Question 1

Linear equation solvers can sometimes be used to solve constraint systems that are not exactly linear like:

$$\begin{aligned} x + y &= 2 \text{ or } 6 \\ x + 2y &= 1 \text{ or } 7 \\ x + 3y &= 0 \text{ or } 2 \\ 2x + y &= 1 \text{ or } 5 \\ 3x + y &= 7 \text{ or } 8. \end{aligned} \tag{1}$$

(a) Express (1) as a *linear* system of equations in the five "variables" x^2, y^2, xy, x, and y. (**Hint:** Write each constraint "$e = a$ or b" as $(e - a)(e - b) = 0$ and expand.)

Solution: In matrix form the system is

$$\begin{bmatrix} 1 & 1 & 2 & -8 & -8 \\ 1 & 4 & 4 & -8 & -16 \\ 1 & 9 & 6 & -2 & -6 \\ 4 & 1 & 4 & -12 & -6 \\ 9 & 1 & 6 & -45 & -15 \end{bmatrix} \cdot \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \end{bmatrix} = \begin{bmatrix} -12 \\ -7 \\ 0 \\ -5 \\ -56 \end{bmatrix}.$$

(b) Treating these variables as independent, solve the system. You may want to use a computer for this. What are x and y? Verify that they satisfy all constraints in (1).

Solution: It solves to $x^2 = 9, y^2 = 1, xy = -3, x = 3, y = -1$. Indeed $x = 3, y = -1$ satisfy all the constraints with right-hand side values $2, 1, 0, 5, 8$ in order.

(c) Here is an alternative method for solving (1). Take any two constraints in (1). For all four possible pairs of values of the right-hand side, find x and y. Among these four, keep the one that is consistent with the other three constraints.

Solution: I take the first two constraints and obtain the following solutions:

right-hand values	$2, 1$	$2, 7$	$6, 1$	$6, 7$
x, y	$3, -1$	$-3, 5$	$1, 5$	$5, 1$

The first solution-pair is the only one that is consistent with all equations. The other three all violate the third equation.

(d) Suppose you have a system with 50 unknowns x_1 to x_{50}, and 5000 constraints of the type $a_1x_1 + \cdots + a_{50}x_{50} = b$ or c. Which of the methods do you think is preferable for solving such a system? Justify your answer.

Solution: The method in part (b) transforms it into a system with $\binom{50}{2}$ "variables" x_ix_j with $i \neq j$, plus 50 "variables" x_i^2 and that many x_i. The total number of variables is $n = \binom{50}{2} + 2 \cdot 50 = 1325$. Assuming that, among the 5000 constraints, there are at least 1325 that are linearly independent, the time it takes to solve for all the variables via Gaussian elimination is $O(n^3)$, which is on the order of 2^{31}. In contrast, the method from part (c) would need to try all possible assignments to some 50 of the equations and its time complexity is on the order of 2^{50}. The method from part (b) appears more efficient.

The drawback of method (b) is that it requires 1325 linearly independent constraints. If fewer are available it is unclear how to make it work. In contrast, method (c) works as long as the left-hand side of some 50 equations in the original system are linearly independent. Then every choice for the right-hand side values will uniquely determine a solution. One of these is guaranteed to work assuming the system had a solution in the first place.

Question 2

In this question you will investigate Gradient Descent on underdetermined linear systems.

(a) Write down the sum of squares loss for the equation $x + y = 1$ and calculate its gradient.

Solution: The sum-of-squares loss is $f(x, y) = (x + y - 1)^2$. Its gradient is $(\partial f/\partial x, \partial x/\partial y) = (2(x + y - 1), 2(x + y - 1))$.

(b) Suppose you run gradient descent with rate ρ on part (a). How do the values of $x + y - 1$ and $x - y$ change in each iteration? What is the maximum rate that guarantees convergence?

Solution: (x, y) changes by $-\rho(2(x + y - 1), 2(x + y - 1))$. Therefore $x + y$ changes by $-4\rho(x + y - 1)$, and so does $x + y - 1$ (as -1 doesn't change). Namely

$$(x + y - 1)' = (x + y - 1) - 4\rho(x + y - 1) = (1 - 4\rho)(x + y - 1). \tag{2}$$

So $x + y - 1$ is scaled by $1 - 4\rho$ in each iteration. $x - y$ doesn't change because the x-change cancels out the y-change.

