



## ISTITUTO NAZIONALE DI RICERCA METROLOGICA Repository Istituzionale

A simple algorithm to find the L-curve corner in the regularisation of ill-posed inverse problems

*Original*

A simple algorithm to find the L-curve corner in the regularisation of ill-posed inverse problems / Cultrera, Alessandro; Callegaro, Luca. - In: IOP SCINOTES. - ISSN 2633-1357. - 1:2(2020), p. 025004. [10.1088/2633-1357/abad0d]

*Availability:*

This version is available at: 11696/64814 since: 2020-12-30T09:14:04Z

*Publisher:*

IOP

*Published*

DOI:10.1088/2633-1357/abad0d

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

ARTICLE • OPEN ACCESS

## A simple algorithm to find the L-curve corner in the regularisation of ill-posed inverse problems

To cite this article: Alessandro Cultrera and Luca Callegaro 2020 *IOPSciNotes* 1 025004

View the [article online](#) for updates and enhancements.

### Recent citations

- [Mapping Time-Dependent Conductivity of Metallic Nanowire Networks by Electrical Resistance Tomography toward Transparent Conductive Materials](#)  
Gianluca Milano *et al*



## ARTICLE

# A simple algorithm to find the L-curve corner in the regularisation of ill-posed inverse problems

## OPEN ACCESS

RECEIVED  
8 May 2020REVISED  
22 July 2020ACCEPTED FOR PUBLICATION  
6 August 2020PUBLISHED  
28 August 2020

Alessandro Cultrera and Luca Callegaro

INRIM—Istituto Nazionale di Ricerca Metrologica, Strada delle Cacce, 91—10135, Torino, Italy

E-mail: [a.cultrera@inrim.it](mailto:a.cultrera@inrim.it)**Keywords:** inverse problems, L-curve, electrical resistance tomography, regularisation parameter

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

**Abstract**

We propose a simple algorithm to locate the ‘corner’ of an L-curve, a function often used to select the regularisation parameter for the solution of ill-posed inverse problems. The algorithm involves the Menger curvature of a circumcircle and the golden section search method. It efficiently finds the regularisation parameter value corresponding to the maximum positive curvature region of the L-curve. The algorithm is applied to some commonly available test problems and compared to the typical way of locating the L-curve corner by means of its analytical curvature. The application of the algorithm to the data processing of an electrical resistance tomography experiment on thin conductive films is also reported.

**1. Introduction**

The solution  $\hat{\mathbf{x}}$  of an ill-posed inverse problem is often searched by means of a regularized least squares functional of the type

$$\hat{\mathbf{x}}_\lambda = \arg \min_{\mathbf{x}} \{ \|\mathbf{Ax} - \mathbf{b}\|^2 + \lambda \mathbf{R}(\mathbf{x}) \}, \quad \lambda \in \mathbb{R}, \quad \lambda \geq 0 \quad (1)$$

where  $\mathbf{Ax} - \mathbf{b}$  is the vector of residuals between the experimental data vector  $\mathbf{b}$  and the reconstructed data  $\mathbf{Ax}$  for a given  $\mathbf{x}$ . The regularisation term  $\mathbf{R}(\mathbf{x})$  renders the problem less sensitive to the noise of  $\mathbf{b}$  and find a stable solution.  $\mathbf{R}(\mathbf{x})$  represents a cost function, which usually includes prior information about the solution. The scalar factor  $\lambda$  is the *regularisation parameter*, is a weighing factor of  $\mathbf{R}(\mathbf{x})$ . The choice of  $\lambda$  is crucial for a meaningful solution. As an example, we consider the regularisation method of Tikhonov [1], in which  $\mathbf{R}(\mathbf{x}) = \|\mathbf{x}\|^2$ . Several methods (see [2, section 7]) have been developed in order to find an optimal tuning of  $\lambda$  for a given problem. Of particular interest is the L-curve method [2, section 7.5] [3], which is one of the best-known heuristic methods for the selection of  $\lambda$ . The L-curve is two-dimensional, parametric in  $\lambda$ , defined by points with cartesian coordinates

$$P(\lambda) = (\xi(\lambda), \eta(\lambda)) \rightarrow \begin{cases} \xi(\lambda) = \log \|\mathbf{Ax} - \mathbf{b}\|^2 \\ \eta(\lambda) = \log \|\mathbf{x}\|^2 \end{cases} \quad (2)$$

