ISSN: 2307-7743
www.scienceasia.asia

Asian J. Math. Appl. (2022) 2022:12

Exact Camera Location via Gauss Newton Method

Imen Hassairi[1], Chen Junqi[1], Amin Hassairi[2]

[1]College of Sciences and Technology, Wenzhou-Kean University, Wenzhou, China
[2]John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary

Correspondence should be addressed to Imen Hassairi: himen@kean.edu

Email(s): himen@kean.edu (I. Hassairi), chenjunq@kean.edu (C. Junqi), amin-hassairi@hotmail.fr (A. Hassairi)

Abstract
The aim of this present work is to develop an iterative method called Gauss Newton Method to resolve inverse problems. The inverse problem is formulated as an optimization problem in the sense of least squares. In order to minimize the computation time related to resolving this inverse problem, a method with direction of descent (method of Gauss Newton) is chosen. This algorithm allows a good compromise between accuracy and computation time. Directional methods descent that may present numerical instabilities, the Gauss Newton algorithm is stabilized in order to be able to identify the model parameters considered.

Keywords: Gauss Newton method; least squares problem; overdetermined systems.

Introduction

In practice, we are often confronted with the resolution of a system of nonlinear equations. That is, for a given function $f:\mathbb{R}^n\longrightarrow\mathbb{R}^n$, we seek $x\in\mathbb{R}^n$ such that

$$f(x)=0. \tag{1.1}$$

In general, there is no algorithm using a direct method to find a solution of (1.1). We are therefore forced to use iterative methods. Either then

$$f:\Omega\subset\mathbb{R}^n\longrightarrow\mathbb{R}^n, \textit{with } \Omega \textit{ open of } \mathbb{R}^n$$

$$x=\begin{pmatrix} x_1\\ .\\ .\\ x_n\end{pmatrix}\longmapsto\begin{pmatrix} f_1(x)\\ .\\ .\\ f_n(x)\end{pmatrix}$$

We approximate the zero of the function for $x_0\in\mathbb{R}^n$, we establish a Taylor expansion to order 1 of the function f around x_0,

$$f(x)=f(x_0)+Df(x_0)(x-x_0)+o(\|x-x_0\|).$$

Received: 2022-02-11

By repeating this procedure n times, while neglecting the error term, we get

$$\begin{cases} x_0 \in \mathbb{R}^n, \\ x_{n+1} = x_n - (Df(x_n))^{-1} f(x_n) \end{cases}$$

that $(Df(x_n))^{-1}$ is the inverse of the Jacobian matrix of f at the point x_n such that,

$$Df(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & . & . & . & \frac{\partial f_1}{\partial x_n}(x) \\ . & . & . & . & . \\ . & . & \frac{\partial f_j}{\partial x_j}(x) & . & . \\ . & . & . & . & . \\ \frac{\partial f_n}{\partial x_1}(x) & . & . & . & \frac{\partial f_n}{\partial x_n}(x) \end{pmatrix}.$$

The Gauss Newton algorithm is an iterative method regularly used for solving nonlinear least squares problems. It is particularly well-suited to the treatment of very large scale variational data assimilation problems that arise in atmosphere and ocean forecasting. The procedure consists of a sequence of linear least squares approximations to the nonlinear problem, each of which is solved by an iterative process. In comparison with Newton method, the algorithm is attractive because it does not require the evaluation of second-order derivatives in the Hessian of the objective function. In practice the exact Gauss Newton method is too expensive to apply operationally in meteorological forecasting and various approximations are made in order to reduce computational costs and to solve the problems in real time. There is a program in [2] called Photo location and EXIF data viewer, where you can upload whatever the picture and it can give you some information about the brand of the smart phone or tablet that took this picture, the camera characteristics, pixels, date and time. However it is still missing the exact location of the photographer since we have tried to submit different pictures from different countries but never give us the exact place neither the country nor the city. The novelty in our actual paper, we make the link between existing theoretical results made by Gauss Newton and the problem setting up as a real world application. This work gives us a very nice result that helps people, for example, to recognize the exact location of a plane before its explosion or disappearance once we get the last recorded data on its black box.

In the next section, we present general results related to Gauss Newton algorithm and how the method consists in solving a sequence of linearized least squares approximations to the nonlinear problem, each of which can be solved efficiently by an iterative process. We study also the convergence of the Gauss Newton method. We then state the problem in details: both theoretical and algorithmic parts are well studied. Finally, numerical results demonstrating and validating the solution are presented.

Gauss Newton Method

Least squares solutions for overdetermined systems

Let us consider a system of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots.. + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots.. + a_{2n}x_n &= b_2 \\ . \qquad\qquad & \quad . \\ . \qquad\qquad & \quad . \\ . \qquad\qquad & \quad . \end{aligned} \tag{1.2}$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots.. + a_{mn}x_n \quad = \quad b_m$$

where $m \geq n$. In matrix terms, the system (1.2) can be written as, $Ax = b$ with $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$; A is a matrix $m \times n$. Obviously, the system (1.2) has no solution in general. The idea is to look up a vector x such that $\|Ax - b\|_2$ be minimal for the euclidean norm, i.e. we will minimize the sum of the error squares; this is the principle of the method of least squares.

Theorem 1.1. *Let H be an Hilbert space, K a convex set, closed $\neq \emptyset$ and $f \in H$. Then there exists a unique $u \in K$ such that $\|f - u\| = \inf_{v \in K} \|f - v\|$. Moreover, $< f - u, v - u > \leq 0$ for all $v \in K$.*

Corollary 1.2. *Let H be an Hilbert space, K a closed vector subspace of H and $f \in H$. Then there exists a unique $u \in K$ such that $\|f - u\| = \inf_{v \in K} \|f - v\|$. Moreover, $< f - u, v >= 0$ for all $v \in K$.*

Theorem 1.3. *Let A be a matrix $m \times n$(with $m \geq n$) and let $b \in \mathbb{R}^m$. Then*

$$\|Ax - b\|_2 = \min_{\mathbb{R}^n} \|Az - b\|_2 \ if \ and \ only \ if \ A^T Ax = A^T b. \tag{1.3}$$

The equations of the system (1.3) are called normal equations.

