

Real Time Cutter Path Generation and Control of a Hydraulically Actuated Tunneling Robot

Paul Zsombor-Murray
Mechanical Engineering
McGill University
Canada

Gerhard Rath, Paul O’Leary
Automation
University of Leoben
Austria

A two degree of freedom automated tunneling machine uses a relatively slow but extreme environment hardened digital controller (PLC) to control cutter path on a toroidal surface. A geodesic curve is the shortest distance between two points on any surface. We show how to compute a series of points, approximating those on the shortest geodesic, by projecting points, on a straight line connecting the terminal points, normally onto the surface. This entire cutter trajectory is represented with precision by a sequence of such point-to-point straight line segments of various length. Motion interpolation occurs in increments of equal displacement during an equal time interval. The conventional Bresenham straight line algorithm for plotters and graphical devices is adapted to use time as the unconditional step, i.e., independent variable cardinal axis. Thus the motion of two or more joint axes of a tunneller may be easily coordinated.

1 Introduction

1.1 The Machine

A miniature sewer tunnel re boring machine is shown in Fig 1. The spherical cutting head on the right, equipped with conical carbide teeth, rotates about the longitudinal axis of the arm that supports the head and removes a layer of predetermined thickness from the tunnel wall as the machine moves backwards, to the left. Between small steps of backward progress, the arm must be moved so the cutter sphere centre makes a complete circuit around a closed curve described by the intersection of a cylinder, offset with respect to tunnel surface by a distance equal to the sphere radius less the depth of desired cut, and the surface of a torus defined by its two characteristic radii. This torus has a horizontal “doughnut hole” axis provided by a revolute joint mounted on the machine frame. A lesser radius r separates this joint axis from a perpendicular one that moves tangentially to the torus ring centre line; a circle that remains in an essentially vertical plane. The greater, ring centre circle radius R is the distance between this second axis and the centre of the cutter head sphere.



Figure 1: Robotic Tunneller

One leg of a bell crank, just forward of the horizontal revolute joint axis, supports the tangential axis joint and a clearly

visible pair of hydraulic cylinders working in push-pull tandem. They move the cutter centre along circles that rule the torus meridially and lie in a pencil of planes on the torus centre line. These excursions extend between about $-40^\circ \geq \phi \leq 40^\circ$. The bell crank pivots on the horizontal revolute joint and is actuated by a single hydraulic cylinder in the machine base. Excursions about the horizontal axis extend between about $-60^\circ \geq \theta \leq 35^\circ$. Note that -60° represents elevation of the cutter *above* the horizontal plane in order to preserve right-handedness of the Cartesian coordinate system to be introduced later.

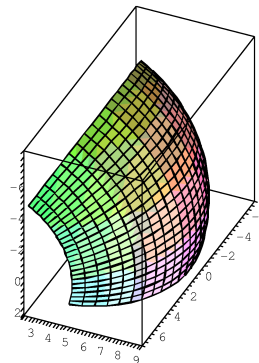


Figure 2: Torus Surface on Cutting Face

Above, one sees in Fig. 2 the cutter sphere centre working surface subject to joint limits but exaggerated to $-60^\circ \geq \phi \leq 60^\circ$ so as to show the quadrilateral grid of equal angular increments and how the subtended distances on the surface diminish towards the polar regions as the magnitude of ϕ increases. This plays an important part in the design of “straight line” trajectory motion planning. Below, Fig. 3 shows the entire exterior surface of the torus and its nearly spherical shape when $R/r = 8$. This ratio is somewhat less

than that used in designing the actual machine.

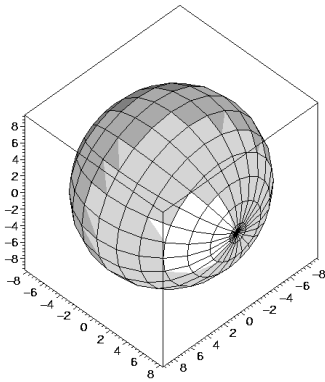


Figure 3: Entire Torus

2 The Tunnel

Large sewer tunnel cross-section shape and proportions, throughout the world, have been standardized since the mid-19th century when extensive modern sanitary drainage system installation began in England [1]. A cylindrical section is shown in Fig. 4. This is a composite of four C_1 blended circular arcs of roof radius a , two of wall radius $3a$ and with floor trough radius $a/2$. A typical cross section size, $a = 0.5\text{m}$, is inscribed in a rectangle, width and height, of $1 \times 1.5\text{m}$. Fig. 5 shows the tunnel together with a portion of the surface swept by the robot's horizontal, θ , and tangential, ϕ , axis revolute joint angle displacements.

