**m --S**

**W 1ai)**

U is an upper bound for a sequence iff for all n, U ≥ an

1 ≥ sin(n) ≥ -1 for all n ∈ N

1/n ≥ sin(n)/n = an for all n ∈ N

1 ≥ 1/n for all n ∈ N and !(n=0)

1 ≥ an for all n ∈ N

So 1 is an upper bound

**ii)**

Take un = 1/n and ln = -1/n and an = sin(n)/n

By Sandwich Theorem we get -1/n <= sin(n)/n <= 1/n

We need to prove that un is greater than an for all n as we know that un converges to 0

We need to prove that ln is greater than an for all n as we know that ln converges to 0

We get -1 <= sin(n) <= 1 which is defined by sin.

And so we have that sin(n)/n converges to 0

**b)**

Using an = (4x - 1)n / nn we need to calculate the radius of convergence by doing an+1 / an

We end up with (4x - 1) \* nn / (n+1)n+1

The n part of the equation tends to 0 as n reaches infinity and so the overall thing tends to 0

This means that the radius of convergence is infinity

And so the series converges for all values of x

**c)**

sin(t) = t - t3/3! + t5/5! - …

sin(t)/t = 1 - t2/3! + t4/5! - …

integral(sin(t)/t) = t - t3/(3\*3!) + t5/(5\*5!) - …

Maclaurin series expansion = x2n+1 \* (-1)n / [(2n+1) \* (2n+1)!] when summing between n=0 and infinity

Apply D’Alembert’s LRT to get a limit L = 0 irrespective of the value chosen for x. Then L < 1 for all choices of x. The series converges for all x. So the radius of convergence is r = infinity.

ln(1+x2) = x2 - x4/2 + x6/3 - …

ln(1+x2)/x2 = 1 - x2/2 + x4/3 - …

Maclaurin series expansion = x2n \* (-1)n / (n+1) when summing between n=0 and infinity

