A)

```
function s=Hermite(A,n)
  coefs=zeros(n-1,4);
  for i=1:n-1
     x0=A(1,i);
     x1=A(1,i+1);
     y0=A(2,i);
     y1=A(2,i+1);
     dy0=A(3,i);
     dy1=A(3,i+1);
     coefs(i,:)=Divided_diff(x0,x1,y0,y1,dy0,dy1);
  end
  s=mkpp(A(1,:),coefs);
end
function coef=Divided_diff(x0,x1,y0,y1,dy0,dy1)
  x=[x0,x0,x1,x1];
  y=[y0,y0,y1,y1];
  ydiff=[dy0,dy1];
  n=4;
  table=zeros(n,n);
  table(:,1)=y(:);
  cnt=1;
  for j=2:n
     for i=1:n-j+1
        if(x(i+j-1)==x(i))
           table(i,j)=ydiff(1,cnt);
           cnt=cnt+1;
        else
           table(i,j)=(table(i+1,j-1)-table(i,j-1))/(x(i+j-1)-x(i));
```

```
            end
        end
    end

    coef=[table(1,1),table(1,2),table(1,3),table(1,4)];
    coef = flip(coef);
    coef(1,2) = coef(1,2) - (coef(1,1)*(x1-x0));
end
```

B)

```
%settings
f=@(x)exp(x-5);
g=@(x) sin(x);
h=@(x) log(x-3);
k=@(x)abs(x-5).^(3/2);
a=4; b=6; t=linspace(4,6);
%function f
n=3;
X=linspace(a,b,n);
Y=f(X);
A=[X;Y;Y];
s=Hermite(A,n);
```

//——————————————————————————-table for first and second interval——————————————————————————-//

```
table =

   0.367879441171442   0.367879441171442   0.264241117657115   0.103638323514327
   0.367879441171442   0.632120558828558   0.367879441171442                   0
   1.000000000000000   1.000000000000000                   0                   0
   1.000000000000000                   0                   0                   0

table =

   1.000000000000000   1.000000000000000   0.718281828459045   0.281718171540955
```

| | | | |
|---|---|---|---|
| 1.000000000000000 | 1.718281828459045 | 1.000000000000000 | 0 |
| 2.718281828459045 | 2.718281828459045 | 0 | 0 |
| 2.718281828459045 | 0 | 0 | 0 |

Now we can use the first row of the table for coefficients.

//———————————————————————————————————————————//

max(abs(ppval(s,t)-f(t))) =0.004370746538173

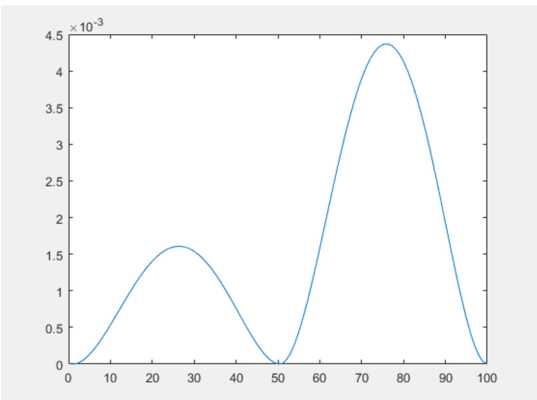
mean(abs(ppval(s,t)-f(t))) = 0.001578365909698

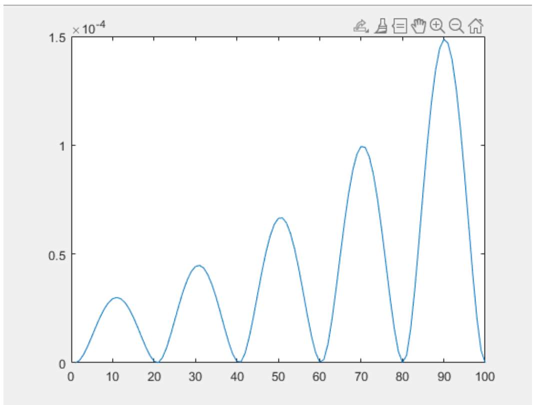
% now n=6 points,5 interval

max(abs(ppval(s,t)-f(t)))  =1.486750864114761e-04

mean(abs(ppval(s,t)-f(t))) = 4.121030630701017e-05

**Error functions**




**Observation**: As the number of interval goes from 2 to 5 the error gets smaller.

%function g

n=3;

X=linspace(a,b,n);

Y=g(X);

dY=cos(X);

A=[X;Y;dY];

s=Hermite(A,n);

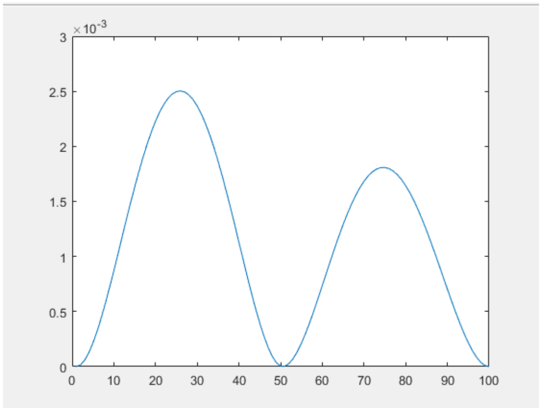
max(abs(ppval(s,t)-g(t)))=0.002503541715719

