

Math 235: Calculus Lab

Prof. Doug Hundley

Whitman College

Week 10

This Week:

- ▶ A piecewise defined path.
- ▶ Optimizing over a family of paths.
- ▶ Discussion of the Lab.

This Week:

- ▶ A piecewise defined path.
- ▶ Optimizing over a family of paths.
- ▶ Discussion of the Lab.

Next Week: Nothing new. Continue (or start) typesetting your paper. Be sure to think about the comments from earlier papers!

A Piecewise Defined Path

In the (β, α) plane, we have three parts:

- ▶ $(0, 0)$ to $(0, \pi)$

Parametrization:

A Piecewise Defined Path

In the (β, α) plane, we have three parts:

- ▶ $(0, 0)$ to $(0, \pi)$

Parametrization: $\beta(t) = 0, \alpha(t) = \pi t$

A Piecewise Defined Path

In the (β, α) plane, we have three parts:

- ▶ $(0, 0)$ to $(0, \pi)$

Parametrization: $\beta(t) = 0, \alpha(t) = \pi t$

- ▶ $(0, \pi)$ to (π, π) ,

Parametrization:

A Piecewise Defined Path

In the (β, α) plane, we have three parts:

- ▶ $(0, 0)$ to $(0, \pi)$

Parametrization: $\beta(t) = 0, \alpha(t) = \pi t$

- ▶ $(0, \pi)$ to (π, π) ,

Parametrization: $\beta(t) = \pi t, \alpha(t) = \pi$

A Piecewise Defined Path

In the (β, α) plane, we have three parts:

- ▶ $(0, 0)$ to $(0, \pi)$

Parametrization: $\beta(t) = 0, \alpha(t) = \pi t$

- ▶ $(0, \pi)$ to (π, π) ,

Parametrization: $\beta(t) = \pi t, \alpha(t) = \pi$

- ▶ (π, π) to $(\pi, 0)$,

Parametrization:

A Piecewise Defined Path

In the (β, α) plane, we have three parts:

- ▶ $(0, 0)$ to $(0, \pi)$

Parametrization: $\beta(t) = 0, \alpha(t) = \pi t$

- ▶ $(0, \pi)$ to (π, π) ,

Parametrization: $\beta(t) = \pi t, \alpha(t) = \pi$

- ▶ (π, π) to $(\pi, 0)$,

Parametrization: $\beta(t) = \pi, \alpha(t) = \pi(1 - t)$

Implementation in Maple (See worksheet)

The Torus

```
with(plots):  
f:=cos(beta)*(cos(alpha)+2):  
g:=sin(beta)*(cos(alpha)+2):  
h:=sin(alpha):  
Torus1:=plot3d([f,g,h],alpha=0..2*Pi,beta=0..2*Pi,  
    scaling=constrained,style=wireframe,shading=ZHUE):
```

The Paths

```
xt:=0: yt:=Pi*t:
Path2AF:=subs({beta=xt,alpha=yt},f),
          subs({beta=xt,alpha=yt},g),
          subs({beta=xt, alpha=yt},h):
xt:=Pi*t: yt:=Pi:
Path2BF:=subs({beta=xt,alpha=yt},f),
          subs({beta=xt,alpha=yt},g),
          subs({beta=xt, alpha=yt},h):
xt:=Pi: yt:=Pi*t:
Path2CF:=subs({beta=xt,alpha=yt},f),
          subs({beta=xt,alpha=yt},g),
          subs({beta=xt, alpha=yt},h):
```

The Path Lengths

```
dP1:=diff([Path2AF],t);  
dP2:=diff([Path2BF],t);  
dP3:=diff([Path2CF], t);  
Integrand1:=simplify(dP1[1]^2+dP1[2]^2+dP1[3]^2);  
Integrand2:=simplify(dP2[1]^2+dP2[2]^2+dP2[3]^2);  
Integrand3:=simplify(dP3[1]^2+dP3[2]^2+dP3[3]^2);  
PathLength:=int(sqrt(Integrand1),t=0..1)+  
               int(sqrt(Integrand2),t=0..1)+  
               int(sqrt(Integrand3),t=0..1);
```

Why the square brackets?

The Path Lengths

```
dP1:=diff([Path2AF],t);  
dP2:=diff([Path2BF],t);  
dP3:=diff([Path2CF], t);  
Integrand1:=simplify(dP1[1]^2+dP1[2]^2+dP1[3]^2);  
Integrand2:=simplify(dP2[1]^2+dP2[2]^2+dP2[3]^2);  
Integrand3:=simplify(dP3[1]^2+dP3[2]^2+dP3[3]^2);  
PathLength:=int(sqrt(Integrand1),t=0..1)+  
               int(sqrt(Integrand2),t=0..1)+  
               int(sqrt(Integrand3),t=0..1);
```

Why the square brackets?

The variable `Path2AF` is a list of three things (with no delimiters around them).

Putting square brackets around it make it one “thing”, so the derivative operation will work.

Graphics

```
Path2A:=spacecurve([Path2AF,t=0..1],color=black,  
    thickness=5):  
Path2B:=spacecurve([Path2BF,t=0..1],color=black,  
    thickness=5):  
Path2C:=spacecurve([Path2CF,t=0..1],color=black,  
    thickness=5):  
display3d({Torus1, Path2A, Path2B, Path2C});
```

Optimizing over a Family of Paths

Rather than going from $(0, 0)$ to $(\pi/2, \pi)$ as we did in Path 3, let β go from 0 to an unknown value, k as α runs from 0 to π .

Path 4 in the (β, α) plane:

$$(0, 0) \rightarrow (k, \pi) \rightarrow (?, \pi) \rightarrow (\pi, 0)$$

To be symmetric, the unknown should be:

Optimizing over a Family of Paths

Rather than going from $(0, 0)$ to $(\pi/2, \pi)$ as we did in Path 3, let β go from 0 to an unknown value, k as α runs from 0 to π .

Path 4 in the (β, α) plane:

$$(0, 0) \rightarrow (k, \pi) \rightarrow (?, \pi) \rightarrow (\pi, 0)$$

To be symmetric, the unknown should be: $\pi - k$.

Optimizing over a Family of Paths

Rather than going from $(0, 0)$ to $(\pi/2, \pi)$ as we did in Path 3, let β go from 0 to an unknown value, k as α runs from 0 to π .

Path 4 in the (β, α) plane:

$$(0, 0) \rightarrow (k, \pi) \rightarrow (?, \pi) \rightarrow (\pi, 0)$$

To be symmetric, the unknown should be: $\pi - k$.

Path 4A: $(0, 0)$ to (k, π)

Optimizing over a Family of Paths

Rather than going from $(0, 0)$ to $(\pi/2, \pi)$ as we did in Path 3, let β go from 0 to an unknown value, k as α runs from 0 to π .

Path 4 in the (β, α) plane:

$$(0, 0) \rightarrow (k, \pi) \rightarrow (?, \pi) \rightarrow (\pi, 0)$$

To be symmetric, the unknown should be: $\pi - k$.

Path 4A: $(0, 0)$ to (k, π)

Path 4B: (k, π) to $(\pi - k, \pi)$

Optimizing over a Family of Paths

Rather than going from $(0, 0)$ to $(\pi/2, \pi)$ as we did in Path 3, let β go from 0 to an unknown value, k as α runs from 0 to π .

Path 4 in the (β, α) plane:

$$(0, 0) \rightarrow (k, \pi) \rightarrow (?, \pi) \rightarrow (\pi, 0)$$

To be symmetric, the unknown should be: $\pi - k$.

Path 4A: $(0, 0)$ to (k, π)

Path 4B: (k, π) to $(\pi - k, \pi)$

Path 4C: $(\pi - k, \pi)$ to $(\pi, 0)$.

