0. Read sections 11.3, 11.4 A.6, and A.7 in the book.

1. Page 363, #2.9. Let $f(\mathbf{x}) = 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_1^4$. Determine the minimizers/maximizers of f and indicate what kind of minima or maxima (local, global, etc.) they are.

   Solution. We need to find where the gradient is zero. At those points, if the Hessian is positive definite, the point is a minimum, and if the Hessian is negative-definite, the point is a maximum. We compute

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 4x_1 - 2x_2 + 6x_1^2 + 4x_1^3 \\ 2x_2 - 2x_1 \end{bmatrix}$$

   and

$$H(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{bmatrix} 4 + 12x_1 + 12x_1^2 & -2 \\ -2 & 2 \end{bmatrix}$$

   Setting $\nabla f(\mathbf{x}) = \mathbf{0}$, we see that

$$\begin{aligned} 4x_1 - 2x_2 + 6x_1^2 + 4x_1^3 &= 0 \\ 2x_2 - 2x_1 &= 0 \end{aligned}$$

   so that, from the second equation, $x_1 = x_2$. Substituting this back into the first equation yields

$$\begin{aligned} 0 &= 4x_1 - 2x_1 + 6x_1^2 + 4x_1^3 = 2x_1 + 6x_1^2 + 4x_1^3 = 2x_1(1 + 3x_1 + 2x_1^2) \\ &= 2x_1(2x_1 + 1)(x_1 + 1) \end{aligned}$$

   Thus, $x_1 = 0, -\frac{1}{2}$, or -1. Since $x_2 = x_1$, the critical points are $(0, 0)$, $(-\frac{1}{2}, -\frac{1}{2})$, and $(-1, -1)$. At these points, the Hessian is:

$$\begin{aligned} H((0,0)) &= \begin{bmatrix} 4 & -2 \\ -2 & 2 \end{bmatrix} \\ H((-\tfrac{1}{2}, -\tfrac{1}{2})) &= \begin{bmatrix} 1 & -2 \\ -2 & 2 \end{bmatrix} \\ H((-1,-1)) &= \begin{bmatrix} 4 & -2 \\ -2 & 2 \end{bmatrix} \end{aligned}$$

   Coincidentally, $H((0, 0)) = H((-1, -1))$, so our work is reduced. To check it is positive definite, we can use Sylvester's criterion to notice that $h_{1,1}(0, 0) =$

$4 > 0$ and $\det(H((0,0))) = (4)(2) - (-2)(-2) = 4 > 0$, so the matrix is positive definite, and both $(0,0)$ and $(-1,-1)$ are local minima. Furthermore, the function tends ∞ as $\|\mathbf{x}\| \to \infty$, so at least one local minimum must be a global mimimum. Since $f((0,0)) = f((-1,-1)) = 0$, they are both local minima. Next, notice that $h_{1,1}(-\frac{1}{2},-\frac{1}{2}) = 1 > 0$, but $\det(H((-\frac{1}{2},-\frac{1}{2}))) = (1)(2) - (-2)(-2) = -2 < 0$, so $H((-\frac{1}{2},-\frac{1}{2}))$ is not positive definite. In fact, it is indefinite, and one can see in the plot below that f has a saddle-point at $(-\frac{1}{2},-\frac{1}{2})$.

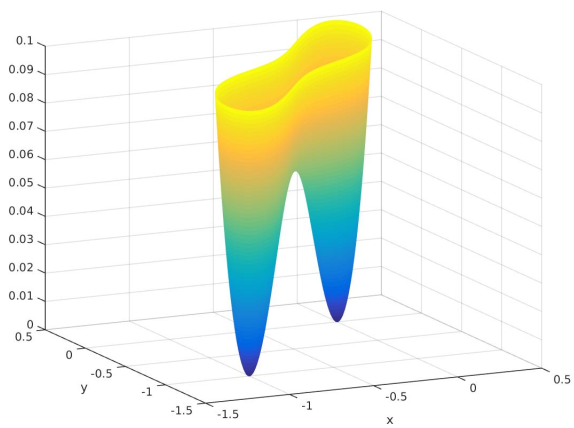


Figure 1: Plot of the function $f(\mathbf{x}) = 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_1^4$

By the way, here is how I plotted that in Matlab:

```
h = figure;
X = linspace(-1.5,0.5,200);
Y = linspace(-1.5,0.5,200);
[x y] = meshgrid(X,Y);
surf(x,y,2*x.^2 + y.^2-2*x.*y+2*x.^3+x.^4);
xlabel('x');
ylabel('y');
axis([-1.5 0.5 -1.5 0.5 0 0.1]);
shading interp;
lighting phong;
caxis([0,0.1]);
print(h,'-djpeg','HW4prob1.jpg'); % Save plot to a file.
```

2. Page 369, #3.2. Use the program you wrote in Assignment #3 for this one. Write a very brief description (no more than a short paragraph) of your methods, and report your results.

