
Table of Contents

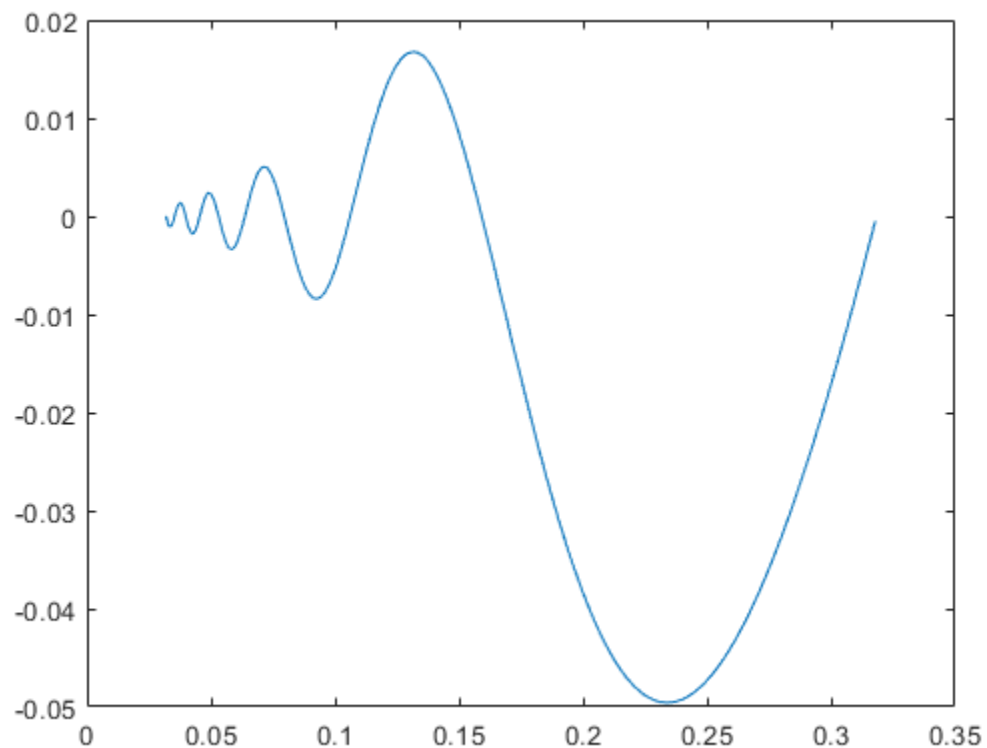
Damià Casas Casajuana p3	1
a)	1
b1)	2
b2)	3
c)	4

Damià Casas Casajuana p3

```
close all;  
clear all;  
format long g;
```

a)

```
xrepr = [1/(10*pi):1/1000:1/pi];  
yrepr = (xrepr.^2).*sin(1./xrepr);  
figure();  
plot(xrepr, yrepr);
```



