Project B02: Status of the Kronos Library

Travis Whyte

in collaboration with

Eric B. Gregory, Bartosz Kostrzewa, Stefan Krieg, Giovanni Pederiva, Simon Schlepphorst, Aniket Sen, Estela Suarez

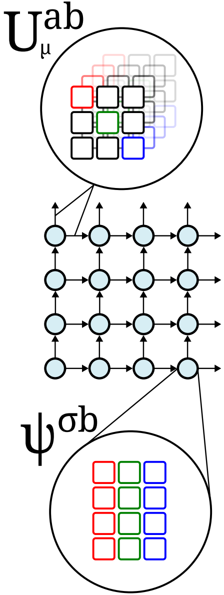
HISKP, University of Bonn & JSC, Forschungszentrum Jülich

NuMeriQs Retreat, 2025, Kloster Steinfeld, Germany

Lattice Quantum Field Theory

example: Lattice QCD

4D Cartesian Grid

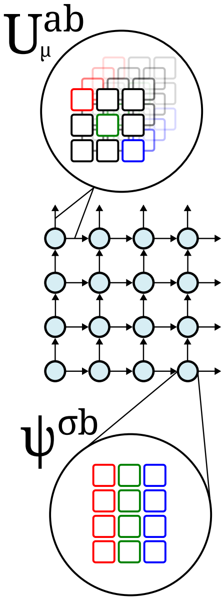


- 3×3 complex matrices $U_\mu^{ab}(x)$ "connecting" vertices
  - "gauge links" or "gauge field" (one per direction μ)

Lattice Quantum Field Theory

example: Lattice QCD

4D Cartesian Grid

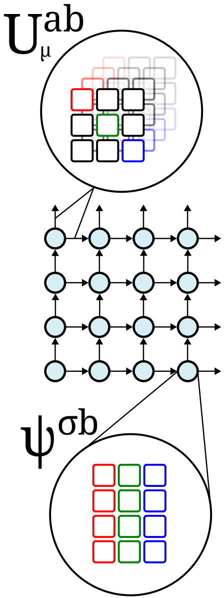


- 3×3 complex matrices $U_\mu^{ab}(x)$ “connecting” vertices
  - “gauge links” or “gauge field” (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices (“spinor field”)

Lattice Quantum Field Theory

example: Lattice QCD

4D Cartesian Grid

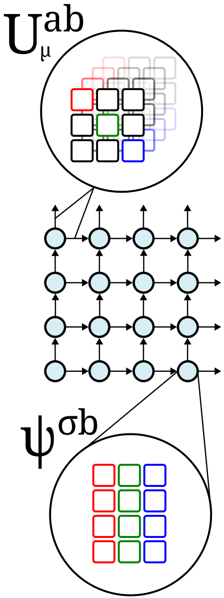


- 3×3 complex matrices $U^{ab}_{\mu}(x)$ "connecting" vertices
  - "gauge links" or "gauge field" (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices ("spinor field")
- Operations involving U and ψ

Lattice Quantum Field Theory

example: Lattice QCD

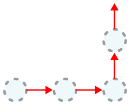
4D Cartesian Grid



- 3×3 complex matrices $U_\mu^{ab}(x)$ "connecting" vertices
  - "gauge links" or "gauge field" (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices ("spinor field")
- Operations involving U and ψ

chains of links

$$V = U_0(x) \cdot U_1(x') \cdot U_1(x'') \ldots$$

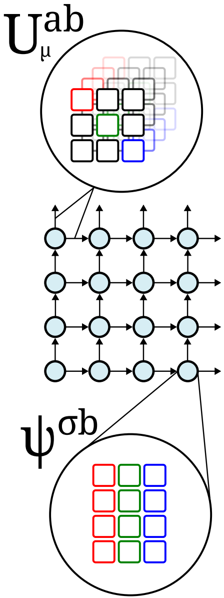


(and traces thereof)

Lattice Quantum Field Theory

example: Lattice QCD

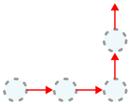
4D Cartesian Grid



- 3×3 complex matrices $U^{ab}_{\mu}(x)$ “connecting” vertices
  - “gauge links” or “gauge field” (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices (“spinor field”)
- Operations involving U and ψ

chains of links

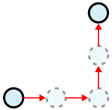
$$V = U_0(x) \cdot U_1(x') \cdot U_1(x'') \ldots$$