   Use Newton's method to solve:

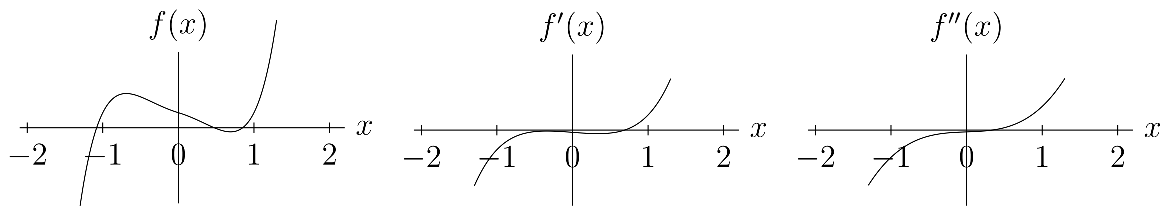
   $$\text{minimize} f(x) = 5x^5 + 2x^3 - 4x^2 - 3x + 2$$

   Look for the solution in the interval $[-2, 2]$. Make sure you have found a minimum and not a maximum.

   Solution. Newton's method for minimization (via solving $f'(x) = 0$) takes the form:

   $$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{25x^4 + 6x^2 - 8x - 3}{100x^3 + 12x - 8}$$

   It may help to see the plots below. Notice the very subtle oscillation near the origin.



   Here's a way to do it by brute force (which is totally overkill): check a few thousand initial guesses on the interval using a loop. (See previous homework's answer key for a coded Newton's method.)

```
df = @(x) 25*x^4+6*x^2-8*x-3;
ddf = @(x) 100*x^3+12*x-8, x0;

i=1;
x = zeros(1,1000);
for x0 = linspace(-2,2,1000);
    x(i) = Newton(df, ddf, x0, 10^(-15), 1000);
    i = i+1;
end;
display(unique(x)');
```

   I found $x = -0.289897948556636$ and $x = 0.689897948556636$ from this search. However,

```
>> ddf(-0.289897948556636)
ans =
         -13.9151015307185
>> ddf(0.689897948556636)
ans =
          33.1151015307186
```

so by the second derivative test, only $x = 0.689897948556636$ is a local minimum.

3. Page 370, #3.6 (You may want to read Appendix B.5 starting on page 696.) Consider the problem

$$\text{minimize } f(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{c}^T\mathbf{x}$$

Where Q is a (symmetric) positive definite matrix. Prove that Newton's method will converge in one step, regardless of the starting point.

Solution. As we saw in class,

$$\begin{aligned} \nabla f(\mathbf{x}) &= Q\mathbf{x} - \mathbf{c} \\ \nabla^2 f(\mathbf{x}) &= Q \end{aligned}$$

Thus, the critical points are solutions to $\mathbf{0} = \nabla f(\mathbf{x}) = Q\mathbf{x} - \mathbf{c}$, that is, solutions to $Q\mathbf{x} = \mathbf{c}$. Recall that very SPD matrix is intervible. Thus, the solution to $Q\mathbf{x} = \mathbf{c}$ is unique, and it is give by $\mathbf{x}_* = Q^{-1}\mathbf{c}$. Also, since $\nabla^2 f(\mathbf{x}) = Q$ is SPD, the function f is (strictly) convex, so the critical point $\mathbf{x}_* = Q^{-1}\mathbf{c}$ must be the unique global minimum, and are there are no other local minima.

Now,, Newton's method for minimization is

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k))^{-1}(\nabla f(\mathbf{x}_k)) \\ &= \mathbf{x}_k - Q^{-1}(Q\mathbf{x}_k - \mathbf{c}) \\ &= \mathbf{x}_k - \mathbf{x}_k + Q^{-1}\mathbf{c} \\ &= Q^{-1}\mathbf{c} = \mathbf{x}_* \end{aligned}$$

Thus, no mater what the starting position is, Newton's method converges in one step. However, this is not especially useful in this case, since to perform the first step, we need to solve $Q\mathbf{x} = \mathbf{c}$ anyway, which is the main difficulty to begin with. Applying Newton's method therefore doesn't reduce the complexity of this problem. It is more useful for problems with worse nonlinearity.

