1 Matlab correction

1.1 Binary image generation

```
n = 512+256; % size of image
R  =200; % radius of circle
[X, Y] = meshgrid(-n/2:n/2-1, -n/2:n/2-1);
I = double(X.^2+Y.^2 >= R^2);
```

1.2 Initial contour

The choice of the initial contour is crucial in this method. The parameters used in this example ensure the convergence of the snake.

```
% ellipse formula
x = n/2 + 300 * cos(0:.1:2*pi);
y = n/2 + 150 * sin(0:.1:2*pi);

% in order to be consistent with the following code, one has to take the
    ↪ transposed vector.
x = x';
y = y';
```

The different parameters are defined by:

```
alpha = .00001;
beta  = .05;
gamma = 200;
iterations = 1000;
```

1.3 Matrix construction

This is maybe the hardest part of this code, with the use of the spdiags function.

```
N = length(x);
X = [-beta alpha+4*beta -2*alpha-6*beta alpha+4*beta -beta -beta alpha+4*
    ↪ beta -beta alpha+4*beta];
B = repmat(X, N, 1);
A = full(spdiags(B, [-2 -1 0 1 2 N-2 N-1 -N+2 -N+1], N, N));
```

```
AA = eye(N) - gamma*A;
```

Be aware that in MATLAB®, for efficiency reasons, the invert of the matrix can be written as inv(A)*y=A\ y.

1.4 External forces

```
% define convolution kernels
hgauss = fspecial('gaussian', 100, 30);
hprewitt  = fspecial('prewitt');

% gaussian filter
G = imfilter(I, hgauss, 'replicate');
% gradient (prewitt) and its norm
Fy= imfilter(G, hprewitt, 'symmetric');
Fx= imfilter(G, hprewitt', 'symmetric');
G = sqrt(Fx.^2+Fy.^2);

% orientation of previous gradient
Fy= imfilter(-G, hprewitt, 'symmetric');
Fx= imfilter(-G, hprewitt', 'symmetric');
```

1.5 Display results

To enhance the role of the external forces, the arrows showing the force are displayed (quiver function).

```
imshow(I,[])
hold on
plot([x;x(1)], [y; y(1)], 'g', 'linewidth', 3);

%% display arrows for external forces
step=20;
subx = 1:step:size(I,1);
suby = 1:step:size(I,2);
[Xa, Ya] = meshgrid(subx, suby);
quiver(Xa, Ya, Fx(subx, suby), Fy(subx,suby));
```

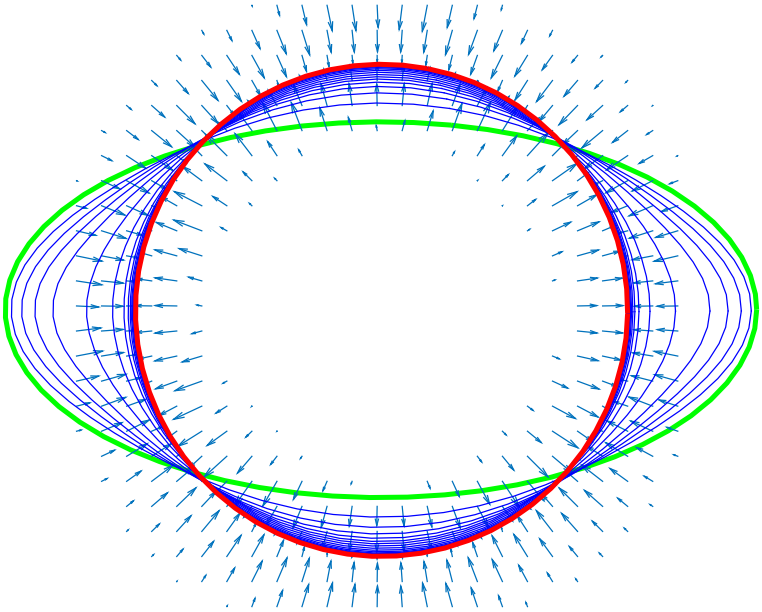

Figure 1: Result of the snake converging toward the disk, after 1000 iterations with the proposed parameters. The green ellipse is the original snake, the red snake shows the final result. Blue snakes are intermediate results.

1.6 Convergence algorithm

```
h = waitbar(0, 'snake converging...');
% iterations
for index = 1:iterations,
    % interpolate values of forces
    fex = interp2(Fx, x, y, 'linear');
    fey = interp2(Fy, x, y);

    x = AA\(x+gamma*fex);
    y = AA\(y+gamma*fey);
    % display
    if mod(index,10)==0
        plot([x;x(1)], [y;y(1)], 'b');
    end
    waitbar(index/iterations);
end
plot([x;x(1)], [y;y(1)], 'r', 'linewidth', 3);
close(h)
```

The results are displayed in Fig.**??**.