(and traces thereof)

Wilson lines

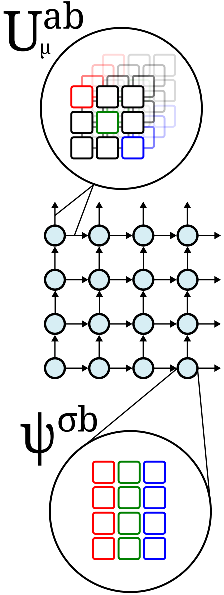
$$\bar{\psi}(x) \cdot V \cdot \psi(y)$$



Lattice Quantum Field Theory

example: Lattice QCD

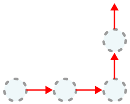
4D Cartesian Grid



- 3×3 complex matrices $U_\mu^{ab}(x)$ "connecting" vertices
  - "gauge links" or "gauge field" (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices ("spinor field")
- Operations involving U and ψ

chains of links

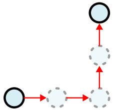
$$V = U_0(x) \cdot U_1(x') \cdot U_1(x'') \ldots$$



(and traces thereof)

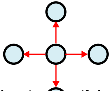
Wilson lines

$$\bar{\psi}(x) \cdot V \cdot \psi(y)$$



Finite differences

$$\psi'(x) = \psi(x) + \sum_\mu U_\mu(x)\psi(x+\hat{\mu}) - U_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu})$$

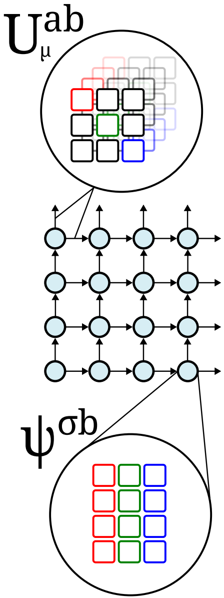


(stencil operator) (Non-local op.)

Lattice Quantum Field Theory

example: Lattice QCD

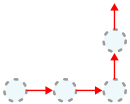
4D Cartesian Grid



- 3×3 complex matrices $U_\mu^{ab}(x)$ "connecting" vertices
  - "gauge links" or "gauge field" (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices ("spinor field")
- Operations involving U and ψ

chains of links

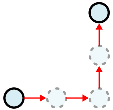
$$V = U_0(x) \cdot U_1(x') \cdot U_1(x'') \dots$$



(and traces thereof)

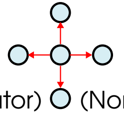
Wilson lines

$$\bar{\psi}(x) \cdot V \cdot \psi(y)$$



Finite differences

$$\psi'(x) = \psi(x) + \sum_\mu U_\mu(x)\psi(x+\hat{\mu}) - U_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu})$$

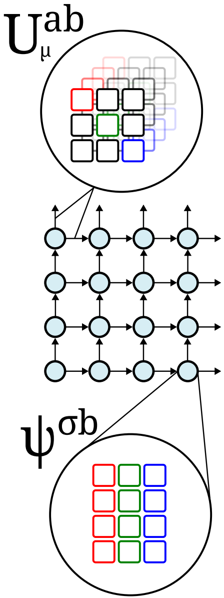


(stencil operator) (Non-local op.)

Lattice Quantum Field Theory

example: Lattice QCD

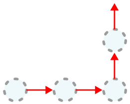
4D Cartesian Grid



- 3×3 complex matrices $U^{ab}_{\mu}(x)$ "connecting" vertices
  - "gauge links" or "gauge field" (one per direction μ)
- 4×3 complex matrices $\psi^{\sigma b}(x)$ at vertices ("spinor field")
- Operations involving U and ψ

chains of links

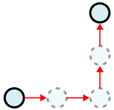
$$V = U_0(x) \cdot U_1(x') \cdot U_1(x'') \ldots$$



(and traces thereof)