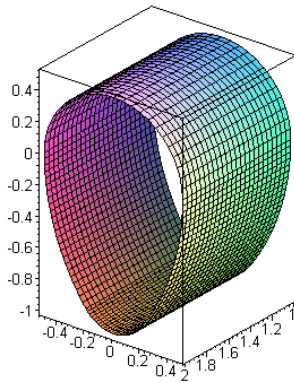


Figure 4: The Tunnel

3 Kinematics, Motion Planning and Control

Before dealing with the control of two electro-hydraulic servo-valves that manage fluid flow to and from the linear actuators, *i.e.*, cylinders, attached between frame and bell crank, thereby affecting angle θ , and between bell crank and cutter arm, thereby producing angular motion ϕ , one must consider the geometry of tunnel cylinder and cutter torus. It is upon

this that direct and inverse kinematics of cutter sphere centre point motion are based. Given the tunneller 2R joint, actuator and arm design layout and that of the tunnel, it may be presumed that motion planning and control are trivial tasks. However the requirement of "straight line" motion implies that the centre point should preferably move along minimum length geodesic arcs on the torus. To investigate these trajectories properly, it is useful to begin with more basic aspects of torus geometry.

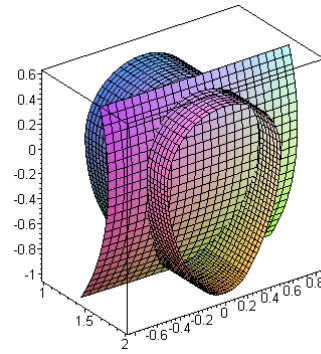


Figure 5: Tunnel and Torus; Front

3.1 Torus Geometry

Given definitive dimensions r and R , the Cartesian coordinates of any point (x, y, z) on the swept cutter centre surface generated by joint variables θ and ϕ are given by Eq. 2.

$$\begin{aligned} x &= (r + R \cos \phi) \cos \theta \\ y &= R \sin \phi \\ z &= -(r + R \cos \phi) \sin \theta \end{aligned} \quad (1)$$

Note that this torus is in standard form, with the origin at the centre, the positive x -axis pointing forward, with respect to the machine, the y -axis on the horizontal R-joint and positive z upward. Eq. 2 is the implicit equation.

$$[x^2 + y^2 + z^2 - (r^2 + R^2)]^2 - (2r)^2(R^2 - y^2) = 0 \quad (2)$$

Eq. 2 is easily obtained by recalling that the torus is a special cyclide, a member of the family of fourth order surfaces that can always be represented by the square of a sphere augmented by another surface of order three or less. Using homogeneous coordinates, a cyclide is given by Eq. 3.

$$\begin{aligned} (-\rho^2 w^2 + x^2 + y^2 + z^2)^2 + \sum k_{ijkl} w^i x^j y^k z^l &= 0 \quad (3) \\ w &= 1, \quad i, j, k, l = 0, 1, 2, 3, \quad i + j + k + l = 3 \end{aligned}$$

Squaring Eqs. 2 and adding these produces

$$\begin{aligned} x^2 + y^2 + z^2 &= (r + R \cos \phi)^2 \cos^2 \theta + R^2 \sin^2 \theta \\ &\quad + (r + R \cos \phi)^2 \sin^2 \theta \end{aligned}$$

Then by collecting $(r + R \cos \phi)^2$ and R^2 and simplifying terms like $\cos^2 \alpha + \sin^2 \alpha = 1$ one gets

$$x^2 + y^2 + z^2 = r^2 + 2rR \cos \phi + R^2$$

Using the second of Eq. 2 to eliminate $\cos \phi$ with $y = R\sqrt{1 - \cos^2 \phi}$ and squaring to eliminate the root furnishes Eq. 2.

3.2 Direct and Inverse Kinematics

3.2.1 Direct Position

Given the two joint angles θ and ϕ the spherical cutter centre location is immediately available in Cartesian coordinates as expressed by Eq. 2. This *direct kinematic* information is required as the controller makes hydraulic fluid admission adjustments in the course of following some desired cutter trajectory. It is also required to specify a starting point, subsequently always referred to as P , of some approximation to, say, a desired straight line cutter path to a given destination Q .

3.2.2 Inverse Pose

A destination point, always referred to subsequently as Q , is provided by a point on a ruling line of one of the cylindrical tunnel wall regions. That line may be defined in terms of a segment parameterized on two known points A and B . Expressing A , B and Q as position vectors \mathbf{a} , \mathbf{b} and \mathbf{q} , *i.e.*, $\mathbf{q} = \mathbf{a} + t(\mathbf{b} - \mathbf{a})$. One finds Q by solving Eq. 2 in terms of t and back-substituting into the vector equation. The set of constraints are given in Eq. 4.

$$(q_1^2 + q_2^2 + q_3^2 - r^2 - R^2)^2 - 4r^2(R^2 - q_2^2) = 0$$

$$q_i = a_i + (b_i - a_i)t, \quad i = 1, 2, 3 \quad (4)$$

The solution is a quartic. That is inevitable because a line intersects a torus on four points. The point to be taken as Q is, because of the choice of torus frame that is identical to that of the tunneller, the one with greatest positive magnitude of the coordinate $x \equiv q_1$. For computation, the univariate quartic in t is given, together with its coefficients, in Eq. 5.