The point of maximum positive curvature  $P(\lambda_{\text{opt}})$ , the ‘corner’, can be associated to the optimal reconstruction parameter, say  $\lambda_{\text{opt}}$ . The underlying concept is that the ‘corner’ represents a compromise between the fitting to the data and the amount of regularisation applied to the problem [4]. Numerical search algorithms have been proposed for the estimation of  $\lambda_{\text{opt}}$ ; among them, we mention the splines method [3, 5], the triangle method [6] and the L-ribbon method [7]. The adoption of the L-curve approach to deal with diverse ill-posed inverse problems is an ongoing research topic [8, 9]. Here we propose an alternative method, and its very simple implementation, to locate the L-curve corner. It is based on an iterative estimation of the local curvature of the L-curve from three sampled points with an update rule based on the golden section search. The method has a small computational effort since it reduces the number points of the L-curve explicitly computed. The following gives a description of the algorithm and its application on both typical test problems, and a reconstruction problem of electrical resistance tomography.

## 2. Algorithm

The algorithm 1 is written in pseudo-code. Algorithm 1 calls two functions. Function  $P=1\_curve\_P(\lambda)$  is based on the the specific regularisation problem being solved; it is assumed that at each call, given as input the regularisation parameter  $\lambda$  it solves the system (1) and provides as output the point  $P(\lambda)$ , i.e. the coordinates  $\xi(\lambda)$  and  $\eta(\lambda)$  of the L-curve. The function  $C_k = menger(P_j, P_k, P_\ell)$  is defined below in section 2.1. The algorithm is iterative and identifies the estimate  $\lambda_{opt}$  in the following  $\lambda_{MC}$ , by means of the definition of curvature given in section 2.1 and the golden section search method, described in section 2.2. ‘MC’ stays for ‘Menger Curvature’.

### 2.1. Curvature

The function  $C_k = menger(P_j, P_k, P_\ell)$  is based on the definition of the curvature of a circle by three points given by Menger [10, 11]. In our case three values  $\lambda_j < \lambda_k < \lambda_\ell$  of the regularisation parameter identify three points  $P(\lambda_j)$ ,  $P(\lambda_k)$  and  $P(\lambda_\ell)$  on the L-curve. We follow the notation of (2) for the coordinates of a generic point  $P(\lambda)$ . For notational simplicity we make the substitution:

$$\begin{aligned}\xi(\lambda_i) &\rightarrow \xi_i, \\ \eta(\lambda_i) &\rightarrow \eta_i, \\ P(\lambda_i) &\rightarrow P_i.\end{aligned}\tag{3}$$

We define a signed curvature  $C_k$  of the circuncircle as

$$C_k = \frac{2 \cdot (\xi_j \eta_k + \xi_k \eta_\ell + \xi_\ell \eta_j - \xi_j \eta_\ell - \xi_k \eta_j - \xi_\ell \eta_k)}{(\overline{P_j P_k} \cdot \overline{P_k P_\ell} \cdot \overline{P_\ell P_j})^{1/2}},\tag{4}$$

where

$$\begin{aligned}\overline{P_j P_k} &= (\xi_k - \xi_j)^2 + (\eta_k - \eta_j)^2, \\ \overline{P_k P_\ell} &= (\xi_\ell - \xi_k)^2 + (\eta_\ell - \eta_k)^2, \\ \overline{P_\ell P_j} &= (\xi_j - \xi_\ell)^2 + (\eta_j - \eta_\ell)^2,\end{aligned}\tag{5}$$

are the euclidean distances between the sampled L-curve points. Note that we choose to index the curvature with the intermediate index ( $k$ ) of the three points.

