PROCEEDINGS OF SCIENCE

Review on Algorithms for dynamical fermions

Jacob Finkenrath[a,b,*]

[a] *Bergische Universität Wuppertal,*
Gaußstraße 20, Wuppertal, Germany
[b] *The Cyprus Institute,*
20 Konstantinou Kavafi Street, 2121 Nicosia, Cyprus

E-mail: finkenrath@uni-wuppertal.de

This review discusses the current and next steps in research of algorithms for dynamical fermions in large scale lattice QCD simulations.
First a short overview on the state-of-the-art of ensemble generation at the physical point is given. Followed by an overview on required steps towards simulation of large lattices for the Hybrid Monte Carlo algorithm. Here, in particular, the status of iterative solvers and tuning procedures for numerical integrators within the molecular dynamics are discussed.
This is followed by a review on the on-going developments for algorithms, with a focus on methods which are potentially useful to simulate gauge theories at very fine lattice spacings, i.e. well suited to overcome freezing of the topological charge. This includes modification of the HMC algorithm as well as a discussion of algorithms which includes the fermion weight via global correction steps. Parts of the discussions are on the application of generative models via gauge equivariant flows as well as multi-level algorithms.

The 39th International Symposium on Lattice Field Theory (Lattice2022),
8-13 August, 2022
Bonn, Germany

*Speaker

PoS(LATTICE2022)227

1. Introduction

The strong interaction between quarks and gluons is described by Quantum Chromodynamics (QCD). While the theory is asymptotic free towards high energies, at low energies the coupling becomes strong and quarks are confined within color-neutral hadronic states. In the low energy regime QCD can be solved on a discretised Euclidean space-time lattice with High Performance Computing (HPC) via lattice QCD, the so-far only known *ab-initio* approach.

An observable O at a set of *bare physics* and *lattice* parameters $\{\beta, am_i, \ldots\}$ can be evaluated via the path-integral

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}[U]\, O(U) \cdot p(U) \qquad \text{with} \qquad p(U) = \left(\prod_i^{N_f} \det D(U, m_i) \right) \cdot e^{-\beta S_g(U)} \tag{1}$$

for N_f number of quarks. At a line of constant physics, i.e. appropriate tuned quark masses m_i and a constant physical volume, continuum QCD can be recovered by an extrapolation in the lattice spacing a by using a set of different gauge couplings β. The integral measure is given by $\mathcal{D}[U] = \prod_{x=0,\mu=1}^{V,d} dH(U_\mu(x))$ with dH the Haar measure of SU(3), $V = L^3 \times T$ the lattice volume and $d = 4$ the dimensions. The Boltzmann factor $p(U)$ depends on the discretised Dirac operator $D(U, m_i)$ and the pure gauge action $S_g(U)$.

The standard method to solve eq. (1), given by a high dimensional integral, is to use Markov Chain Monte Carlo simulations (MCMC). The most common MCMC approaches can be roughly decomposed in two steps, a proposal step, where a new gauge configuration U' is generated with a conditional weight $q(U)$ with the proposal probability $T(U \to U')$, and a correction step

$$P_{acc}(U, U') = \min \left[1, \frac{p(U')q(U)}{p(U)q(U')} \right] . \tag{2}$$

If the ratio $r(U, U') = (p(U')q(U))/(p(U)q(U'))$ is log-normal distributed, the acceptance rate [1] is given by

$$\langle P_{acc}(U, U') \rangle = \text{erfc} \left\{ \sqrt{\sigma^2/8} \right\} \qquad \text{with} \qquad \sigma^2 = \langle r(U, U')^2 \rangle - \langle r(U, U') \rangle^2 . \tag{3}$$

A set of N configuration U_i in the thermodynamical equilibrium with weight $p(U)$ is called ensemble, and if the MCMC procedure (see for more details [2]) fulfils the fix-point or stability condition

$$\int \mathcal{D}[U] T(U \to U') p(U) = p(U') \qquad \text{for all } \; U' , \tag{4}$$

it follows

$$\langle O \rangle = \frac{1}{N} \sum_{i=1}^{N} O(U_i) + O\left(\sqrt{\frac{2\tau_{int}}{N}} \right) \tag{5}$$

where the autocorrelation time τ_{int} depends on the MCMC method.

