Towards a Unified Numerical Framework for Quantum Mechanical Simulations

C. Hinz, S. Bauch, S. Hermanns, N. Schlünzen and M. Bonitz

Institute for Theoretical Physics and Astrophysics, CAU Kiel, Germany

Bundesministerium für Bildung und Forschung

Abstract

- State-of-the-art simulations often need to utilize high-performance computing (HPC) to be feasible...
- ..., but HPC architectures are getting more complex and diverse.
- No optimization → 4 - 5 orders of magnitude slowdown
- Manual optimization might not be viable for everybody.
- We have developed the Qubus tensor framework to mitigate this problem.

Key features

- Syntax inspired by mathematical notation.
- Architecture-independent interface + auto-parallelizing optimizer and runtime
- →no need to ...
  - know about architecture specifics (SIMD, Caches, ...).
  - manually parallelize and synchronize execution.
  - worry about data distribution and communication via MPI.
- Out-of-the-box support for a diverse set of architectures (x86, CUDA, Xeon Phi, ...).

Efficiency via high-level optimizations

Math

$$\Sigma_{ij}^{<,\mathrm{2B}}(t_a,t_b) = \sum_{klmnrs} w_{ikms}\left(w_{ljrm} - w_{njrl}\right) G_{kl}^{<}(t_a,t_b) G_{mn}^{<}(t_a,t_b) G_{rs}^{>}(t_b,t_a)$$

User code

```
tensor_expr<double, 4> w = get_interaction(params); // Polymorphic function

tensor_expr<complex<double>, 4> Sigma =
   def_tensor (i, j, ta, tb) [ sum(w(i, k, m, s) * (w(l, j, r, n) - w(n, j, r, l)) *
                                   Gl(k, l, ta, tb) * Gl(m, n, ta, tb) * Gg(r, s, tb, ta), {k, l, m, n, r, s}) ];
```

Interaction tensor

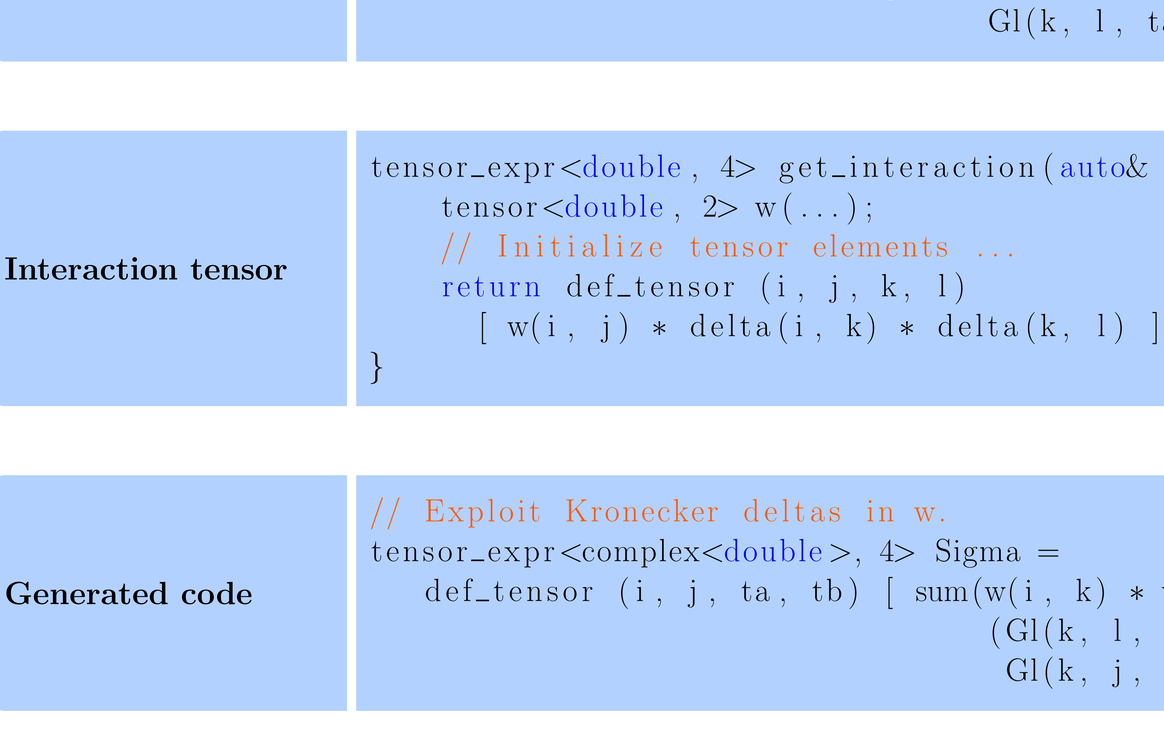
```
tensor_expr<double, 4> get_interaction(auto& params) {
    tensor<double, 2> w(...);
    // Initialize tensor elements ...
    return def_tensor (i, j, k, l)
      [ w(i, j) * delta(i, k) * delta(k, l) ];
}
```

Just one possible implementation.

Generated code

```
// Exploit Kronecker deltas in w.
tensor_expr<complex<double>, 4> Sigma =
   def_tensor (i, j, ta, tb) [ sum(w(i, k) * w(l, j) * Gg(l, k, tb, ta) *
                                   (Gl(k, l, ta, tb) * Gl(i, j, ta, tb) -
                                    Gl(k, j, ta, tb) * Gl(i, l, ta, tb)) , {k, l}) ];
```

Matrix multiplication benchmark



Matrix multiplication using Qubus

Math $C_{ij} = \sum_k A_{ik} B_{kj}$

Code

```
index i, j, k; // Declare abstract indices.

// Declare double-precision tensor variables of order 2.
tensor<double, 2> A(N, N);
tensor<double, 2> B(N, N);
tensor<double, 2> C(N, N);

C = def_tensor(i, j) [ sum(A(i, k) * B(k, j), k) ];
```

Example is fully parallelized.

Contact

Mail: hinz@theo-physik.uni-kiel.de

LaTeX TikZposter