Wick Contractions for Nucleon-Nucleon Scattering and Matrix Elements

E. Berkowitz[1], T. Kurth[2], and M. Strother[2]

[1]Lawrence Livermore National Laboratory, [2]Lawrence Berkeley National Laboratory

Introduction

- A Wick contraction is a way to tie creation and annihilation operators into propagators. Each contraction corresponds to a Feynman diagram.
- The proton-proton scattering by the $\Delta I = 2$ weak operator insertion has 4896 contractions. These reduce to 2208 contractions for degenerate up- and down-quark masses.
- Each contraction comes with a sign. These signs are
  - important for the physics: the signs control the interference between the diagrams.
  - tricky to keep track of by hand: they sensitively depend on operator ordering.
  - absolutely critical to get right: a single error casts the whole result into doubt.

Baryon Block Formalism

- Once we have done the Wick contractions, we are left with a number of quark propagators whose indices need to be tied up appropriately.
- We can take advantage of some physics knowledge to simplify and speed up the contractions. For example, we know that baryons are color singlets. This lets us build a baryon block,

$$BB^{\mu}(x_f, \{x_{i1}, x_{i2}, x_{i3}\})^{\beta\delta\phi}_{bdf} = \sum_{\substack{\alpha\gamma \\ ace}} \epsilon^{ace}\Gamma_{\alpha\gamma} S(x_f, x_{i1})^{\alpha\beta}_{ab} S(x_f, x_{i2})^{\gamma\delta}_{cd} S(x_f, x_{i2})^{\mu\phi}_{ef}$$

  which is color neutral at the sink.
- We can factor every contraction into baryon blocks, possibly some left-over quark propagators, and constant, sparse tensors (which we can pre-compute). See below for a simple example.
- All nucleon-nucleon partial-wave scattering can ultimately be reduced to a handful of sums like

$$\sum_{i,j} BB^{i}(x+\Delta x) BB^{j}(x) T_{ij}$$

  where i, j are collective indices, and T is the tensor which tells us how to connect them.

Sources

- In order to measure scattering in particular partial waves, we need sources with good overlap with the corresponding initial spatial wavefunction.
- For p-wave (and higher-order) scattering, this requires sources where the baryons are spatially separated.
- We put each baryon on a single lattice site, but displace the sites relative to one another.

$$\bar{N}(x+\Delta x, t)\bar{N}(x, t)$$

  - $\Delta x = 0$ has good overlap with the s-wave (and some to $L = 4$ and higher) but none with the p-wave
  - $\Delta x \propto \pm\hat{x}, \pm\hat{y}, \pm\hat{z}$ allows us to construct in post-processing
    - nonlocal sources with good s-wave overlap (via the isotropic combination)
    - nonlocal sources with good p-wave overlap (via antisymmetric combinations)

Sinks

- Similarly, we need sinks with good overlap with the corresponding final wavefunction.
- For partial-wave scattering, we store the full momentum dependence of each correlator. This makes it cheap to try any desired sink, because we can pull out particular displacements

$$\sum_x N(x+\Delta x, t) N(x, t) \propto \sum_p e^{-ip\,\Delta x}\tilde{N}(p, t)\tilde{N}(-p, t)$$

  or fold in any spatial wavefunction.
- For nucleon-nucleon matrix elements, this approach is prohibitively expensive, and so we fix our sinks in the same way we fix our sources.

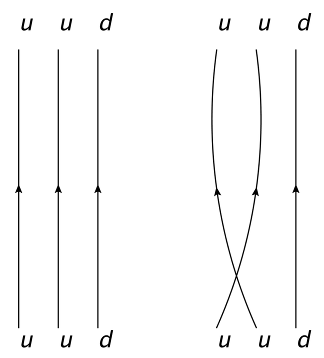
Operator Insertion

- The parity-violating process we are interested in conserves momentum, and thus our operator must be projected to zero momentum.
- This can be accomplished
  - Stochastically (allowing us to use arbitrary sinks), at the cost of increased noise.
  - Exactly (which requires a sum over the lattice volume). This is feasible with fixed sinks.

Automatic Contraction Generation

- It is impractical to write code for 2208 contractions by hand.
- If calculated by hand, the chance of 2208 correct signs would be low.
- We need software to help write the correct code.
- A good contraction generator would make this tedious task mindless.
- It should:
  - Parse familiar physics expressions,
  - Be easy to extend,
  - Algorithmically generate all Wick contractions,
  - Generate code for a variety of lattice QCD software.

