The University of Texas at Austin
Dept. of Electrical and Computer Engineering

Midterm #2 **Take-Home Exam** ***Solution Set 1.0***

Prof. Brian L. Evans

Date: May 6, 2020 Course: EE 445S

Name: ____
Last, First

Please sign your name below to certify that you did not receive any help, directly or indirectly, on this test from another human other your instructor, Prof. Brian L. Evans, and to certify that you did not provide help, directly or indirectly, to another student taking this exam.

(please sign here) ________________________________

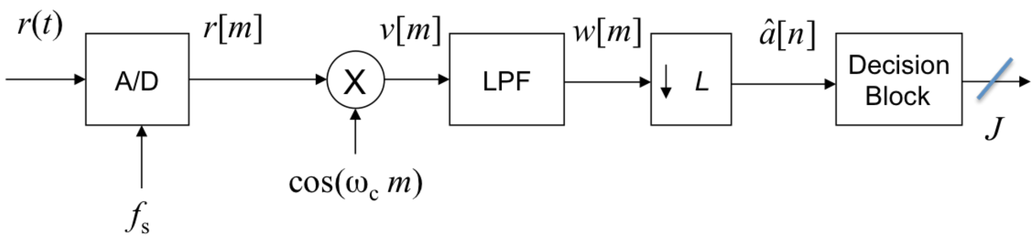
- **Take-home exam** is scheduled for Wednesday, May 6, 2020, from noon to 11:59pm.
  - The exam will be available on the course Canvas page at noon on May 6, 2020.
  - Please upload your solution to the course Canvas page by 11:59pm on May 6, 2020.
- **Perform all work on test.** All work should be performed on the exam. If more space is needed, then use the backs of the pages or scan in the extra page(s) with each problem.
- **Fully justify your answers**. When justifying your answers, reference your source and page number as well as quote the particular content in the source for your justification.
- **Internet access.** Yes, you may fully access the Internet when answering exam questions provided that you comply with the other instructions on this page.
- **Academic integrity.** You shall not receive help directly or indirectly on this test from another human except your instructor, Prof. Evans. You shall not provide help, directly or indirectly, to another student taking this exam.
- **Send questions to Prof. Evans.** You may send any questions or concerns about midterm #2 to Prof. Evans by e-mail at bevans@ece.utexas.edu.
- **Contact by Prof. Evans.** Prof. Evans might contact all students in the class during the exam through Canvas announcements. Please periodically monitor those announcements.

Problem	Point Value	Your score	Topic
1	24		Bandpass PAM Receiver Tradeoffs
2	30		QAM Communication Performance
3	28		Nonlinear Channel Equalization
4	18		Potpourri
Total	100		

Problem 2.1. *Bandpass Pulse Amplitude Modulation Receiver Tradeoffs. 24 points.*

A bandpass pulse amplitude modulation (PAM) receiver is described as

Lectures 13 & 14
JSK Ch. 8 & 11
Lab #5 & WWM Ch. 17
HW 5.2, 5.3, 6.1, 6.2
Midterm 2.1 F19
Handout P



where m is the sampling index and n is the symbol index, and has **system parameters**

$a[n]$ transmitted symbol amplitude	$\hat{a}[n]$ received symbol amplitude	
$2d$ constellation spacing	f_s sampling rate	f_{sym} symbol rate
$g[m]$ raised cosine pulse with rolloff α	J bits/symbol	L samples/symbol
M number of levels, i.e. $M = 2^J$	N_g symbol periods in $g[m]$	ω_c carrier freq. in rad/sample

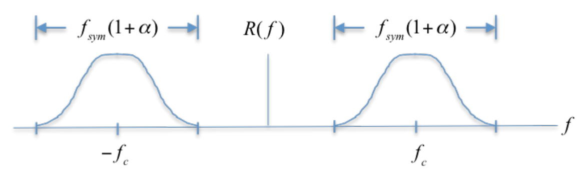
The only impairment is additive thermal noise $w(t)$ modeled as zero-mean Gaussian with variance σ^2.

