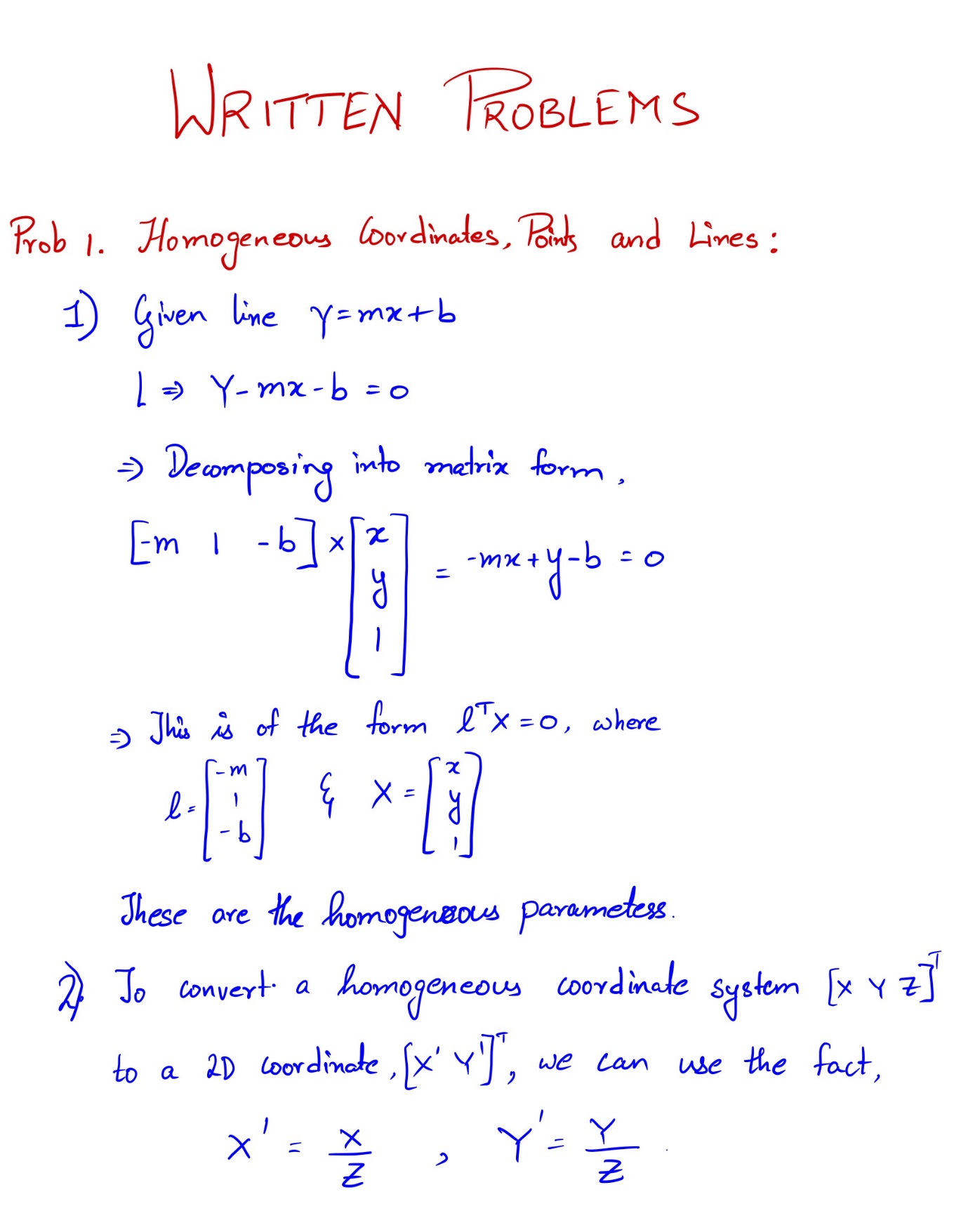
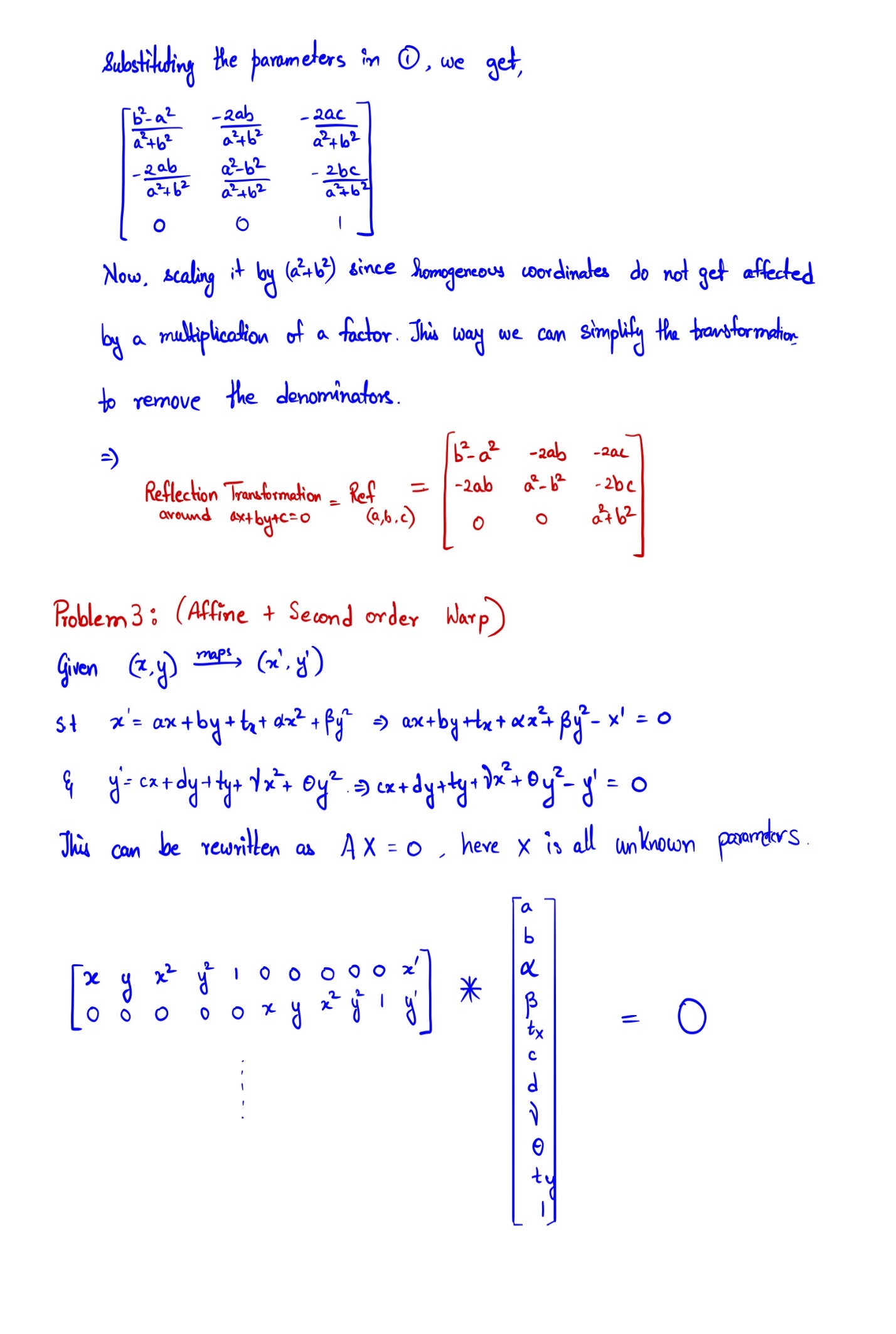
