

**Research Paper** 

# ADAMS-MOULTON METHOD APPROACH TO GEODESIC PATHS ON 2D SURFACES

ESA SHARAHI\*, ESMAIEL PEYGHAN, AND AMIR BAGHBAN

ABSTRACT. Our aim in this paper, is applying Adams-Moulton algorithm to find the geodesics as the answers of the classical system of ordinary differential equations on a 2-dimensional surface for which a Riemannian metric is defined.

**MSC(2010):** 53B50.

**Keywords:** Adams-Moulton algorithm, Euclidean space, geodesic, navigation problem, Riemannian manifold, Runge-Kutta algorithm.

### 1. Introduction

Short paths are known as geodesics for a manifold equipped with a connection. After fixing the shape of a manifold by a connection, one can compute the short paths between two points and get the Euler-Lagrange equation. Discussions related to the geodesic are of the most important topics in differential and computational geometry. Geodesics are completely local and intrinsic concepts and their related theorems, intensify control optimum approaches in differential manifolds (see [6]). For a historical story about remarkable facts on geodesics, take a look at [2]. Moreover, there are extremely open problems about them (see e.g. [3]) that only experiential observations can motivate just only a numerical solution. Geodesics applied in various aspects of science also. For example, grid technology, manifold learning, navigation problems, image processing and optimal control problems are dealing with them amazingly (see [5, 10, 11]). There are several approaches to meet formulas describing geodesics in a space equipped with geometrical objects. But all of them reach to a way parallel to the solving of a quasi-linear system of ordinary differential equations. In real-world situations, there is no solution for these systems of ODEs analytically feasible. So, numerical perception handles changing the approach. Indeed, the survivor way is to detect the geodesic trajectories passing a specified point in a certain direction using numerical tips. See [8, 9] for some classical methods. In

Date: Received: August 23, 2020, Accepted: December 13, 2020.

<sup>\*</sup>Corresponding author.

this paper, we use Adams-Moulton algorithm to solve the generator system of ODEs of geodesics embedded in a 3-dimensional Riemannian space. This algorithm has a high insurance of convergence. It is remarkable that one can use all of these methods and specially the code in the Finslerian cases only by a small change in geodesics equations. For a complete discussion about geodesics in Finsler geometry, see [1].

In the next section, we have a concise view on some needed definitions and also explain our manner to solve the problem. Section 3 is devoted to four famous examples. We drop details of computations on finding the geodesic formulas because of simplicity in reading. Finally the last section exhibits the algorithm and code in details.

## 2. Preliminaries

Equipping a manifold by an (affine) connection makes it possible to fix some of the features like curvature and minimal trajectories on that. It is well-known that any paracompact manifold, has a Riemannian metric. Let  $(M^n, g)$  be a smooth Riemannian manifold by the unique associated Levi-Civita connection  $\nabla$ .

Roughly, a curve is the locus of points whose positions are represented by a vector-valued function of a single parameter varying on a subdomain of  $\mathbb{R}$ . This function is a parametrization of the curve. It is remarkable that the Riemannian metric g induces a distance metric  $d_g(p,q) = \inf(\ell(c))$ where  $c : [a,b] \longrightarrow M$  varies on the set of all curves joining p to q and  $\ell(c) = \int_a^b \sqrt{g(\dot{c}(t), \dot{c}(t))} dt$  in a natural way (nowadays, it is a conventional assumption that M be a Hausdorff space to carry this distance metric). This functional is compatible with the topological structure of M as a topological manifold. Namely, it is symmetric, satisfies triangle inequality and bounded. Many of geometrical objects can be generalized using this distance function. But, l(c) is difficult to compute in general and is usually overlooked in practice. In a rather weak situation, to approximate the geodesic distance between nearby points in manifolds with sufficient dense atlas, divergence methods can be applied.