Hence, $r(t) = s(t) + w(t)$ where $s(t)$ is the transmitted bandpass PAM signal.

Give a formula for each quantity below in terms of the symbol rate f_{sym} and describe how much the quantity changes when the symbol rate increases.

(a) Bit rate in bits/s. *4 points.* **Bit rate $J\, f_{sym}$ increases linearly when f_{sym} increases.**

(b) Transmission bandwidth in Hz. *4 points.*

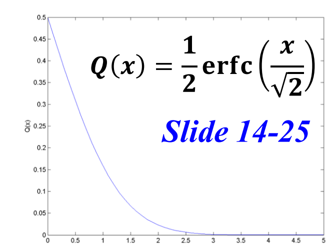


Transmission bandwidth $f_{sym}\,(1 + \alpha)$ increases linearly when f_{sym} increases.

(c) Sampling rate f_s. *4 points.*

Sampling rate $L\, f_{sym}$ increases linearly when f_{sym} increases. (Alternately, $f_s > 2\, f_{max}$ where $f_{max} = f_c + ½\, f_{sym}$ and $f_c > f_{sym}$. The product $2\, f_{max}$ increases linearly when f_{sym} increases.)

(d) Probability of symbol error. *4 points.* **Recalling that $T_{sym} = 1 / f_{sym}$,**



$P_{error}^{PAM} = 2\,\frac{M-1}{M}\, Q\left(\frac{d}{\sigma}\sqrt{T_{sym}}\right) = 2\,\frac{M-1}{M}\, Q\left(\frac{d}{\sigma\sqrt{f_{sym}}}\right)$ **increases when f_{sym} increases because $Q(x)$ is monotonically decreasing vs. x → Increase in P_{error}^{PAM} is proportional to $\sqrt{T_{sym}}$ in low SNR (small positive x) and faster than exponential in T_{sym} for high SNR (large positive x). Slide 14-26: $Q(\sqrt{\rho}) \leq 1/\sqrt{2\pi}\ e\text{^}(-\rho/2)/\sqrt{\rho}$ for large positive ρ.**

(e) Implementation complexity in multiplications per second. *8 points.* **(r) means r multiplications/s.**

Analog-to-digital (A/D) converter would have at least an analog filter, sampler, quantizer (0).

Generate $\cos(\omega_c m)$ via lookup table (0), diff. equ. ($2\, L\, f_{sym}$) or math library call ($30\, L\, f_{sym}$).

Multiply $r[m]$ and $\cos(\omega_c m)$ to produce $v[m]$ ($L\, f_{sym}$)

LPF plus downsampler takes ($L^2\, N_g\, f_{sym}$) in direct form or ($L\, N_g\, f_{sym}$) in polyphase form

Decision block algorithm: Euclidean distance ($2\, M\, f_{sym}$) or using comparisons (0)

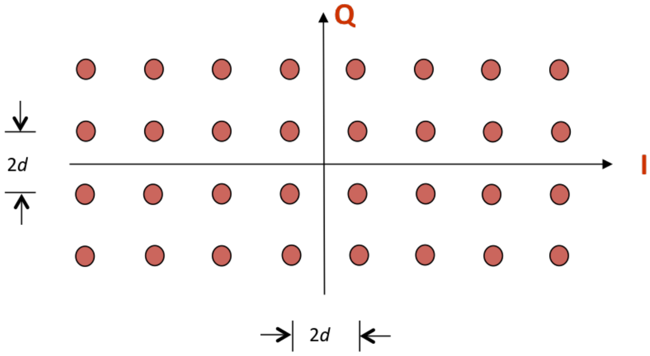
Implementation complexity increases linearly when f_{sym} increases

Lectures 15 & 16; JSK Ch. 16; Lab #6; HW 6.3 & 7.3; Handout P; Midterm 2.2 on 32-QAM in Sp13; Other Midterm 2.2 problems in F14, Sp15, F15, Sp16, F16, Sp17, F17, Sp18, F18, Sp19, F19