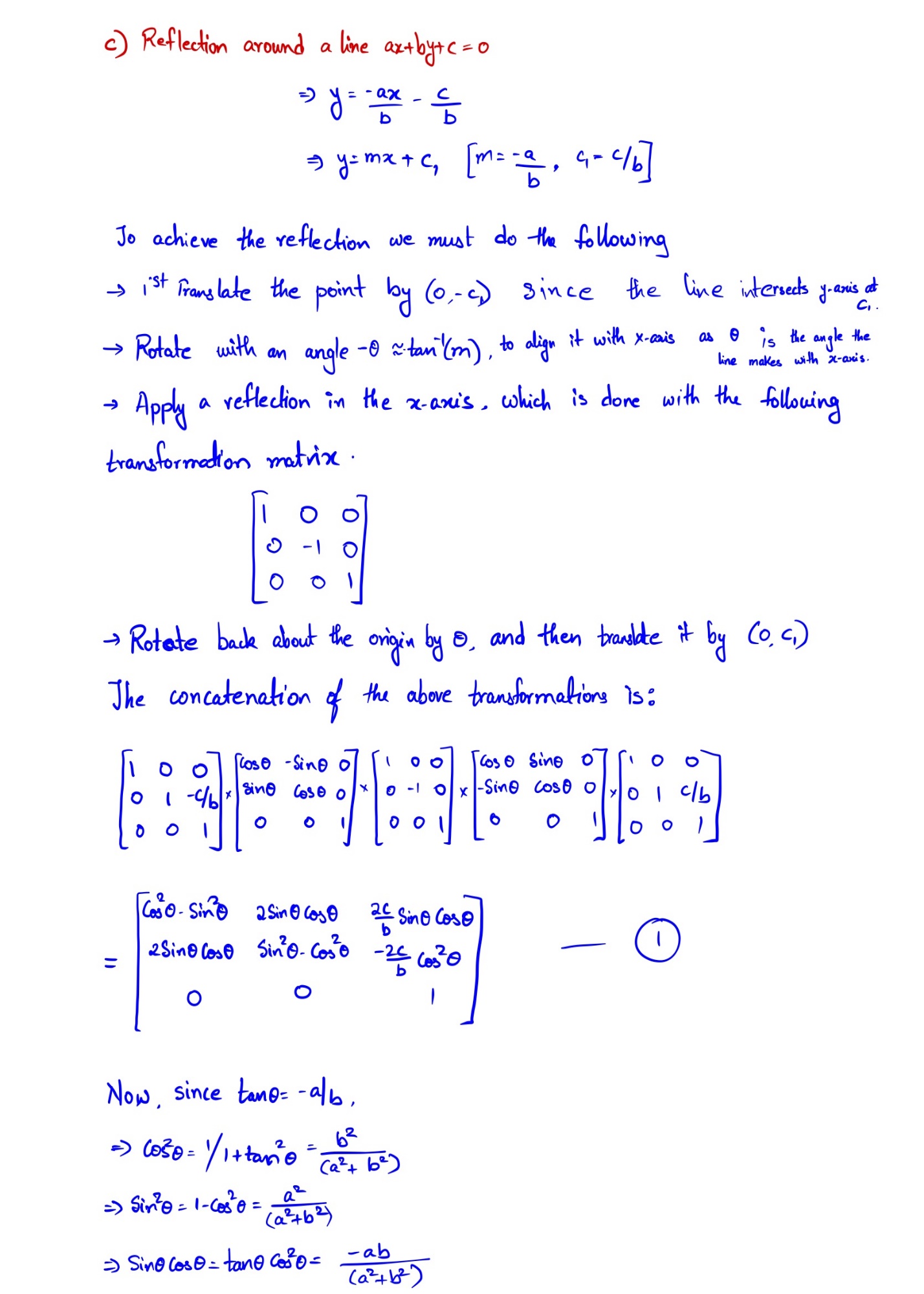
