PROCEEDINGS OF SCIENCE

Novel Algorithms for Computing Correlation Functions of Nuclei

Nabil Humphrey,[a,*] William Detmold,[b,c] Ross D. Young[a] and James M. Zanotti[a]

[a] *CSSM, Department of Physics, University of Adelaide,*
Adelaide SA 5005, Australia

[b] *Center for Theoretical Physics,*
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA

[c] *The NSF AI Institute for Artificial Intelligence and Fundamental Interactions*

E-mail: nabil.humphrey@adelaide.edu.au

The computational cost required to calculate nuclear correlation functions grows factorially in the number of quarks, making the study of large nuclei inaccessible to *ab initio* study using lattice QCD at the present time. However, the tensor expressions corresponding to many of these correlation functions exhibit a high degree of permutation symmetry that can be exploited to reduce computational work. We present promising speed-ups for certain choices of interpolating operators using two new algorithms for computing nuclear correlation functions.

MIT-CTP/5381
ADP-21-24/T1171

The 38th International Symposium on Lattice Field Theory, LATTICE2021 26th-30th July, 2021
Zoom/Gather@Massachusetts Institute of Technology

[*] Speaker

1. Introduction

This work will explore the use of lattice QCD in the *ab initio* study of the properties and interactions of multi-hadron systems directly from QCD. There are deep physical and numerical challenges in this regard, including a proliferation of possibilities for interpolating operators that couple to low energy states, factorial scaling of contraction complexity in general, signal-to-noise scaling [1–3], and potentially dominating floating-point precision errors [4]. The focus of this work is the computational resource scaling associated with numerically evaluating correlation functions built from interpolating operators possessing the quantum numbers of a multi-hadron system. For a given set of quantum numbers, a two-point correlation function takes the form:

$$C = \langle \mathcal{O}'\mathcal{O}^\dagger \rangle = \sum_{(\vec{a},\vec{a}')\in\mathcal{I}} W^{a_1,\dots,a_{n_q}}_{a'_1,\dots,a'_{n_q}} \left\langle q(a_1)\dots q(a_{n_q})\overline{q}(a'_1)\dots\overline{q}(a'_{n_q})\right\rangle, \tag{1}$$

where $\vec{a} = (a_1,\dots,a_{n_q})$ combine flavour f_i, colour indices c_i, spinor indices α_i, and spacetime indices x_i of quark fields $q(a_i) \equiv q^{f_i,c_i}_{\alpha_i}(x_i)$; $W^{a_1,\dots,a_{n_q}}_{a'_1,\dots,a'_{n_q}}$ combines tensors in $\mathcal{O}$ which project onto the correct set of quantum numbers; and $n_q = n_u + n_d$, where n_u (n_d) denotes the number of up (down) quark fields in operator $\mathcal{O}$. The set of values over which a_i, a'_j range, given by $\mathcal{I} = \{(\vec{a},\vec{a}') \mid W^{\vec{a}}_{\vec{a}'} \neq 0\}$, is referred to as the *index set* and its cardinality $|\mathcal{I}|$ is referred to as the *index size*. The number of floating-point operations required to directly evaluate C scales as the product of the index size $|\mathcal{I}|$ and the number of quark-level Wick contractions, which is given by $n_u!n_d!$ in the case where $\mathcal{O}$ contains only quark fields rather than their adjoints.

In order to construct nuclear correlation functions, we must first construct quark-level interpolating operators with the desired set of quantum numbers. A convenient approach for multi-baryon systems is to combine local baryon interpolating operators by tying together uncontracted spinor indices. We consider the following set of single baryon operators (examples of $\mathcal{O}$):

$$\begin{aligned}
p^\alpha(x) &= \epsilon_{abc}(u^T_a(x)(C\gamma_5)d_b(x))u^\alpha_c(x) && \text{(proton)}\\
n^\alpha(x) &= \epsilon_{abc}(d^T_a(x)(C\gamma_5)u_b(x))d^\alpha_c(x) && \text{(neutron)}\\
p^\alpha_\pm(x) &= \epsilon_{abc}(u^T_a(x)(C\gamma_5 P_\pm)d_b(x))u^\alpha_c(x) && \text{(non-relativistic proton)}\\
n^\alpha_\pm(x) &= \epsilon_{abc}(d^T_a(x)(C\gamma_5 P_\pm)u_b(x))d^\alpha_c(x) && \text{(non-relativistic neutron),}
\end{aligned}$$

where $P_\pm = \frac{1}{2}(1 \pm \gamma_4)$. From single baryon operators, we may construct a selection of multi-baryon operators (examples of $\mathcal{O}$):