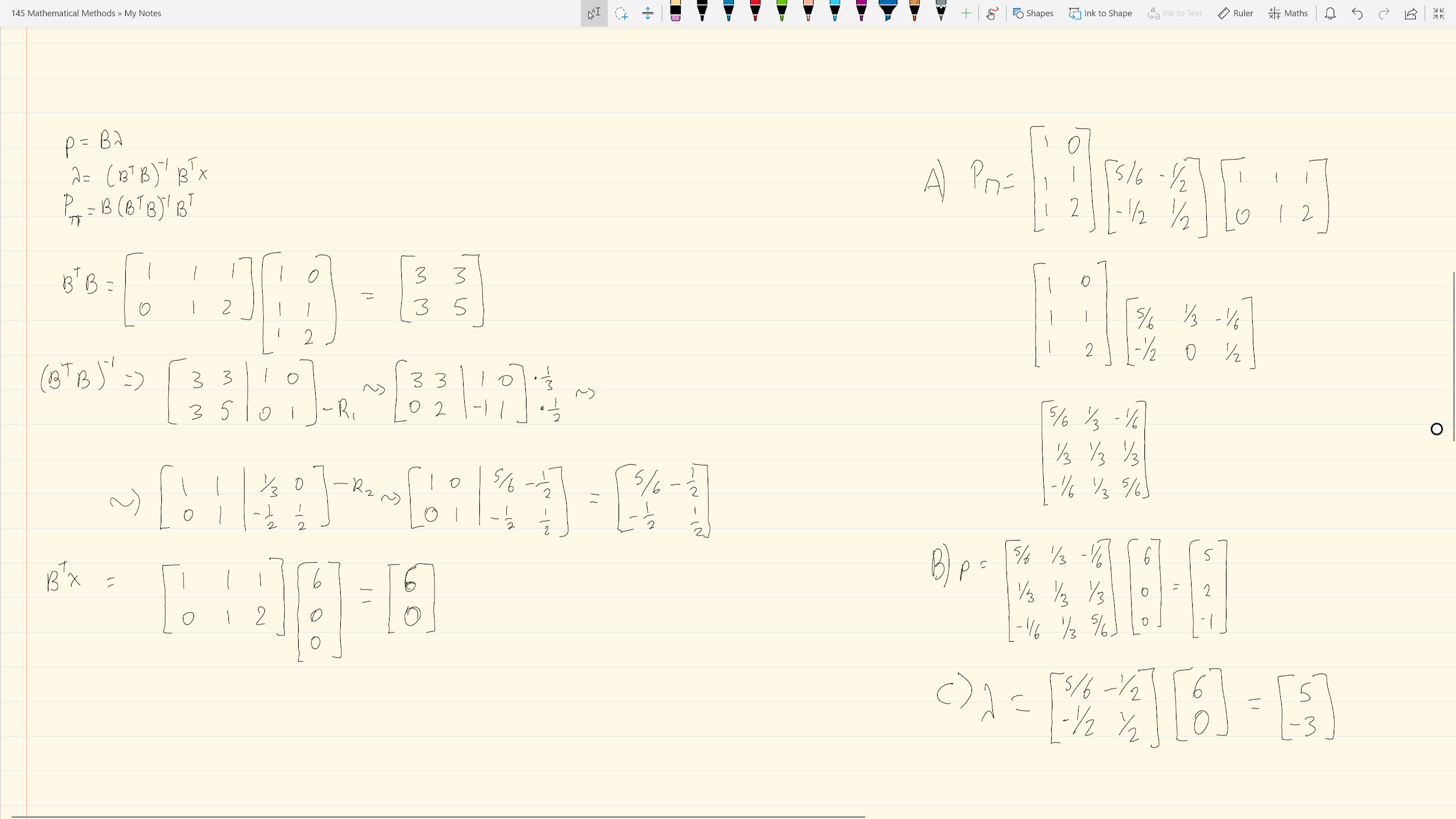
H

The expansion seems right but Wolfram Alpha gives a radius of convergence of 1 (|x|<1)

Power series of ln(1+x) is only valid when |x|<1

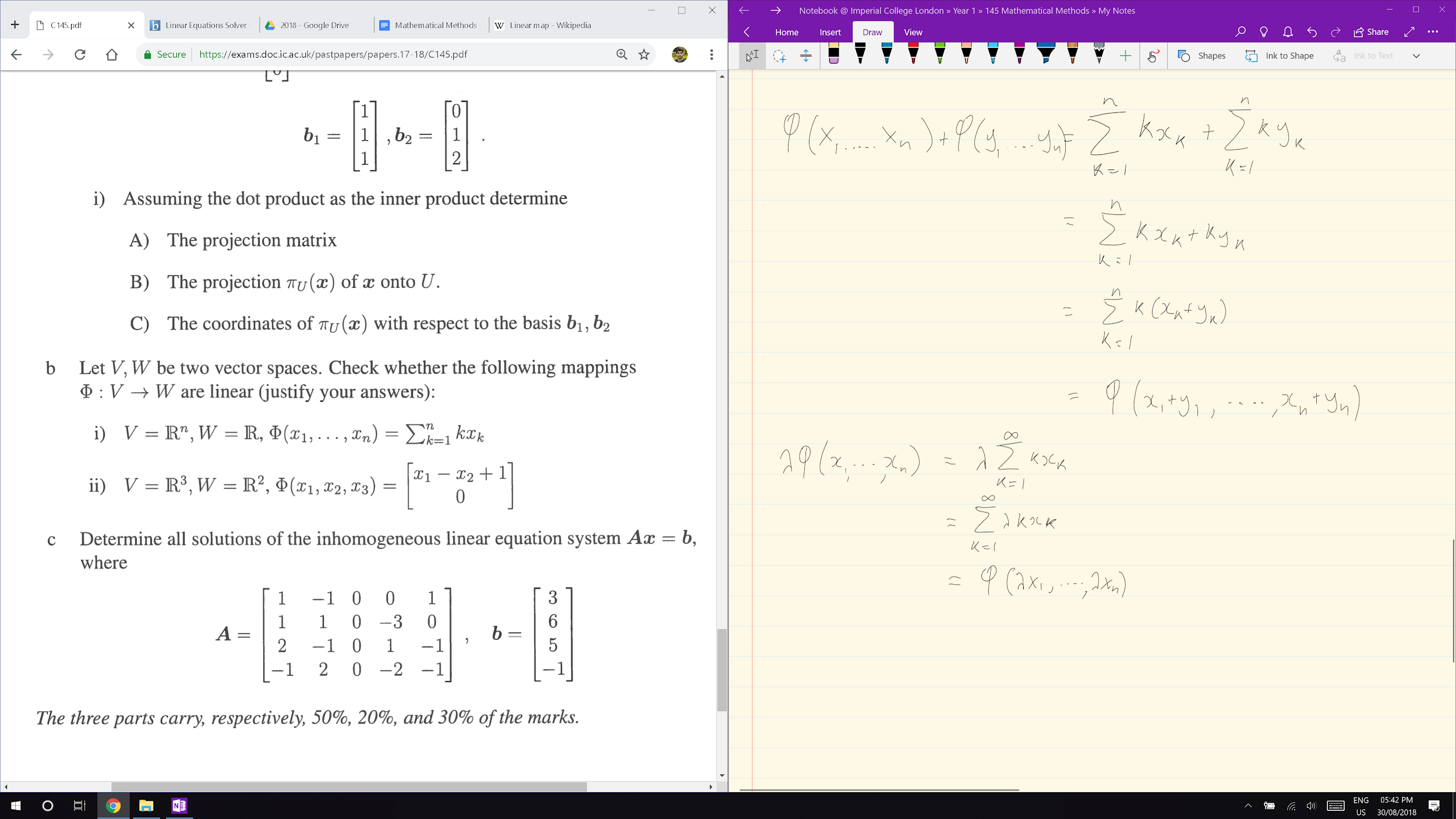
This next part is super dodgy so take it with a giant lump of salt if I end up putting anything here

