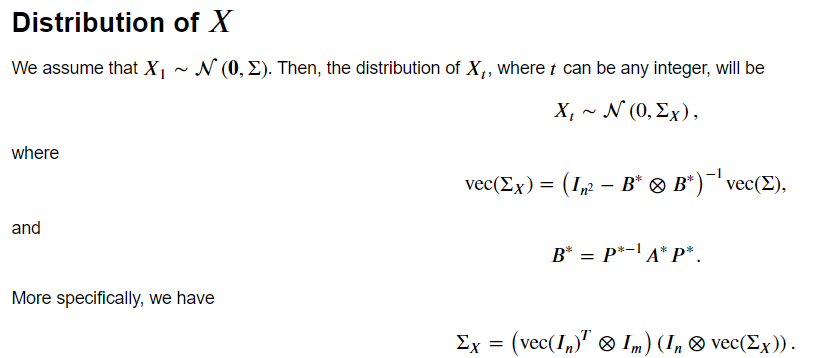
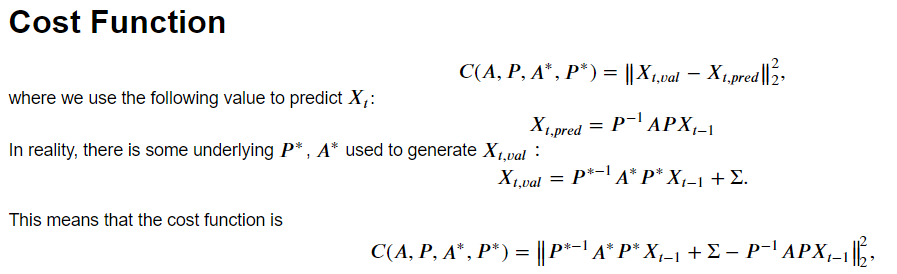
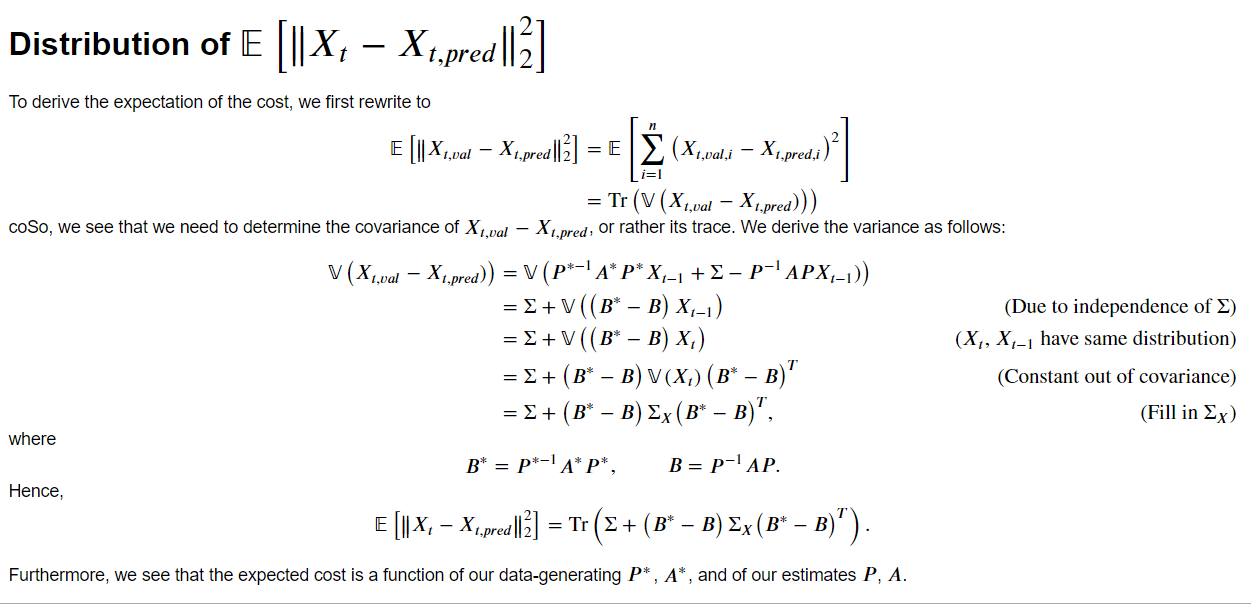
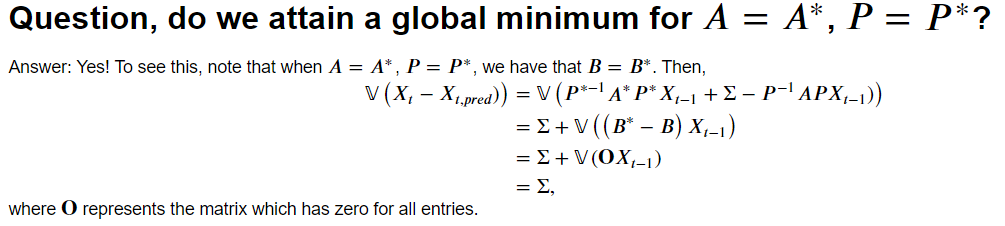
Prep Meeting 7

