Automated lattice perturbation theory

Chris Monahan

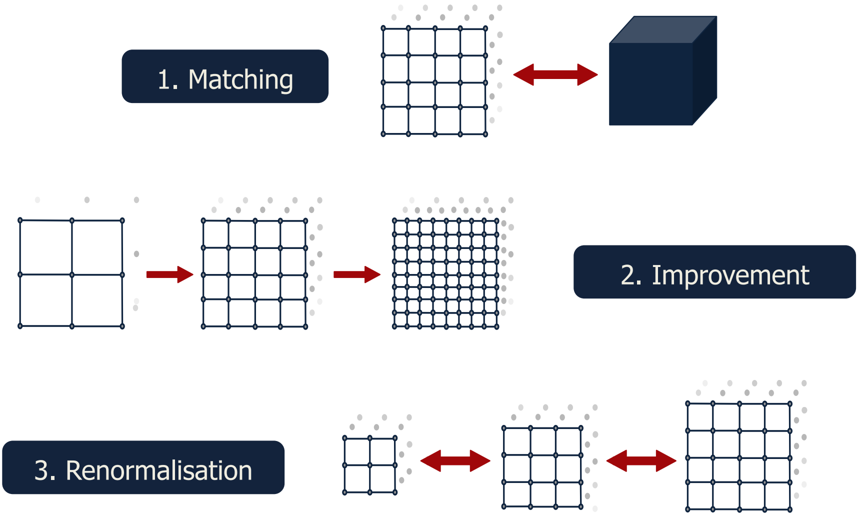
College of William and Mary/JLab

LATTICE 13

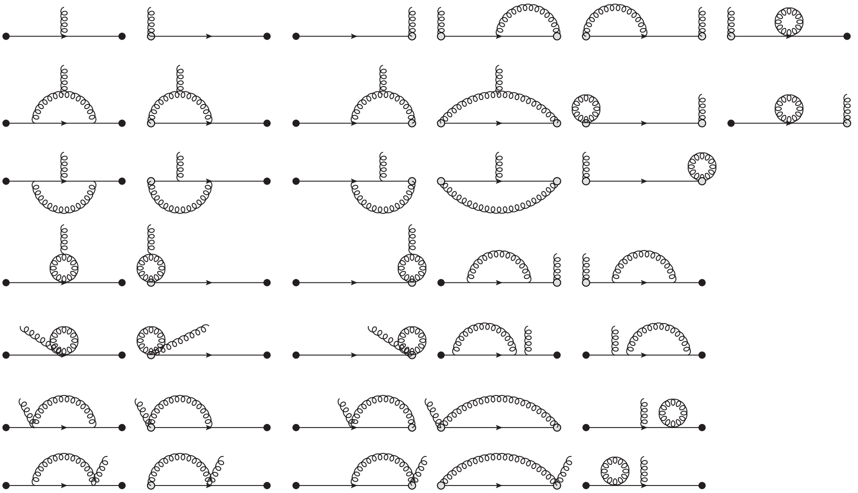
JULY 29 – AUGUST 03 2013
MAINZ, GERMANY

Jefferson Lab

Motivation

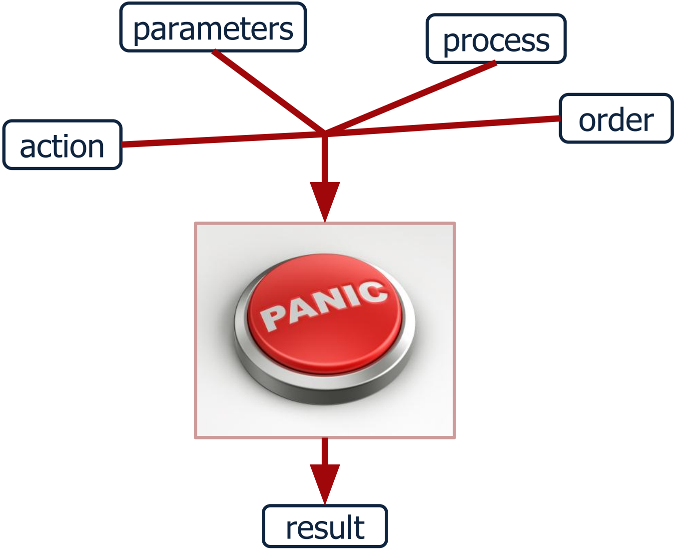


Automation: Motivation

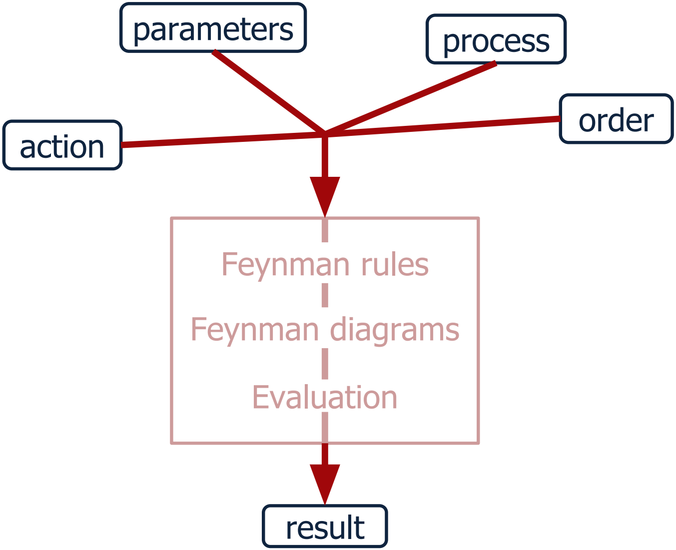


C. Lehner: Tues. 14:40, Parallel 3C

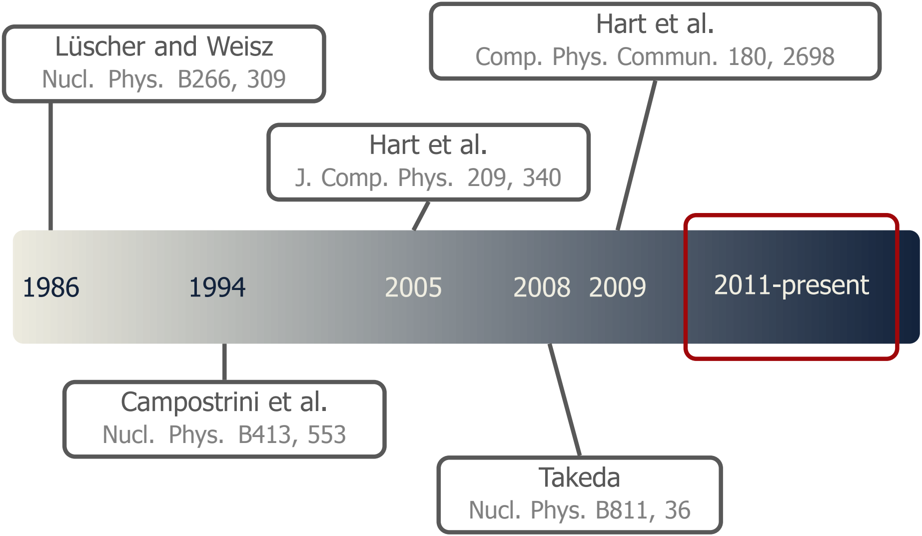
Automation: An Ideal



Automation: An Ideal



Automation: A Short History



2011-present

- HiPPy/HPsrc
- Pastor
- PhySyHCAl

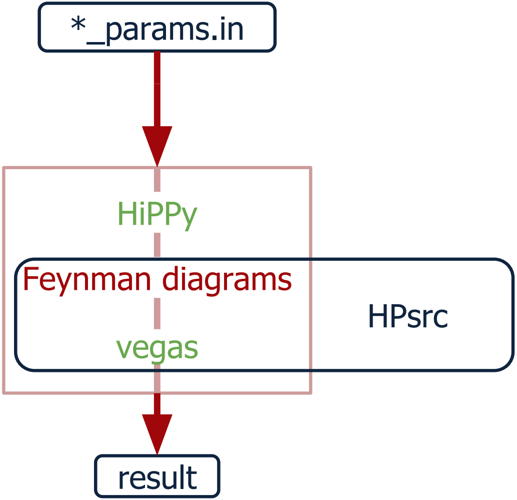
- Mathematica-based approaches
- Numerical stochastic perturbation theory

HiPPy/HPsrc

- Two stage procedure
- Generation of vertices via `HiPPy`
- Generation and evaluation of Feynman diagrams in `HPsrc`
- Widely tested on a range of perturbative calculations

Wilson & clover quarks
Asqtad & HISQ
NRQCD
Background field gauge

[Hart, Horgan and von Hippel, Comp. Phys. Comm. 180 (2009) 2698]
[Hart et al., J. Comp. Phys. 209 (2005) 340]


*_params.in
HiPPy
Feynman diagrams
HPsrc
vegas
result

Example

Improving nonrelativistic QCD (NRQCD)

$$\mathcal{S}_{\mathrm{NRQCD}} = \sum_{x,t} Q^\dagger(x,t)\left[Q(x,t) - K(t)Q(x,t-a)\right]$$

$$K(t) = \left(1 - \frac{a\delta H}{2}\right)\left(1 - \frac{aH_0}{2n}\right)^n U_4^\dagger \left(1 - \frac{aH_0}{2n}\right)^n \left(1 - \frac{a\delta H}{2}\right)$$