$$T_4 t^4 + T_3 t^3 + T_2 t^2 + T_1 t + T_0 = 0 \quad (5)$$

$$t_4 = (b_1 - a_1)^2 + (b_2 - a_2)^2 + (b_3 - a_3)^2$$

$$t_3 = a_1(b_1 - a_1) + a_2(b_2 - a_2) + a_3(b_3 - a_3)$$

$$t_2 = a_1^2 + a_2^2 + a_3^2 - r^2 - R^2$$

$$T_4 = t_4^2$$

$$T_3 = 4t_3 t_4$$

$$T_2 = 4r^2(b_2 - a_2)^2 + 2t_2 t_4 + 2t_3^2$$

$$T_1 = 8r^2 a_2 (b_2 - a_2) + 4t_2 t_3$$

$$T_0 = t_2 - 4r^2(R^2 - a_2^2)$$

3.3 Straight Lines

Since there are no straight lines on a torus these must be approximated by trajectories on the curved surface. Ideally one might wish to invoke the shortest path between P and Q , *i.e.*, one of ∞^1 of such paths radiating in every direction from any point P . A number of these pass through Q . To further complicate matters the computation requires

the iterative solution of a simultaneous pair of nonlinear ordinary differential equations. There is considerable literature [2, 3, 4] and websites providing good description and software for computation [5, 6, 7] available that treats the subject of geodesics quite thoroughly. Although calculating the shortest path between two points on a torus can be done with great precision using commercial codes it is nevertheless by means of a hit a miss algorithmic process. That is eminently unsuited to real time computation; the task at hand. Therefore what is proposed here is to follow a path such that the discrete points along it, the ones that are to be computed as trajectory landmarks for the two actuator controllers, are points on the surface *closest* to the unattainable straight line joining P and Q . We begin with spheres centred on a given points M between P and Q and tangent to the torus and continue with a parameterization on PQ to produce M at intervals corresponding to a fairly equal separation of successive points N on the surface. Then there will be a comparison between the proposed approximation scheme an easy-to-obtain but not entirely trivial geodesic on a developable surface, *i.e.*, a cone of revolution. Finally the hydraulic fluid admission control will be described in terms of pseudo-finite steps, the time based Bresenham observer.

3.3.1 Tangent Spheres to the Torus

Before selecting any parameterization scheme to move point M along the line segment PQ consider the segment MN that is normal to, with point N on, the surface. It may be seen as the radius of a tangent sphere centred on M and touching the torus. The problem is defined by the following constraints.

$$f(n_i) = (n_1^2 + n_2^2 + n_3^2 - r^2 - R^2)^2 - 4r^2(R^2 - n_2^2) = 0, \quad i = 1, 2, 3 \quad (6)$$

This is just the torus equation again with the coordinates of N substituted as unknowns. Next the surface normal direction numbers are determined.

$$\frac{\partial f}{\partial n_1} = (n_1^2 + n_2^2 + n_3^2 - r^2 - R^2)n_1$$

$$\frac{\partial f}{\partial n_2} = (n_1^2 + n_2^2 + n_3^2 + r^2 - R^2)n_2$$

$$\frac{\partial f}{\partial n_3} = (n_1^2 + n_2^2 + n_3^2 - r^2 - R^2)n_3 \quad (7)$$

These are put in two equations, Eq. 8 and 9, that express the proportionality between these direction numbers and the difference in the point position vectors of M and N , *i.e.*, $\mathbf{n} - \mathbf{m}$. This procedure is well documented in any advanced calculus text [8].

$$(n_1^2 + n_2^2 + n_3^2 + r^2 - R^2)(n_3 - m_3)n_2 - (n_1^2 + n_2^2 + n_3^2 - r^2 - R^2)(n_2 - m_2)n_3 = 0 \quad (8)$$

$$(n_1^2 + n_2^2 + n_3^2 + r^2 - R^2)(n_1 - m_1)n_2 - (n_1^2 + n_2^2 + n_3^2 - r^2 - R^2)(n_2 - m_2)n_1 = 0 \quad (9)$$

Eliminating n_2 and n_3 among Eqs. 6, 8 and 9 produces a univariate in n_1 of degree 64. This is not nearly so hopeless as it may appear at first glance. The equation factors into a triple product each element of which is only quartic. Furthermore only one of these represents a valid solution because it contains all three coordinates of the sphere centre. The other two express only m_1 and m_2 . Still more appetizing is absence of the cubic coefficient in the valid factor. So the solution appears as Eq. 10.

$$N_4 n_1^4 + N_3 n_1^3 + N_2 n_1^2 + N_1 n_1 + N_0 = 0 \quad (10)$$

The four coefficients are recorded below to show that, although lengthy, they are still computationally manageable.