**2ai)**



**bi)**

Yes, hard to explain



Is this enough?

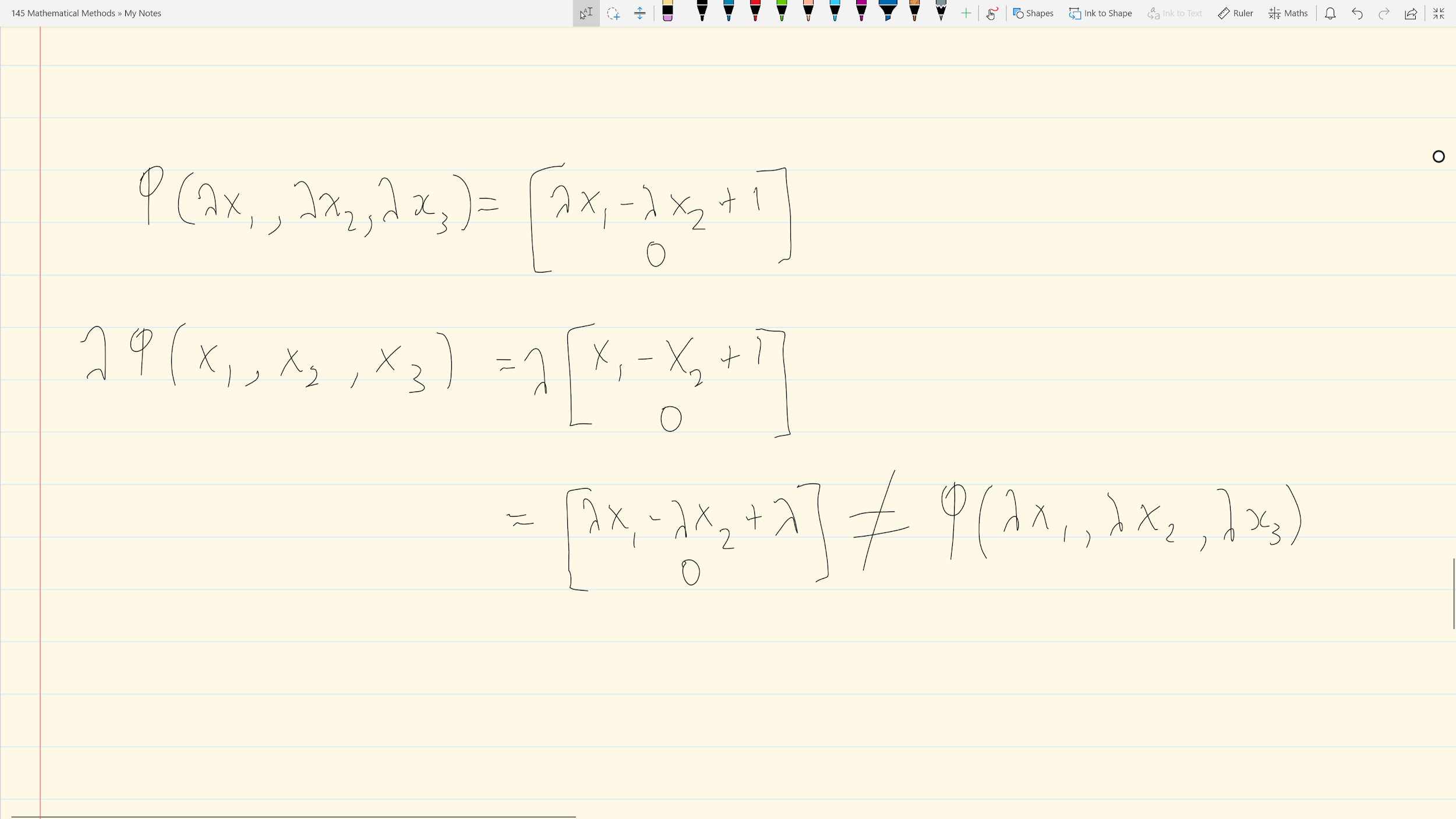
Looks like it yeah

**ii)**

No,hard to explain but let’s have a go

Something to do with either not working over addition, multiplication or some form of associativity