### 2.2. Golden section search

The algorithm is initialized by assigning the search interval  $[\lambda_1, \lambda_4]$ . Two other values  $\lambda_2$  and  $\lambda_3$  are calculated following the golden section method; the calculation is done on the exponents of  $\lambda$  (given  $\lambda_i = 10^{x_i}$ ) to maintain a uniform spacing along the many orders of magnitude covered,

$$\begin{aligned}x_2 &= (x_4 + \varphi \cdot x_1) / (1 + \varphi), \\ x_3 &= x_1 + (x_4 - x_2),\end{aligned}\tag{6}$$

where  $\varphi = (1 + \sqrt{5})/2$  is the golden section [12]. Four values of  $\lambda$  define four points on the L-curve and allow to calculate two curvatures,  $C_2$  from  $\{P(\lambda_1), P(\lambda_2), P(\lambda_3)\}$  and  $C_3$  from  $\{P(\lambda_2), P(\lambda_3), P(\lambda_4)\}$ . The curvatures  $C_2$  and  $C_3$  are compared; consistent reassignment and recalculation are done in order to work at each iteration with four points  $P(\lambda_1) \dots P(\lambda_4)$ . The algorithm terminates when the search interval  $[\lambda_1, \lambda_4]$  is smaller than a specified threshold  $\epsilon$  and returns  $\lambda_{MC}$ .

It may happen that the curvature  $C_3$  associated to the right-hand circle is negative at the initial stage of the search, since  $C_k$  is defined with sign in (4). By definition, the corner corresponds to a positive curvature and it lays on the left-side of the plot. Hence, the algorithm performs a check, and while  $C_3 < 0$  the search extreme  $\lambda_1$  is kept fixed,  $\lambda_4$  is shifted toward smaller values and  $\lambda_2$  and  $\lambda_3$  are recalculated. The condition on  $C_3$  is strong enough that even in case of both negative curvatures it guarantees the convergence towards the corner.

Some considerations: (a) according to the golden section search method, the algorithm needs to recalculate only one  $P(\lambda)$  at each iteration (except for the first iteration), the other can be simply reassigned; this limits the calculation effort; (b) as  $P(\lambda_1)$  and  $P(\lambda_4)$  are distant at the first iterations,  $C_2$  and  $C_3$  are just rough approximations of the curvature of the L-curve in different regions, but become more accurate as the distance between the search extremes decreases.

#### Algorithm 1. L-curve corner search

- 
- 1: Initialize  $\lambda_1$  and  $\lambda_4$ ; { search extremes }
  - 2: Assign  $\epsilon$ ; { termination threshold }
  - 3:  $\varphi \leftarrow (1 + \sqrt{5})/2$ ; { golden section }
  - 4:  $\lambda_2 \leftarrow 10^{(x_4 + \varphi \cdot x_1) / (1 + \varphi)}$ ;
-

```

(Continued.)
5:  $\lambda_3 \leftarrow 10^{x_1+(x_4-x_2)}$ ;
6: for  $i = 1$  to 4 do
7:  $P_i \leftarrow \text{l\_curve\_P}(\lambda_i)$ ; {l_curve_Preturns (2)}
8: end for
9: repeat
10:  $C_2 \leftarrow \text{menger}(P_1, P_2, P_3)$ ; {mengercalls (4)}
11:  $C_3 \leftarrow \text{menger}(P_2, P_3, P_4)$ ;
12: repeat
13:  $\lambda_4 \leftarrow \lambda_3$ ;  $P_4 \leftarrow P_3$ ;
14:  $\lambda_3 \leftarrow \lambda_2$ ;  $P_3 \leftarrow P_2$ ;
15:  $\lambda_2 \leftarrow 10^{(x_4+\varphi x_1)/(1+\varphi)}$ ;
16:  $P_2 \leftarrow \text{l\_curve\_P}(\lambda_2)$ ;
17:  $C_3 \leftarrow \text{menger}(P_2, P_3, P_4)$ ;
18: until  $C_3 > 0$ 
19: if  $C_2 > C_3$  then
20:  $\lambda \leftarrow \lambda_2$ ; {store  $\lambda$ }
21:  $\lambda_4 \leftarrow \lambda_3$ ;  $P_4 \leftarrow P_3$ ;
22:  $\lambda_3 \leftarrow \lambda_2$ ;  $P_3 \leftarrow P_2$ ;
23:  $\lambda_2 \leftarrow 10^{(x_4+\varphi x_1)/(1+\varphi)}$ ;
24:  $P_2 \leftarrow \text{l\_curve\_P}(\lambda_2)$ ; {only  $P_2$  is recalculated}
25: else
26:  $\lambda \leftarrow \lambda_3$ 
27:  $\lambda_1 \leftarrow \lambda_2$ ;  $P_1 \leftarrow P_2$ ;
28:  $\lambda_2 \leftarrow \lambda_3$ ;  $P_2 \leftarrow P_3$ ;
29:  $\lambda_3 \leftarrow 10^{x_1+(x_4-x_2)}$ ;
30:  $P_3 \leftarrow \text{l\_curve\_P}(\lambda_3)$ ; {only  $P_3$  is recalculated}
31: end if
32: until  $(\lambda_4 - \lambda_1)/\lambda_4 < \epsilon$ 
33: return  $\lambda_{MC} \leftarrow \lambda$ 