$$\begin{aligned} N_4 &= m_1^4(m_1^2 + m_3^2)\nu_4 & (11) \\ \nu_4 &= r^4 + (m_3^2 - 2r^2)m_3^2 \\ &+ [2(m_3^2 + r^2) + m_2^2]m_2^2 + \\ &+ [2(m_3^2 + m_3^2 - r^2) + m_1^2]m_1^2 \\ N_3 &= 0 \\ N_2 &= -2m_1^6(m_1^2 + m_3^2)\nu_2 \\ \nu_2 &= (r^2 + R^2)[r^4 + (m_3^2 - 2r^2)m_3^2] \\ &+ [r^2m_2^2 + (2r^2 + R^2)(r^2 + m_3^2)]m_2^2 \\ &+ [(r^2 + R^2)(2m_3^2 + m_1^2 - 2r^2) + (2r^2 + R^2)m_2^2]m_1^2 \\ N_1 &= 8r^2R^2(m_1^2 + m_3^2)m_1^7m_2^2 \\ N_0 &= m_1^8\nu_0 \\ \nu_0 &= (r^2 - R^2)(r^2 + R^2)[r^4 + (m_3^2 - 2r^2)m_3^2]m_3^2 \\ &+ r^2[2(r^2 - R^2)(r^2 + m_3^2) + r^2m_2^2]m_2^2 + (r^2 - R^2) \cdot \\ &\cdot [(r^2 + R^2)(-2r^2 + 2m_3^2 + m_1^2) + 2r^2m_2^2]m_1^2 \end{aligned}$$

3.3.2 Advancing the Cutting Head

Using the direct and inverse kinematics relationship one extracts successive points N by, say, parameterizing in equal increments from P to Q along the straight line. If the predicted volume removal is too great or small the parameter step is adjusted appropriately in a single change. This volume is estimated as follows. The tunneller has a spherical cutting head. In any time increment its advance is dictated by tunnel surface properties and the power available to rotate the head. The following analysis is meant to correlate the volume of material removed during an interval during which the centre of a sphere of radius R advances a distance c . The situation is illustrated in Fig. 6. The volume so removed is $V_C + V_R$, the combination of the “bowl” of crescent shaped cross section and the “vee” cusp ring that fills the circumferential groove at the intersection of the two images of the sphere whose centres are separated by c . The outer surface of this region is bounded by the cylinder tangent to both images. But the cutter does not normally excavate a path of cross section radius R . Rather it leaves behind a groove of depth g . Therefore the actual volume removed is

$$V = 2 \cos^{-1} \left(\frac{R-g}{R} \right) (V_C + V_R)$$

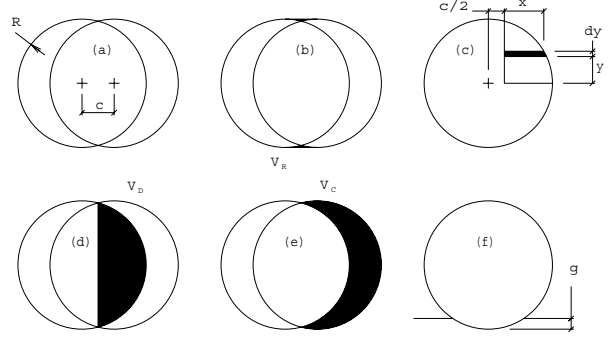


Figure 6: Cutter Groove Geometry

To get volume V the first step is to compute the first moment of half the “D” shaped area, shown in inset (d) of Fig. 6 by means of the integration scheme illustrated in inset (c). This is, according to Pappus’ second theorem, multiplied by 2π to obtain spherical volume V_D .

$$\begin{aligned} V_D &= 2\pi \int_0^{\sqrt{R^2 - c^2/4}} y \left(\sqrt{R^2 - y^2} - \frac{c}{2} \right) dy \\ &= 2\pi \left[R^2 \left(\frac{4R - 3c}{12} \right) + \frac{c^3}{48} \right] \quad (12) \end{aligned}$$

The “bowl” part, V_C , of the excavated volume is $V_C = V_S - 2V_D$, where $V_S = 4\pi R^3/3$, the sphere volume, while the “ring” part, V_R , is $V_R = V_L + 2V_D - V_S$, where $V_L = \pi R^2 c$, the volume of a cylinder of radius R and height c . Although it may seem counter intuitive $V_L = V_C + V_R$!

4 Motion Control

Using adjacent points N the joint angle increments $\Delta\theta$ and $\Delta\phi$ are computed. The actual values θ and ϕ are measured with rotation encoders and fed into software-implemented position controllers, which adjust the control voltages for two hydraulic servo-valves in order to minimise the deviations to setvalues θ_s and ϕ_s [10].

Path control is done by feeding the independent controllers with a sequence setvalue pairs in time. The time intervals have to be short enough not to excite resonances of the mechanical system. As described previously, the path is divided into straight line segments. The task of the real-time computer system is to interpolate the points on these lines.

