CIMAT

Centro de Investigación en Matemáticas, A.C.
Optimización

Tarea 11

José Miguel Saavedra Aguilar

Abstract

In this homework we prove some properties useful for non-linear least squares and implement the Levenberg-Marquart algorithm

1 Problem 1

Let

$$f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$$

with $r_j : \mathbb{R}^n \to \mathbb{R}$ for $j = 1, \ldots, m$. Let $R : \mathbb{R}^n \to \mathbb{R}^m$ be defined as $[r_1(x), \ldots, r_m(x)]^\top$ and $J = J(x)$ be the jacobian matrix of f. First, lets consider

$$J(x) = \begin{pmatrix} \nabla r_1(x)^\top \\ \vdots \\ \nabla r_m(x)^\top \end{pmatrix}$$

Then

$$J^\top J = \begin{pmatrix} \nabla r_1(x) & \cdots & \nabla r_m(x) \end{pmatrix} \begin{pmatrix} \nabla r_1(x)^\top \\ \vdots \\ \nabla r_m(x)^\top \end{pmatrix}$$

For entry j, k we have:

$$\left[J^\top J\right]_{j,k} = \sum_{i=1}^{m} \frac{\partial r_i}{\partial x_j}\frac{\partial r_i}{\partial x_k}$$

Note also

$$\left[\sum_{i=1}^{m} \nabla r_i \nabla r_i^\top\right]_{j,k} = \sum_{i=1}^{m} \frac{\partial r_i}{\partial x_j}\frac{\partial r_i}{\partial x_k}$$

So that

$$J^\top J = \sum_{i=1}^{m} \nabla r_i \nabla r_i^\top$$

Now, let's consider

$$J^\top R = \begin{pmatrix} \nabla r_1(x) & \cdots & \nabla r_m(x) \end{pmatrix} R$$

Now, the product of a matrix and a vector can be decomposed as a sum of the columns of the matrix times the vector entries, so we have:

$$J^\top R = \sum_{i=1}^{m} \nabla r_i(x) r_i$$

2 Problem 2

Let $R \in \mathbb{R}^m, J \in \mathbb{R}^{m \times n}, I \in \mathbb{R}n \times n$ the identity matrix and $\mu > 0$. Let $A \in \mathbb{R}^{(m+n)\times n}$ and $b \in \mathbb{R}^{m+n}$ be defined as:

$$A = \begin{bmatrix} J \\ \sqrt{\mu} I \end{bmatrix}, \qquad b = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

consider the problem:

$$s^* \underset{s \in \mathbb{R}^n}{\arg\min} \quad \|As + b\|_2^2$$

In order to solve this problem, we shall take the derivative with respect to s and find the critical points:

$$2A^\top As + +2s^\top A^\top b = 0$$
$$s^* = -(A^\top A)^{-1} A^\top b$$

Now, note:

$$A^\top A = J^\top J + \mu I$$
$$A^\top b = J^\top R + \sqrt{\mu} 0$$
$$A^\top b = J^\top R$$

Then, we substitute this values for s^*:

$$s^* = -(J^\top J + \mu I)^{-1} J^\top R$$

Since $\|As + b\|_2^2$ is a quadratic function for s, we have s^* minimizes $\|As + b\|_2^2$ if and only if $\nabla^2 \|As + b\|_2^2$ is positive definite. Since

$$\nabla^2 \|As + b\|_2^2 = J^\top J + \mu I$$

Since $J^\top J$ is positive semi-definite, and $\mu > 0$, we have $\nabla^2 \|As + b\|_2^2$ is positive definite.

3 Levenberg-Marquart algorithm

We implemented the Levenberg-Marquart algorithm in Julia [1] and fit the points (x_i, y_i) given in file `puntos2D_ej3.npy` by the model:

$$A\sin(\omega x + \phi)$$

so $z = \begin{pmatrix} A & \omega & \phi \end{pmatrix}$, with residuals $r_i(z) = A\sin(\omega x_i + \phi) - y_i$. The jacobian matrix for this residual function is given by:

$$\nabla r_i(z) = \begin{bmatrix} \sin(\omega x_i + \phi) \\ Ax_i\cos(\omega x_i + \phi) \\ A\cos(\omega x_i + \phi) \end{bmatrix}$$

$$J = \begin{pmatrix} \nabla r_1^\top \\ \vdots \\ \nabla r_i^\top \\ \vdots \\ \nabla r_m^\top \end{pmatrix}$$

4 Results

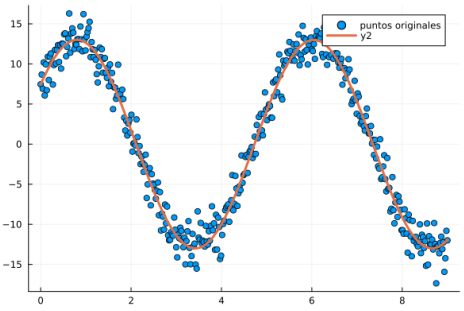
We fit the points (x_i, y_i) starting from the three following starting points:

$$z_0 = \begin{bmatrix} 15 & 0.6 & 0 \end{bmatrix}$$
$$z_0 = \begin{bmatrix} 15 & 1.0 & 0 \end{bmatrix}$$
$$z_0 = \begin{bmatrix} 15 & 0.6 & 1.6 \end{bmatrix}$$

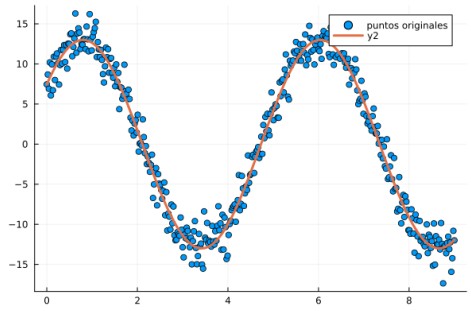
We show the fitment results on table 1 and show the plots fitted function vs the points on image 1

Starting point	μ_{ref}	$f(z)$	A	ω	ϕ	k	$\|p\|_2$
First	0.001	457.169	13.00	1.20	-5.67	9	1.54685×10^{-10}
Second	100	457.169	13.00	1.20	0.61	30	5.48528×10^{-9}
Third	10	457.169	13.00	1.20	0.61	14	5.756×10^{-10}

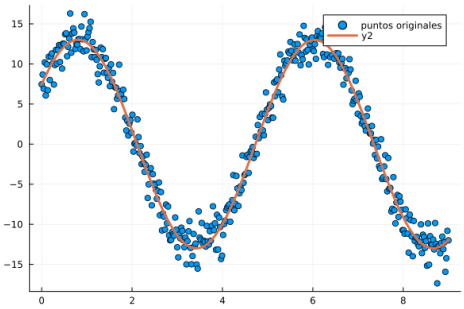
Table 1: Levenberg-Marquart algorithm fitment results



(a) First starting point



(b) Second starting point



(c) Third starting point

Figure 1: Fitted function and the sampled points.

We can conclude this method works well since all functions were fitted correctly, but we must point out this algorithm wouldn't converge from the second and third starting points for some starting values μ_{ref}, so we must be careful.

References

[1] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: https://epubs.siam.org/doi/10.1137/141000671