

```
% This code demonstrates example shown in class
```

```
clear all;
```

```
V = 1;
```

```
Y=[ 0; 1; 4; 1; 0];
```

```
for i=2:4,  
    Uy(i)=Y(i+1)-Y(i-1);
```

```
end
```

```
Uy(1)=Y(2)-Y(1);
```

```
Uy(5)=Y(5)-Y(4);
```

```
Uy = 3 * Uy'; %makes it Column !!
```

```
M=[2 1 0 0 0;  
    1 4 1 0 0;  
    0 1 4 1 0;  
    0 0 1 4 1;  
    0 0 0 1 2];
```

```
invM= inv(M);
```

```
Sy = invM * Uy % prints Sy !!
```

```
Value = Y(1);
```

```
for i=1:4,  
    Uo = Y(i);  
    U1 = Y(i+1);  
    So = Sy(i);  
    S1 = Sy(i+1);
```

```
for t=0.01:0.01:1,  
    v1 = Uo * (t-1)^2 * (2*t+1) + So * (t-1)^2 * t;  
    v = v1 + U1 * t^2 * (3-2*t) - S1 * t^2 * (1-t);  
    Value = [Value v];
```

```
end
```

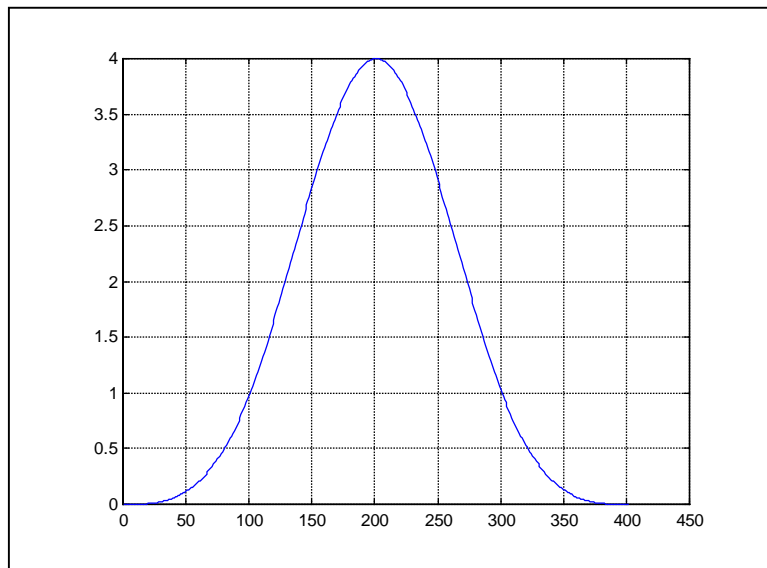
```
end
```

```
% end of file
```

It prints:

Sy =

```
0.0000  
3.0000  
0.0000  
-3.0000  
-0.0000
```



SPLINES function using algorithms posted by 10/18/00 which were explained in class !!!