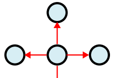
Wilson lines

$$\bar{\psi}(x) \cdot V \cdot \psi(y)$$



Finite differences

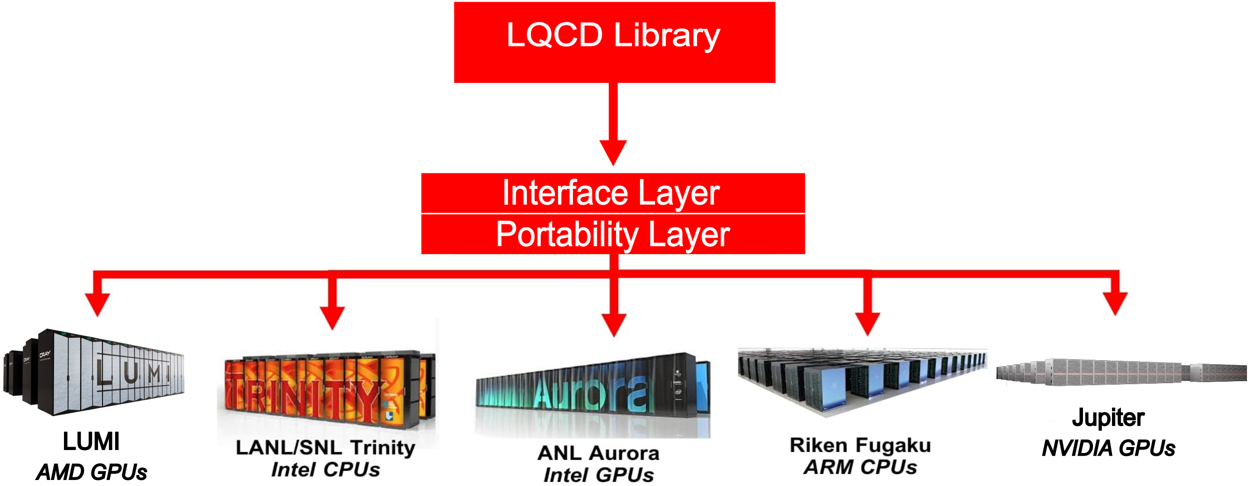
$$\psi'(x) = \psi(x) + \sum_{\mu} U_{\mu}(x)\psi(x+\hat{\mu}) - U^{\dagger}_{\mu}(x-\hat{\mu})\psi(x-\hat{\mu})$$



(stencil operator) (Non-local op.)

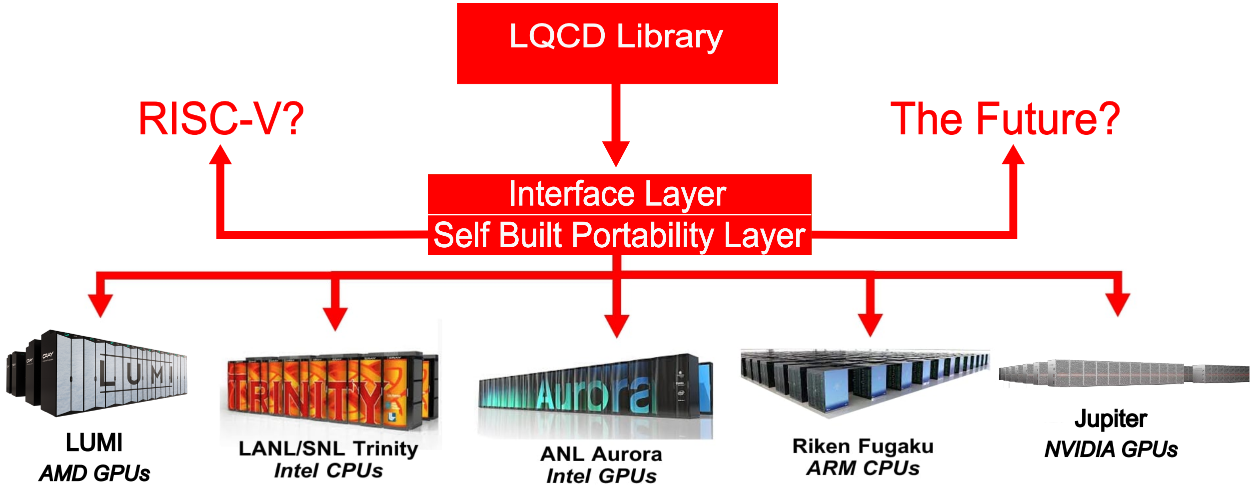
- Need to be able to easily express
  - $U \cdot U'$, $U \cdot \psi$, $U + U'$, $\psi + c \cdot \psi'$, $\langle \psi | \psi \rangle$, ...
  - loops over lattice sites and multidimensional slices thereof

