Elementary Animated Illustrations of Geodesics using POV-Ray

Michael R. Gallis
Penn State University, Capital College,
200 University Drive,
Schuylkill Haven, Pennsylvania 17972

June 4, 2003

Abstract

This note is a description of the development of a simple and mathematically accurate animation depicting geodesics on curved surfaces. The project was motivated by my desire to give students in a conceptual astronomy class a clear view of the nature and effect of curved space as part of a discussion of general relativity and cosmology.

1 Introduction

The concept of geodesics and space-time curvature are part of any discussion of cosmology or black holes. Such mathematically advanced abstractions are generally difficult to grasp for the typical student in a conceptual physics or astronomy class. Animations can help visual learners by providing dynamic illustrations. To this end POV-Ray[1] (a freely available 3-D rendering program) was used to create a series of images used as frames for geodesic animations as part of a developing collection of physics and astronomy animations[2] developed for introductory classes.

2 Mathematics of Surfaces and Geodesics

A geodesic path in curved space is a path which takes covers a minimal distance between its endpoints. Quite often these ideas are introduced using two dimensional curved surfaces such as spheres and hyperbolic paraboloids which are imbedded in a (flat) three dimensional volume. Paths are then curves which are constrained to lie on the surfaces, and geodesics are those paths which minimize the distance subject to the constraint. We wish to show how the path of a geodesic depends upon the curvature of a surface, given the geodesics starting point and initial direction. We begin by reviewing the necessary mathematical description of geodesics and curved surfaces.

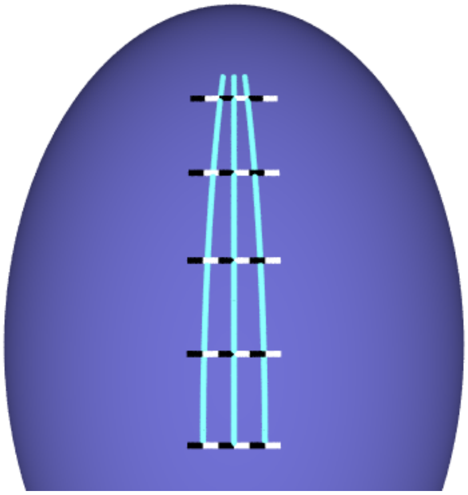

Figure 1: Geodesic on a Positively Curved Surface

Two methods of describing surfaces were useful in this project. The first method is a constraint equation on the coordinates x_1, x_2 and x_3 in three dimensions:

$$\phi(\vec{x}) = \phi(x_1, x_2, x_3) = 0. \tag{1}$$

For example, a spherical surface could be described by

$$\phi(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 - R^2 = 0.$$

The surface's (outward) normal is parallel to $-\vec{\nabla}\phi$. The second method of describing surfaces is to parameterize the the coordinates $\{x_i\}$ in terms of generalized coordinates $\{q_\mu\}$:

$$x_i = x_i(q_1, \ldots, q_\nu). \tag{2}$$

Only two generalized coordinates are for a two dimensional surface needed. For example, a spherical surface of radius R could be described by

$$\begin{aligned} x_1 &= R\cos q_1 \cos q_2 \\ x_2 &= R\sin q_1 \cos q_2 \\ x_3 &= R\sin q_2 \end{aligned}$$

The coordinates of a parameterized path (or trajectory) on the surface can be written

$$x_i = x_i(t) = x_i(q_1(t), \ldots, q_\nu(t)). \tag{3}$$

The distance covered by this trajectory is

$$S = \int ds = \int \frac{ds}{dt} dt. \tag{4}$$

An infinitesimal section of arc has length ds related to the change in the cartesian coordinates x_i by

$$ds^2 = \sum_i dx_i^2,$$

which can be written in terms of the generalized coordinates as

$$ds^2 = \sum_{i\mu\nu} \frac{\partial x_i}{\partial q_\mu} dq_\mu \frac{\partial x_i}{\partial q_\nu} dq_\nu = \sum_{\mu\nu} g_{\mu\nu} dq_\mu dq_\nu,$$

which defines the metric tensor

$$g_{\mu\nu} \equiv \sum_i \frac{\partial x_i}{\partial q_\mu} \frac{\partial x_i}{\partial q_\nu} \tag{5}$$

and so that

$$\frac{ds}{dt} = (\sum_{\mu\nu} g_{\mu\nu} \dot{q}_\mu \dot{q}_\nu)^{\frac{1}{2}} \tag{6}$$

where a dot is used to indicate time derivative.