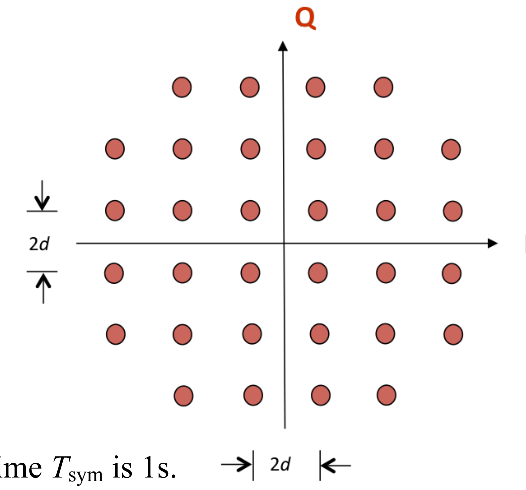
Problem 2.2 *QAM Communication Performance. 30 points. Problem continues onto the next page.*

Part I. Consider choosing a 32-QAM constellation.

Constellation #1: 8x4 rectangular



Constellation #2: Power Efficient



Constellation spacing is $2d$. Pulse shape energy is 1. Symbol time T_{sym} is 1s.

Compute the peak and average power for constellation #2. *3 points each.*

	Constellation #1	**Constellation #2**	
(a) Peak transmit power	$58d^2$	$\mathbf{34d^2}$	*3 points*
(b) Average transmit power	$26d^2$	$\mathbf{20d^2}$	*3 points*

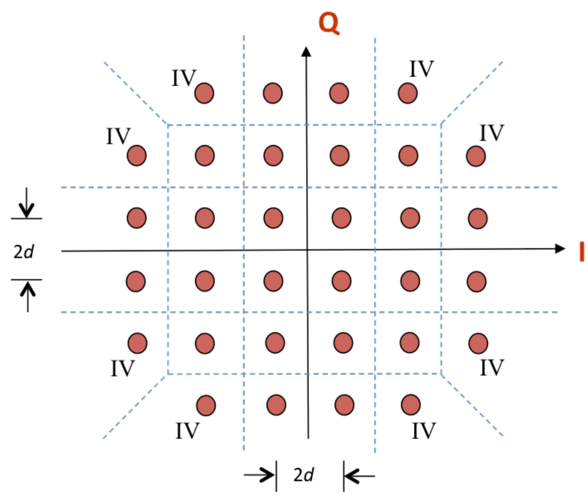
Constellation #2 has the lowest peak and average power possible, and is commonly used in practice.

Part II. For constellation #2, we're going to introduce a type IV constellation region. We'll also use it in Part III.

On the right, decision region boundaries are shown by the in-phase (I) axis, quadrature (Q) axis and dashed lines.

Type IV region has a diagonal line separating two nearest neighbors on a corner. It's a union of a type II region (finite in one dimension and infinite in the other) and half of a type III region (quarter plane).

We now have eight type IV regions instead of four type III regions (quarter planes at corner points).



	Constellation #1	**Constellation #2**	
(c) Number of type I regions	12	**16**	*3 points*
(d) Number of type II regions	16	**8**	*3 points*
(e) Number of type III regions	4	**0**	
(f) Number of type IV regions	0	**8**	
(g) Symbol error probability for additive Gaussian noise, zero mean & variance σ^2	$\frac{13}{4} Q\left(\frac{d}{\sigma}\right) - \frac{21}{8} Q^2\left(\frac{d}{\sigma}\right)$	**see next page**	*6 points*
(h) Express d/σ as a function of Signal-to-Noise Ratio (SNR) in linear units	$\text{SNR} = 26\left(\frac{d^2}{\sigma^2}\right)$ $\frac{d}{\sigma} = \sqrt{\frac{\text{SNR}}{26}} \approx 0.196\sqrt{\text{SNR}}$	$\mathbf{SNR = 20\left(\frac{d^2}{\sigma^2}\right)}$ $\boldsymbol{\frac{d}{\sigma} = \sqrt{\frac{SNR}{20}} \approx 0.224\sqrt{SNR}}$	*3 points*

(a) We find the constellation points with the largest radii. There are 8 such points. The two in the upper right quadrant are $3d + j5d$ and $5d + j3d$. Their radius is $\sqrt{34d^2}$. Peak power is $34d^2$.

