

## SOFTWARE METAPAPER

# DiracQ: A Package for Algebraic Manipulation of Non-Commuting Quantum Variables

John G. Wright<sup>1</sup> and B. Sriram Shastry<sup>2</sup><sup>1</sup> Research Assistant, Physics Department, University of California, Santa Cruz 95064, US  
jgwright1986@gmail.com<sup>2</sup> Professor, Physics Department, University of California, Santa Cruz 95064, US  
b.sriram.shastry@gmail.com

In several problems of quantum many body physics, one is required to handle complex expressions originating in the non-commutative nature of quantum operators. Their manipulation requires precise ordering and application of simplification rules. This can be cumbersome, tedious and error prone, and often a challenge to the most expert researcher. In this paper we present a software package DiracQ to facilitate such manipulations. The package DiracQ consists of functions based upon and extending considerably the symbolic capabilities of *Mathematica*. With DiracQ, one can proceed with large scale algebraic manipulations of expressions containing combinations of ordinary numbers or symbols (the c-numbers) and arbitrary sets of non-commuting variables (the q-numbers) with user defined properties. The DiracQ package is user extendable and comes encoded with the algebraic properties of several standard operators in popular usage. These include Fermionic and Bosonic creation and annihilation operators, spin operators, and canonical position and momentum operators. An example book is provided with some suggestive calculations of large-scale algebraic manipulations.

**Keywords:** Non-commuting Operators; Manipulation and Simplification of Expressions; Quantum Many-Body Physics; *Mathematica*; Symbolic Computer Algebra

## (1) Overview

### Introduction

The programming philosophy underlying the software package DiracQ is inspired by P A M Dirac's notation of c-numbers and q- numbers, denoting objects that are analogous to regular numbers and to non commuting quantum operators [1]. Dirac's notation pervades much of theoretical physics, and underlies its characteristic informality of syntax, relative to formal mathematical literature. The package DiracQ permits user defined expressions to be similarly informal combinations of (symbolic) commuting variables and non-commuting operators. DiracQ consists of functions designed to extend and compliment the abilities of *Mathematica* to perform manipulations with non-commuting quantum operators. Once loaded within *Mathematica*, the package DiracQ allows the user to perform algebraic operations with the most frequently encountered quantum objects. The package enables the user to evaluate commutators, anticommutators or products of expressions, and to manipulate and often greatly simplify the resulting expressions.

In summary, DiracQ works with expressions in a fashion that is very close to a theoretical physicists "natural" way of formulating quantum problems. This feature makes it especially easy to use for physicists and can be

very advantageous in pedagogical settings, where it may facilitate student concentration on physics related logic instead of the specificities of programming.

### An Elementary Example

We now provide a simple example of *Mathematica* input and output, to demonstrate the motivation and usage of the package. In this example, we will take a single spin-1/2 particle in an arbitrary magnetic field. The Zeeman Hamiltonian of this system is given by

$$H = A \sigma_x + B \sigma_y + C \sigma_z, \quad (1)$$

where A, B, and C are the components of the magnetic field in the three Cartesian directions in suitable units, the set  $\{\sigma_x, \sigma_y, \sigma_z\}$  are the usual Pauli matrices. By a specific rotation of the Cartesian axes, H can be diagonalized. However, if we are interested only in the eigenvalues of H, and not its eigenfunctions, we can avoid the diagonalization altogether. A simple shortcut may be used, exploiting the two properties of Pauli matrices (a) for any component j,  $(\sigma_j)^2 = \mathbf{e}$  the identity matrix, and (b) the anti-commuting property  $\{\sigma_i, \sigma_j\} = 0$  for distinct i, j. Therefore by squaring H, we obtain identity times the square of the net field. This is easily done by hand and yields the net field

$h = \sqrt{A^2 + B^2 + C^2}$  so that the eigenvalues are  $h$  and  $-h$ . Let us next see how this problem is done using DiracQ. We define this Hamiltonian in Mathematica using the input below<sup>1</sup>.