Generation in Mathematica



- Mathematica makes it easy to parse and manipulate symbolic expressions, and is interactive and fast enough to allow error correction and fast iteration.
- We have written an automatic contraction generator using Mathematica.
- Proton two-point function has two Wick contractions ↗.
  - Begin with physics expression: $\langle\, p^{\mu}(x_f) \mid p_{\mu}(x_i)\,\rangle$
  - Translate into simple mathematica expression:

```
pp = Proton[sink][xf, μ] ** bar[Proton[source][xi, ν]] ** SpinProjector[Same, Spin[ν, μ]];
```

  - Symbolically generate contractions:

```
contractions = Contract[pp];
Notation[contractions]
```

$$\mathrm{Same}_{\nu\mu}\, C\gamma_{5\,\nu''\rho''}\, C\gamma_{5\,\nu'\rho'}\, S[\mathrm{down}, xf - xi]^{b'b''}_{\nu'\nu''}\, S[\mathrm{up}, xf - xi]^{a'c''}_{\mu\rho''}\, S[\mathrm{up}, xf - xi]^{c'a''}_{\rho'\nu}\, \epsilon^{a''b''c''}\, \epsilon^{a'b'c'} -$$
$$\mathrm{Same}_{\nu\mu}\, C\gamma_{5\,\nu''\rho''}\, C\gamma_{5\,\nu'\rho'}\, S[\mathrm{down}, xf - xi]^{b'b''}_{\nu'\nu''}\, S[\mathrm{up}, xf - xi]^{a'a''}_{\mu\nu}\, S[\mathrm{up}, xf - xi]^{c'c''}_{\rho'\rho''}\, \epsilon^{a''b''c''}\, \epsilon^{a'b'c'}$$

  - Take generated contractions and generate code
    - using QDP primitives

```
Generate[QDP][contractions]

 -colorTrace(spinTrace(S_up_xf_from_xi * projectorSame) *
spinTrace(quarkContract24(S_up_xf_from_xi, S_down_xf_from_xi * gammaC5) * gammaC5))
-trace(gammaC5 * S_up_xf_from_xi * projectorSame *
quarkContract24(S_up_xf_from_xi, S_down_xf_from_xi * gammaC5))
```

    - for creating tensors for the baryon block formalism

```
Generate[BaryonBlocks][contractions] // Notation

for_color(b){
for_color(d){
for_color(f){
    for_spin(epsilon){
    for_spin(   beta){
    for_spin(  delta){
    for_spin(    phi){
        temp  = 0;
        temp -= epsilon3(b, d, f) * diquark[beta][phi] * spin_projector_Same[delta][epsilon];
        temp -= epsilon3(b, d, f) * diquark[beta][delta] * spin_projector_Same[phi][epsilon];
        tensor_singleProton.add(temp, tensor_index(b, d, f, epsilon, beta, delta, phi));
    }}}}
}}}
```

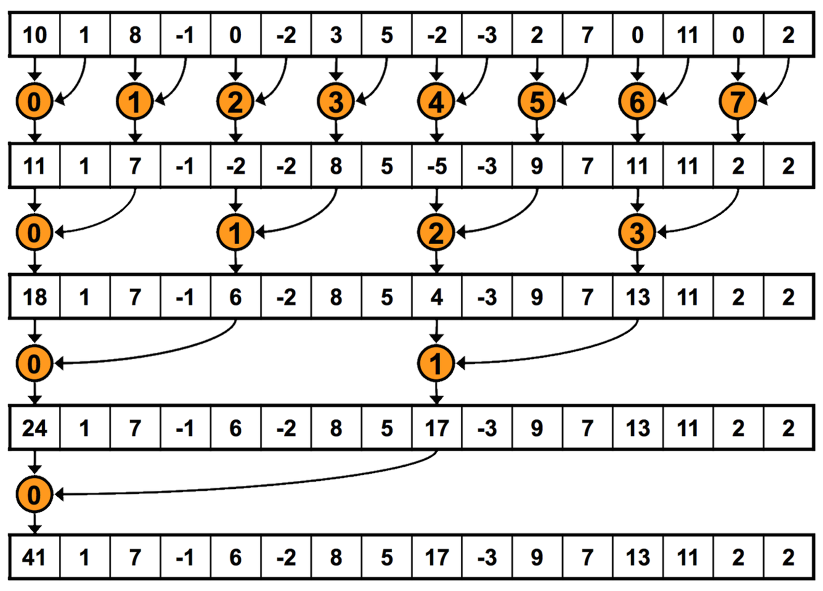
$$-\left(\mathrm{Same}_{\phi\epsilon}\, C\gamma_{5\,\beta\delta} + \mathrm{Same}_{\delta\epsilon}\, C\gamma_{5\,\beta\phi}\right) BB^{\epsilon}[xf, \{xi, xi, xi\}]^{\beta\delta\phi}_{bdf}\, \epsilon^{bdf}$$

Using CUDA

- The two-baryon contraction problem can be viewed as a calculation of the dense-sparse-dense bilinear product

$$BB(x+\Delta x)^{T} \cdot T \cdot BB(x)$$

- using indirect addressing, we have to compute for all sites x a product such as `v[ I1[k] ] * T[k] * w[ I2[k] ]` and sum over `k`.



- compute `I1`, `I2` once and store them along with `v`, `w` and `T` in global GPU memory
- read `v[I1[k]]`, `w[I2[k]]` and `T[k]` from global memory and compute the product, one `k` per thread
- perform efficient binary reduction
- for each x, create separate CUDA stream to maximize concurrency
- host thread concurrently computes all FFTs and baryon blocks relevant for next step of the calculation

Acknowledgements

- Support for this work was provided through Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Nuclear Physics.
- Prepared by LLNL under Contract DE-AC52-07NA27344.
- LLNL-POST-657635