COMPUTING MONGE SURFACES AND WOOD STRIP MODELS

DAVID BRANDER AND JENS GRAVESEN

A *Monge surface* ([3], §XXV), is a generalization of a surface of revolution, if we replace the circle that describes the rotation about the axis with an arbitrary curve. It is generated by a planar *profile curve* α, and a 3D *spine curve* $\mathbf{r}$. The plane is swept through space perpendicularly to the spine curve, and the movement of the profile curve sweeps out a surface (which may have singularities and self-intersections).

This is a quick guide on how to use the Matlab function *monge.m* to compute the Monge surface generated by a given spine curve and profile curve, as well as another function *plotstrips.m* to plot a wood-strip model of the surface. This code was used to produce the images in the Bridges 2017 article [2]. The function *monge.m* uses a fast algorithm from [4] to compute an approximation of the rotation minimizing frame along the spine curve. The Monge surface is then found by moving the plane of the profile curve with this frame. Both functions can be downloaded at http://geometry.compute.dtu.dk/software.

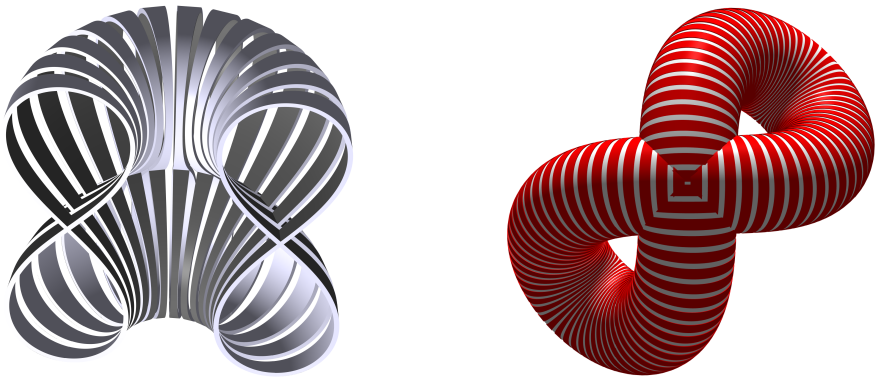

FIGURE 1. The Monge surface on the left is (half of) a surface of revolution: the spine curve is a circle and the profile curve is a planar elastica figure 8 curve. On the other hand, the surface on the right is generated by a spine curve that is a Viviani figure-8 curve, (the intersection of a cylinder and a sphere), while the profile curve is a circle.

Computing a Monge surface: The solution can be computed with the command

[X, Y, Z]= monge(alpha, r, th0, xx, yy),

where alpha(t) = @(t) [x(t); y(t)] and r(t) = @(t)[a(t); b(t); c(t)] are a pair of vector-valued function handles describing the profile and spine curves respectively, and xx= [x1, x2, ..., xn] and yy=[y1, y2, ..., ym] are row vectors describing the parameter values for α and $\mathbf{r}$ respectively, and *th0* is a rotation in the initial plane of the profile curve. If $th0 = 0$, then the x-axis of the initial plane will be directed along the normal to the spine curve $\mathbf{r}$ curve at the initial point. If the acceleration of the spine curve curve at the initial point is zero, so that there is no normal, then $th0 = 0$ corresponds to a direction in the plane orthogonal to the spine curve that is chosen automatically selecting, from the three directions $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$, that which is

Date: April 25, 2017.

closest to being orthogonal to the tangent to **r**, then projecting to the orthogonal plane. The commands:

```
[X,Y,Z]=monge(@(t)[t;t.^2], @(t)[cos(t);sin(t);t],0,linspace(-1,1,100),linspace(0,2*pi,400));
[X,Y,Z]=monge(@(t)[t;t.^2], @(t)[cos(t);sin(t);t],pi,linspace(-1,1,100),linspace(0,2*pi,400));
```

produce the first two images in Figure 2. Note that we used *t.^2* for t^2, and we would similarly use a dot on any operation with functions of t, e.g. t.*cos(t). This is because the function handles will be evaluated on vectors of points, not on scalars.

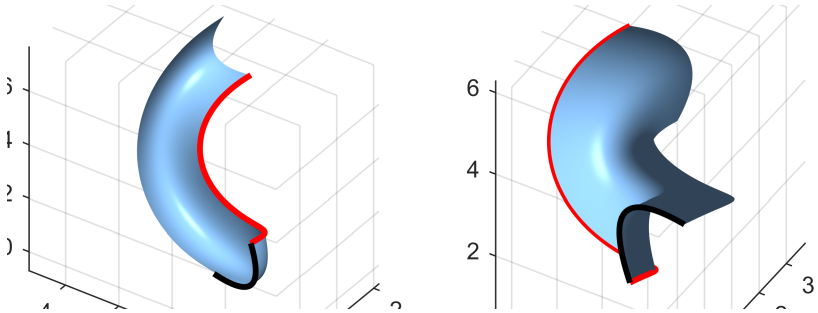

FIGURE 2.