Proof Let $G(x) = \|Ax - b\|_2^2 = (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b$.
For $DG(x) = 0$, we get $2(x^T A^T A - b^T A) = 0$, consequently, $A^T Ax = A^T b$.
Geometric interpretation : Let $E = \{Ax, x \in \mathbb{R}^n\}$ a subspace, $\neq \emptyset$, convex, closed of $\mathbb{R}^m$ and $b \in \mathbb{R}^m$. According to the corollary (1.2) we have Ax is the orthogonal projection of b on E. In other words $Ax - b \perp Az$ for all $z \in \mathbb{R}^n$. We deduce that $A^T(Ax - b) = 0$.

Description of Gauss Newton Method

In this part, we are interested in nonlinear overdetermined systems. We consider a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ where $m \geq n$ and we find out a solution of $f(x) = 0$. As we have more constraints than unknowns, we therefore content ourselves with finding a vector $x \in \mathbb{R}^n$ such that $\|f(x)\|_2$ be minimal. If $f(x)$ is a class C^1 function then for $\bar{x}$ to be a local minimal it must $DF(\bar{x}) = 0$, where $F(x) = \|f(x)\|_2^2 = \sum_{i=1}^{m} f_i^2(x)$ and $DF(\bar{x}) = \sum_{i=1}^{m} 2\frac{\partial f_i}{\partial x_j}(\bar{x}) f_i(\bar{x})$.
Writing $DF(\bar{x}) = 0$, we get the following formula

$$(Df(\bar{x}))^T f(\bar{x}) = 0.$$

We linearized $f(x)$ around an approximation x_0 of the solution and we compute x_1 so that the quantity $\|f(x_0) + Df(x_0)(x_1 - x_0)\|_2$ be minimal. A repetition of this process gives the following algorithm:

$$\begin{cases} for \ k = 0, 1, 2... \\ Compute \ f(x_k) \ et \ Df(x_k) \\ determine \ \Delta x_k \\ x_{k+1} = x_k + \Delta x_k \\ end \ for. \end{cases}$$

Remark 1.4. *For a given x_k, Δx_k is determined by minimizing $\|f(x_k) + Df(x_k)h\|_2$ using the least squares method.*

Convergence of the Gauss Newton Method

Theorem 1.5. *Assume that $f : \mathbb{R}^n \to \mathbb{R}^m$ (with $m > n$) be 3 times continuously differentiable, that the rank of $Df(x)$ be maximal and $\tilde{x}$ the vector for which the $\|f(x)\|_2$ is minimal. Then, for x_k close enough of $\tilde{x}$, the error $e_k = x_k - \tilde{x}$ of the Gauss Newton Method satisfies:*

$$e_{k+1} = -((Df(x_k))^T Df(x_k))^{-1} B(x_k)(f(x_k), e_k) + o(\|e_k\|^2),$$

with $B(x) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ is the defined bi-linear application as :

$$(B(x)(u,v))_i = \sum_{l=1}^{n} \sum_{j=1}^{m} \frac{\partial^2 f_j}{\partial x_l \partial x_i}(x) u_j v_l.$$

Proof Let $g(x) = (Df(x))^T f(x)$, we have $g_i(x) = \sum_{i=1}^{n} \frac{\partial f_j}{\partial x_i}(x) f_j(x)$, that gives

$$\frac{\partial g_i}{\partial x_l} = \sum_{j=1}^{m} \frac{\partial^2 f_j}{\partial x_l \partial x_i}(x) f_j(x) + \sum_{j=1}^{m} \frac{\partial f_j}{\partial x_i}(x) \frac{\partial f_j}{\partial x_l}(x).$$

In matrix form, this formula is written

$$g'(x) = B(x)(f(x), .) + (Df(x))^T Df(x).$$

By performing the Taylor series expansion of g at the point x_k, we get

$$\begin{aligned} g(x) &= g(x_k) + g'(x_k)(x - x_k) + o(\|x - x_k\|^2), \\ &= (Df(x_k))^T f(x_k) + (Df(x_k))^T Df(x_k)(x - x_k) + B(x_k)(f(x_k), x - x_k) + o(\|x - x_k\|^2). \end{aligned}$$

Setting $x = \tilde{x}$ and using $(Df(x_k))^T Df(x_k)\Delta x_k + (Df(x_k))^T f(x_k) = 0$, we get

$$\begin{aligned} g(\tilde{x}) &= (Df(x_k))^T f(x_k) + (Df(x_k))^T Df(x_k)(\tilde{x} - x_k) + B(x_k)(f(x_k), \tilde{x} - x_k) - \\ &\quad Df(x_k)^T Df(x_k)\Delta x_k - (Df(x_k))^T f(x_k) + o(\|\tilde{x} - x_k\|^2), \end{aligned}$$

therefore

$$\begin{aligned} o(\|e_k\|^2) &= (Df(x_k))^T Df(x_k)(\tilde{x} - x_k - \Delta x_k) + B(x_k)(f(x_k), -e_k), \\ &= (Df(x_k))^T Df(x_k)(-e_{k+1}) - B(x_k)(f(x_k), e_k), \end{aligned}$$