Precision physics requires $\mathcal{O}(\alpha_s)$ improvement

calculated in Dowdall et al., Phys. Rev. D 85 (2012) 054509

$$a\delta H = -c_1 \frac{(\Delta^{(2)})^2}{8(am_0)^3} + c_5 \frac{a^2 \Delta^{(4)}}{24 am_0} - c_6 \frac{a(\Delta^{(2)})^2}{16n(am_0)^2} - c_3 \frac{g}{8(am_0)^2} \sigma \cdot (\tilde{\nabla} \times \tilde{\mathbf{E}} - \tilde{\mathbf{E}} \times \tilde{\nabla})$$
$$+ c_2 \frac{ig}{8(am_0)^2} (\nabla \cdot \tilde{\mathbf{E}} - \tilde{\mathbf{E}} \cdot \nabla) - c_4 \frac{g}{2am_0} \sigma \cdot \tilde{\mathbf{B}}$$

Hammant et al. arXiv:1303.3234

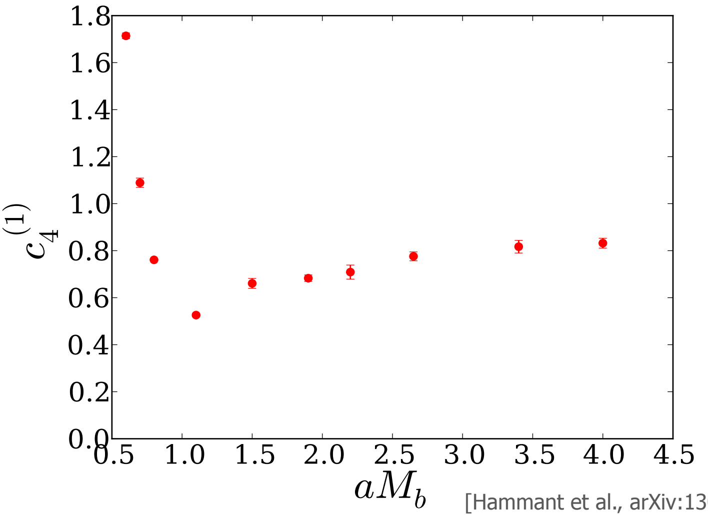
Background Field Gauge

Well established tool with nice properties for QCD:
- Ward identities constrain renormalisation parameters
- residual gauge invariance constrains operators

Indispensable in determination of c_2, c_4 for NRQCD:
- ensures only gauge-covariant $D > 4$ operators appear
- no UV logarithms in coefficients of effective action operators
- render 1PI vertex function ultraviolet finite
- enables us to match NRQCD to QCD using different regulators

Within `HiPPy/HPsrc` background field gauge requires:
- distinguishing background field gluons from quantum gluons
- ordering background field gluons and quantum gluons
- new vertex symmetrisation

Sample Results: $c_4^{(1)}$



[Hammant et al., arXiv:1303.3234]

2011-present

- HiPPy/HPsrc
- Pastor
- PhySyHCAl

- Mathematica-based approaches
- Numerical stochastic perturbation theory

Pastor

- Optimised for the Schrödinger functional

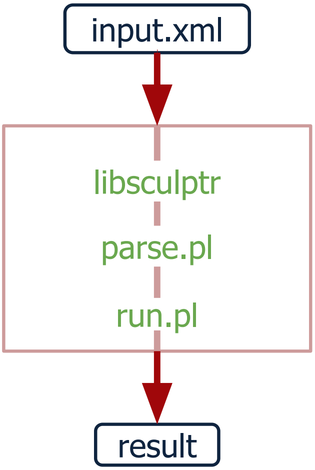
Wilson & clover actions
HQET: static & $\mathcal{O}(1/m_Q)$

- Automated generation of vertices using `C++` routines

- Automated generation of Feynman diagrams in `Python`

[Hesse, Sommer and von Hippel, PoS(Latt. 2011) 229]
[Hesse and Sommer, 1211.0866]

P. Korcyl: Tues. 15:00, Parallel 3C


input.xml
libsculptr
parse.pl
run.pl
result

Tuning HQET for B physics

$$\mathcal{L}_{\mathrm{HQET}} = \mathcal{L}_{\mathrm{stat}} - \left(\omega_{\mathrm{kin}}\mathcal{L}_{\mathrm{kin}} + \omega_{\mathrm{spin}}\mathcal{L}_{\mathrm{spin}}\right) + \mathcal{O}(1/m_b^2)$$

$$\langle O\rangle_{\mathrm{HQET}} = \langle O\rangle_{\mathrm{stat}} + \omega_{\mathrm{kin}} \sum_x \langle O\mathcal{L}_{\mathrm{kin}}(x)\rangle_{\mathrm{stat}} + \omega_{\mathrm{spin}} \sum_x \langle O\mathcal{L}_{\mathrm{spin}}(x)\rangle_{\mathrm{stat}}$$

