```
%
%  BesselMovie.m -- djm -- 03 feb 2019
%
% - this script make an animated gif
%  file that is a solution to the
%  'wave equation' on a disc
% - you do NOT have to understand this
%  script at all
% - to use it, the matrix Zmk with size
%  16x16 must be defined by YOUR code

%  REMOVE THIS "clear"!

%  bessel (integer) index
kP = 16;  dr  = 1/(kP+1);
mP = kP;  dth = pi/mP;

%  INSTRUCTOR SOLUTION CODE
%  bessel_roots

if (exist('Zmk','var')==0)
 %  proxy matrix of bogus values
 Zmk = transpose((1:16)*1.3)*ones(1,16);
else
 %  check if student matrix is 16x16
 if (sum(size(Zmk)==[16 16])~=2)
  disp('matrix is not correct size!')
  return
 end
end

th = (0:2*mP-1)*dth;
rr = (1:kP)*dr;

[rg,tg] = meshgrid(rr,th);
[xg,yg] = pol2cart(tg,rg);

%  make initial condition
pk = 16;
yy = exp(-pk*((xg-0.5).^2 + yg.^2)).*((1 - rg.^4).^2);
yp = -2*pk*yg.*yy ...
     -8*exp(-pk*((xg-0.5).^2 + yg.^2)).*(1 - rg.^4).*(rg.^2).*yg;

%  fourier transforms
yyfft = fft(yy,[],1);
ypfft = fft(yp,[],1);

%  build coefficients

Cmk = zeros(mP,kP);
Dmk = zeros(mP,kP);
```

```
for mm = 0:mP-1;
 for kk = 1:kP
  alpha = Zmk(mm+1,kk);
  Cmk(mm+1,kk) = sum(yyfft(mm+1,:).*besselj(mm,alpha*rr).*rr)*dr;
  Dmk(mm+1,kk) = sum(ypfft(mm+1,:).*besselj(mm,alpha*rr).*rr)*dr;

  bprime = (besselj(mm+1,alpha) - besselj(mm-1,alpha)) / 2;

  if (mm==0)
   Cmk(mm+1,kk) = 1*Cmk(mm+1,kk)/(bprime^2);
   Dmk(mm+1,kk) = 1*Dmk(mm+1,kk)/(bprime^2);
  else
   Cmk(mm+1,kk) = 2*Cmk(mm+1,kk)/(bprime^2);
   Dmk(mm+1,kk) = 2*Dmk(mm+1,kk)/(bprime^2);
  end
 end
end

%  plot coordinates
[rp,tp] = meshgrid([0 rr 1],[th 2*pi]);
[xp,yp] = pol2cart(tp,rp);

%  zero check plot
test = zeros(size(rp));

for mm = 0:mP-1
 for kk = 1:kP
  test = test + Cmk(mm+1,kk)*besselj(mm,Zmk(mm
+1,kk)*rp).*exp(1i*mm*tp);
 end
end

%  Don't create any figures if 'nofigs' variable exists and is true
if(exist('nofigs','var') == 1)
    if nofigs == true
        return
    end
end
figure(3000);  clf
plot(real(test(:,end))/mP,'kx')
title('this plot should show numerical zeros! (what does this mean?)')
xlabel(['\theta-axis (max abs val = '
 num2str(max(abs(real(test(:,end))/mP))) ')'])
ylabel('surface values at r=1 as a function of \theta')

%  initialize for animated gif
h = figure(3001);
axis tight manual % this ensures that getframe() returns a consistent
 size
filename = 'BesselMovie.gif';

%  make animation
figure(3001);  clf
for tt = [(0:48)*0.1]
```

```
    soln = zeros(size(rp));
    for mm = 0:mP-1
        for kk = 1:kP
            next = Cmk(mm+1,kk)*besselj(mm,Zmk(mm
+1,kk)*rp).*exp(1i*mm*tp);
            soln = soln + next*cos(Zmk(mm+1,kk)*tt);
            next = Dmk(mm+1,kk)*besselj(mm,Zmk(mm
+1,kk)*rp).*exp(1i*mm*tp);
            soln = soln + next*sin(Zmk(mm+1,kk)*tt)/Zmk(mm+1,kk);
        end
    end

    surf(xp,yp,real(soln)/mP);  hold on
    plot3(xp(:,end),yp(:,end),real(soln(:,end)/mP),'b','linewidth',3);
  hold off
    title(['waves on a disc, (time = ' num2str(tt) ')'])
    xlabel(['x-axis : ' num2str(sum(Zmk(:,end)/pi))])
    cl = clock;
    zlabel(['z-axis ; ' num2str(cl(3)) '.' num2str(cl(4)) '.'
 num2str(cl(5))])
    ylabel('y-axis')
    axis equal
    axis([-1 1 -1 1 -1 1])
    caxis([-1 1])
    colormap(hot)
    colorbar
    drawnow

    % Capture the plot as an image
    frame = getframe(h);
    im = frame2im(frame);
    [imind,cm] = rgb2ind(im,256);

    % Write to the GIF File
    if (tt==0)
        imwrite(imind,cm,filename,'gif', 'Loopcount',1);
    else
        imwrite(imind,cm,filename,'gif','WriteMode','append');
    end
end
```

Published with MATLAB® R2018a