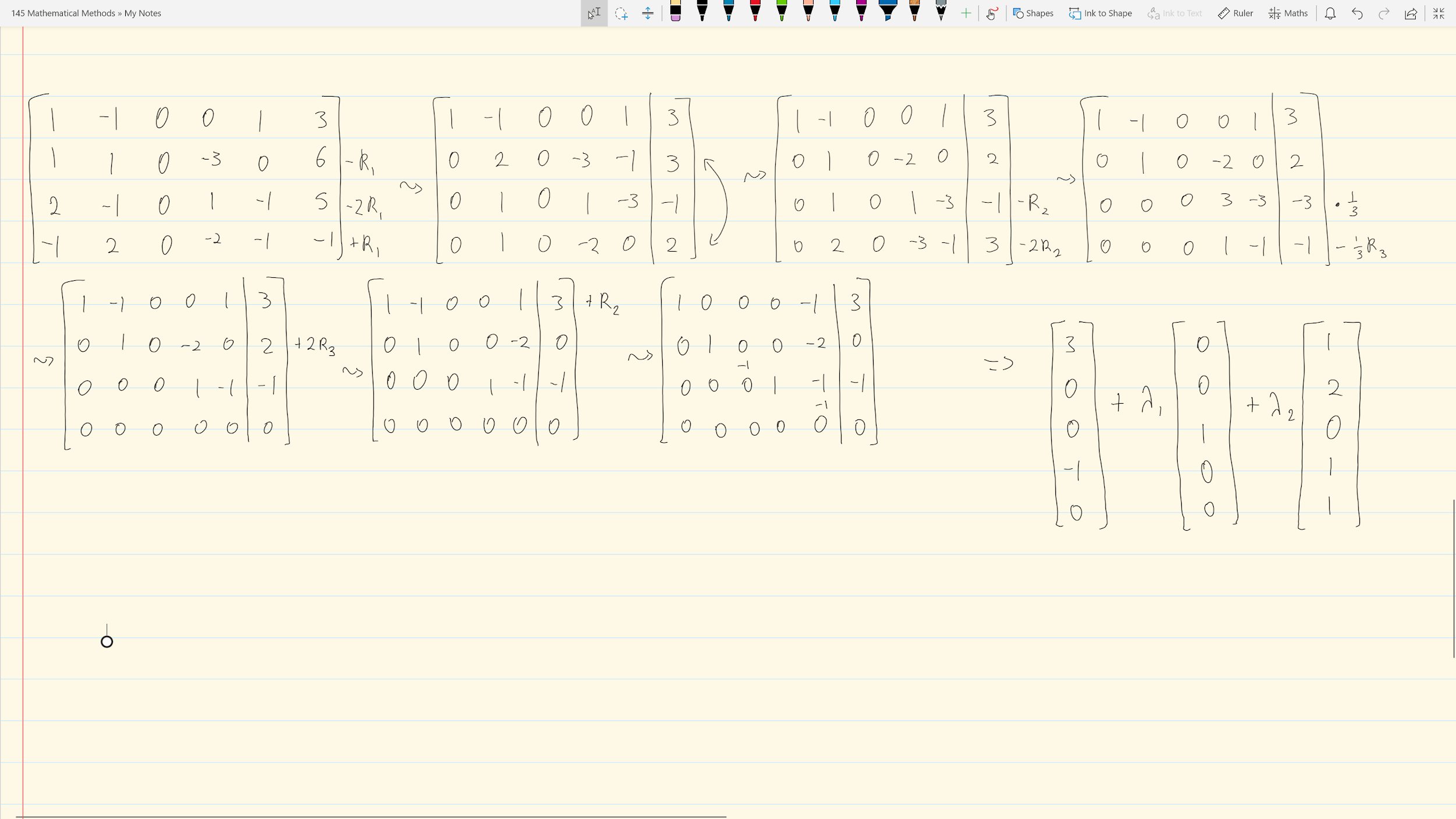
Eh



Is this enough?

Looks like it is

**c)**

+