Assignment 3: Animating Warping Surfaces

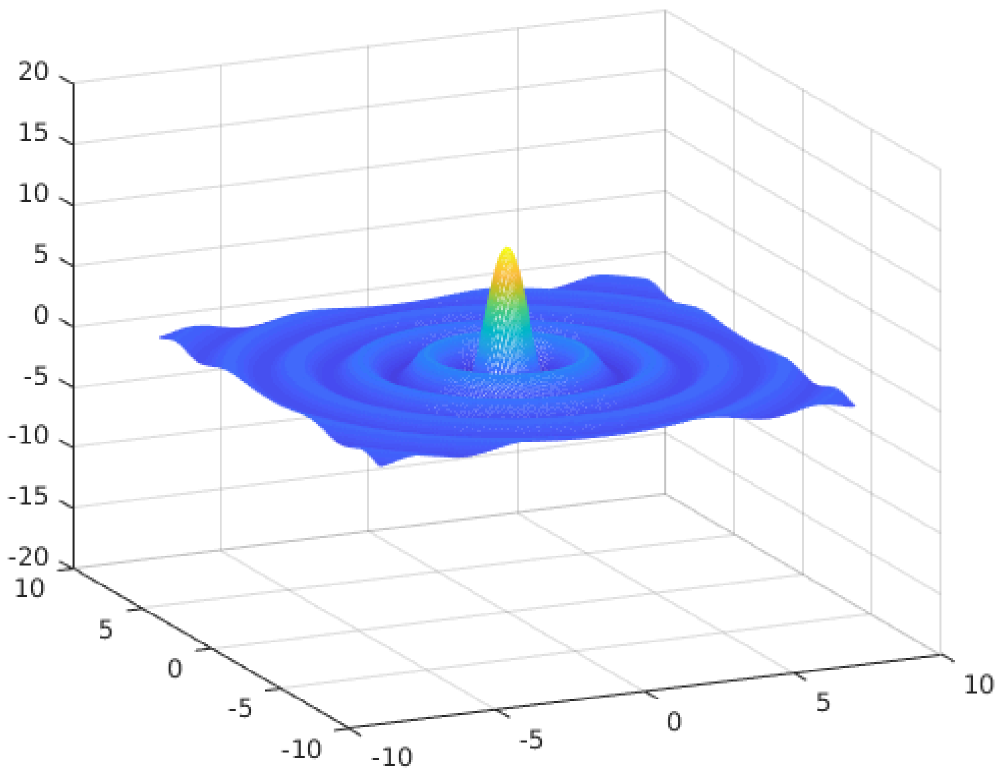
Table of Contents

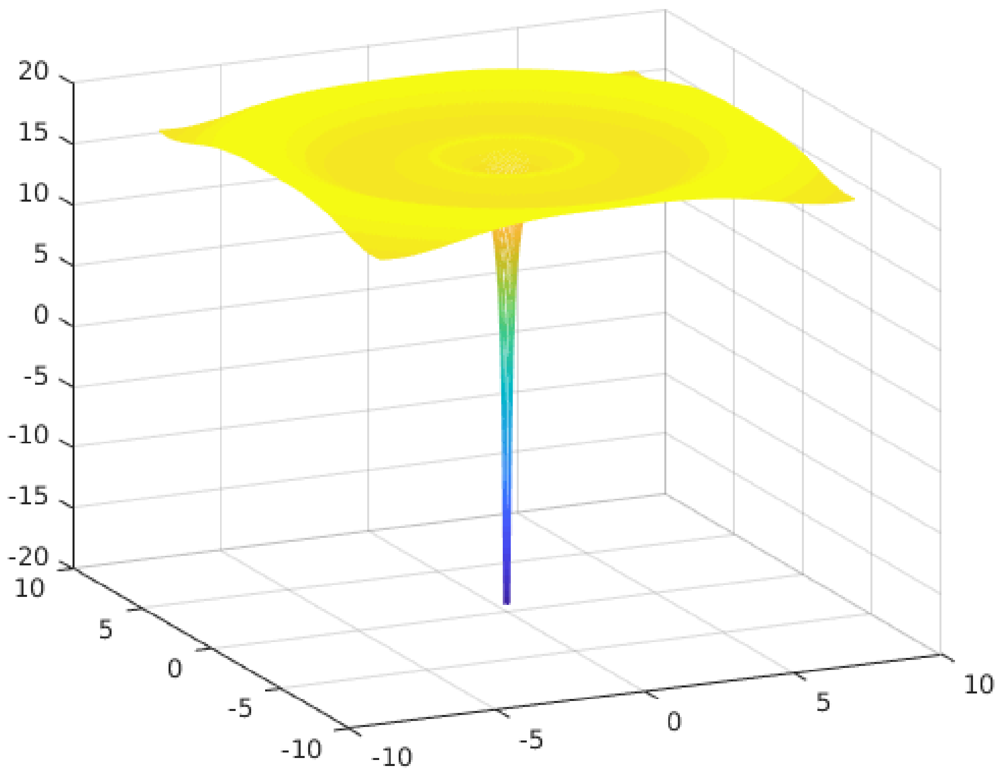
Robert Moir: 0123456789

Initial Plotting of the Surfaces

```
% x and y boundary value
val = 8;
% set x, y and z boundary values
xmin = -val; xmax = val;
ymin = -val; ymax = val;
zmin = -20; zmax = 20;
% number of points for plots
npoints = 201;
% positions for text
xpos1 = 1.1*xmin;
ypos1 = 0.8*ymin;
zpos1 = 0.8*zmin;
xpos2 = 1.2*xmin;
ypos2 = 0.6*ymin;
zpos2 = zpos1;
% The two surfaces to interpolate
f1 = @(x,y)(10*sinc(sqrt(x.^2+y.^2)));
%f1 = @(x,y)(10*(sin(pi*sqrt(x.^2+y.^2))./(pi*sqrt(x.^2+y.^2))));
f2 = @(X,Y)(18-3./(sqrt(X.^2+Y.^2))+sin(sqrt(X.^2+Y.^2))+...
    sqrt(200-(X.^2+Y.^2)+10*sin(X)+10*sin(Y))/1000);
xs = linspace(xmin,xmax,npoints);
ys = linspace(ymin,ymax,npoints);
[X,Y] = meshgrid(xs,ys);
% plot surface 1
Z1 = f1(X,Y);
mesh(X,Y,Z1)
shading interp
view(-25,20)
zlim([zmin,zmax])
% compute integral of surface 1 for use in next part
num_area1=integral2(f1,xmin,xmax,ymin,ymax);
print 'f1.png' -dpng
% plot surface 2
figure
Z2 = f2(X,Y);
```

```
mesh(X,Y,Z2)
rotate3d on
shading interp
view(-25,20)
zlim([zmin,zmax])
% compute integral of surface 2 for use in next part
num_area2=integral2(f2,xmin,xmax,ymin,ymax);
print 'f2.png' -dpng
```



