Optimisation of the L-neighbour approach for a QD-cavity system

In the LN approach

$$F^{(s+1)}_{i_L \dots i_1} = \sum_{e=0,1} G_{i_L \dots i_1 e} F^{(s)}_{i_{L-1} \dots i_1 e}$$

with the initial $F^{(1)}_{i_L \dots i_1} = M_{i_1 i_0}$

and

$$G_{i_L \dots i_1 e} = M_{i_1 e} e^{\delta_{e1}(K_0 + 2\delta_{i_1 1} K_1 + \dots 2\delta_{i_L 1} K_L)}$$

and the polarisation has the form

$$P_{i_N i_0}(t) = e^{\delta_{i_N 1} K_0} F^{(N)}_{0 \dots 0 i_N}$$

Here $i=0$ $(i=1)$ corresponds to the cavity (exciton) state.

Let us first introduce at each iteration s a normal matrix array $\mathcal{F}$:

$$F^{(s)}_{i_L \dots i_1} = \mathcal{F}_{nm}$$

This is not necessarily the case and will not be the case for some iterations, but let us ~~take~~ choose for clarity n and m such that

in n-array the values of i_1 to $i_{L/2}$ do not change

in m-array the values of $i_{\frac{L}{2}+1}$ to i_L do not change

For example,

$$m = 2^{\frac{L}{2}-1} i_{L/2} + 2^{\frac{L}{2}-2} i_{\frac{L}{2}-1} + \dots + 2^0 i_1$$

so m depends on $i_{L/2} \dots i_1$ only

$$n = 2^{\frac{L}{2}-1} i_L + 2^{\frac{L}{2}-2} i_{L-1} + \dots + 2^0 i_{\frac{L}{2}+1}$$

so n depends on $i_L \dots i_{\frac{L}{2}+1}$ only

At this point one has to introduce arrays of i_p, with $p = 1, \dots L$

Assuming for simplicity the ~~siz~~ n- and m-arrays are of the same size $\bar{N}$, these can be 2D arrays

$$i_p[0 \dots 1][0 \dots \bar{N}] \quad \text{with } \bar{N} = 2^{L/2} - 1 \ (L \text{ even})$$

where the first index says whether i_p is belongs to n or m and the second is n or m itself. For example if i_p belongs to n-array, then

$i_p[0][n] = 0$ or 1 (depending on n)

$i_p[1][m] = 2$ (or any other number)

the latter indicating that m-array is independent of i_p. See also an example in the Appendix.

Let us now rewrite the recursive formulas as

$$F^{(s+1)}_{p\, i_L \dots i_2} = \sum_e Q^{(L)}_{pe}\, Q^{(L-1)}_{i_L e} \dots Q^{(2)}_{i_3 e}\, Q^{(1)}_{i_2 e}\, F^{(s)}_{i_L \dots i_2 e}$$

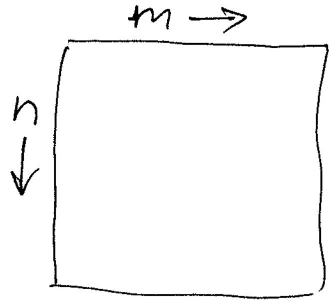
where $\ell = i_1$ and p take the values of 0 or 1 and

$$Q^{(r)}_{i\ell} = e^{\delta_{\ell 1}\delta_{i1} 2K_r} \qquad (1 < r \leq L)$$

$$Q^{(1)}_{i\ell} = M_{i\ell}\, e^{\delta_{\ell 1} K_0}\, e^{\delta_{\ell 1}\delta_{i1} 2K_1} \qquad (r=1)$$

To implement this, we take the SVD of normal matrix

$\mathcal{F}_{nm} =$

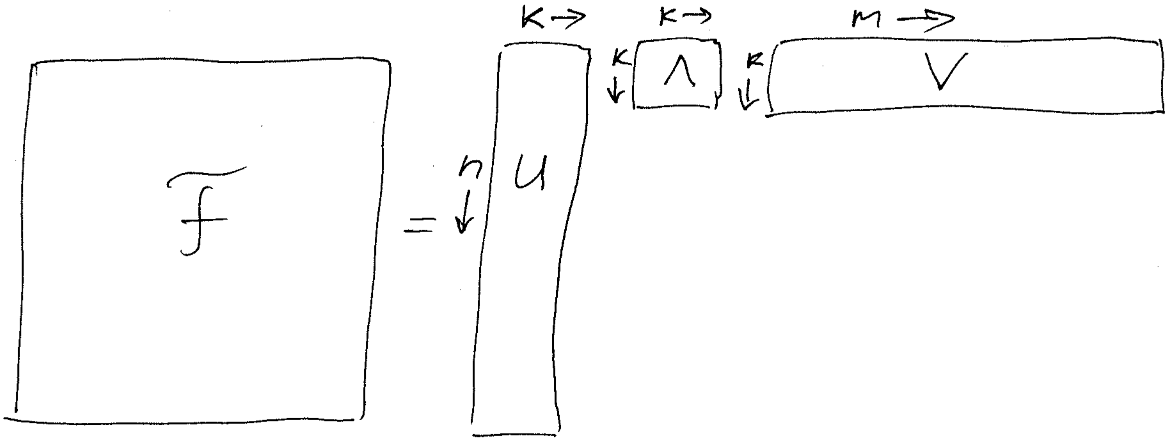


$i_1 - i_{L/2}$ not changing in n

$i_{\frac{L}{2}+1} - i_L$ not changing in m

representing $F^{(s)}_{i_L \cdots i_1}$, which can be visualised as

$\mathcal{F} =$



and written as

$$\mathcal{F}_{nm} = \sum_K U_{nK} \Lambda_K V_{Km}$$

Let us separate $i_1 = 0$ and $i_1 = 1$ elements of $\mathcal{F}$:

