MATEMATIKA, 2011, Volume 27, Number 2, 159–167

Minimum Energy Curve in Polynomial Interpolation

[1]**Chan Chiu Ling**, [2]**Muhammad Abbas and** [3]**Jamaludin Md. Ali**
[1,2,3]School of Mathematical Science, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia
e-mail: [1]stella_ccl@yahoo.com, [2]malikabbas007@gmail.com, [3]jamaluma@csm.usm.my

Abstract Construction of smooth or more technically fair interpolating curve has been of great interest among researchers. The concept of smoothness involves more than tangent and curvature continuity. One plausible suggestion for smoothness is that the strain energy should attain minimum value. In this paper, planar minimum energy curve is constructed using numerical optimization technique. Magnitude of tangent and second derivatives vector are varied to find the curve that minimizes the curvature functional measuring the strain energy. The resulting curve is free of undesirable shape and exhibited local control.

Keywords Minimum energy curve; curve optimization; smooth curve interpolation.

2010 Mathematics Subject Classification 41A10; 41A15; 65D07; 65D10; 65D05

1 Introduction

Minimum energy curve (MEC) has it root in the draftsman's spline. Draftsman's spline is a flexible piece of thin wooden or metal strip used to smoothly connect a number of points in drawing curves. When the thin strip is bent there will be an internal strain energy created in the cross section area of the strip. Malcom [1] defined that the simplest way to characterize a spline mathematically is with the fact that a spline assumes shape which minimizes its strain energy. Levein [2,3] had described the history of elastica and stated the minimum energy curve (MEC) as one of the classic curve. The study of elastic strain energy was started by Bernoulli in 1691 and followed by a number of researchers. At the early stage, the studies of elastic concentrate on the physic aspects. More recent, numerical technique is applied.

Glass [4] had published the first paper on construction of discrete MEC. The method is time consuming because it involved solving multipoint-boundary value problem. Woodford [5] had further improved Glass's method through solving linear multi-boundary value problem. Jerome [6] discussed the existence of MEC. Lee and Forsythe [7] had defined a nonlinear spline as those curves whose energy is local minimum among all curves with continuous tangent and curvature continuity on a finite set of points. They derived algebraic and differential equation relating curvature, angle, arc length and tangent force for both open and closed nonlinear spline. Malcom [1] had computed the MEC by using finite difference method. In the more recent literatures, Moreton [8] had constructed the minimum energy networks by varying the free parameter at the end point but there is lack of discussion on method used to control the curve such that it is free of undesirable curve shape. Yong [9] had defined that an optimum curve should be mathematically smooth, that is of minimum strain energy and geometrically smooth if free from loops, cusps or folds. He derived a new class of curves called optimal geometric hermite curves. Yong adopted the cubic Hermite curve which only interpolates the point position and first derivatives.

Here we described a method for computation of optimum curve using the quintic Hermite curve which interpolates the point position, first and second derivatives in an addition it is free of loops, cups and folds.

2 Quintic Hermite

A quintic Hermite curve $r(t)$,$t \in [t_0, t_1]$ (where $t_0, t_1 \in$ R and $t_0 < t_1$) is a parametric curve satisfying end point position, first derivatives and second derivatives(P, P', P''). The quintic Hermite curve can be expressed as:

$$r(t) = P_0H_0 + P_0'H_1 + P_0''H_2 + P_1''H_3 + P_1'H_4 + P_1H_5$$

H_i are quintic blending function as described in [10].

$$\begin{aligned}
H_0 &= B_0^5 + B_1^5 + B_2^5 \\
H_1 &= \tfrac{1}{5}[B_1^5 + 2B_2^5] \\
H_2 &= \tfrac{1}{20}B_2^5 \\
H_3 &= \tfrac{1}{20}B_3^5 \\
H_4 &= -\tfrac{1}{5}\left[2B_3^5 + B_4^5\right] \\
H_5 &= B_3^5 + B_4^5 + B_5^5
\end{aligned}$$

$B_i^n$are Bernstein polynomials, defined by

$$B_i^n(t) = \left(\begin{array}{c} n \\ i \end{array}\right) t^i(1-t)^{n-i}$$

Refer [6] the first and second derivatives are defined as:

$$P' = m\hat{t} \tag{1}$$

$$P'' = m^2\hat{n}k + \alpha m\hat{t} \tag{2}$$

where

$\hat{t} =$ unit tangent vector
$\hat{n} =$ unit normal vector
$k =$ curvature
$m =$ magnitude of tangent vector
$\alpha =$ parameter that completes the relation between curvature and the second derivatives

The derivatives are definite in this way, in order that the resulting curves interpolation satisfies second order geometric continuity G^2.

