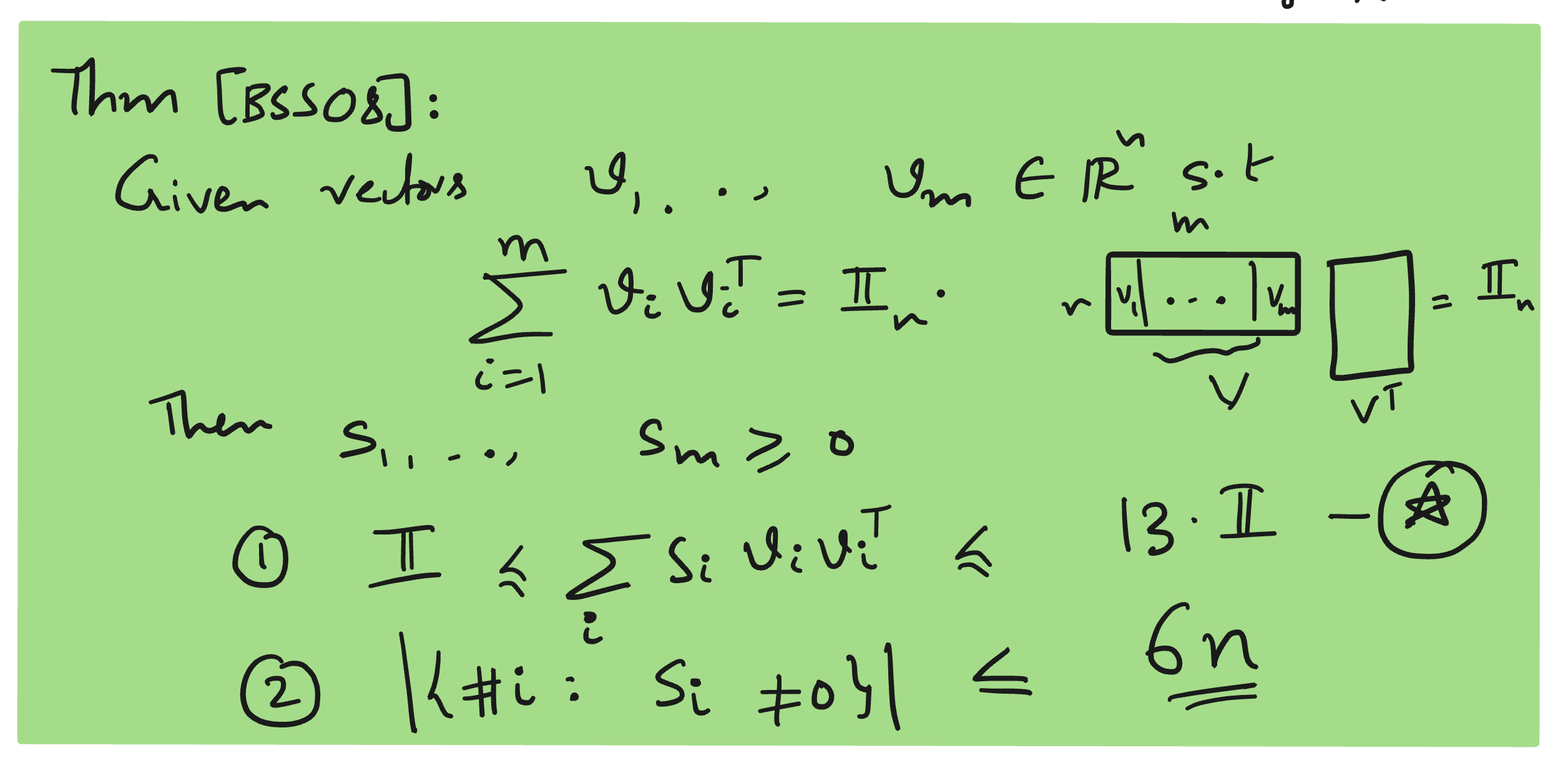
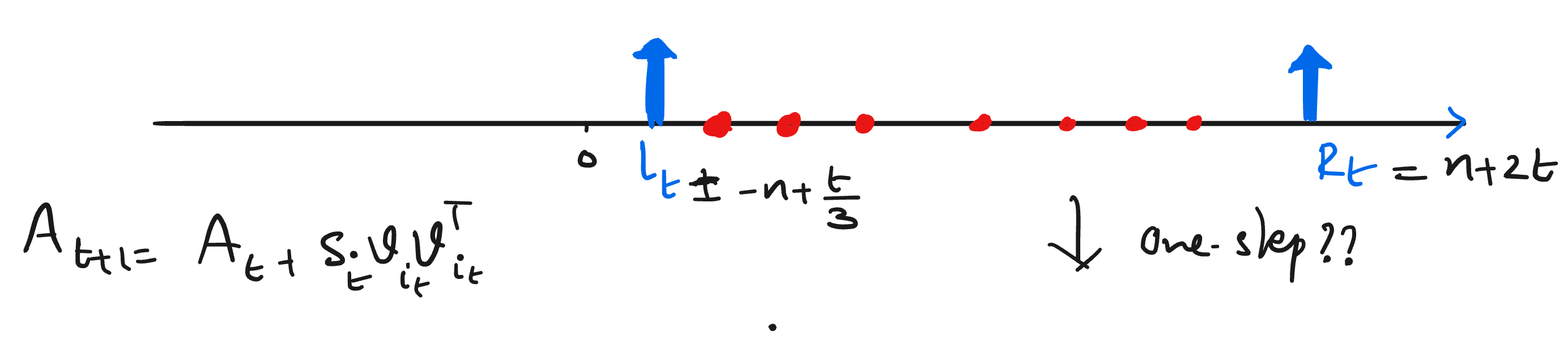
# Problem 1