$$\mathcal{F}^{(0)}_{nm'} = \sum_K U_{nK} \Lambda_K V^{(0)}_{Km'}$$

$$\mathcal{F}^{(1)}_{nm'} = \sum_K U_{nK} \Lambda_K V^{(1)}_{Km'}$$

so $F_{nm} = F^{(0)}_{nm'}$ if $i_1 = 0$

and $F_{nm} = F^{(1)}_{nm'}$ if $i_1 = 1$

since n-array does not depend on i_1

U_{nk} is the same in both cases

but $V^{(0)}_{km'} = V_{km}$ if $i_1 = 0$

and $V^{(1)}_{km'} = V_{km}$ if $i_1 = 1$

Clearly m'-array is 2 times shorter than m-array as it is independent of i_1 (but m-array depends on i_1)

Then multiply $\underbrace{U_{nk}}_{\text{with } Qs}$ to obtain two new matrices:

$$\tilde{U}^{(0)}_{nk} = Q^{(L-1)}_{i_L 0} \cdots Q^{(L/2)}_{i_{L/2+1} 0} U_{nk}$$

$$\tilde{U}^{(1)}_{nk} = Q^{(L-1)}_{i_L 1} \cdots Q^{(L/2)}_{i_{L/2+1} 1} U_{nk}$$

again corresponding to $i_1 = 0$ and $i_1 = 1$, respectively. Similarly,

$$\tilde{V}^{(0,p)}_{km'} = Q^{(L)}_{p0}\, Q^{(\frac{L}{2}-1)}_{i_{L/2} 0} \cdots Q^{(1)}_{i_2 0} V^{(0)}_{km'}$$

$$\tilde{V}^{(1,p)}_{km'} = Q^{(L)}_{p1}\, Q^{(\frac{L}{2}-1)}_{i_{L/2} 1} \cdots Q^{(1)}_{i_2 1} V^{(1)}_{km'}$$

with $p = 0$ or 1, introducing a new index, due to G.

Now a new $\tilde{\mathcal{F}}^{(p)}_{nm'}$ representing $F^{(s+1)}_{p i_L \dots i_2}$ takes the form

$$\tilde{\mathcal{F}}^{(p)}_{nm'} = \sum_k \left(\tilde{U}^{(0)}_{nk} \Lambda_k \tilde{V}^{(0,p)}_{km'} + \tilde{U}^{(1)}_{nk} \Lambda_k \tilde{V}^{(1,p)}_{km'} \right)$$

We can introduce a new matrix $\mathcal{F}_{nm}$ on step $s+1$ as a product of matrices:

$$\mathcal{F} = \begin{bmatrix} \tilde{U}^{(0)} & \tilde{U}^{(1)} \end{bmatrix} \begin{bmatrix} \Lambda & \\ & \Lambda \end{bmatrix} \begin{bmatrix} \tilde{V}^{(0,0)} & \tilde{V}^{(0,1)} \\ \tilde{V}^{(1,0)} & \tilde{V}^{(1,1)} \end{bmatrix}$$

Clearly the m-array is now restored to accomodate the new index p. At this stage all the arrays ip[ ][ ] should be redefined, in order to go from

$$F^{(s+1)}_{p i_L \dots i_2} \to F^{(s+1)}_{i_L \dots i_2 i_1}$$

The new $\mathcal{F}$ has the form of SVD with a double size of the diagonal matrix and can be optimised by applying SVD to different parts of $\mathcal{F}$.

For instance, one can SVD the matrix

$$\begin{matrix} \Lambda & & \tilde{V}^{(0,0)} & \tilde{V}^{(0,1)} \\ & \Lambda & \tilde{V}^{(1,0)} & \tilde{V}^{(1,1)} \end{matrix} = A \; \Lambda_1 \; V_1$$

where Λ_1 has a smaller size due to the threshold condition that

if $|\lambda_k| < \varepsilon$ then $\lambda_k = 0$

Next, one can apply an SVD to

$$\tilde{U}^{(0)} \; \tilde{U}^{(1)} \; A \; \Lambda_1 = U_2 \; \Lambda_2 \cdot B$$

This procedure can be repeated several times until matrices A and B are reduced to their minimal size. After that we can go from $F^{(s+1)}$ to $F^{(s+2)}$
Clearly, at some point (for some s) the index $i_1 = e$ will be found in the n-array. In this case one should do with matrix U the same what was done with V, and vice versa.
Arrays ip[][] should be reorganised ~~for~~ each s.

