初始化零水平集：

```
a=3;b=4;
initial=6*ones(nrow,ncol);
initial(a:end-b,a:end-b)=0;
initial(a+1:end-b+1,a+1:end-b+1)=-6;
phi_0 = initial;
```

设置超参数：

```
numIter = 3;
delta_t = 5;
lambda = 3;
nu = 5;
mu = 0.04;
epsilon = 0.4;
```

计算初始梯度：

```
sigma = 0.5;
G = fspecial('gaussian',15,sigma);
II = conv2(I,G,'same');
[Ix,Iy]=gradient(II);
f = Ix.^2+Iy.^2;
g=1./( 1 + f );
```

进行水平集分割迭代，若零水平集包围区域不变化，则终止迭代。每 10 次迭代保存一次结果，同时终止迭代时也保存一次结果：

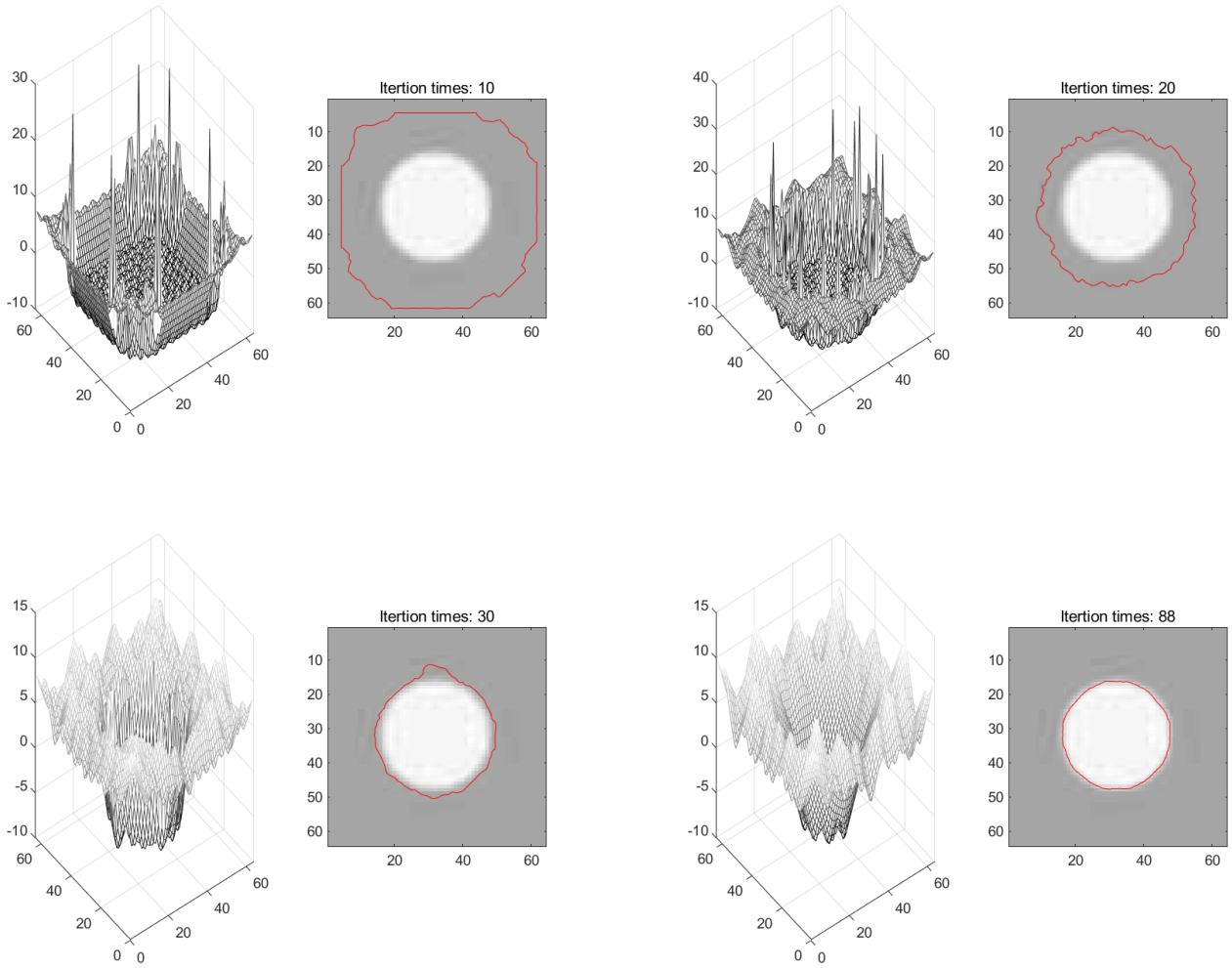
```
for k=1:300
    phi = evolution_cv(G, phi, mu, nu, lambda, delta_t,epsilon, numIter, g);   % update lev
    if mod(k,2)==0
        figure(2); clc; axis equal;
        title(sprintf('Itertion times: %d', k));
        subplot(1,2,1); mesh(phi);
        subplot(1,2,2); imagesc(uint8(I));colormap(gray)
        hold on; plotLevelSet(phi,0,'r');

        if k == 2
            seg_region_old = (phi < 0);
        else
            seg_region_new = (phi < 0);
            dif_pixNum = sum(sum(abs(seg_region_old - seg_region_new)));
            if dif_pixNum < 1 % 零水平集包围的区域不再变化，则终止迭代
                fprintf('Level set evolution is converged.\n');
                break;
            else
                seg_region_old = seg_region_new;
            end
          if mod(k,10)==0
              saveas(gcf, num2str(k), 'png');
          end
        end
    end
end
saveas(gcf, num2str(k), 'png');
```

演化方程的定义：

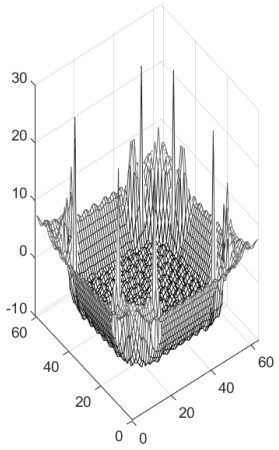
```
function phi = EVOLUTION_CV(I, phi0, mu, nu, lambda, delta_t, epsilon, numIter, g)

I = BoundMirrorExpand(I); % 镜像边缘延拓
phi = BoundMirrorExpand(phi0);
g = BoundMirrorExpand(g);
for k = 1 : numIter
    phi = BoundMirrorEnsure(phi);
    g = BoundMirrorEnsure(g);
    Curv = curvature(phi);

    fphi=(0.5/epsilon)*(1+cos(pi*phi/epsilon));
    DELTAPHI = fphi.*(phi<=epsilon)&(phi>=-epsilon);

    PHIxxyy = del2(phi);

    [gx,gy]=gradient(g);
    [phix,phiy]=gradient(phi);

    norm=sqrt(phix.^2 + phiy.^2 + 1e-10);
    phixn=phix./norm;phiyn=phiy./norm;



    % updating the phi function
    phi=phi+delta_t*(mu*(4*PHIxxyy-Curv)+lambda*DELTAPHI.*(gx.*phixn+gy.*phiyn+g.*Curv)+nu*g.*DELTAPHI);
end
phi = BoundMirrorShrink(phi); % 去掉延拓的边缘
```

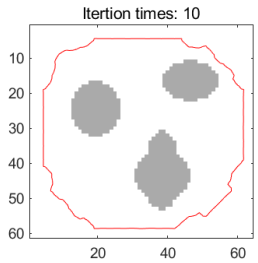
实验结果：

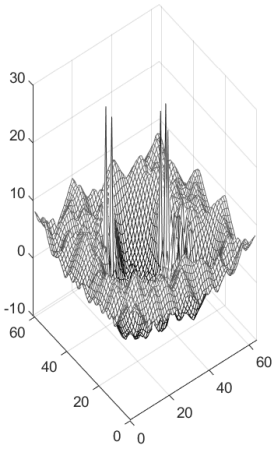
圆：

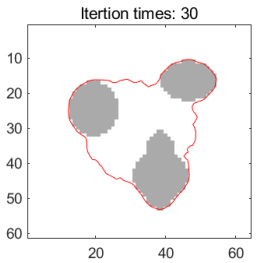


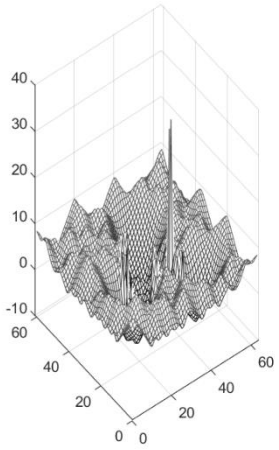
三个圆：

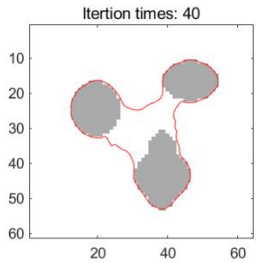


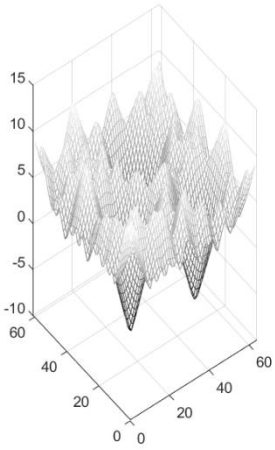

Itertion times: 10

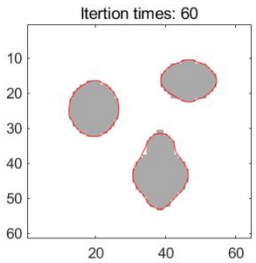



Itertion times: 30

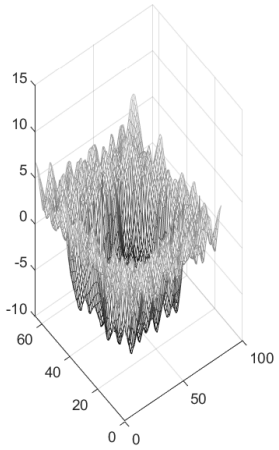


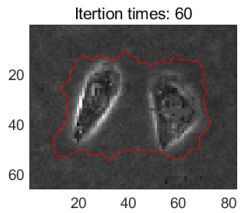

Itertion times: 40

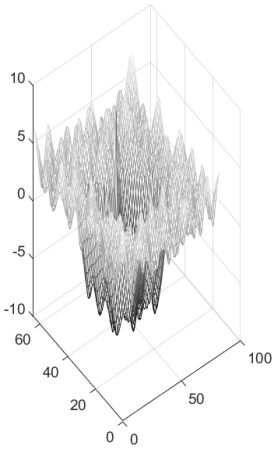


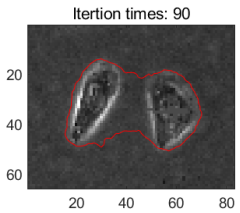

Itertion times: 60

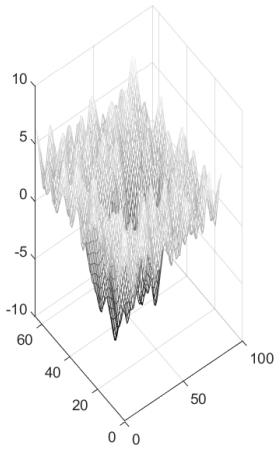
细胞：

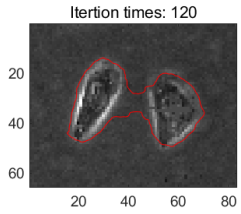


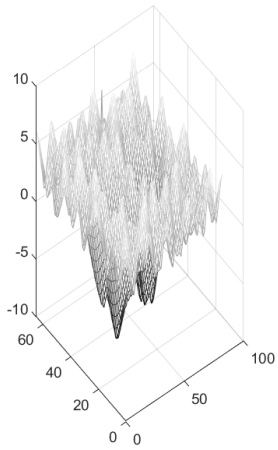

Itertion times: 60

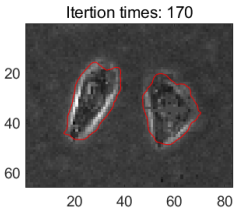



Itertion times: 90

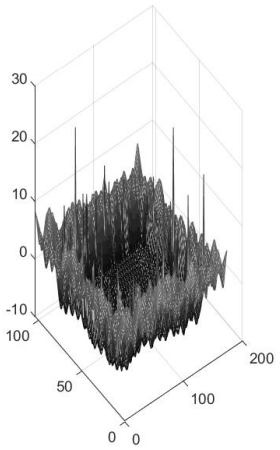


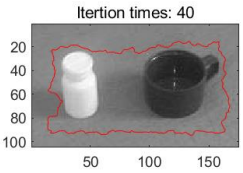

Itertion times: 120

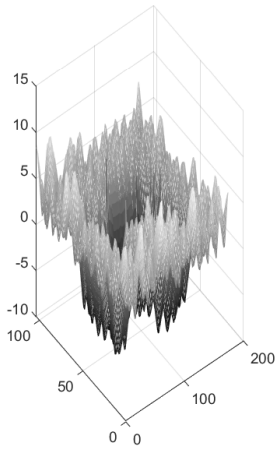


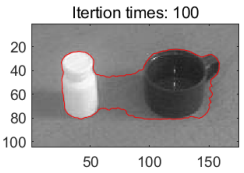

Itertion times: 170

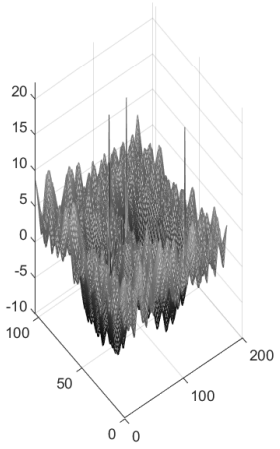
两个杯子：

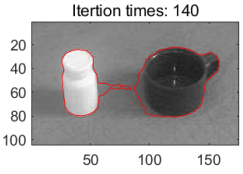


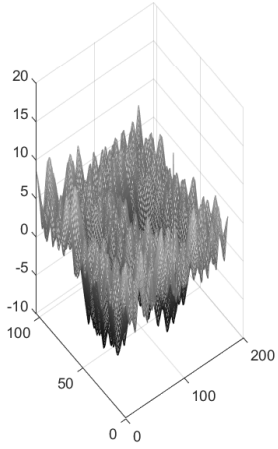

Itertion times: 40

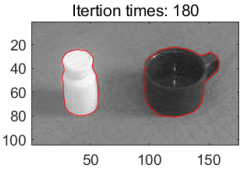



Itertion times: 100




Itertion times: 140




Itertion times: 180