```

### 3. Application to test problems

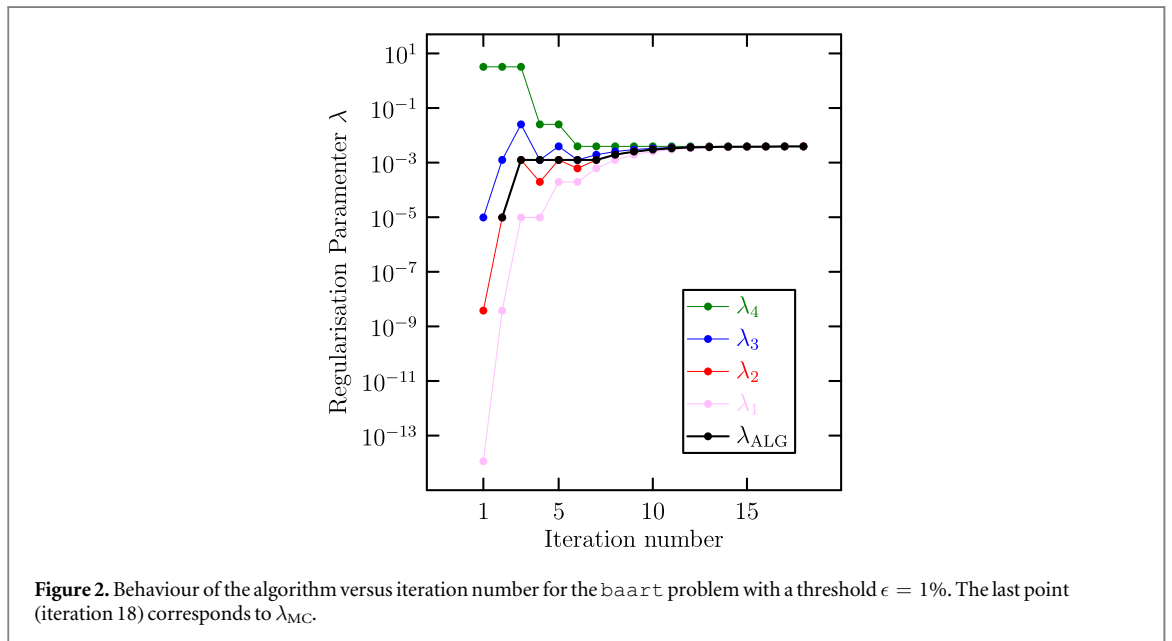
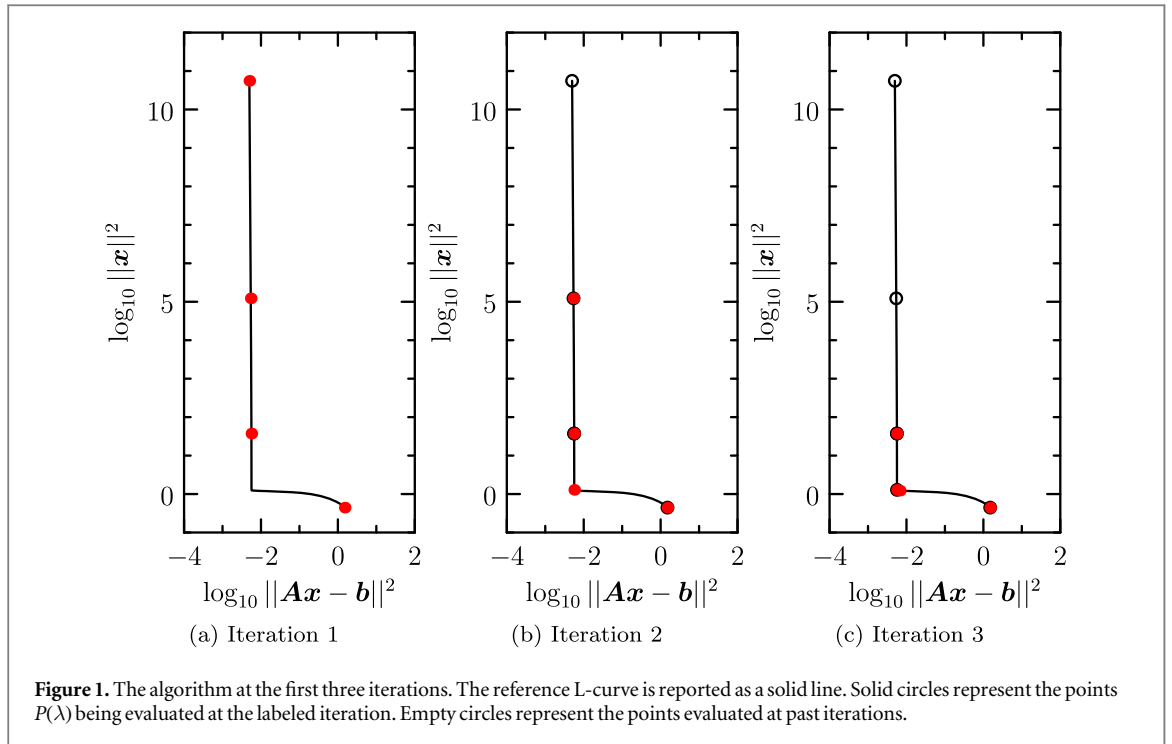
We tested the algorithm on small demonstrative problems, some (baart, blur, shaw and spike) chosen from the function library Regularisation Tools (RT), implemented in MATLAB [13]. This library is also employed to implement a function of the algorithm ( $\text{l\_curve\_P}(\lambda)$ ) which evaluates a single point of the L-curve for a given  $\lambda$ . Algorithm 1 is implemented in MATLAB as well.