4. (a) Find an LDL^T factorization of the following symmetric positive-definite matrix

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 6 \\ 4 & 6 & 22 \end{bmatrix},$$

that is, write $A = LDL^T$ where L is lower-triangular, and D is diagonal.

Solution. First, we find an LU factorization using Gaussian elimination without pivoting.

$$\begin{aligned} A &= \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 6 \\ 4 & 6 & 22 \end{bmatrix} \\ &\sim \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -2 \\ 0 & -2 & 6 \end{bmatrix}; \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & * & 1 \end{bmatrix} \\ &\sim \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 2 \end{bmatrix} = U; \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & 1 \end{bmatrix} \end{aligned}$$

In class, we saw that $D = L^{-1}U^T$. Now, what does L^{-1} look like? Since L is lower-triangular, so is L^{-1} (to see why, write $LM = I$ where I is the identity, and see what the entries of M must be). Also, since L has only 1's on the diagonal, the same is true about L^{-1}, since

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I = LL^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ * & b & 0 \\ * & * & c \end{bmatrix}$$

so the only options are $a = b = c = 1$. Thus,

$$D = L^{-1}U^T = \begin{bmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Note: We did <u>not</u> compute L^{-1} here. We only used properties of it. We now have,

$$A = LDL^T = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & 1 \end{bmatrix}^T.$$

(b) Rewrite the LDL^T factorization to form an LL^T (i.e. Cholesky) factorization.

Solution. Using D from above, we find

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} := \sqrt{D}\sqrt{D}.$$

Therefore,

$$\begin{aligned} A &= LDL^T = L\sqrt{D}\sqrt{D}L^T \\ &= \left(\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \right) \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & \sqrt{2} \end{bmatrix} = \widetilde{L}\widetilde{L}^T. \end{aligned}$$

This is the Cholesky factorization, which is unique.

(c) Use either LDL^T or the LL^T the factorization of the above matrix to solve $A\mathbf{x} = \mathbf{b}$, where

$$\mathbf{b} = \begin{bmatrix} 1 \\ 6 \\ -6 \end{bmatrix}.$$

Hint: Use forward substitution and back substitution.

Solution. We only demonstrate the Cholesky solve here; LDL^T is similar. The idea is to solve is stages. First, note that solving $A\mathbf{x} = \mathbf{b}$ is the same as solving $LL^T\mathbf{x} = \mathbf{b}$, since $A = LL^T$. Thus, we solve first $L\mathbf{y} = \mathbf{b}$, and then $L^T\mathbf{x} = \mathbf{y}$, and we do it **not by inverting a matrix**, but by simple substitution, which is ***much*** faster.

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & -2 & \sqrt{2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ -6 \end{bmatrix}$$

We quickly see that $y_1 = 1$. Then $2y_1 + 1y_2 = 6$, so $y_2 = 4$. Then $4y_1 - 2y_2 + \sqrt{2}y_3 = -6$, so $y_3 = -\sqrt{2}$. Next, we solve

$$\begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & -2 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -6 \\ -\sqrt{2} \end{bmatrix}$$

This solve goes backwards. We first get $x_3 = \frac{1}{\sqrt{2}}(-\sqrt{2}) = -1$. Next, we work on x_2 to find $1x_2 - 2x_3 = -6$ so that $x_2 = 2$. Finally, $1x_2 + 2x_2 + 4x_3 = 1$, so $x_3 = 1$. Thus

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}.$$

5. **Bonus (1 point):** Write a program that computes the Cholesky factorization. It is OK to look online or at other resources to get ideas, but then close the webpage, book, etc., and write your code on your own. You will learn more and grow stronger if you do it this way.

   Solution. See the following Matlab code, saved as `cholesky.m`, for one possible solution.

```
function L = cholesky(A);
% Compute the Cholesky decomposition of an SPD matrix A.
n = size(A,1);
L = zeros(n,n); % Wasteful, since we only need to store the lower-t

tol = 1e-15;

for j = 1:n
    LjjSquared = A(j,j) - L(j,:)*L(j,:)';
    if LjjSquared < tol
        error('Matrix is either not SPD, or badly scaled.');
    end
    L(j,j) = sqrt(LjjSquared);

    for i = (j + 1):n
        L(i,j) = (A(i,j) - L(j,:)*L(i,:)')/L(j, j);
    end
end

end
```