Input:  $H = A \sigma[i, x] + B \sigma[i, y] + C \sigma[i, z]$ ;

Note that  $H$  is a typical mix of commuting ( $A, B, C$ ), and non-commuting (Pauli matrix) objects. The standard Mathematica function **NonCommutativeMultiply** (**\*\***) is not very useful for evaluating the square of the Hamiltonian;  $H**H$  leads to a sum over nine terms of the type  $(A \sigma[i, x])**(B \sigma[i, y])$ . This output can be further processed by the standard Mathematica function **Simplify**, which gives back the same result. One would like to implement further rules that declare  $A$  and  $B$  to be commuting objects that can be moved to any position in the product, while the ordering and simplification should only affect the product of Pauli matrices  $\sigma[i, x]**\sigma[i, y]$ . This is achieved in DiracQ, where one instructs the program that  $A, B$  and  $C$  are  $c$ -numbers, whereas the Pauli matrices are  $q$ -number operators, with their well known simplification rules<sup>2</sup>. With these preliminaries, the expression is immediately simplified to the correct answer<sup>3</sup>.

In: ProductQ[H,H]  
Out:  $A^2+B^2+C^2$

This rather elementary example illustrates DiracQ's ability to distinguish and separate  $c$ -numbers from  $q$ -numbers, and apply special algebraic properties of quantum operators. The problem at hand is trivial and hardly requires symbolic computation. However, similar tasks involving many copies of the Pauli matrices can compound to unmanageable proportions and require greater processing power<sup>4</sup>.

### Problems involving Fermions

DiracQ recognizes most common quantum operators and knows their algebraic properties. For example, when instructed by the user, the package will recognize  $f[i]$  and  $f^\dagger[j]$  to represent the Fermi annihilation and creation operators indexed by  $i$  and  $j$ , representing sites on a lattice. Functions of the package will utilize their algebraic properties, such as product rules and anticommutators appropriately, or as overridden/directed by the user. Input expressions can include standard summation notation using multiple summation indices. Summation indices interact appropriately with Kronecker delta functions in user input, or those that arise during evaluation or simplification of expressions.

Many important problems in quantum many body physics require the diagonalization of standard models, such as the Hubbard model. These models are defined on varying lattices with Fermi operators assigned to each site, and often with different spin or flavor indices. A typical numerical application requires setting up and diagonalizing the Hamiltonian matrix within a subspace defined by a fixed number of particles. While the diagonalization of a numerical matrix is a standard problem in numerical

analysis, where much progress has been made, we are interested in the other end of the problem: setting up the matrix. Here the physicist is expected to produce the numerical matrix starting from the abstract Hamiltonian on an appropriate lattice. This is often a tedious and error prone procedure. DiracQ efficiently handles this aspect of the problem. The example notebook provided in the DiracQ package distribution folder shows how to construct such a matrix in a typical case. For a small cluster of four sites, the Hubbard Hamiltonian and the basis states within the Fock space are set up, and the numerical matrix written out at the end. This procedure is easily extended to larger lattices and to other models. The eigensystem of the resulting matrix can be computed either within Mathematica itself, or if required, in a suitable external program.

### Advanced Problems

Users can not only use the predefined operators in DiracQ, but also define additional operators and provide their algebraic properties to DiracQ. Such operators and their algebraic properties will be recognized and implemented by all functions of the package. In this way DiracQ provides a new language for formulating algebraic quantum problems.

The DiracQ package distribution folder includes demonstration problems involving popular systems in statistical mechanics and many body physics. For example, we reproduce some of the crucial algebra contained in the seminal paper by R. J. Baxter on the integrability of the 8-vertex model [2], and B. S. Shastry's analogous proof of the integrability of the 1-dimensional Hubbard model [3], and its later extensions [4]. A recent work by Bukov et. al. illustrates the use of the DiracQ package for the evaluation of commutators in the problem of high-frequency periodically driven systems [5].

### Summary

The DiracQ package should find application in any research that involves manipulations of long string of non-commuting operators. We typically expect that these manipulations will arise in the fields of quantum condensed matter physics, quantum statistical mechanics, quantum field theory and nuclear physics, and also in some problems of quantum chemistry

### Implementation and architecture

The goal of our project was to develop a library of functions that would enable users to perform algebraic manipulations of expressions that include non-commuting operators as well as commuting numbers. The functions of the package all operate within the same underlying framework. User input expressions are first separated into individual components. Input expressions are broken into individual non-commuting operators, commuting symbols, numbers, and summation indices. These components of an input expression are stored in a nested list organized according to the type of objects found in the input expression. All functions of the package utilize this organizational framework for manipulation and

combination of expressions. After manipulation, the individual components of an expression are recombined to yield a result in familiar notation.