(b) We can use the fact that the constellation has quadrant symmetry to compute the average power in the upper right quadrant. The instantaneous power calculations for the constellation points are $2d^2$, $10d^2$, $10d^2$, $18d^2$, $26d^2$, $26d^2$, $34d^2$, $34d^2$, which has an average of $20d^2$.

(g) As a shorthand notation, we'll use $q = Q\left(\frac{d}{\sigma}\right)$. The probabilities of correct detection for the type I, II, and III constellation regions, respectively, follow from Lecture Slides 15-13 and 15-14:

Type I. $P_{correct}^{type\ I} = (1 - 2q)^2$

Type II. $P_{correct}^{type\ II} = (1 - 2q)(1 - q)$

Type III. $P_{correct}^{type\ III} = (1 - q)^2$

For the type IV constellation region, we'll use the constellation point $5d + j3d$:

$$P_{correct}^{type\ IV} = P\left(\left(v_I(nT_{sym}) > -d\right) \,\&\, \left(-d < v_Q(nT_{sym}) < v_I(nT_{sym}) + 2d\right)\right)$$

$$P_{correct}^{type\ IV} = P\left(\left(v_I(nT_{sym}) > -d\right) \,\&\left(v_Q(nT_{sym}) > -d\right) \,\&\, \left(v_Q(nT_{sym}) - v_I(nT_{sym}) < 2d\right)\right)$$

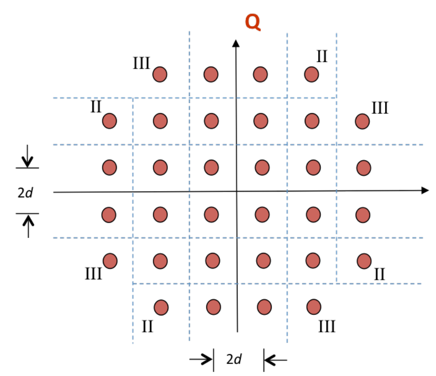
Let $z = v_Q(nT_{sym}) - v_I(nT_{sym})$ be a random variable. Assuming that $v_Q(nT_{sym})$ and $v_I(nT_{sym})$ are statistically independent, z has zero mean and its variance is equal to the variance of $v_Q(nT_{sym})$ plus the variance of $v_I(nT_{sym})$, per Spring 2016 Midterm #2 Problem 2.3.

$$P_{correct}^{type\ IV} = P\left(\left(v_I(nT_{sym}) > -d\right) \,\&\left(v_Q(nT_{sym}) > -d\right) \,\&\, (z < 2d)\right)$$

$$P_{error}^{total} = 1 - P_{correct}^{total}$$

Given the difficulty in finding a closed-form solution for the symbol error probability using type I, II and IV constellation regions, we'll find the lower and upper bounds instead.

<u>Lower bound</u>: Using only type I, II, III regions, we have $\frac{16}{32}$, $\frac{12}{32}$ and $\frac{4}{32}$ probabilities for each region:



$$P_{correct}^{total} = \frac{1}{2}(1 - 2q)^2 + \frac{3}{8}(1 - 2q)(1 - q) + \frac{1}{8}(1 - q)^2$$

$$P_{correct}^{total} = \frac{1}{2}(1 - 4q + 4q^2) + \frac{3}{8}(1 - 3q + 2q^2) + \frac{1}{8}(1 - 2q + q^2)$$

$$P_{correct}^{total} = 1 - \frac{27}{8}q + \frac{23}{8}q^2$$

$$P_{error}^{total} = 1 - P_{correct}^{total} = \frac{27}{8}q - \frac{23}{8}q^2 = \frac{27}{8}Q\left(\frac{d}{\sigma}\right) - \frac{23}{8}Q^2\left(\frac{d}{\sigma}\right)$$