$$\begin{aligned}
D_I(x) &= n^T(x)(C\gamma_5)p(x) && \text{(Dinucleon I)}\\
D_{II}(x) &= \frac{1}{\sqrt{2}}\left[n^T(x)(C\gamma_5)p(x) - p^T(x)(C\gamma_5)n(x)\right] && \text{(Dinucleon II)}\\
{}^3He^j_I(x) &= p^T_-(x)(C\gamma_5)n_+(x)p^j_+(x) && \text{(Helium-3 I)}\\
{}^3He^j_{II}(x) &= \frac{1}{\sqrt{6}}\Big[p^T_-(x)(C\gamma_5)n_+(x)p^j_+(x) - p^T_+(x)(C\gamma_5)n_+(x)p^j_-(x)\\
&\quad +n^T_+(x)(C\gamma_5)p_+(x)p^j_-(x) - n^T_+(x)(C\gamma_5)p_-(x)p^j_+(x)\\
&\quad +p^T_+(x)(C\gamma_5)p_-(x)n^j_+(x) - p^T_-(x)(C\gamma_5)p_+(x)n^j_+(x)\Big] && \text{(Helium-3 II)}
\end{aligned} \tag{2}$$

(note that D_{II} and ${}^3He_{II}$ project spin and isospin quantum numbers while D_I and ${}^3He_I$ do not).

The literature explores and evaluates a number of existing algorithms to reduce the numerical work associated with multi-hadron contractions. The formulation into *hadron blocks*, as described and evaluated in § 2, has been explored in Refs. [5–8] in the context of nuclei. The construction of *Index Lists*, as described in Ref. [6], pre-computes the minimal subset of the index set of a numerically expensive tensor which has a non-vanishing contribution to the correlator. We construct and use Index Lists in our implementation of hadron blocks in § 2. The matrix determinant formulation, as devised in Ref. [5], re-casts the quark-level Wick contraction $\left\langle q(a_1)\dots q(a_{n_q})\overline{q}(a'_1)\dots\overline{q}(a'_{n_q})\right\rangle$ as the determinant of a block matrix containing quark propagator elements $S(a_i;a'_j) \equiv \underbrace{q(a_i)\overline{q}(a'_j)}$. When combined with LU factorisation [9], the asymptotic scaling of the number of operations is reduced from $O\left(n_u!n_d!\right)$ to $O\left(n_u^3 n_d^3\right)$. The index size scaling, which for light nuclei is the dominant scaling factor, is left unchanged.

2. Hadron Blocks

The formulation of multi-hadron correlation function contractions in terms of single hadron building blocks has been explored in Refs. [5–8]. A hadron block is a tensor, $f^h_{\vec{p}}(x',\vec{\xi})$, constructed from the constituent quarks of a hadron h with colour and spinor indices bundled into $\vec{\xi}$, created at a fixed source spacetime location x' and annihilated as a (momentum-projected to $\vec{p}$ and colour-contracted) hadron at the sink. In the case of a single proton block, $f^P_{\vec{p}}(x',\vec{\xi})$, we can compute the necessary Wick contractions using explicit colour and spinor indices for both the hadron block $\left(\vec{\xi} = (a',\beta',b',\gamma',c',\alpha',\alpha)\right)$ and the quark propagator $\left(S^{f,cc'}_{\alpha\alpha'}(x,x') \equiv \underbrace{q^{f,c}_{\alpha}(x)\overline{q}}{}^{f,c'}_{\alpha'}(x')\right)$ via:

$$
\begin{aligned}
f^P_{\vec{p}}(x',\vec{\xi}) &\equiv f^P_{\vec{p}}(x',a',\beta',b',\gamma',c',\alpha')_\alpha \\
&:= \left\langle \sum_x e^{-i\vec{p}\cdot\vec{x}} p_\alpha(x)\overline{u}^{a'}_{\beta'}(x')\overline{d}^{b'}_{\gamma'}(x')\overline{u}^{c'}_{\alpha'}(x') \right\rangle \\
&= \sum_x e^{-i\vec{p}\cdot\vec{x}} \epsilon^{abc}\Gamma_{\beta\gamma} \left\langle u^a_\beta(x) d^b_\gamma(x) u^c_\alpha(x) \overline{u}^{a'}_{\beta'}(x')\overline{d}^{b'}_{\gamma'}(x')\overline{u}^{c'}_{\alpha'}(x') \right\rangle \\
&= \sum_x e^{-i\vec{p}\cdot\vec{x}} \epsilon^{abc}\Gamma_{\beta\gamma} \left[S^{u,ac'}_{\beta\alpha'}(x,x') S^{u,ca'}_{\alpha\beta'}(x,x') - S^{u,aa'}_{\beta\beta'}(x,x') S^{u,cc'}_{\alpha\alpha'}(x,x') \right] S^{d,bb'}_{\gamma\gamma'}(x,x').
\end{aligned}
\tag{3}
$$