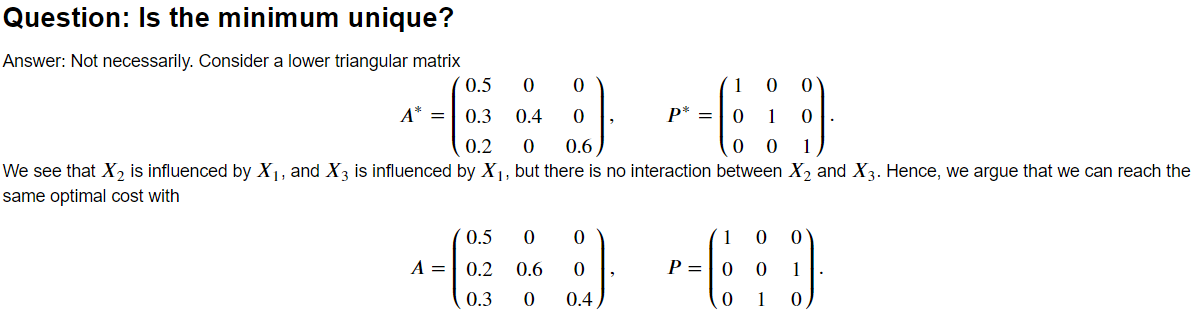
1. **Derivation of X\_t, Derivation of expected error.**











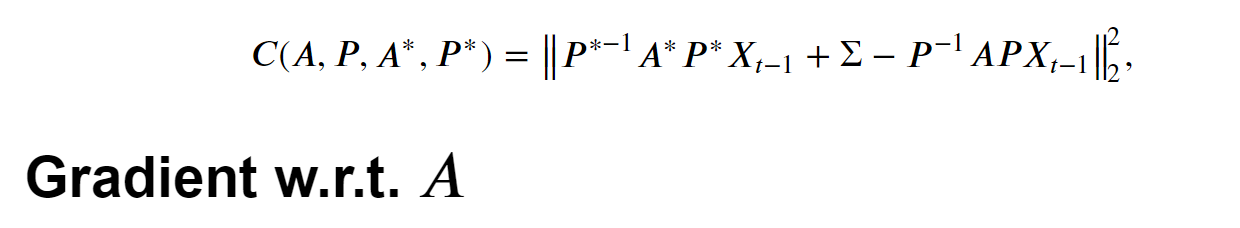
**Conclusion**: Using this format, we know that we can never attain a cost lower than Tr(Sigma). Furthermore, the difference

