

Topic : Additional Interpolation Topics

Simulation : The Need for Spline Interpolation

Language : Matlab r12

Authors : Nathan Collier, Autar Kaw

Date : 15 November 2002

Abstract : This simulation shows the need for spline interpolation as opposed to using polynomial interpolation. The function chosen was first used by Runge in 1901 when he wanted to show that higher order interpolation is a bad idea. He took a simple looking function $f(x)=1/(1+25x^2)$ and chose equidistantly spaced points to interpolate the function.

```
clear all
```

```
% INPUTS: Enter the following
```

```
% Enter the number of points chosen for interpolation in [-1,1]
```

```
n=9;
```

```
% NOTE: For the number scheme to work properly n needs to be an odd number
```

```
% SOLUTION:
```

```
% This calculates window size to be used in figures
```

```
set(0,'Units','pixels')
```

```
scnsize = get(0,'ScreenSize');
```

```
wid = round(scnsize(3));
```

```
hei = round(0.95*scnsize(4));
```

```
wind = [1, 1, wid, hei];
```

```
figure('Position',wind)
```

```
frunge=inline('1/(1+25*x^2)');
```

```
nn=1;
```

```
for i =-1:0.01:1
```

```
    xf(nn)=i;
```

```
    yf(nn)=frunge(xf(nn));
```

```
    nn=nn+1;
```

```
end
```

```
plot(xf,yf,'k','LineWidth',3)
```

```
title('Runge^s function interpolated using polynomial interpolation and cubic spline  
interpolation','Fontweight','bold','FontSize',14)
```

```
hold on
```

```
xlim([-1,1])
```

```
ylim([-1.1,1.1])
```

```
for i=1:n
```

```
    X(i)=2/(n-1)*(i-1)-1;
```

```
    Y(i)=frunge(X(i));
```

```
end
```

```
p=polyfit(X,Y,numel(X)-1);  
x1=-1:0.01:1;  
y1=polyval(p,x1);  
plot(x1,y1,'b','LineWidth',3)
```

```
xx=-1:0.01:1;  
yy=spline(X,Y,xx);  
plot(xx,yy,'g','LineWidth',3)
```

```
plot(X,Y,'ro','MarkerSize',8,'MarkerFaceColor',[1,0,0])
```

```
legend('Runge^s function','Polynomial fit','Spline fit','Data points',4)
```