For convergence to happen $1 - 4\rho$ needs to be bounded by one in absolute value. Otherwise $x + y - 1$ will blow out of control. Therefore ρ can be anywhere between 0 and 1/2.

(c) To which target (x^*, y^*) does gradient descent in part (b) converge to under initialization $x = 0, y = 0$? How about $x = 1, y = -1$?

Solution: Assuming convergence, $x+y-1$ eventually vanishes but $x-y$ stays invariant. If the initialization is $x = y = 0$ the target $(x^\star, y^\star)$ must satisfy both $x^\star + y^\star - 1 = 0$ and $x^\star - y^\star = 0 - 0 = 0$. It is $(1/2, 1/2)$. If $x = 1$ and $y = -1$ the target equations are $x^\star + y^\star - 1 = 0$ and $x^\star - y^\star = 1 - (-1) = 2$, so the target is $(3/2, -1/2)$.

(d) For each of the initializations in part (b), calculate the distance between the state (x_t, y_t) of Gradient Descent at step t and (x^*, y^*) as a function of ρ. At which step does the distance dip below 0.01 when $\rho = 0.1$?

Solution: When initialized with $\rho = .1$ and $x = y = 0$, the first five steps are $(0, 0) \to (.2, .2) \to (.32, .32) \to (.392, .392) \to (.395, .395)$. At step 8, $x_8 = y_8 \approx .4916$. The distance between (x_8, y_8) and (x^*, y^*) is about $\sqrt{2 \cdot .0084^2} \approx .012$. At step 9, $x_9 = y_9 \approx .495$ and the distance is about .007.

We can reach the same conclusion analytically. As $x - y$ starts at zero and remains the same, x must equals y throughout Gradient Descent. Equation (2) then tells us that $x+y-1 = 2x-1$ shrinks by $1-4\rho$ in every step, and so does $2y - 1$. After t steps, $2x_t - 1$ equals $(1 - 4\rho)^t \cdot (2 \cdot 0 - 1) = -(1 - 4\rho)^t$. As $2x^\star - 1$ drops to zero, $2x_t - 2x^\star$ has to be $-(1 - 4\rho)^t$. Same holds for $2y_t - 2y^\star$. Finally, the distance from (x_t, y_t) to $(x^\star, y^\star)$ is

$$\sqrt{(x_t - x^\star)^2 + (y_t - y^\star)^2} = \sqrt{\tfrac{1}{4}(1 - 4\rho)^{2t} + \tfrac{1}{4}(1 - 4\rho)^{2t}} = \frac{1}{\sqrt{2}} \cdot (1 - 4\rho)^t.$$

When ρ is 0.1, it drops below 0.1 at the first step t when $0.6^t/\sqrt{2} \le 0.01$, which is $t = \lceil \log .01\sqrt{2}/\log .6 \rceil = 9$.

When the initial point is $(1, -1)$, the invariant quantity is $x - y = 1 - (-1) = 2$. Equation (2) now tells us that $x+y-1 = 2x-3$ shrinks by $1-4\rho$ per step and so does $2y+2$. Now $(1-4\rho)^t(2x_t - 3) = 2x_0 - 3 = -1$. So $2x_t - 3 = -(1-4\rho)^t$, and so must be $2(x_t - x^\star)$. As for y, $(1-4\rho)^t(2y_t+2) = 2y_0+2 = 0$, and y_t will never change. The distance at time t is $|x_t - x^\star| = (1/2)(1 - 4\rho)^t$. It drops below 0.01 as soon as $1/2 \cdot .6^t \le .01$ which is also $t = 9$.

(e) (**Extra credit**) Prove that, in general, the target $\mathbf{x}^\star$ of gradient descent is affected by the choice of initialization if and only if the columns of the input matrix A are linearly dependent.