First, we need to change this part:



The first and second bullet points will need to be written as

The left and right barriers at each iteration will also need to be changed



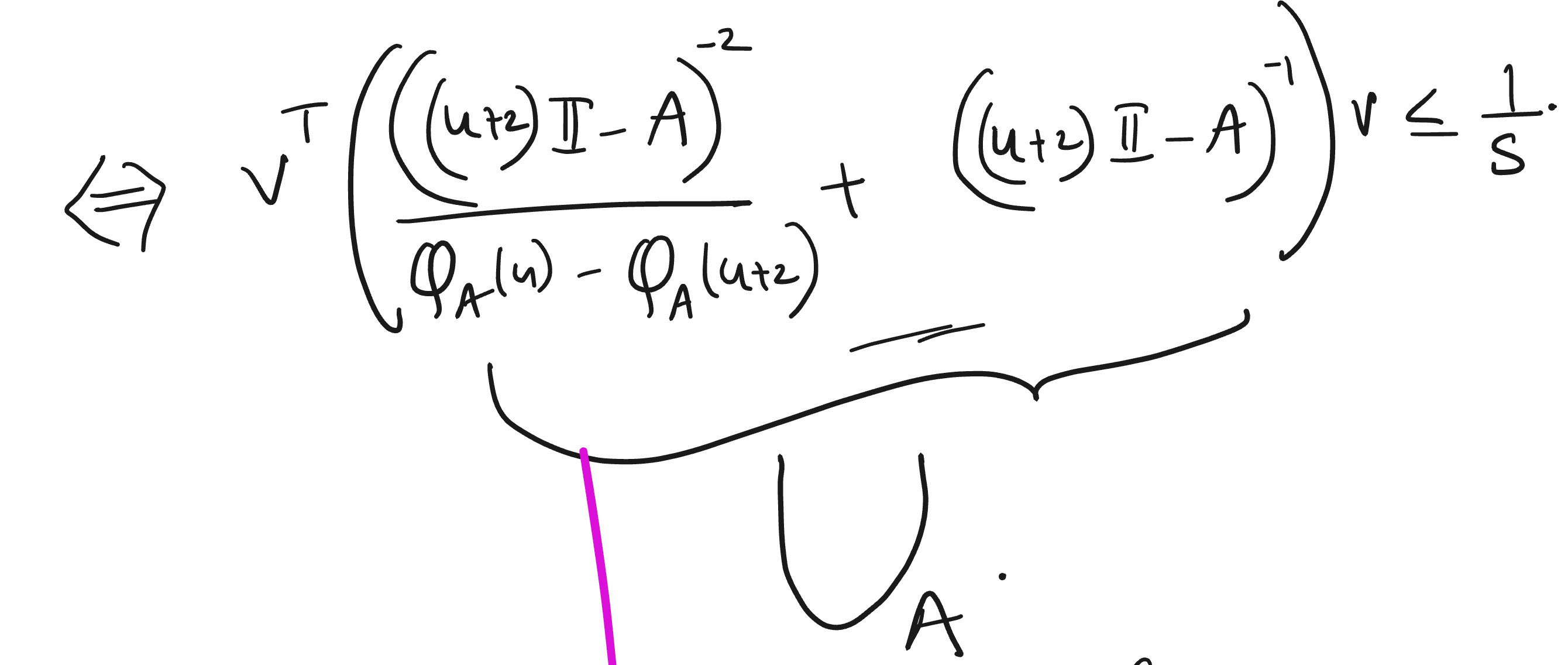
Change and as

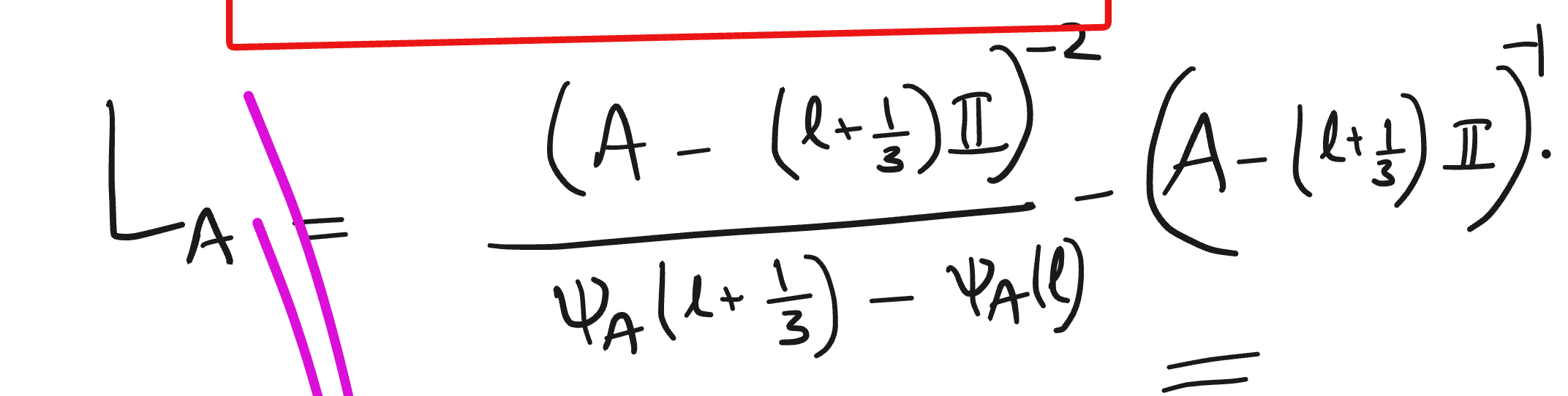
Notice that with these new and , we have and which is as desired by .

Also, we need to change part circled in part in the main inductive step:



Lastly, the and matrices need to be updated:





Lastly, we need to verify with the modified and .

# Problem 2

We have the left-hand side

So,

Notice that so we can put a trace operator in front. Also note that expectation and trace commute:

Now, using the Jacobi’s identity, we have

Using the above equation, it can be shown that the right-hand side

This completes the proof…

# Problem 3

With the help from <https://cstheory.stackexchange.com/questions/18756/whats-the-meaning-of-input-size-for-3-sat>, here is a way to encode a 3-CNF formula with variables:

|  |
| --- |
| Here's how to code a 3-CNF formula with variables in bits. Every clause has either 0, 1, 2 or 3 negations; we may permute so that the negated literals appear first in any clause. Now, fix an enumeration (say, lexicographic) of all the possible 3-tuples of variables. For each kind of clause (0, 1, 2 or 3 negations), make a list of bits such that bit is 1 if, and only if, the 3-tuple is a clause with that number of negations. The input is those four lists. |

With the above encoding scheme (call it ) and a given , we now have a new encoding scheme to encode all possible 3-CNF along with its variable assignment. This encoding scheme will take up bits. The first bits represent the 3-CNF and the last bits represent the associated variable assignment; you just concatenate them. Call this new encoding scheme .