The package therefore contains two sets of functions: those functions whose purpose is decomposing input expressions into nested lists or composing output expressions from nested lists, and functions that users call to perform manipulations which rely on the former functions. The package is extensible in that users can relatively easily write new functions to manipulate expressions utilizing the foundational organization system.

### Quality control

Each function of the package has been tested individually to ensure that the algebraic manipulations carried out are correct. Combinations of functions have been tested with known examples. Series of manipulations using a large number of functions have been carried out to ensure they produce some known results. A notebook supplied in the package distribution folder provides several examples of the packages use and demonstrations of the functions of the package being used to obtain non-trivial known results.

## (2) Availability

### Operating system

Any system capable of running *Mathematica* 8 or higher.

### Programming language

*Mathematica* 8 or higher.

### Dependencies

*Mathematica* 8 and higher.

### Software location

#### Archive

**Name:** arXiv.org

**Persistent identifier:** arXiv:1301.4494, and <http://physics.ucsc.edu/~sriram/DiracQ>,

**License:** Open license under which the software is licensed

**Publisher:** John G Wright and B Sriram Shastry

**Date published:** 2/9/2015

#### Code repository GitHub

**Name:** DiracQ

**Identifier:** <https://github.com/jgwright1986/DiracQ>

**License:** GNU General Public License Version 2

**Date published:** 2/9/2015

### Language

English.

## (3) Reuse potential

The DiracQ package will find use in any research that requires the algebraic manipulation of expressions containing non-commuting quantum operators, especially in settings where the expressions or manipulations are particularly large or complex.

### Competing Interests

The authors declare that they have no competing interests.

### Notes

<sup>1</sup> The dummy index “i” is needed here since DiracQ defines the Pauli matrices as having two indices, the Cartesian indices (x,y,z) and the additional site or flavor index.

<sup>2</sup> Before running DiracQ, we must of course invoke it within the *Mathematica* session. For example we may use the command `Get["DiracQ_V1.m"]`, or use alternative schemes as described more fully in the writeup. We must also specify that the  $\sigma$  variables represent the Pauli operators using the control palette that pops up with the invocation of DiracQ, (otherwise they will be treated as c-numbers).

<sup>3</sup> Note that the DiracQ function `ProductQ[A,B]` can be replaced with the alternate DiracQ command `A ⊗ B`.

<sup>4</sup> A sledgehammer approach to the class of problems involving many spin half particles is to represent the non commuting objects as matrices in a suitable dimension. We presume that the readers of this article are looking for more elegant solutions.

### References

1. **Dirac, P A M** 1927 Proc. Roy. Soc., A113: 621. DOI: <http://dx.doi.org/10.1098/rspa.1927.0012>
2. **Baxter, R J** 1971 Eight-Vertex Model in Lattice Statistics. *Phys. Rev. Lett.*, 26: 832–833. DOI: <http://dx.doi.org/10.1103/PhysRevLett.26.832>
3. **Shastry, B S** 1988 Decorated Star Triangle relations and the Exact Integrability of the 1-d Hubbard model. *Jour. Stat. Phys.*, 50: 57–79. DOI: <http://dx.doi.org/10.1007/BF01022987>
4. **Shiroishi, M** and **Wadati, M** 1995 Yang-Baxter Equation for the R-Matrix of the One-Dimensional Hubbard Model. *Jour. Phys. Soc. Japan*, 64: 57–63. DOI: <http://dx.doi.org/10.1143/JPSJ.64.57>
5. **Bukov, M, D'Alessio, L** and **Polkovnikov, A** 2014 Universal High-Frequency Behavior of Periodically Driven Systems: from Dynamical Stabilization to Floquet Engineering. *Advances in Physics*, 64(2): 139–226. DOI: <http://dx.doi.org/10.1080/00018732.2015.1055918>

**How to cite this article:** Wright, J G and Shastry, B S 2015 DiracQ: A Package for Algebraic Manipulation of Non-Commuting Quantum Variables. *Journal of Open Research Software*, 3: e13, DOI: <http://dx.doi.org/10.5334/jors.cb>

**Published:** 20 November 2015

**Copyright:** © 2015 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License (CC-BY 3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/3.0/>.

 *Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS 