

```

% evolveCurve: evolve the given shape
%
% Input parameters:
%   shape = the shape to be evolved. It can be any value of ('circle',
%   'dumbbell').
%   speed = constant speed
%   t0 = initial time
%   tMax = final time
%   m = matrix size
function [phi grid phi0] = evolveCurve(shape, speed, tMax, m)

%-----
% Initialize evolution parameters
plotStep = 0.05;           % plot at t = 0, 0.05, 0.1, 0.15, 0.2, and 0.25
t0 = 0;                    % Start at time t = 0

% Don't calculate the error by default
calculateError = false;

% Colors array for plotting
colors = ['r' 'g' 'b' 'c' 'm' 'k'];

% Start time
startTime = cputime;

%-----
% Determine which curve to evolve
if(strcmpi(shape, 'circle'))
    grid = constructGrid(m);
    phi0 = shapeCone(0.15, [0.5 0.5], grid);
    calculateError = true;
elseif(strcmpi(shape, 'dumbbell'))
    plotStep = 0.025;
    grid = constructGrid(m);
    circle1 = shapeCone(0.15, [0.25 0.5], grid);
    circle2 = shapeCone(0.15, [0.75 0.5], grid);
    rectangle = shapeRectangle(grid, [0.25 0.45], [0.75 0.55]);
    phi0 = shapeUnion(circle1, rectangle, circle2);
else
    error('??? Error. Unknown shape.');
```

end

```

% Initialize phi with phi0
phi = phi0;

%-----
% Evolve the curve
t = t0;
deltaT = (grid.upperRightCorner(1) - grid.lowerLeftCorner(1)) / m;

% There's a little round off here, because while testing the code for m =
% 80, the sixth plot was not shown. Therefore, I guessed if I increased the
% terminating condition of the loop, maybe it will be able to plot it. And
% it did!
while(t <= tMax + 100 * eps)
    % plot at t = 0, 0.05, 0.1, 0.15, 0.2, and 0.25
    if(mod(t, plotStep) == 0)
        index = mod(int32(t/plotStep), size(colors, 2)) + 1;
        contour(grid.axes{1}, grid.axes{2}, phi, [0 0], colors(index));
        % Draw on the same plot
        hold on;
    end
    phi = finiteDifference(phi, speed, deltaT, grid);
end

```

```

    t = t + deltaT;
end

% Create the legend
legendHandle = legend('t = 0', 't = 0.05', 't = 0.1', 't = 0.15', 't = 0.2', 't = 0.25',
'Location', 'NorthEastOutside');
set(legendHandle, 'FontWeight', 'bold');

% Release the hold on the plot
hold off;

if(calculateError)
    % The exact area of the circle if we calculate it analytically.
    % Since  $F = 1$ , this means at any given time  $t$ , the radius should increase by
    %  $(t - t_0)$  to keep the ratio  $dx/dt = 1$  (i.e.  $F = 1$ ). So at  $t_{\text{Max}}$  the
    % radius would be  $= 0.15 + t_{\text{Max}} - t_0$ .
    exactArea = pi * (0.15 + tMax - t0)^2;

    % The area calculated using the final phi
    phiArea = dblquad(@(x,y) heaviside(-1*(phi(ceil(x*(m-1)) + 1, ceil(y*(m-1)) + 1))),
grid.lowerLeftCorner(1) , grid.upperRightCorner(1), grid.lowerLeftCorner(2), grid.
upperRightCorner(2));

    fprintf('AreaPhi - AreaExact = %g - %g = %g\n', exactArea, phiArea, exactArea -
phiArea);
end

% End time
endTime = cputime;

% Total execution time
fprintf('Total execution time = %g seconds.\n', endTime - startTime);

```