PROCEEDINGS OF SCIENCE

Automated lattice perturbation theory and relativistic heavy quarks in the Columbia formulation

Christoph Lehner*
RIKEN BNL Research Center, Brookhaven National Laboratory, Upton, NY 11973, USA
E-mail: clehner@quark.phy.bnl.gov

We introduce a new computer algebra system optimized for use in lattice perturbation theory as well as continuum perturbation theory and a new framework to perform automated perturbative calculations on top of said computer algebra system. The new framework is used to tune the relativistic heavy quark action in the Columbia formulation at one loop in meanfield-improved perturbation theory. Preliminary results for the matching and O(a)-improvement of heavy-light axial vector currents with light domain-wall quarks are also presented.

The 30th International Symposium on Lattice Field Theory
June 24 – 29, 2012
Cairns, Australia

*Speaker.

PoS(Lattice 2012)126

1. Introduction

The automation of lattice perturbation theory (LPT) was pioneered by Lüscher and Weisz in Ref. [1], where lists were used to represent the information required to evaluate gluonic Feynman diagrams numerically. Using a similar data structure, these early results were extended in Ref. [2] to the case of fermionic actions, enhanced in Ref. [3] to also allow for complex smearing, and further refined in Refs. [4] and [5] for various fermionic actions.

Here we propose to use a more general data structure and represent the gluonic and fermionic actions as well as operators and matrix elements in a symbolic manner using a computer algebra system (CAS).[1] This approach has found wide use in the automation of continuum perturbation theory, where many high-loop calculations are preformed using FORM [6]. Unfortunately, FORM lacks features to efficiently handle the complicated vertices that arise in lattice actions. In this work, we introduce a new CAS optimized for LPT. We present a flexible framework on top of said CAS that is capable of performing perturbative calculations using a lattice regulator as well as a continuum regulator and to generate contractions for non-perturbative computations.

In the following we briefly describe the layout of the new framework and present first results for the one-loop O(a)-improvement of relativistic heavy quarks (RHQ) in the Columbia formulation [7] and the matching and O(a)-improvement of heavy-light axial vector operators in the on-shell limit.

2. Layout of the framework

In this section we give a brief overview of the new framework that consists of three C++ libraries: **libcas**, a computer algebra system, **libqft**, a quantum field theory library that sits on top of libcas, and **libint**, a library for the numerical evaluation of algebraic expressions of lattice loop integrals.

The library libcas parses algebraic expressions from text and XML input, implements a pattern matching algorithm that is largely compatible with FORM, and provides further methods to manipulate and simplify general algebraic expressions. One crucial feature of libcas that allows for the efficient use in the context of LPT is the *function map*. Compared to continuum perturbation theory the vertices of lattice actions are complicated functions of momenta. A straightforward expansion of all terms that appear in a typical Feynman diagram leads to an explosion of individual terms that is prohibitive for LPT. This issue is addressed within libcas by the function map that allows to replace sub-expressions (such as vertices) by references to unique functions in the function map in a straightforward manner. Hence the complicated momentum-dependence is represented as a separate function that only has to be evaluated once, and each instance of the vertex is represented by a mere function call with respective momenta as arguments. We make extensive use of the function map not only to handle the complexity of lattice vertices but also, e.g., to perform a general common subexpression elimination within the lattice loop integrands.

The library libqft extracts vertices from text-representations of lattice and continuum actions and operators, defines common lattice actions, performs Wick contractions, and computes loop integrals in dimensional regularization using Passarino-Veltman reduction [8].

[1] The possibility of such an alternative approach was briefly mentioned already in Ref. [1].

```
// create action from formula string
from(" sum(x)*Qb(x)*( "
     " + sum(i1,4)*zeta(i1)*Ngamma(i1)*aD(i1,x) + am0 "
     " - sum(i1,4)*r(i1)*aD(i1,i1,x)/2 "
     " + sum(i1,4)*sum(i2,4)*i_*cp(i1,i2)/4"
     "   *Nsigma(i1,i2)*aF(i1,i2,x) "
     " )*Q(x) ");
```

Figure 1: The definition of the RHQ action of Eq. (3.1) within libqft is shown above. The function *from* is defined within libqft and extracts the Feynman rules from the text representation of the action. Operators such as the covariant derivative *aD* or the field-strength tensor *aF* are defined within libqft. Special functions such as *sum* and symbols such as the complex *i_* are defined within libcas. The γ-algebra is represented by non-commuting functions *Ngamma* and *Nsigma*. For a more detailed explanation of the above code, we refer to Ref. [9].