precision physics requires matching at $\mathcal{O}(1/m_b)$

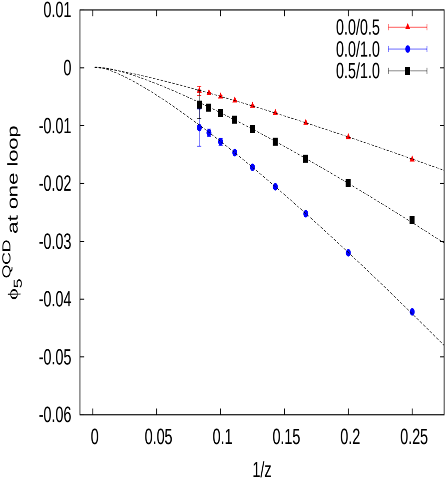
19 free parameters at this order

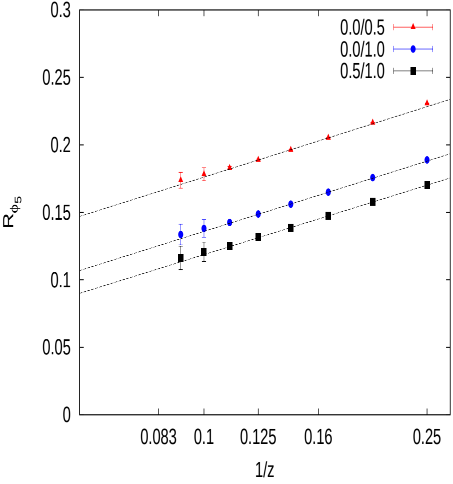
Observables must be precisely computable in QCD and HQET

Perturbative expansion ambiguous $\Rightarrow$ nonperturbative matching required

automated lattice perturbation theory used to test quality of observables

Sample results: ϕ_5





P. Korcyl: Tues. 15:00, Parallel 3C

2011-present

- `HiPPy/HPsrc`
- `Pastor`
- `PhySyHCAl`
- `Mathematica`-based approaches
- Numerical stochastic perturbation theory

PhySyHCAl

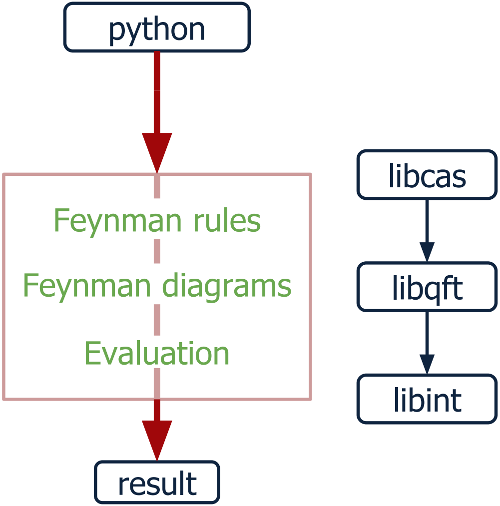
Computer Algebra System (CAS):
software package for symbolic manipulation of mathematical expressions

- Analogous to FORM
- Optimised for lattice perturbation theory
- CAS implemented as C++ library
- Unified lattice and continuum perturbation theory framework on top
- Documentation at www.lhnr.de/physyhcal

Wilson & RHQ actions
Domain wall fermions
Schrödinger functional

[Lehner, 1211.4013]

C. Lehner: Tues. 14:40, Parallel 3C


python
Feynman rules
Feynman diagrams
Evaluation
result
libcas
libqft
libint

Tuning relativistic heavy quarks in the Columbia formulation

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

can be tuned to remove $\mathcal{O}(a\vec{p})$ discretisation errors in on-shell quantities

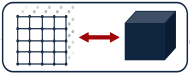
$$Q'(x) = Q(x) + d_1 \sum_{i=1}^{3}\gamma_i D_i Q(x)$$

tune to match to continuum

RHQ Matching Conditions

Quark bilinear for onshell momenta

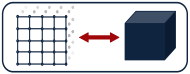
$$S(p) = \sum_q \langle Q'(p)\overline{Q}'(q) \rangle$$



- pole mass $\rightarrow m_0$
- dispersion relation $\rightarrow \zeta$
- spinor structure $\rightarrow d_1$