By definition, if a smooth curve  $\gamma : [a, b] \subseteq \mathbb{R} \longrightarrow M$  parallel-transports it own tangent vectors, then it is a geodesic on M. This means that  $\nabla_{\dot{\gamma}} \dot{\gamma} = 0$ . The latter equality leads to the following system of ODEs

(2.1) 
$$\frac{\mathrm{d}^2 \gamma^k(t)}{\mathrm{d}t^2} + \Gamma^k_{ij}(t) \frac{\mathrm{d}\gamma^i(t)}{\mathrm{d}t} \frac{\mathrm{d}\gamma^j(t)}{\mathrm{d}t} = 0,$$

where  $i, j, k \in \{1, \dots, n\}$ ,  $\gamma^{i}$ 's are the components of  $\gamma$  and  $\nabla_{\frac{\partial}{\partial x^{i}}} \frac{\partial}{\partial x^{j}} = \Gamma^{k}{}_{ij}\frac{\partial}{\partial x^{k}}$ . The parameter t in  $\gamma(t)$  in differential system of geodesic has the homogeniety property. Indeed, if  $\gamma(t)$  is a geodesic, then for any nonzero constant  $\lambda$ , the curve  $\gamma(\lambda t)$  is also a geodesic.

By the way, this system of ODEs, retrieves the chance of compute the l(c). It is a consequence of the famous Hopf-Rinow theorem that if M be a complete, finite dimension and connected Riemannian manifold, then any two

83

points in M can join by a length minimizing geodesic. Detailed discussions could be found in almost all Riemannian geometry books. For example, one can see a complete survey in [7]. However, system (2.1) obtained by real-world examples is not solvable analitically and instantly requires computing numerical solutions.

Now, we restrict our attention to the case of surface S laying in  $\mathbb{R}^3$ . Consider such a surface with arbitrary coordinate system (x, y) and with an arbitrary Riemannian metric reads

$$g = E \mathrm{d}x \otimes \mathrm{d}x + F \mathrm{d}x \otimes \mathrm{d}y + F \mathrm{d}y \otimes \mathrm{d}x + G \mathrm{d}y \otimes \mathrm{d}y,$$

where  $E, F, G \in C^{\infty}(M)$ . Using the well-known formula for Christoffel symbols

$$\Gamma_{ij}^{k} = \frac{1}{2}g^{kl}\left\{\frac{\partial g_{li}}{\partial x^{j}} + \frac{\partial g_{lj}}{\partial x^{i}} - \frac{\partial g_{ji}}{\partial x^{l}}\right\},$$

(where we suppose that  $x^1 = x$ ,  $x^2 = y$ ) it is not hard to investigate that we can write the system (2.1) by this terminology as follow

(2.2) 
$$\alpha'' + \frac{GE_x - 2FF_x + FE_y}{2(EG - F^2)} (\alpha')^2 + \frac{GE_y - FG_x}{EG - F^2} \alpha' \beta' + \frac{2GF_y - GG_x - FG_y}{2(EG - F^2)} (\beta')^2 = 0,$$

and

(2.3) 
$$\beta'' + \frac{2EF_x - EE_y - FE_x}{2(EG - F^2)} (\alpha')^2 + \frac{EG_x - FE_y}{EG - F^2} \alpha' \beta' + \frac{EG_y - 2FF_y + FG_x}{2(EG - F^2)} (\beta')^2 = 0,$$

where  $\gamma(t) = (\alpha(t), \beta(t), S(\alpha(t), \beta(t)))$  is a parametrization for  $\gamma = \gamma(t)$  on the surface S. Latter relations are the underlyings of our approach to the problem. Indeed, we pass this data to the computer by mentioned algorithm and the output produce geodesic on surface (under existence initial conditions).

#### 3. Examples

In this section, we use modified formulas (2.2) and (2.3) to study some famous examples. The figures are obtained by numerical analysis using Adams-Moulton algorithm described in the last section by code. Before move on, we remind that the computations are omitted and we write the results briefly. It is notable that we focus on more prevalent Euclidean examples to check the code result intuitively more simple whereas one can apply the method for non-Euclidean cases obviously.

Here, there is a remarkable fact about how justify equations to pass them to code. Indeed, the computational approaches are heavily benchmarked by entirely eroding iterations among other algorithmic facts. So, modifying the equations in such a way that makes reductions on iterations, makes the code faster and so more applicable in real-time situations. Therefore, instead of using strict equations of geodesics, we modify them such that the code benefits from the pattern repeated among them. This way, we derive a more sharply computer code.

3.1. Sphere. Assume the upper half sphere defined by  $x^2 + y^2 + z^2 = 1$ , z > 0 and so the geodesic parametric equation shall be

$$\gamma(t) = (\alpha(t), \beta(t), \sqrt{1 - \alpha^2(t) - \beta^2(t)}).$$

By straightforward computations, (2.2) and (2.3) convert to

$$\begin{split} \alpha^{\prime\prime} &= -\frac{\alpha(\beta^2-1)}{1-\alpha^2-\beta^2} (\alpha^\prime)^2 - \frac{2\alpha^2\beta}{1-\alpha^2-\beta^2} \alpha^\prime\beta^\prime \\ &- \frac{\alpha(1-\alpha^2)}{1-\alpha^2-\beta^2} (\beta^\prime)^2, \end{split}$$

and

$$(\beta)'' = -\frac{\beta(\alpha^2 - 1)}{1 - \alpha^2 - \beta^2} (\beta')^2 - \frac{2\beta^2 \alpha}{1 - \alpha^2 - \beta^2} \alpha' \beta' - \frac{\beta(1 - \beta^2)}{1 - \alpha^2 - \beta^2} (\beta')^2.$$

By initial data

(3.1) 
$$\alpha(0) = \beta(0) = 0; \qquad \alpha'(0) = 1; \qquad \beta'(0) = 0,$$

we will have the following result.



FIGURE 1. Plot for initial conditions (3.1)

85

3.2. **Ellipsoid.** For the ellipsoid given by  $x^2 + \frac{y^2}{4} + z^2 = 1$  where it supposed that z is positive, straight computations lead to the equations

$$\begin{aligned} \alpha'' + \frac{16\alpha(1-\frac{\beta^2}{4})}{(1-\alpha^2-\frac{\beta^2}{4})\left(16(1-\alpha^2-\frac{\beta^2}{4})+16\alpha^2+\beta^2\right)}(\alpha')^2 \\ + \frac{8\alpha^2\beta}{(1-\alpha^2-\frac{\beta^2}{4})\left(16(1-\alpha^2-\frac{\beta^2}{4})+16\alpha^2+\beta^2\right)}\alpha'\beta' \\ \frac{4\alpha(1-\alpha^2)}{(1-\alpha^2-\frac{\beta^2}{4})\left(16(1-\alpha^2-\frac{\beta^2}{4})+16\alpha^2+\beta^2\right)}(\beta')^2 &= 0, \end{aligned}$$

and

$$\beta'' + \frac{4\beta(1-\frac{\beta^2}{4})}{(1-\alpha^2-\frac{\beta^2}{4})\left(16(1-\alpha^2-\frac{\beta^2}{4})+16\alpha^2+\beta^2\right)}(\alpha')^2 + \frac{2\alpha\beta^2}{(1-\alpha^2-\frac{\beta^2}{4})\left(16(1-\alpha^2-\frac{\beta^2}{4})+16\alpha^2+\beta^2\right)}\alpha'\beta' + \frac{\beta(1-\alpha^2)}{(1-\alpha^2-\frac{\beta^2}{4})\left(16(1-\alpha^2-\frac{\beta^2}{4})+16\alpha^2+\beta^2\right)}(\beta'')^2 = 0,$$

giving the geodesic

$$\gamma(t) = (\alpha(t), \beta(t), \sqrt{1 - \alpha^2(t) - \frac{\beta(t)^2}{4}}).$$

By initial data

(3.2) 
$$\alpha(0) = \beta(0) = 0; \qquad \alpha'(0) = 1; \qquad \beta'(0) = 0,$$

we have the following result.



FIGURE 2. Plot for initial conditions (3.2)

3.3. Cylinder. Take the cylinder with local definition  $X(u, v) = (\cos u, \sin u, v)$ where  $(u, v) \in (0, \pi) \times (-\infty, \infty)$ . If  $\gamma(t) = (\cos(\alpha(t)), \sin(\alpha(t)), \beta(t))$  be a geodesic, it satisfies the equation  $\gamma'' = 0$  equivalent with  $\alpha'' = \beta'' = 0$ . For this well-known situation, we consider three following cases of initial conditions

(3.3) 
$$\alpha(0) = \frac{\pi}{2}, \quad \beta(0) = 0, \quad \alpha'(0) = 1, \quad \beta'(0) = 0,$$

(3.4) 
$$\alpha(0) = \frac{\pi}{2}, \quad \beta(0) = 0, \quad \alpha'(0) = 0, \quad \beta'(0) = 1,$$

and

(3.5) 
$$\alpha(0) = \frac{\pi}{2}, \quad \beta(0) = 0, \quad \alpha'(0) = \frac{1}{\sqrt{2}}, \quad \beta'(0) = \frac{1}{\sqrt{2}},$$

to behold all types of geodesics in cylinder.



3.4. Saddle Surface. Let (x, y, z) be the saddle given by  $z = x^3 - 3xy^2$ . We suppose  $\gamma(t) = (\alpha(t), \beta(t), \alpha^3 - 3\alpha\beta^2(t))$  is a geodesic on saddle. So, after some computations, it results that the geodesic can be given by the following equations

$$\begin{aligned} \alpha'' + \frac{648\alpha\beta^2(\alpha^2 - \beta^2)^2}{2 + 18(\alpha^2 + \beta^2)^2}(\alpha')^2 + \frac{-36(\alpha^2 - \beta^2)}{2 + 18(\alpha^2 + \beta^2)^2}\alpha'\beta' \\ + \frac{36(1 + 36\alpha^2\beta^2)(-\alpha^3 + 3\alpha\beta^2) - 72\alpha\beta^2(1 + 36\alpha^2\beta^2) + 1296\alpha^3\beta^2(\alpha^2 - \beta^2)}{2 + 18(\alpha^2 + \beta^2)^2}(\beta')^2 &= 0, \end{aligned}$$

$$\beta'' + \frac{36(1+9(\alpha^2-\beta^2)^2)(\beta^3-3\beta\alpha^2)+36\beta(\alpha^2-\beta^2)(1+9(\alpha^2-\beta^2)^2)+648\alpha^2\beta(\alpha^2-\beta^2)^2}{2+18(\alpha^2+\beta^2)^2}(\alpha')^2 + \frac{72\alpha\beta^2}{2+18(\alpha^2+\beta^2)^2}\alpha'\beta' + \frac{(72\alpha^2\beta)(1+9(\alpha^2-\beta^2)^2)+648\alpha\beta(\alpha^2-\beta^2)(3\alpha\beta^2-\alpha^3)-1296\alpha^2\beta^3(\alpha^2-\beta^2)}{2+18(\alpha^2+\beta^2)^2}(\beta')^2 = 0.$$

We should plot the minimum trajectory curve in  $\mathbb{R}^3$  using explicit formula  $(\alpha, \beta, \alpha^3 - 3\alpha\beta^2)$  with the following 3 cases of initial conditions

(3.6) 
$$\alpha(0) = 0, \quad \beta(0) = 0, \quad \alpha'(0) = 1, \quad \beta'(0) = 0,$$

(3.7) 
$$\alpha(0) = 0, \quad \beta(0) = 0, \quad \alpha'(0) = 0, \quad \beta'(0) = 1,$$
  
and

(3.8) 
$$\alpha(0) = 0, \quad \beta(0) = 0, \quad \alpha'(0) = \frac{1}{\sqrt{2}}, \quad \beta'(0) = \frac{1}{\sqrt{2}}.$$



FIGURE	FIGURE	FIGURE
6. Plot	7. Plot	8. Plot
for ini-	for ini-	for ini-
tial	tial	tial
condi-	condi-	condi-
tions	tions	tions
( <b>3.6</b> )	(3.7)	(3.8)

## 4. Implementation details

Adams-Moulton algorithm as a classical time integration technique in solving initial conditioned systems of ODEs is categorised as a multistep method. Remind that for any given set of n + 1 distinct points in the space, there is a unique polynomial of degree n that fits them. Adams-Moulton technique structure enjoys from feeding previous computational data at each step and constructs the Lagrange interpolating polynomial as soon as possible. See [4] for a detailed discussion on method. We use Adams-Bashforth methods as a predictor-corrector and pass the main iteration to Adams-Moulton method. The code is sufficiently flexible to import any other example and try other situations. So, it can be use as a user-friendly library to import to any other project.

All of the following listings are about the complete implementations of examples presented in this paper.

We stop the iterations until the local truncation error for every example cuts the 10E - 7 off.

File "Run.m":

```
clc;
1
\mathbf{2}
      clear;
3
      addpath('Surface');
4
5
6
      P = input('Enter surface name: ', 's');
      O = input('Enter order of -AdamsMoulton solver: ');
\overline{7}
      Step = input('Enter number of steps: ');
8
      Dlta = input('Enter step size: ');
9
10
      switch P
11
12
         case 'Sphere'
            Func = @SphereD;
13
            Surf = @SphereS;
14
         case 'Ellipse'
15
             Func = @EllipseD;
16
             Surf = @EllipseS;
17
         case 'Cylinder
18
             Func = @CylinderD;
19
             Surf = @CylinderS;
20
         case 'Saddle
21
            Func = @SaddleD;
22
23
             Surf = @SaddleS;
24
      end
25
26
      Data = zeros(Step + 1, 4);
27
      Data(1,1) = input('Enter initial condition "Alpha": ');
28
      Data(1,2) = input('Enter initial condition "Beta" : ');
29
      Data(1,3) = input('Enter initial condition "a" : '); Setup
30
                                                             : ');
      Data(1,4) = input('Enter initial condition "b"
31
32
      for i = 1:Step
33
         if(i \ll 0)
34
35
            N = i;
36
             Cons = Adams(i);
37
         end
         Data(i + 1,:) = AdamsMoulton(Func, Cons, Dlta, Data(i + 1 - N \leftrightarrow
38
               : i,:));
39
      end
40
```

88

```
41 figure;
42 XYZ = Surf(Data(1:Step,1),Data(1:Step,2));
43 plot3(XYZ(:,1),XYZ(:,2),XYZ(:,3));
```

File "Adams.m":

```
function Cons = Adams(Ordr)
 1
 \mathbf{2}
 3
        Cons = zeros(Ordr + 1, 2);
        syms u v;
 4
 5
        \% Adams-Bashforth
 \mathbf{6}
        for j = 0: \text{Ordr} - 1
 \overline{7}
             v = 1 / (u + j);
 8
             for i = 0: \text{Ordr} - 1
 9
                      v = v * (u + i);
10
             end
11
             Cons(Ordr - j, 1) = (-1) \hat{j} / factorial(j) / factorial(\leftrightarrow
12
                   Ordr - j - 1) * double(int(v, u, 0, 1));
13
        end
14
        \% Adams-Moulton
15
        for j = 0:Ordr
16
             v = 1 / (u + j - 1);
17
             for i = 0: Ordr
18
                      \mathbf{v} = \mathbf{v} * (\mathbf{u} + \mathbf{i} - 1);
19
             end
20
             \texttt{Cons}(\texttt{Ordr} - \texttt{j} + 1, 2) = (-1) \hat{\texttt{j}} / \texttt{factorial}(\texttt{j}) / \texttt{factorial} \leftrightarrow
21
                   (\operatorname{Ordr} - j) * \operatorname{double}(\operatorname{int}(v, u, 0, 1));
22
        end
23
24
        end
```