Taking the limits towards continuum physics is extremely computational challenging, i.e. roughly the computational cost increases $\propto a^{-\gamma_0} m^{-\gamma_1} L^{\gamma_2}$ with power laws $\gamma_0, \gamma_1, \gamma_2 > 1$. A major part of the computations are the fermionic contributions, which can be represented by the inverse of the Dirac operator, e.g. in case of the Boltzmann-factor via pseudofermions [3]

$$\det D(m_i, U) = \int \mathcal{D}[\eta] \exp\{-\eta^\dagger D(m_i, U)^{-1} \eta\} \quad \text{if} \quad x^\dagger (D + D^\dagger) x / x^\dagger x > 0 \quad \forall x \in \mathbb{C}^{12V} . \tag{6}$$

2. State of the art

Due to the computational cost of computing the inverse operator D^{-1}, fermion determinants were neglected in the first major simulation efforts, in the so-called quenched approximation. In the early years of the millennium this led to ~10% systematic effects in the hadron spectrum [4]. This precision is not enough for advances in the low energy regime of the standard model, where search for new physics is driven by increased precision at the so-called precision frontier. To detect signs for new physics, i.e. by deviations with experiments, lattice QCD quantities have to be measured at sub-percentage precision, e.g. in case of the hadronic vector contribution (HVP) to the anomalous magnetic moment of the muon. To reach this precision, lattice QCD simulations have to include fermions in the simulation, also called using dynamical fermions, and control all major systematic effects, like finite size, finite discretization and light quark mass effects. The later effect is eliminated by directly simulating at the physical point, where the quark masses are tuned to reproduce the physical meson masses, such as pion and kaon masses. Directly simulating at the physical point is possible due to advances on the algorithmic level as well as on the hardware site. Nowadays these physical point ensembles are generated by various lattice collaborations around the globe. The selected actions of the collaborations differs by the used gauge as well as the used fermion discretization, but most ensembles are generated at the isosymmetric point, e.g. with two mass-degenerated light quarks, and a dynamical strange and in most cases with a dynamical charm quark, denoted as $N_f = 2 + 1(+1)$. Most of the generated ensembles, see Fig. 1 for an overview, have an lattice size of > 5fm and are generated at lattice spacings in the range of $[0.05 - 0.2]$ fm. A set of ensembles in this range enables to control the major systematic effects, i.e. finite volume and cut-off effects, in order to reach $O(1\%)$ precision in observables.

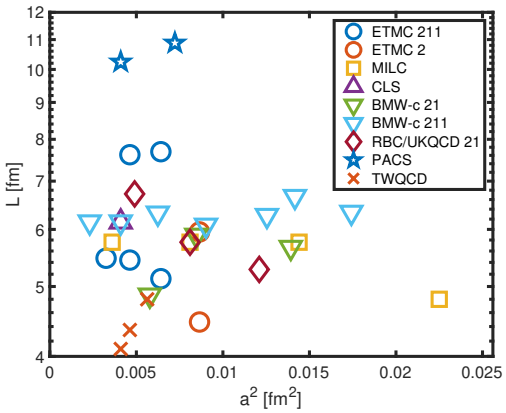

Figure 1: The figure shows the physical point ensembles generated by various collaborations each using different fermions discretizations (see also [11]).

The MCMC algorithm of choice to generate these ensembles is given by the Hybrid Monte Carlo (HMC) [5, 6] algorithm. The method is used by all collaboration with different variants to improve computational efficiency, i.e. which are based on Infra-red/ultra-violet (IR/UV) preconditioning. The most common techniques are given by even-odd-reduction, by Hasenbusch-mass-preconditioning [7] in the light quark sector and by rational HMC [8], based on rational approximation, in the heavy quark sector. A subvariant used in particular for Domain Wall fermions is given by the Exact One Flavor algorithm [9, 10], which decompose the Dirac operator into two hermitian operators in spin space making use of the γ_5 hermicity. Note that a detailed description on the used fermion discretizations as well as on the used preconditioning methods for the ensemble generation of the different collaborations can be found in [11].

Despite the elimination of quark mass effects, major challenges are remaining at the precision

frontier. To make advances the remaining systematic effects need to be further minimized to be sensible in the search for new physics beyond the standard model. The major contributions are given by finite size effects, e.g. in the HVP of the anomalous magnetic moment [15], and finite discretization effects present in continuum extrapolations, see e.g. [12–14].

Larger and finer lattices are needed to extend our knowledge in various directions, such as in the field of bottom- and charm-physics, in measurements of multiple particle scattering or in improving lattice results for the anomalous magnetic moment g-2. The next step here is to generate new ensembles with larger lattice size, such as > 8 fm as well as new ensembles at finer lattice spacings < 0.05 fm. These steps are already consider by various lattice groups, for example by the US *snowmass study* in targeting ensemble generation at a fine lattice spacing of $a = 0.04$ fm and a lattice extent of $L = 256$ [16, 17] or is already under production by the PACS collaboration at $a = 0.043$ fm with $L = 256$ [11].

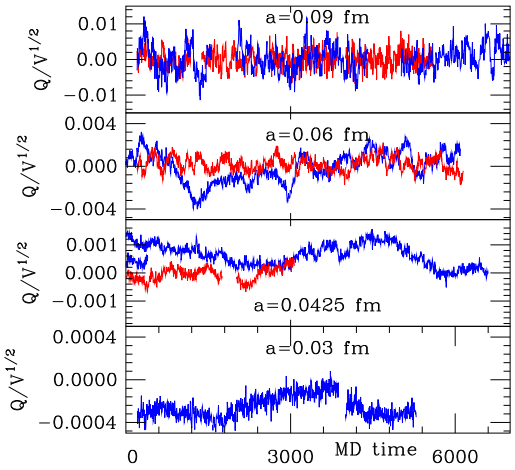

Figure 2: Figure shows the fluctuation of the topological charge in case of the MILC $N_f = 2+1+1$ HISQ staggered ensembles [19] (red history is obtained at physical pion masses).