Now, we can define the polytope . Max-3SAT then can be described using the above formulation. The cost function follows (roughly speaking, you encode the 3CNF formula for represents using ).

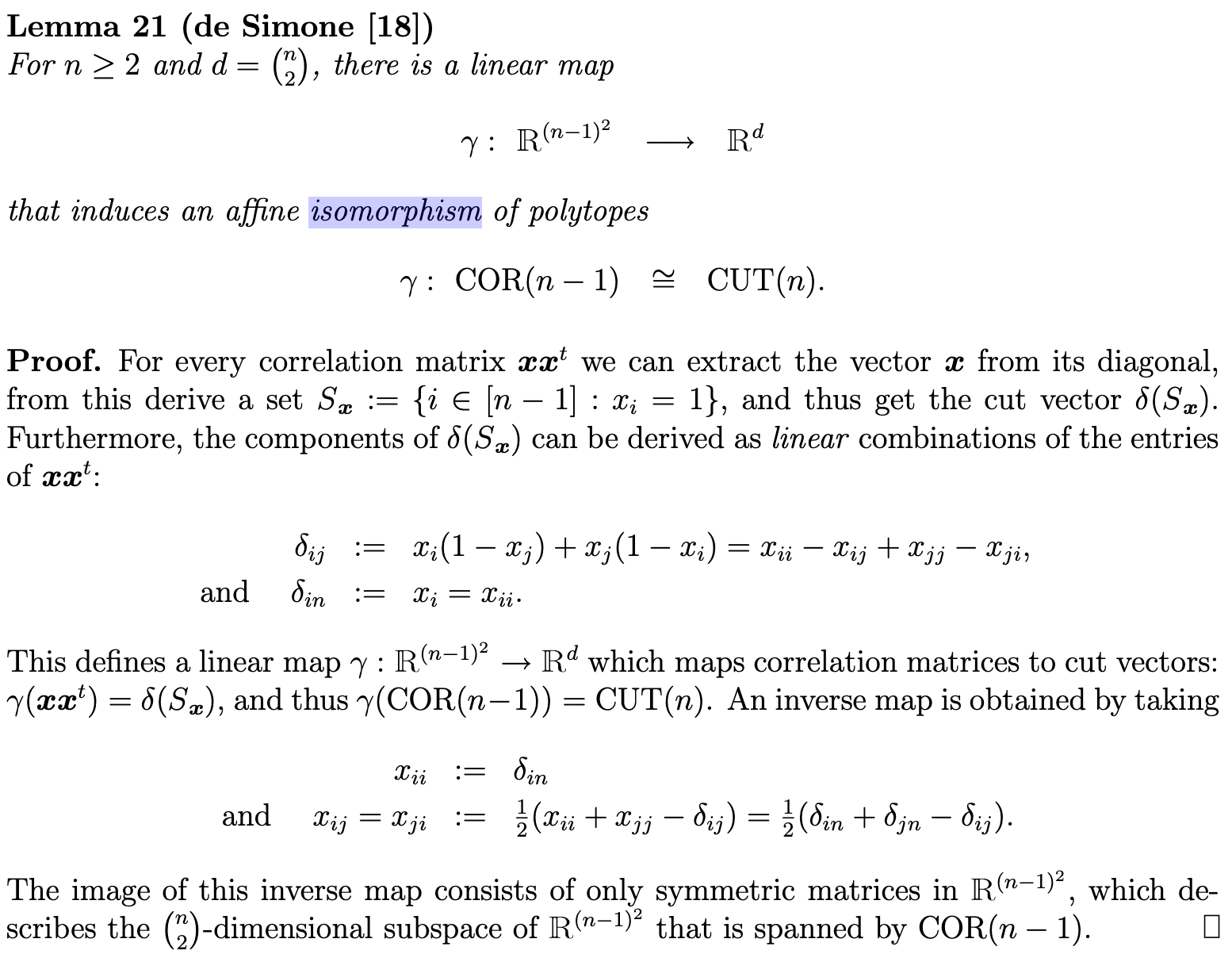
Define

The problem can then be written as . This works because the polytope doesn’t depend on .