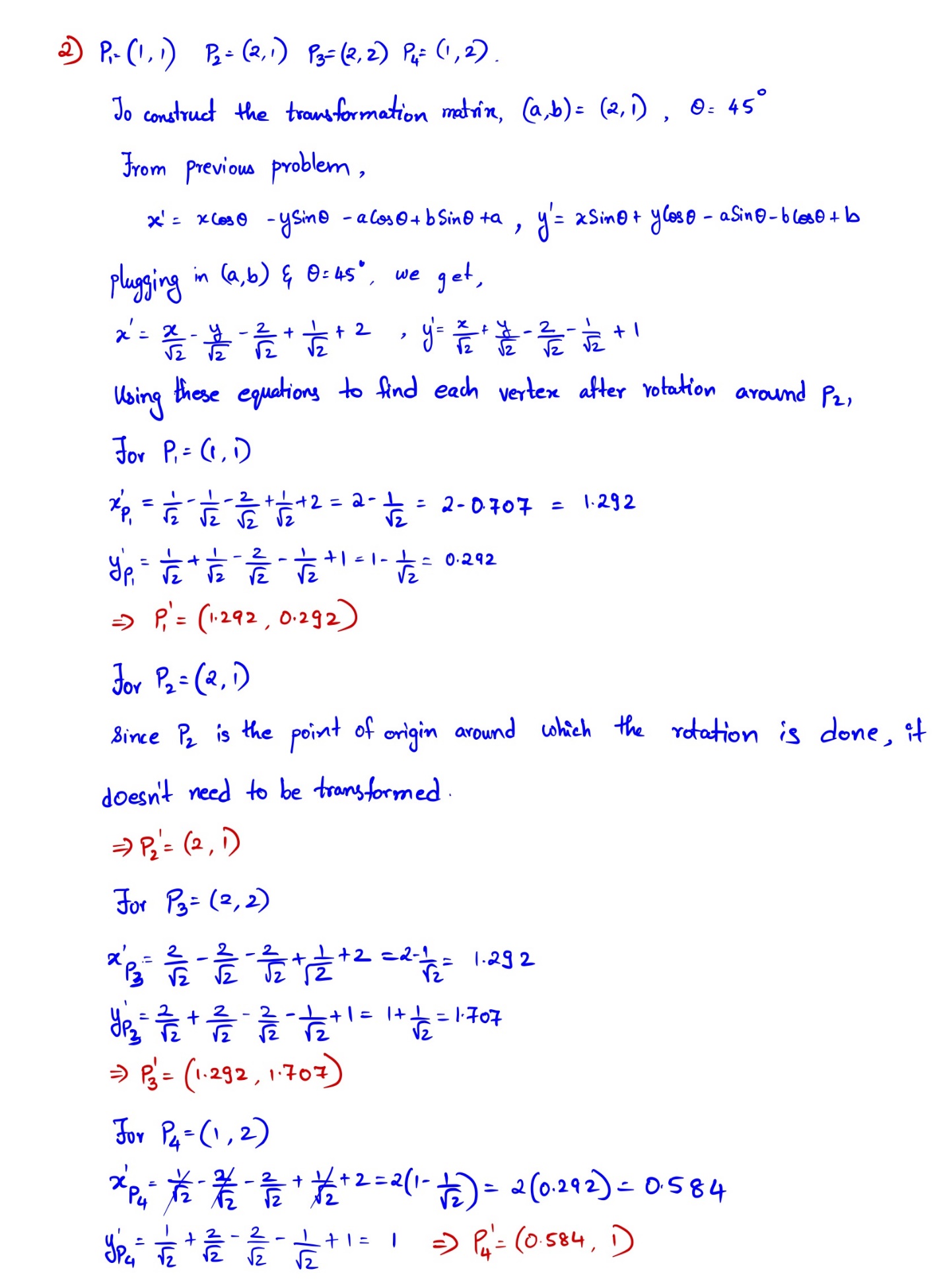
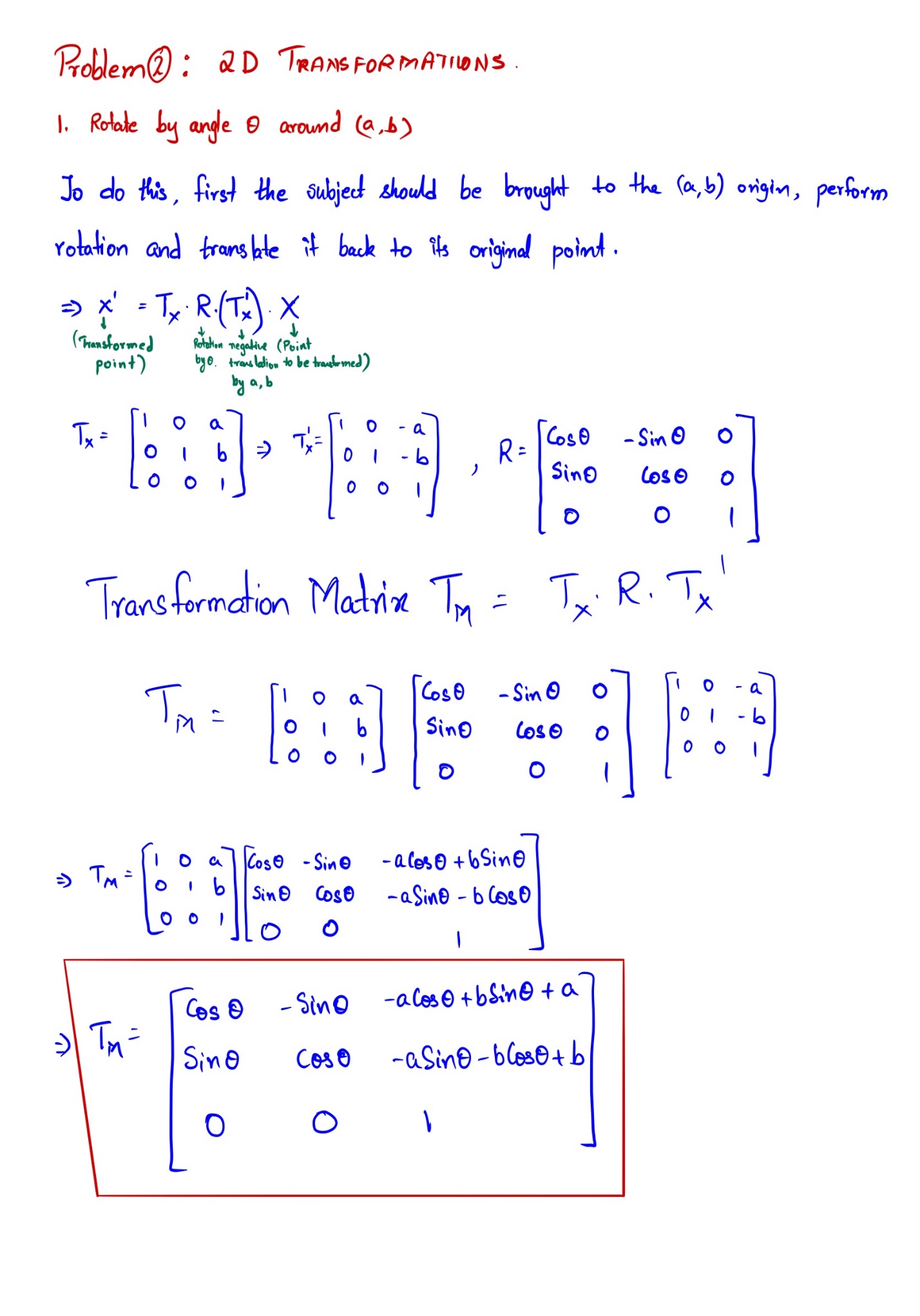
