Calculation of derivative properties in XMS-CASPT2

Peter John Cherry
Shiozaki group meeting
November 28th 2017

Calculation of matrix elements for many electron operators

- We need a generic approach for evaluating terms of the form

$$\sum_{\substack{ijkl\\wxyz}} \sum_{J} T^{\dagger}_{ijkl} g_{wxyz} q_{uv} \langle I | a_i a_j a^{\dagger}_k a^{\dagger}_l a^{\dagger}_w a^{\dagger}_x a_y a_z a^{\dagger}_u a_v | J \rangle c_J$$

- These terms can be decomposed into a sum of expressions of the following form:

$$\sum_{ijklmn}^{act} \sum_{J} \langle I | a^{\dagger}_i a^{\dagger}_j a^{\dagger}_k a_l a_m a_n | J \rangle c_J A_{ijklmn}$$

- Jae used code from SMITH3 to develop an efficient method of doing this in a non-relativistic framework.

- However, for cases involving spin-flipping interactions, we need to develop a method which takes advantage of spin symmetry.

Outline of method

Step 1 : Get operator information.

- Split input tensors up into blocks and identify symmetries between blocks.

Step 2 : Simplify expression

- Use expression manipulator to reduce the number of terms to be calculated.

← Changes here

Step 3 : Generate computational proceedure.

- Generate a task list to obtain contracted A-tensors and corresponding RDM/RDM derivatives.

← Changes here

Step 4 : Execute computational proceedure.

- Obtain A-tensors/RDM matrices and contract.

← Changes here

Step 5 : Loop over spin sectors and states.

- Repeat for all necessary spin sectors.

Outline of method

Step 1 : Get operator information.
- Split input tensors up into blocks and identify symmetries between blocks.

Step 2 : Simplify expression
- Use expression manipulator to reduce the number of terms to be calculated.

← Working*

Step 3 : Generate computational proceedure.
- Generate a task list to obtain contracted A-tensors and corresponding RDM/RDM derivatives.

← Working*

Step 4 : Execute computational proceedure.
- Obtain A-tensors/RDM matrices and contract.

← Working*

Step 5 : Loop over spin sectors and states.
- Repeat for all necessary spin sectors.

← Needs retesting

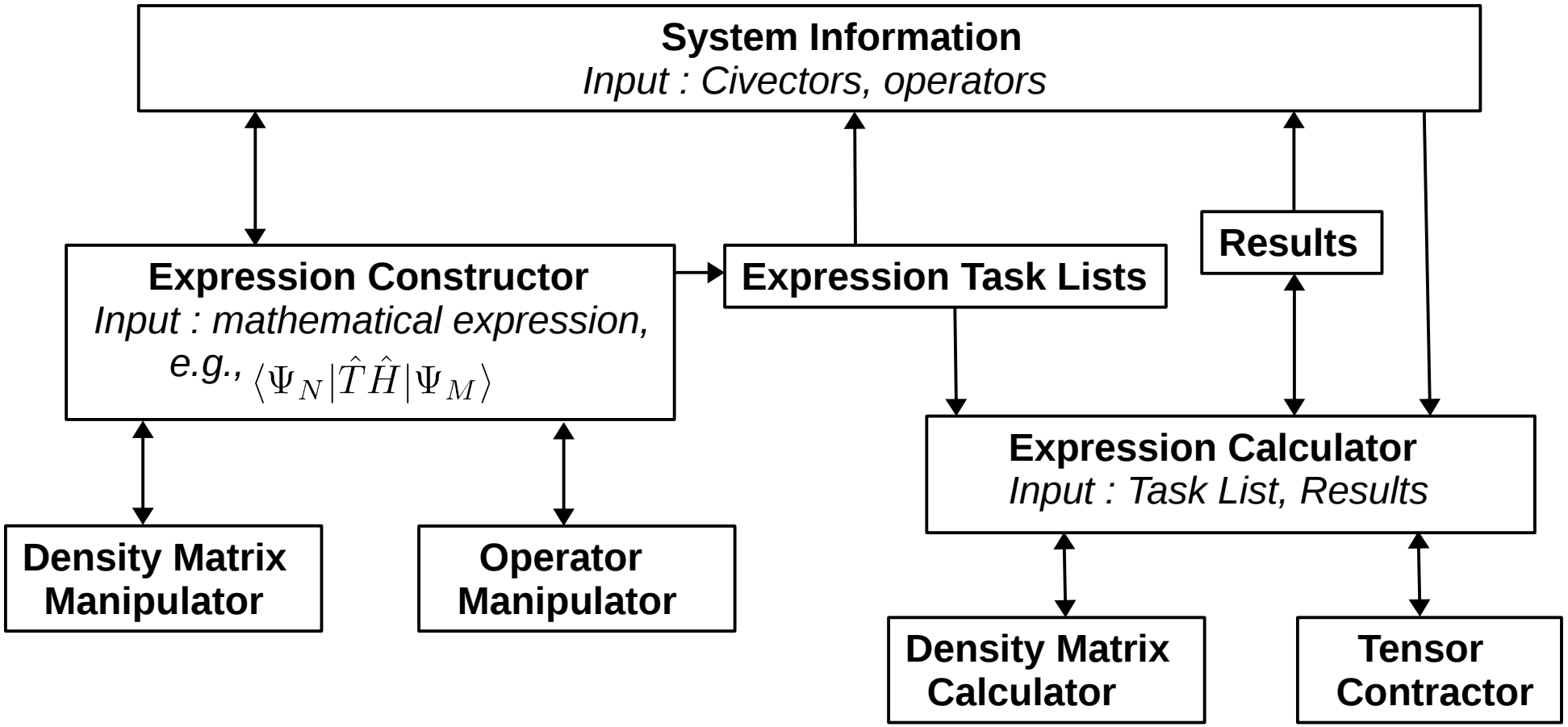
Step 6 : Implement CI derivative expression