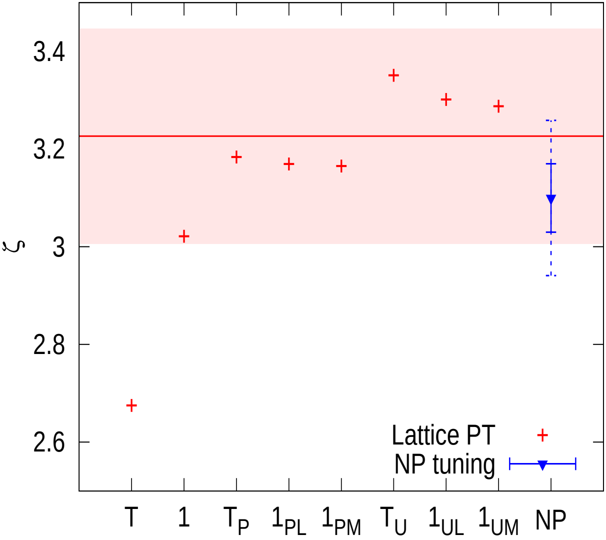
Three-point function in the onshell limit

$$\Lambda^a_\mu(p,q) = \sum_k \langle Q'(q) A^a_\mu(k) \overline{Q}' \rangle$$



c_P

Sample results: ζ



Tree level

1 loop

Plaquette improvement

U: Landau gauge link improvement

Lattice coupling expansion

$\overline{\mathrm{MS}}$ coupling expansion

NonPerturbative tuning

Lehner, 1211.4013

2011-present

- HiPPy/HPsrc
- Pastor
- PhySyHCAl

- Mathematica-based approaches
- Numerical stochastic perturbation theory

Mathematica-based Approaches

Feynman diagrams handled algebraically

Divergent integrals reduced to miminal set

Basis integrals evaluated numerically via `FORTRAN` routines

Various actions:

- Wilson and clover actions
- Twisted mass fermions
- Staggered fermions

Calculations with Wilson-like quarks carried out to $\mathcal{O}(a^2)$

[Constantinou et al., JHEP 10 (2009) 064]

[Constantinou et al., PRD 83 (2011) 074503]

[Alexandrou et al., PRD 86 (2012) 014505]

[Constantinou, Costa and Panagopoulos, 1305.1870]

[Constantinou et al., PRD 87 (2013) 096019]

Mathematica at Lattice 2013

- H. Perlt: Wed. 09:50, Parallel 5C
- M. Constantinou: Tues. 15:00, Parallel 3B
- H. Panagopoulos: Fri. 15:20, Parallel 9C
- M. Costa: “Perturbative renormalization functions”

Example 1

Chromomagnetic operator renormalisation for twisted mass fermions

$$g\overline{q}_s \sigma_{\mu\nu} G^{\mu\nu} q_d$$

Occurs in the $\Delta S = 1$ effective Hamiltonian relevant for

- $K^0 - \overline{K}^0$ mixing
- ϵ'/ϵ and $\Delta I = 1/2$
- $K \to 3\pi$

Potentially mixes with ~ 40 lattice operators of dimension $D \leq 5$

Symmetries of twisted mass action constrain operators to just 13

Mixing matrix calculated at one-loop

H. Panagopoulos: Fri. 15:20, Parallel 9C

EXAMPLE 2

Renormalisation of twisted mass fermion bilinears

$$\mathcal{O}^a_\chi = \overline{\chi} O^a \chi \qquad O^a \in \{\tau^a, \gamma_5\tau^a, \gamma_\mu\tau^a, \gamma_5\gamma_\mu\tau^a, \sigma_{\mu\nu}\tau^a, \gamma_5\sigma_{\mu\nu}\tau^a\}$$

$\psi = e^{i\pi\gamma_5\tau^3/4}\chi$ and $\overline{\psi} = \overline{\chi}\, e^{i\pi\gamma_5\tau^3/4}$

Renormalisation prescription, at critical mass and vanishing twisted mass:

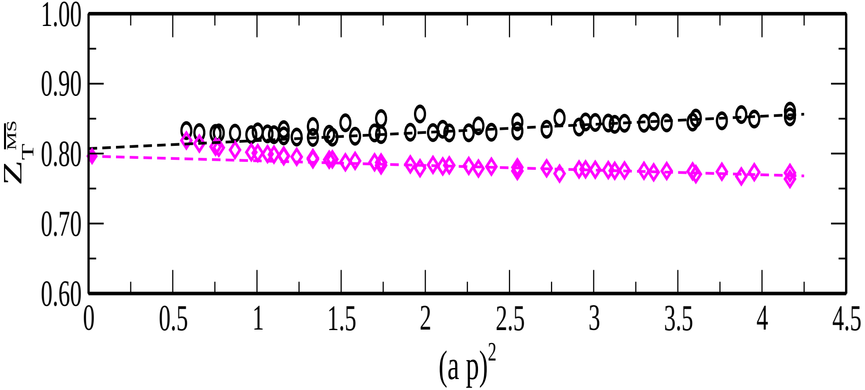
$$Z_{\mathcal{O}} = \left. \frac{\mathrm{Tr}\left\{ \left(S^L(p)\right)^{-1} S^{(0)}(p) \right\}}{\mathrm{Tr}\left\{ \Gamma^L(p) \left(\Gamma^{(0)}(p)\right)^{-1} \right\}} \right|_{p^2=\mu^2}$$