<u>Upper bound</u>: Using only type I and II regions, which would leave quarter plane gaps in each quadrant at $5d + j5d$, $-5d + j5d$, etc.,

$$P_{correct}^{total} = \frac{1}{2}(1 - 2q)^2 + \frac{1}{2}(1 - 2q)(1 - q) = 1 - \frac{7}{2}q + 3q^2$$

$$P_{error}^{total} = 1 - P_{correct}^{total} = \frac{7}{2}q - 3q^2 = \frac{7}{2}Q\left(\frac{d}{\sigma}\right) - 3Q^2\left(\frac{d}{\sigma}\right) = \frac{28}{8}Q\left(\frac{d}{\sigma}\right) - \frac{24}{8}Q^2\left(\frac{d}{\sigma}\right)$$

Summary: There is very little difference between the lower and upper bounds. Almost all of the symbol error probability is in the type I and type II regions.

Note: The symbol error probability expression has the form $c_0\, Q\left(\frac{d}{\sigma}\right) - c_1\, Q^2\left(\frac{d}{\sigma}\right)$. As $\sigma \to \infty$, knowing that d is positive, the symbol error probability goes to $c_0\, Q(0) - c_1\, Q^2(0)$ where $Q(0) = 0.5$. Generally, $c_0 > c_1$. As $\sigma \to 0$, the symbol error probability goes to 0. Randomly guessing a M-PAM symbol amplitude gives an symbol error probability of $\frac{M-1}{M}$.

Part III. In the receiver, finding the nearest constellation point to the received QAM symbol amplitude using Euclidean distance provides high-accuracy in the symbol detection. Complexity is proportional to the number of levels $M = 2^J = 32$ where $J=5$ is the number of bits in a symbol: 64 multiplications, 96 additions, and 64 memory reads in words for each (I, Q) symbol amplitude.

We introduced the type IV region in part II to unlock a low-complexity divide-and-conquer method that is just as accurate as using Euclidean distance but only needs to use comparison operations. Describe the method and compare its complexity with the Euclidean distance method. *9 points.*

Midterm 2.2(i) Sp16

Midterm 2.1(f) F18

Let (i, q) be the received symbol amplitude.

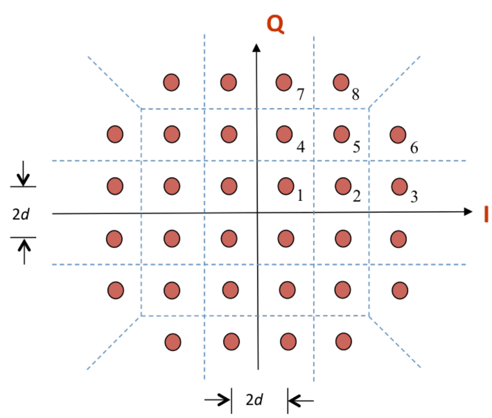
1. **If $i > 0$, then in right-half plane; otherwise, in left-half plane**
2. **If $q > 0$, then in upper half plane; otherwise, in bottom half plane**

Remaining steps are for the upper right quarter plane (i.e. for $i > 0$ and $q > 0$):

3. **If $i > 4d$,**
    - **If $q < 2d$, then point 3**
    - **Else If $q > i$, then point 8**
    - **Else point 6**

   Else If $i < 2d$,
    - **If $q > 4d$, then point 7**
    - **Else If q < $2d$, then point 1**
    - **Else point 4**

   Else If $q > 4d$, then point 8
    - **Else If $q < 2d$, then point 2**
    - **Else point 5**



The above divide-and-conquer algorithm uses 4, 5 or 6 comparisons (5.375 on average) and takes J+1 comparisons and memory reads.

Euclidean distance method computes the Euclidean distance squared from the received symbol amplitude (i, q) to each symbol amplitude in the constellation map (I, Q): $(i - I)^2 + (q - Q)^2$. The square root is an unnecessary calculation—it doesn't change the ordering among the distances.

