Practica_18_Casas_Mercade

Table of Contents

Section A

Last page of the pdf

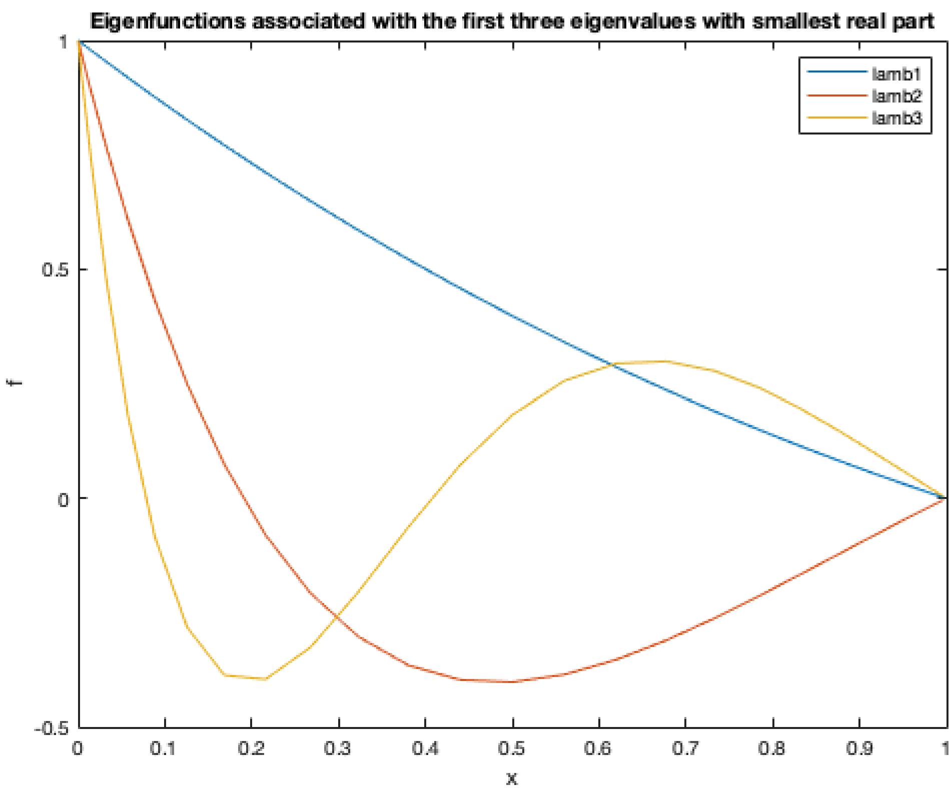
Section B

```
clear all;
clc;
close all;

g=1;
l = 1;
% Functions of the ODE
p = @(x)(x);
q =@(x)(0.*x + 1);
r = @(x)(0.*x);
n = 26;
a=0; b=1;
C=[0, 0, 0; l, 0, 0];


[F, x, lamb] = resoldreODEeig(n, C, p, q,r, a, b);
disp('The first three eigenvalues with smallest real part');
lamb = sqrt(lamb);
disp(lamb(1:3))

%The codes used to solve the ODE are the followings:

%{
function [M1,M2,M3,Lhat, x] = crearMatriusODE(n, C, p, q, r,a,b)
% Input: n: nombre de nodes Chebyshev,
% C: Matriu 2*3 amb les Robin conditions
% Funcions p, q, r:  En cas que sigui una funcio constant, cal que
 apareixi la x en el function handler

% Fem la diferenciacio Chebyshev, que ens retorna tambÃ© els nodes:
[D,x] = chebdiff(n,a,b);
P = diag(p(x));
Q=diag(q(x));
R=diag(r(x));
L=P*D^2+Q*D+R;
```

```
%Cas en que nomes tenim una condicio de Robin corresponent a x=1.
 Haurem de
%treure el primer element ja que els nodes chebyshev estan al reves.

Lhat=L(2:end,2:end); %treiem primera fila i primera columna de L


M1=-D(1,2:end); %Primera fila de D, sense agafar element de la primera
 columna


M2 = [C(2,1) + C(2,2)*D(1,1)];

M3= [L(2:end,1)];

end
%}


%{
function [F, x, lamb] = resoldreODEeig(n, C, p, q,r, a, b)
% Les funcions entregades han de ser les corresponents al domini -1, 1
% Totes les funcions han ser aptes per vectors
[M1,M2,M3,Lhat, x] = crearMatriusODE(n, C, p, q, r, a, b);
mCoef = [ C(2,2) ];
esquerra = Lhat + M3*inv(M2)*mCoef*M1;

[EVEC, EVAL] = eig(-esquerra);
lamb = diag(EVAL); % Lamb es un vector perq diag funciona en les dos
 direccions de conversio.
[foo,ii] = sort(lamb) ; lamb = lamb(ii) ; EVEC = EVEC(:,ii);
eval=diag(lamb);

F = EVEC';
end
%}



%Once we have the eigenvectors we normalize, make them positive and
 plot the first three of
%them

f = [];
for i=1:3
    if abs(max(F(i,:)))< abs(min(F(i,:)))
        f(i,:)=F(i,:)/min(F(i,:));
    else
        f(i,:)=F(i,:)/max(F(i,:));
    end
end
```

```
figure;
plot(x(2:end),f)
title('Eigenfunctions associated with the first three eigenvalues with
 smallest real part')
legend('lamb1','lamb2','lamb3')
xlabel('x')
ylabel('f')
graphLamb = lamb(1:3);
```

```
The first three eigenvalues with smallest real part
    1.2024
    2.7600
    4.3269
```