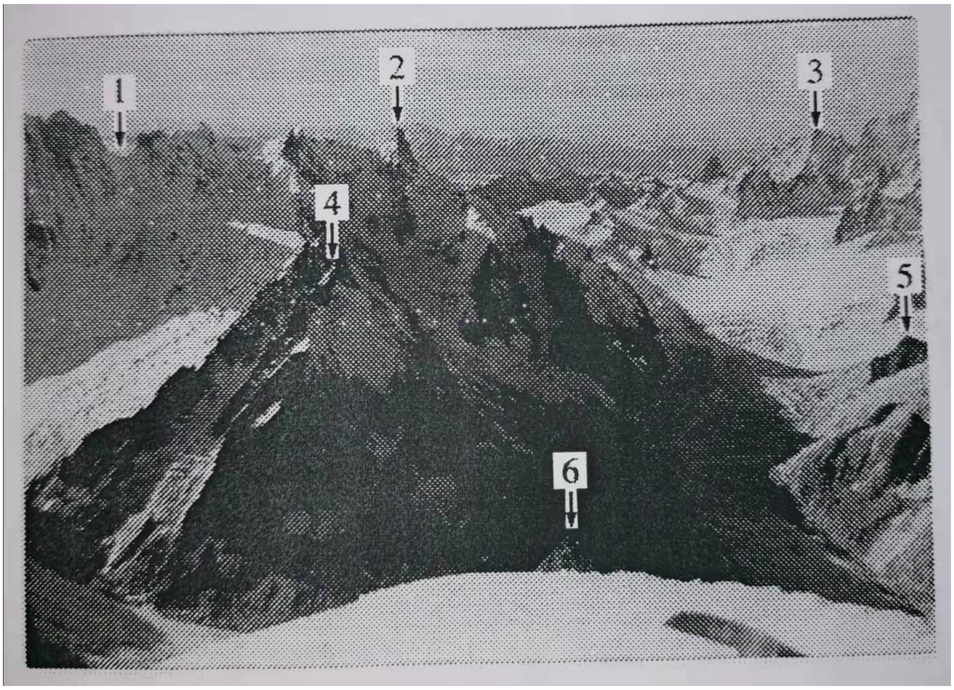
consequently

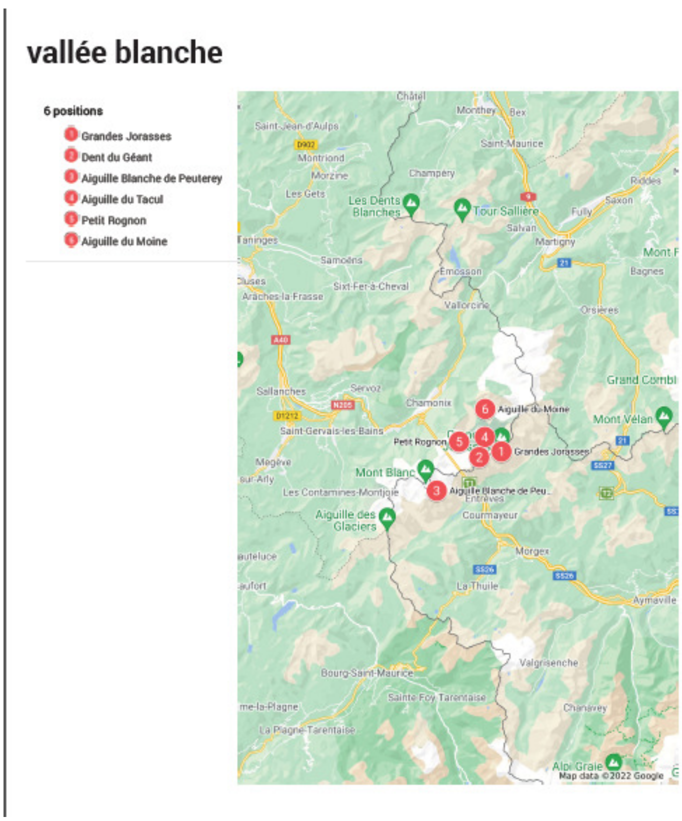
$$e_{k+1} = -((Df(x_k))^T Df(x_k))^{-1} B(x_k)(f(x_k), e_k) + o(\|e_k\|^2).$$

Application

Set up the problem

The figures below shows a photo of the Vallée Blanche on the borders between France and Italy. We can recognize the Grandes Jorasses, the Dent du Géant, the Aiguille Blanche de Peuterey, the Aiguille du Tacul, Petit Rognon and the Aiguille du Moine. And the next figure is a geographic map of this region. The problem consists in finding the position of the camera, its characteristics (focus) and angles of inclination.





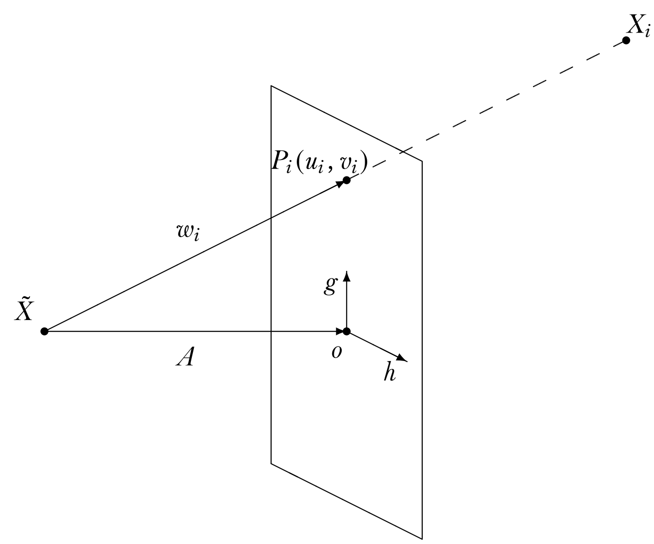
Modelisation

To formulate this problem mathematically, let $\mathcal{R}$ an orthonormal coordinate system of $\mathbb{R}^3$. We have set coordinates (u_i, v_i) on the photo (for which we choose an orthonormal plane $(O, \vec{h}, \vec{g})$ for a suitable choice of h and g), and coordinates $X_i = \begin{pmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{pmatrix}_{\mathcal{R}}$ on the map (x_{3i} represents the altitude). The measured values for the 6 recognized points are given in the following table.