← Doing now

Possible use of SMITH3

- Project has been progressing slowly; should reconsider direction.
- With some modification, existing algorithms in SMITH3 could probably be used, however...
    - Modification could be significant, or at least would require substantial effort to verify I had done it correctly.
    - Not so easily extensible to other properties.
    - Likely to be more computationally expensive due to lack of incorporation of symmetry
        - ➔ Would like to implement MRCI
    - Current approach is starting to work better, and will hopefully be more productive in the longer term.

Program Structure



Advantage for other property calculations

- Having overall system info class simplifies efficient reuse of previously calculated quantities.
- Not specific to XMS-CASPT2; generic format, e.g., can be used for MRCI.
- For most properties, the additional work should only be in the programming of the integrals.
- Should be possible to extend frame work to include solution of basic equations (e.g., the linear solver currently used to obtain the T-amplitudes).
- Ultimate aim is to have the option to enter mathematical expressions for properties in the input file
  - ➔ Not totally unreasonable as the expressions for many different properties use the same underlying operators.

Why MRCI and not XMS-CASPT2?

- Concern about the use of state-averaging in XMS-CASPT2.

Significance of state averaging in relativistic XMS-CASPT2

- XMS-CASPT2 uses a state averaged density

$$\hat{f}^{sa} = \hat{\tilde{f}}(\mathbf{d}^{(sa)}) = \hat{\tilde{f}}\left(\frac{1}{N}\sum_{n}^{N_{states}} \mathbf{d}_n\right)$$

Possible influence of state averaging on T-amplitudes

- XMS-CASPT2 wavefunction is given by:

$$|\Psi_L^{(1)}\rangle = \sum_N \hat{T}_{LN}|\tilde{\Psi}_N\rangle = \sum_N \sum_{abcd} T_{LN,abcd}\hat{E}_{abcd}|\tilde{\Psi}_N\rangle$$

- The $T_{LN,abcd}$ (T-amplitudes) satisfy the equation :

$$\sum_N \langle\tilde{\Psi}_M|(\hat{E}^\dagger_{ijkl}(\hat{f}^{(sa)} - \epsilon_L^{(0)} + \epsilon_{shift})\hat{T}_{LN}|\tilde{\Psi}_N\rangle + \langle\tilde{\Psi}_M|\hat{E}^\dagger_{ijkl}\hat{H}|\tilde{\Psi}_L\rangle = 0$$

- Where

$$\sum_M \langle\Psi_L|\hat{f}^{sa}|\Psi_M\rangle U_{MN} = U_{LN}\tilde{E}_N \quad \text{and} \quad |\tilde{\Psi}_L\rangle = \sum_M U_{LM}|\Psi_M\rangle$$

- State averaging plays an important role in XMS-CASPT2.