Motivation for a New LQCD Library



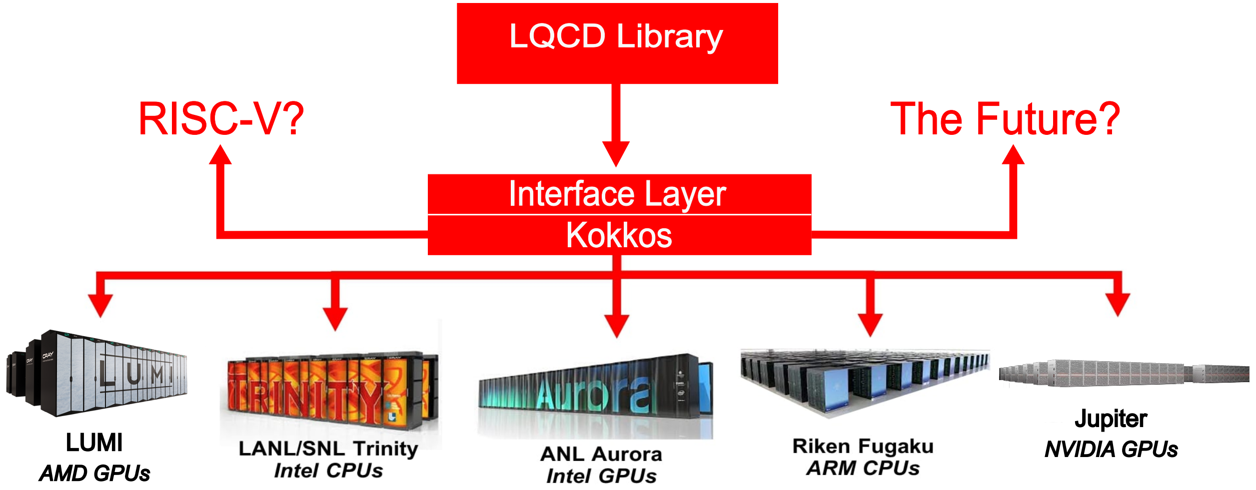
Lattice QCD is in need of a portable, performant, maintainable, extensible library with a low barrier of entry for new users in the exascale era

Motivation for a New LQCD Library



Portable (✓) Performant (✓) Maintainable (×) Extensible (×)

Motivation for a New LQCD Library



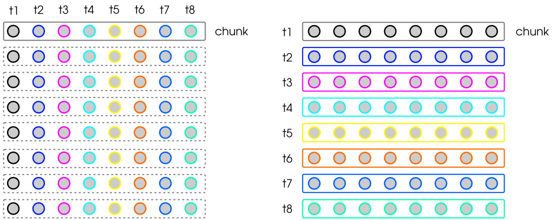
Portable (✓) Performant (✓) Maintainable (✓) Extensible (✓)

Kokkos

A C++ Performance Portability Ecosystem for Hardware Agnostic Code

The View

- A smart pointer to a chunk of memory allocated in some memory space
- Mapping of indexing in memory for coalesced access on GPUs and blocked access on CPUs
  - GPU: `U(x,mu,a,b)` → `U[b][a][mu][x]`
  - CPU: `U(x,mu,a,b)` → `U[x][mu][a][b]`



Supported HPC Architectures

- CUDA
- HIP
- SYCL
- HPX
- OpenMP

Execution Policies

Abstractions for loops over multi-dimensional index spaces with good mapping to hardware parallelism

Parallel Dispatch

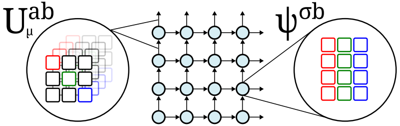
- `parallel_for`
- `parallel_reduce`
- ...