Table 1: The data of the problem "Vallée Blanche"

i	u_i	v_i	x_{1i}	x_{2i}	x_{3i}
1.Grandes Jorasses	$-0,0480$	$0,0290$	9855	5680	3825
2.Dent du Géant	$-0,0100$	$0,0305$	8170	5020	4013
3.Aiguille Blanche de Peuterey	$0,0490$	$0,0285$	2885	730	4107
4.Aiguille du Tacul	$-0,0190$	$0,0115$	8900	7530	3444
5.Petit Rognon	$0,0600$	$-0,0005$	5700	7025	3008
6.Aiguille du Moine	$0,0125$	$-0,0270$	8980	11120	3412

To fix the unknowns, let us denote by $\tilde{X} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{pmatrix}_{\mathcal{R}}$ the position of the camera focus, and by $A = \begin{pmatrix} a \\ b \\ c \end{pmatrix}_{\mathcal{R}}$ the vector of the view direction corresponding to the distance from the projection plane.



Central Projection

Theoretical Part

Consider the orthonormal basis formed by vectors A, h and g with

$$A = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, h = \frac{1}{\sqrt{a^2+b^2}} \begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}, \text{ and } g = \frac{1}{\sqrt{(a^2+b^2)(a^2+b^2+c^2)}} \begin{pmatrix} -ac \\ -bc \\ a^2+b^2 \end{pmatrix}.$$

The plan of the film is generated by h and g, (h being horizontal and g vertical) which gives that the coordinates $(O, \vec{h}, \vec{g})$ is orthonormal so we obtain a graduation of the photo. Let $w_i = \begin{pmatrix} w_{1i} \\ w_{2i} \\ w_{3i} \end{pmatrix}$ the vector that goes up from the center $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ to the point $P_i = (u_i, v_i)$.
We have $OP_i = \alpha_i h + \beta_i g$. The vector w is written as follows $w_i = OP_i + A = \alpha_i h + \beta_i g + A$ avec $\alpha_i = w_i h$ et $\beta_i = w_i g$.
In case our photographer didn't take his camera horizontally, a rotation could be followed by an angle θ.

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix}.$$

The vectors w and $X - \tilde{X}$ are collinear then

$$w = \lambda(X - \tilde{X}). \tag{1.4}$$

The coefficient λ is determined by the condition $w - A$ should be orthogonal to A that gives us $\lambda = \frac{<A,A>}{<X-\tilde{X},A>}$.
From (1.4) we get, $w \wedge (X - \tilde{X}) = 0$, then

$$\begin{pmatrix} w_{2i}(x_{3i} - \tilde{x}_3) - w_{3i}(x_{2i} - \tilde{x}_2) \\ w_{3i}(x_{1i} - \tilde{x}_1) - w_{1i}(x_{3i} - \tilde{x}_3) \\ w_{1i}(x_{2i} - \tilde{x}_2) - w_{2i}(x_{1i} - \tilde{x}_1) \end{pmatrix} = 0.$$

Let $f : \mathbb{R}^7 \to \mathbb{R}^{18}$ and $f(y) = (f_{11}(y), f_{12}(y), f_{13}(y), \ldots, f_{61}(y), f_{62}(y), f_{63}(y))$, where $y = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$.

$f_{i1} = [y_5 - (\cos y_7\, u_i + \sin y_7\, v_i)\frac{y_4}{\sqrt{y_4^2+y_5^2}} + (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_5 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}]x_{3i} - y_3) - [y_6 - (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4^2+y_5^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{2i} - y_2)$.

$f_{i2} = [y_6 - (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4^2+y_5^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{1i} - y_1) - [y_4 + (\cos y_7\, u_i + \sin y_7\, v_i)\frac{y_5}{\sqrt{y_4^2+y_5^2}} + (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{3i} - y_3)$.

$f_{i3} = [y_4 + (\cos y_7\, u_i + \sin y_7\, v_i)\frac{y_5}{\sqrt{y_4^2+y_5^2}} + (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{2i} - y_2) - [y_5 - (\cos y_7\, u_i + \sin y_7\, v_i)\frac{y_4}{\sqrt{y_4^2+y_5^2}} + (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_5 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{1i} - y_1)$.

It is clear that the function f is differentiable for $a, b \neq 0$, we compute now the partial derivatives for f_{i1}, f_{i2} and f_{i3}.

$\frac{\partial f_{i1}}{\partial y_1} = 0$.

$\frac{\partial f_{i1}}{\partial y_2} = y_6 - (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4^2+y_5^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}$.

$\frac{\partial f_{i1}}{\partial y_3} = -y_5 + (\cos y_7\, u_i + \sin y_7 v_i)\frac{y_4}{\sqrt{y_4^2+y_5^2}} + (-\sin y_7 u_i + \cos y_7\, v_i)\frac{y_5 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}$.

$\frac{\partial f_{i1}}{\partial y_4} = [-(\cos y_7\, u_i + \sin y_7\, v_i)\frac{y_5^2}{(y_4^2+y_5^2)^{3/2}} - (\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4 y_5 y_6(2y_4^2+2y_5^2+y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{3i} - y_3) + [(\sin y_7\, u_i - \cos y_7\, v_i)\frac{y_4 y_6^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)^{3/2}}}](x_{2i} - y_2)$.