Significance of state averaging in relativistic XMS-CASPT2

- XMS-CASPT2 uses a state averaged density

$$\hat{f}^{sa} = \hat{\tilde{f}}(\mathbf{d}^{(sa)}) = \hat{\tilde{f}}\left(\frac{1}{N}\sum_{n}^{N_{states}} \mathbf{d}_n\right)$$

- The state averaging impacts two electron terms:

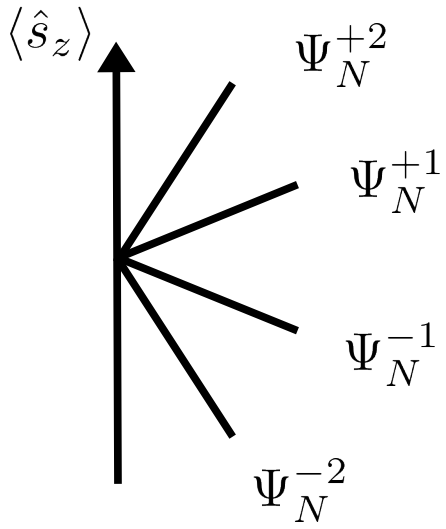
$$\hat{f}^{sa} = \hat{f}^{(1el)} + \hat{\tilde{f}}^{(2el)}(\mathbf{d}^{sa})$$

- The charge densities for different states are typically quite similar; they cannot differ by more than the number of electrons in the active space.

- States within the same spin multiplet have the same charge density.

Significance of state averaging in relativistic XMS-CASPT2

- The spin densities are typically very different for all states in a multiplet :



Charge density :

$$\rho_N(\mathbf{r}) = \Psi_N^{\dagger}(\mathbf{r})\Psi_N(\mathbf{r})$$

Spin density :

$$\boldsymbol{\sigma}_N(\mathbf{r}) = \Psi_N^{\dagger}(\mathbf{r})\hat{\mathbf{s}}\Psi_N(\mathbf{r})$$

- Can split Fock-operator into spin-dependent and spin-independent terms :

$$\hat{f}^{sa} = \hat{h}^{(1el)} + \hat{\tilde{g}}^{(2el)}_{nonrel}(\rho_{sa}) + \hat{\tilde{g}}^{(2el)}_{rel}(\rho_{sa}, \boldsymbol{\sigma}_{sa})$$

- It is not clear (to me) what the influence of the state averaging will be on the spin dependent terms.

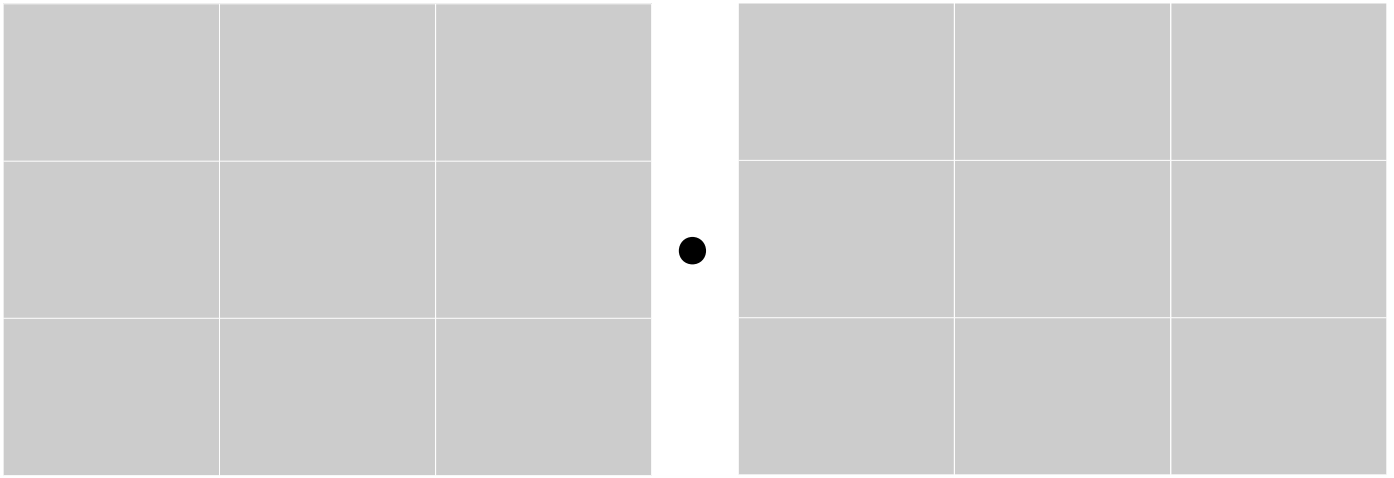
Why MRCI and not XMS-CASPT2?