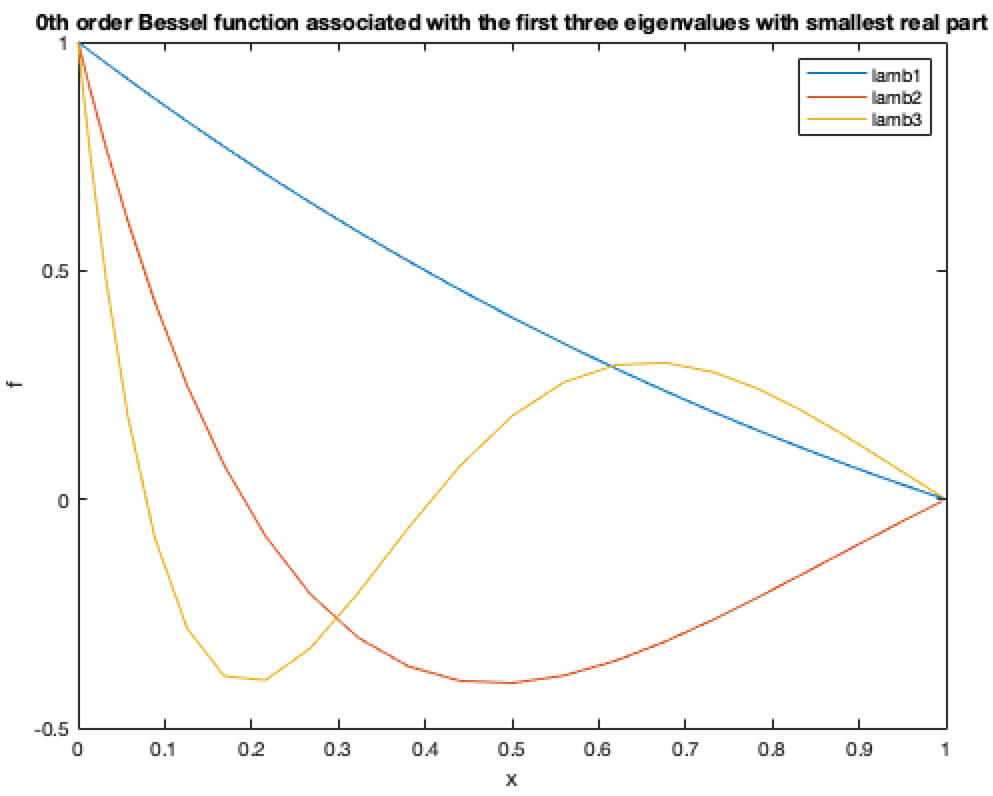
Section C

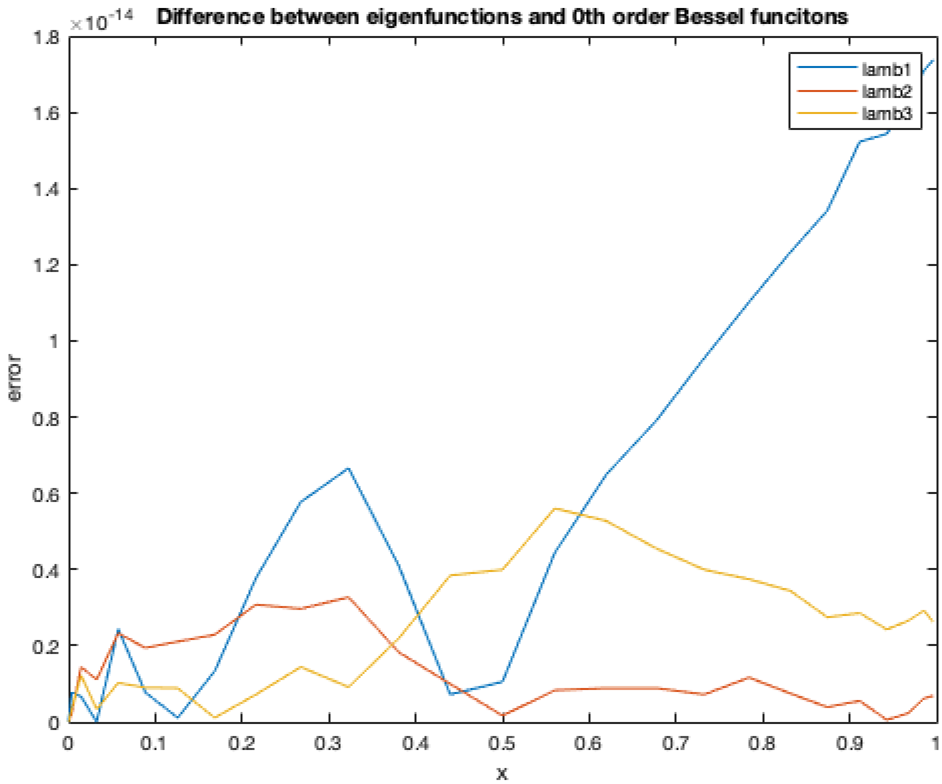
```
% Now we check that the 0th order Bessel functions evaluated in the
% specific nodes compute below (z are the solution to the ODE:
z = 2.*lamb(1:3)'.*sqrt(x(2:end));
y = besselj(0, z);
figure;
plot(x(2:end),y)
title('0th order Bessel function associated with the first three
 eigenvalues with smallest real part')
legend('lamb1','lamb2','lamb3')
```

```
xlabel('x')
ylabel('f')

%The plot of the bessel functions is exactly the same that we obtained
%using the eigenvectors.


%Now we check if using the nodes 2*lamb the bessel funcitons is 0.
j=2*lamb(1:3);
J = besselj(0, j);
disp('The 0th order Bessel functiona of first kind, and their
 associated eigenvalues lamb_k satisfy J(2*lamb)=0')
disp(J)

% We now compute the error:
figure;
error = abs(f-y');
plot(x(2:end), error);
title('Difference between eigenfunctions and 0th order Bessel
 funcitons')
legend('lamb1','lamb2','lamb3')
xlabel('x')
ylabel('error')

% Using the Newton iteration we can find the nodes for the three first
% lambdas:

itmax = 100;
tol = 1e-10;
% As there will be always a 0 at x = 1:
XJs = flip(x(2:end));


% Finally we find the nodes. As we have seen that using evaluatinf the
% bessel function on  2*lamb  the result is 0, we find the nodes (x)
 for each
% lambda with: lambdai = lambdaj*sqrt(x)

for i = 1:3
   for j = 1:i
       disp(strcat('Lambda ' , int2str(i) , ' nodes:'))
       nodeAt = (lamb(j)./lamb(i)).^2;
       disp(nodeAt);
   end
end
```

```
The 0th order Bessel functiona of first kind, and their associated
 eigenvalues lamb_k satisfy J(2*lamb)=0
   1.0e-13 *

   -0.1741
    0.0084
   -0.0301
```

```
Lambda1 nodes:
     1

Lambda2 nodes:
    0.1898

Lambda2 nodes:
     1

Lambda3 nodes:
    0.0772

Lambda3 nodes:
    0.4069

Lambda3 nodes:
     1
```





Published with MATLAB® R2018b