Automated Perturbation Theory for staggered and minimally doubled fermions

G.M. von Hippel

Mainz Institute of Theoretical Physics
Johannes Gutenberg-Universität Mainz

Mainz, 8 March 2023

Why is Lattice Perturbation Theory hard?

- Feynman rules become very complex
  - $U_\mu(x) = e^{igaA_\mu(x+\frac{1}{2}\hat{\mu})} \Rightarrow$ vertices of arbitrary adicity
  - Discrete Fourier-transform $\Rightarrow$ trigonometric functions appear
  - Improved actions $\Rightarrow$ many additional operators contribute
  - No Lorentz symmetry $\Rightarrow$ more complicated Lorentz index structures allowed
- Number of diagrams grows very quickly
  - Vertices of arbitrary adicity $\Rightarrow$ many possible contractions from high-adicity vertices
  - Book-keeping effort grows rapidly
- Standard tricks don't work
  - Trigonometric functions $\Rightarrow$ inverse propagator is not a quadratic form
  - No Lorentz symmetry $\Rightarrow$ cannot use it to simplify numerators
  - Cannot complete squares; no Schwinger parameter representation; no Feynman trick

Why is Lattice Perturbation Theory hard?

$$
\begin{aligned}
W^{abcd}_{\mu\nu\lambda\rho}(p,q,r,s) \quad = \quad & -g_0^2 \Bigg\{ \sum_e f_{abe} f_{cde} \Bigg[\delta_{\mu\lambda}\delta_{\nu\rho} \left(\cos\frac{a(q-s)_\mu}{2} \cos\frac{a(k-r)_\nu}{2} - \frac{a^4}{12}\, \widehat{k}_\nu \widehat{q}_\mu \widehat{r}_\nu \widehat{s}_\mu \right) \\
& -\delta_{\mu\rho}\delta_{\nu\lambda} \left(\cos\frac{a(q-r)_\mu}{2} \cos\frac{a(k-s)_\nu}{2} - \frac{a^4}{12}\, \widehat{k}_\nu \widehat{q}_\mu \widehat{r}_\mu \widehat{s}_\nu \right) \\
& +\frac{1}{6}\delta_{\nu\lambda}\delta_{\nu\rho} a^2\, (\widehat{s-r})_\mu\, \widehat{k}_\nu \, \cos\frac{aq_\mu}{2} - \frac{1}{6}\delta_{\mu\lambda}\delta_{\mu\rho} a^2\, (\widehat{s-r})_\nu\, \widehat{q}_\mu \, \cos\frac{ak_\nu}{2} \\
& +\frac{1}{6}\delta_{\mu\nu}\delta_{\mu\rho} a^2\, (\widehat{q-k})_\lambda\, \widehat{r}_\rho \, \cos\frac{as_\lambda}{2} - \frac{1}{6}\delta_{\mu\nu}\delta_{\mu\lambda} a^2\, (\widehat{q-k})_\rho\, \widehat{s}_\lambda \, \cos\frac{ar_\rho}{2} \\
& +\frac{1}{12}\delta_{\mu\nu}\delta_{\mu\lambda}\delta_{\mu\rho} a^2 \sum_\sigma (\widehat{q-k})_\sigma (\widehat{s-r})_\sigma \Bigg] \\
& +(b\leftrightarrow c, \nu\leftrightarrow\lambda, q\leftrightarrow r) + (b\leftrightarrow d, \nu\leftrightarrow\rho, q\leftrightarrow s) \Bigg\} \\
& +\frac{g_0^2}{12}\, a^4 \left\{ \frac{2}{3}(\delta_{ab}\delta_{cd}+\delta_{ac}\delta_{bd}+\delta_{ad}\delta_{bc}) + \sum_e (\delta_{abe}\delta_{cde}+\delta_{ace}\delta_{bde}+\delta_{ade}\delta_{bce}) \right\} \\
& \times \Bigg\{ \delta_{\mu\nu}\delta_{\mu\lambda}\delta_{\mu\rho} \sum_\sigma \widehat{k}_\sigma \widehat{q}_\sigma \widehat{r}_\sigma \widehat{s}_\sigma - \delta_{\mu\nu}\delta_{\mu\lambda}\, \widehat{k}_\rho \widehat{q}_\rho \widehat{r}_\rho \widehat{s}_\mu \\
& -\delta_{\mu\nu}\delta_{\mu\rho}\, \widehat{k}_\lambda \widehat{q}_\lambda \widehat{s}_\lambda \widehat{r}_\mu - \delta_{\mu\lambda}\delta_{\mu\rho}\, \widehat{k}_\nu \widehat{r}_\nu \widehat{s}_\nu \widehat{q}_\mu - \delta_{\nu\lambda}\delta_{\nu\rho}\, \widehat{q}_\mu \widehat{r}_\mu \widehat{s}_\mu \widehat{k}_\nu \\
& +\delta_{\mu\nu}\delta_{\lambda\rho}\, \widehat{k}_\lambda \widehat{q}_\lambda \widehat{r}_\mu \widehat{s}_\mu + \delta_{\mu\lambda}\delta_{\nu\rho}\, \widehat{k}_\nu \widehat{r}_\nu \widehat{q}_\mu \widehat{s}_\mu + \delta_{\mu\rho}\delta_{\nu\lambda}\, \widehat{k}_\nu \widehat{s}_\nu \widehat{q}_\mu \widehat{r}_\mu \Bigg\}
\end{aligned}
$$