- $S^{(0)}$ and $\Gamma^{(0)}$ are tree-level propagator and fermion operators
- S^L and Γ^L correspond to perturbative and nonperturbative results

$Z_{\mathcal{O}}$ determined perturbatively and nonperturbatively

Perturbative results correct $\mathcal{O}(a^2p^2)$ errors in nonperturbative results

Sample results: $Z_T^{\overline{MS}}$



M. Constantinou: Tues. 15:00, Parallel 3B

2011-present

- HiPPy/HPsrc
- Pastor
- PhySyHCAl
- Mathematica-based approaches
- Numerical stochastic perturbation theory

NSPT

Established technique for higher order perturbative calculations

Di Renzo et al., NPB(PS) 34 (1994) 795

Di Renzo et al., NPB 426 (1994) 675

Stochastic quantisation:

- Fields evolve in fictitious time according to a Langevin equation
- Stochastic perturbation theory converges to standard expansion

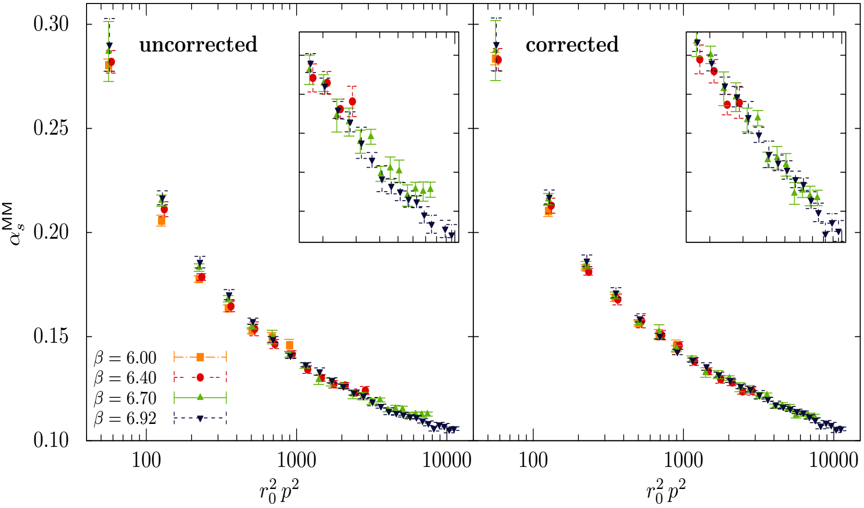
NSPT: the numerical application of stochastic perturbation theory

NSPT reviewed in, e.g., Di Renzo and Scorzato, JHEP 0410 (2004) 073

NSPT at Lattice 2013

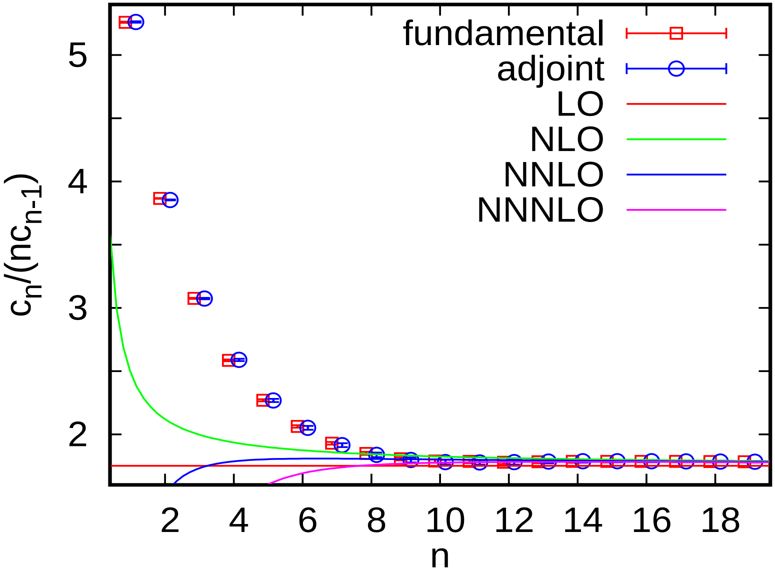
- D. Hesse: Mon. 14:20, Parallel 1B
- M. Dalla Brida: Mon. 14:40, Parallel 1B
- F. Di Renzo: Mon. 15:20, Parallel 1C
- A. Pineda: Mon. 17:10, Parallel 2E
- M. Brambilla: Fri. 18:10, Parallel 10G
- J. Simeth: "Quantifying discretization errors for the gluon and ghost propagators using stochastic perturbation theory"

Sample Results: MiniMom α_s



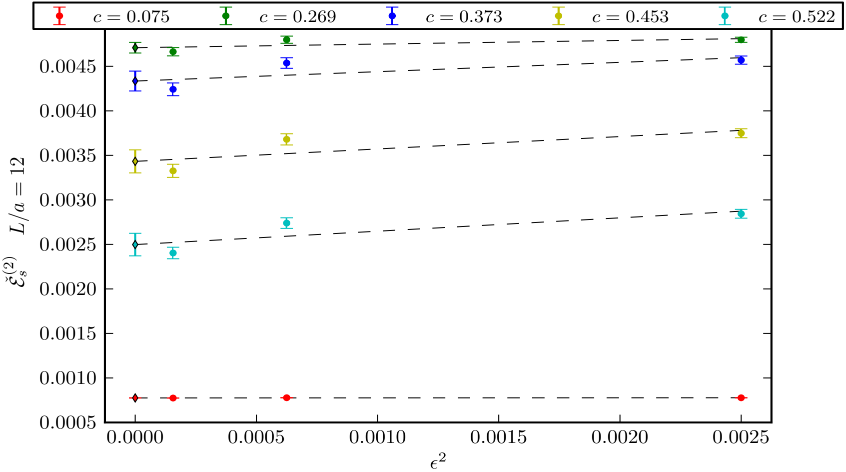
Simeth and Sternbeck: Poster

Sample Results: Static Source Selfenergy



[Plot simplified from Bali et al., PRD 87 (2013) 094517]