Introducing Kronos

$$\phi^{\sigma a}(x) = \sum_{\mu} \sum_{b} U_{\mu}^{ab}(x) \cdot \psi^{\sigma b}(x), \ \forall x, \sigma, a$$

Introducing Kronos

- MPI-parallel lattice field types which wrap Kokkos Views

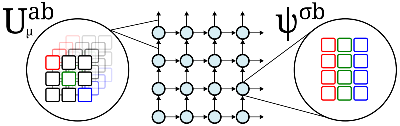


$$\boxed{\phi^{\sigma a}(x) = \sum_\mu \sum_b U_\mu^{ab}(x) \cdot \psi^{\sigma b}(x), \ \forall x, \sigma, a}$$

```
SpinorField4D<double, 4, 3> phi, psi;
GaugeField4D<double, 3> g;
```

Introducing Kronos

- MPI-parallel lattice field types which wrap Kokkos Views

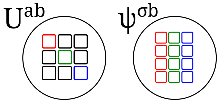


$$\phi^{\sigma a}(x) = \sum_\mu \sum_b U_\mu^{ab}(x) \cdot \psi^{\sigma b}(x), \ \forall x, \sigma, a$$

```
SpinorField4D<double, 4, 3> phi, psi;
GaugeField4D<double, 3> g;
```

- Site References

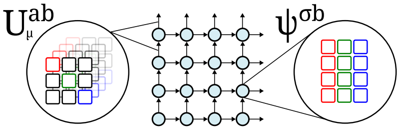
Objects which represent **access** to a spinor or gauge field at a particular lattice site




```
auto out = phi.site(i);
auto in = psi.site(i);
auto U = g.sitevector(i, 0);
```

Introducing Kronos

- MPI-parallel lattice field types which wrap Kokkos Views

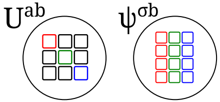


$$\phi^{\sigma a}(x) = \sum_{\mu} \sum_{b} U^{ab}_{\mu}(x) \cdot \psi^{\sigma b}(x), \ \forall x, \sigma, a$$

```
SpinorField4D<double, 4, 3> phi, psi;
GaugeField4D<double, 3> g;
```

- Site References

Objects which represent **access** to a spinor or gauge field at a particular lattice site

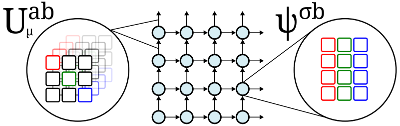


- Enable operator overloading
  - combine with stack-allocated temporaries in kernels
  - preserve coalesced access to views allocated on heap
  - C++20 concepts to generate all overloads automatically

```
auto out = phi.site(i);
auto in = psi.site(i);
auto U = g.sitevector(i, 0);
auto tmp = phi.site_type();
tmp = U * in;
for (int mu = 1; mu < 4; ++mu)
  tmp += U(i, mu) * in;
out = tmp;
```

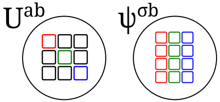
Introducing Kronos

- MPI-parallel lattice field types which wrap Kokkos Views



- Site References

Objects which represent **access** to a spinor or gauge field at a particular lattice site



  - Enable operator overloading
    - combine with stack-allocated temporaries in kernels
    - preserve coalesced access to views allocated on heap
    - C++20 concepts to generate all overloads automatically

- Loop over lattice sites using Kokkos parallel dispatch

$$\boxed{\phi^{\sigma a}(x) = \sum_{\mu} \sum_{b} U_{\mu}^{ab}(x) \cdot \psi^{\sigma b}(x), \ \forall x, \sigma, a}$$

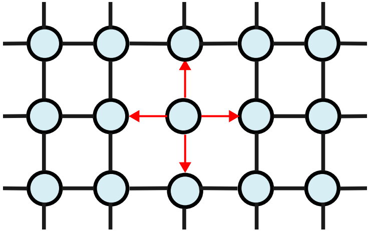
```
SpinorField4D<double, 4, 3> phi, psi;
GaugeField4D<double, 3> g;

Kokkos::parallel_for(
  N,
  KOKKOS_LAMBDA(int i) {
    auto out = phi.site(i);
    auto in = psi.site(i);
    auto U = g.sitevector(i, 0);
    auto tmp = phi.site_type();
    tmp = U * in;
    for (int mu = 1; mu < 4; ++mu)
      tmp += U(i, mu) * in;
    out = tmp;
  }
);
```