The primary advantage of this construction is that block expressions are typically re-used many times during the course of evaluating particular choices of multi-hadron correlators. The factorial number of Wick contractions in the correlator is suppressed by a factor of 2^A for A baryons, but without block expression re-use this computation cost is merely transferred to the hadron block evaluation. To measure the performance improvement associated with using hadron blocks, Figure 1 compares wall-clock time for the correlator computation (excluding propagator computation) on a 64^3 volume using the dinucleon II operator in Eq. (2) and the helium-4 operator given in Ref. [10].

Figure 1 demonstrates that hadron blocks offer a clear performance improvement, even for two-hadron systems. Note that the cost to compute the block expressions (shown in blue) remains

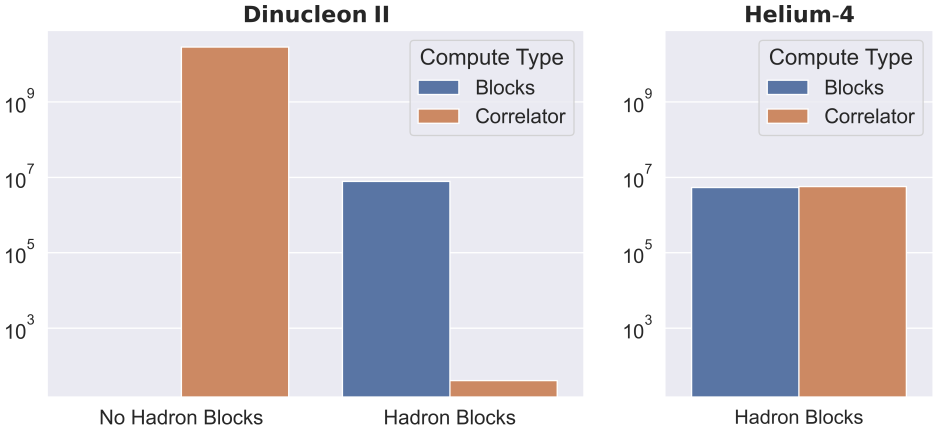


Figure 1: Hadron block benchmark, measuring wall-clock time in milliseconds on a single core of an Intel Xeon Scalable Cascade Lake processor; lattice volume 64^3. The helium-4 correlator without hadron blocks was too costly to perform here, but an upper bound on the number of floating point operations required is 10^{22}.

constant[1] in baryon number, whilst the cost to contract these blocks to form the multi-hadron correlator (shown in brown) scales factorially. The block expression cost does not benefit significantly from further symmetry-based optimisation, so the focus of the remaining sections will be the correlator cost itself (shown in brown).

3. Empirical Symmetry Properties

The following investigates some of the term-wise symmetry properties of the particular multi-baryon correlators under consideration. Under the hadron block construction, correlators of A baryons take the general form:

$$C = \sum_{k=1}^{N_w} w_k \epsilon_{a_1 b_1 c_1} \dots \epsilon_{a_A b_A c_A} \Gamma^{(1,k)}_{\alpha_1 \beta_1} \dots \Gamma^{(N_\Gamma,k)}_{\alpha_{N_\Gamma} \beta_{N_\Gamma}} f^{h_1}_{\vec{p}_{(1,k)}} \left(x'_{(1,k)}, \vec{\xi}_{(1,k)} \right) \dots f^{h_A}_{\vec{p}_{(A,k)}} \left(x'_{(A,k)}, \vec{\xi}_{(A,k)} \right), \tag{4}$$

For example, the correlator for dinucleon I in Eq. (2) has $\mathcal{O} = D_I(x)$, $A = 2$, $N_\Gamma = 4$, $N_w = 9$, $h_1 = P$, $h_2 = N$, and the $k = 1$ term for Eqn. (4) is:

$$\begin{aligned} &\epsilon_{a_1 b_1, c_1} \epsilon_{a_2 b_2 c_2} (C\gamma_5)_{\alpha_1 \beta_1} (C\gamma_5)_{\alpha_2 \beta_2} (C\gamma_5)_{\alpha_3 \beta_3} (C\gamma_5)_{\alpha_4 \beta_4} \times \\ &\qquad \times f^P_{\vec{0}}(x', a_1, \alpha_2, b_1, \beta_2, c_1, \alpha_3)_{\alpha_1} f^N_{\vec{0}}(x', a_2, \alpha_4, b_2, \beta_4, c_2, \beta_3)_{\beta_1}. \end{aligned} \tag{5}$$

[1] Hadron Block cost differs between relativistic (e.g. dinucleon II) and non-relativistic (e.g. helium-4) forms.