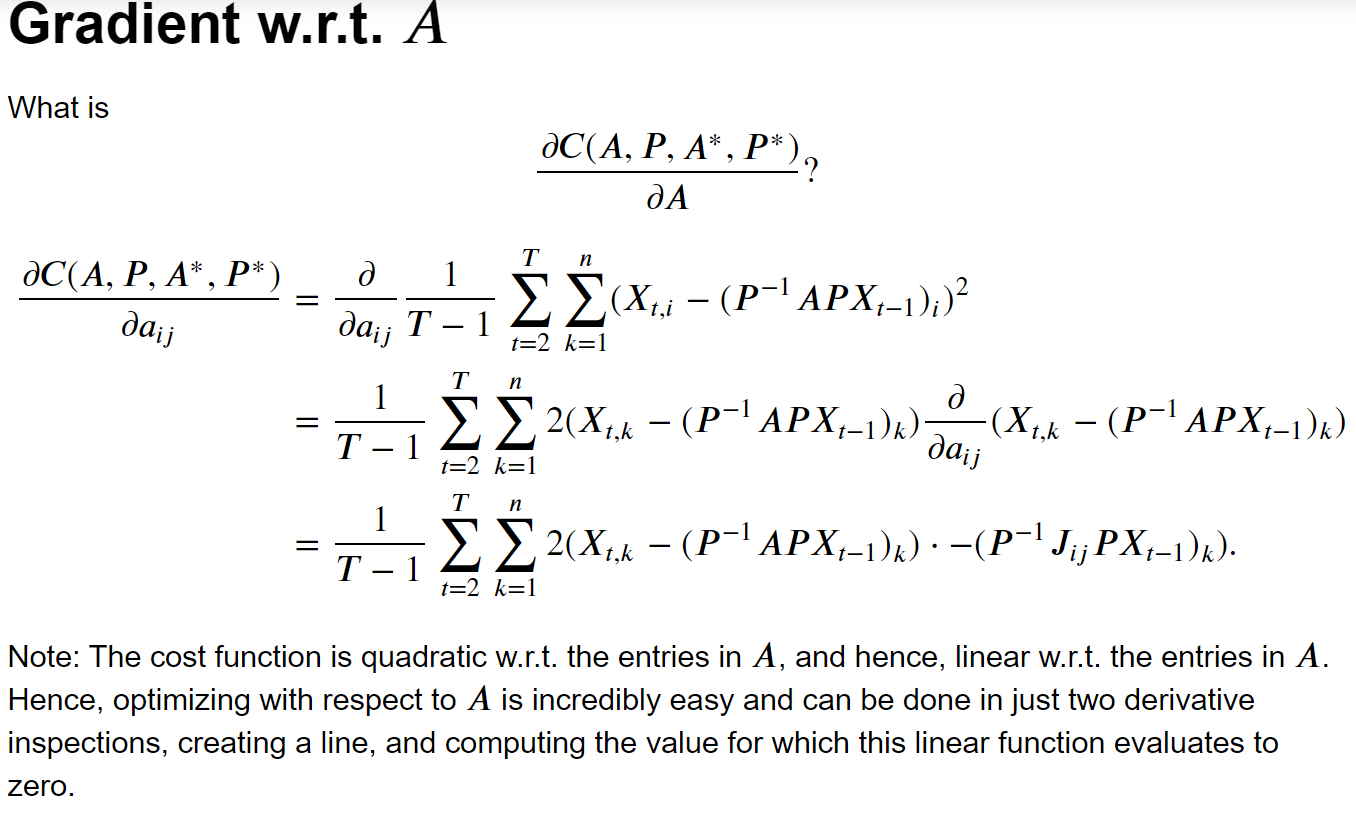
**NP-Hardness of our problem:**

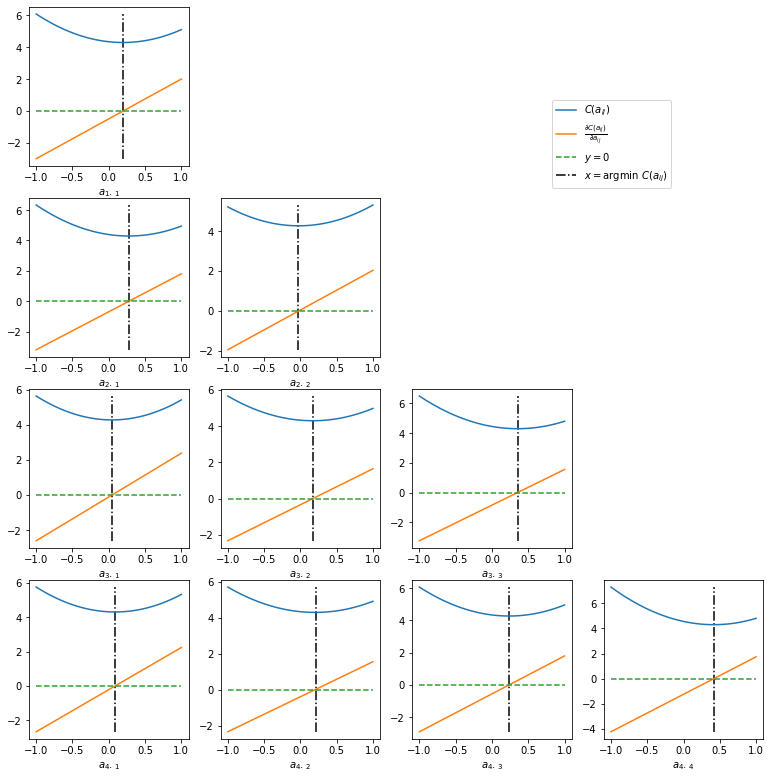
Chickering et al (2004): “*…, we therefore immediately conclude that the optimization problem of identifying the optimal DAG is hard as well.*” They prove this by reducing the feedback arc set problem to this problem. <https://www.jmlr.org/papers/volume5/chickering04a/chickering04a.pdf>