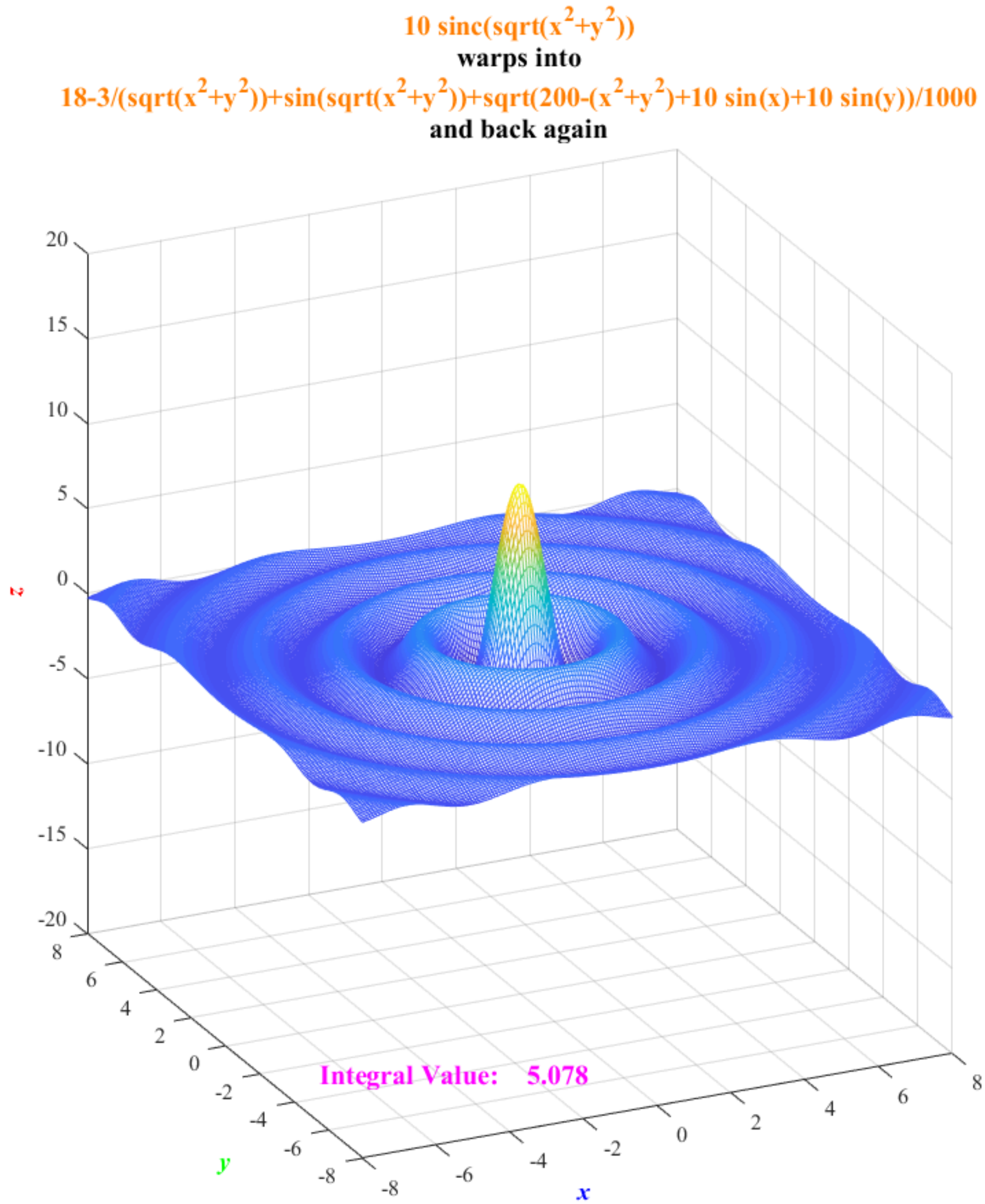


Animation of Surface Warping

```
% setup figure window size based on screen size
set(0,'units','pixels');
screenSizePixels=get(0,'screensize');
screenWidth=screenSizePixels(3);
screenHeight=screenSizePixels(4);
figureAspectRatio=16/14; % height to width
figureHeight=screenHeight*0.85;
figureWidth=figureHeight/figureAspectRatio;
% shift left 5% of the screen width
leftx=screenWidth*0.05;
% shift up 15% of the screen height
lefty=screenHeight*0.15;
ha=figure;
set(ha,'Position',[leftx lefty figureWidth figureHeight]);

% setup animation sequence
delta_t=0.01;
time = 0:delta_t:1;
time = [time 1-delta_t:-delta_t:0];
for t=time
    % interpolate the surfaces
    Z=Z1*(1-t)+Z2*(t);
```

```
    % plot current surface
    mesh(X,Y,Z)
    % reset axis object
    axis([xmin xmax ymin ymax zmin zmax]);
    % print current integal by interpolating integrals computed above
    text(xpos1,ypos1,zpos1,['\fontsize{15}\bf \color{magenta} ' ...
        'Integral Value: ' ...
        sprintf('%8.3f',num_area1*(1-
t)+num_area2*t)],'FontName','Times');
    % label axes
    xlabel('\it\bf x','color','blue');
    ylabel('\it\bf y','color','green');
    zlabel('\it\bf z','color','red');
    set(gca,'FontName','Times','FontSize',12);
    % print specialized title
    title({'{\color{orange}10 sinc(sqrt(x^2+y^2))}'; 'warps into';...
        ['{\color{orange}18-3/(sqrt(x^2+y^2))+sin(sqrt(x^2+y^2))+',...
        'sqrt(200-(x^2+y^2)+10 sin(x)+10 sin(y))/1000}'];...
        'and back again'},'FontSize',15);
    shading interp
    view(-25,20)
    drawnow;
    % condition to print flower surface in middle of animation
    if (t == 1)
        pause(1)
        print 'f2_frame.png' -dpng
    end
end % for t
% print water surface at end of animation
print 'f1_frame.png' -dpng
```



Symbolic Integration

From the fact that MATLAB returns the input expressions for the double integrals, we see that MATLAB is not able to evaluate the integrals of f_1 and f_2 to a closed form expression. [This is sufficient for a basic response to this question].