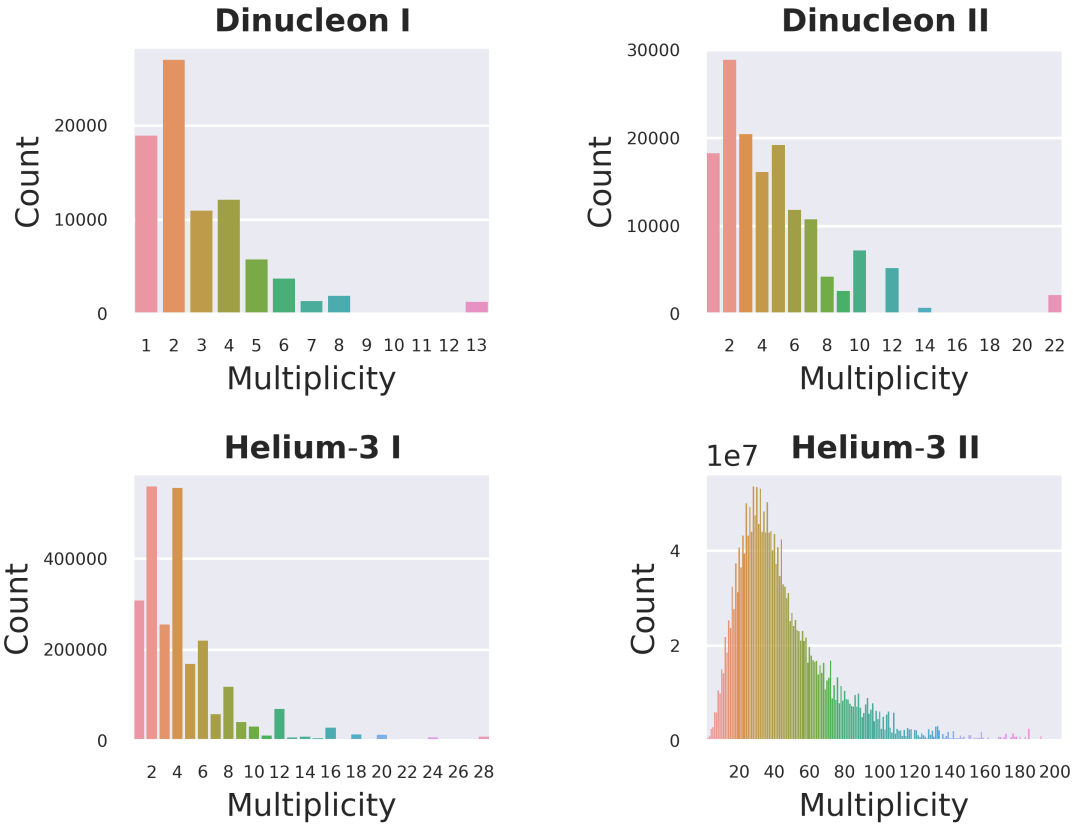


Figure 2: Multiplicity histograms for dinucleon I/II and helium-3 I/II operators in Eq. (2), where multiplicities are computed as the number of identical terms in the expansion of Eq. (4) after canonical ordering of terms.

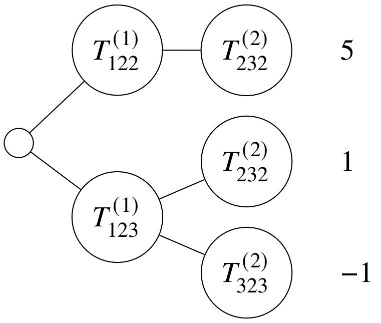
When all internal indices $\{a_i\}$, $\{b_i\}$, $\{c_i\}$, $\{\alpha_i\}$, $\{\beta_i\}$ are summed over, we are left with strings of $f^h_{\vec{q}}(\widetilde{x}\,', \vec{\eta})$ factors:

$$C = \sum_{k=1}^{\widetilde{N}_w} \widetilde{w}_k f^{h_1}_{\vec{q}_{(1,k)}} \left(\widetilde{x}'_{(1,k)}, \vec{\eta}_{(1,k)}\right) \dots f^{h_A}_{\vec{q}_{(A,k)}} \left(\widetilde{x}'_{(A,k)}, \vec{\eta}_{(A,k)}\right), \tag{6}$$