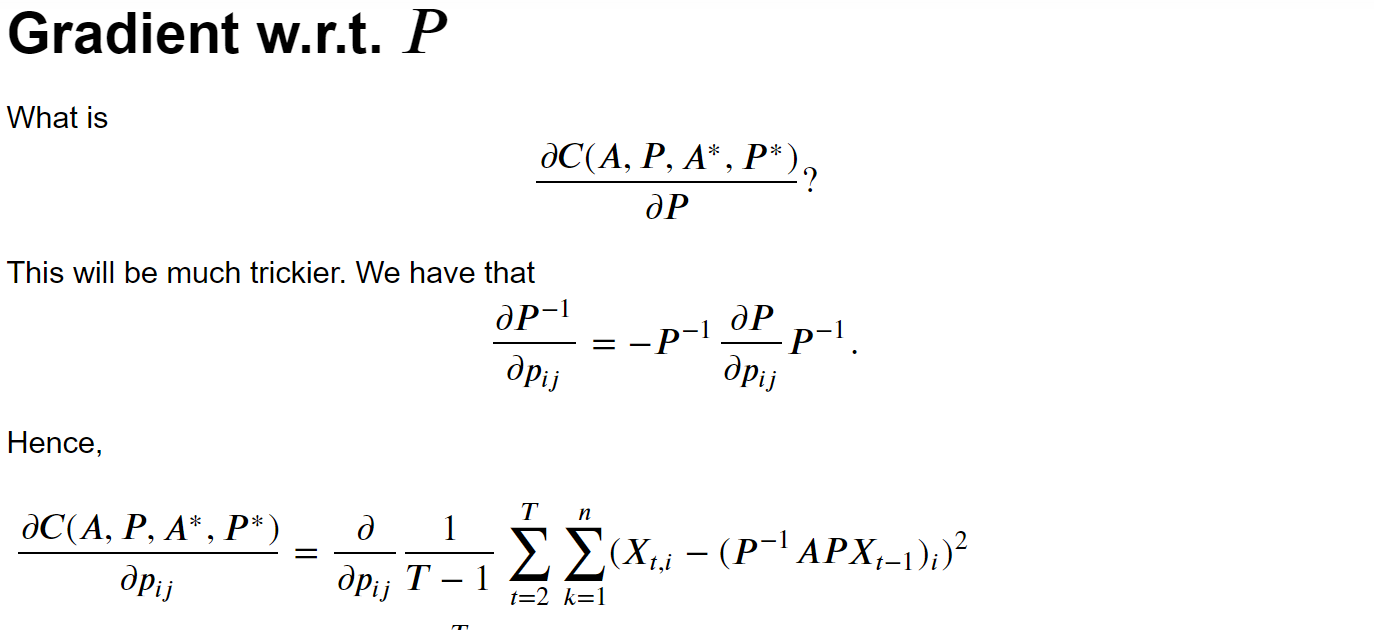
This also makes sense: Exact solution to this problem are all exponential and work until ~12 nodes. All the other methods are approximate solutions, who are only guaranteed to find a local optimum. However, the “useful” algorithms find a local optimum whose DAG is close to the global optimum.