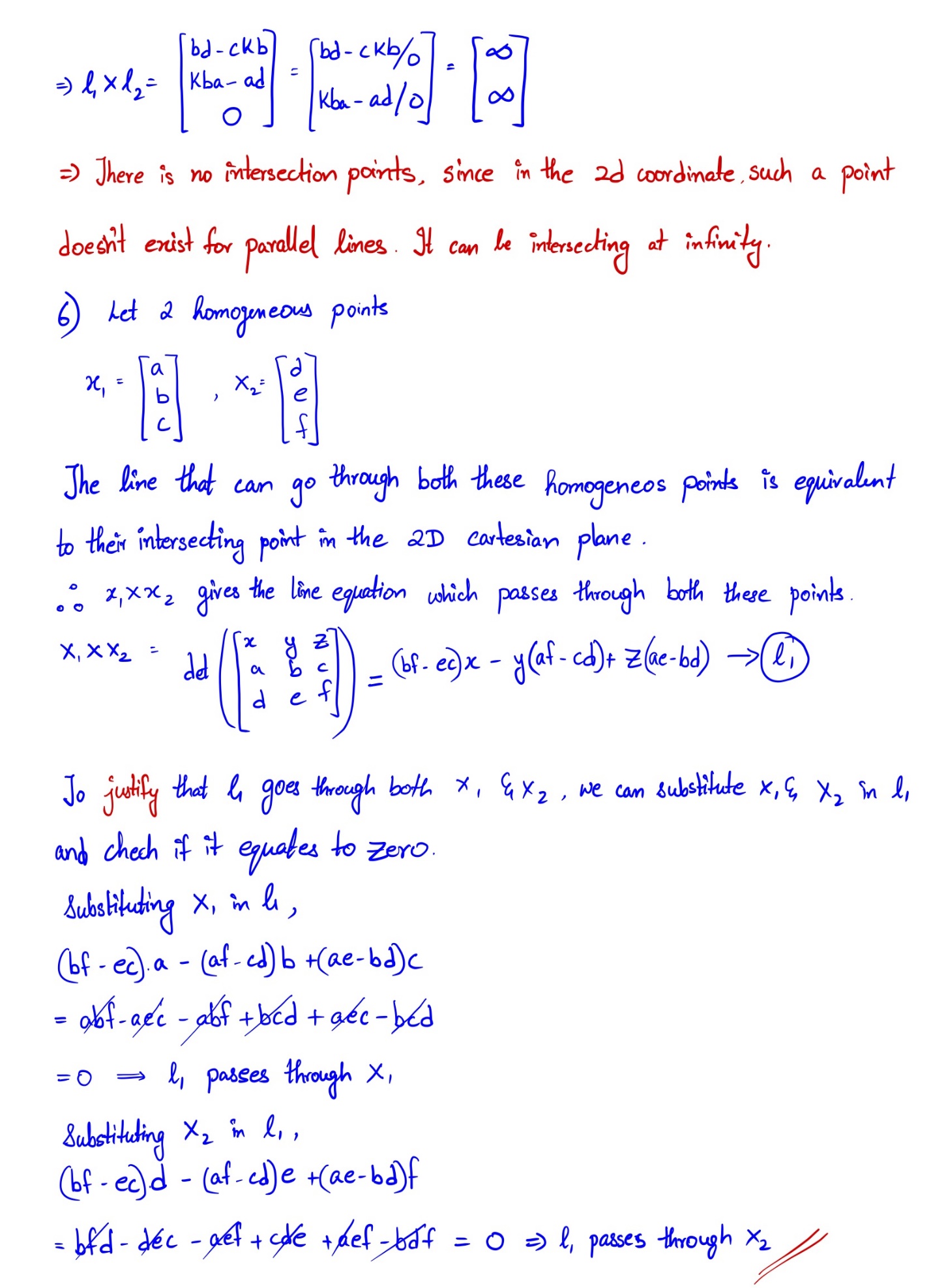
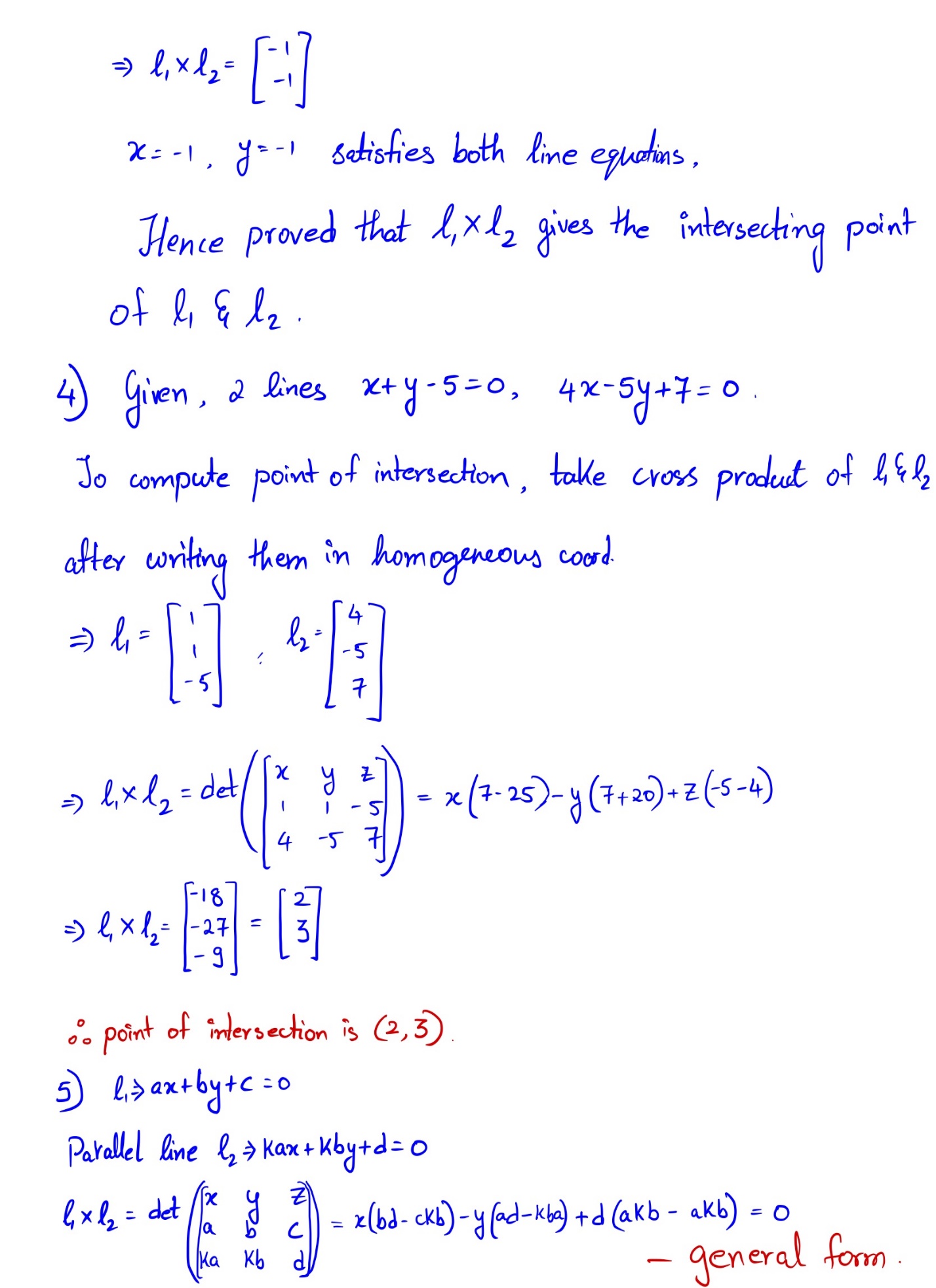