$\frac{\partial f_{i1}}{\partial y_5} = [1 + (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4 y_5}{(y_4^2+y_5^2)^{3/2}} - (-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_6(y_4^4 - y_5^4 + y_4^2 y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{3i} - y_3) + [(\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_5 y_6^2}{\sqrt{(y_4^2+y_5^2)}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{2i} - y_2).$

$\frac{\partial f_{i1}}{\partial y_6} = [(\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_5\sqrt{(y_4^2+y_5^2)}}{(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{3i} - y_3) - [1 + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_6\sqrt{(y_4^2+y_5^2)}}{(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{2i} - y_2).$

$\frac{\partial f_{i1}}{\partial y_7} = [\frac{-y_4}{\sqrt{y_4^2+y_5^2}}(-\sin y_7\ u_i + \cos y_7\ v_i) + \frac{y_5 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}(\cos y_7\ u_i + \sin y_7\ v_i)](x_{3i} - y_3) + [\frac{y_4^2+y_5^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}(\cos y_7\ u_i + \sin y_7\ v_i)](x_{2i} - y_2).$

$\frac{\partial f_{i2}}{\partial y_1} = -y_6 + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4^2+y_5^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}.$

$\frac{\partial f_{i2}}{\partial y_2} = 0.$

$\frac{\partial f_{i2}}{\partial y_3} = y_4 + (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_5}{\sqrt{y_4^2+y_5^2}} + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}.$

$\frac{\partial f_{i2}}{\partial y_4} = [-(\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4 y_6^2}{\sqrt{y_4^2+y_5^2}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{1i} - y_1) - [1 - (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4 y_5}{(y_4^2+y_5^2)^{3/2}} + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_6(-y_4^4 + y_5^4 + y_5^2 y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{3i} - y_3).$

$\frac{\partial f_{i2}}{\partial y_5} = [-(\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_5 y_6^2}{\sqrt{y_4^2+y_5^2}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{1i} - y_1) - [(\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4^2}{(y_4^2+y_5^2)^{3/2}} - (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4 y_5 y_6(2y_4^2 + 2y_5^2 + y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{3i} - y_3).$

$\frac{\partial f_{i2}}{\partial y_6} = [1 + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_6\sqrt{y_4^2+y_5^2}}{(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{1i} - y_1) - [(\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4\sqrt{y_4^2+y_5^2}}{(y_4^2+y_5^2+y_6^2)^{3/2}}]$
$(x_{3i} - y_3).$

$\frac{\partial f_{i2}}{\partial y_7} = [-(\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4^2+y_5^2}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{1i} - y_1) - [(-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_5}{\sqrt{y_4^2+y_5^2}} + (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{3i} - y_3).$

$\frac{\partial f_{i3}}{\partial y_1} = y_5 - (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4}{\sqrt{y_4^2+y_5^2}} + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_5 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}.$

$\frac{\partial f_{i3}}{\partial y_2} = -y_4 - (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_5}{\sqrt{y_4^2+y_5^2}} - (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}.$

$\frac{\partial f_{i3}}{\partial y_3} = 0.$

$\frac{\partial f_{i3}}{\partial y_4} = [1 - (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4 y_5}{(y_4^2+y_5^2)^{3/2}} + (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_6(-y_4^4 + y_5^4 + y_5^2 y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}]$
$(x_{2i} - y_2) + [(\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_5^2}{(y_4^2+y_5^2)^{3/2}} - (-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_4 y_5 y_6(2y_4^2 + 2y_5^2 + y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}]$
$(x_{1i} - y_1).$

$\frac{\partial f_{i3}}{\partial y_5} = [(\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4^2}{(y_4^2+y_5^2)^{3/2}} - (\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4 y_5 y_6(2y_4^2 + 2y_5^2 + y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}]$
$(x_{2i} - y_2) - [1 + (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4 y_5}{(y_4^2+y_5^2)^{3/2}} - (-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_6(y_4^4 - y_5^4 + y_4^2 y_6^2)}{(y_4^2+y_5^2)^{3/2}(y_4^2+y_5^2+y_6^2)^{3/2}}]$
$(x_{1i} - y_1).$

$\frac{\partial f_{i3}}{\partial y_6} = [(\sin y_7\ u_i - \cos y_7\ v_i)\frac{y_4\sqrt{y_4^2+y_5^2}}{(y_4^2+y_5^2+y_6^2)^{3/2}}](x_{2i} - y_2) + [(-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_5\sqrt{y_4^2+y_5^2}}{(y_4^2+y_5^2+y_6^2)^{3/2}}]$ $(x_{1i} - y_1)$.

$\frac{\partial f_{i3}}{\partial y_7} = [(-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_5}{\sqrt{y_4^2+y_5^2}} + (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_4 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}]$ $(x_{2i} - y_2) + [(-\sin y_7\ u_i + \cos y_7\ v_i)\frac{y_4}{\sqrt{y_4^2+y_5^2}} - (\cos y_7\ u_i + \sin y_7\ v_i)\frac{y_5 y_6}{\sqrt{(y_4^2+y_5^2)(y_4^2+y_5^2+y_6^2)}}](x_{1i} - y_1)$.

