CMPSC/Math 451, Numerical Computation

Wen Shen

Department of Mathematics, Penn State University

Natural cubic spline

Computing z_i:

```
function z = cspline(t,y)
n = length(t);
z = zeros(n,1); h = zeros(n-1,1); b = zeros(n-1,1);
u = zeros(n,1); v = zeros(n,1);

h = t(2:n)-t(1:n-1); b = (y(2:n)-y(1:n-1))./h;
u(2) = 2*(h(1)+h(2)); v(2) = 6*(b(2)-b(1));

for i=3:n-1
  u(i) = 2*(h(i)+h(i-1))-h(i-1)^2/u(i-1);
  v(i) = 6*(b(i)-b(i-1))-h(i-1)*v(i-1)/u(i-1);
end

for i=n-1:-1:2
  z(i) = (v(i)-h(i)*z(i+1))/u(i);
end
```

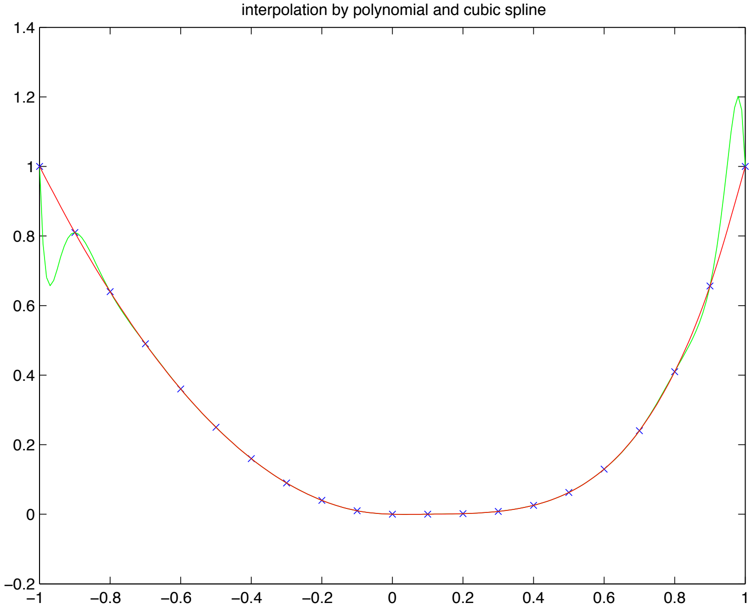
Computing $S(x)$ for a given x:

```
function S = cspline_eval(t,y,z,x)
m = length(x);
n = length(t);
for i=n-1:-1:1
  if (x-t(i)) >= 0
    break
  end
end
h = t(i+1)-t(i);
S = z(i+1)/(6*h)*(x-t(i))^3 ...
   -z(i)/(6*h)*(x-t(i+1))^3 ...
  +(y(i+1)/h-z(i+1)*h/6)*(x-t(i)) ...
  -(y(i)/h-z(i)*h/6)*(x-t(i+1));
```

Use your functions:

```
>> t = [0.9,1.3,1.9,2.1]
t =
    0.9000    1.3000    1.9000    2.1000
>> y = [1.3,1.5,1.85,2.1]
y =
    1.3000    1.5000    1.8500    2.1000
>> z = cspline(t,y)
z =
         0
   -0.5634
    2.7113
         0
>> cspline_eval(t,y,z,1.5)
ans =
    1.5810
```

Comparing polynomial interpolation with cubic spline:



knots (x), polynomial interpolation (🟩), cubic spline (🟥)