3 Strain Energy

The curvature functional measuring strain energy of a curve is defined as the integral of curvature square over arc length.

$$\int_0^l k(s)^2 ds$$

It is converted into integral function of parameter t in $r(t)$form as follow:

$$\int_0^l k(s)^2 ds = \int_0^t \left(\frac{\dot{r}(t) \times \ddot{r}(t)}{\|\dot{r}(t)\|}\right)^2 \|\dot{r}(t)\|\, dt = \int_0^t \frac{(\dot{r}(t) \times \ddot{r}(t))^2}{(\dot{r}(t) \cdot \dot{r}(t))^{5/2}} dt \qquad (3)$$

where $\dot{r}(t)$ and $\ddot{r}(t)$ are the first and second derivatives of $r(t)$ with respect to t.

4 Optimization

In the optimization process, the strain energy is minimized by varying the magnitude of the tangent and second derivatives vectors. Curvature functional (3) measuring the strain energy is considered as the objective function $f(X)$ in the optimization process. Conjugate gradient descent method is used to optimize the objective function. In general, the idea of this optimization method is to search for the suitable direction in order to locate the minimum of $f(X)$.

The conjugate gradient descent method performs well for any continuous objective function provided the gradient can be computed. The gradient is set as the first partial derivatives of the objective function with respect to the free parameter measuring the magnitude of the tangent vector and the second derivatives at the end points. The next step follows up after computation of gradient is the computation of direction. There are several methods available for computation of the direction; here the Polak-Ribiere method because it is known to perform better for non quadratic function. For each direction a one dimensional optimization process is performed to determine the step length.

Since the objective function in this study is a non-quadratic function, the optimization is operated in cycle. The function is said to be converge when the gradient approach to zero. The algorithm is showed in appendix A.

During the optimization process, the magnitude of the tangent vector in (1) and (2) must be constrained to be positive such that the tangent of $\vec{H}(t)$ at t_0 and t_1 would have the same direction as $\hat{t}_0$and $\hat{t}_1$ respectively. We impose this constraint by controlling the range of step lengths search.

4.1 Initialization

The optimization process described in the previous section starts with an initial curve. Referring to [6], the magnitude of the tangent and second derivatives is initialized to the incident chord length $\|P_i - P_{i+1}\|$. The other parameters are set as follow

- Free parameter $\alpha_{i0}$and α_{i1} are set to be zero such that the curve is arc length parameterized to its end point.

- Tangents vector are set to the average of the incident cord direction.

$$t_i = \frac{P_{i-1} - P_i}{\|P_{i-1} - P_i\|} + \frac{P_i - P_{i+1}}{\|P_i - P_{i+1}\|}, \qquad \hat{t}_i = \frac{t_i}{\|t_i\|}$$

- Curvature are set to be curvature of circle passing through P_{i-1}, P_i and P_{i+1}.

$$k_i = \frac{2\,(P_i - P_{i-1}) \times (P_{i+1} - P_i)}{\|P_i - P_{i-1}\|\,\|P_{i+1} - P_1\|\,\|P_{i+1} - P_{i-1}\|}$$

In the case where the end point does not have two neighbors, tangent direction are set to be the image of inverse direction of neighbor's tangent reflecting from the line passing through midpoint and is perpendicular to the intervening cord (See Figure 1). Curvatures are set to be zero.