where $\vec{\eta}_{i,k}$ are bundles of fixed colour/spinor values (note the notation change $\vec{\xi} \longrightarrow \vec{\eta}$ to differentiate contracted and fixed index bundles), $\widetilde{w}_k$ are the new weights after expansion, $\widetilde{N}_w$ is the number of terms in the expanded expression for C, and $\left\{\widetilde{x}'_{i,k}\right\}$ are the source spacetime points of the expanded expression (noting that $\widetilde{x}'_{i,k} \neq x'_{i,k}$ in general). For example, the correlator for dinucleon I in Eq. (2) has the $k = 1$ term for Eqn. (6) given by:

$$f^P_{\vec{0}}(x', 1, 1, 2, 2, 3, 1)_1 f^N_{\vec{0}}(x', 1, 1, 2, 2, 3, 2)_2. \tag{7}$$

We may then canonically order the factors $f^h_{\vec{q}}(\widetilde{x}\,', \vec{\eta})$ using lexicographic order on $(h, \vec{q}, \widetilde{x}\,', \vec{\eta})$ tuples, and assign multiplicities to identical terms. Performing this process with the operators



Abstract Factor Tree

$$\longleftrightarrow \quad \begin{aligned} E &= 5\,T^{(1)}_{122}T^{(2)}_{232} + T^{(1)}_{123}T^{(2)}_{232} - T^{(1)}_{123}T^{(2)}_{323} \\ &= 5\,T^{(1)}_{122}T^{(2)}_{232} + T^{(1)}_{123}\left(T^{(2)}_{232} - T^{(2)}_{323}\right) \end{aligned}$$

Figure 3: Abstract Factor Tree example (left) with corresponding tensor expression E (right). $T^{(1)}_{ijk}, T^{(2)}_{ijk}$ are rank-3 tensors.

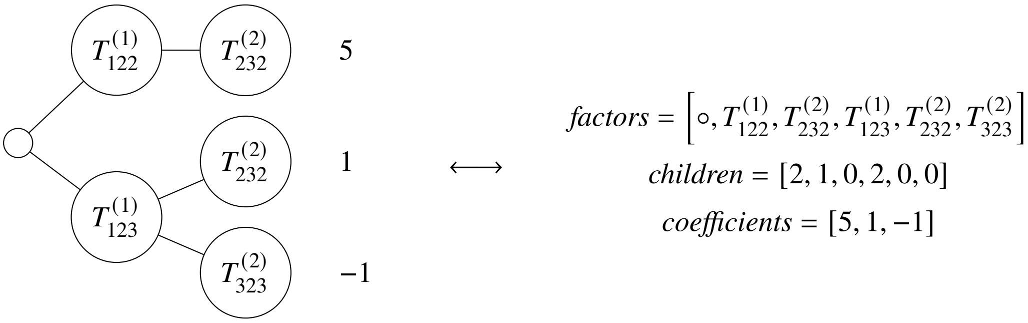
for dinucleon I/II and helium-3 I/II in Eq. (2) yields multiplicity histograms in Figure 2. The multiplicity power law relationship seen in Figure 2 suggests that significant computational savings can be made by computing each degenerate term once, adjusting the coefficients by multiplicity. This idea also motivates *factor trees* as explored in § 4.

4. Factor Trees

Starting with the observations of empirical symmetry properties, as explored in § 3, together with the property that the set of factors $\{f^h_{\vec{q}}(\widetilde{x}\,', \vec{\eta})\}$ is small compared with the set of terms in C, we can expect a high degree of both term-wise and factor-wise 'compression' in multiplicity accumulation and factorisation. To target expressions of this type we construct *factor trees*, in which the sum of all root-to-leaf paths through the tree, weighted by the leaf coefficients, is equal to the original tensor expression.

A natural way to represent an abstract factor tree that corresponds to a tensor expression E as in Figure 3 on a computer might be to represent each node as a structure and to form links between the nodes using pointers. This would be highly inefficient in two senses: first, it doesn't exploit the predictable access pattern (depth-first traversal) that evaluation of the factor trees entails and would as a result be significantly memory access bound; and second, it would have a large memory footprint, taken up almost entirely by pointers of $parent \to child$ relationships. A representation that is much more efficient on both counts is given by a *linearised factor tree*. As shown in Figure 4, a linearised factor tree is represented by three arrays: *factors*, *children*, and *coefficients*. The *factors* array is given by the depth-first traversal of the vertices of the abstract factor tree. The *children* array is given by the number of children of the vertices in depth-first traversal order. The *coefficients* array is the list of leaf coefficients in depth-first traversal order.