b1)

```
% Utilitzo els nodes Chebychev:
N = 10;
j = [0:1:N+1];
a = 1/(10*pi);
b = 1/pi;
aux = cos((j*pi)/(N+1));
xCheby = a + ((b-a)/2)*(1+ aux);

% 1000 punts equiespaciats:
z = linspace(a, b, 1000);
funCheby = (xCheby.^2).*sin(1./xCheby);

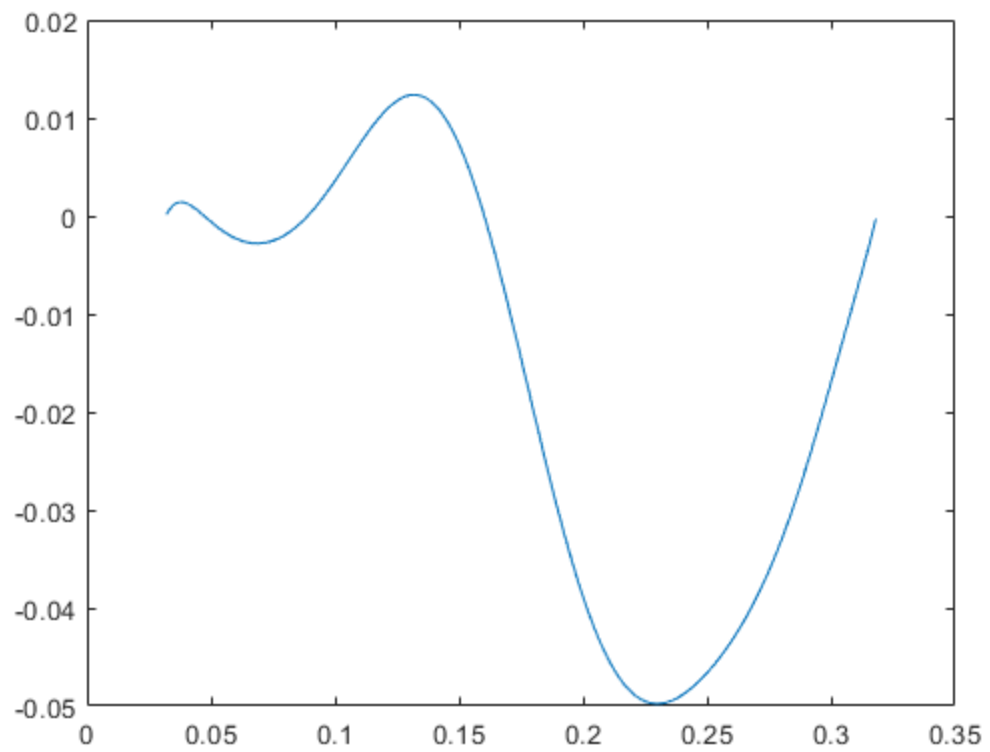
% Aplico directament la interpolacio baricentrica:

%function B = baricentrica2(z, xCheby, funCheby)
n = length(xCheby);
m = length(z);
den = zeros(length(z),1);
num = zeros(length(z),1);

for j = [1:1:n]
    if j == 1 || j == n
        numj = ((-1)^j)*funCheby(j)*(z-xCheby(j)).^(-1).*(1/2);
        num = num + numj';
        denj = ((-1)^j)*(z-xCheby(j)).^(-1).*(1/2);
        den = den + denj';
    else
        num = num + (((-1)^j)*funCheby(j)*(z-xCheby(j)).^(-1)))';
        den = den + (((-1)^j)*(z-xCheby(j)).^(-1)))';
    end
end

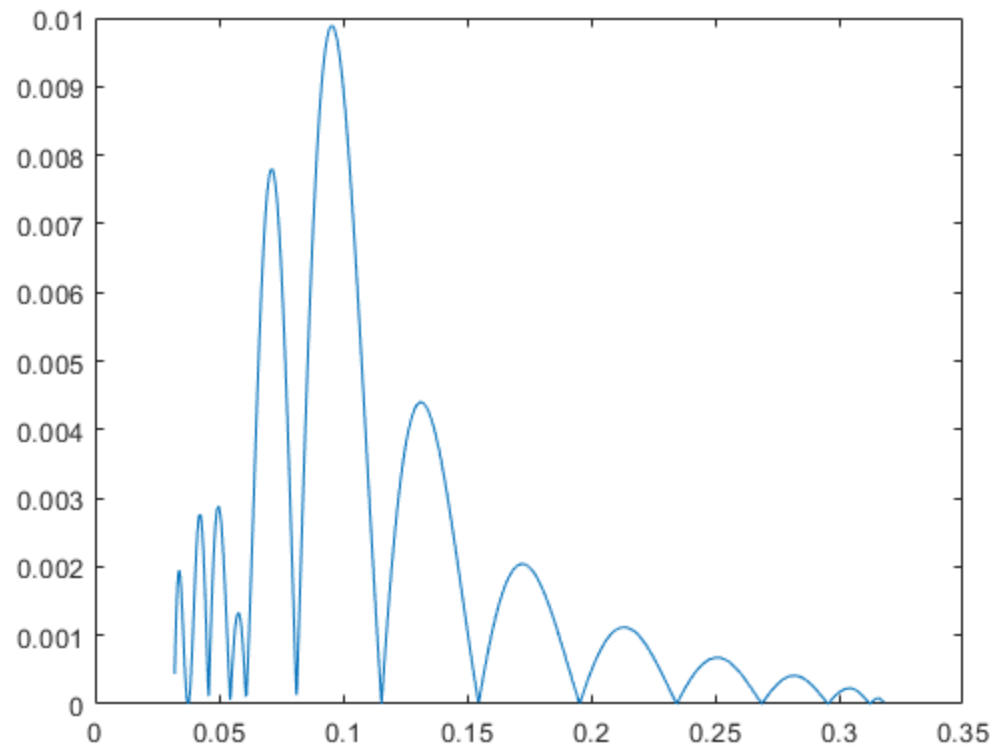
B = num./den;

% Finalment represento:
figure;
plot(z, B);
```



b2)

```
funz = (z.^2).*sin(1./z);  
error = abs(funz- B');  
figure;  
plot(z, error);
```



c)

```
errors = [];  
  
funz = (z.^2).*sin(1./z);  
  
for N = 10:5:130  
    % Aquesta part ha sigut copiada de l'apartat b.  
    j = [0:1:N+1];  
    a = 1/(10*pi);  
    b = 1/pi;  
    aux = cos((j*pi)/(N+1));  
    xCheby = a + ((b-a)/2)*(1+ aux);  
  
    % 1000 punts equiespaciats:  
    z = linspace(a, b, 1000);  
    funCheby = (xCheby.^2).*sin(1./xCheby);  
  
    % Aplico directament la interpolacio baricentrica:  
  
    %function B = baricentrica2(z, xCheby, funCheby)  
    n = length(xCheby);  
    m = length(z);  
    den = zeros(length(z),1);
```

```

num = zeros(length(z),1);

for j = [1:1:n]
    if j == 1 || j == n
        numj = ((-1)^j)*funCheby(j)*(z-xCheby(j)).^(-1).*(1/2);
        num = num + numj';
        denj = ((-1)^j)*(z-xCheby(j)).^(-1).*(1/2);
        den = den + denj';
    else
        num = num + (((-1)^j)*funCheby(j)*(z-xCheby(j)).^(-1)))';
        den = den + (((-1)^j)*(z-xCheby(j)).^(-1)))';
    end
end

B = num./den;
error = abs(funz- B');
errors = [errors max(error)];
end
figure;
semilogy([10:5:130], errors);
hold off;

% Com es pot observar a la grafica, podem assolir un error mínim de
% 10^-16,
% que és exactament el error de la màquina. Aquest error es trobarà
% quan la
% N sigui igual a 110 i a partir d'aquesta N el error s'estabilitzarà
% (mai
% serà inferior al de la màquina).

Undefined function 'hold' for input arguments of type 'char'.

Error in p3DamiaCasas (line 104)
hold off;

```

Published with MATLAB® R2018b