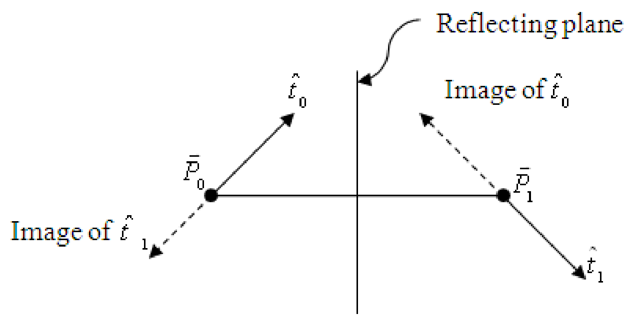


Figure 1: Tangent Direction at the End Point of the Curve

4.2 Range of Step Length Search

In the optimization process, the refined magnitude of tangent vector and second derivatives vector is expressed as follows

$$X_{k+1} = X_k + \lambda d_k$$

where X_{k+1} is the refine magnitude X_k and d_k is the magnitude and direction of the previous iteration. λ is the step length.

Since the magnitude must be positive, the refined magnitude is constrained to be greater than zero. In addition, to avoid undesirable shape that is loop; cusps and folds in the resulting curve, the refined magnitude is constrained to be smaller then twice the incident chord length. Other then these two constrains, the sign of the direction shall also be considered. The range for step length search is considered using the direction of first iteration.

When the direction of first iteration is positive, $d_1 > 0$, the range is set as follows,

$$\frac{-X_1}{d_1} < \lambda < \frac{2\,\|P_0 - P_1\| - X_1}{d_1}$$

When the direction of first iteration is negative, $d_1 < 0$, the range is set as follows,

$$\frac{2\,\|P_0 - P_1\| - X_1}{d_1} < \lambda < \frac{-X_1}{d_1}$$

5 Numerical Integration

In the optimization process, we need to compute the strain energy and their partial derivatives with respect to the magnitude. Gauss Legendre integration is used to perform this computation.

$$\int_0^1 f(t)dt \approx \frac{1}{2}\sum_k w(\xi_k) f(\frac{1}{2}\xi_k + \frac{1}{2})$$

where ξ_k and $w(\xi_k)$ are abscissas and weights of the Gauss Legendre integration. The advantage of using Gauss Legendre integration is it has high precision.

6 Result and Discussion

This section shows a sample of MEC computed using six data points

$$[1.2, 1.85],\ [2.7, 0.85]\ [4.45, 2.3]\ [1.85, 5]\ [3, 6.15] \text{ and } [4.5, 5.5].$$

The magnitude of tangent vector and second derivatives vector is refined until it falls below the gradient tolerance of 1×10^{-6}. The energy and arc length is evaluated by using 20 integration points. The sample curve is illustrated in Figure 2 and the iteration proceeded for the first piece is showed separately in Table 1 in Appendix B. In Figure 2, the dash line represents the initial curve and the dark line represents the optimized curve. Figure 3 compared the resulting optimized curve (dark line) with the cubic Bezier curve (dash line), both satisfying the same end point constrains and second order geometric continuity.

Figure 4 showed sample curves computed using eight data points $[2, 2.4]$,$[2, 3.8]$,$[3.4, 3.6]$, $[3.9, 2.3]$,$[2.6, 1]$,$[1, 2]$,$[1.3, 4.5]$ and $[3.9, 4]$. The magnitude of tangent vector and second derivatives vector of optimum curve is refined until it fall below the gradient tolerance of 1×10^{-4}. Sample curve (a) is the optimized quintic hermite interpolation and sample curve (b) is the cubic Bezier interpolation. Figure 4 shows that the optimized curve has better local control compared to the cubic Bezier curve. In Figure 4(a), when a point is moved, only the shape near to the moved point changed while in Figure 4(b), a moved point causes more effect to the curve shape of neighboring pieces.

7 Conclusion

We have presented the technique for computing minimum energy curves using piece wise interpolation. The resulting curve is free of undesirable shape such as loops, cusps and folds. In an addition the resulting curve exhibited better local control compare to the cubic Bezier interpolation.

Acknowledgement

This work is supported by USM Fellowship.