However, generating these ensembles requires to meet two major challenges at the computational and algorithmic level. For $L > 128$ it requires to manage large computational costs per HMC-step at simultaneous relative large real simulation time up to 10h-24h per molecular dynamics integration. For $a < 0.05$ fm it requires new algorithmic advances towards finer lattice spacings because the HMC algorithm develops very large autocorrelation times due to topological charge freezing [18, 19] at very fine lattice spacings, see Fig. 2. This makes it practically impossible to generate an equilibrated ensemble with available computational resources below $a \ll 0.05$ fm with periodic boundary conditions.

In this review, we will take a closer look to this two challenges. We will discuss the status of generating ensembles using the HMC algorithms in Sec. 3, in particular towards larger lattices. In the second part in Sec. 4 of the review, we will discuss algorithmic approaches, which can overcome limitations of the commonly used HMC algorithm, i.e. by modifications or alternative approaches.

3. Scaling of Hybrid Monte Carlo

The Hybrid Monte Carlo algorithm [5, 6] is a MCMC algorithms, which samples configurations weighted with the Boltzmann weight $p(U)$. This is done by iterating the following steps (see also [2]) .

1. Generating conjugated momentas P and pseudofermion via heat-bath steps.

2. Obtaining a proposal of a new set (U, P) at Monte Carlo (MC)-time $t = \tau$ by integrating

Hamilton's Equations via

$$\frac{dP}{dt} = -\frac{\partial H}{\partial U} \qquad \text{and} \qquad \frac{dU}{dt} = \frac{\partial H}{\partial P} \,. \tag{7}$$

3. Followed by an acceptance step with accept-reject probability

$$P_{acc} = \min\left[1, e^{-H(U_\tau, P_\tau)+H(U_0,P_0)}\right], \tag{8}$$

where $H = P^2/2+\phi^\dagger[D^\dagger(U)D(U)]^{-1}\phi+\beta S_g(U)$ is a basic Hamiltonian for mass-degenerated Wilson light quarks.

The major part of the computational effort to generate an ensemble is roughly given by the size N, i.e. the number of configuration, and the cost per molecular dynamics (MD) . The cost for the MD can be hierarchical decomposed into the number of integration steps N_{step} per MD (see subsec. 3.3) and the numerical cost to invert the fermion matrix cost$^{(inv)}$ (see subsec. 3.2). The computational cost of the inversion can be further decomposed into linear algebra operations, which are dominated by the matrix vector product (see subsec. 3.1). Note that the same hierarchy is introduced at the software level. Namely, the lowest level consists of linear algebraic function, the intermediate level of linear solver methods and the highest level of numerical integrators.

The challenge towards larger lattices does not only depend on the increasing computational cost, but also on how the computation can be parallelized and how it performs on novel HPC hardware. In case of MCMC simulations the potential parallelization is limited by the sequential nature of a Markov Chain. In general this means speed up can only be achieved by parallelization at fixed lattice size, i.e. limited by the strong scaling of the used algorithms. In case the upper bound of the window, where the algorithm still scales, does not scale as well as the computational cost, it results into larger simulation times even if computational resources are not limited.

Another challenge is given by the HPC architecture which comes with a relative short life cycle with new architectures entering HPC on a roughly regular two year basis while remaining at the top for roughly 6 years. This requires a constant effort in adapting the computational kernels to be performend on the updated and sometimes completely novel hardware, e.g. on the up-coming HPC machines equipped with next-gen GPUs from different vendors such as Nvidia, AMD or Intel.

3.1 Wilson Dirac stencil on European Supercomputers

The lowest level of the lattice QCD software stack consists of linear algebra operations, such as the matrix vector product given by the Dirac stencil $D(U)x$. The Dirac operator $D(U)$ can be represented as a huge sparse matrix with next-neighbour interaction, which requires exchange of boundary terms if parallelised. The arithmetic intensity of the stencil is roughly given by 1.0 and the computation cost grows with the lattice volume V. On current HPC-hardware the operator is bandwidth bound, while it becomes latency bound within the strong scaling limit. The computational kernel, often within a conjugate gradient procedure, is used to benchmark HPC machines, e.g. Fig. 3 shows results from the European supercomputers by using various software packages. Namely, for Intel Xeon (Phi) architecture QPhiX [34] was used, while for machines with Nvidia GPUs QUDA [22] and for the ARM machine the kernel from grid [20, 21]. The