To measure the performance improvement associated with using factor trees, Figure 5 compares wall-clock time for correlator computation (excluding propagator and hadron block computation) using the helium-3 I/II operators as in Eq. (2). Figure 5 demonstrates that factor trees offer between


$T^{(1)}_{122}$
$T^{(2)}_{232}$
5
$T^{(2)}_{232}$
1
$T^{(1)}_{123}$
$T^{(2)}_{323}$
-1
$\longleftrightarrow$
$factors = \left[\circ, T^{(1)}_{122}, T^{(2)}_{232}, T^{(1)}_{123}, T^{(2)}_{232}, T^{(2)}_{323}\right]$
$children = [2, 1, 0, 2, 0, 0]$
$coefficients = [5, 1, -1]$

(a) Abstract Factor Tree

(b) Linearised Factor Tree

Figure 4: Relationship between an abstract factor tree (a) and the corresponding linearised factor tree (b). Example as in Figure 3.

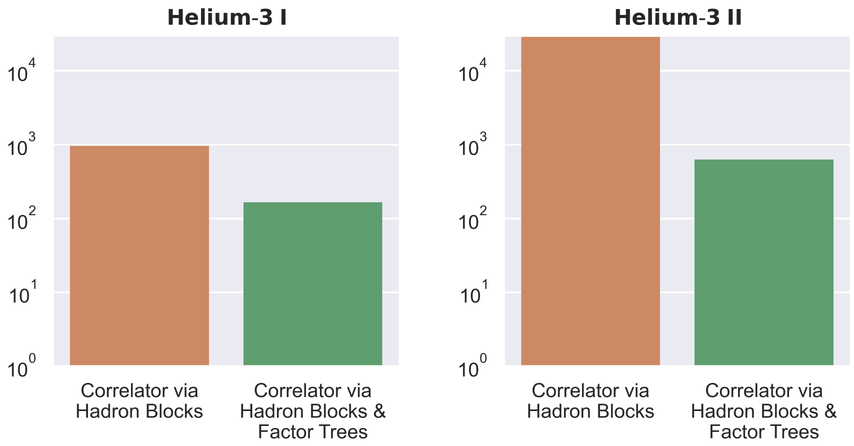


Figure 5: Factor tree benchmark, measuring wall-clock time in milliseconds on a single core of an Intel Xeon Scalable Cascade Lake processor; lattice volume 64^3.

one and two orders of magnitude improvement over unoptimised correlator computations for light nuclei using Hadron Blocks. An up-front computational cost for constructing the factor tree not included in Figure 5 is required and proportional to the unoptimised correlator computational cost. The extra calculation, however, is only required before the first configuration is analysed and hence the cost may be amortised over the full set of configurations. Although the factor tree method shows promising speed-ups for small nuclei, it should be noted that the memory used will become prohibitive for large nuclei[2].

[2] Constructing factor trees directly to hard-disk could extend the method to larger nuclei.

5. Tensor E-graphs

E-graphs (equality graphs) compactly represent and compute a congruence relation over a set of expressions defined by a *re-write rule*. E-graphs were originally developed for automated theorem provers, and have recently shown success when used in program optimisation [11]. An *e-graph* is a tuple (U, M, H) where:

- U is a union-find data structure (see Refs. [11, 12]) storing an equivalence relation over *e-class ids* (an *e-class* is a set of equivalent *e-nodes*; an *e-node* is a function symbol paired with a list of children e-classes),
- M is a map from *e-class ids* to *e-classes*; all equivalent *e-class ids* map to the same *e-class*,
- H is a hash-cons (see Ref. [11]) mapping *e-nodes* to *e-class ids*.

Here we present *tensor e-graphs*, in which each e-node is a tensor expression E and the initial e-graph is the set of tensor expressions representing the correlator C. The full tensor e-graph is built by repeatedly applying to each e-node E the re-write rule given in Figure 6 to produce candidate common subexpressions $\widetilde{E}_1$, $\widetilde{E}_2$.

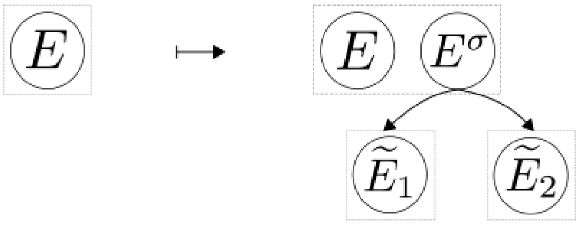


Figure 6: Tensor e-graph re-write rule for a tensor expression E, with $\sigma \in S^{\pm}_{|E|}$ (signed symmetric group[3]) permuting the index slots of E through action $E^\sigma = E_1 E_2$ for subexpressions E_1, E_2 related to child e-nodes through tensor expression canonicalisation $E_k \longrightarrow \widetilde{E}_k$ (see Ref. [13]). E-nodes are notated by circles and e-classes are notated by dashed rectangles.