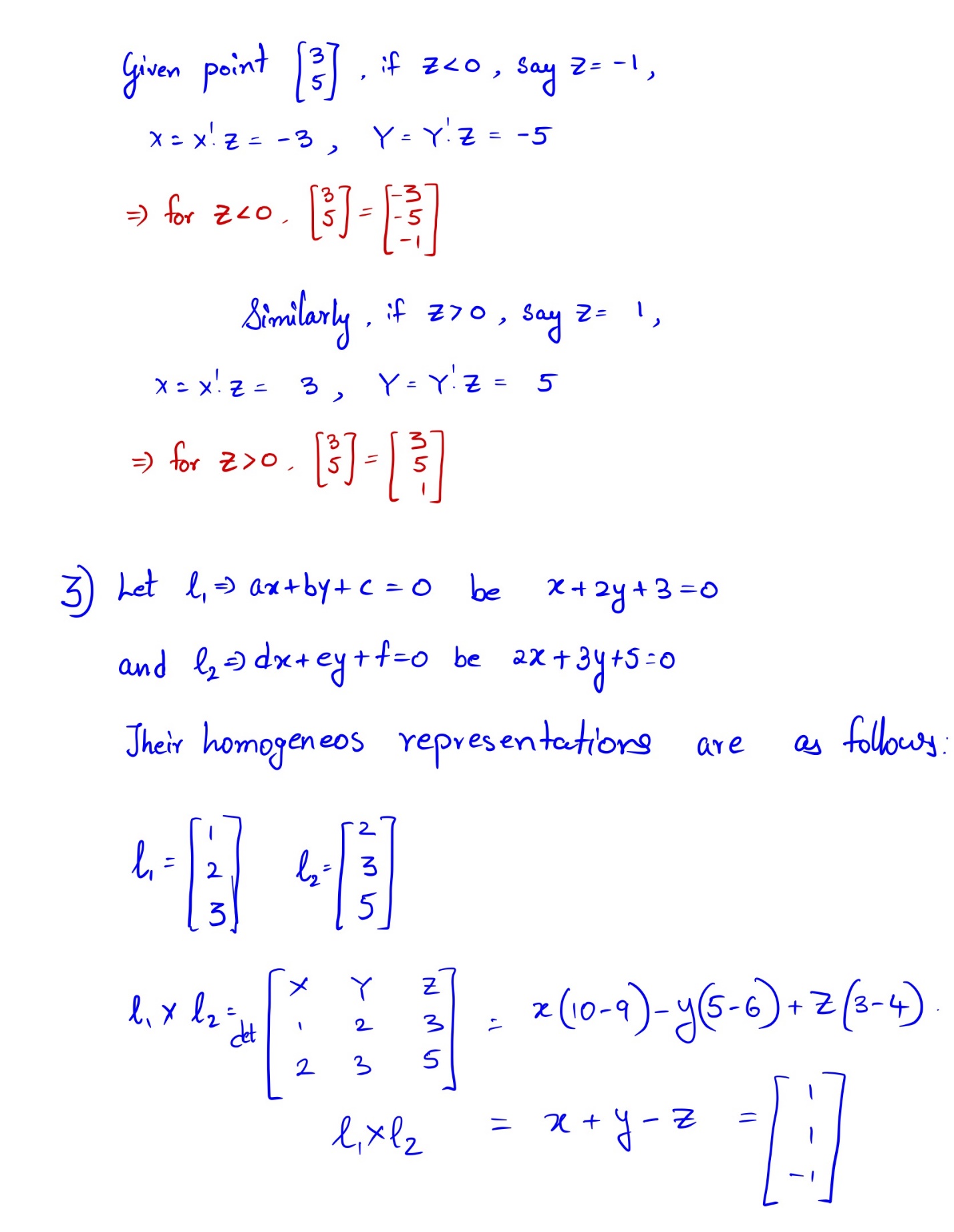
**EEE 598 ASSIGNMENT - 2**