4.1 Bresenham’s Line Interpolation

Linear interpolation means to solve the line equation $y = kx + d$ while incrementing x . Simple, but robust microcontrollers still today need much more time for floating point than for integer calculation. An algorithm called DDA (Digital Differential Analyzer) [12] avoids the dot operation by consecutively adding a value to y while incrementing x . With this, the problem of cumulative rounding errors occurs.

Bresenham's algorithm [11] does not run into this problem. Here the integer and the fractional part of the addition are treated separately. Scaling the fractional part properly to an integer number avoids rounding errors without the need of floating point arithmetics. When x is the major axis, the slope k of the line always will be less than or equal to 1. When the addition of k produces a carry, this is added to the integer and subtracted from the fractional part. Since with the conventional Bresenham algorithm, the increment has to be done always with the major axis, several further developments were done, for example the MAI (Multiple Axis Interpolation) [13], which is applied in graphical display devices.

4.2 Multiple Axis Interpolation

Coordinating more than two axes with the MAI algorithm must handle slopes greater than 1 [13]. If the slope is unrestricted, the addition sample has an integer and a fractional part. This implies the need of one division and one modulo function to chop the result into its integer and fractional portions again, which is a disadvantage at the first glance. But a fast integer division is in the command set of most modern controllers, and the modulo is the remainder of the division obtained by a simple subtraction.

4.3 Multiple Axis Interpolation in Time

While in computer graphics this algorithm is used to implement position and colour information of a line, for the tunnelling machine the time was chosen as cardinal axis, and the setvalues θ_s and ϕ_s are the axes to be coordinated. The following code in MATLAB¹ shows how the interpolation works for one axis.

```
%motion parameters:
t1 = 0; t2 = 10; y1 = 5; y2 = 30;
%system parameters:
T = 1; %sampling time
%algorithm parameters:
emax = 0.5;
cstep = 1; %correction step
%some MATLAB stuff:
t_array = zeros(1000,1);
y_array = zeros(1000,1);
%preparation phase for the motion:
j = t2 - t1; k = y2 - y1;
nx = k/j;
nxi = floor (nx); % simulate int division
% command of modern CPUs
e = 0; y = y1; n = 1;
for t=t1: T: t2,
    if e > emax
        y = y + cstep;
        e = e - cstep;
    end
    t_array(n) = t;
    y_array(n) = y;
    y = y + nxi;
    e = e + nx - nxi;
```

¹Trademark of the Mathworks company

```
n = n + 1;
end
%the following plots the results:
plot (t_array(1:n-1),y_array(1:n-1),'o');
n = 1;
for t=t1: T: t2,
    rectangle('Position',
        [t-T/2 y_array(n)-nxi/2 T nxi])
    y_ref(n) = y1 + t*nx; %ideal line
    n = n+1;
end
hold on;
plot (t_array(1:n-1),y_ref(1:n-1));
grid on;
xlabel ('t [s]'); ylabel ('y'); hold off;
axis ([0 t2+1 0 y2+5]);
```

A simple completion of some "if" constructs enables the algorithm to work in all four quadrants. For every axis one such piece of code is necessary.

4.4 Multitasking

Generating setvalue pairs on a straight line works numerically efficient on microcontrollers with the Bresenham-based method. The other tasks, as the line segmentation or the geodesic computation, seem not to fit into this concept. But modern processing units provide time-sharing multitasking facilities that allow to solve this problem with a configuration shown in Fig. 7. While the line interpolation is running in a fast, high-priority schedule, the data for the following line segment can be prepared in a low-priority task. A certain amount of complex mathematical function as square-roots and floating-point operations is possible, since the mechanical system needs some time to reach the end of the present line segment.

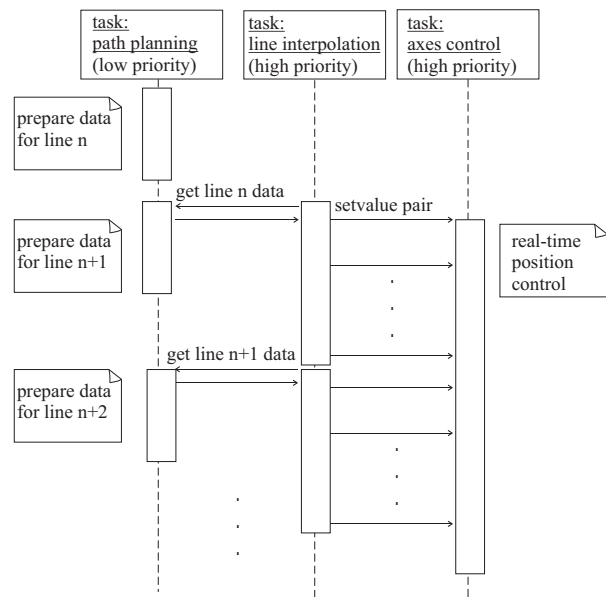


Figure 7: Path generation and control sequence

5 Summary and Conclusions

A novel scheme for the approximation of a geodesic that is the shortest distance on a surface is shown by projecting points on a straight line, which is connecting the terminal points, normally onto the surface. For a toroidal surface a computational efficient implementation is demonstrated for a small tunnelling machine. Furthermore, the geodesic path is broken into straight line segments that are interpolated by a modified Bresenham algorithm using time as cardinal axis to accomplish the coordination of the two machine axes, or even more, in real-time.