The library libint provides a convenient framework to numerically evaluate the lattice loop integrands with a variety of integration methods.

All three libraries are designed to be easily extensible and will eventually be released as open source; a complete list of features will be given in an upcoming publication [9].

3. Example: Applications to heavy quark physics

The framework found its first applications within the heavy-quark physics project of the RBC and UKQCD collaborations that uses RHQ in the Columbia formulation to describe the heavy quarks. Recent results have been presented in Refs. [10] and [11]. Relativistic heavy quarks, first proposed in Ref. [12] and further refined in Refs. [13] and [7], provide an effective heavy-quark action for large quark masses that is smoothly connected to a fully relativistic quark action as the quark mass becomes small compared to the lattice cutoff. The Columbia formulation [7] corresponds to the lattice action

$$S = \sum_x \overline{Q}(x) \left((\gamma_0 D_0 - \frac{1}{2} D_0^2) + \zeta \sum_{i=1}^{3} (\gamma_i D_i - \frac{1}{2} D_i^2) + m_0 + c_P \sum_{\mu,\nu=0}^{3} \frac{i}{4} \sigma_{\mu\nu} F_{\mu\nu}(x) \right) Q(x) \tag{3.1}$$

with heavy-quark fields Q. The parameters m_0, ζ, and c_P can be tuned to remove $\mathrm{O}(a\vec{p})$ discretization errors in on-shell quantities, where $a\vec{p}$ corresponds to the spatial momentum of the heavy quark in lattice units. Figure 1 shows how this action is defined within libqft from a text representation. If we allow for a field rotation

$$Q'(x) = Q(x) + d_1 \sum_{i=1,2,3} \gamma_i D_i Q(x) \tag{3.2}$$

with parameter d_1, we can match the quark fields Q' to continuum fields. Such a field rotation, however, leaves the mass spectrum of the theory invariant. Therefore the parameters m_0, ζ, and c_P can be tuned non-perturbatively without knowledge of d_1; results for the tuning of bottom quarks were recently presented in Ref. [14].

In order to test the framework, we also performed a perturbative tuning of the parameters. We match the bilinear