If the profile curve and/or spine curve are only available as data points, then the solution can still be computed from this input. For example, discrete data for α and **r** in the first example are:

```
t1=linspace(-1, 1, 100);     alpha=[t1; t1.^2];
t2=linspace(0, 2*pi, 400);     r=[cos(t2); sin(t2);t2];
```

and the solution can be computed with:

```
[X,Y,Z]=monge(alpha, r,0);
```

Since the 2nd derivative is estimated from the discrete values at the initial point, and the first derivative along the entire curve, the finer the mesh for **r**, the more accurate will be the solution. (In this implementation, the function handle input will give more accurate results than the discrete input method). If many points are used, the image produced should be indistinguishable from the true solution.

More options: *ProfilePos* (sixth parameter, default 1) specifies which point on the profile curve is positioned at the initial point on the spine curve. *RPfactor* (7th parameter, default 10) controls a reparameterization of the spine curve. This makes the strips in the wood-strip plots described below about the same width. To turn this option off, add 0 as a 7th parameter. Example:

```
[X,Y,Z]=monge(@(t)[t;t.^2], @(t)[cos(t);sin(t);t],0,linspace(-1,1,100),linspace(0,2*pi,400), 1, 0);
```

Plotting a wood-strip model: The command

```
plotstrips(X,Y,Z,offset,width,numstrips,color1,color2,style,color3);
```

will plot a series of (thickened) ruled strips along a subset of the isocurves $v = constant$ (if v is the parameter in the sweeping direction). On a Monge surface, these curves are planar curves congruent to the profile curve α, and are geodesics in the surface. The thickness of the strip is determined by *offset*, in the same units as the geometry of the surface. The width of each strip is determined by *width*, which should be between 0 (no wood) and 1 (no gaps) . The number of strips is *numstrips*, and the function works best if the number of v grid points is large compared with the number of strips (e.g. 800 pts to 80 strips). A larger number of v points also makes the computation of the surface more accurate.

The other parameters are optional: the default is: color1=[1 1 1], color2=[0.6 0.6 0.6], style=1 and color3=color2. The parameter *style* can be 1, 2 or 3. The commands

```
plotstrips(X,Y,Z,0.06, 0.7, 40);
plotstrips(X,Y,Z,0.06, 0.7, 40, [1 1 1], [0. 0.9 0.5], 1, [1 0 0]);
plotstrips(X,Y,Z,0.06, 0.7, 40, [1 1 1], [0. 0.9 0.5], 2, [1 0 0]);
plotstrips(X,Y,Z,0.06, 0.7, 40, [1 1 1],[ 0. 0.9 0.5], 3, [1 0 0]);
```

produce the examples in Figure 3.

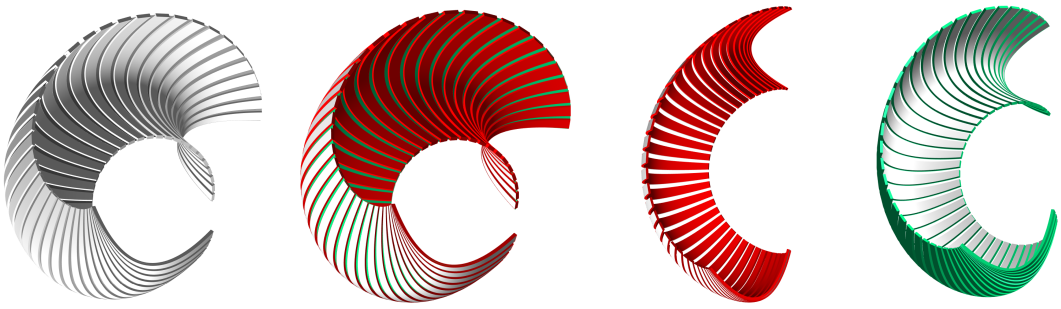

FIGURE 3. Examples of plots of the same surface using different styles. (The geometry of the surfaces are identical, but they are viewed here from slightly different angles).

Styles 1 and 2 offset the surface half the offset distance in both the positive and negative normal direction. Style 1 also plots the underlying surface, and uses all three colors, if three colors are specified.

Style 3 offsets in only one direction, which might be wanted in some cases. The difference in offsets can easily be seen by plotting a surface with a cuspidal edge singularity, such as this one:

```
[X,Y,Z]=monge(@(t)[t;0*t], @(t)[cos(t);sin(t);t],0,linspace(0,10,100),linspace(0,pi,400));
```

Note that we entered @(t)[t; 0*t] for the function handle @(t)[t; 0]. As with the dot operations, this is necessary because the function handle will operate on vectors, not scalars, and 0*t will be a row vector of the correct length. The results of plotting this surface with the following four commands:

```
plotstrips(X,Y,Z,0.06, 0.7,40,[1 1 1],[0.6 0.2 0.1],1,[0.6 0.7 1]);
plotstrips(X,Y,Z,0.06, 0.7,40,[1 1 1],[0.6 0.2 0.1],2,[0.6 0.7 1]);
plotstrips(X,Y,Z,0.06,0.7,40,[1 1 1],[0.6 0.2 0.1],3,[0.6 0.7 1]);
plotstrips(-X,-Y,-Z,0.06,0.7,40,[1 1 1],[0.6 0.2 0.1],3,[0.6 0.7 1]);
```

are shown in Figure 4. Note that we replaced X, Y , Z with -X, -Y, -Z in the last plot. This is the orientation-