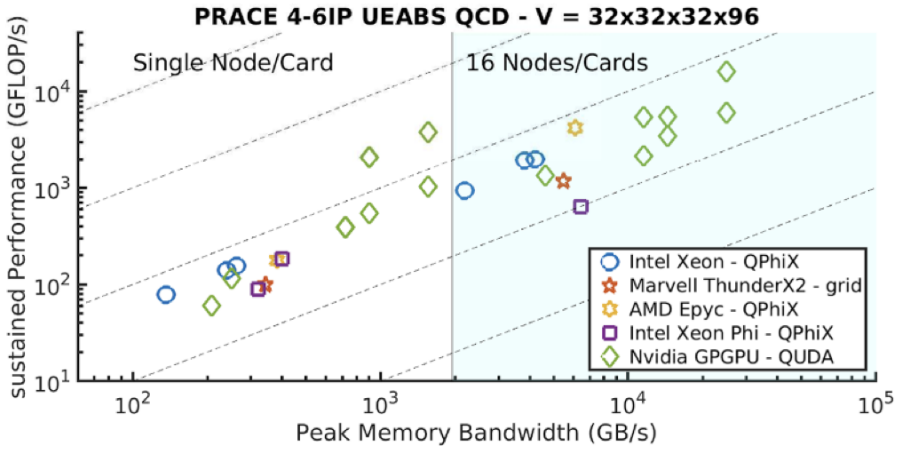

Figure 3: The figure shows the sustained performance in GFLOP/s of CG benchmarks on European HPC systems Tier 0 obtained within PRACE 4IP-6IP on a single and 16 nodes. The two QUDA outlayers corresponds to using 4 GPUs per node.

results illustrate that the single node performance increased as well as strong scalability improved continuously over the last 10 years.

Note that performance results for novel HPC hardware are still missing, however several lattice QCD packages already offer optimized stencil operation such as QUDA or grid. Strong scalability for QCD kernels on Fugakus Fujitsu A64fx chips using Bridge++ can be found under [24]. The algorithmic intensity can be reduced by introducing multiple right hand sides (rhs), while this increases the computational effort per application arithmetic intensity can be reduces and strong scalability as well as performance can be improved. Several packages such as DDalphaAMG [37], QUDA [26] or grid [25] are providing these kernels.

3.2 Solvers

The integration of the MD during the HMC requires to solve frequently the Dirac equation

$$D(U)\cdot x = b \quad \text{with} \quad x, b \in \mathbb{C}^{12V} \quad \text{and} \quad D(U) \in \mathbb{C}^{12V\times 12V}. \tag{9}$$

with known right hand side (rhs) b. For large sparse matrices, like the Dirac operator $D(u)$, the common procedure is based on iterative methods which builds a Krylov space $\mathcal{K}_{n+1}(A, x_0) = \{x_i | i = 0, \ldots, n \text{ with } x_i = A^i x_0\}$ to find the solution x. The conjugate gradient (CG) solver requires a hermitian matrix, which we can derive by writing $A = D^\dagger D$, while for more flexible methods, such as the flexible generalized minimal residual method (FGMRES), we directly use $A = D$. Note that in many cases, using the better conditioned even-odd reduced operator yields an effective speedup of the inversion. For this review we will use configurations of different ensembles at different lattice sizes and lattice spacings generated by the ETM collaboration at physical quark masses, here referred to as cB64, cB96, cC80 and cD96 where B : 0.08 fm, C : 0.069 fm and D : 0.058 fm and for example cB64 corresponds to an lattice extent of $L = 64$ (see also [99, 100]). The ensembles will be used to discuss results obtained by the CG solver as well as for different implementation of the multigrid (MG) method.

3.2.1 Conjugate Gradient solver

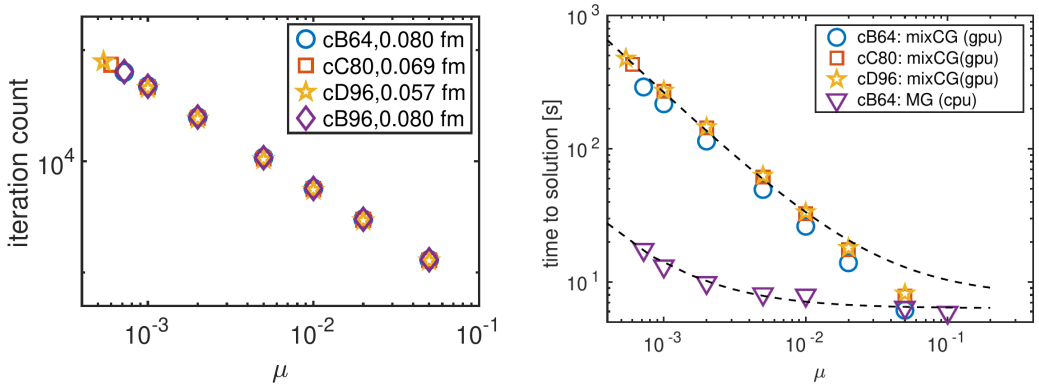

Figure 4: The left panel shows the iteration per solution of the CG solver measured on the ETMC physical point ensembles in dependence of the quark mass. The right figure shows the time per solution obtained using the CG implemented in QUDA and MG solver implemented in DDalphaAMG. All results were obtain using JUWELS Boosters equipped with Nvidia A100 while the CPU data is obtain on Intel Skylakes CPUs on SuperMUC.