$$S(p) = \sum_q \langle Q'(p) \overline{Q}'(q) \rangle \tag{3.3}$$

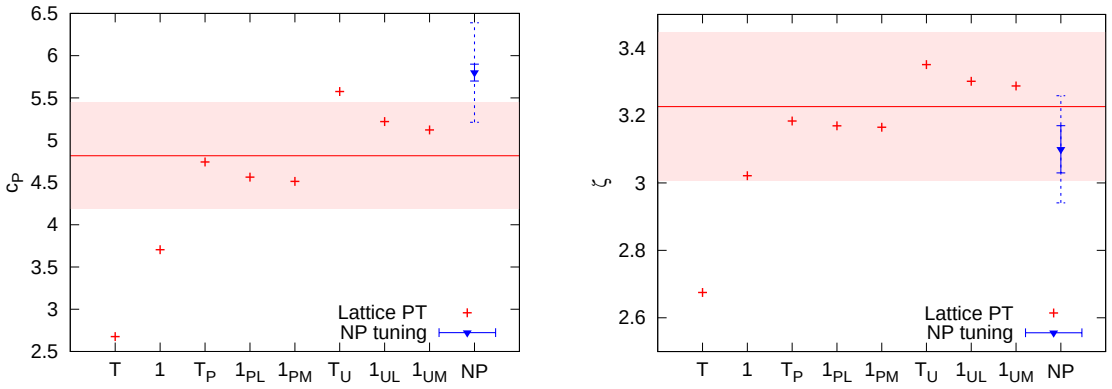

Figure 2: Results for the perturbative tuning of parameters c_P and ζ on the 24^3 ensembles of Ref. [14] for bottom quarks. We compare perturbative results with results of non-perturbative (NP) tuning. Perturbative results are given at tree level (T) or at one-loop level (1). The subscript P indicates that we use the average value of the plaquette for meanfield improvement, the subscript U indicates that the Landau gauge-fixed average link value was used for meanfield improvement. The subscript L denotes expansion in the bare lattice coupling, the subscript M denotes expansion in the $\overline{\text{MS}}$ coupling constant at scale $1/a$, where a is the lattice spacing. The estimation of the errors is described in detail in Ref. [14].

to the continuum for on-shell momenta and determine m_0 from the position of the pole, ζ from the dispersion relation, and d_1 from the spinor structure. The parameter c_P is obtained by matching the three-point function

$$\Lambda^a_\mu(p,q) = \left[\sum_k \langle Q'(q) A^a_\mu(k) \overline{Q}'(-p)\rangle\right]_{\text{amp}} = \left(\sum_{p'} \langle Q'(q)\overline{Q}'(p')\rangle\right)^{-1} \left(\sum_k \langle Q'(q) A^b_\nu(k)\overline{Q}'(-p)\rangle\right) \times \left(\sum_{q'} \langle Q'(p)\overline{Q}'(q')\rangle\right)^{-1} [D(p+q)^{-1}]^{ba}_{\nu\mu} \tag{3.4}$$

to the continuum in the on-shell limit, where

$$D(k)^{ab}_{\mu\nu} = \sum_{k'} \langle A^a_\mu(k) A^b_\nu(k')\rangle \tag{3.5}$$

is the lattice gluon propagator with color indices a, b and Lorentz indices μ, ν. Figure 2 compares the perturbative and non-perturbative results for c_P and ζ on the 24^3 lattices of Ref. [14]. For more details of the calculation, we refer to Refs. [14] and [9]. Within estimated errors, the perturbative and non-perturbative results agree.

The corresponding C++ code for the calculation of the necessary diagrams is shown in Fig. 3, and the resulting diagrams for the vertex are displayed in Fig. 4. The one-loop integral for the propagator is evaluated to 10^{-3} relative accuracy on a regular desktop computer in less than ten seconds.

In Ref. [11] preliminary results for f_B and f_{B_s} were presented with light domain-wall fermions. The decay constants are extracted from matrix elements of

$$A_0^{\text{cont}}(x) = \rho_{bl}\sqrt{Z_V^{bb} Z_V^{ll}}\left[\overline{q}(x)\gamma_0\gamma_5 Q(x) + c_1 \sum_{i=1,2,3} \overline{q}(x)\gamma_0\gamma_5\gamma_i D_i Q(x)\right], \tag{3.6}$$

```
Context c;

// use rhq + gauge action
ActionRHQ rhq(&c,"Q");
ActionGAUGE gauge(&c);

// define field rotations
c.coefficients << "d1FT";
const char* QimpD =
 "(1 + sum(i,4)*d1FT(i)*Ngamma(i)*aD(i,x))*Q(x)";
FieldRotationRHQ Qimp(&c,"Q","QimpmomT",QimpD);
const char* QbimpD =
 "Qb(x)*(1 - sum(j,4)*d1FT(j)*Ngamma(j)*aDl(j,x))";
FieldRotationRHQ Qbimp(&c,"Qb","QbimpmomT",QbimpD);

// perform wick contractions
Wick w(&c);
w << rhq << gauge << Qimp << Qbimp;
Expression* vertex = w.contract(
 "sum(k,mom)*QimpmomT(q)*aACmom(mu1,a1,k)*QbimpmomT(-p)",3);
Expression* prop = w.contract(
 "sum(q,mom)*QimpmomT(p)*QbimpmomT(q)",2);
```

Figure 3: The results of Fig. 2 are obtained using the code displayed above. The code defines the field rotations of Eq. (3.2) and generates the one-loop diagrams for the propagator of Eq. (3.3) and the vertex of Eq. (3.4). The resulting diagrams for the vertex are shown in Fig. 4. A more detailed explanation of the code shown above is given in Ref. [9].