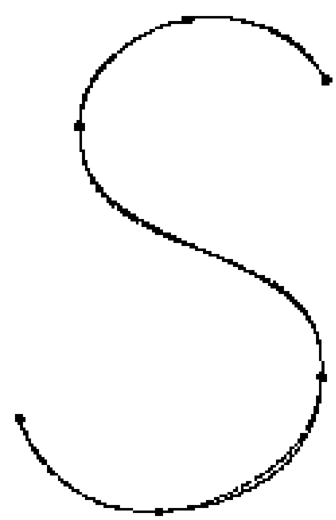
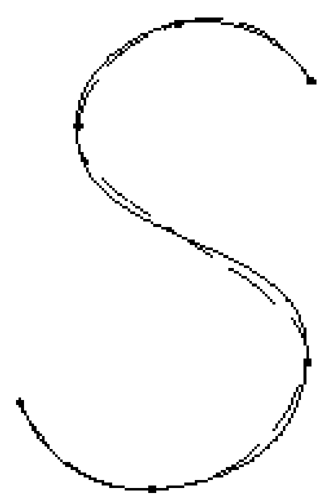
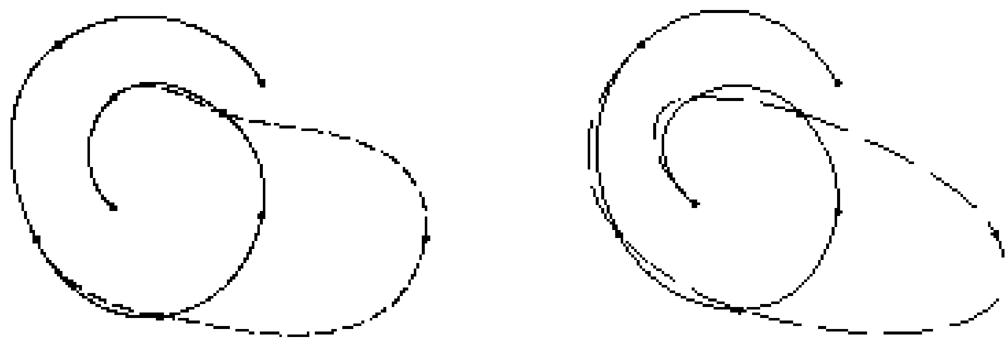


Figure 2: Interpolation of Initial and Optimized Curve in Font's Design



.

Figure 3: Comparison of Cubic Bezier Interpolation and MEC Interpolation



(a) (b)

Figure 4: (a) Optimum Curve with Better Local Control and (b) Cubic Bezier Curve with Less Local Control

References

[1] Malcolm, M. A. On the computation of nonlinear spline functions, *SIAM Journal on Numerical Analysis.* 1977. 14(2): 254-282.

[2] Levien, R. The Elastica: a mathematical history. Technical Report UCB/EECS- 2008-103, EECS Department, University of California, Berkeley. 2008.

[3] Levein, R. and Sequin, C. H. Interpolating splines: which is the fairest of them all. *Computer-Aided Design and Applications.* 2009. 6: 1-4.

[4] Glass, J. M. Smooth-curve interpolation: a generalized spline-fit procedure. *BIT Numerical Mathematics.* 1966. 6(4): 277-293.

[5] Woodford, C. H. Smooth curve interpolation. *BIT Numerical Mathematics.* 1969. 9(1): 69-77.

[6] Jerome, J. W. Minimization problems and linear and nonlinear spline functions i: existence. *SIAM Journal on Numerical Analysis.* 1973. 10(5): 808-819.

[7] Lee, E. H. and Forsythe, G. E. Variational study of nonlinear spline curves. *SIAM Review.* 1973. 15(1): 120-133.

[8] Moreton, H. and Séquin, C. *Surface design with minimum energy networks.* In *Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications.*

[9] Yong, J. H. and Cheng, F. Geometric Hermite curves with minimum strain energy. *Computer Aided Geometric Design.* 2004. 21(3): 281-301.

[10] Farin, G. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide.* 3rd Edition. New York: Academic Press. 1993.

Appendix A

Algorithm 1: Conjugate gradients descent method for non-quadratic function using the Polak-Ribiére formula.

