For Eq 10 in [Forde24] for the finite-option case, set

$$
\begin{aligned}
u(x) &= \sum_{i=0}^{n} w_i^X (x - K_i^X)^+ \\
v(y) &= \sum_{i=0}^{n} w_i^Y (y - K_i^Y)^+ \\
w(z) &= \sum_{i=0}^{n} w_i^Z (z - K_i^Z)^+
\end{aligned}
$$

where $n = 5$ and $K_0^X = K_0^X = K_0^X = 0$ (which correspond to the forward contracts which pay X, Y and Z respectively); we then maximize over the 3*6=18 weights to get e^{u+v+yw}. Also need to normalize e^{u+v+yw} as in Eq 11 in [F24]. Then if done correctly, $\mu^*(x,y) = \frac{e^{u(x)+v(y)+yw(y/x)}}{\mathbb{E}(e^{u(X)+v(Y)+Yw(X/Y)})}\mu_X(x)\mu_Y(y)$ will correctly price all three smiles (note this requires 2d Gauss-Legendre quadrature as in the final part of Part 2). This approach does not require Sinkhorn iterations or the SVI formula or any interpolation/extrapolation with splines, but you may need to run the `scipy` minimizer for a few thousand iterations to get good results.

You can also try doing this using the **MOSEK** solver as in Project 3 (which will likely be quicker since MOSEK uses **interior point methods** which are specifically designed for convex optimization problems). You can also look at the **forward-starter** option problem using this approach (see **chap 5** in FM14 2024), in which case you will need to maximize over an additional N parameters to enforce the **martingale condition** (where N is your Number of quadrature weights), for which MOSEK would likely be much faster.

Also remember the SVI formula is not a model, it is just a parametrization for implied volatility at a single maturity. For Part 1, you should also try to discuss no-arbitrage conditions for SVI, specifically the meaning of **butterfly arbitrage** which arises if the SVI density comes out negative.