The application of algorithm 1 to the problem baart is shown explicitly in the following. This problem represents the discretization of a Fredholm integral equation of first kind of order  $n$ . The matrix  $A$  in (1) is therefore  $n \times n$ . The chosen size of the problem is  $n = 32$ . In this example we added random noise of relative standard deviation of  $10^{-3}$  to the exact data. The corner of the L-curve generated by this problem is located with both algorithm 1 and the  $\text{l\_corner}$  routine from RT<sup>1</sup>. Figure 1 shows the first three iterations of the algorithm, and displays also a full L-curve obtained by dense sampling of  $\text{l\_curve\_P}(\lambda)$  as a reference. Empty circles represent points visited at previous iterations, while filled circles represent the four points  $P_1 \dots P_4$  of the given iteration. The algorithm runs by choosing as initial search extremes the default choice of the  $\text{l\_corner}$  routine ( $\lambda_1 = 10^{-14}$  and  $\lambda_4 = 10^{-1}$ ). Running the algorithm on the other three mentioned problems gives similar results in term of accuracy compared to the native  $\text{l\_corner}$  routine of the RT library. The optimal regularisation parameter obtained with this routine is called  $\lambda_{RT}$  in the following. Table 1 summarizes the results of solving the four test problems with algorithm 1 and with RT's function  $\text{l\_corner}$ .  $\lambda_{MC}$  is the optimal regularisation parameter returned by our algorithm while  $\lambda_{RT}$  is the one returned by the RT routine. The MATLAB profiler was used to get the corresponding net timing<sup>2</sup> of algorithm 1 ( $t_{MC}$ ), and the  $\text{l\_corner}$  routine ( $t_{RT}$ ). figure 2 shows the evolution of the algorithm towards convergence.