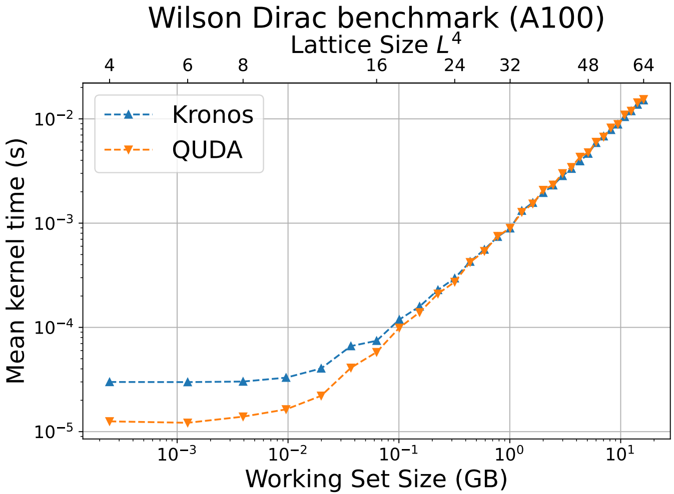
Putting It All Together - 4D Finite Difference (Wilson Dirac) Operator

$$\phi^{\sigma a}(x) = \psi^{\sigma a}(x) - \kappa \sum_{\mu} \sum_{\beta} \sum_{b} \left\{ (1-\gamma_\mu)^{\sigma\beta} U_\mu^{ab}(x) \psi^{\beta b}(x+\hat{\mu}) + (1+\gamma_\mu)^{\sigma\beta} U_\mu^{*ba}(x-\hat{\mu}) \psi^{\beta b}(x-\hat{\mu}) \right\}$$

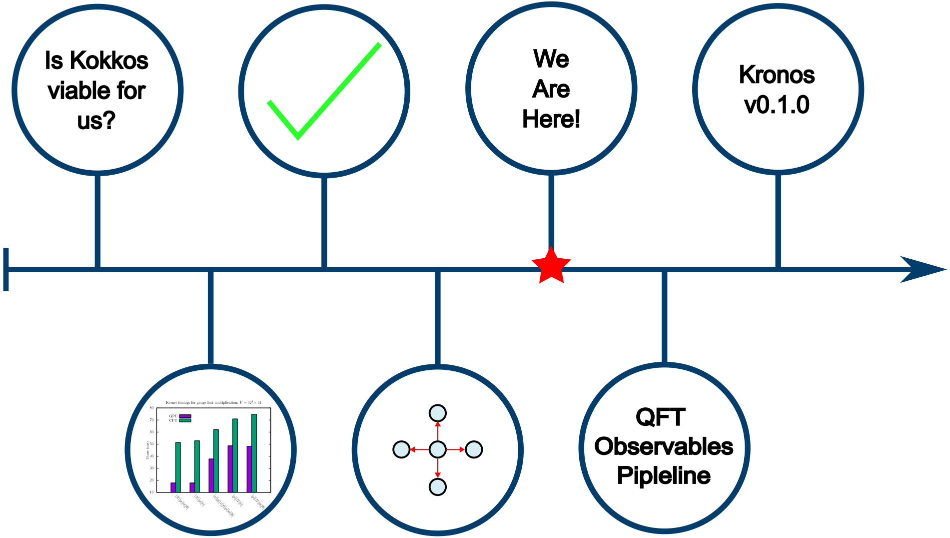
```
for (int mu = 0; mu < 4; ++mu) {
  tmp += ImGamma(U(i, mu) * in(ip[mu]),
         DslashGammas<Ns>(mu));
  tmp += IpGamma(conj(U(im[mu], mu)) *
         in(im[mu]), DslashGammas<Ns>(mu));
}
out = in(i) - ( m_kappa * tmp );
```



- Loop over color and spin abstracted away via $*$
- Halo exchange for U and ψ also included



The Roadmap



Thanks for your attention - Come See Our Poster!

Optimizing Performance

Auto Tuner

- A wrapper around Kokkos' parallel dispatch
- Runs a benchmark for a given kernel upon first use with different chunk and tiling sizes and caches the best result
- Tuning is distributed among different MPI processes
- Results in upwards of 50% decrease in execution time for some kernels