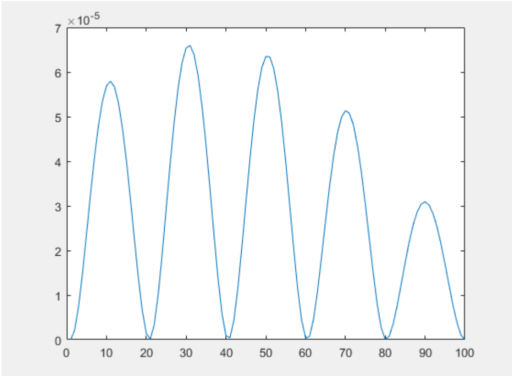
mean(abs(ppval(s,t)-g(t)))=0.001136591190992

%Now with n=6

max(abs(ppval(s,t)-g(t)))= 6.595726378433309e-05 mean(abs(ppval(s,t)-g(t)))=2.851160315315904e-05

**Error functions**




**Observation:** The error got smaller as n increased.

%function h

n=3;

X=linspace(a,b,n);

Y=h(X);

dY= 1./(X-3);

A=[X;Y;dY];

s=Hermite(A,n);

max(abs(ppval(s,t)-h(t))) = 0.003641976248656

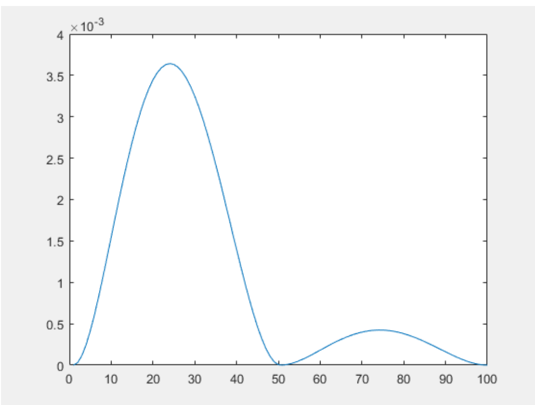
mean(abs(ppval(s,t)-h(t))) = 0.001075147047493

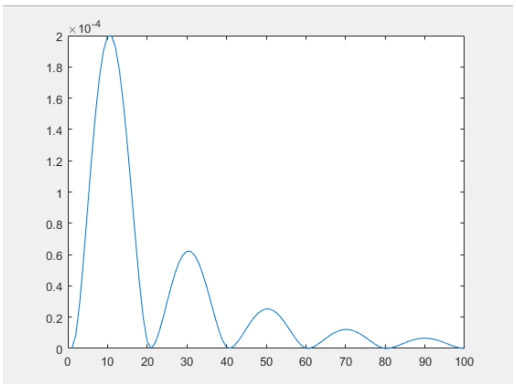
%Now with n=6

max(abs(ppval(s,t)-h(t))) = 1.999649170678075e-04

mean(abs(ppval(s,t)-h(t))) = 3.245452420306480e-05

**Error functions**



**Observation:** The error got smaller as n increased.

**% function k**

dk=@(t)(3*(t-5))/(2*sqrt(abs(t-5)));

n=3;

X=linspace(a,b,n);

Y=k(X);

dY= [-1.5,0,1.5]; %approx

A=[X;Y;dY];

s=Hermite(A,n);

max(abs(ppval(s,t)-k(t)))=0.045063155139943

mean(abs(ppval(s,t)-k(t)))=0.024750944910461

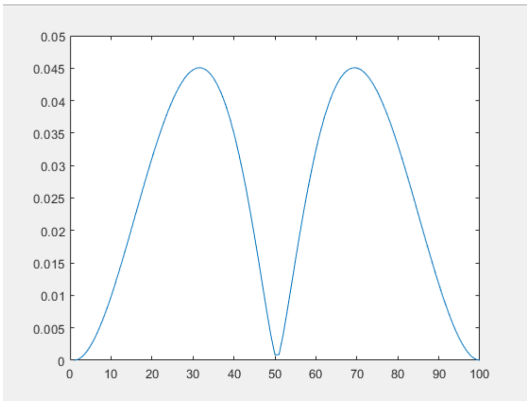
%now n=6

dY= [-1.5, -1.162,-0.671, 0.671,1.162,1.5]; %approx

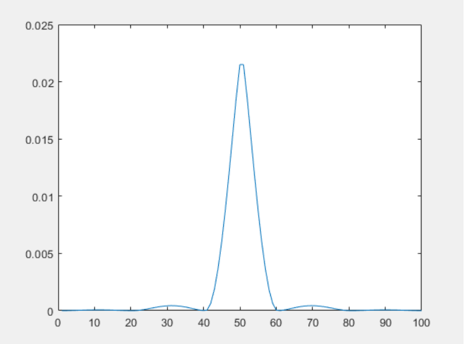
max(abs(ppval(s,t)-k(t))) = 0.021498685392098

mean(abs(ppval(s,t)-k(t))) = 0.001872700152697

**Error functions**




→




**Observation:** The error got smaller as n increased.

In theory, when we use cubic Hermite interpolation, regardless of what function we are interpolating, it should converge. It is because the error formula is 5/384*max(f””)[a,b]*h^4 and h always gets smaller when the number of interval increases. As we can see above when n increased the error got smaller for all 4 functions, So **convergence is quite certain.** If we compare with HW5’s results, cubic Hermite interpolation has lower errors than cubic spline interpolation. However, since Hermite interpolation requires derivates for each interval, the cost is more expensive.