Algorithmic part

Data $y_0 = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$, $0 < \epsilon < 1$, $N > 1$, M, u, v, x_1, x_2, x_3.

for $i = 1$ to M do
$y = y_0$;
$n = 0$;
$e = 1$;
While (($e > \epsilon$) and ($n <= N$)) do
Compute $f(y)$ and $Df(y)$
solve $(Df(y))^T Df(y)h = -(Df(y))^T f(y)$
$z = y + h$;
$y = z$;
$e = \|h\|_2$;
$n = n + 1$;
end while
$y = y$;
end for

Matlab programming

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%  The data   %%%%%%%%%%%%%%%%%%%%%%%%%%
y_0=input('y_0=')
M=input('M=')
u=input('u=')
v=input('v=')
x1=input('x1=')
x2=input('x2=')
x3=input('x3=')
epsilon=input('epsilon=')
N=input('N=')

for i=1:M
    disp(['For the position ', num2str(i)])
    y=y_0;
    n=0;
    e=1;
    while((e>epsilon)&&(n<=N))

    %%%%%%%%%%%%%%%%%%%%%%%%%  Computation of f_i1(y) ,f_i2(y) and f_i3(y) %%%%%%
```

```
fi(1)=(y(5)-(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(4)/sqrt(y(4).^2+y(5).^2))+...
      (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(5)*y(6))/sqrt((y(4).^2+y(5).^2)...
      *(y(4).^2+y(5).^2+y(6).^2))))*(x3(i)-y(3))-(y(6)-(sin(y(7))*u(i)-...
      cos(y(7))*v(i))*((y(4).^2+y(5).^2)/sqrt((y(4).^2+y(5).^2)*(y(4).^2+...
      y(5).^2+y(6).^2))))*(x2(i)-y(2));

fi(2)=(y(6)-(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4).^2+y(5).^2)/sqrt((y(4)...
      .^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2))))*(x1(i)-y(1))-(y(4)+...
      (cos(y(7))*u(i)+sin(y(7))*v(i))*(y(5)/sqrt(y(4).^2+y(5).^2))+...
      (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(6))/sqrt((y(4).^2+y(5).^2)*...
      (y(4).^2+y(5).^2+y(6).^2))))*(x3(i)-y(3));

fi(3)=(y(4)+(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(5)/sqrt(y(4).^2+y(5).^2))+...
      (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(6))/sqrt((y(4).^2+y(5).^2)*...
      (y(4).^2+y(5).^2+y(6).^2))))*(x2(i)-y(2))-(y(5)-(cos(y(7))*u(i)+...
      sin(y(7))*v(i))*(y(4)/sqrt(y(4).^2+y(5).^2))+(sin(y(7))*u(i)-...
      cos(y(7))*v(i))*((y(5)*y(6))/sqrt((y(4).^2+y(5).^2)*(y(4).^2+...
      y(5).^2+y(6).^2))))*(x1(i)-y(1));

     %%%%%%%%%%%%%%%%%%%%%%%%%  compuattion of Df(y)   %%%%%%%%%%%%%%%%%%%%%%%%%

Df(1,1)=0;

Df(1,2)=y(6)-(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4).^2+y(5).^2)/sqrt((y(4)...
        .^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)));

Df(1,3)=-y(5)+(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(4)/sqrt(y(4).^2+y(5).^2))-...
        (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(5)*y(6))/sqrt((y(4).^2+y(5).^2)...
        *(y(4).^2+y(5).^2+y(6).^2)));

Df(1,4)=(-(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(5).^2/(y(4).^2+y(5).^2).^(3/2))...
        -(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(5)*y(6)*(2*y(4).^2+2*y(5)...
        .^2+y(6).^2))/((y(4).^2+y(5).^2).^(3/2)*(y(4).^2+y(5).^2+y(6).^2)...
        .^(3/2))))*(x3(i)-y(3))+((sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(6)...
        .^2)/(sqrt(y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*...
        (x2(i)-y(2));

Df(1,5)=((1+(cos(y(7))*u(i)+sin(y(7))*v(i))*((y(4)*y(5))/(y(4).^2+y(5).^2)...
        .^(3/2))+(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(6)*(y(4).^4-y(5).^4+...
        y(4).^2*y(6).^2))/(y(4).^2+y(5).^2).^(3/2)*(y(4).^2+y(5).^2+y(6).^2)...
        .^(3/2))))*(x3(i)-y(3))+((sin(y(7))*u(i)-cos(y(7))*v(i))*((y(5)*y(6)...
        .^2)/(sqrt(y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*...
        (x2(i)-y(2));

Df(1,6)=((sin(y(7))*u(i)-cos(y(7))*v(i))*((y(5)*sqrt(y(4).^2+y(5).^2))/(y(4)...
        .^2+y(5).^2+y(6).^2).^(3/2)))*(x3(i)-y(3))-(1+(sin(y(7))*u(i)-...
        cos(y(7))*v(i))*((y(6)*sqrt(y(4).^2+y(5).^2))/(y(4).^2+y(5).^2+...
```

```
        y(6).^2).^(3/2)))*(x2(i)-y(2));

Df(1,7)=((y(4)/sqrt(y(4).^2+y(5).^2))*(sin(y(7))*u(i)-cos(y(7))*v(i))+...
        ((y(5)*y(6))/sqrt((y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)))*...
        (cos(y(7))*u(i)+sin(y(7))*v(i)))*(x3(i)-y(3))+(((y(4).^2+y(5).^2)/...
        sqrt((y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)))*(cos(y(7))*u(i)+...
        sin(y(7))*v(i)))*(x2(i)-y(2));

Df(2,1)=-y(6)+(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4).^2+y(5).^2)/sqrt((y(4)...
        .^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)));

Df(2,2)=0;

Df(2,3)=y(4)+(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(5)/sqrt(y(4).^2+y(5).^2))+...
        (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(6))/sqrt((y(4).^2+y(5).^2)*...
        (y(4).^2+y(5).^2+y(6).^2)));

Df(2,4)=(-(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(6).^2)/(sqrt(y(4).^2+y(5)...
        .^2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*(x1(i)-y(1))-((1-(cos(y(7))*...
        u(i)+sin(y(7))*v(i))*(y(4)*y(5))/(y(4).^2+y(5).^2).^(3/2))+(sin(y(7))...
        *u(i)-cos(y(7))*v(i))*((y(6)*(-y(4).^4+y(5).^4+y(5).^2*y(6).^2))/...
        ((y(4).^2+y(5).^2).^(3/2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*...
        (x3(i)-y(3));

Df(2,5)=(-(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(5)*y(6).^2)/(sqrt(y(4).^2+y(5)...
        .^2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*(x1(i)-y(1))-((cos(y(7))*...
        u(i)+sin(y(7))*v(i))*(y(4).^2/(y(4).^2+y(5).^2).^(3/2))-(sin(y(7))*...
        u(i)-cos(y(7))*v(i))*((y(4)*y(5)*y(6)*(2*y(4).^2+2*y(5).^2+y(6).^2))/...
        ((y(4).^2+y(5).^2).^(3/2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*...
        (x3(i)-y(3));

Df(2,6)=(1+(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(6)*sqrt(y(4).^2+y(5).^2))/...
        (y(4).^2+y(5).^2+y(6).^2).^(3/2)))*(x1(i)-y(1))-((sin(y(7))*u(i)-...
        cos(y(7))*v(i))*((y(4)*sqrt(y(4).^2+y(5).^2))/(y(4).^2+y(5).^2+...
        y(6).^2).^(3/2)))*(x3(i)-y(3));

Df(2,7)=(-((y(4).^2+y(5).^2)/sqrt((y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6)...
        .^2)))*(cos(y(7))*u(i)+sin(y(7))*v(i)))*(x1(i)-y(1))-((y(5)/...
        sqrt(y(4).^2+y(5).^2))*(-sin(y(7))*u(i)+cos(y(7))*v(i))+((y(4)*...
        y(6))/sqrt((y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)))*(cos(y(7))*...
        u(i)+sin(y(7))*v(i)))*(x3(i)-y(3));

Df(3,1)=y(5)-(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(4)/sqrt(y(4).^2+y(5).^2))+...
        (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(5)*y(6))/sqrt((y(4).^2+y(5).^2)*...
        (y(4).^2+y(5).^2+y(6).^2)));

Df(3,2)=-y(4)-(cos(y(7))*u(i)+sin(y(7))*v(i))*(y(5)/sqrt(y(4).^2+y(5).^2))-...
        (sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(6))/sqrt((y(4).^2+y(5).^2)*...
```

```
        (y(4).^2+y(5).^2+y(6).^2)));

Df(3,3)=0;

Df(3,4)=(1-(cos(y(7))*u(i)+sin(y(7))*v(i))*((y(4)*y(5))/(y(4).^2+y(5).^2)...
        .^(3/2))+(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(6)*(-y(4).^4+y(5).^4+...
        y(5).^2*y(6).^2))/((y(4).^2+y(5).^2).^(3/2)*(y(4).^2+y(5).^2+...
        y(6).^2).^(3/2))))*(x2(i)-y(2))+((cos(y(7))*u(i)+sin(y(7))*v(i))*...
        ((y(5).^2)/(y(4).^2+y(5).^2).^(3/2))-(-sin(y(7))*u(i)+cos(y(7))*...
        v(i))*((y(4)*y(5)*y(6)*(2*y(4).^2+2*y(5).^2+y(6).^2))/((y(4).^2+...
        y(5).^2).^(3/2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*(x1(i)-y(1));

Df(3,5)=((cos(y(7))*u(i)+sin(y(7))*v(i))*((y(4).^2)/(y(4).^2+y(5).^2)...
        .^(3/2))-(sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*y(5)*y(6)*(2*y(4)...
        .^2+2*y(5).^2+y(6).^2))/((y(4).^2+y(5).^2).^(3/2)*(y(4).^2+y(5).^2+...
        y(6).^2).^(3/2))))*(x2(i)-y(2))-(1+(cos(y(7))*u(i)+sin(y(7))*v(i))*...
        ((y(4)*y(5))/(y(4).^2+y(5).^2).^(3/2))+(sin(y(7))*u(i)-cos(y(7))*...
        v(i))*((y(6)*(y(4).^4-y(5).^4+y(4).^2*y(6).^2))/((y(4).^2+y(5).^2)...
        .^(3/2)*(y(4).^2+y(5).^2+y(6).^2).^(3/2))))*(x1(i)-y(1));

Df(3,6)=((sin(y(7))*u(i)-cos(y(7))*v(i))*((y(4)*sqrt(y(4).^2+y(5).^2))/...
        (y(4).^2+y(5).^2+y(6).^2).^(3/2)))*(x2(i)-y(2))+((-sin(y(7))*u(i)+...
        cos(y(7))*v(i))*((y(5)*sqrt(y(4).^2+y(5).^2))/(y(4).^2+y(5).^2+...
        y(6).^2).^(3/2)))*(x1(i)-y(1));

Df(3,7)=((y(5)/sqrt(y(4).^2+y(5).^2))*(-sin(y(7))*u(i)+cos(y(7))*v(i))+...
        ((y(4)*y(6))/sqrt((y(4).^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)))*...
        (cos(y(7))*u(i)+sin(y(7))*v(i)))*(x2(i)-y(2))+((y(4)/sqrt(y(4).^2+...
        y(5).^2))*(-sin(y(7))*u(i)+cos(y(7))*v(i))-((y(5)*y(6))/sqrt((y(4)...
        .^2+y(5).^2)*(y(4).^2+y(5).^2+y(6).^2)))*(cos(y(7))*u(i)+sin(y(7))*...
        v(i)))*(x1(i)-y(1));

       %%%%%%%%%%%%%%%%%%%%%%%%%  computation of h   %%%%%%%%%%%%%%%%%%%%%%%%%

Df_transpose=Df';
w1=Df_transpose*Df;
r=(-1)*Df_transpose;
fi_transpose=fi';
w2=r*fi_transpose;
h_column=w1\w2;
h=h_column';