References

- [1] Moore, K. (2005) "Buried Treasure", *IMechE Professional Engineer*, v.181, p.44.
- [2] Lipschultz, M. (1969) *Differential Geometry*, ISBN 0-07-037985-8, McGraw-Hill.
- [3] Wrede, R.C. (1972) *Introduction to Vector and Tensor Analysis*, ISBN 0-486-61879-X, Dover.
- [4] Guggenheimer, H.W. (1977) *Differential Geometry*, ISBN 0-486-63433-7, Dover.
- [5] Anderson, W.L. (2005) *Geodesic Computation*, <<http://www.elesoft.com/geodcomp/>>.
- [6] Irons, M.L. (2004) *The Curvature and Geodesics of the Torus*, <<http://www.rdrop.com/half/math/torus/>>.
- [7] Knill, A. (1999) *Geodesics*, <<http://abel.math.harvard.edu/archive/118r-spring-05/handouts/geodesics.pdf>>.
- [8] Spiegel, M.R. (1994) *Advanced Calculus*, ISBN 07-060229-8, McGraw-Hill, p.161.
- [9] Newman, W.M. and Sproull, R.F., (1979) *Principles of Interactive Computer Graphics*, ISBN 0-07-046638-7, McGraw-Hill, pp.25-27.
- [10] Rath, G. (2003) *Dynamics of High-Precision Position Controlled Hydraulic Cylinders Subjected to Elastic Load*, Proceedings of The Eighth Scandinavian International Conference on Fluid Power (SICFP'03), Tampere, Finland, pp.383-396.
- [11] Bresenham, J. E. (1965) *Algorithm for Computer Control of a Digital Plotter*, IBM Systems Journal 4(1), pp.25-30.
- [12] Watt, A. (1989) *Fundamentals of Three-Dimensional Computer Graphics*, Addison-Wesley.
- [13] Swanson, R.W. and Thayer, L.J. (1986) *A Fast Shaded-Polygon Renderer*, Siggraph '86, Vol.20, Number4, pp.95-101.

Appendix

Introduction

In order to compare the result, and therefore the suitability, of the tangent sphere approximation to the shortest path length geodesic arc between two given points, P and Q , on various surfaces of revolution, consider a cone of revolution. This surface is developable so points along a true geodesic path may be easily computed. The proposed approximating method involves stepping along the straight line chord connecting P and Q . At each point M along the chord a perpendicular to the surface is constructed to yield points N , the normally projected images of M . Let us examine a method to find the sequence of locations of M such that the stepping parameter u , the fraction of the length between P and Q , as it appears on the development, is taken in steps of equal length increment. This will yield successive points N at close to equal distance measured along the surface. Choosing a step size Δt of constant length along the line from P to Q obviously will not produce equally spaced points N .

Equations

We begin with the constraint equations for tangent spheres, their solution and a numerical example, then go on to find a sequence of locations for M at equal spacing along the surface

Standard Cone and Normal

Consider the following implicit equations.

$$n_1^2 + n_2^2 - kn_3^2 = 0 \quad (13)$$

$$n_2(n_1 - m_1) - n_1(n_2 - m_2) = 0 \quad (14)$$

$$kn_3(n_1 - m_1) + n_1(n_3 - m_3) = 0 \quad (15)$$

The first, Eq. 13, represents a cone of revolution, apex on the origin and axis vertical. The point $M(m_1, m_2, m_3)$ is given and point $N(n_1, n_2, n_3)$ is the point on the surface that we seek. Subscripts 1, 2, 3 refer to directions x, y, z . The parameter k is $k = \tan^2 \frac{\alpha}{2}$ where α is the cone apex half angle. Eq. 14 and Eq. 15 express the proportionality between the direction numbers of line segment MN that is perpendicular to the surface and the respective partial derivatives of Eq. 13 with respect to $n_i, i = 1, 2, 3$.