- Concern about the use of state-averaging in XMS-CASPT2.
- Methods such as Douglas-Kroll-Hess are often very good for 1-electron relativistic effects, but can struggle with spin-spin and spin-other-orbit interactions.
- Accurate spin-spin interactions are important for justifying the large computational cost of four-component methods.
- However, memory demands of four component MRCI are large.
- Storing T-amplitudes is expensive enough to cause problems; storing rdm-derivatives could be a major issue.

Reducing memory cost

- Currently calculate the complete A-tensor, then the complete reduced density matrix and then perform the contraction :

$$\sum_{ijklmn} A^{LN}_{abcdef}\Gamma^{LN}_{abcdef} = \langle \Psi_L | \hat{T}^{\dagger}_{LN} \hat{H} | \Psi_N \rangle$$



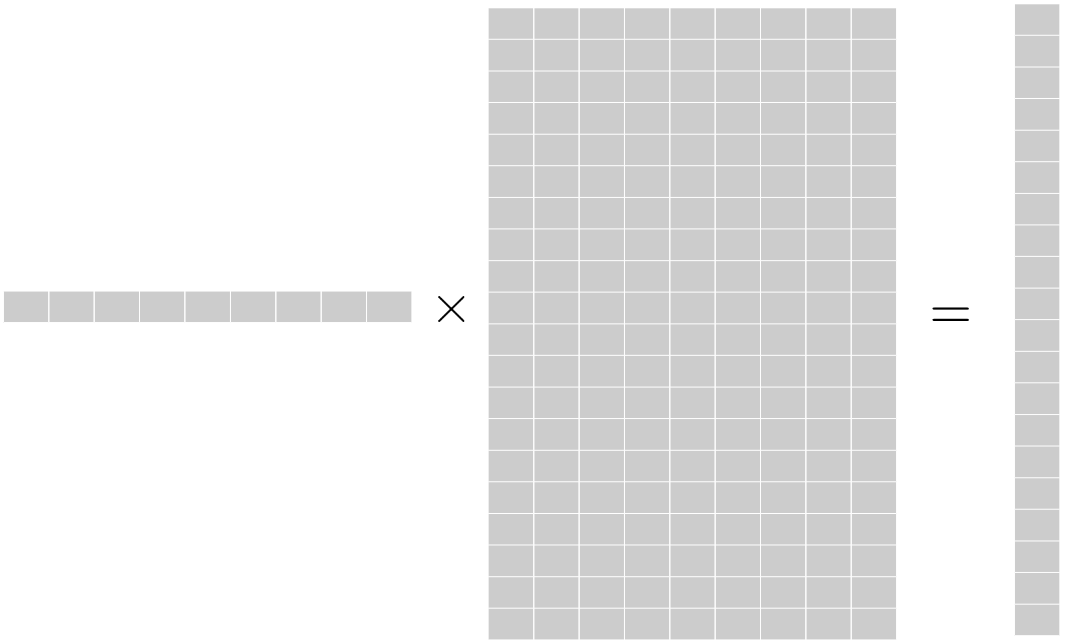
$[A^{LN}]$ $[\Gamma^{LN}]$ $= X^{LN}$

- *Must store both the density matrix and the A-tensor.*

Reducing memory cost

- Currently calculate the complete A-tensor, then the complete reduced density matrix derivative and then perform the contraction :

