Cosmological Coherent State Expectation Values in LQG
A Mathematica package for expectation values of (polynomial) operators

Klaus Liegener,* and Łukasz Rudnicki,†

(Dated: November 27, 2021)

I. INTRODUCTION

This package presents a step-by-step implementation of the algorithm proposed in [1]. We follow the same structure and notation and thus refer to there for all detailed steps.
The usef is able to implement the operator of interest in the section "User Input: Operator" of *Main* (the file to be run eventually). We provide details on how to do this in the next section.

Important comments / requirements:

- Ensure that supplementary file *DiracQV_mod* is contained in the same folder as the file *Main*
- Output of *Main* is the expectation value in form of a text file: "ExpValue_Result_[Title]". Make sure to have writing right to the folder where *Main* is located or change the outputs file destination at the very end of *Main*.

II. HOW TO USE *MAIN*

The code is designed in such a way that in order to compute the expectation value of any operator of your choice, all relevant settings can be specified in the section "User Input: Operator" of *Main* (the file to be run eventually).
If you do not know what you do, do not alter other parts of the file.

[1] The main operator is specified as "Oper=..." (see figure 2.1) for an operator of interest

$$Oper \;=\; O(\{h\},\{V\},\{Q\},) \tag{II.1}$$

which is polynomial in

- holonomies $h(e) = h[\mu, \{v1, v2, v3\}, a, b, J]$,

*Electronic address: klaus.liegener@tum.de
†Electronic address: lukasz.rudnicki@ug.edu.pl

```
Title = "CE";
$RecursionLimit = 16 390;
|Rekursionsgrenze
Oper =
  pEuc *
   Sum[Sign[d3] s1 s2 (HEuclCompact[{0, 0, 0}, a1, a2, a3, a4, a5, aa1, aa2, aa3,
   |s···  |Vorzeichen
        {s1 Mod[Abs[d3] + 1, 3, 1], s2 Mod[Abs[d3] + 2, 3, 1], d3}]), {d3, L}] ** Ω[];
            |M···  |Absolutwert                |M···  |Absolutwert
M = 4;
Jmax = 2;
MVolAppr = 3;
IndicesIni0 = {{a1, -1/2, 1/2}, {a2, -1/2, 1/2}, {a3, -1/2, 1/2}, {a4, -1/2, 1/2},
   {a5, -1/2, 1/2}, {aa1, -1/2, 1/2}, {aa2, -1/2, 1/2}, {aa3, -1/2, 1/2}};
VertList = {{s1, {-1, 1}}, {s2, {-1, 1}}};
```

FIG. 2.1: Section "User Input: Operator" contains the presented crucial lines, where the input is defined. The defintion of the several parameters are specified in the main text.

- Volume operators $V(\vec{v}) = V[\{v1, v2, v3\}]$
- powers of Q operators $Q^N(\vec{v}) = Q[v1, v2, v3, N]$

where any edge e is parametrised by its starting vertex $\vec{v}$ and its direction μ and J denotes some irreducible representation of $SU(2)$.

[2] Utilizing the above building blocks, use ** to the denote the product of two operators, i.e.

$$\hat{X}\hat{Y} \quad \Longleftrightarrow \quad X^{**}Y \tag{II.2}$$

Use Commutator$[X, Y]$ to include the commutator.

Example: The following is valid input, typical for applications in LQG:

$$h[d, \vec{v}, a, b, 1/2]\ ^{**}\text{Commutator}[h[-d, \vec{v}', b, c, 1/2], V[\vec{v}]]/t \tag{II.3}$$

(Note that we divide by t in order to compensate by the higher classical order due to the commutator).

[3] At the end, multiply the operator by $^{**}\Omega\,[\,]$.

[4] The total size of the periodic lattice is M (operator extending theis range are wrapped around the torus). The highest appearing irrep of SU(2) in the operator is $Jmax$. The Volume Operator is approximated to order $MVolAppr$.

[5] For simplicity, do not provide the sums over the indices (magnetic indices and vertices). Instead these are stored in $IndicesIni0$ and $VerList$, respectively. The notation for an index s running from $a, ..., a+b$ is $\{s, \{a, a+1, ..., a+b\}\}$.

[6] For simplicity, some parameters of the examples (e.g. the overall prefactor, pEuc) can kept general throughout the script.

[7] One can increase RecursionLimit as much as required by the further computation. By default $ RecursionLimit=1024.

[8] Output will be stored in file with specified name "ExpValue_Result_[Title]". The output is a text file containing $f(\xi, \eta)$ of $\langle \Psi, \hat{O}\Psi \rangle = f(\xi, \eta) + \mathcal{O}(\sqcup)$, with ξ, η following the notation of [1].

In order to ease useage of the code, sections "UI-Examples" provide an extensive list of examples for interesting operators (within the context of LQG).

III. FURTHER COMMENTS

- *DiracQV_mod* differs form its original version DiracQV1 [2] by not invoking the user interface. Instead, required settings are hard-coded in the file.
- The code has been tested with Mathematica v9 and v11.2. Upwards compatibility has not been verified so far.
- The code has been tested on various versions of HPC instances. As any Mathematica code utilizing parallel computing, it will need to be told what kernels are available. This requires changes to the preamble of *Main*. See, e.g., [3] for discussions on this topic.

Acknowledgements:

K.L. acknowledges support German National Merit Foundation as well as the German Research Foundation (DFG) under Germanys Excellence Strategy - EXC 2121 "Quantum Universe" - 390833306 and DFG-project BA 4966/1-2. Ł.R. would like to acknowledge support by the Foundation for Polish Science (IRAP project, ICTQT, Contract No. 2018/MAB/5, cofinanced by the EU within the Smart Growth Operational Programme).

The code uses *DiracQ: A Quantum Many-Body Physics Package* by J.G. Wright and B.S. Shastry [2].

[1] K. Liegener, Ł. Rudnicki, Algorithmic approach to Cosmological Coherent State Expectation Values in LQG, CQG 38 205001 (2021), https://arxiv.org/abs/2012.07813

[2] J.G. Wright and B.S. Shastry, DiracQ: A Quantum Many-Body Physics Package, arXiv:1301 (2013), https://github.com/jgwright1986/DiracQ

[3] https://community.wolfram.com/groups/-/m/t/984003