

1 Overview

1.1 Purpose

The program is intended to help with evaluation of the many index expressions which arise in multi-reference perturbation theory. Essentially, it aims to evaluate one of the following three kinds of expressions, the simplest of which is

$$\langle M | \hat{X} \hat{Y} \dots | N \rangle, \quad (1)$$

here \hat{X} and \hat{Y} are some arbitrary operators, and $|N\rangle$ is a multireference wavefunction represented as a linear combination of determinants, $|I\rangle$;

$$|N\rangle = \sum_I c_I^N |I\rangle. \quad (2)$$

There are three main types of terms the program can calculate. The simplest is :

$$\sum_{\substack{x_1 x_2 \dots \\ y_1 y_2 \dots \\ \dots}} X_{x_1 x_2 \dots} Y_{y_1 y_2 \dots} \sum_I \sum_J \langle I | a_{x_1}^\dagger a_{x_2} \dots a_{y_1}^\dagger a_{y_2} \dots | J \rangle c_I^{M\dagger} c_J^N \quad (3)$$

Which is just (1) written using second quantization; $X_{x_1, x_2 \dots}$ is a representation of operator \hat{X} in a basis of molecular orbitals.

It is also possible to calculate derivatives of (3) with respect to ci-coefficients, c_I^N ;

$$\sum_{\substack{x_1 x_2 \dots \\ y_1 y_2 \dots \\ \dots}} X_{x_1 x_2 \dots} Y_{y_1 y_2 \dots} \sum_J \langle I | a_{x_1}^\dagger a_{x_2} \dots a_{y_1}^\dagger a_{y_2} \dots | J \rangle c_J^N \quad (4)$$

In a similar vein, it is possible to expressions of form:

$$\begin{aligned} & \sum_{\substack{x_1 x_2 \dots \\ y_1 y_2 \dots \\ \dots}} X_{x_1 x_2 \dots} Y_{y_1 y_2 \dots} \sum_{\Omega} \sum_I \sum_J \langle I | \hat{E}_{\Omega}^\dagger a_{x_1}^\dagger a_{x_2} \dots a_{y_1}^\dagger a_{y_2} \dots | J \rangle c_I^{M\dagger} c_J^N \\ &= \sum_{\substack{x_1 x_2 \dots \\ y_1 y_2 \dots \\ \dots}} X_{x_1 x_2 \dots} Y_{y_1 y_2 \dots} \sum_I \sum_J \langle I | a_{\omega_1} a_{\omega_2} \dots a_{x_1}^\dagger a_{x_2} \dots a_{y_1}^\dagger a_{y_2} \dots | J \rangle c_I^{M\dagger} c_J^N \end{aligned} \quad (5)$$

where the \hat{E}_{Ω}^\dagger is an projection operator which excites electrons from the orbitals used to construct the space in which the reference wavefunctions, $\{|N\rangle\}$, were originally defined into some virtual space.

The program is intended to be flexible; and instead of asking for specific perturbation

theories or properties, the user has the option of specifying algebraic expressions directly via the input file.

1.2 Structural outline

The program is split into three main components; an algebraic manipulator, an fci and tensor contraction library, and an task manager (make figure).

The algebraic manipulator should the input expression, and rearranges it into a series of expressions which are more suited to computational evaluation. To accomplish this it makes extensive use of the commutation relations of the creation and annihilation operators, as well as the physical symmetries of the operators, in order to minimize the number of terms, and avoid the need for calculation of terms which are likely to be computationally expensive to evaluate (e.g., those with large numbers of indexes). This series of expressions is used to form an algebraic task list; a sequence of mathematical operations which need to be evaluated. It is important to note that the algebraic manipulator is entirely symbolic manipulation, and does not handle any of the large data structures, e.g., civectors, matrices of molecular orbital indices used. The expressions in the algebraic task list do not in any way specify what kind of data structures need to be used to store the quantities necessary for their evaluation.

The FCI and tensor libraries contains generic routines for calculating density matrices and their derivatives, and performing tensor contractions (and other various tensor manipulations). The tensor and FCI library are completely disctinct from the algebraic manipulator, and operate totally independently. The FCI library is designed such that it can calculate density matrices (and derivatives) or arbitrary order. In a similar vein, the tensor library can operate on tensor of arbitrary rank, with each dimension being an arbitrary and different size. At present both the FCI uses the dvec and determinant classes defined in bagel, but this should be phased out. Similarly, the tensor contraction makes use of the SMITH::Tensor_ class.

The final component of the program is the task manager which is intended to faciliate communication between the routines. This is necessary, as the neither the arithmetic component has now knowledge of the classes used in the algebraic manipulator, and vice versa. Whilst this has complicated the design slightly, it should enable greater portability, as well as making the program substantially easier to extend and upgrade. In a sense there is a fourth component, a driving routine, which performs some communication between the three routines, and which deals with interpretation of the input.

However, whilst this component is practically important, and not particularly small, it is rather simple, and not of interest, so I will not discuss it significantly.

2 Specific components