Z=y+h; y=Z; e=norm(h); n=n+1; end y=y; disp('h=') disp(h)
disp(['The number of iterations is n=',num2str(n)]) disp('e=')
disp(e) disp('The solution is y=') disp(y) end
```

Matlab results

```
>> format short g
y_0=[9660 13100 4120 -0.040 -0.17 -0.030 -0.072]

y_0 =

  Columns 1 through 6

        9660        13100         4120        -0.04        -0.17        -0.03

  Column 7

      -0.072

M=6

M =

     6

u=[-0.0480 -0.0100 0.0490 -0.0190 0.0600 0.0125]

u =

      -0.048        -0.01        0.049       -0.019         0.06       0.0125

v=[0.029 0.0305 0.0285 0.0115 -0.0005 -0.0270]

v =

       0.029       0.0305       0.0285       0.0115      -0.0005       -0.027

x1=[9855 8170 2885 8900 5700 8980]

x1 =

        9855         8170         2885         8900         5700         8980

x2=[5680 5020 730 7530 7025 11120]

x2 =

        5680         5020          730         7530         7025        11120

x3=[3825 4013 4107 3444 3008 3412]

x3 =
```

```
        3825         4013         4107         3444         3008         3412

epsilon=10^(-3)

epsilon =

        0.001

N=50

N =

    50

For the position 1
h=
  Columns 1 through 6

  1.6815e-012  1.0422e-009  1.1771e-010  7.7685e-016 -2.1417e-017 -1.0893e-015

  Column 7

 -1.2129e-014

The number of iterations is n=8
e=
  1.0488e-009

The solution is y=
  Columns 1 through 6

        10984        14487       4638.6    -0.046126     -0.06209     -0.03364

  Column 7

    -0.049187

For the position 2
h=
  Columns 1 through 6

  5.5218e-009  1.8661e-008 -3.7265e-009  2.4892e-014  1.4626e-016  2.6297e-013

  Column 7

  7.2954e-013
```

```
The number of iterations is n=4
e=
  1.9815e-008

The solution is y=
  Columns 1 through 6

       9515.3        13297       4037.3    -0.041745     -0.16999    -0.028852

  Column 7

     -0.14666

For the position 3
h=
  Columns 1 through 6

 -2.9937e-006  2.8168e-006 -5.4884e-007   1.875e-011  3.6565e-016  1.7091e-010

  Column 7

 -5.0073e-010

The number of iterations is n=5
e=
   4.147e-006

The solution is y=
  Columns 1 through 6

       8282.7        14402       3408.6    -0.048728     -0.17605    -0.042128

  Column 7

     -0.67027

For the position 4
h=
  Columns 1 through 6

  1.3667e-009  3.0994e-008   1.876e-009  1.8466e-013  9.1976e-016 -7.1559e-015

  Column 7

  7.8447e-012

The number of iterations is n=4
e=
```

```
  3.1081e-008

The solution is y=
  Columns 1 through 6

       9548.6        13620       4098.4     -0.03852      -0.1762    -0.029223

  Column 7

    -0.087565

For the position 5

h=
  Columns 1 through 6

  7.4012e-005   0.00031172   4.932e-005  2.7957e-009   1.139e-015  1.1551e-010

  Column 7

  3.7323e-009

The number of iterations is n=5
e=
   0.00032416

The solution is y=
  Columns 1 through 6

       8779.7        12228       3929.4    -0.045938     -0.17937    -0.051743

  Column 7

     -0.38771

For the position 6
h=
  Columns 1 through 6

  6.9646e-009  5.9936e-008 -1.9183e-010  1.1385e-012  9.0077e-016  1.2816e-012

  Column 7

  9.3466e-012

The number of iterations is n=4
e=
   6.034e-008
```

```
The solution is y=
  Columns 1 through 6

       9650.5        13128       4102.1     -0.040675      -0.16464     -0.027755

  Column 7

    -0.028034
```

After few iterations, the Gauss-Newton method converges for this choice of initial values (see the table 2). We have so good found the position of the photographer: This photo was taken at the Aiguille Verte.

Table 2: Convergence of Gauss Newton's method

i	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	a	b	c	θ
0	9660	13100	4120	-0.0400	-0.1700	-0.0300	-0.0720
1	10984	14487	4638.6	-0.0461	-0.0620	-0.0336	-0.0491
2	9515.3	13297	4037.3	-0.0417	-0.1699	-0.0288	-0.1466
3	8282.7	14402	3408.6	-0.0487	-0.1760	-0.0421	-0.6702
4	9548.6	13620	4098.4	-0.0385	-0.1762	-0.0292	-0.0875
5	8779.7	12228	3929.4	-0.0459	-0.1793	-0.0517	-0.3877
6	9650.5	13128	4102.1	-0.0406	-0.1646	-0.0277	-0.0280

Conclusion: The choice of initial values is essential, it directly influences the convergence of the method. (see the array(3)).

Table 3: Influence of the initial values on the convergence of the Gauss-Newton method

Initial values	Nature
9660 13100 4120 -0.040 -0.170 -0.030 -0.072	convergence
9660 13000 4100 -0.040 -0.170 -0.030 -0.072	divergence
9000 13000 1000 -0.040 -0.200 0.000 0.000	divergence
9600 14000 3000 -0.040 -0.200 -0.050 0.000	divergence

References

[1] H. Brezis, Functional analysis, Sobolev spaces and partial differential equations, Springer, New York, NY, 2011. https://doi.org/10.1007/978-0-387-70914-7.

[2] Pic2Map. https://www.pic2map.com.

[3] K.Madsen, H.B. Nielsen, O. Tingleff Methods for nonlinear least squares problems, 2nd edition, April 2004.