File "AdamsBashforth.m":

```
function Y = AdamsBashforth(Func,Cons,h,X)
1
\mathbf{2}
3
     Ordr = size(X, 1);
     Y = X(Ordr, :);
4
     for i = 1: Ordr
\mathbf{5}
         Y = Y + h * Cons(i,1) * Func(X(i,:));
6
     end
7
8
9
     end
```

File "AdamsMoulton.m":

```
1 function Y = AdamsMoulton(Func, Cons, h, X)

2

3 Ordr = size(X,1);

4 Z = AdamsBashforth(Func, Cons, h, X);

5 Conv = 1;

6 while(Conv > 1e-10)
```

```
7 Y = X(Ordr,:) + h * Cons(Ordr + 1,2) * Z;
8 for i = 1:Ordr
9 Y = Y + h * Cons(i,2) * Func(X(i,:));
10 end
11 Conv = (norm(Y) - norm(Z)) / norm(Y); Z = Y;
12 end
13
14 end
```

File "Surface/SphereS.m":

1 function XYZ = SphereS(U,V) 2 3 XYZ = [U, V, sqrt(1 - U .\* U - V .\* V)]; 4 5 end

File "Surface/SphereD.m":

function Y = SphereD(X)1  $\mathbf{2}$  $A = 1 - X(1)^2;$ 3  $B = 1 - X(2)^2;$ 4  $\mathsf{D} = \mathsf{A} + \mathsf{B} - 1;$  $\mathbf{5}$ 6 A = A / D; $\overline{7}$ B = B / D;8 C = (2 \* X(1) \* X(2)) / D;9 10 L = X(3) \* X(3);11  $\mathbb{M} = \mathbb{X}(3) * \mathbb{X}(4);$ 1213 $\mathbb{N} = \mathbb{X}(4) * \mathbb{X}(4);$ 14Y(1) = X(3);15 $\mathbf{Y}(2) = \mathbf{X}(4);$ 16Y(3) = X(1) \* (B \* L - C \* M - A \* N);17Y(4) = X(2) \* (-B \* L - C \* M + A \* N);181920end

File "Surface/EllipseS.m":

1 function XYZ = EllipseS(U,V)
2
3 XYZ = [U, V, sqrt(1 - U .\* U - V .\* V / 4)];
4
5 end

File "Surface/EllipseD.m":

1 function Y = EllipseD(X)2  $A = X(1)^2;$ 

90

```
B = X(2)^2;
4
      C = 1 - A - B / 4;
5
      D = C * (16 * (C + A) + B);
6
7
      A = (1 - A) / D;
8
      B = (1 - B / 4) / D;
9
      C = (X(1) * X(2)) / D;
10
11
      L = X(3) * X(3);
12
      M = X(3) * X(4);
13
      \mathbf{N} = \mathbf{X}(4) * \mathbf{X}(4);
14
15
      Y(1) = X(3);
16
      \mathbf{Y}(2) = \mathbf{X}(4);
17
18
      Y(3) = X(1) * (-16 * B * L - 8 * C * M - 4 * A * N);
      Y(4) = X(2) * (-4 * B * L - 2 * C * M - 
19
                                                          A * N);
20
21
      end
```

File "Surface/CylinderS.m":

```
1 function XYZ = CylinderS(U,V)
2
3 XYZ = [sin(U), cos(U), V];
4
5 end
```

File "Surface/CylinderD.m":

1 function Y = CylinderD(X)
2
3 Y(1) = X(3);
4 Y(2) = X(4);
5 Y(3) = 0;
6 Y(4) = 0;
7
8 end

File "Surface/SaddleS.m":

1 function XYZ = SaddleS(U,V) 2 3 XYZ = [U, V, U .^ 3 - 3 \* U .\* V .^ 2]; 4 5 end

File "Surface/SaddleD.m":