The SVD of the initial tensor $(s=1)$ is analytical:

$$F^{(1)}_{i_L \dots i_1} = M_{i_1 i_0} = \mathcal{F}_{nm}$$

where i_0 is fixed. Then

$$\mathcal{F}_{nm} = \sum_{k} U_{nk} \Lambda_k V_{km}$$

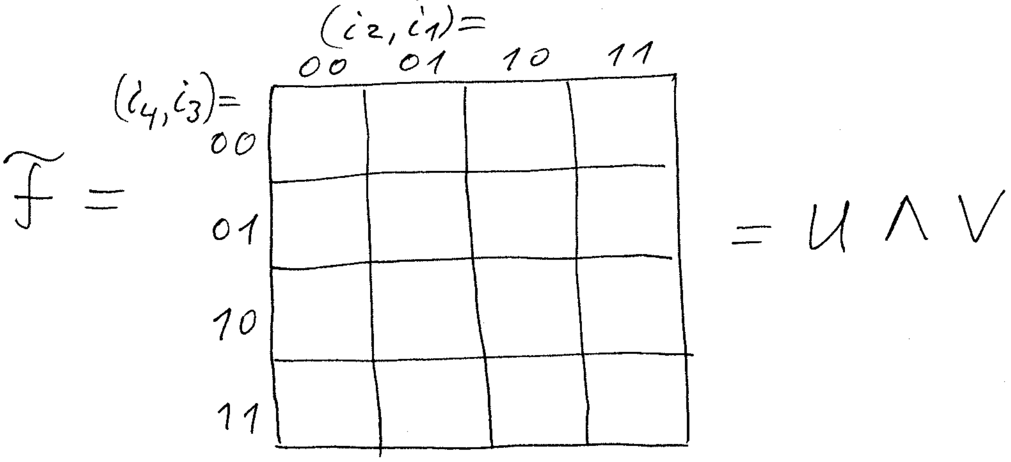
with

$$U_{n0} = 1 \qquad \Lambda_k = \delta_{k0} \qquad V_{0m} = M_{i_1 i_0}$$

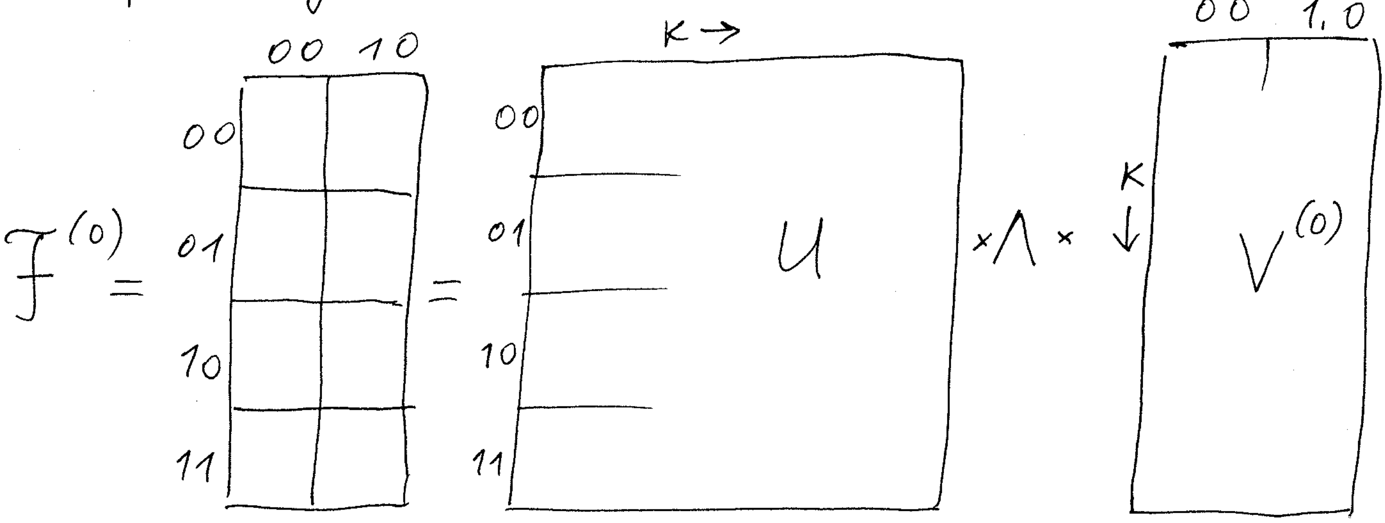
so Λ is a 1×1 matrix.

Appendix: Example of $F_{i_4 i_3 i_2 i_1}$ tensor

$m = 2i_2 + i_1 \qquad n = 2i_4 + i_3$



Separating $i_1 = 0$ elements, obtain:



Here it is assumed Λ has the maximum size (4×4 in this case).
However, for the whole procedure to work one probably has to require that Λ is 4 times smaller than its maximum size (i.e. 2×2 in this case.)
Or maybe this is not needed because of the SVD applied, see page 6.