The geodesic paths can be found by determining the generalized coordinates $q_\mu(t)$ for which Eq. (4) will be minimized. Variational methods generate the corresponding Euler-Lagrange equations:

$$\frac{\partial}{\partial q_\mu}(\frac{ds}{dt}) - \frac{d}{dt}(\frac{\partial}{\partial \dot{q}_\mu}(\frac{ds}{dt}) = 0$$

which become

$$\sum_{\alpha\beta} \frac{1}{2\frac{ds}{dt}} \frac{\partial g_{\alpha\beta}}{\partial q_\mu} \dot{q}_\alpha \dot{q}_\beta - \sum_\alpha \frac{d}{dt}(\frac{1}{\frac{ds}{dt}} g_{\alpha\mu} \dot{q}_\alpha) = 0 \tag{7}$$

Changing from dependance on t to dependance on distance s yields

$$\sum_{\alpha\beta} \frac{1}{2} \frac{\partial g_{\alpha\beta}}{\partial q_\mu} q'_\alpha q'_\beta - \sum_\alpha \frac{d}{ds}(g_{\alpha\mu} q'_\alpha) = 0. \tag{8}$$

Using using the matrix inverse of the metric as well as its symmetry properties, Eq. (8) can be written

$$q''_\nu = -\sum_{\alpha\beta} \Gamma_{\nu\alpha\beta}\, q'_\alpha q'_\beta, \tag{9}$$

where the Christoffel symbols $\Gamma_{\nu\alpha\beta}$ are given by

$$\Gamma_{\nu\alpha\beta} \equiv \frac{1}{2} \sum_\mu g^{-1}_{\nu\mu} (\frac{\partial g_{\mu\alpha}}{\partial q_\beta} + \frac{\partial g_{\mu\beta}}{\partial q_\alpha} - \frac{\partial g_{\alpha\beta}}{\partial q_\mu}) \tag{10}$$

The "velocity" of a geodesic along the arc parameterized by s in terms of the cartesian coordiantes can be obtained via the chain rule:

$$x'_i = \sum_\mu \frac{\partial x_i}{\partial q_\mu} q'_\mu \tag{11}$$

3 Illustrating Geodesics with POV-Ray

The POV-Ray ray tracing program source file scripting language allows the placement and manipulation of objects and their appearance in a virtual three dimensional space. There are some programming limitations (such as the lack of arrays) that will force some tedious coding, however the graphics tools are (of course) incredible.

Drawing the curved two dimensional surface is easy using the isosurface object; all that needs to be specified is the function $f(x, y, z)$ which specifies the surface and the bounding volume. The corresponding statement for a parabolic hyperboloid is given by:

```
#declare fns = function(xp,yp,zp) {xp*xp/(a*a)-zp*zp/(b*b)-c*yp}
```

where the parameters `a`, `b` and `c` are defined elsewhere in the code. Since the gradient of this function is parallel to the surface normal (which will be useful in some of the geometric constructs in the illustration), we estimate the components of the gradient with the function definitions

```
#declare grad_fns[1]=function(xp,yp,zp)
     {(fns(xp+delter,yp,zp)-fns(xp,yp,zp))/delter}
#declare grad_fns[2]=function(xp,yp,zp)
     {(fns(xp,yp+delter,zp)-fns(xp,yp,zp))/delter}
#declare grad_fns[3]=function(xp,yp,zp)
     {(fns(xp,yp,zp+delter)-fns(xp,yp,zp))/delter}
```

where the (small) parameter `delter` is specified elsewhere in the code. The value chosen for this particular set of figures was 10^{-6}.