# Problem 4

From <https://arxiv.org/pdf/math/9909177.pdf>, correlation polytope is isomorphic to

Below is the excerpt:

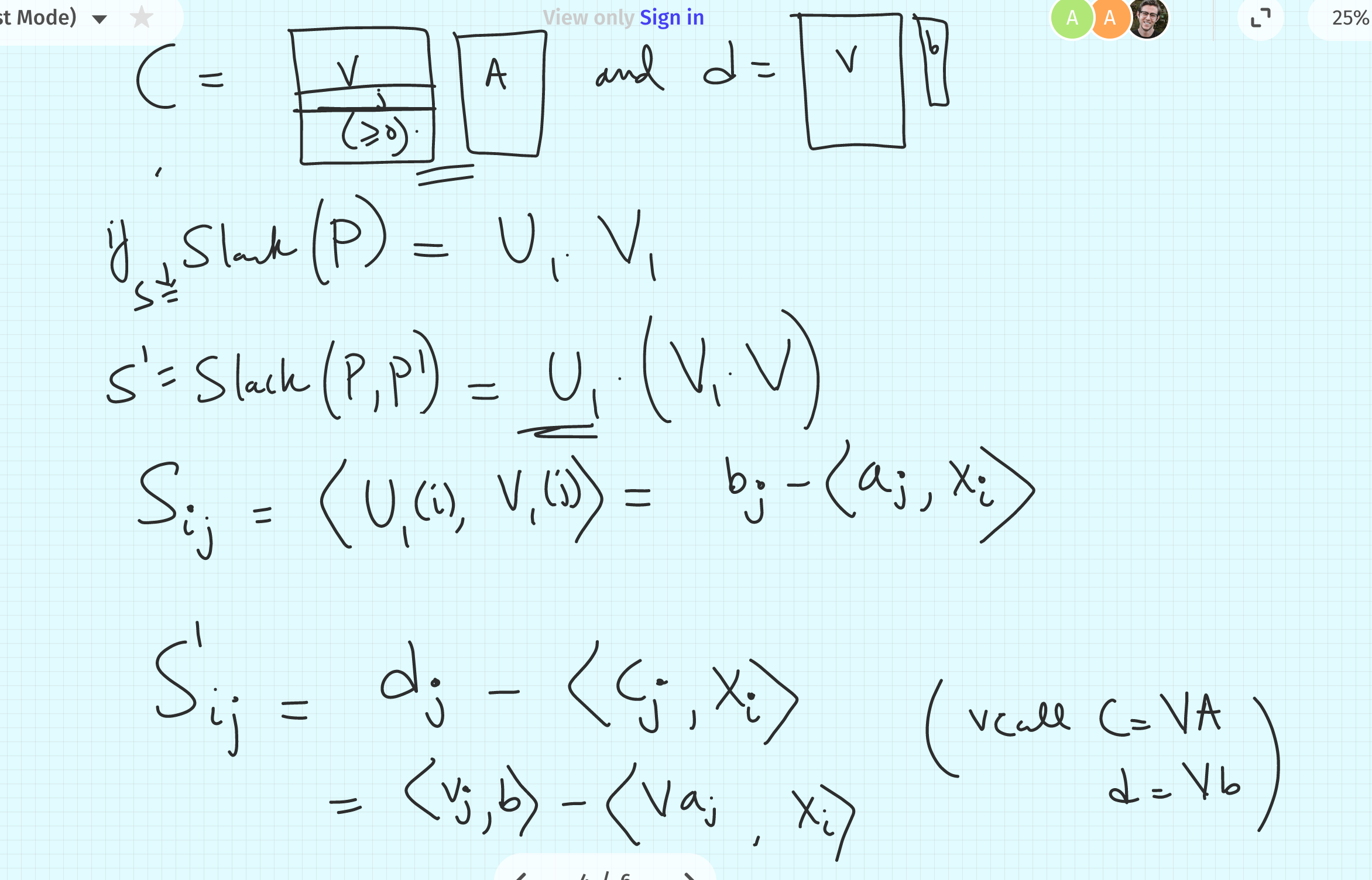


Because:

1. we derived in class that the extension complexity of is
2. The above excerpt shows that correlation polytope is isomorphic to cut polytope,

We can conclude that the extension complexity of cut polytope is also

# Problem 5



Following in class derivation, we have .

Now, assume has non-negative rank with decomposition . We can write .

Write

This tells us that there exists a non-negative decomposition for where and

This completes the proof…