 a), b), and c); it uses workspace4.mat. test_eigenvalues.m: MATLAB code for the generation of Figures 8 a) and b); it uses workspace_eigenvalues.mat. (Supplementary Materials)

References

- [1] A. Bacigalupo, M. Lepidi, G. Gnecco, and L. Gambarotta, "Optimal design of auxetic hexachiral metamaterials with local resonators," *Smart Materials and Structures*, vol. 25, no. 5, 2016.
- [2] H. Men, R. M. Freund, N. C. Nguyen, J. Saa-Seoane, and J. Peraire, "Fabrication-adaptive optimization with an application to photonic crystal design," *Operations Research*, vol. 62, no. 2, pp. 418–434, 2014.
- [3] D. Kressner, *Numerical methods for general and structured eigenvalue problems*, vol. 46, Springer-Verlag, Berlin, 2005.
- [4] K. Kolykhalova, G. Gnecco, M. Sanguineti, A. Camurri, and G. Volpe, "Graph-restricted game approach for investigating human movement qualities," in *Proceedings of the 4th International Conference on Movement Computing, MOCO 2017*, UK, June 2017.
- [5] L. D. Akulenko and S. V. Nesterov, *High-precision methods in eigenvalue problems and their applications*, vol. 6 of *Differential and Integral Equations and Their Applications*, Chapman & Hall/CRC, Boca Raton, FL, 2005.
- [6] K. Raghupathy and T. W. Parks, "Improved curve tracing in images," in *Proceedings of the Proceedings - IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. III581–III584, Canada, May 2004.
- [7] D. B. Reid, "An algorithm for tracking multiple target," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [8] K. Raghupathy, *Curve tracing and curve detection in images, Thesis presented to the Faculty of the Graduate School of Cornell University*, 2004, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.571.3262rep=rep1type=pdf>.
- [9] G. Steger, "An unbiased detector of curvilinear structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 2, pp. 113–125, 1998.
- [10] P. M. Vaidya, "Geometry helps in matching," *SIAM Journal on Computing*, vol. 18, no. 6, pp. 1201–1225, 1989.
- [11] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, 1998.
- [12] *The Mathworks, Inc., Natick, Massachusetts, MATLAB 9.2 (R2017a)*, 2017.
- [13] T. Katö, *Perturbation Theory for Linear Operators*, vol. 132 of *Classics in Mathematics*, Springer, 1995.
- [14] M. Trzuppek, M. R. Ogiela, and R. Tadeusiewicz, "Intelligent image content semantic description for cardiac 3D visualisations," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 8, pp. 1410–1418, 2011.
- [15] F. Yao, G. Shao, and J. Yi, "Extracting the trajectory of writing brush in Chinese character calligraphy," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 6, pp. 631–644, 2004.
- [16] A. Bacigalupo, G. Gnecco, M. Lepidi, and L. Gambarotta, "Optimal design of low-frequency band gaps in anti-tetrachiral lattice meta-materials," *Composites Part B: Engineering*, vol. 115, pp. 341–359, 2017.
- [17] G. Gnecco and M. Sanguineti, "The weight-decay technique in learning from data: an optimization point of view," *Computational Management Science*, vol. 6, no. 1, pp. 53–79, 2009.
- [18] G. Gnecco and M. Sanguineti, "Regularization techniques and suboptimal solutions to optimization problems in learning from data," *Neural Computation*, vol. 22, no. 3, pp. 793–829, 2010.
- [19] G. Gnecco and M. Sanguineti, "On spectral windows in supervised learning from data," *Information Processing Letters*, vol. 110, no. 23, pp. 1031–1036, 2010.
- [20] G. Gnecco, M. Gori, S. Melacci, and M. Sanguineti, "Learning with mixed hard/soft pointwise constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2019–2032, 2015.
- [21] G. Gnecco, "Symmetry and antisymmetry properties of optimal solutions to regression problems," *Optimization Letters*, vol. 11, no. 7, pp. 1427–1442, 2017.
- [22] I. J. Cox and S. L. Hingorani, "Efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.



Hindawi

Submit your manuscripts at
www.hindawi.com