Why is Lattice Perturbation Theory hard?

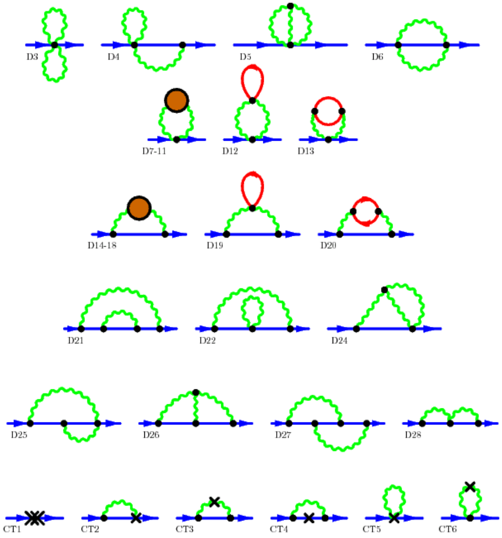
- This was the simplest possible gauge action (Wilson)!
- Improvement makes things much worse:

	# of terms for				
	prop.	3-gluon	4-gluon	5-gluon	6-gluon
Wilson	76	240	826	1583	3184
Symanzik	252	1456	6022	15488	36148

- Quark actions become even worse after improvement:

	# of terms for			
	prop.	1-gluon	2-gluon	3-gluon
Naive	9	8	8	8
Asqtad	17	1248	8376	29480
Fat3	9	152	2888	54584
Fat5	9	632	49928	?
Fat7	9	1080	145800	?

Why is Lattice Perturbation Theory hard?



Why is Lattice Perturbation Theory hard?

- The number of Feynman diagrams grows quickly:

	continuum	lattice
1-loop quark s.e.	1	2
1-loop gluon s.e.	4	7
1-loop qqg vertex corr.	2	6
1-loop ggg vertex corr.	8	16
2-loop quark s.e.	11	29

- If one wishes to simplify the numerator algebra on the lattice, the lack of Lorentz invariance makes the calculations much more involved.
- Ultimately, lattice integrals cannot be evaluated analytically, and have to be treated with numerical integration methods.

What can we do about it?

- Automate the derivation of the Feynman rules
  - Avoid human error in the extremely complex calculation
  - Enable use of highly improved actions, iterated smearing and higher orders
  - Algorithm easily implementable in any language with list and dictionary types (e.g. Python)
  - Output in machine-readable format for use by separate generic vertex code
- Automate the generation of the Feynman diagrams
  - Take advantage of generic vertex code
  - At least for low orders, we can simply apply Wick's theorem
- Automate the evaluation of the Feynman diagrams
  - Use numerical integration (e.g. VEGAS) or finite-volume sums
  - Employ techniques of automatic differentiation

Automating the derivation of lattice Feynman rules

- Lattice gauge action is given as a sum of Wilson loops:

$$S_g = \beta \sum_i c_i \sum_x \mathrm{Re}\,\mathrm{Tr}\left(U_{\mathcal{C}_i}(x)\right)$$

- We want the action in vertex form:

$$S_g = \sum_r \frac{g_0^{r-2}}{r!} \sum_{x,v_1,\dots,v_r} V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x,v_1,\dots,v_r) A^{a_1}_{\mu_1}(v_1)\cdots A^{a_r}_{\mu_r}(v_r)$$

- From this, the vertex functions in momentum space are obtained as

$$\begin{aligned} V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(k_1,\dots,k_r) &= \mathrm{Tr}\left(t^{a_1}\cdots t^{a_r}\right) \\ &\times \sum_i f_{\{\mu\},i}\, \mathrm{e}^{i\sum_j k_j\cdot v_{\{\mu\},i,j}} \end{aligned}$$

Automating the derivation of lattice Feynman rules

- Lattice fermion action is given as a sum of Wilson lines capped by fermion fields:

$$S_f = \sum_i c_i \sum_x \bar{\psi}(y_i)\Gamma_i U_{\mathcal{D}_i}(y_i, x)\psi(x)$$

- We want the action in vertex form:

$$S_f = \sum_r \frac{g_0^r}{r!} \sum_{y,x,v_1,\ldots,v_r} \overline{\psi}^b(y) W^{a_1\cdots a_r,bc}_{\mu_1\cdots\mu_r}(x,y,v_1,\ldots,v_r) A^{a_1}_{\mu_1}(v_1)\cdots A^{a_r}_{\mu_r}(v_r)\psi^c(x)$$

- From this, the fermionic vertex functions are obtained as

$$\begin{aligned} W^{a_1\cdots a_r,bc}_{\mu_1\cdots\mu_r}(k_1,\ldots,k_r;p,q) &= (t^{a_1}\cdots t^{a_r})_{bc} \\ &\times \sum_i f_{\{\mu\},i}\Gamma_{\alpha_{\{\mu\},i}} \mathrm{e}^{i\left(p\cdot x_{\{\mu\},i}+q\cdot y_{\{\mu\},i}+\sum_j k_j\cdot v_{\{\mu\},i,j}\right)} \end{aligned}$$

Automating the derivation of lattice Feynman rules

- Need to automate translation [Lüscher, Weisz, NPB **266** (1986) 309; Hart, vH, Horgan, Müller, arXiv:0904.0375]
  $\{(c, \Gamma, \mathcal{C})\} \mapsto \{(\{\mu\}, \{v\}, y, \Gamma, f)\}$
- The basic object is a gauge link

$$U_\mu(x) = e^{igaA_\mu(x+\frac{1}{2}\hat{\mu})} = \sum_{r=0}^{\infty} \frac{g^r}{r!} t^{a_1} \cdots t^{a_r} A_\mu^{a_1}(x + \tfrac{1}{2}\hat{\mu}) \cdots A_\mu^{a_r}(x + \tfrac{1}{2}\hat{\mu})$$

- Hence, basic mapping is

$$F(U_\mu) = \{(\mu; \tfrac{1}{2}\hat{\mu}; \hat{\mu}; \mathbb{1}; 1), (\mu, \mu; \tfrac{1}{2}\hat{\mu}, \tfrac{1}{2}\hat{\mu}; \hat{\mu}; \mathbb{1}; 1), \ldots\}$$

- Build action from links by using

$$\begin{aligned}
F(U_{\mathcal{C}} + U_{\mathcal{C}'}) &= F(U_{\mathcal{C}}) \cup F(U_{\mathcal{C}'}) \\
F(cU_{\mathcal{C}}) &= \{cE : E \in F(U_{\mathcal{C}})\} \\
F(\Gamma U_{\mathcal{C}}) &= \{\Gamma E : E \in F(U_{\mathcal{C}})\} \\
F(U_{\mathcal{C}} U_{\mathcal{C}'}) &= \{E * E' : E \in F(U_{\mathcal{C}}), E' \in F(U_{\mathcal{C}'})\}
\end{aligned}$$

Automating the derivation of lattice Feynman rules

- We need algebra of expansion entities $E = (\{\mu\}; \{v\}; y; \Gamma; f)$:

$$
\begin{aligned}
cE &= (\{\mu\}; \{v\}; y; \Gamma; cf) \\
\Gamma' E &= ((\{\mu\}; \{v\}; y; \Gamma'\Gamma; f) \\
E * E' &= (\{\mu\} \cup \{\mu'\}; \{v\} \cup \{v' + y\}; y' + y; \Gamma\Gamma'; Cff')
\end{aligned}
$$

- In practice, naive implementation of link algebra in terms of entity algebra will yield many redundant entities
- To avoid associated cost, implement link algebra in terms of dictionaries mapping partial entities $\hat{E} = (\{\mu\}; \{v\}; y; \Gamma)$ to their associated amplitude f
- Momentum conservation means that $(\{\mu\}; \{v\}; y; \Gamma)$ and $(\{\mu\}; \{v + c\}; y + c; \Gamma)$ give same contribution to vertex
- Identify and combine entities that differ only by a constant translation of their vectors