The geodesics are "drawn" by a placing a sequence of cylinders end to end along the geodesic path. Determining the endpoints of the cylinders thus requires (numerically) integrating Eq. (9). Over a short segment of the path, the "acceleration" can be treated as constant, so that

$$\begin{aligned}
q_\mu(s+ds) &\approx q_\mu(s) + {q'}_\mu(s)ds + \frac{1}{2}{q''}_\mu(s)(ds)^2 \\
{q'}_\mu(s+ds) &\approx {q''}_\mu(s)ds \\
{q''}_\mu(s) &= -\sum_{\alpha\beta} \Gamma_{\nu\alpha\beta}\, q'_\alpha(s) q'_\beta(s), \qquad (12)
\end{aligned}$$

where $\{\Gamma_{\nu\alpha\beta}\}$ as functions of $\{q_\nu(s)\}$ are determined via Eq. (10).

The implementation of the mathematics has been accomplished by defining the basic description of the surface in terms of the parameterizations of the cartesian coordinates in terms of the generalized coordinates. For example, the code segment

```
#declare xv[1]=function(q1,q2) { a*q1}
#declare xv[2]=function(q1,q2) { (q1*q1-q2*q2)/c}
#declare xv[3]=function(q1,q2) { b*q2}
```

defines the x, y and z coordinates on the surface as functions of the generalized coordinates `q1` and `q2` for the particular case of the hyperbolic paraboloid (the constants `a`, `b` and `c` are specified elsewhere). The remaining implementation is completely general, so that varying the surface type only requires modifying these function definitions.

The metric is defined in Eq. (5) in terms of partial derivatives of the coordinate functions. The estimating functions are defined (while looping over `i`) as

```
#declare xvd[i][1]=function(q1,q2)
     {(xv[i](q1+delter,q2)-xv[i](q1,q2))/delter}
#declare xvd[i][2]=function(q1,q2)
     {(xv[i](q1,q2+delter)-xv[i](q1,q2))/delter}
```

which in turn allows the the definition of the metric component functions (looping over `i` and `j`) from

```
#declare g[i][j]=function(q1,q2) {
      xvd[1][i](q1,q2)*xvd[1][j](q1,q2) +
      xvd[2][i](q1,q2)*xvd[2][j](q1,q2) +
      xvd[3][i](q1,q2)*xvd[3][j](q1,q2)}
```

In order to define the Christoffel symbol functions from Eq. (10), we define the estimates of the derivatives of the metric components (looping over `i` and `j`) via

```
#declare gd[i][j][1]=function(q1,q2)
     {(g[i][j](q1+delter,q2)-g[i][j](q1,q2))/delter}
#declare gd[i][j][2]=function(q1,q2)
     {(g[i][j](q1,q2+delter)-g[i][j](q1,q2))/delter}
```

and the inverse of the metric via the general inverse of a 2 by 2 matrix:

```
#declare gdet=function(q1,q2)
     {g[1][1](q1,q2)*g[2][2](q1,q2)-g[1][2](q1,q2)*g[2][1](q1,q2)}
#declare ginv[1][1]= function(q1,q2)
     {  g[2][2](q1,q2)/gdet(q1,q2) }
#declare ginv[1][2]= function(q1,q2)
     { -g[1][2](q1,q2)/gdet(q1,q2) }
#declare ginv[2][1]= function(q1,q2)
     { -g[2][1](q1,q2)/gdet(q1,q2) }
#declare ginv[2][2]= function(q1,q2)
     {  g[1][1](q1,q2)/gdet(q1,q2) }
```

so that the definition of the Christoffel symbols becomes

```
#declare G[i][j][k]=function(q1,q2)
     {.5*(
      ginv[i][1](q1,q2)*(
            gd[1][j][k](q1,q2)
           +gd[1][k][j](q1,q2)
           -gd[j][k][1](q1,q2))
     +ginv[i][2](q1,q2)*(
            gd[2][j][k](q1,q2)
           +gd[2][k][j](q1,q2)
           -gd[j][k][2](q1,q2))
     )}
```