Solution: There is an explanation for the calculations in part (d). The "directions" $(1, 1)$ and $(1, -1)$ of $x+y$ and $x-y$ are precisely the (unnormalized) eigenvectors of A^TA. Whenever $\mathbf{v}$ is an eigenvector of A^TA with eigenvalue λ, $\mathbf{v} \cdot (\mathbf{x} - \mathbf{x}^\star)$ scales by $1 - 2\rho\lambda$ in each iteration because

$$\mathbf{v} \cdot (\mathbf{x}' - \mathbf{x}^\star) = \mathbf{v} \cdot (I - 2\rho A^TA)(\mathbf{x} - \mathbf{x}^\star) = \mathbf{v}(1 - 2\rho\lambda)(\mathbf{x} - \mathbf{x}^\star) = (1 - 2\rho\lambda)\mathbf{v} \cdot (\mathbf{x} - \mathbf{x}^\star). \tag{3}$$

Viewed from the basis of the eigenvectors of A^TA, gradient descent is a very simple algorithm. If we move the origin to $\mathbf{x}^\star$, along each eigenvector, $\mathbf{x}$ scales by $1 - 2\rho \cdot$ eigenvalue.

The directions in which $\mathbf{x}$ does not change are then those eigenvectors associated to zero eigenvalues. So if A^TA has eigenvalue zero, $\mathbf{x}^\star$ will depend on $\mathbf{x}$ in the direction of the corresponding eigenvector. Conversely, if all eigenvalues are nonzero there cannot be two different convergence targets $\mathbf{x}$ and $\mathbf{x}^\star$ because they should be invariant under the application of (3). But (3) will bring $\mathbf{x}$ closer to $\mathbf{x}^\star$ in each iteration if when all eigenvalues are nonzero (and ρ is sufficiently small).

If this argument didn't make sense to you, there is a more direct way to prove it. The points of convergence of gradient descent are those $\mathbf{x}$ for which $\nabla f(\mathbf{x}) = 2A^T(A\mathbf{x} - b)$ is zero. If A's columns have a linear dependence $\mathbf{y}$ then $A\mathbf{y}$ equals zero, so if any point $\mathbf{x}^\star$ is a target so must be $\mathbf{x}^\star + \mathbf{y}$. It is not unique.

Conversely, if there are two such points $\mathbf{x} \neq \mathbf{x}^\star$ then $\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star) = 2A^TA(\mathbf{x} - \mathbf{x}^\star)$, so $\mathbf{y} = \mathbf{x} - \mathbf{x}^\star$ satisfies $A^TA\mathbf{y} = 0$. Multiplying by the row vector $\mathbf{y}$ on the left gives $0 = \mathbf{y}A^TA\mathbf{y} = \|A\mathbf{y}\|^2$. So $A\mathbf{y}$ must be zero and $\mathbf{y}$ is a linear dependence of the columns of A.

Question 3

The condition number κ of a linear system is a measure of proximity to linear dependence. In Lecture 2 we argued that it controls the convergence rate of Gradient Descent. You will investigate it in this question.

(a) The condition number of a PSD matrix S is defined as the ratio between its largest and its smallest eigenvalues. Find the condition number of the matrix

$$S = \begin{bmatrix} 1 & 1 \\ 1 & 1.01 \end{bmatrix}.$$

You may use any method you like, but you must explain how you arrived at your answer.

Solution: I ran QR iteration. The first two steps give

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.01 \end{bmatrix} \to \begin{bmatrix} 2.005 & -.005 \\ -.005 & .005 \end{bmatrix} \to \begin{bmatrix} 2.005 & 10^{-5} \\ 10^{-5} & .005 \end{bmatrix}.$$

The eigenvalue estimates quickly converge to 2.005 and 0.005. The condition number is their ratio, which is around 401.

(b) The condition number of a general matrix A is defined as the square root of the condition number of the PSD matrix A^TA: $\kappa(A) = \sqrt{\kappa(A^TA)}$. Prove that when S is PSD this is consistent with the definition in part (a), i.e., $\kappa(S) = \sqrt{\kappa(S^TS)}$.