As a side note, a similar implementation of the presented algorithm could be made also using a Fibonacci search to pick the  $x_i$  in (6). In fact the Fibonacci search interval reduction ratio converges to the golden section very quickly [12].

<sup>1</sup> We tested our algorithm with noise relative standard deviation levels over a wide range, from  $10^{-10}$  to  $10^{-1}$ . Our results always matched with negligible deviation the algorithm of Regularisation Tools, taken as reference.

<sup>2</sup> Eventual plotting time not considered.

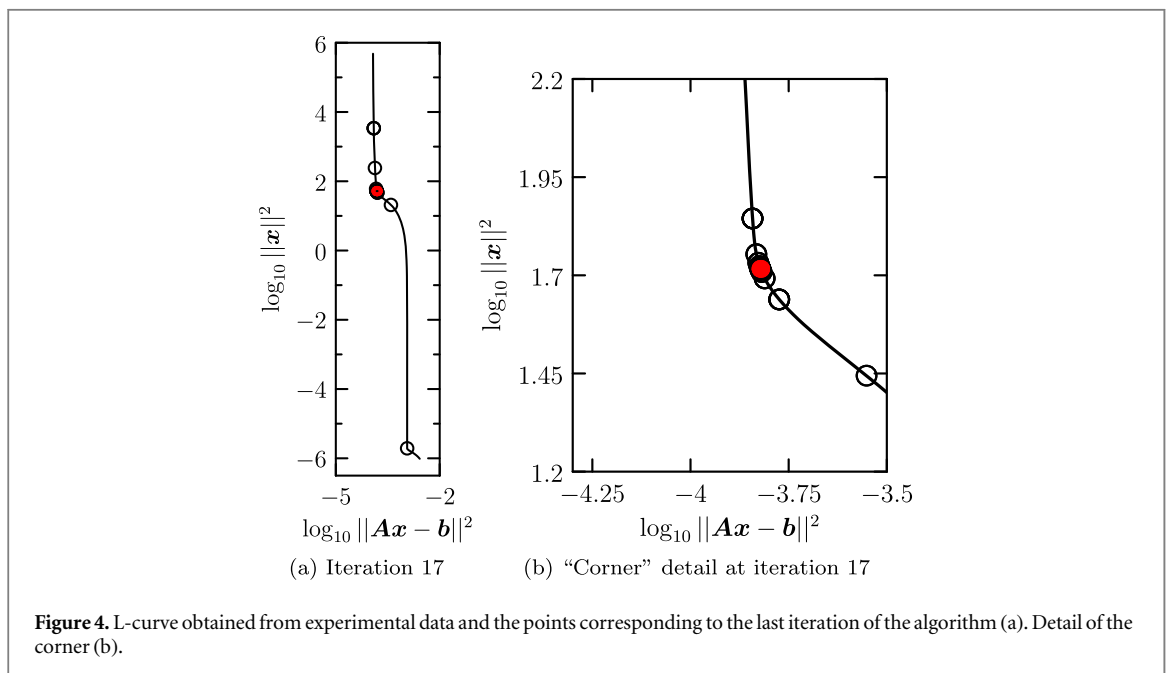
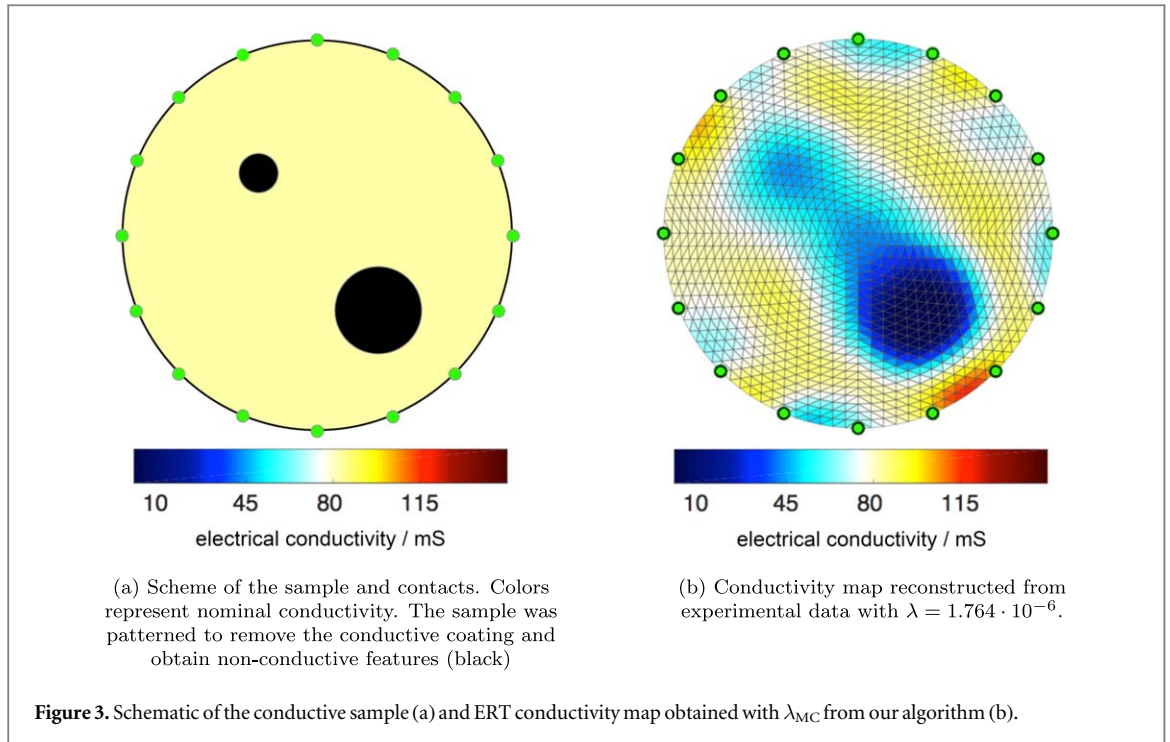