The most basic Krylov subspace solver is given by the conjugate gradient (CG) method, which is commonly used for large masses, e.g. see Fig. 4. It depends mainly on the matrix-vector stencil and can be efficiently parallelized with very good scalability compared to other QCD kernels. The algorithm can be further sped up by using mix-precision methods, i.e. performing most iteration using low-precision arithmetics. This leads to a speed up by 40% using single and 50% half precision on Juwels Booster A100 GPUs, as shown on the left panel in Fig. 4 . As pointed out by [23] this can be further optimised using a suitable representation of low-precision numbers.

The number of needed iterations of the CG solver is proportional to the condition number of the operator, i.e. in case of lattice QCD this reduces to the dependence on the smallest eigenvalue. It is independent from the density. Towards the physical point the iteration count drastically increases towards $70k$ iteration. For the computational costs we found

$$cost_{CG} \approx V \cdot \left(\frac{b}{\mu} + a\right) \approx V\frac{b}{\mu} \tag{10}$$

with $b/a \sim 0.04$ as depicted on the right panel in Fig. 4.

3.2.2 Multigrid approaches

The high iteration count of the CG solver at the physical light quark mass can be overcome by using preconditioned Krylov subspace solvers. A very effective method in reducing the iteration count to $O(10)$ is given by algebraic multigrid (MG) procedures. This is done by using a flexible solver such as FGMRES and treating the highly fluctuating UV modes via a smoother, e.g. based on Gauss-Seidel iterations or on a Schwarz-alternating procedure, and the low fluctuating IR modes via a coarse grid correction based on an algebraic multigrid approach.

Multigrid procedures outperform the basic CG methods by up to two orders of magnitude for several versions of lattice fermion actions. Examples are Wilson fermions [27–30] and twisted

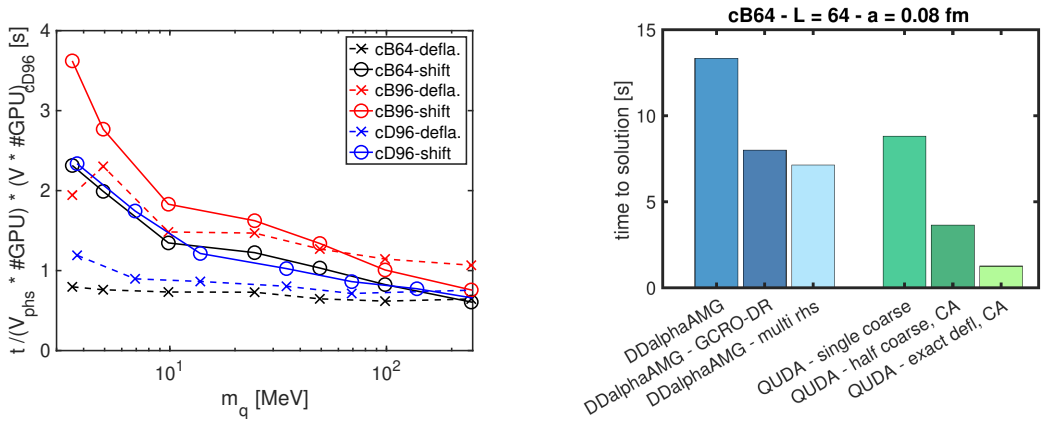

Figure 5: The left figure shows the mass-dependence of a 3 level MG solver implemented in QUDA using a coarse grid shift in μ (solid lines) and exact deflation (dotted line) relative to the solution time on cD96 at $m_q \sim 250$ MeV. The right figure shows recent performance improvements of the DDalphaAMG solver (in blue) and of the QUDA MG solver (in green)

mass fermions [31]. While for other fermion types like staggered fermions [32, 33] or Domain Wall fermions [35, 36] effective MG methods became only recently known and their improvements are less significant (up to one order of magnitude). Note that a MG procedure comes with an additional overhead, given by the setup time for building the coarse grid projection and restriction operator.

As shown on the right panel of Fig. 4 MG methods are suppressing the quark mass dependence. We found for the DDalphaAMG solver $\propto V(0.001/\mu+1)$ [31]. A similar behaviour is found in case of the MG solver implemented by QUDA, see left panel of Fig. 5. Here, the quark mass dependence can be even further suppressed by using exact deflation on the coarsest grid. In contrast to the CG solver, we found that the numerical costs increases more than linear with the volume at constant lattice spacing, i.e. comparing cB64 to cB96, while keeping the physical volume constant the cost scales as expected, i.e. comparing cB64 to cD96. A possible explanation is that the cost scales additionally with the density of the low eigenmodes.