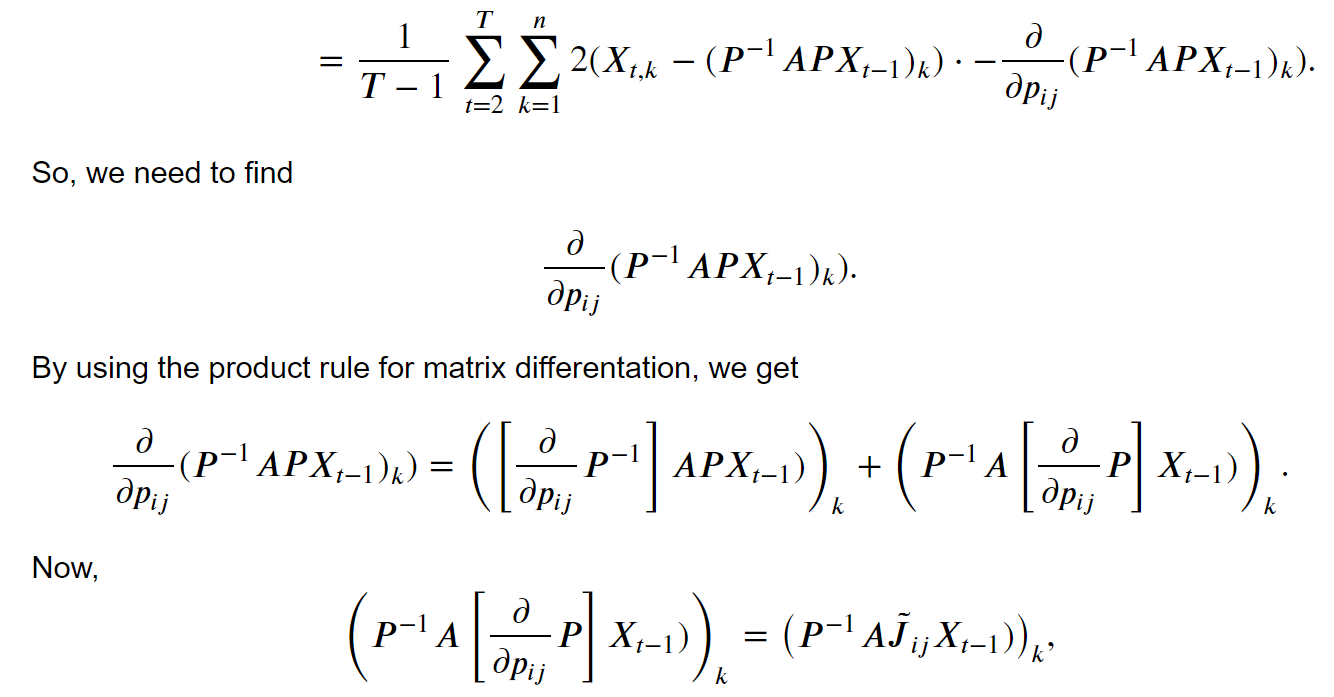
**Derivation of gradient**

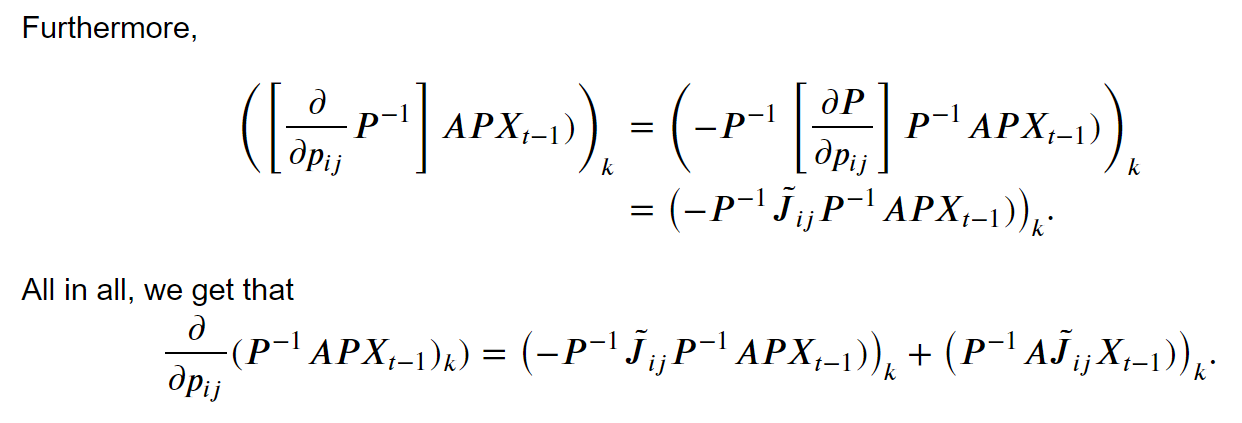


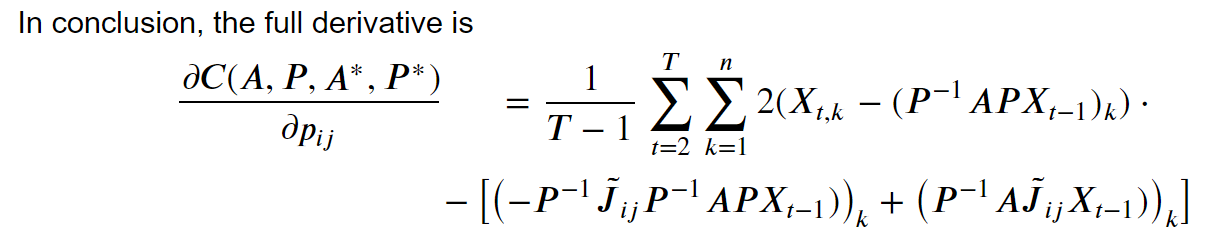


****

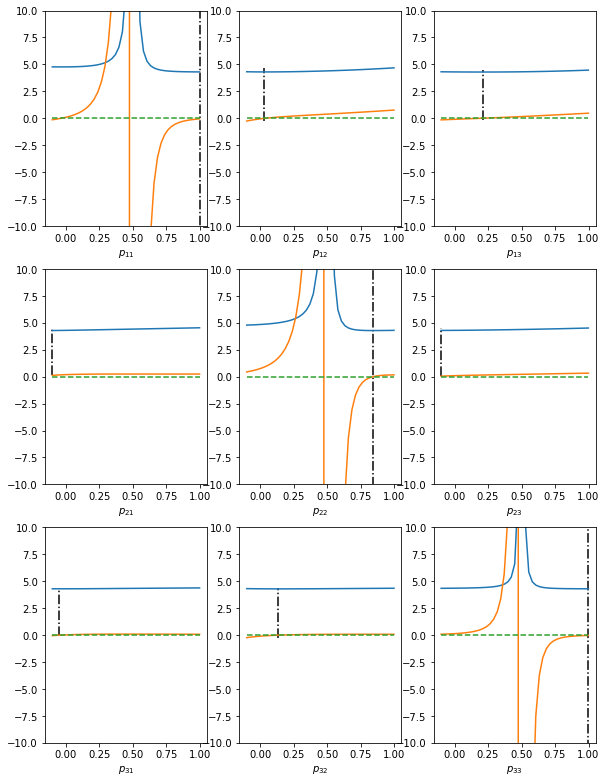