Sample Results: Gradient Flow



M. Dalla Brida: Mon. 14:40, Parallel 1B

Summary

2011-present

- HiPPy/HPsrc
- Pastor
- PhySyHCAl

- Mathematica-based approaches
- Numerical stochastic perturbation theory

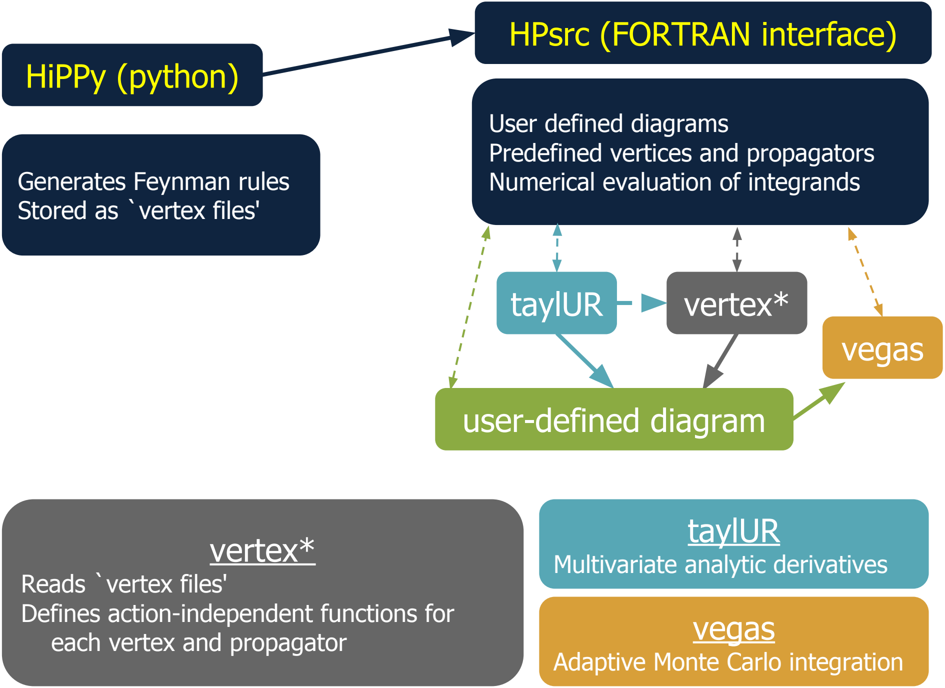
Thank you

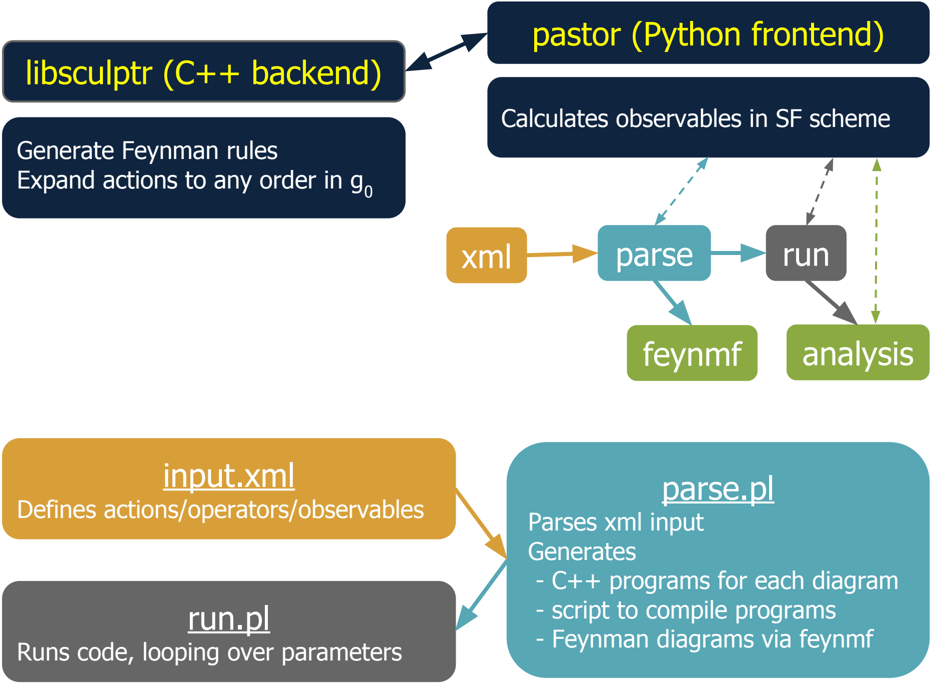
LATTICE 13

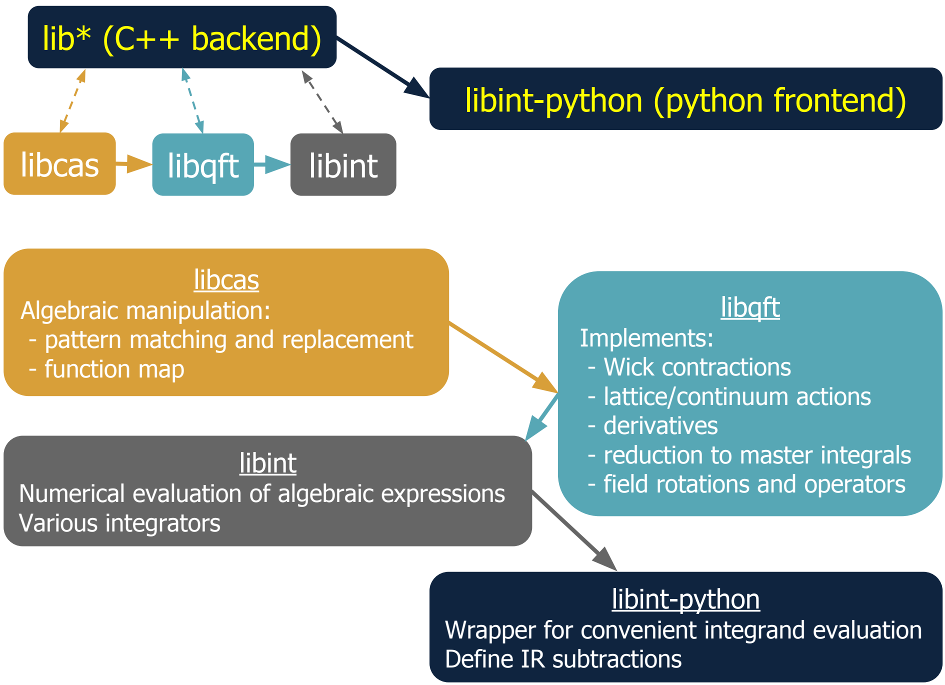
JULY 29 – AUGUST 03 2013
MAINZ, GERMANY

Chris Monahan
cjmonahan@wm.edu

www.latticeperturbationtheory.org


HiPPy (python)
HPsrc (FORTRAN interface)
Generates Feynman rules
Stored as `vertex files'
User defined diagrams
Predefined vertices and propagators
Numerical evaluation of integrands
taylUR
vertex*
vegas
user-defined diagram
vertex*
Reads `vertex files'
Defines action-independent functions for each vertex and propagator
taylUR
Multivariate analytic derivatives
vegas
Adaptive Monte Carlo integration


libsculptr (C++ backend)
pastor (Python frontend)
Generate Feynman rules
Expand actions to any order in g0
Calculates observables in SF scheme
xml
parse
run
feynmf
analysis
input.xml
Defines actions/operators/observables
run.pl
Runs code, looping over parameters
parse.pl
Parses xml input
Generates
- C++ programs for each diagram
- script to compile programs
- Feynman diagrams via feynmf


lib* (C++ backend)
libint-python (python frontend)
libcas
libqft
libint
libcas
Algebraic manipulation:
- pattern matching and replacement
- function map
libqft
Implements:
- Wick contractions
- lattice/continuum actions
- derivatives
- reduction to master integrals
- field rotations and operators
libint
Numerical evaluation of algebraic expressions
Various integrators
libint-python
Wrapper for convenient integrand evaluation
Define IR subtractions