**Table 1.** Comparison between algorithm 1 and the analytic curvature approach on test problems.

Problem	$\lambda_{MC}$	$\lambda_{RT}$	$t_{MC}$ (ms)	$t_{RT}$ (ms)	iterations
baart (32)	$3.92 \times 10^{-3}$	$4.02 \times 10^{-3}$	73	469	18
blur (16, 4, 5)	$3.18 \times 10^{-4}$	$3.20 \times 10^{-4}$	82	459	15
shaw (32)	$8.65 \times 10^{-4}$	$8.28 \times 10^{-4}$	69	473	17
spike (32, 5)	$1.65 \times 10^{-4}$	$1.60 \times 10^{-4}$	61	461	17

#### 4. Application to electrical resistance tomography

The following shows the application of algorithm 1 to electrical resistance tomography (ERT) [14]. In this experiment we used a patterned tin-oxide conductive sample of circular geometry, with electrical contacts on its



boundary (see figure 3(a)). Four-terminal resistance measurements are performed with a scanning setup; the measurements are the elements of the data vector  $\mathbf{b}$ . A detailed description of the experiment is given in [15, 16]. The ERT problem solution is obtained by solving a discretized Laplace equation with Tikhonov regularisation, a formulation compatible with the calculation of a continuous L-curve. EIDORS [17] routines are used to generate a two-dimensional circular mesh (2304 elements) with 16 contact points at the boundary (corresponding to a  $\mathbf{b}$  of size 208), to discretize the Laplace equation and obtain matrix  $\mathbf{A}$ . The reconstructed image shown in figure 3(b). Figure 4 reports the main results of the application of algorithm 1 to ERT experimental data. Figure 4(a) shows the L-curve and the last iteration of algorithm 1; figure 4(b) the detail of the corner. The optimal regularisation parameter returned by the algorithm with  $\epsilon = 1\%$  is  $\lambda_{MC} = 1.8 \cdot 10^{-6}$ . The relative difference between  $\lambda_{MC}$  and  $\lambda_{RT}$  is negligible (1 part in  $10^{13}$ ).