Consider the objective function as $f(X)$ where $X = \begin{bmatrix} m_0 \\ m_1 \end{bmatrix}$ and gradient

$$\nabla f(X) = \begin{bmatrix} \frac{\partial f(X)}{m_0} \\ \frac{\partial f(X)}{m_1} \end{bmatrix}.$$

INPUT initializes magnitude m_0,m_1

OUTPUT Optimum magnitude m_0,m_1

Initializing $i = 1$

Step 1 Set gradient $g_i = \nabla f(X_i)$;

Direction $d_i = -g_i$.

Step 2 Determine the step length $\lambda_i = \min f(X_i + \lambda_i d_i)$;

Step 3 Calculate the new magnitudes $X_{i+1} = X_i + \lambda_i d_i$;

If $g_i > TOL$ then
Continue with step 4
else STOP.

Step 4 Calculate the gradient $g_{i+1} = \nabla f(X_{i+1})$;

Calculate the direction $d_{i+1} = -g_{i+1} + \beta_i d_i$;

$$\beta_i = \frac{g_{i+1}^T \cdot (g_{i+1} - g_i)}{g_i^T \cdot g_i}.$$

Step 6 Determine the step length $\lambda_i = \min f(X_i + \lambda_i d_i)$;

Step 7 Calculate the new magnitudes$X_{i+1} = X_i + \lambda_i d_i$;

If $g_{i+1} > TOL$ then
$i = i + 1$, repeat with step 1
else STOP.

Appendix B

Table 1: Iteration Procedure for *C1* in Figure 2

Iteration	Magnitude $\begin{bmatrix} m_0 \\ m_1 \end{bmatrix}$	Energy	Arc length	Gradient $d = \begin{bmatrix} x \\ y \end{bmatrix}$	Step length
1	$\begin{bmatrix} 1.802776 \\ 1.802776 \end{bmatrix}$	0.884701	1.944961	$\begin{bmatrix} 0.038063 \\ -0.000782 \end{bmatrix}$	2.653469
2	$\begin{bmatrix} 1.701775 \\ 1.804850 \end{bmatrix}$	0.882785	1.938850	$\begin{bmatrix} -0.0006584 \\ -0.032051 \end{bmatrix}$	27.639370
3	$\begin{bmatrix} 0.974015 \\ 2.706054 \end{bmatrix}$	0.862407	1.942035	$\begin{bmatrix} 0.044940 \\ 0.036291 \end{bmatrix}$	1.285383
4	$\begin{bmatrix} 0.916251 \\ 2.659407 \end{bmatrix}$	0.860235	1.937135	$\begin{bmatrix} 0.007969 \\ -0.009868 \end{bmatrix}$	33.392902
5	$\begin{bmatrix} 0.577801 \\ 2.930491 \end{bmatrix}$	0.856723	1.937353	$\begin{bmatrix} 0.013633 \\ 0.017021 \end{bmatrix}$	0.809609
6	$\begin{bmatrix} 0.566764 \\ 2.916711 \end{bmatrix}$	0.856529	1.936171	$\begin{bmatrix} 0.002269 \\ \text{-}0.001817 \end{bmatrix}$	20.475822
7	$\begin{bmatrix} 0.515348 \\ 2.947728 \end{bmatrix}$	0.856438	1.93589	$\begin{bmatrix} 0.000076 \\ 0.000126 \end{bmatrix}$	0.713392
8	$\begin{bmatrix} 0.515294 \\ 2.947638 \end{bmatrix}$	0.856438	1.935887	$\begin{bmatrix} 0.2619\text{e-}5 \\ \text{-}0.1578\text{e-}5 \end{bmatrix}$	17.882610
9	$\begin{bmatrix} 0.515247 \\ 2.947665 \end{bmatrix}$	0.856438	1.935887	$\begin{bmatrix} \text{-}0.3160\text{e-}8 \\ \text{-}0.4780\text{e-}8 \end{bmatrix}$	-1.329784
10	$\begin{bmatrix} 0.515247 \\ 2.947665 \end{bmatrix}$	0.856438	1.935887		