Resultants and a Univariate in n_1

Eliminating n_2 and n_3 among the three constraint equations produces a quadratic univariate, Eq. 16, in n_1 after a common factor n_1^2 has been removed.

$$(k+1)^2(m_1^2 + m_2^2)n_1^2 - 2k(k+1)m_1(m_1^2 + m_2^2)n_1 + km_1^2[k(m_1^2 + m_2^2) - m_3^2] = 0 \quad (16)$$

The other two coordinates of N are available through linear back substitution into Eq.14 and Eq. 15

$$n_2 = \frac{m_2 n_1}{m_1}, \quad n_3 = \frac{m_3 n_1}{k(n_1 - m_1) + n_1}$$

The cone and two tangent spheres, given $k = 1/4$ and $M(-\sqrt{2}, -3\sqrt{2}, -12)$, are shown in Fig. 8. Centre, points

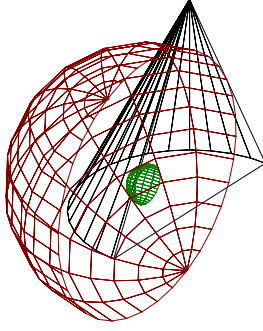


Figure 8: Concentric Spheres Tangent to Cone of Revolution

of tangency and radii for this example are given below, as computed.

$$M(-1.414213562, -4.242640687, -12)$$

$$N(-1.800735989, -5.402207965, -11.38885438),$$

$$r = 1.366563$$

$$N(1.23505056, 3.705151691, -7.811145618),$$

$$r = 8.39997$$

Straight Line on Developed Cone

Referring to Fig. 9 one sees a cone of revolution, apex on the origin and the centre of its base truncation plane, a circle of radius R , with the apex at a height of H above it. Some

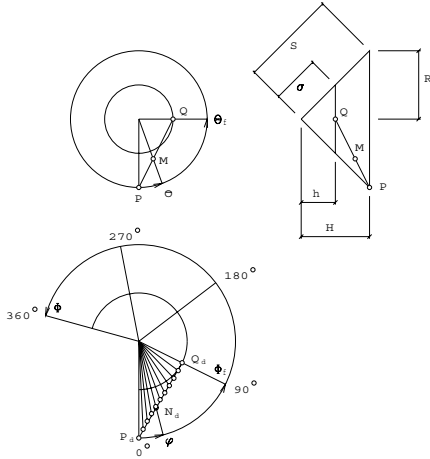


Figure 9: Cone Development

useful dimensions, derived from these two, are summarized below.

$$\tan \alpha = R/H, \quad S = \sqrt{H^2 + R^2}, \quad \sigma = hS/H, \quad \Phi = 2\pi R/S$$

Rather than follow a course of parameterization on the straight line PQ in space, the line segment P_dQ_d in the development

will be divided into equal increments, *i.e.*, parameterized on u . It is seen that

$$P_d(S, 0), \quad Q_d(\sigma \cos \Phi_f, \sigma \sin \Phi_f), \quad \mathbf{n}_d = \mathbf{p}_d + u(\mathbf{q}_d - \mathbf{p}_d)$$

so the point N_d on the geodesic is given in terms of P_d , Q_d and the parameter u by the expanded vector equation Eq.17. Subscript f means “some fraction of”.

$$\begin{bmatrix} d_{d1} \\ n_{d2} \end{bmatrix} = \begin{bmatrix} S \\ 0 \end{bmatrix} + u \begin{bmatrix} \sigma \cos \Phi_f - S \\ \sigma \sin \Phi_f \end{bmatrix} = \quad (17)$$

$$\begin{bmatrix} \sqrt{n_{d1}^2 + n_{d2}^2} \cos \Phi \\ \sqrt{n_{d1}^2 + n_{d2}^2} \sin \Phi \end{bmatrix}$$

Noting that ϕ is the angle parameter defined on $N_d(u)$ and dividing n_{d2}/n_{d1} yields an expression in $\tan \phi$.

$$\tan \phi = \frac{(\sigma \sin \Phi_f)u}{S + (\sigma \cos \Phi_f - S)u}$$

So now the angle ϕ can be defined by Eq. 18 and with it the angle parameter θ that establishes the vertical cutting plane intersecting PQ at M .

$$\theta = \frac{\Theta_f}{\Phi_f} \phi = \frac{2\pi}{2\pi R/S} = S/R$$

$$\theta = \frac{S}{R} \phi = \frac{S}{R} \tan^{-1} \left\{ \frac{(\sigma \sin \Phi_f)u}{S + (\sigma \cos \Phi_f - S)u} \right\} \quad (18)$$

Cutting Planes on the Cone Axis

One may obtain points at equal intervals on the cone surface via the distances $\sqrt{n_{d1}^2 + n_{d2}^2}$ computed at equal increments of u , Δu , and transferring these to the cone surface via Eq. 18. To compare the precision obtained with the proposed approximating method the line segment must be cut by a fan of planes m on the cone axis, parameterized as $\theta = \theta(u)$, *i.e.*, if the approximation were perfect an M -centred sphere's tangent point N would coincide exactly with the one obtained. As it stands, this is not the case. This pencil of planes is normal to the surface but not to line $\mathcal{M} \equiv PQ$. Let us examine the intersection

$$M = \mathcal{M}_r \cap m, \quad m_i = \sum_{j=0}^3 m_{ij} M_j$$

where we have the following homogeneous coordinates of points M, P, Q , plane m and radial line \mathcal{M}_r . The latter is given in terms of Plücker coordinates.