## 5. Conclusions

The proposed algorithm allows, given an inverse problem having the form (1), the determination of the regularisation parameter  $\lambda_{MC}$  corresponding to the maximum positive curvature of the L-curve. The algorithm is designed for maximum simplicity of implementation on already existing solvers. On both test problems or in a real electrical resistance tomography problem, convergence is achieved in less than 20 iterations. Compared to a common routine for the location of the L-curve corner such as `Regularisation Tools` the present algorithm returns strongly compatible results with a reduced calculation effort.

## Acknowledgments

The work has been developed within the Joint Research Project 16NRM01 GRACE: Developing electrical characterisation methods for future graphene electronics. This project has received funding from the EMPIR programme co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation programme.

## Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

## ORCID iDs

Alessandro Cultrera  <https://orcid.org/0000-0001-8965-9116>

Luca Callegaro  <https://orcid.org/0000-0001-5997-9960>

## References

- [1] Tikhonov A N, Goncharsky A, Stepanov V and Yagola A G 2013 *Numerical Methods for the Solution of Ill-Posed Problems* (New York, US: Springer Science & Business Media)
- [2] Hansen P C 1998 *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion* (Philadelphia, US: Society for Industrial and Applied Mathematics)
- [3] Hansen P C 1992 Analysis of discrete ill-posed problems by means of the L-curve *SIAM Rev.* **34** 561
- [4] Hansen P C and O'Leary D P 1993 The use of the L-curve in the regularization of discrete ill-posed problems *SIAM J. Sci. Comput.* **14** 1487
- [5] Hansen P C, Jensen T K and Rodriguez G 2007 An adaptive pruning algorithm for the discrete L-curve criterion *J. Comput. Appl. Math.* **198** 483
- [6] Castellanos J L, Gómez S and Guerra V 2002 The triangle method for finding the corner of the L-curve *Appl. Numer. Math.* **43** 359
- [7] Calvetti D, Golub G H and Reichel L 1999 Estimation of the L-curve via lanczos bidiagonalization *BIT Numer. Math.* **39** 603
- [8] Choi M-B, Shin J, Ji H-I, Kim H, Son J-W, Lee J-H, Kim B-K, Lee H-W and Yoon K J 2019 Interpretation of impedance spectra of solid oxide fuel cells: L-curve criterion for determination of regularization parameter in distribution function of relaxation times technique *JOM* **71** 3825
- [9] Xu Y, Pei Y and Dong F 2016 An extended L-curve method for choosing a regularization parameter in electrical resistance tomography *Meas. Sci. Technol.* **27** 114002
- [10] Menger K 1930 Untersuchungen über Allgemeine Metrik *Math. Ann.* **103** 466
- [11] Pajot H 2002 *Analytic Capacity, Rectifiability, Menger Curvature and Cauchy Integral* (Berlin, DE: Springer Science & Business Media)
- [12] Kiefer J 1953 Sequential minimax search for a maximum *P. Am. Math. Soc.* **4** 502
- [13] Hansen P C 1994 Regularization tools: a MATLAB package for analysis and solution of discrete ill-posed problems *Numer. Algorithms* **6** 1
- [14] Seo J K and Woo E J 2013 *Nonlinear Inverse Problems in Imaging* (Chichester, UK: John Wiley & Sons, Ltd) Chap. Electrical Impedance Tomography
- [15] Cultrera A et al 2019 Mapping the conductivity of graphene with Electrical Resistance Tomography *Sci. Rep.* **9** 10655
- [16] Cultrera A and Callegaro L 2016 Electrical resistance tomography of conductive thin films *IEEE Trans. Instrum. Meas.* **65** 2101
- [17] Adler A and Lionheart W R 2005 EIDORS: Towards a community-based extensible software base for EIT *VI Conf. on Biomedical Applications of Electrical Impedance Tomography (London, UK, 22–24 June 2005)*