Solution: If S is symmetric, i.e., $S^T = S$, then the eigenvalues of S^TS are the squares of the eigenvalues of S: If $\lambda, \mathbf{v}$ is an eigenvalue-eigenvector pair of S, then

$$S^TS\mathbf{v} = S^T\lambda\mathbf{v} = \lambda S\mathbf{v} = \lambda^2\mathbf{v},$$

so $\lambda^2, \mathbf{v}$ is an eigenvalue-eigenvector pair of S^TS. As S is positive semidefinite its largest and smallest eigenvalues are positive, so the largest and smallest eigenvalues of S^TS must be their squares. The condition number $\kappa(S^TS)$ is their ratio, so it is the square of $\kappa(S)$: $\kappa(S^TS) = \kappa(S)^2$.

(c) Use part (a) to calculate the condition number of

$$A = \begin{bmatrix} 1 & 0.9 \\ 1 & 1.1 \end{bmatrix}.$$

Solution: A^TA equals the matrix S in part (a), so $\kappa(A^TA)$ is about 20.

(d) Prove that the condition number of a square matrix A is finite if and only if its rows are linearly independent. Use this equivalence to explain qualitatively why the answer in part (c) is so large. You may use the fact that if the rows of A are linearly dependent then so are its columns.

Solution:

If the rows of A are linearly dependent then so are its columns. Then $A\mathbf{x}$ is zero for some nonzero $\mathbf{x}$ representing this linear dependence on the columns, and so is $A^TA\mathbf{x}$. So A^TA must have zero as its smallest eigenvalue. Its condition number is something divided by zero which is infinite.

If A^TA has infinite condition number then $A^TA\mathbf{x}$ is zero for some nonzero $\mathbf{x}$. If $A\mathbf{x}$ is nonzero, then it is a linear dependence between the columns of A^T which are the same as the rows of A. If it is zero, then $\mathbf{x}$ is a linear dependence between the columns of A. The rows of A must then also be linearly dependent.

The answer in (c) is relatively large because the rows of A are "almost" linearly dependent. If we perturb the second column by $(+0.1, -0.1)$ a dependence is created and the smallest eigenvalue is zero. As 0.1 is a small number relative to the scale of the matrix we expect the smallest eigenvalue of the original A^TA to be no larger than this perturbation, namely on the order of $0.1^2 = 0.01$. Its spectral norm, in contrast, is around 4; power iteration on A^TA should stabilize close to the direction $(1, 1)$. The condition number of A is indeed close to $\sqrt{4/0.01} = 20$.

Question 4

Find your personalized hidden parity instance here: `https://andrejb.net/csi4103/hw/25H01.html`

The instance consists of 20 equations in 12 unknowns. + and − stand for the numbers +1 and −1, respectively. You need to find a subset of the columns that multiplies to the right-hand side.

Write down your solution in the form of indices of the relevant columns (in increasing order). For instance, the solution to the example in Section 7 of Lecture Notes 1 is: `2, 4`.

Explain clearly how you arrived at your solution. Undocumented computer code will not be entertained as a satisfactory explanation.

Solution: I solved my instance using simple Gaussian elimination. The program produced the solution $3, 5, 7, 10$. I verified that the corresponding columns of the left-hand side add to the right-hand side modulo two.

To implement Gaussian elimination, I first converted each row of the instance into a number whose i-th bit is the parity of the i-th symbol on the left-hand side and whose zeroth bit is the parity of the right-hand side. I wrote Gaussian elimination in two steps: A forward pass and a backward pass.

The objective of the forward pass is to reduce the system into a diagonal form in which variable i appears in the i-th equation but no lower-indexed variables appear in it. To implement it I went over the variables in order, looked for the first equation in which the variable appears, swapped it with the first unused equation in the system, and then added it to all the remaining equations in which it also appears by XORing their number representations.

In the backward pass I read off the variables in reverse order. The variables that evaluate to 1 were included in the assignment and were eliminated from all equations in which they appear by XORing the corresponding zeroth bits which represent the right-hand side.

This instance is small enough that a brute force search algorithm that tries all 2^{12} possible candidate solutions should have worked just as well.