****

****

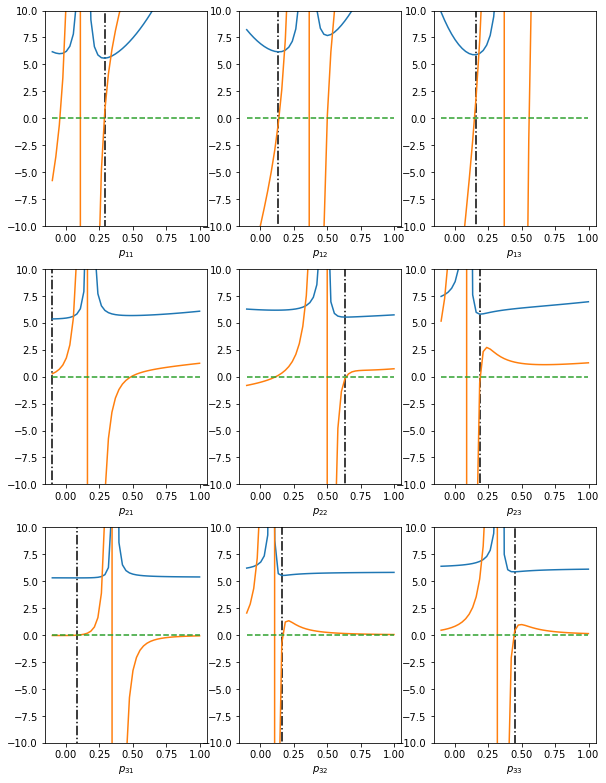
****

****

**Makes sense starting from the correct initialization (duh)**

****

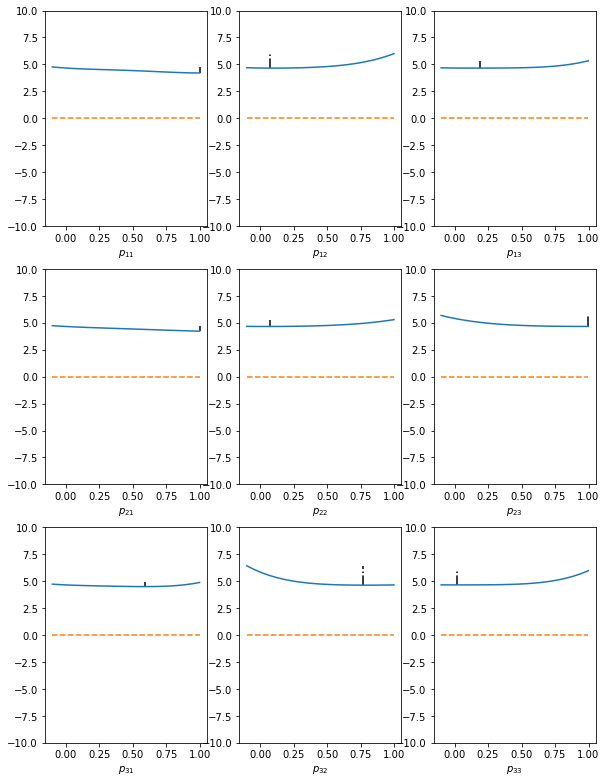
**Does not make sense from a random initialization**

****

So, given that the gradient of the P^{-1}AP is discontinuous and far from smooth, we seem to be unable to escape local optima.