$$\mathcal{M}\{m_0 : m_1 : m_2 : m_3\}, \quad m\{M_0 : M_1 : M_2 : M_3\} \equiv$$

$$\{0 : -\sin \theta : \cos \theta : 0\},$$

$$P\{p_0 : p_1 : p_2 : p_3\} \equiv \{1 : R : 0 : -H\},$$

$$Q\{q_0 : q_1 : q_2 : q_3\} \equiv \{1 : 0 : hR/H : -h\},$$

$$\mathcal{M}\{m_{01} : m_{02} : m_{03} : m_{23} : m_{31} : m_{12}\}$$

$$\equiv \{p_0q_1 - p_1q_0 : p_0q_2 - p_2q_0 : p_0q_3 - p_3q_0 :$$

$$p_2q_3 - p_3q_2 : p_3q_1 - p_1q_3 : p_1q_2 - p_2q_1\};$$

The point coordinates m_i are obtained by expanding the summation as Eq. 19, subject to the conditions $m_{ij} = 0$ where $i = j$ and $m_{ij} = -m_{ji}$.

$$\begin{aligned} m_0 &= m_{00}M_0 + m_{01}M_1 + m_{02}M_2 + m_{03}M_3 \\ m_1 &= -m_{01}M_0 + m_{11}M_1 + m_{12}M_2 - m_{31}M_3 \\ m_2 &= -m_{02}M_0 - m_{12}M_1 + m_{22}M_2 + m_{23}M_3 \\ m_3 &= -m_{03}M_0 + m_{31}M_1 + m_{23}M_2 - m_{33}M_3 \end{aligned} \quad (19)$$

After making appropriate substitutions and simplification and dividing by m_0 we arrive at the ordinary Cartesian coordinates of M .

$$M \left(\frac{hR}{H \tan \theta + h}, \frac{hR \tan \theta}{H \tan \theta + h}, \frac{-hH(\tan \theta + 1)}{H \tan \theta + h} \right)$$

Computation and Comparison

The data points that appear in Fig. 9 as small circles, almost on the straight line representing the ideal geodesic, are shown to be quite good approximations. The computational procedure for ten steps in u is coded below.

```

100 INPUT R,HH,H,F,J:PI=3.141592654:
      SS=SQR(HH^2+r*r)
110 LPRINT:LPRINT:LPRINT R,HH,H,F
120 S=H*SS/HH:K=(R/HH)^2:PH=2*PI/SS:PF=PH*F
130 FOR I=0 TO 10
140 U=I/J:ND1=SS+(S*COS(PF)-SS)*U:
      ND2=S*SIN(PF)*U
150 PHI=ATN(ND2/ND1):TH=SS*PHI/R:
      TTH=TAN(TH):D=HH*TTH+H
160 NN=SQR(ND1^2+ND2^2):M1=H*R/D:
      M2=H*R*TTH/D:M3=-H*HH*(TTH+1)/D
170 K1=K+1:M12=M1^2+M2^2:A=K1^2*M12:
      B=-2*K*K1*M1*M12
180 C=K*M1^2*(K*M12-M3^2):DS=B*B-4*A*C:
      A2=A*A:IF DS<0 THEN STOP
190 DQ=SQR(DS):N1P=(-B+DQ)/A2:N1M=(-B-DQ)/A2:
      M21=M2/M1
200 N2P=M21*N1P:N2M=M21*N1M:N3P=M3*N1P/
      (K*(N1P-M1)+N1P)
210 N3M=M3*N1M/(K*(N1M-M1)+N1M):
      LN=SQR(N1P^2+N2P^2+N3P^2)
220 M=(M1-N1M)^2+(M2-N2M)^2+(M3-N3M)^2:
      N=(M1-N1P)^2+(M2-N2P)^2+(M3-N3P)^2
230 IF M<N THEN LN=SQR(N1M^2+N2M^2+N3M^2)
240 LPRINT U,180*PHI/PI, LN, NN, NN-LN
250 NEXT I

```

The following results were obtained with $R = H = 1$, $h = 0.5$ and $f = 0.25$. Successive values, of u , ϕ and the radii from the development apex to the geodesic line and to the approximating data points, respectively, and the difference between these latter two, appear in columns in the tabulation below.

0.0	0.0	1.4142	1.4142	0.0
0.1	2.7813	1.2875	1.3057	0.0183
0.2	6.0572	1.1748	1.2009	0.0261
0.3	9.9442	1.0747	1.1007	0.0259
0.4	14.5832	0.9863	1.0065	0.0204
0.5	20.1338	0.9090	0.9203	0.0113
0.6	26.7542	0.8427	0.8445	0.0018
0.7	34.5525	0.7878	0.7820	-0.0057
0.8	43.5058	0.7454	0.7363	-0.0091
0.9	53.3672	0.7174	0.7106	-0.0069
1.0	63.6396	0.7071	0.7071	0.0