If MG approaches are used within the MD of the HMC, the constant update of the setup limits the effectiveness of the solvers. The overhead can be limited by updating the previous setup with less setup iterations. This leads, however, to additional reversibility violations, which can be minimise with higher residual precision. Moreover the window, where the multigrid approach scales strongly, is limited, towards large parallelization by the coarse grid size and towards low parallelization by the memory.

While the usage of a MG method can improve the total cost per trajectory, as shown in Fig. 7, the limitation given for the maximal parallelization increases the simulation time, i.e. we found that the upper bound of the strong scaling window increases roughly with L^3 while the cost for the HMC increases with $L^{4.5}$. This leads to time per trajectories of around 6-8h on $L = 96$ lattices on 384 Skylake nodes of SuperMUC-NG.

However this can be improved by developing more efficient coarse grid algorithm or utilizing GPU machines. Examples for algorithmic improvements are given in case of DDalphaAMG by a multiple rhs version, see [37], and by additional coarse grid improvements [38], based on pipelined

polynominal preconditioning of a deflated restart GCR method. The results are shown on the right panel of Fig. 5. An even more impressive improvement is seen in case of the performance improvements of the QUDA MG solver in the past years, see right panel of Fig. 5. Using communicating avoiding CG in combination with reduced precision, the method improved the time to solution from 8 sec on 64 Nvidia P100 nodes to 3.7 secs on 16 Nvidia A100. Deflating additional 800 exact coarse grid eigenvalues further reduces inversion times to 1.26 secs. This also reduces significantly the quark mass dependence, as shown on the left panel of Fig. 5. Moreover further improvements are expected by using multiple rhs, e.g. shown in case of grid [20, 21] on Juwels Booster, see [25], which found significant speed up.

While improvements of the coarse grid directly improve observable calculations, MG improvements, which comes with an additional overhead, might be not directly useful for the MD integration, e.g. exact deflation of QUDA MG procedure would require a relative large setup time such that the standard variant is significant faster.

3.3 Integrators

The HMC algorithm propose a new gauge configuration via MD integration, where the total cost of the HMC algorithm is directly proportional to the number of required integration steps N_{step}. The integration can be done via a symplectic, reversible integrator as required by the fix-point condition. The most common integrator approach is given by the second minimal norm scheme [43], while a large collection of higher order schemes can be found in the work by Omelyan, Mryglod and Folk [44], which also includes force gradient approaches.

To correct for the error of the numerical integration, an accept-reject step is done after each trajectory. The acceptance rate of the accept-reject step is given by eq. (3), here $\langle P_{acc}\rangle = \mathrm{erfc}(\sqrt{\sigma^2/8})$ and the related variance $\sigma^2(N_{steps}, \mu, V, \ldots) = var(\delta H)$ can be written as a function of integrator as well as lattice parameters [1, 45, 46]. For a n-th order integrator, we find

$$\mathrm{var}(\delta H) = \frac{1}{N_{steps}^{2n}}\sigma^2(1, m, V, \ldots) \propto h^{2n}\, V\, m^{-2\alpha_0} \tag{11}$$

with $\alpha_0 > 1$ and $h = 1/N_{step}$. Now, a suitable tuning condition is given by fixing the acceptance rate $\langle P_{acc}\rangle$ and minimizing the cost function which depends on the number of steps and the cost per force computation. Using IR/UV preconditioning techniques, such as Hasenbusch mass preconditioning or rational approximation, can further reduce the cost but increases the search space.

A suitable approach to calculate $\mathrm{var}(\delta H)$ is given the first error terms, which can be calculated via Poisson Brackets and are the first non-zero higher order terms of the Shadow-Hamiltonian [46]. In case of the second minimal norm scheme, the Shadow-Hamiltonian is given by

$$\tilde{H} = T + S + h^2\left(\frac{6\lambda^2 - 6\lambda + 1}{12}\{S,\{S,T\}\} + \frac{1-6\lambda}{24}\{T,\{S,T\}\}\right) + O(h^4)\,. \tag{12}$$

with the freely selectable parameter λ, which roughly minimizes the $O(h^2)$-term by setting to $\lambda \sim 0.19$. If we set $\lambda = 1/6$ the leading order term is given by the force $\{S,\{S,T\}\} = \mathrm{tr}(F^2)/a$. This relation can be now used to tune the step size h.