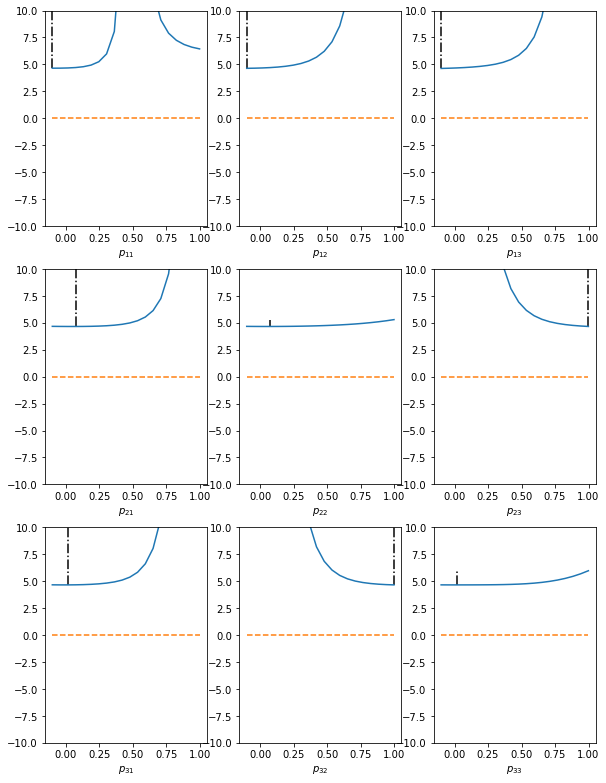
Other approaches:

* P^TAP -> Should make the derivative much nicer, as well as the cost function. However, are the results as we want?



Indeed, cost function is now polynomial, and I think the derivative will be third / second degree. However, we can clearly see that the minima do not lie where we expect them to be. We expect p\_ii to have a minimum at 1, and the others at 0. However, even when we start with the perfect P, even then e.g. p\_22 finds its optimum nowhere near 1, and some others find its optima nowhere near 0.

* P^TAP / detP -> Should resemble P^{-1}AP a little bit better, but dividing by det P might not be the best solution.



**Application to VAR models**

Seem more promising, e.g. for n = 3, we do not seem to get stuck in other local optima.

**Application to SEM models**

**X = AX + Σ**

Rather than X\_t = P^{-1}APX\_{t-1}, where A is (not strictly) lower triangular, I also investigated SEM.

X\_t = P^{-1}APX\_t, where A is not strictly lower triangular, as in the setting of some other papers, e.g. NOTEARS.

*Results*: It seems that there are a lot of local optima where the minimizer gets stuck in. When the starting point *x0* is close the actual global minimum, it will easily converge to that global minimum. However, when *x0* is off, e.g. *P0* resembles more a different permutation matrix, then it has a very hard time getting out of this local optimum.

Example: When P = I, starting with P0 = [[0.51, 0.49, 0], [0.49, 0.51, 0], [0, 0, 0]] always yields optimal results, as the closest permutation matrix is I. However, when starting with P0 = [[0.49, 0.51, 0], [0.51, 0.49, 0], [0, 0, 0]], then we will always get stuck in a local optimum [[0.5, 0.5, 0], [ 0.5, 0.45, 0.05], [0, 0.05, 0.95]]. Hence, we need to find a way to avoid / reduce the number of these local optima.

X\_1 = 2X\_2 + noise

X\_2 = noise

X\_3 = X1 + X2 + noise

Discussed the derivations, and all agreed. Closer to B\* the better. Define the setting clearer: Topological ordering should be unique.

TODO NEXT WEEK:

LOOK AT POPULATION SETTING

PERHAPS POPULATION SETTING IS NOT NP HARD

OTHER METHODS, SUCH AS PTAP / SOMETHING ELSE

Alex thinks NOTEARS is indeed a good idea, as acyclicity is a soft assignment here.