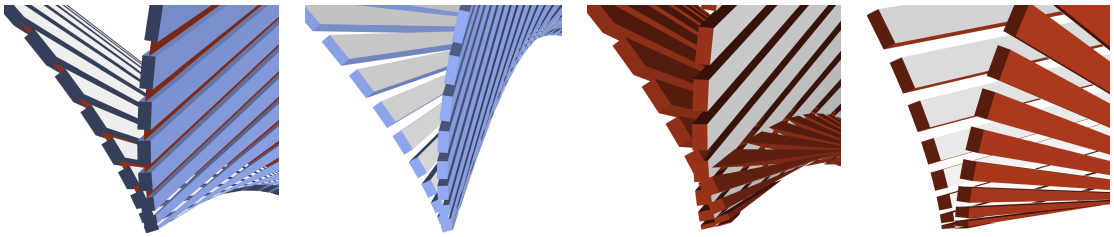

FIGURE 4.

reversing isometry of $\mathbb{R}^3$ given by $p \mapsto -p$, which has the effect of flipping the normal. The normal is

pointing "into" the surface in the third figure, and "out" in the fourth, giving a very different offset.

Closed spine curves: A monge surface is determined by a minimal rotating frame along the spine curve. If the spine curve is closed, it is necessary that the minimal rotating frame of the spine curve closes up in order that the surface closes up. In general, it is necessary and sufficient that the integral of the torsion of $\mathbf{r}$ around the curve is an integer multiple of 2π. We discuss this problem and give a method for producing spine curves with closed minimal rotating frames in [1]. However, there is a simple way to produce some examples: any closed curve that lies in a sphere has the property that the integral of its torsion is zero. Hence any such curve can be used to generate Monge cylinders. If the profile curve is also closed, we get a Monge torus. For example, the curve $\mathbf{r}(t) = (\cos(u)\cos(v), \sin(u)\cos(v), \sin(v))$ where $u(t) = 2+\cos(t)$, and $v(t) = \sin(t)$, is a closed curve in the 2-sphere. The commands:

```
u=@(t)2+cos(t); v=@(t)sin(t);
[X,Y,Z]=monge(@(t)0.25*[cos(t); sin(t)], @(t)[cos(u(t)).*cos(v(t));sin(u(t)).*cos(v(t));sin(v(t))],...
 0,linspace(0,2*pi,200),linspace(0,2*pi,800));
plotstrips(X,Y,Z,0.01,0.625,50,[1 1 1],[0.5 0.5 0.5],1);
```

produces the Monge torus shown to the right in Figure 5.

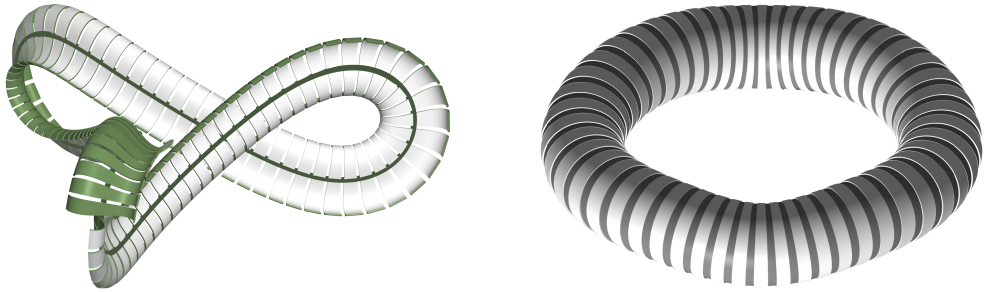

FIGURE 5. The Monge surface to the left is generated from a closed spine curve, but the minimal rotating frame is not closed on this curve. The surface on the right is generated by a closed spine curve that lies in a sphere.

REFERENCES

1. D Brander and J Gravesen, *Monge surfaces and planar geodesic foliations*, In preparation.
2. ———, *Surfaces foliated by planar geodesics: a model for curved wood design*, Proceedings of Bridges 2017: Mathematics, Music, Art, Architecture, Culture (2017).
3. G Monge, *Application de l'analyse a la géométrie*, fifth ed., Bachelier, Paris, 1850.
4. W Wang, B Jütler, D Zheng, and Y Liu, *Computation of rotation minimizing frames*, ACM Transactions on Graphics **27** (2008).

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTER SCIENCE, MATEMATIKTORVET, BUILDING 303 B, TECHNICAL UNIVERSITY OF DENMARK, DK-2800 KGS. LYNGBY, DENMARK
E-mail address: dbra@dtu.dk

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTER SCIENCE, MATEMATIKTORVET, BUILDING 303 B, TECHNICAL UNIVERSITY OF DENMARK, DK-2800 KGS. LYNGBY, DENMARK
E-mail address: jgra@dtu.dk