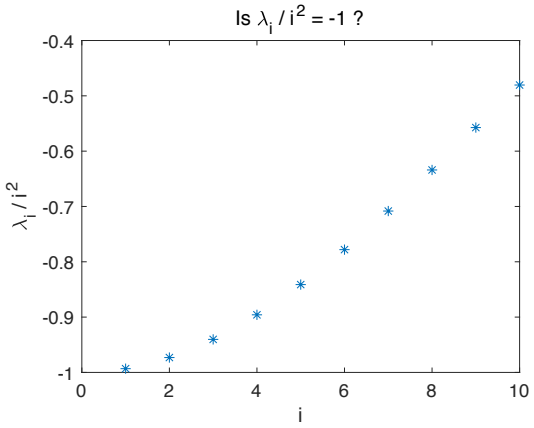

Figure 1: Eigenvalues of the differential eigenvalue problem (A.1), scaled by i^2 so that the analytical result is -1.

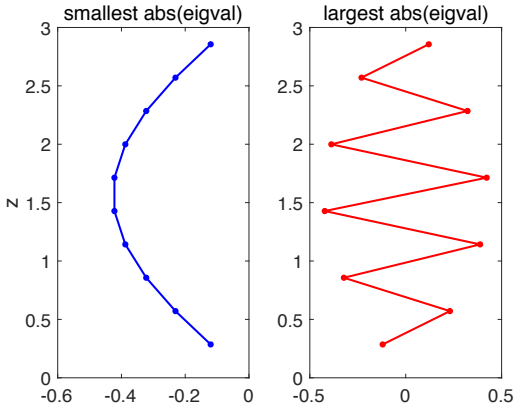

Figure 2: Eigenvectors corresponding to the smallest and largest (absolute) eigenvalues.

3: Differential eigenvalue problem

(a) Eigenvalues of the differential equation are $-i^2; i = 1, 2, ..., N$.

(b) When you solve the matrix eigenvalue problem, the first few eigenvalues are close to -1, -4, ..., but things degenerate as you go to higher eigenvalues. This is because the approximate derivative in ddz2 is only accurate for functions that are smooth enough to be well-resolved by the number of z values you've used. To be more specific, the truncation error is approximately proportional to $\Delta^2 f^{(4)}$. If there's a lot of small-scale structure, the fourth derivative can make this error large even when Δ is small. Also, higher-order terms can contribute to the error.

For N=10, you get the eigenvalues shown in figure 1. These eigenvalues have been sorted and normalized by i^2 so that each would equal -1 if the solution were perfectly accurate. The first few eigenvalues are fairly accurate, but things degenerate fast after that. The final eigenvalue is wrong by a factor of two!

The eigenvector corresponding to $i = 1$ ($\sigma = -0.993$) is shown on the left in figure 1. It is clearly well resolved by the ten z values. Note that it is headed for zero at $z = 0$ and $z = \pi$, i.e. it obeys the boundary conditions.

On the right is the eigenvector corresponding to $i = 10$. This function is very badly resolved, i.e. there is too much small-scale structure to be accurately represented by ten points, and the truncation error in ddz2 is

expected to be large. This is why the eigenvalue was inaccurate.

The moral of the story is, always keep in mind that the differential eigenvalue problem that you want to solve and the algebraic eigenvalue problem that you actually do solve are two different things. A well-designed algebraic equation should give good approximations to solutions of the corresponding differential equation, but you must always interpret the results with care. If the function you get is not well resolved by your spatial discretization (i.e. if the scale on which it varies is not much bigger than Δ, as in the red curve shown above), it is probably not a good approximate solution to the differential equation.

Here is the code we used:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% HMWK Problem 3
%
clear
close all
fs=20;

% define z values
N=10;
z=pi*[1:N]'/(N+1);

% compute derivative matrix
d=ddz2(z);
dz=z(2)-z(1);

%%%
% Implement boundary conditions.
% (To use 1-sided derivatives, comment out the next three lines.)
d(N,:)=0;d(N,N-1)=1/dz^2;d(N,N)=-2/dz^2;
d(1,:)=0;d(1,1)=-2/dz^2;d(1,2)=1/dz^2;
%%%

% compute eigvals & eigvecs
[v ee]=eig(d);e=diag(ee);

% sort
[~,ind] =sort(abs(e),'ascend');
e=e(ind);
v=v(:,ind);

% Plot eigenvalues /i^2.
% If the eigfn is well-resolved, this will be close to -1.
figure
plot([1:N],e./[1:N].^2','*','markersize',10)
xlabel('i','fontsize',fs);
ylabel('\lambda_i / i^2','fontsize',fs)
title('Is \lambda_i / i^2 = -1 ?','fontsize',fs,'fontweight','normal')
set(gca,'fontsize',fs-2)
```

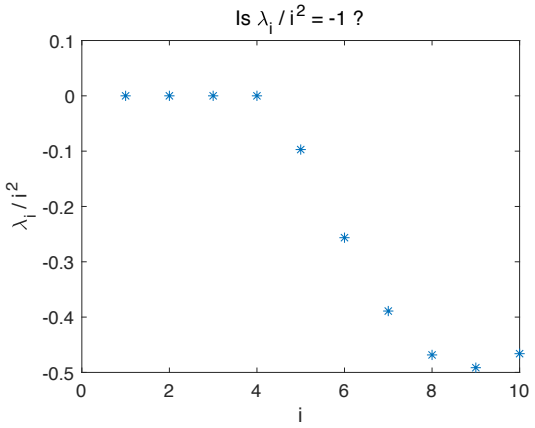

Figure 3: Eigenvalues using 1-sided derivatives at the boundaries.

```
% Plot first and last eigvecs.
% The first is well-resolved, and its eigval is close to -i^2.
% The last is poorly-resolved, and the eigval is not close to -i^2.
figure
subplot(1,2,1)
plot(v(:,1),z,'b*')
hold on
plot(v(:,1),z,'b')
ylabel('z','fontsize',fs)
title('smallest abs(eigval)','fontsize',fs,'fontweight','normal')
set(gca,'fontsize',fs-2)

subplot(1,2,2)
plot(v(:,end),z,'r*')
hold on
plot(v(:,end),z,'r')
title('largest abs(eigval)','fontsize',fs,'fontweight','normal')
set(gca,'fontsize',fs-2)
```

Replacing the boundary conditions with one-sided derivatives gives utter nonsense. The eigenvalues are nothing like $-i^2$. Looking at the eigenvectors, one can see why - they do not obey the boundary conditions.

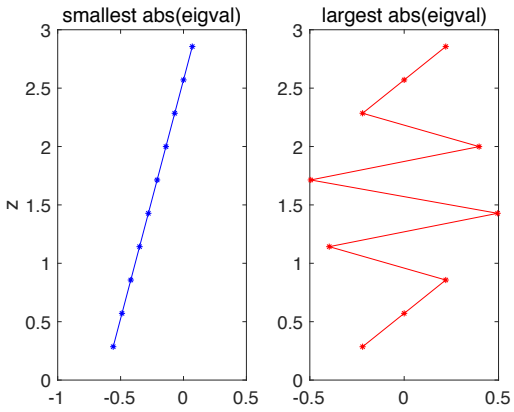

Figure 4: Eigenvectors using 1-sided derivatives at the boundaries.