Converting expressions into matrices

As you have probably seen in the course, it is often useful to convert expressions into matrix operations.  
  
E.g. instead of writing

we can equivalently define a matrix and and calculate

As you can already see above, a matrix representation of a given expression is generally not unique. This is not surprising, because even simple numbers have a multitude of ways to be expressed. For example,

That above equality is true can be easily proven by utilizing tools to solve geometric series. Any real number can trivially be represented by a matrix . In this case would even be multiplicative commutable! If we are only concerned with preserving additive operations, we have an even broader amount of suitable matrices available:

would also be isomorphic representations of concerning addition.  
This is often done in complex analysis, where we see as a pair belonging to . Of course this isomorphism is only valid if we do additive operations and stops if we compare both representations concerning e.g. differentiability (a function in representation might be real-but not complex differentiable). Perhaps surprisingly, there is a matrix representation of the complex numbers that really “behaves” like the complex numbers even regarding multiplication. See <https://math.stackexchange.com/questions/180849/why-is-the-complex-number-z-abi-equivalent-to-the-matrix-form-left-begins>)

But how can we find a matrix representation of a given expression? For example, what if we have the following expression:

We now want to convert its logarithmic derivative into matrix form:

I usually start with each term sequentially (including the summation):

This is almost a matrix expression. The only thing we have to somehow convert is the sum. If we introduce the design matrix this becomes:

As you see, the entire process is trial and error.

TO BE CONTINUED