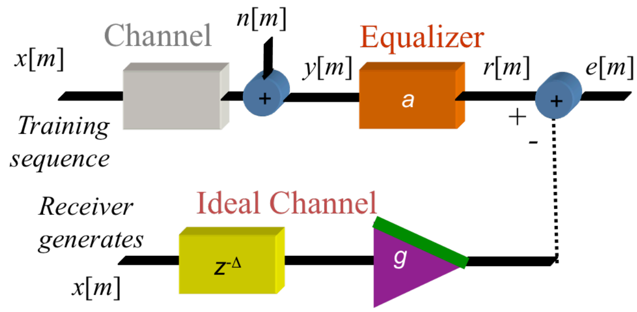
Its complexity is 2×2^J multiplications, 3×2^J additions, 2×2^J memory reads and 2^J comparisons.

In practice, QAM constellations are as large as $J = 8$ in cellular and Wi-Fi, and $J = 15$ for ADSL. The difference between linear and exponential complexity can be significant.

JSK Sec. 2.12 & 13.1-13.3; HW 5.1, 6.1, 6.2, 7.2; Slides 16-6 to 16-8; In-Lecture Problems 2&3; Midterm #2 Problems: 2.2 F12; 2.1 Sp13; 2.1 F13; 2.1 Sp14; 2.3 Sp17; 2.4 Sp17; 2.3 F17; 2.3 F19

Problem 2.3. *Nonlinear Channel Equalization. 28 points.*

In the discrete-time system on the right, the equalizer operates at the sampling rate.



The channel has significant **nonlinear distortion**.

We're going to use a **nonlinear equalizer** of the form

$r[m] = a_0 + a_1\, y[m] + a_2\, y^2[m] + \ldots + a_N\, y^N[m]$

where $a_0, a_1, a_2, \ldots a_N$ are real-valued coefficients.

The channel model includes additive noise $n[m]$ that has a Gaussian distribution with zero mean and variance σ^2.

(a) Give a training sequence for $x[m]$ that you would use? Why? *3 points.*
Pseudo-noise sequences and chirp sequences have all discrete-time frequencies present in them. Either can be independently generated by the receiver. A pseudo-noise sequence can be generated using only logical operations and memory.

(b) For one of the training sequences in part (a), describe how you would estimate the transmission delay parameter Δ in the ideal channel model. *3 points.*
The receiver can correlate the received signal $y[m]$ against the anticipated training sequence $x[m]$, and we can take the location of the first peak to be Δ samples.

(c) What objective function would you use? Why? *6 points.*
We would like to minimize the error between $r[m]$ and $s[m]$: $J(e[m]) = \frac{1}{2} e^2[m]$

By driving $e^2[m]$ to zero, we also drive $e[m]$ to zero.

Another advantage is that the amount of the offset in the parameter update at each iteration will be proportional to the error— the update will rapidly converge for large errors and slowly converge for small errors, provided that the step size has been chosen correctly.

Another advantage is relatively low computational complexity.

(d) For an adaptive nonlinear equalizer, **derive the update equation** for the vector of coefficients $\vec{a}$ for the objective function in part (c). Here, $\vec{a} = [\, a_0 \;\; a_1 \;\; a_2 \;\; \cdots \;\; a_N \,]$. *12 points.*

$$e[m] = r[m] - s[m]$$

$$\vec{a}[m+1] = \vec{a}[m] - \mu \left. \frac{dJ(e[m])}{d\vec{a}} \right|_{\vec{a}=\vec{a}[m]} = \vec{a}[m] - \mu\, e[m]\, \vec{y}[m]$$

where $\vec{y}[m] = [\, 1 \quad y[m] \quad y^2[m] \quad \cdots \quad y^N[m] \,]$.

(e) For your answer in (c), what values of the step size (learning rate) μ would you use? *4 points.*

Use a small positive value for the step size μ, such as 0.01 or 0.001, for convergence of the steepest descent algorithm. A step size of zero will prevent any updates. A negative step size and a large positive step size will cause divergence.