Moreover, the measurement of the force $f = \partial H/\partial U$, can be used to evaluate methods. An example is given by the RHMC [8] with Block Krylov solvers and multiple pseudofermions [47]. Here, a variant of RHMC is introduced by splitting up $\det[D^\dagger D] = \det[(D^\dagger D)^{1/n_{pf}}]^{n_{pf}}$ and for the variance of the force follows [47] $\mathrm{var}(F^2(n_{pf})) = c_s n_{pf}^{-1} + c_3 n_{pf}^{-2} + O(n_{pf}^{-3})$. This reduces the required steps at given acceptance and is ideal to combine with Block Krylov solvers. An example are given by the shifted Block CGrQ method, [48], which leads to faster convergence by increasing the search space and is, in general, ideally to combine with multiple rhs kernels, e.g. [37]. As depicted in Fig. 6 the RHMC combined with a Block solver leads to speed ups of ~ 6 achieved in case of $N_f = 4$ on a $L = 8$ lattice.

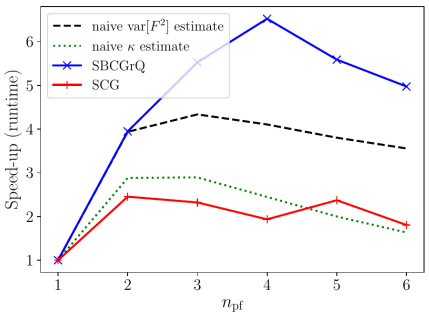

Figure 6: The figure shows the relative speed up using higher roots n_{pf} in combination with a Block Krylov solver (taken from [47]).

IR/UV splitting of the action are naturally combined with nested integration schemes, where the high fluctuating UV terms can be integrated with a smaller step size and expensive IR terms with a larger step size. Let us write the Hamiltonian as $H = S_1 + S_0 + P$, with S_1 containing the IR modes and S_0 the UV modes. Then, a nested integration scheme based on the second minimal norm scheme with $\lambda = 1/6$ can be written as [57, 58]

$$\Delta(h) = e^{\frac{h}{6}\hat{S}_1}\Delta(h/2)e^{\frac{2h}{3}\hat{S}_1 - \frac{h^3}{72}\hat{C}_1}\Delta(h/2)e^{\frac{h}{6}\hat{S}_1} \tag{13}$$

with $\Delta(h/2) = e^{\frac{h}{12}\hat{S}_0}e^{\frac{h}{4}\hat{P}}e^{\frac{h}{3}\hat{S}_0}e^{\frac{h}{4}\hat{P}}e^{\frac{h}{12}\hat{S}_0}$ and $C_1 = 2\sum_{x=1,\nu=0}^{V,3}\frac{\partial S_1}{\partial U_\nu(x)}\frac{\partial^2 S_1}{\partial U_\nu(x)\partial U_\mu(x)}$ the force gradient term, which comes with a second derivative. If the integration errors from S_0 are sufficiently suppressed, this leads to an fourth order integrator. As noted by Lin and Mawhinney [49], the force gradient term can be approximated numerically, which avoids the implementation of second derivative terms. The approximated variant can be implemented based on the force terms and only requires more memory for an additional gaugefield. Moreover the total cost of inversions of the fermion matrix is even reduces by 4/3 compared to the exact force gradient term, see [58] .

A very common IR/UV-spliting method is given by Hasenbusch mass preconditioning [7], which introduces additional mass terms μ_i. In principle, these additional parameters can be now tuned, if the first Poisson-Brackets are known. In case of a fourth order integrators like eq. 13 the brackets requires the calculation of derivative of up to the fourth order [46]. In practice the shifts μ_i are tuned empirically. We found roughly a dependence of $\propto (\frac{\Delta^2 m_i}{\mu_i^2})^k$ with $\Delta^2 m_i = \mu_{i+1}^2 - \mu_i^2$ and $k \in [3, 4]$ for corresponding contributions to the variance of δH eq. 13. This leads to a relative shallow nested integration setup by using a fourth order scheme with deepth of three or even only two levels with Hasenbusch mass shifts roughly given by $\mu_i \in \{\mu + \mu[0, 1, 10, 100, \ldots]\}$. Note that this choice also suppresses effectively the quark mass dependence, i.e. $O(\Delta^2 m_0) \approx O(\mu^2)$.

3.3.1 HMC on GPUs

The relative long (real) time per MD of up to 8 hours can be sped up by porting the major computational kernels to novel HPC architecture or to accelerator cards, such as GPUs. For GPUs,