Automating the derivation of lattice Feynman rules

- The resulting expansion of the action in terms of entities is written out in a machine-readable format
- The HPSRC library of generic vertex functions `vertex_*(k,mu,a)` is used to construct Feynman diagrams
- The HPSRC library reads in the list of entities at runtime and constructs vertices from entities on the fly
- This separates the coding of the Feynman diagrams from the derivation of the Feynman rules
- Note that this allows to rerun a calculation with a different action with as little overhead as possible
- For very complex actions, the HPSRC library allows to factor the action into parts whose Feynman rules are derived separately and combined on the fly at runtime
- Also includes hand-written Feynman rules for Fadeev-Popov ghosts and for counterterms from the Haar measure

Automating the generation of lattice Feynman diagrams

- Feynman diagrams can be generated by applying Wick's theorem
- For one-loop calculations, it is generally easier to code the diagrams by hand
- For two-loop calculations, a straightforward implementation of Wick's theorem is currently used
- Standard tools like QGRAF could be used, but these would need adapting for Lattice QCD
- It is extremely desirable to output the diagrams directly as sequences of function calls to the HPSRC generic vertex routines
- Further work needed in this area if one wants to go to even higher orders

Automating the evaluation of lattice Feynman diagrams

- For the numerical evaluation of the lattice integrals we use parallel VEGAS (for lattices with infinite extent) or trivially parallelised finite-volume mode sums (for finite lattices)
- We usually need wavefunction renormalisation constants, i.e. derivatives of self-energy diagrams
- Require methods of automatic differentiations [vH, arXiv:0910.5111]
- Implement a type that encodes a function along with its first few derivatives w.r.t. external momentum
- Overload arithmetic operations to fulfill Leibniz's and Faà di Bruno's rules

$$\frac{d^n}{dx^n}(fg)(x) = \sum_{k=0}^{n} \binom{n}{k} f^k(x) g^{(n-k)}(x)$$

$$\frac{d^n}{dx^n} f(g(x)) = \sum_{\substack{0 \le k_\mu \le n \\ \sum_\mu \mu k_\mu = n}} \frac{n!}{\prod_\nu k_\nu! \nu!^{k_\nu}} \frac{d^{|k|} f}{dy^{|k|}} \prod_{\mu=1}^{n} \left(\frac{d^\mu g}{dx^\mu} \right)^{k_\mu}$$

- Derivatives of vertex functions are simple to compute (exponentials!)
- Overloaded operations take care of the rest

Applications

- Unquenching contributions from improved staggered quarks
  - asqtad [Hao, vH, Horgan, Mason, Trottier, arXiv:0705.4660]
  - HISQ [Hart, vH, Horgan, arXiv:0812.0503]
- Improving the NRQCD action
  - hyperfine splittings [Hammant, Hart, vH, Horgan, Monahan, arXiv:1105.5309, arXiv:1303.3234]
  - kinetic couplings [Davier, Harrison, Hughes, Horgan, vH, Wingate, arXiv:1812.11639]
  - hindered M1 radiative decays of $\Upsilon(2S)$ [Hughes, Dowdall, Davies, Horgan, vH, Wingate, arXiv:1508.01694]
  - leptonic widths of S-wave quarkonia [Hart, vH, Horgan, hep-lat/0605007]
- Perturbative subtraction of lattice artifacts in NPR
  - with Wilson quarks [Harris, vH, Junnarkar, Meyer, Ottnad, Wilhelm, Wittig, Wrang, arXiv:1905.01291]

Did you say staggered?

- Actually, the staggered calculations were done with naive fermions.
- But this is essentially the same thing (except easier):
  - The naive propagator is

$$S(x,y) = g(x,y)\Omega(x)\Omega(y)^\dagger$$

    in terms of the single-component staggered propagator g and the Kawamoto-Smit transform

$$\Omega(x) = \prod_\mu (\gamma_\mu)^{\frac{x_\mu}{a}}$$

  - whence the vertices become

$$\Omega(x+\hat{\mu})^\dagger \Gamma_\mu \Omega(x) = \alpha_\mu(x)\mathbb{1}$$

    in terms of staggered phase factors

$$\alpha_\mu(x) = (-1)^{\sum_{\nu<\mu} \frac{x_\nu}{a}}$$

- This implies that
  - naive loops are four times the corresponding staggered loop,
  - naive incoming (outgoing) legs give a factor Ω ($\Omega^\dagger$),
  - everything else just goes through.

Did you say staggered?

- There is actually a version of Reisz's power-counting theorem for staggered fermions [Giedt, hep-lat/0606003]
- No completed proof of perturbative renormalizability, though
- But no reason to expect serious obstacles

And what about minimally-doubled?

- No direct technical obstacles to applying automated methods
- Four-component spinors, Wilson-like structure (naive + symmetry-breaking)
- Breaking of discrete spacetime symmetries not a problem in automated treatment
- Taste structure may require some interpreting, though