Optimizing over a Family of Paths

Rather than going from $(0, 0)$ to $(\pi/2, \pi)$ as we did in Path 3, let β go from 0 to an unknown value, k as α runs from 0 to π .

Path 4 in the (β, α) plane:

$$(0, 0) \rightarrow (k, \pi) \rightarrow (?, \pi) \rightarrow (\pi, 0)$$

To be symmetric, the unknown should be: $\pi - k$.

Path 4A: $(0, 0)$ to (k, π)

Path 4B: (k, π) to $(\pi - k, \pi)$

Path 4C: $(\pi - k, \pi)$ to $(\pi, 0)$.

Make the appropriate changes to the Maple file. What values should we allow k to take?

Once we get the paths:

- ▶ Path 4A: $x_t := k * t$; $y_t := P_i * t$
- ▶ Path 4B: $x_t := k * (1 - t) + (P_i - k) * t$
- ▶ Path 4C: $x_t := (P_i - k) * (1 - t) + P_i * t$ $y_t := P_i * (1 - t)$

Be Sure To Use capital I for the Integral!

Once we get the paths:

- ▶ Path 4A: $x_t := k \cdot t$; $y_t := \pi \cdot t$
- ▶ Path 4B: $x_t := k \cdot (1-t) + (\pi - k) \cdot t$
- ▶ Path 4C: $x_t := (\pi - k) \cdot (1-t) + \pi \cdot t$ $y_t := \pi \cdot (1-t)$

Be Sure To Use capital I for the Integral!

- ▶ The path length depends on k . Plot it!
- ▶ Now find the optimal value of the path length!

Sample solution in Maple:

```
xt:=k*t; yt:=Pi*t;  
Path3AF:=subs... (Same as before)  
xt:=k*(1-t)+(Pi-k)*t; yt:=Pi;  
Path3BF:=subs... (Same as before)  
xt:=(Pi-k)*(1-t)+Pi*t; yt:=Pi*(1-t);  
Path3CF:=subs... (Same as before)  
dP1:=diff([Path3AF],t); dP2:=diff([Path3BF],t); dP3:=diff([
```

```
Integrand1:=simplify(dP1[1]^2+dP1[2]^2+dP1[3]^2);  
Integrand2:=simplify(dP2[1]^2+dP2[2]^2+dP2[3]^2);  
Integrand3:=simplify(dP3[1]^2+dP3[2]^2+dP3[3]^2);  
PathLength:=Int(sqrt(Integrand1),t=0..1)+  
               Int(sqrt(Integrand2),t=0..1)+  
               Int(sqrt(Integrand3),t=0..1);  
plot(PathLength,k=0..Pi/2);
```

What should your graph be? At $k = 0$? $k = \pi/2$?

To find the minimum, we set the derivative to zero. We'll need to do it numerically, so we need an approximate answer.

- ▶ Find an expression for the derivative.
- ▶ Plot the derivative to get an approximate answer.
- ▶ Use the approximation in `fsolve`
- ▶ Find the numerical value of the best path.

In Maple:

```
dPath:=diff(PathLength,k);  
plot(dPath,k=0..Pi/2);  
BestK:=fsolve(dPath=0,k=Pi/4..5*Pi/16);  
evalf(subs(k=BestK,PathLength));
```

Continuing: Plot the resulting path in Maple

```
Path3A:=subs(k=BestK,[Path3AF]);  
Path3B:=subs(k=BestK,[Path3BF]);  
Path3C:=subs(k=BestK,[Path3CF]);  
P1:=spacecurve(Path3A,t=0..1,color=black,thickness=5):  
P2:=spacecurve(Path3B,t=0..1,color=black,thickness=5):  
P3:=spacecurve(Path3C,t=0..1,color=black,thickness=5):  
display3d(Torus1,P1,P2,P3);
```