In order to restrict the rather large class of possible re-writes $E \xrightarrow{\sigma} E^\sigma = E_1 E_2$, we introduce two conditions:

1. Both E_1 and E_2 must contain at least one summed index
2. At least one of E_1 or E_2 must contain at least one fully summed tensor.

For example, take $E = \epsilon_{ijk}\epsilon_{lmn}T_{ijl}T_{nmk}$ for any rank-3 tensor T and consider the sub-expressions $E_1 = \epsilon_{ijk}T_{ijl}$ and $E_2 = \epsilon_{lmn}T_{nmk}$. The canonical forms of these sub-expressions are $\widetilde{E}_1 = \epsilon_{ijk}T_{ijl} = \widetilde{E}_2$, noting that $E_2 \longrightarrow \widetilde{E}_2$ introduces a relative minus sign from the exchange of anti-symmetric indices $\epsilon_{lmn} = -\epsilon_{nml}$. The signed permutation σ keeps track of both the rearrangement of tensors $\epsilon\epsilon TT \longrightarrow \epsilon T \epsilon T$ and the relative sign induced by the tensor expression canonicalisation process. In this case, $\sigma = -(4\ 7)(5\ 8)(6\ 9)(7\ 9)$ using (signed) cycle notation. This re-write, as depicted in Figure 7 can be used to extract a common sub-expression $B_{kl} = \epsilon_{ijk}T_{ijl}$ so that by

[3] Formally, $S^{\pm}_{|E|}$ is defined by the external wreath product $\mathbb{Z}_2 \wr S_{|E|}$ for symmetric group $S_{|E|}$; see Ref. [14].

permuting the index slots of E by σ (i.e. $ijklmnijlnmk \to -ijkijlnmlnmk$) and discarding the index slots summed in B, we may express $E = -B_{kl}B_{lk}$.

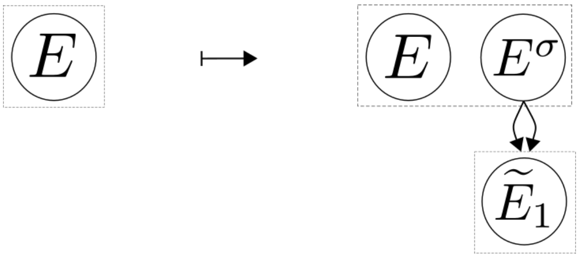


Figure 7: Tensor e-graph re-write rule for tensor expression $E = \epsilon_{ijk}\epsilon_{lmn}T_{ijl}T_{nmk}$ with a rank-3 tensor T. Common sub-expression $\widetilde{E}_1 = \epsilon_{ijk}T_{ijl}$ only appears once in the e-graph, enabling its re-use.

A tensor e-graph is constructed by repeated application of the re-write rule until the *saturation limit*, where no further non-redundant re-writes may be performed, is reached. Often, it is not computationally feasible to reach the saturation limit, and the tensor e-graph construction process terminates after a set e-node limit has been reached.

After a tensor e-graph has been constructed, the optimal sub-expression decomposition scheme (i.e. the desired computational strategy) may be extracted by making a recursive selection within each e-class of the e-node maximising an objective function. Here we use the objective function such that when evaluated on an e-node is given by the number of excess parents from distinct e-classes, multiplied by the number of floating point operations required to evaluate the tensor expression represented by that e-node. This objective function is chosen to maximise the re-use of common sub-expressions weighted by how expensive they are to compute, so as to minimise the total computational cost. The provably optimal objective function remains open.

Applying this process to the correlator for the dinucleon II operator as in Eq. (2) yields the following two example common subexpressions:

$$\mathcal{B}_1(\alpha',\beta',\gamma',\delta') = \epsilon_{abc}\epsilon_{def}\,(C\gamma_5)_{\alpha\beta}(C\gamma_5)_{\gamma\delta}f^{P}_{\alpha,\vec{0}}(\vec{0},a,\alpha',d,\beta',b,\gamma')f^{N}_{\beta,\vec{0}}(\vec{0},e,\gamma,c,\delta',f,\delta), \tag{8}$$

$$\begin{aligned}&\mathcal{B}_2(a',b',c',d',e',f')\\ &\qquad = (C\gamma_5)_{\alpha\beta}(C\gamma_5)_{\gamma\delta}(C\gamma_5)_{\sigma\rho}(C\gamma_5)_{\mu\nu}f^{P}_{\alpha,\vec{0}}(\vec{0},a',\gamma,b',\sigma,c',\rho)f^{N}_{\beta,\vec{0}}(\vec{0},d',\mu,e',\delta,f',\nu).\end{aligned} \tag{9}$$

Note that both $\mathcal{B}_1$ and $\mathcal{B}_2$ satisfy the two re-write restrictions since they both contain the fully summed tensor $(C\gamma_5)_{\alpha\beta}$, for example. The correlator for the dinucleon II operator is optimally computed by first computing $\mathcal{B}_1$ and $\mathcal{B}_2$, and then contracting the remaining indices. Future work will measure the performance improvement attained by reusing $\mathcal{B}_1$ and $\mathcal{B}_2$ in the contraction.