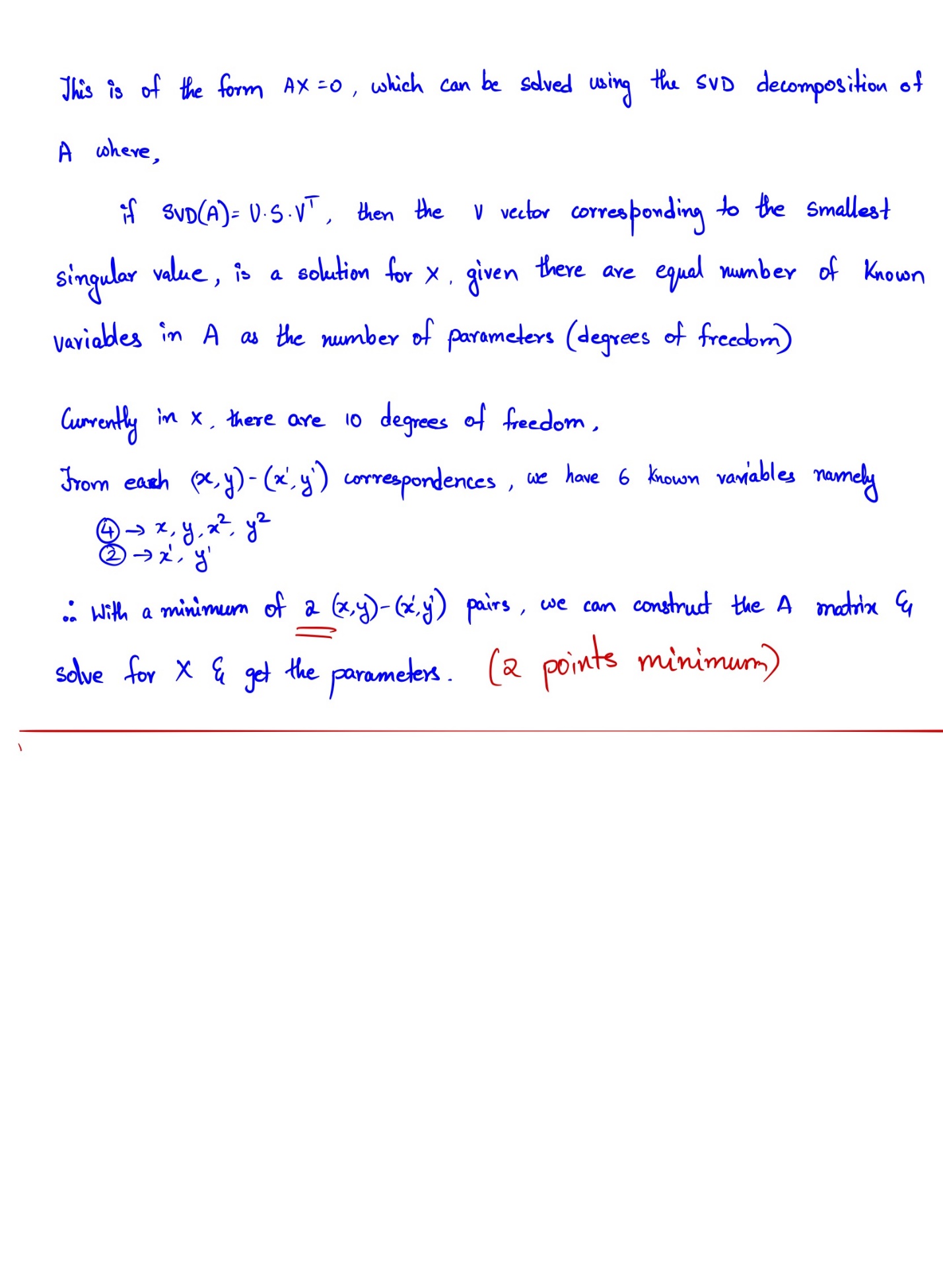
**REPORT**

BY: Pranav Bajoria

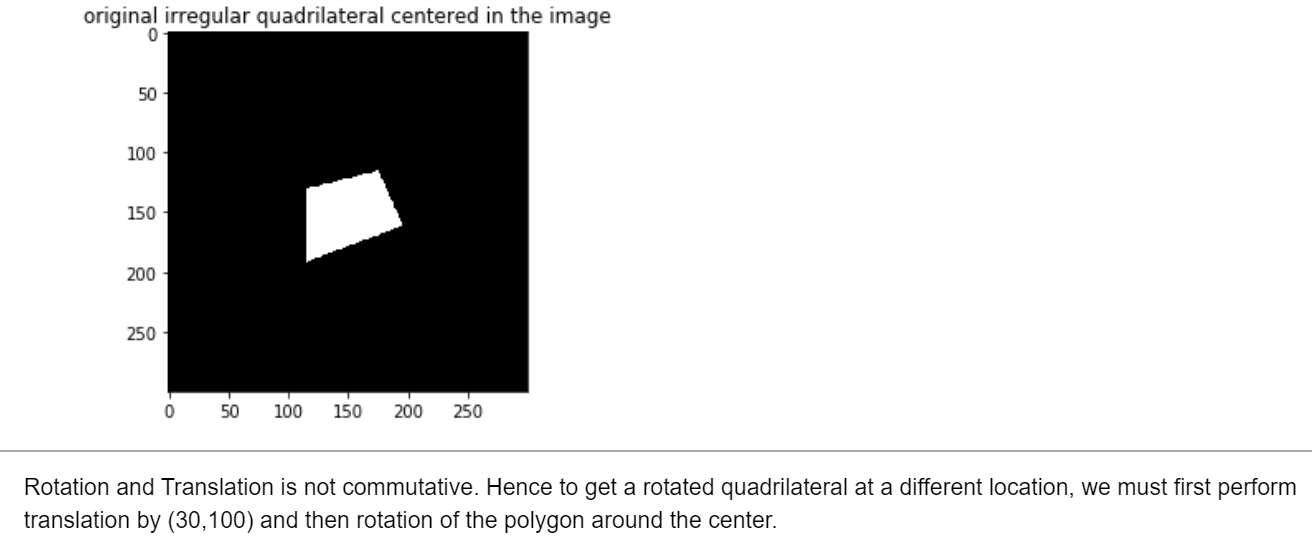
ASU ID: 1215321107



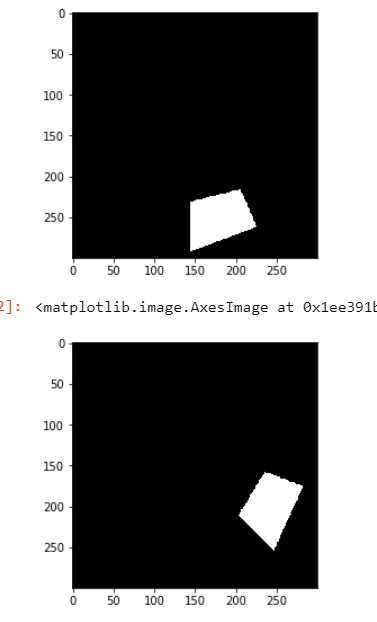
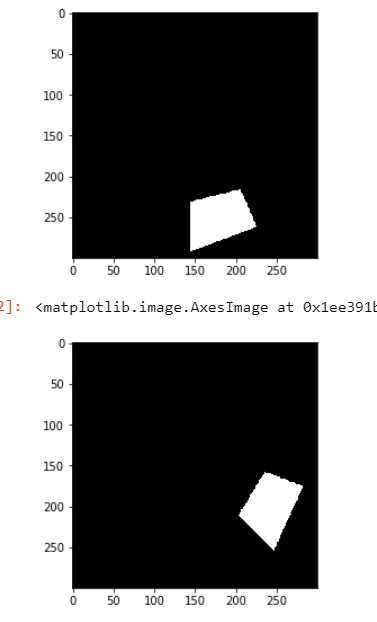




**CODING ASSIGNMENTS**

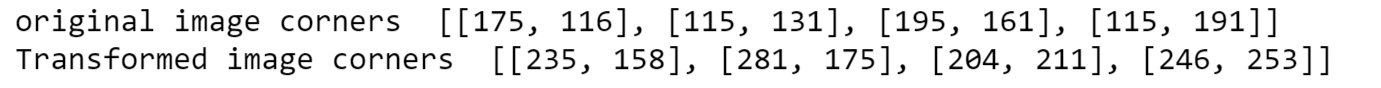
Problem 4. 

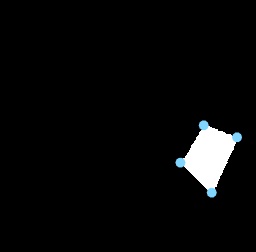
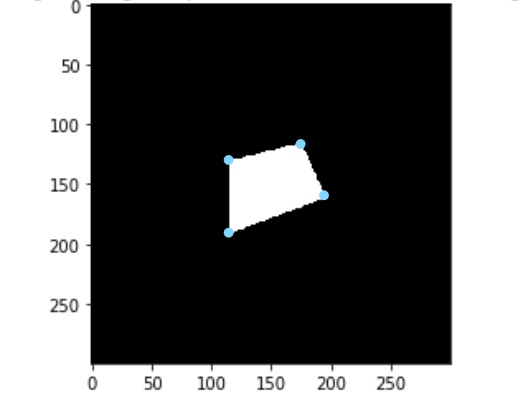
* Translation and rotation done



4.2) I implemented the Harris Corner detection from scratch. Check the jupyter notebook for the code. The most challenging part proved to be thresholding since there were many corners detected due to the scale of the image.

* The corners detected for each image were:

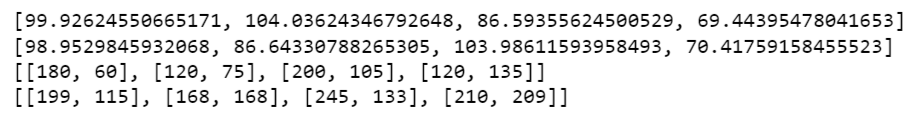




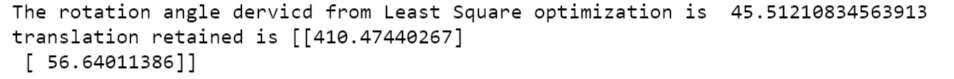
For feature descriptor, I implemented angles at each corner feature. I made sure that the polygons had unique angles while constructing to avoid any false matches.

To implement this, I constructed vectors with corners and added the angles from 1 corner to the other 3. I used the angles between vectors formula as shown below

These were the angles found for both the images and then using some error threshold, corners were matched as shown below.