For a particular frame of the animation, once the surface is placed, three geodesic curves (initially parallel) are draw for a distance s (which depends upon the elapsed time in the animation) by placing cylinders with endpoints along the geodesics. The actual geodesic trajectories are obtained by adding up the results of Eq. (12) over several short subintervals:

```
//calculate all accelerations first
#local j=1;
#while(j<=2)
        #local qa[j]=-(
        G[j][1][1](qn[1],qn[2])*qv[1]*qv[1]+
        G[j][1][2](qn[1],qn[2])*qv[1]*qv[2]+
        G[j][2][1](qn[1],qn[2])*qv[2]*qv[1]+
        G[j][2][2](qn[1],qn[2])*qv[2]*qv[2]
        );
        #local j=j+1;
#end
//update coords and velocities
#local j=1;
#while(j<=2)
        #local qn[j]=qn[j]+qv[j]*dds+.5*qa[j]*dds*dds;//new coord
        #local qv[j]=qv[j]+qa[j]*dds;                 //new vel
        #local j=j+1;
#end
```

Cylinders endpoints are separated by several such subintervals, but are close enough that the rendering of the geodesics appears smoothly curved.

To aid in the perception of the geodesics as the animation progresses, several crossbars are placed tangent to the surface and perpendicular to the central geodesic. The displacement of the end of the crossbar from the central geodesic are taken in the direction of the cross product of the surface normal (taken to be in the direction of the estimated gradient in `grad_fns`) and the "velocity" of the geodesic. To aid in the visibility of the geodesics and crossbars, the cylinders are "lifted" out of the surface by a small translation in the direction of the outward surface normal. The virtual camera viewpoint is translated and rotated to keep the entire geodesic in view in each frame. For all three examples, the initial camera vantage is close in, to give the impression of a flat surface containing three apparently straight and parallel segments while the final frame

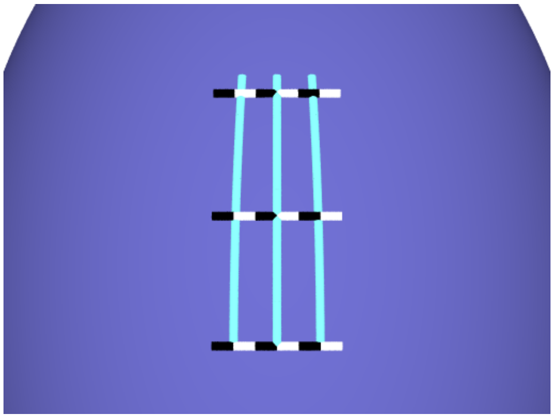

Figure 2: Typical Frame with 3 geodesics and crossbar markers

has panned out enough so that the nature of the surface can be seen. The final three animation files can be seen at

```
http://phys23p.sl.psu.edu/phys_anim/astro/geodesic_neg_curve_anim_new.avi
http://phys23p.sl.psu.edu/phys_anim/astro/geodesic_pos_curve_anim_new.avi
http://phys23p.sl.psu.edu/phys_anim/astro/geodesic_no_curve_anim_new.avi
```

4 Comments

The computational power of the modern desktop computer, coupled with freely available graphics and ray tracing programs like POV-Ray can enable teachers to create the visual aids for difficult concepts. In this note, the creation of a set of brief animations illustrating geodesics and curved space was outlined by reviewing the specific mathematical details as well as some of the POV-Ray code used to implement the animations. The resulting code could readily be adapted to other related projects (such as an illustration of parallel transport, etc) or to extend and enhance the original illustrations. The POV-Ray files can be found at

```
http://phys23p.sl.psu.edu/phys_anim/povray/geodesic.tar.gz
```

References

[1] POV-Ray http://www.povray.org/

[2] Physics and Astronomy Animations
http://phys23p.sl.psu.edu/phys_anim/Phys_anim.htm/phys_anim