You need to fix it !! This algorithm only works for nine given points (x,y). Make it work for any size from 4 to 20 points. Note, intervals is the number of intervals printed between such given points.

```
function [ValuesX,ValuesY] = splines(X,Y,intervals)

%Try:
% intervals = 5;
% X=[ 1; 3; 2; 0.5; 4; 5; 2; 0; -1];
% Y=[ 2; 3; 5; 7; 7; 3; 1; 2; 0];
% splines(X,Y,intervals);

[m,n] = size(X);

for i=2:m-1,
    Ux(i)=X(i+1)-X(i-1);
end
Ux(1)=X(2)-X(1);
Ux(m)=X(m)-X(m-1);

Ux = 3 * Ux';

for i=2:m-1,
    Uy(i)=Y(i+1)-Y(i-1);
end
Uy(1)=Y(2)-Y(1);
Uy(m)=Y(m)-Y(m-1);

Uy = 3 * Uy';

% This part has to be changed !!
M=[2 1 0 0 0 0 0 0 0 0;
    1 4 1 0 0 0 0 0 0 0;
    0 1 4 1 0 0 0 0 0 0;
    0 0 1 4 1 0 0 0 0 0;
    0 0 0 1 4 1 0 0 0 0;
    0 0 0 0 1 4 1 0 0 0;
    0 0 0 0 0 1 4 1 0 0;
    0 0 0 0 0 0 1 4 1 0;
    0 0 0 0 0 0 0 1 4 1;
    0 0 0 0 0 0 0 0 1 2];

invM= inv(M);

Sx = invM * Ux;

Sy = invM * Uy;

ValueX = X(1);

% continue ...
```

```

for i=1:m-1,
    Uo = X(i);
    U1 = X(i+1);
    So = Sx(i);
    S1 = Sx(i+1);

    for t=1/intervals:1/intervals:1,
        v = Uo * (t-1)^2 * (2*t+1) + So * (t-1)^2 * t;
        v = v + U1 * t^2 * (3-2*t) - S1 * t^2 * (1-t);
        ValueX = [ValueX v];
    end
end

ValueY = Y(1);

for i=1:m-1,
    Uo = Y(i);
    U1 = Y(i+1);
    So = Sy(i);
    S1 = Sy(i+1);

    for t=1/intervals:1/intervals:1,
        v = Uo * (t-1)^2 * (2*t+1) + So * (t-1)^2 * t;
        v = v + U1 * t^2 * (3-2*t) - S1 * t^2 * (1-t);
        ValueY = [ValueY v];
    end
end

plot(ValueX,ValueY,'.',ValueX,ValueY,'-'), hold on
plot(X,Y,'or')
grid on

% End of splines !!

```

This code only obtains the slopes on every given point !!

```
function [Sx,Sy] = slopes(X,Y,intervals)
```

```
[m,n] = size(X);
```

```
for i=2:m-1,  
    Ux(i)=X(i+1)-X(i-1);  
end  
Ux(1)=X(2)-X(1);  
Ux(m)=X(m)-X(m-1);
```

```
Ux = 3 * Ux';
```

```
for i=2:m-1,  
    Uy(i)=Y(i+1)-Y(i-1);  
end  
Uy(1)=Y(2)-Y(1);  
Uy(m)=Y(m)-Y(m-1);
```

```
Uy = 3 * Uy';
```

```
M=[2 1 0 0 0 0 0 0 0;  
    1 4 1 0 0 0 0 0 0;  
    0 1 4 1 0 0 0 0 0;  
    0 0 1 4 1 0 0 0 0;  
    0 0 0 1 4 1 0 0 0;  
    0 0 0 0 1 4 1 0 0;  
    0 0 0 0 0 1 4 1 0;  
    0 0 0 0 0 0 1 4 1;  
    0 0 0 0 0 0 0 1 2];
```