Problem 2.4. *Potpourri. 18 points*

(a) A handheld garage door opener transmits a binary request to an automatic garage door control unit. The binary request is to open the garage door if closed, and close the garage door if open.

Please describe the signals based on PN sequences that the garage door opener would transmit to indicate that it is sending its binary request to the automatic garage door control unit (receiver)?

The transmission/reception would have to be

- Reliable -- a very high probability of correct detection
- Secure -- nearly impossible for an opener meant for another unit to work on your garage door.

Lab #4; HW 4.1, 4.2, 4.3, 5.2; Wikipedia "garage door opener"

***Initialization*. At manufacturing, the opener and control unit are initialized with the same random 32-bit number that will be the value of the pseudo-noise (PN) shift register generator in both. This will make it highly unlikely for openers of the same brand to open your garage. *Transmission/Reception:* The PN generator would be used to create a self-synchronizing data scrambler in the opener and descrambler in the control unit. The opener would scramble a text message (converted to bits) that contains a 16-bit number of how many times the opener has been pressed in its lifetime. The number must be equal to or higher than that tracked by the control unit. This prevents an eavesdropper from playing back a recorded transmission to open your garage. An opener will repeat the transmission 10 times to improve reliability.**

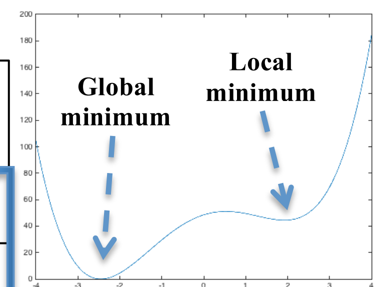
(b) Steepest descent algorithm to minimize an objective (cost) function.

Steepest descent algorithm for parameter x is

$$x[m+1] = x[m] - \mu \left. \frac{dJ(x)}{dx} \right|_{x=x[m]}$$

to minimize the objective function $J(x)$ given a positive value for step size (learning rate) μ.

```
x = -4 : 0.001 : 4;
% p = [1 0 -10 10 48.5];
p = poly( [-3, -1, 1, 3 ] );
p(4) = 10;
p(5) = p(5) + 39.5;
Jx = polyval(p, x);
plot(x, Jx);
```



i. Draw an objective function that has at least one global minimum value and at least one local minimum that is not a global minimum. *3 points.*

ii. Give a way to determine if a steepest descent algorithm converged to an answer. *3 points.*
***Method #1:* Since $J'(x) = 0$ at a minimum, $|J'(x)| \leq 10^{-7}$ for 10 iterations in a row.**
***Method #2:* For 10 iterations in a row, the relative change in $x[m]$ is below a threshold $|x[m+1] - x[m]| \leq 10^{-4}\, |x[m]| + \varepsilon$ where small $\varepsilon > 0$ is added in case $x[m] \approx 0$**

iii. How would you use multiple steepest descent algorithms running in parallel to reach a solution with a lower objective (cost) function? *3 points.*
Start each steepest descent algorithm with a different randomized initial guess and then choose the answer with smallest objective function value.

iv. If you could only run one steepest descent algorithm, how could you modify it to get out of a possible local minimum? *3 points.*
Use the test in part (b)ii to determine if the algorithm might be a local minimum.

***Answer #1:* Negate step size μ to head toward a maximum value and then switch back. Downside is that it will take a long time to get out of a local minimum because near a local minimum, $J'(x)$ is close to zero, and at a local minimum, $J'(x)$ is zero.**

***Answer #2:* Temporarily switch to a large step size μ and then switch back.**

***Answer #3:* Set the parameter to a random number and continue updating. Once the iteration converges, compare the objective function value with the previous solution.**

In-Lecture Assignment 2 & 3; HW 5.1, 6.1, 6.2, 7.2, 7.3; Midterm #2 problems 2.2 F12; 2.1 Sp13; 2.1 Sp14; 2.3 F17