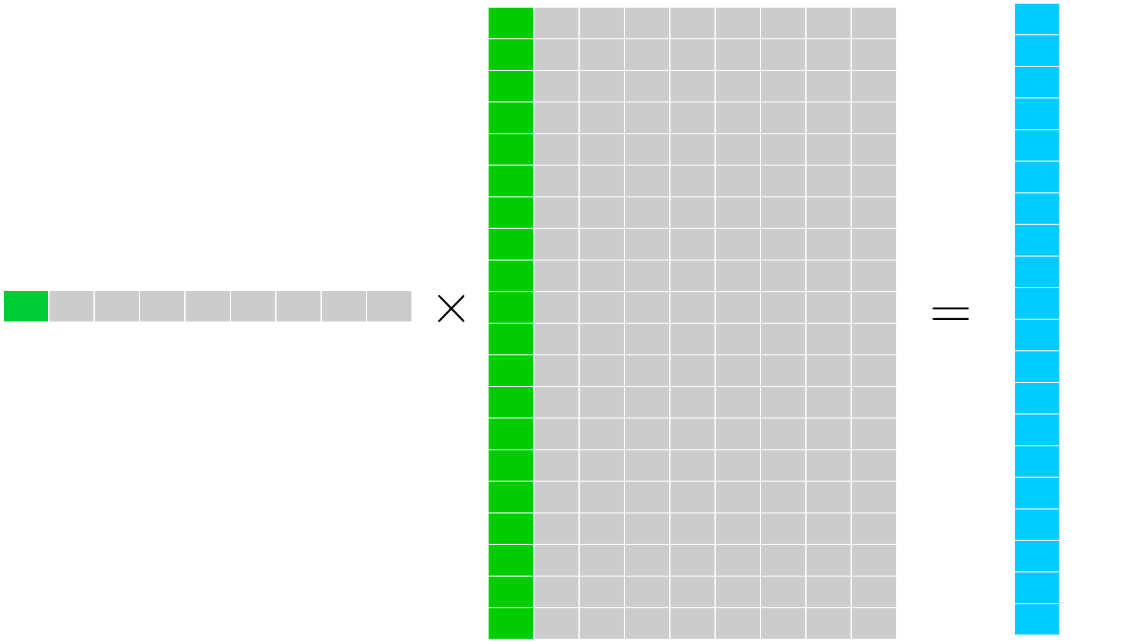
$$\sum_{ijklmn} A^{LN}_{abcdef}\Gamma^{LN,I}_{abcdef} = \langle\Psi_N|\hat{T}^{\dagger}_{LN}\hat{H}|I\rangle$$



$[A^{LN}]$ $[\Gamma^{LN,I}]$ $[\tilde{X}^{LN,I}]$

Blockwise contractions

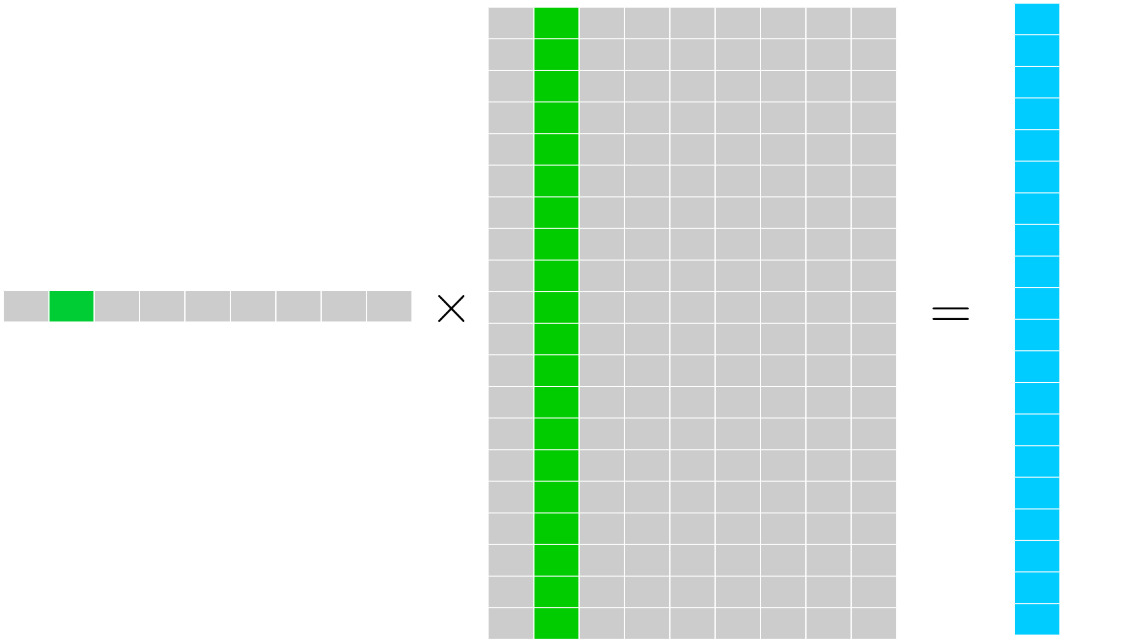
- Can instead calculate one block at a time, and contract immediately, avoiding the need to store the entire tensor :



- *Note : For this to be practical, must have already decomposed A-tensor into different spin sectors.*

Blockwise contractions

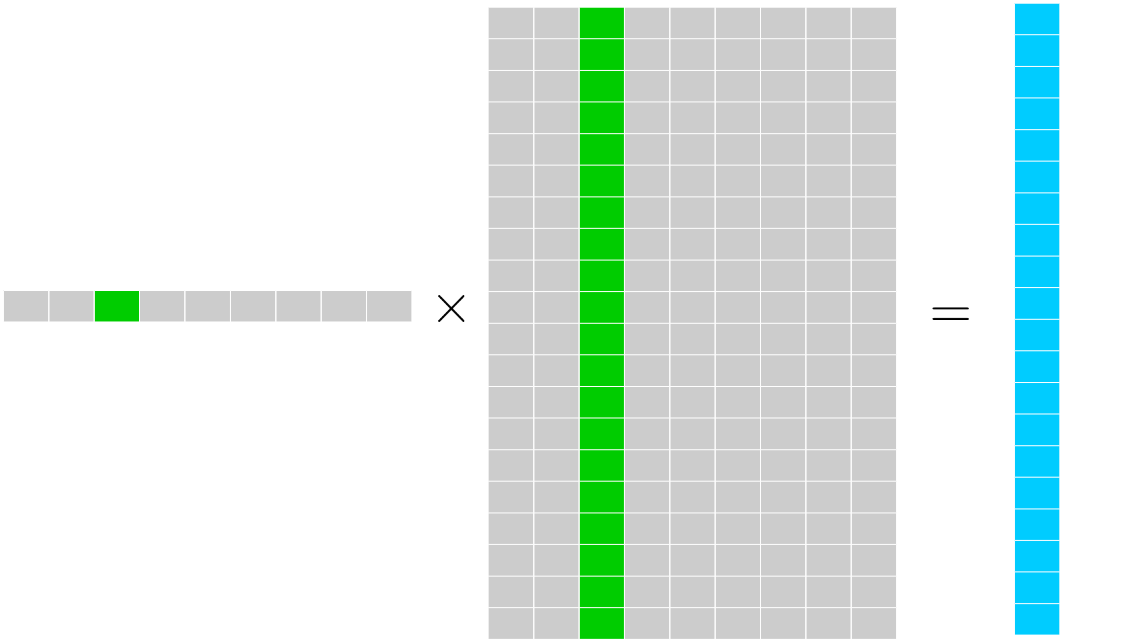
- Can instead calculate one block at a time, and contract immediately, avoiding the need to store the entire tensor :



- *Note : For this to be practical, must have already decomposed A-tensor into different spin sectors.*

Blockwise contractions

- Can instead calculate one block at a time, and contract immediately, avoiding the need to store the entire tensor :



- *Note : For this to be practical, must have already decomposed A-tensor into different spin sectors.*

Summary

- Little change in what I am trying to do, however, parts of it are now working.
- No longer programming underlying machinery, but using the machinery to implement the equations for XMS-CASPT2.