6. Conclusion and Outlook

In this work, we have explored a number of ways to accelerate the computation of correlation functions of multi-hadron systems in the context of lattice QCD. The construction of factor trees shows promising speed-ups for certain choices of interpolating operators of light nuclei once the tree construction cost has been amortised over several iterations, but its applicability to heavier nuclei is constrained by memory limitations. The bulk of the correlator cost for interpolating operators of light nuclei remains in the hadron block evaluation. The construction of optimal evaluation schemes through tensor e-graphs is a potential avenue for performance improvements for larger nuclei, although rigorous demonstration of this possibility is left for future work.

Acknowledgements

We would like to thank Artur Avkhadiev and Phiala Shanahan for many helpful discussions. The calculations were carried out on the NCI National Facility in Canberra, Australia (supported by the Australian Commonwealth Government) and the CSSM/HEP HPC cluster (University of Adelaide). NH is supported by an Australian Government Research Training Program (RTP) Scholarship. RDY and JMZ are supported in part by the Australian Research Council grant DP190100297. WD is supported in part by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under grant Contract Number DE-SC0011090, by the SciDAC4 award DE-SC0018121, and by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/).

References

[1] G. Parisi, *The Strategy for Computing the Hadronic Mass Spectrum*, *Phys. Rept.* **103** (1984) 203.

[2] G.P. Lepage, *The Analysis of Algorithms for Lattice Field Theory*, in *Theoretical Advanced Study Institute in Elementary Particle Physics*, 6, 1989.

[3] S.R. Beane, W. Detmold, T.C. Luu, K. Orginos, A. Parreno, M.J. Savage et al., *High Statistics Analysis using Anisotropic Clover Lattices. II. Three-Baryon Systems*, *Phys. Rev. D* **80** (2009) 074501 [0905.0466].

[4] W. Detmold, M.J. Savage, A. Torok, S.R. Beane, T.C. Luu, K. Orginos et al., *Multi-Pion States in Lattice QCD and the Charged-Pion Condensate*, *Phys. Rev. D* **78** (2008) 014507 [0803.2728].

[5] W. Detmold and K. Orginos, *Nuclear correlation functions in lattice qcd*, *Phys. Rev. D* **87** (2013) 114512.

[6] T. Doi and M.G. Endres, *Unified contraction algorithm for multi-baryon correlators on the lattice*, *Computer Physics Communications* **184** (2013) 117.

[7] J. Günther, B.C. Tóth and L. Varnhorst, *Recursive approach to determine correlation functions in multibaryon systems*, *Phys. Rev. D* **87** (2013) 094513.

[8] H. Nemura, S. Aoki, T. Doi, S. Gongyo, T. Hatsuda, Y. Ikeda et al., *A fast algorithm for lattice hyperonic potentials*, *Proceedings of the 12th International Conference on Hypernuclear and Strange Particle Physics (HYP2015)* (2017) .

[9] G. Lindfield and J. Penny, *Chapter 2 - linear equations and eigensystems*, in *Numerical Methods (Fourth Edition)*, G. Lindfield and J. Penny, eds., pp. 73–156, Academic Press (2019), DOI.

[10] T. Yamazaki, Y. Kuramashi and A. Ukawa, *Helium Nuclei in Quenched Lattice QCD*, *Phys. Rev. D* **81** (2010) 111504 [0912.1383].

[11] M. Willsey, C. Nandi, Y.R. Wang, O. Flatt, Z. Tatlock and P. Panchekha, *egg: Fast and extensible equality saturation*, *Proceedings of the ACM on Programming Languages* **5** (2021) 1–29.

[12] R.E. Tarjan, *Efficiency of a good but not linear set union algorithm*, *J. ACM* **22** (1975) 215–225.

[13] Z. Li, S. Shao and W. Liu, *Classifications and canonical forms of tensor product expressions in the presence of permutation symmetries*, 1604.06156.

[14] M. Baake, *Structure and representations of the hyperoctahedral group*, *Journal of Mathematical Physics* **25** (1984) 3171.