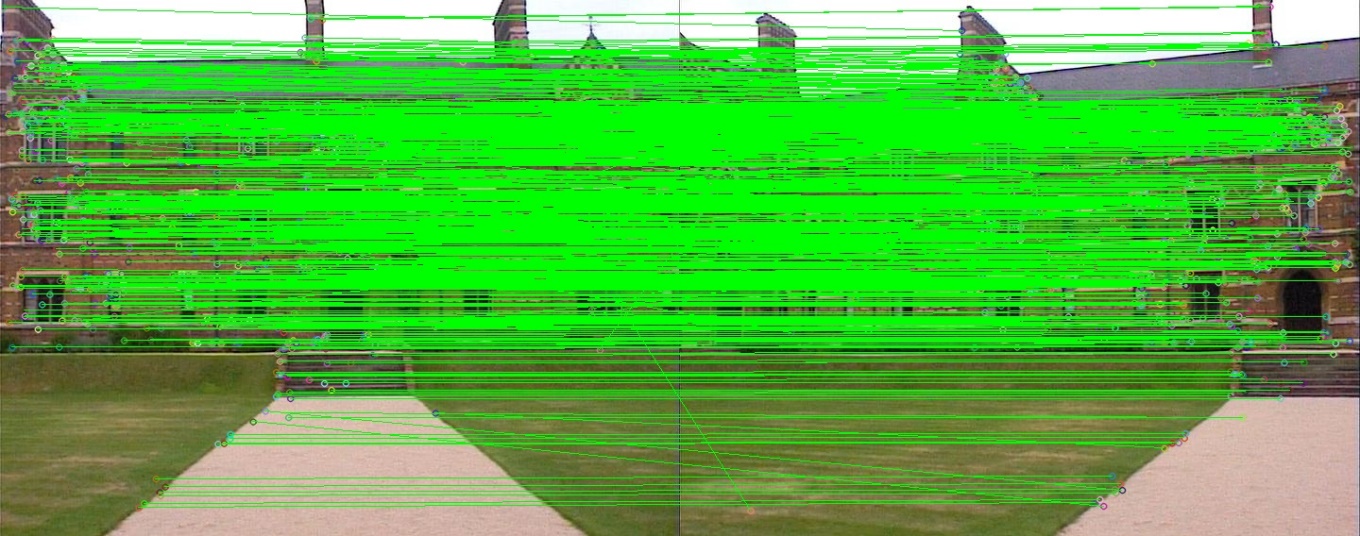


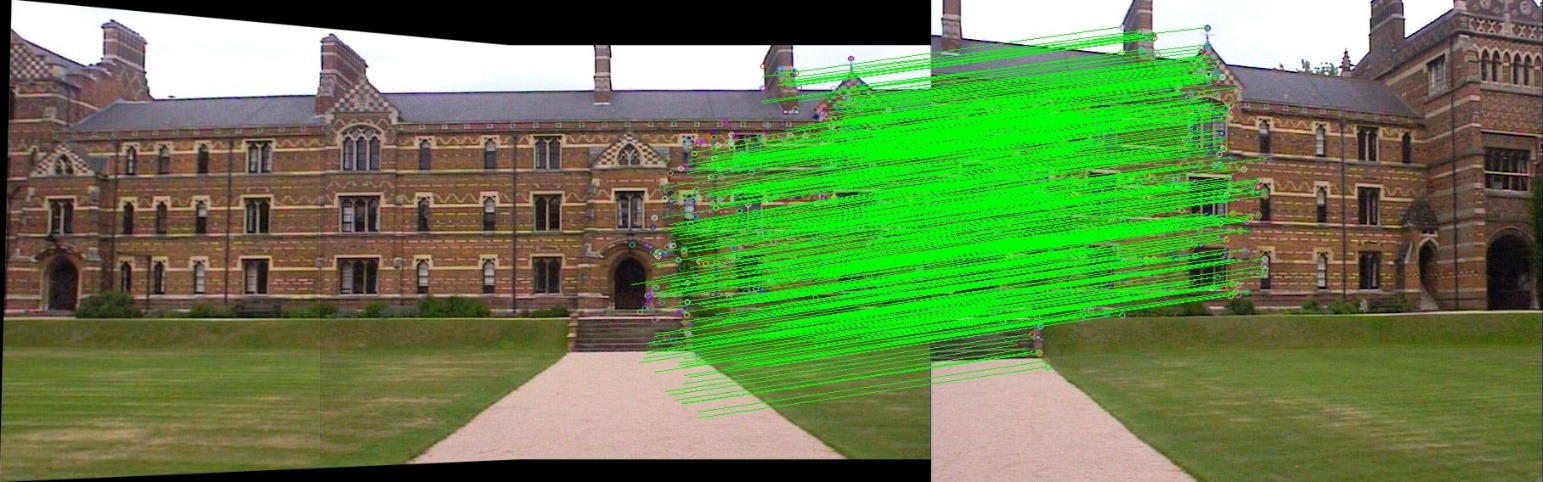
4.3) I implemented the linear least squares to get back the rotation and translation from transformed image w.r.t original image. It can be noted that, LLS optimization can give the rotation and translation relative to the correspondences. Therefore, recovering the actual transformation performed would be possible only if we did rotation first and then translation. However, the algorithm could still retrieve the angle well, but not the translation.



5) **Image Stitching.**

To derive the good matches or tentative matches, I first implemented KNN from scratch on the feature vector after performing SIFT detection operation. Then I used Lowe’s ratio to eliminate matches that had the 2NN too close to the 1NN. This is done to make sure only the matches that has only 1 true correspondence is considered.

****

Then I implemented RANSAC using these good Matches to estimate Homography with the maximum inliers. By keeping the geometric distance threshold of 5 increasing the iterations to 10000, I could get a good 88% inlier accuracy for left and center and 68% when matching the below case. The inliers for the warped left and center image with respect to the right image is illustrated below.

**Final Stitching:**