A more sophisticated response would point out that the failure of MATLAB to produce a response does not mean that a closed form integral is not computable, just that MATLAB does not succeed in finding one. In fact, for the functions we are dealing with, spaces of functions are known in which the integral exists, but algorithms for computing the integral may be too computationally expensive, have not yet been implemented, or may be a subject of current research. [This nice, but is not required to receive full marks for this question.]

There is more subtlety to the situation, though, as MATLAB is able to produce a numerical answer, which is different from the value of the numerial integral of the same function. Thus, MATLAB is able to produce a numerical result if one points out that one is looking for a double value, rather than an exact symbolic result that evaluates to a number. I believe that the reason for this is that MATLAB is able to compute symbolically the single integrals, it is just the double integral it cannot compute. So it may be using the fact that a single integral is available to provide a tight estimate of the value of the double integral. The computation of a numerical value is illustrated by the line of code 'double(sym_area1)' below. Notice that the symbolic value is different from the numerical value. This is not computed for 'sym_area2' as the computation takes too long.

```
format long
syms x y
f1=10*sinc(sqrt(x^2+y^2));
sym_area1=eval(int(int(f1,y,ymin,ymax),x,xmin,xmax))
disp('sym_area1')
double(sym_area1)
disp('num_area1')
num_area1
f2=18-3/sqrt(x^2+y^2)+sin(sqrt(x^2+y^2))+...
    (sqrt(200-(x^2+y^2)+10\sin(x)+10\sin(y)))/1000;
sym_area2=eval(int(int(f2,y,ymin,ymax),x,xmin,xmax))
```

```
sym_area1 =

int(int((10*sin(pi*(x^2 + y^2)^(1/2)))/(pi*(x^2 + y^2)^(1/2)), y, -8,
 8), x, -8, 8)

sym_area1

ans =

   5.077690272301118

num_area1

num_area1 =

   5.077690264880984


sym_area2 =

int(int(sin((x^2 + y^2)^(1/2)) - 3/(x^2 + y^2)^(1/2) + (sin(x)/10 +
 sin(y)/10 - x^2 - y^2 + 200)^(1/2)/1000 + 18, y, -8, 8), x, -8, 8)
```

Published with MATLAB® R2018a