Geodesic curve on the surface的Geogebra程式碼SOP

撰文：104乙 李振岡

工具：Geogebra 5.0R (簡寫GB)

1. 選擇一個要做的曲面來做。
   1. Torus甜甜圈面

f(u,v) = (3 + cos(u)) cos(v)

g(u,v) = (3 + cos(u)) sin(v)

h(u,v) = sin(u)

* 1. Ellipse 橢圓

f(u,v) = 3 sin(u) sin(v)

g(u,v) = 3 sin(u) cos(v)

h(u,v) = 2 cos(u)

* 1. Helicoid

F(u,v)= 3sinh(u) cos(v)

g(u,v) = 3sinh(u) sin(v)

h(u,v) = 3v

* 1. Catenoid

F(u,v)= cosh(u) cos(v)

g(u,v) = cosh(u) sin(v)

h(u,v) = u

GB：選數值滑桿中的「輸入欄位」建立三個輸入欄位如下；

標籤X(u,v)= 相關物件選f

標籤Y(u,v)= 相關物件選g

標籤Z(u,v)= 相關物件選h

1. 顯示曲面的GB程式碼

如果是torus

X(u, v) = Surface[f(u,v), g(u,v), h(u,v), u, 0, 2 pi, v, 0, 2 pi]

如果是ellipse

X(u, v) = Surface[f(u,v), g(u,v), h(u,v), u,0, pi, v, 0, 2 pi]

1. 計算surface的1st fundamental form 的係數E,F,G和 Christoffel symbols (用A\_{kij}代表，網站上的例子是用A\_{ijk}代表)：

先作f, g, h分別對u, v做偏導數

f\_u = Derivative[f, u]

f\_v = Derivative[f, v]

g\_u = Derivative[g, u]

g\_v = Derivative[g, v]

h\_u = Derivative[h, u]

h\_v = Derivative[h, v]

E1= f\_u f\_u + g\_u g\_u + h\_u h\_u

E =Simplify[E1]

F1= f\_u f\_v + g\_u g\_v + h\_u h\_v

F =Simplify[F1]

G1= f\_v f\_v + g\_v g\_v + h\_v h\_v

G =Simplify[G1]

E\_u = Derivative[E, u]

E\_v = Derivative[E, v]

F\_u = Derivative[F, u]

F\_v = Derivative[F, v]

G\_u = Derivative[G, u]

G\_v = Derivative[G, v]

D1= E G - F F

D=Simplify[D1]

B\_{111}= (G E\_u - 2 F F\_u + F E\_v) / (2 D)

B\_{112}= (G E\_v - F G\_u ) / (2 D)

B\_{122}= (2 G F\_v - G G\_u - F G\_v) / (2 D)

B\_{211}= (2 E F\_u - E E\_v - F E\_u) / (2 D)

B\_{212}=(E G\_u - F E\_v) / (2 D)

B\_{222}=(E G\_v - 2 F F\_v + F G\_u) / (2 D)

A\_{111}=Simplify[B\_{111}]

A\_{112}=Simplify[B\_{112}]

A\_{122}=Simplify[B\_{122}]

A\_{211}=Simplify[B\_{211}]

A\_{212}=Simplify[212]

A\_{222}=Simplify[B\_{222}]

到這裡都是可以用手算的部分

1. 開始困難的地方，要解數值ODE，這裡就是用手不一定能算的部分，所以要用電腦。

u’(t, u, v ,u1 ,v1) = u1

v’(t, u, v ,u1 ,v1) = v1

u1’(t, u, v ,u1 ,v1) = - (A\_{111}(u,v) u1 u1 + 2 A\_{112}(u,v) u1 v1 + A\_{122}(u,v) v1 v1)

v1’(t, u, v ,u1 ,v1) = - (A\_{211}(u,v) u1 u1 + 2 A\_{212}(u,v) u1 v1 + A\_{222}(u,v) v1 v1)

在複製到GB的時候，Word上的「’」在Geogebra裡頭會自己變成全形「’」，然後Geogebra會看不懂，所以要記得改掉。

1. 在GB 建立下面數據：

t\_0 = 0

t\_1用數值滑桿，範圍0.1到15，最小值不可以從0開始，最大值大概就好

u\_0用數值滑桿，範圍0到2 pi

v\_0用數值滑桿，範圍0到2 pi

a 用數值滑桿，範圍0到2 pi（就是平常的sigma）

1. 給初始條件，也就是給S上一點P，及P上的一個向量：

給定曲面S上一P點：

P = (f(u\_0 , v\_0) , g(u\_0 , v\_0) , h(u\_0 , v\_0))

在P點上給定一個向量V：

T\_{u0} = (f\_u(u\_0, v\_0) , g\_u(u\_0, v\_0) , h\_u(u\_0, v\_0))

T\_{v0} = (f\_v(u\_0, v\_0) , g\_v(u\_0, v\_0) , h\_v(u\_0, v\_0))

V\_1= Cos(a) T\_{u0} + sin(a) T\_{v0}

V = Vector[P , P + V\_1]

V\_{00} = (cos(a), sin(a))

說明：給定了P點，就固定了切平面。切平面可以由兩條向量T\_{u0}和T\_{v0}給span出來。只要再給一個角度a，就可以在切平面上找到一條向量V，使得V和T\_{u0}的夾角為a。

1. 解數值ODE的指令：

NSolveODE[{u’ , v’ , u1’ , v1’}, t\_0 , {u\_0, v\_0, x(V\_{00}), y(V\_{00})} , t\_1]

記得把’改成’。成功的話，會出現四條曲線，這四條曲線依序是解出來的{u, v, u1, v1}：

numericalIntegral1

numericalIntegral2

numericalIntegral3

numericalIntegral4

指令說明：NSolveODE[應變數所成的集合, 自變數初始值, 應變數初始值所成的集合, 自變數終點值]。其中每個應變數都是函數，而且固定了u\_0, v\_0, a之後，u, v兩個函數都只以t為自變數，t的範圍為[t\_0, t\_1]。

在二維的繪圖區裡，可以看到numericalIntegral1和numericalIntegral2的軌跡。這說明了函數u, v的圖形上的每個點，坐標依序各是(t, u(t))和(t, v(t))。這件事在下面會用到。

1. 作Geodesic curve：

解出了u, v之後，γ(t) = (u(t), v(t))就是我們要的Geodesic curve。我們現在就要做出這條曲線。

想要在numericalIntegral1和numericalIntegral2這兩條曲線上面做partition，因此先用Length指令求出弧長。這個弧長一定是整數，也就是partition的段數。

len = Length[numericalIntegral1]

框框內的nemericalIntegral1只要從1,2,3,4之中隨便選一個就好。

再做一個數值滑桿L，範圍從0到1，間隔隨便。

輸入框框裡面這個

|  |
| --- |
| Sequence[Segment[(f(y(Point[numericalIntegral1, s1]), y(Point[numericalIntegral2, s1])), g(y(Point[numericalIntegral1, s1]), y(Point[numericalIntegral2, s1])), h(y(Point[numericalIntegral1, s1]), y(Point[numericalIntegral2, s1]))),(f(y(Point[numericalIntegral1, s1 + 1/len]), y(Point[numericalIntegral2, s1 + 1/ len])), g(y(Point[numericalIntegral1, s1 + 1/len]), y(Point[numericalIntegral2, s1 + 1 / len])), h(y(Point[numericalIntegral1, s1 + 1/len]), y(Point[numericalIntegral2, s1 + 1 / len])))], s1, 0, L – 1/len, 1/len] |

這是一句話，意思是把Geodesic curve上的點一個個連起來。那麼就畫出了Geodesic curve。

分開來看比較清楚：

Sequence[Segment[

(

f(y(Point[numericalIntegral1, s1]), y(Point[numericalIntegral2, s1])),

g(y(Point[numericalIntegral1, s1]), y(Point[numericalIntegral2, s1])),

h(y(Point[numericalIntegral1, s1]), y(Point[numericalIntegral2, s1]))

), (

f(y(Point[numericalIntegral1, s1 + 1 / len]), y(Point[numericalIntegral2, s1 + 1/ len])),

g(y(Point[numericalIntegral1, s1 + 1 / len]), y(Point[numericalIntegral2, s1 + 1 / len])),

h(y(Point[numericalIntegral1, s1 + 1 / len]), y(Point[numericalIntegral2, s1 + 1 / len]))

)

], s1, 0, L – 1/len, 1/len]

這裡將numericalIntegral1和numericalIntegral2視為兩個有限項數列，這兩個數列都是，從第0項開始，接著是第1 / len項、第2 / len項、第3 / len項、……、一直到第L – 1 / len項是最後一項。總共len個項，也就是partition的段數有len - 1段。

這兩條數列的每一項，都是一組坐標，依序是剛剛說過的(t, u) = (t, u(t))和(t, v) = (t, v(t))。而s1表示兩個數列的各第s1項的值(t, u)和(t, v)，其中u和v是我們要的值，故用y(Point[numericalIntegral1, s1])表示u的值，y(Point[numericalIntegral2, s1])表示v的值。

我們最終的目標就是做出γ(t) = (u(t), v(t))，剛剛已經取出了len個u值和len個v值，現在把他們兩兩放進坐標裡，就會有len組(u, v)，各自代入f(u, v)、g(u, v)、h(u,v)，就會是曲面上len個點。把這len個點一個個連起來，就完成了Geodesic curve。

Sequence指把裡面的東西重複做

（指令：Sequence[要做的事, 參數, 參數起點, 參數終點, 參數間隔]）

Segment指把裡面的東西連起來，就是上面「要做的事」

（指令：Segment[起點，終點]）

會出現list1 = ……，那是因為沒有把Sequence命名。喜歡的話可以換一個名字，不要重複到就好。

Geodesic curve到這裡就做完了。

建議事項：

1. 以下再多設計一個開關功能。

因為電腦計算數值ODE需要花很多記憶體空間，所以要改變P點（即u\_0 和v\_0）和初始向量V（即角度a）之前，建議先停止計算數值ODE。我們的做法是讓解數值ODE的步驟不能執行。也就是，我們要作一個按鈕，按下去就可以讓電腦停止計算數值ODE，再按下去又可以讓電腦開始計算數值ODE。

作法如下：

令一個真假值，初始值是真是假不重要

geodesic = false

並將以前的t\_0改變定義為

t\_0 = If[geodesic =? true, 0]

其中「=?」是等號上面一個問號，要用按的才能按出來。這個意思是如果t\_0是true，那就令t\_0 = 0；如果t\_0是false，那t\_0就未定義。

設定一個按鈕（數值滑桿中的按鈕），標籤隨意，這裡叫做Start or Stop ODE，OnClick程式碼打入下面兩行：

SetValue[geodesic, !geodesic]

StartAnimation[geodesic]

意思是按下去時改變geodesic的真假值。則t\_0會檢驗geodesic是否是true。如果是true，那就使t\_0 = 0，然後數值ODE的條件完整，步驟7就能順利執行，然後就會由步驟8的Sequence指令畫出geodesic curve；如果是false，那就使t\_0未定義，數值ODE的初始條件不完整，步驟7就不能順利執行，所以步驟8也不會畫出geodesic curve。

程式不算數值ODE時，電腦的速度就會變快，可以趁這個時候改變各項參數的值。不信的話可以開著數值ODE改變參數看看。

有時在程式比較多時，經常Geogebra關掉在打開時，有些程式或指令會不見，此時可以新增一個按鈕，在程式、OnClick程式碼內打入需要的程式或指令(可以是很長的程式所組成的程序)，則在按此按鈕時就會執行此程式或指令(或程序)。