Fall 2020 ENG 5300 Computer Assignment

name: Chris Collins

You must show **all** work to receive full credit. All work is to be your own.
Be neat and organized, and use correct notation. Due date: September 9 18:40

The purpose of this assignment is to get familiar with the *gradient search* for finding points of local extrema for functions of the form $z = f(x, y)$ with continuous partial derivatives.

The following paragraph is quoted from a classical text *Linear and Nonlinear Programming*, 3rd ed., 2008, by David G. Luenberger and Yinyu Ye, §8.6, pages 233 and 234:

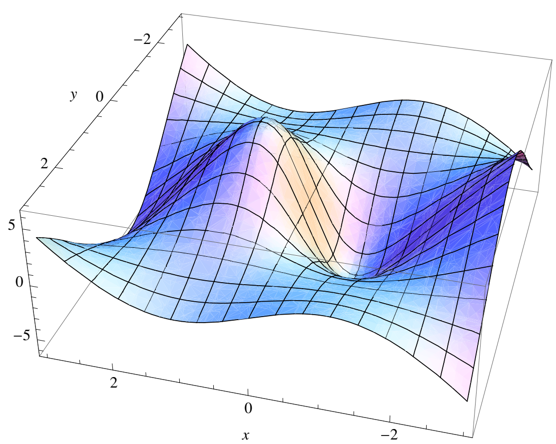
"One of the oldest and most widely known methods for minimizing a function of several variables is the method of steepest descent (often referred to as the gradient method). The method is extremely important from a theoretical viewpoint, since it is one of the simplest for which a satisfactory analysis exists. More advanced algorithms are often motivated by an attempt to modify the basic steepest descent technique in such a way that the new algorithm will have superior convergence properties. The method of steepest descent remains, therefore, not only the technique most often first tried on a new problem but also the standard of reference against which other techniques are measured."

As was discussed in class, the method of steepest ascent/descent is defined by the following iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k \pm \alpha \nabla f(x, y)$$

Where $\alpha > 0$ and $\mathbf{x}_k = \langle x_k, y_k \rangle$. The following example illustrates how to apply *gradient search* to find local minima or local maxima of a given function $z = f(x, y)$.

Consider the surface given by $z = f(x,y) = \frac{x(x^2-4)(y^2-4)}{1+x^2+y^2}$, $-4 \le x \le 4$, $-4 \le y \le 4$



- Perform 64 iterations of *gradient search* for local maxima starting from $(1, 3.2)$, with $\alpha = 0.04$.
- Perform 81 iterations of *gradient search* for local minima starting from $(-1, 3.2)$, with $\alpha = 0.03$.

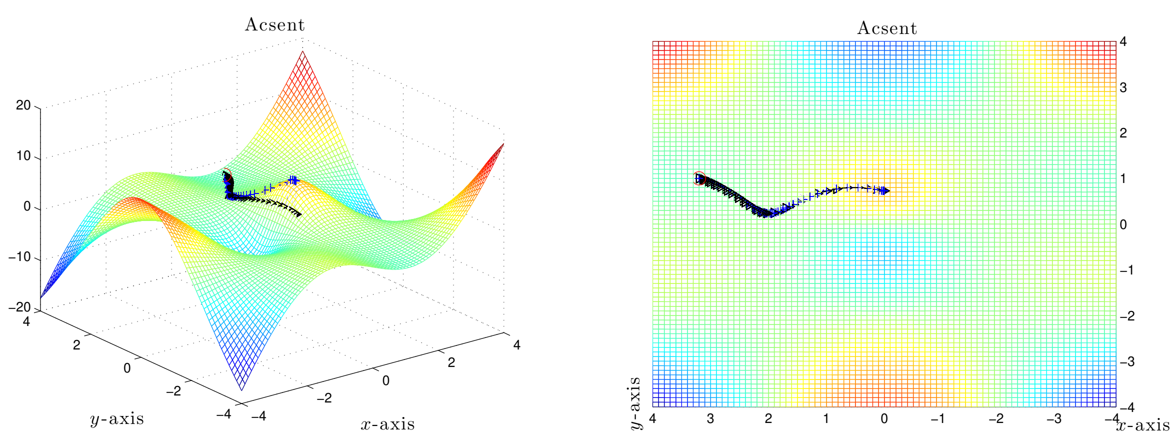
Find the gradient vector first (by hand or using MAPLE).