1 function Y = SaddleD(X)2 3 Al = X(1);4 Bt = X(2);5

```
A12 = X(1) * X(1);
 6
 \overline{7}
      Bt2 = X(2) * X(2);
 8
      TA = A12 - Bt2;
9
      TB = 1 + 36 * A12 * Bt2;
10
      TC = 1 + 9 * TA ^ 2;
11
12
      A1 = 648 * A1 * Bt2 * TA * TA;
13
      B1 = -36 * TA;
14
      C1 = A1 * ( 36 * TB * (3 * Bt2 - A12) \dots
15
                 -72 * Bt2 * TB \dots
16
                 + 1296 * A12 * Bt2 * TA);
17
18
      A2 = Bt * ( 36 * TC * (Bt2 - 3 * A12) \dots
19
20
                 + 36 * TA * TC ...
21
                 + 648 * A12 * TA * TA);
      B2 = 72 * A1 * Bt2;
22
      C2 = Bt * (72 * Al2 * TC \dots
23
                 + 648 * Al2 * TA * (3 * Bt2 - Al2) \dots
24
                 -1296 * Al2 * Bt2 * TA);
25
26
      \mathbf{H} = - 2 * \mathbf{TC};
27
28
      L = X(3) * X(3);
29
      \mathsf{M} = \mathtt{X}(3) \ast \mathtt{X}(4);
30
      \mathbb{N} = \mathbb{X}(4) * \mathbb{X}(4);
31
32
      Y(1) = X(3);
33
34
      Y(2) = X(4);
35
      Y(3) = X(1) * (A1 * L + B1 * M + C1 * N) / H;
      Y(4) = X(2) * (A2 * L + B2 * M + C2 * N) / H;
36
37
38
      end
```

#### References

- D. Bao, S. Chern, Z. Shen, An Introduction to Riemann-Finsler Geometry, Springer-Verlag, 2000.
- [2] M. Berger, A Panoramic View of Riemannian Geometry, Springer-Verlag, 2003.
- [3] K. Burns, V. Matveev, Open Problems and Questions About Geodesics, arXiv:1308.5417v2.
- [4] J. C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley, 2003.
- [5] S. Fiori, Quasi-Geodesic neural learning algorithms over the orthogonal group: A tutorial, Journal of Machine Learning Research, 6, 2005, 743-781.
- [6] D. Gabay, Minimizing a differentiable function over a differential manifold, Journal of Optimization Theory and Applications, 37(2), 1982, 177-219.
- [7] M. M. Postnikov, Geometry VI: Riemannian Geometry, Springer-Verlag, 2001.
- [8] C. Scheffer, J. Vahrenhold, Approximating geodesic distances on 2-manifolds in ℝ<sup>2</sup>, Computational Geometry, 47(2A), 2014, 125-140.
- [9] C. Udrişte, Convex functions and optimization methods in Riemannian manifolds, Mathematics and Its Applications, Vol. 297, Kluwer Academic, Dordrecht, 1994.

92

- [10] S. Vacaru, Nonholonomic algebroids, Finsler geometry, and Lagrange-Hamilton spaces, Math. Sci., 6(18), 2012, doi:10.1186/2251-7456-6-18, arXiv: 0705.0032.
- [11] D. Zosso, X, Bresson, JP. Thiran, Geodesic active fields-a geometric framework for image registration, IEEE Trans Image Process, 20(5), 2011, 1300-1312.

(Esa Sharahi) DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, ARAK UNIVERSITY, ARAK 38156-8-8349, IRAN.

Email address: esasharahi@gmail.com

(Esmaiel Peyghan) DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, ARAK UNIVERSITY, ARAK 38156-8-8349, IRAN. *Email address:* epeyghan@gmail.com

(Amir Baghban) DEPARTMENT OF MATHEMATICS, FACULTY OF SCIENCE, AZARBAIJAN SHAHID MADANI UNIVERSITY, TABRIZ 53751 71379, IRAN.

Email address: amirbaghban87.com