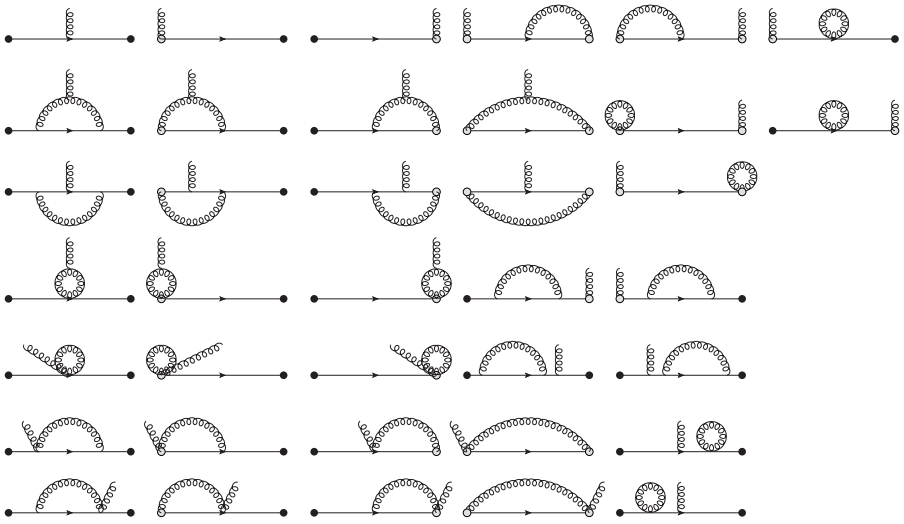

Figure 4: One-loop diagrams of the quark-gluon vertex of the RHQ action in the Columbia formulation defined in Eq. (3.4).

where A_0^{cont} is the temporal component of the heavy-light axial vector operator in the continuum, q is the domain-wall quark field, and Q is the relativistic heavy quark field. The lattice matrix element is related to the continuum version using the non-perturbatively determined light-light vector matching factor Z_V^{ll} and the perturbatively determined coefficients $\eta_{bl} = \rho_{bl}\sqrt{Z_V^{bb}}$ and c_1 with heavy-heavy vector matching factor Z_V^{bb}. The coefficient c_1 is tuned to remove $\mathrm{O}(a\vec{p})$ errors. In Tab. 1 we present preliminary meanfield-improved perturbative results at one loop for the coef-

	$(\eta_{bl})^{(0)}$	$(\eta_{bl})^{(1)}$	$(c_1)^{(0)}$	$(c_1)^{(1)}$
Plaquette	3.2615	−0.0099	0.0727	0.0257
Landau link	3.3195	−0.0107	0.0753	0.0254

Table 1: Preliminary perturbative results for $\eta_{bl} = (\eta_{bl})^{(0)} + g^2(\eta_{bl})^{(1)}$ and $c_1 = (c_1)^{(0)} + g^2(c_1)^{(1)}$, where g is the strong coupling constant, for the 24^3 ensemble used in Ref. [11]. The results are given for meanfield improvement using the average plaquette or the Landau gauge-fixed average link.

```
// use rhq + gauge action
ActionDWF dwf(&c,"L");
ActionRHQ rhq(&c,"Q");
ActionGAUGE gauge(&c);

// define field rotations (RHQ)
c.coefficients << "d1FT";
const char* QbimpD =
"u^(1/2)*Qb(x)*(1 - sum(j,4)*(d1FT(j)/u)*Ngamma(j)*aDl(j,x))";
FieldRotationRHQ Qbimp(&c,"Qb","QbimpmomT",QbimpD);

// define field rotations (DWF 4d fields)
const char* lD =
"u^(1/2)*sum(l,1,N5)*(PL*delta(l,1) + PR*delta(l,N5))*L(l,x)";
FieldRotationDWF l(&c,"L","lmomT",lD);

// define improved vector operator
c.commuting << "c1FT";
FieldOperatorBilinear Vimp(c,"Vimp","lbmomT",
 "Ngamma(rho)*(1 + sum(j,4)*(c1FT(j)/u)*aD(j,x)*Ngamma(j))",
 "QmomT",2);

// perform wick contractions
Wick w(&c);
w << rhq << dwf << gauge << l << Qbimp << Vimp;
Expression* hl_op =
 w.contract("sum(P,mom)*lmomT(q)*Vimp(P)*QbimpmomT(-p)",2);
```

Figure 5: The above code was used to obtain the results of Tab. 1. Field rotations are used to define the four-dimensional domain-wall quark. For a more detailed explanation, we refer to Ref. [9].

ficients η_{bl} and c_1 for the 24^3 ensembles used in Ref. [11]. The perturbative results are based on the code displayed in Fig. 5.

4. Concluding remarks

A main virtue of the new framework presented in this paper is its versatility through the use of the new CAS. It provides a unified environment to perform perturbative calculations with both lattice as well as continuum regulators and is also able to generate contractions for non-perturbative computations. All expressions are stored in an algebraic manner and can be modified in a straightforward way using pattern-matching techniques.

We presented results of a first application in the context of the heavy quark physics program of the RBC and UKQCD collaborations. The framework is designed with higher-loop calculations in mind; a challenge that we plan to address soon. A publication containing further details is in progress [9].

Acknowledgments

I would like to thank my colleagues of the RBC and UKQCD collaborations for many useful discussions and acknowledge support by the RIKEN FPR program.

References

[1] M. Lüscher and P. Weisz, *Efficient numerical techniques for perturbative lattice gauge theory computations*, *Nucl. Phys.* **B266** (1986) 309.

[2] A. Hart, G. von Hippel, R. Horgan, and L. Storoni, *Automatically generating Feynman rules for improved lattice field theories*, *J. Comput. Phys.* **209** (2005) 340 [hep-lat/0411026].

[3] A. Hart, G. von Hippel, R. Horgan, and E. Müller, *Automated generation of lattice QCD Feynman rules*, *Comput. Phys. Commun.* **180** (2009) 2698 [arXiv:0904.0375].

[4] S. Takeda, *Automatic generation of Feynman rules in the Schrödinger functional*, *Nucl. Phys.* **B811** (2009) 36 [arXiv:0808.3065].

[5] D. Hesse, R. Sommer, and G. von Hippel, *Automated lattice perturbation theory applied to HQET*, PoS(Lattice 2011)229.

[6] J. A. M. Vermaseren, *New features of FORM*, math-ph/0010025.

[7] N. H. Christ, M. Li, and H.-W. Lin, *Relativistic heavy quark effective action*, *Phys. Rev.* **D76** (2007) 074505 [hep-lat/0608006].

[8] G. Passarino and M. J. G. Veltman, *One loop corrections for e+ e- annihilation into mu+ mu- in the Weinberg model*, *Nucl. Phys.* **B160** (1979) 151.

[9] C. Lehner, *In preparation*, 2012.

[10] T. Kawanai, R. S. Van de Water, and O. Witzel, *The $B \to pi\ l\ nu$ form factor from unquenched lattice QCD with domain-wall light quarks and relativistic b-quarks*, PoS(Lattice 2012)109 [arXiv:1211.0956].

[11] O. Witzel, *Calculating B-meson decay constants using domain-wall light quarks and nonperturbatively tuned relativistic b-quarks*, PoS(Lattice 2012)103 [arXiv:1211.3180].

[12] A. X. El-Khadra, A. S. Kronfeld, and P. B. Mackenzie, *Massive fermions in lattice gauge theory*, *Phys. Rev.* **D55** (1997) 3933–3957 [hep-lat/9604004].

[13] S. Aoki, Y. Kuramashi, and S.-i. Tominaga, *Relativistic heavy quarks on the lattice*, *Prog. Theor. Phys.* **109** (2003) 383–413 [hep-lat/0107009].

[14] Y. Aoki *et. al.*, *Nonperturbative tuning of an improved relativistic heavy-quark action with application to bottom spectroscopy* [arXiv:1206.2554].