$$\nabla f(x,y) = \langle f_x, f_y \rangle = \left\langle \frac{(y^2-4)(7x^2+x^4+3x^2y^2-4-4y^2)}{(1+x^2+y^2)^2}, \frac{2x(x^2-4)y(5+x^2)}{(1+x^2+y^2)^2} \right\rangle$$

For *gradient ascent*, starting from $(1, 3.2)$, with $\alpha = 0.04$, run the following MATLAB code:

```
[x y]=meshgrid(-4:0.1:4,-4:0.1:4);              % boundaries and resolution of the grid
z=x.*(x.^2-4).*(y.^2-4)./(1+x.^2+y.^2);         % the z=f(x,y) given in the example
mesh(x,y,z); axis image;                         % Draws the surface
options = {'Interpreter','latex','FontSize',14};              % uses LaTeX for labeling
xlabel('$x$-axis',options{:}); ylabel('$y$-axis',options{:});
title('Ascent',options{:});
hold on
p=[1; 3.2];                                      % starting point given in the example
plot(p(1,1),p(2,1),'or','MarkerSize',10);        % red circle label for starting point
for i=1:64
    x=p(1,size(p,2));                            % so that you can copy/paste fx from MAPLE
    y=p(2,size(p,2));                            % so that you can copy/paste fy from MAPLE
    fx=(y^2-4)*(7*x^2+x^4+3*x^2*y ^2-4-4*y^2)/(1+x^2+y^2)^2;               % from MAPLE
    fy=2*x*(x^2-4)*y*(5+x^2)/(1+x^2+y^2)^2;                                 % from MAPLE
    p=[p p(:,size(p,2))+0.04*[ fx ; fy ]];              % the only line that really maters
    x=p(1,:);                   % so that you can copy the z component in stem3 from above
    y=p(2,:);                   % so that you can copy the z component in stem3 from above
    plot(p(1,:),p(2,:),'k>-','MarkerSize',7); % the xy-plane ∇f path towards local max
    stem3(p(1,:),p(2,:),x.*(x.^2-4).*(y.^2-4)./(1+x.^2+y.^2),'b-+');  % artsy cosmetics
    drawnow;
end
```

MATLAB animates the iterations and stops with the following picture on the left. Rotate the picture in 3D using mouse for from-the-top view, as shown on the right.



For your assignment, you have to modify the code highlighted in yellow according to your:

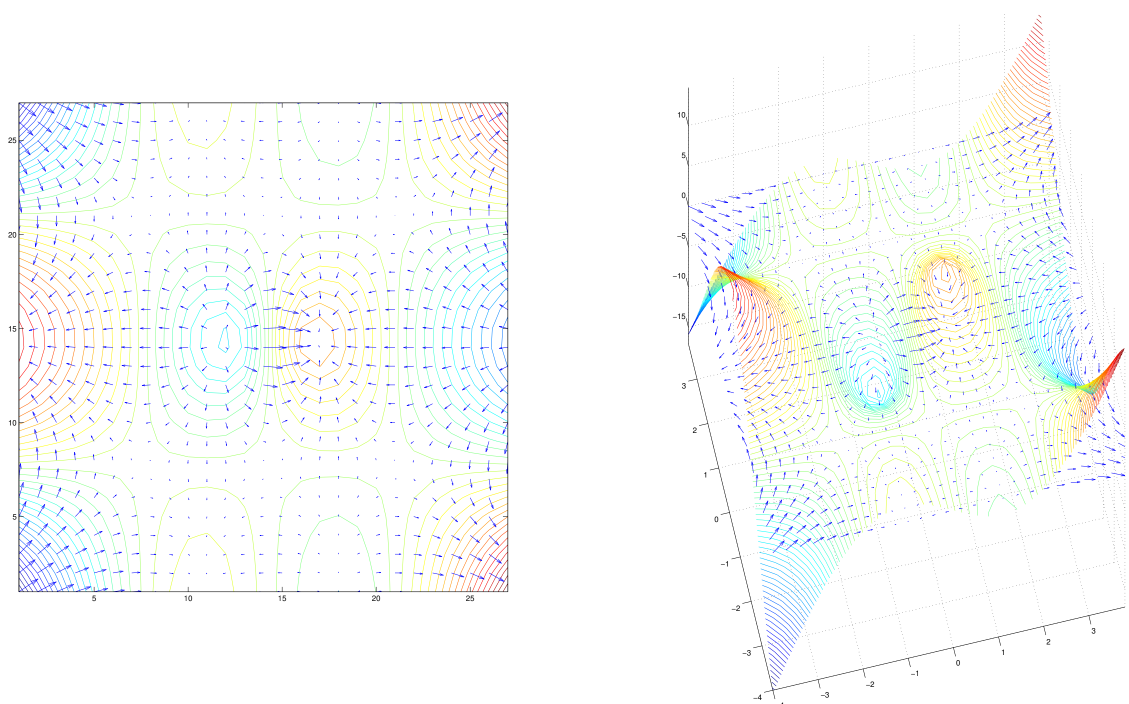
- $z = f(x, y)$.
- Boundaries of the grid.
- Step-scaling scalar α.
- Gradient ascent $\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \nabla f(x, y)}$. Gradient descent $\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(x, y)}$.
- Number of iterations.

The following is a terse (without fancy LaTeX labels) version of the *ascent* MATLAB code.

```
[x y]=meshgrid(-4:0.1:4,-4:0.1:4);              % this is a TERSE VERSION of ASCENT code
z=x.*(x.^2-4).*(y.^2-4)./(1+x.^2+y.^2);
mesh(x,y,z); hold on
x=1; y=3.2;                                      % Starting point
plot(x,y,'or','MarkerSize',10);                  % red circle label for starting point
for i=1:64
    fx=(y^2-4)*(7*x^2+x^4+3*x^2*y^2-4-4*y^2)/(1+x^2+y^2)^2;          % from MAPLE
    fy=2*x*(x^2-4)*y*(5+x^2)/(1+x^2+y^2)^2;                          % from MAPLE
    x=x+0.04*fx;     y=y+0.04*fy;                % x_{k+1} = x_k + α∇f(x,y)
    plot(x,y,'k<-'); drawnow;
end
```

The following MATLAB code generates a plot of gradient field with level curves in 2D on the left and in 3D on the right. The purpose of this visualization is to reinforce the fact that the gradient is perpendicular to level curves.

```
[x y]=meshgrid(-4:0.3:4,-4:0.3:4);          % use dimensions in part 1 on next page
z= x.*(x.^2-4).*(y.^2-4)./(1+x.^2+y.^2);    % use z=f(x,y) from part 1 on next page
[px,py] = gradient(z,2,2);
subplot(1,2,1); contour(z,37), hold on; axis image; quiver(px,py);
subplot(1,2,2); contour3(x,y,z,76); axis image; hold on; quiver(x,y,px,py);
```



Your assignment is to: (5 points for each MATLAB code +20 for explanations) = 45 points

1. Run *gradient ascent*, starting at $(-1.6, 0.3)$, with $\alpha = 0.5$ and 24 iterations, for the surface

$$z = f(x,y) = (x^5 + y^5)e^{-(x^2+y^2)}, \qquad -3 \le x \le 3, \; -3 \le y \le 3$$

2. Modify the following code to plot the level curves and the gradient field in part 1

```
[x y]=meshgrid(-4:0.2:4,-4:0.2:4);          % use dimensions in part 1
z=x.*(x.^2-4).*(y.^2-4)./(1+x.^2+y.^2);    % use z=f(x,y) from part 1
[px,py] = gradient(z,2,2);
subplot(1,2,1); contour(z,37), hold on; axis image; quiver(px,py);
subplot(1,2,2); contour3(x,y,z,37); axis image; hold on; quiver(x,y,px,py);
```

3. Run *gradient descent*, starting at $(1.4, -0.2)$, with $\alpha = 0.08$ and 64 iterations, for the surface

$$z = f(x,y) = (x^5 + y^5)e^{-(x^2+y^2)}, \qquad -3 \le x \le 3, \; -3 \le y \le 3$$

4. Run *gradient ascent*, starting at $(-1.2, 0.0)$, with $\alpha = 0.1$ and 419 iterations, for the surface

$$z = f(x,y) = (x^5 + y^5)e^{-(x^2+y^2)}, \qquad -3 \le x \le 3, \; -3 \le y \le 3$$

   Explain why the algorithm “stalls” on the plateau and does not climb the mountain.

5. Run *gradient ascent*, starting at $(1.3, -1.2)$, with $\alpha = 0.07$ and 128 iterations, for the surface

$$z = f(x,y) = \cos((x^2 + y^4)^{\frac{1}{7}})e^{-(x^2+y^2)}, \qquad -\frac{\pi}{2} \le x \le \frac{\pi}{2}, \; -\frac{\pi}{2} \le y \le \frac{\pi}{2}$$

   Explain why the algorithm misbehaves (goes crazy) near the optima.