UCB Math 128A, Spring 2014: Programming Assignment 2

Solutions

1. The following MATLAB code computes the spline coefficients and plots the curve:

```
t=0:5;
ax=[1,1.5,2,2,2.5,2.5];
ay=[1,.5,1,1.5,1.5,1];
[bx,cx,dx]=ncspline(t,ax);
[by,cy,dy]=ncspline(t,ay);

tt=0:0.01:5;
xx=splineeval(t,ax,bx,cx,dx,tt);
yy=splineeval(t,ay,by,cy,dy,tt);

plot(xx,yy,ax,ay,'o'),axis equal,grid on
```

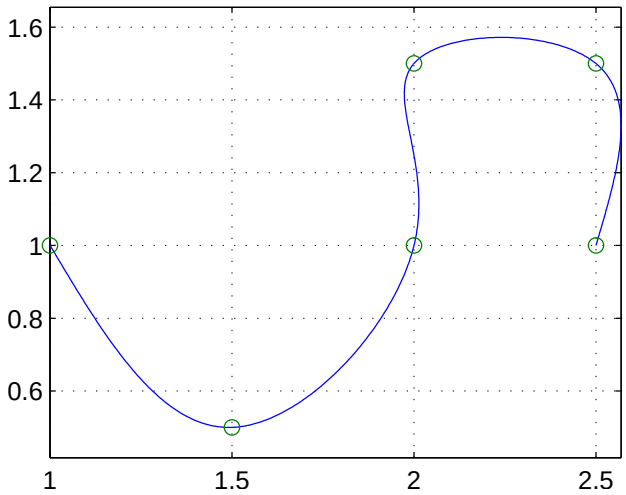
The spline coefficients for $x(t)$ are:

j	t_j	a_j	b_j	c_j	d_j
0	0	1.0	0.4522	0	0.0478
1	1	1.5	0.5957	0.1435	-0.2392
2	2	2.0	0.1651	-0.5742	0.4091
3	3	2.0	0.2440	0.6531	-0.3971
4	4	2.5	0.3589	-0.5383	0.1794
5	5	2.5			

The spline coefficients for $y(t)$ are:

j	t_j	a_j	b_j	c_j	d_j
0	0	1.0	-0.7608	0	0.2608
1	1	0.5	0.0215	0.7823	-0.3038
2	2	1.0	0.6746	-0.1292	-0.0455
3	3	1.5	0.2799	-0.2656	-0.0144
4	4	1.5	-0.2943	-0.3086	0.1029
5	5	1.0			

The plot of the curve is shown below.



2. The following MATLAB code solves for the intersections using Newton's method, with two different starting values:

```
f=@(tt) splineeval(t,ay,by,cy,dy,tt)-1.2;
df=@(tt) diffsplineeval(t,ay,by,cy,dy,tt);
t1=newton(f,df,2.5,1e-12)
t2=newton(f,df,4.5,1e-12)
```

The solutions are $t_1 = 2.31798217$ and $t_2 = 4.66164416$.

3. The following MATLAB code computes the integral using the composite trapezoidal rule with the five values of n:

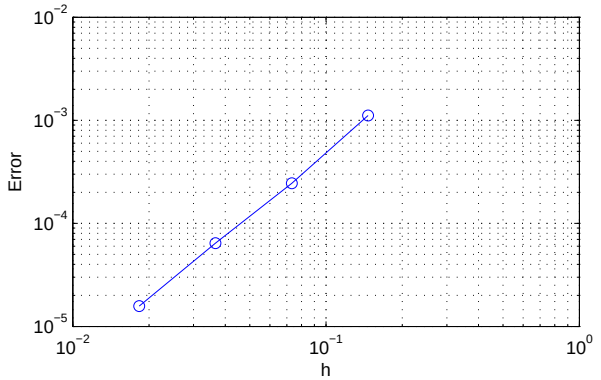
```
ns=[16,32,64,128];
allL=[];
for n=[ns,10000]
  tt=linspace(t1,t2,n+1);
  h=(t2-t1)/n;
  dxx=diffsplineeval(t,ax,bx,cx,dx,tt);
  dyy=diffsplineeval(t,ay,by,cy,dy,tt);
  ds=sqrt(dxx.^2+dyy.^2);
  L=h/2*(2*sum(ds)-ds(1)-ds(end));
  allL=[allL;L];
end
```

The computed integrals are:

n	L_n
16	1.16265486
32	1.16178581
64	1.16160475
128	1.16155626
10000	1.16154050

The errors can be computed and drawn in a log-log plot using the following commands:

```
errors=abs(allL(1:end-1)-allL(end));
hs=(t2-t1)./ns;
loglog(hs,errors,'o-')
xlabel('h')
ylabel('Error')
grid on
```



The slope can be estimated from the 2 most accurate points ($n = 64$ and $n = 128$):

```
slope=(log(errors(end-1))-log(errors(end)))/(log(hs(end-1))-log(hs(end)))
```

The resulting slope is 2.03, which is close to the expected value 2.