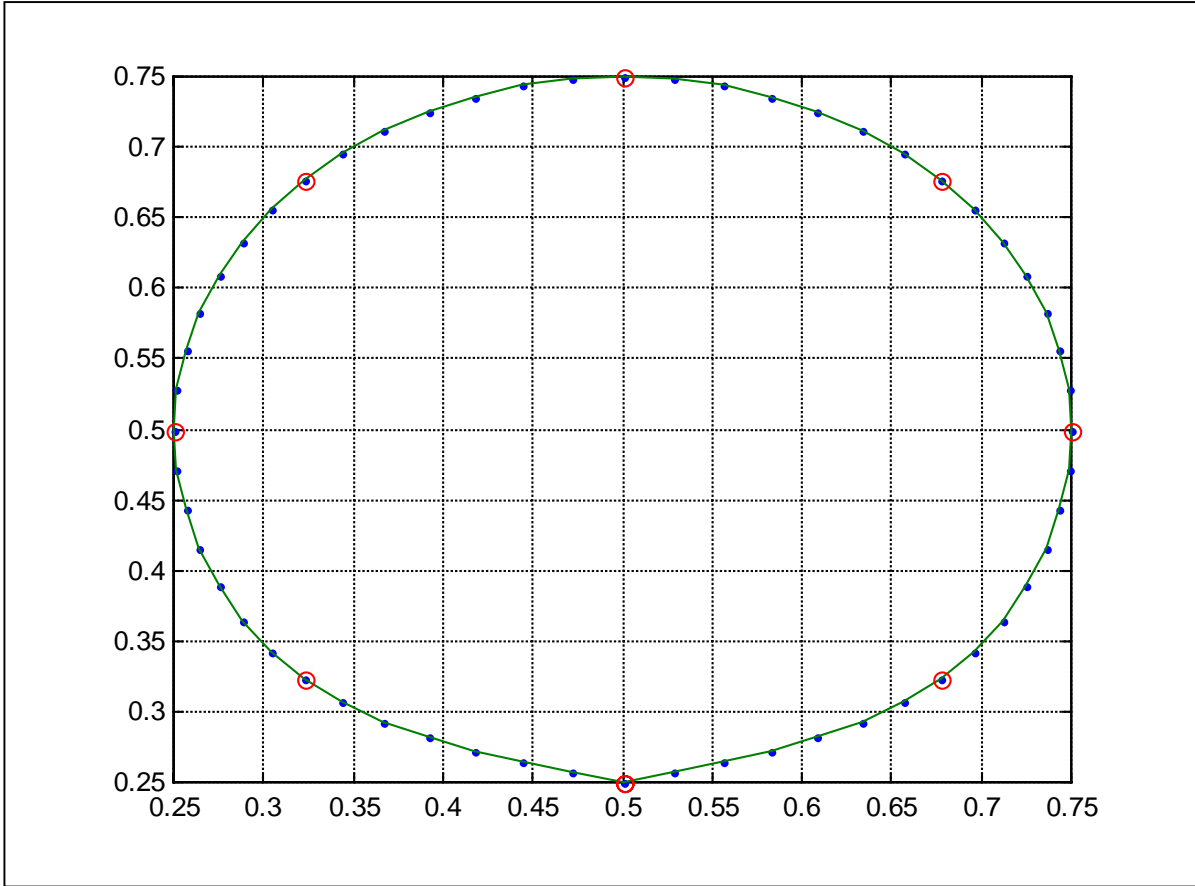
```
invM= inv(M);
```

```
Sx = invM * Ux;
```

```
Sy = invM * Uy;
```

```
% slope = a/b = tan(alpha)  
% alpha = arctan(slope)  
% Vx = V * sin(alpha)  
% they are almost the same !!  
% alphax = atan(Sx);  
% Vx = V * sin(alphax);  
% alphay = atan(Sy);  
% Vy = V * sin(alphay);  
%Vx ~= Sx;  
%Vy ~= Sy;
```

```
» intervals = 7;  
» X=[ 0.500; 0.3232; 0.25; 0.3232; 0.500; 0.6768; 0.750; 0.6768; 0.500];  
» Y=[ 0.250; 0.3232; 0.50; 0.6768; 0.750; 0.6768; 0.500; 0.3232; 0.250];  
» splines(X,Y,intervals);
```





```

intervals = 5;
X=[ 0.500; 0.3232; 0.25; 0.3232; 0.500; 0.6768; 0.750; 0.6768; 0.500];
Y=[ 0.250; 0.3232; 0.50; 0.6768; 0.750; 0.6768; 0.500; 0.3232; 0.250];

```

```

figure(1)
[Px,Py]=splines(X,Y,intervals);

```

```

[Sx,Sy] =slopes(X,Y,intervals);
figure(2)
[Vx,Vy]=splines(Sx,Sy,intervals);

```

```

[Sx,Sy] =slopes(Sx,Sy,intervals);
figure(3)
[Ax,Ay]=splines(Sx,Sy,intervals);

```