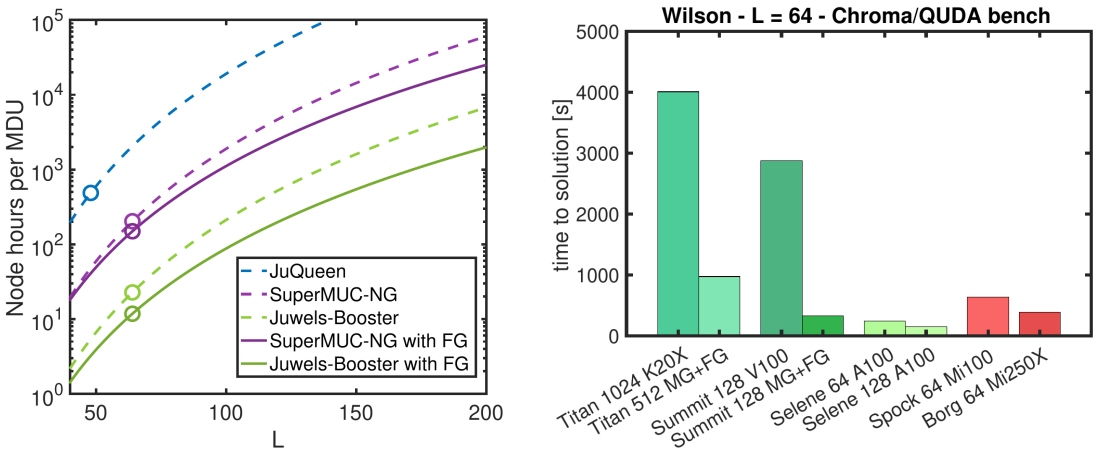

Figure 7: The figure shows the improvements on computational costs of HMC simulation of the last years. On the left side, the improvements of node hours per MDU are shown in case of tmLQCD illustrating the effect using MG solvers, higher order force gradient (FG) integrators and GPUs. On the right side the improvement time to solution of a MDU in case of using Chroma with QUDA is shown.

highly optimized software packages like QUDA [22] or grid [20, 21] exists which can be used to offload major computational tasks, like the inversions of the Dirac operator. QUDA is already used in several frameworks, e.g. Chroma as well as the packages MILC and CPT utilize solvers implemented in QUDA to speed up HMC-simulations, see right panel of Fig. 7. A similar effort is on-going for twisted mass fermion implemented within tmLQCD [39]. Moreover also fully-fledged software packages implements HMC on GPUs, such as grid.

The highly optimized kernels of these package are accessible through python APIs, which enable to interact and use computational kernels in an high level environment, simplifying programming and pushing algorithmic developments. Implementations of the HMC method are available in the python package GPT [42], which uses grid [20, 21] and in the python package lyncs-API [40, 41], which utilises QUDA functions [22] to perform on GPUs.

3.4 Conclusion

The computational cost for the HMC algorithms could be reduced by advances based on algorithms and hardware, such as the usage of MG methods or the utilization of HPC machines equipped with GPUs. In case of twisted mass fermions this leads to a reduction of node hours per trajectory in simulations of physical point ensembles. As illustrated on the left panel of Fig. 7, using a MG solver in combination with Intel Xeon Server CPUs gave a speed-up by a factor 10 compare to the cost required on JuQueen. By utilizing the QUDA solvers another reduction by a factor 10 in node hours were obtained. A comparable improvement is found in case of Wilson fermions using a Chroma driven HMC, which utilizes QUDA. As shown on the right panel of Fig. 7, a speed-up of a factor 40 was obtained using 128 Nvidia Ampere 100 of Selene compare to 1024 Titan nodes each equipped with a Nvidia K20X GPU.

The total costs for ensemble generation by the HMC algorithm using N_H Hasenbusch terms

can be written as

$$\text{cost}_{HMC} = N(\tau_{int}) \sum_{i=1}^{N_H} N_{steps,i} \times \text{cost}_i^{(inv)} \propto a^{-\gamma} \frac{V^{1/2n}}{a^{2/n}} \frac{V}{a^4} = a^{-(4+2/n+\gamma)} V^{4+1/2n} \tag{14}$$

if we assume $\tau_{int} \propto a^{-\gamma}$ where $\gamma \approx 5$ for periodic boundaries [18]. The dependence on the quark mass is reduced by using MG solvers and Hasenbusch mass preconditioning. The volume scaling can be reduced by using integrators of higher order n. For a lattice with extent $L = 192$ follows that a trajectory requires $O(1000)$ node hours on a HPC machine equipped with Nvidia A100. If strong scalability of the solvers towards this lattice sizes does not deteriorate, large lattice sizes are in reach with exascale computing (see Fig. 7).

4. Towards fine lattices

Towards fine lattice spacings $a < 0.05$ fm the autocorrelation time of observables calculated on configuration sampled with the HMC algorithm increases drastically with $\tau_{int} \propto a^{-5}$ or even exponentially [18], which leads to freezing of the topological charge (see Fig. 2). The phenomenon is related to the gauge group and is also present in case of pure gauge simulations. Overcoming this critical slowing down of the MCMC algorithms towards the continuum is under active research and several different approaches are investigated. These different methods can be roughly categorised into two subcategories, methods not based on the HMC proposal by changing the proposal procedure $T(U \to U')$ and including corrections via an accept-reject step, and methods which adapt the HMC either by modifying the procedure of the MD integration or by changing the action or boundary conditions.

The major obstacle for developing efficient methods is given by the acceptance rate eq. (3). If the new configuration is not proposed with care, the variance of the ratio $\text{var}(r(U, U'))$ is proportional to the volume and the acceptance rate is decreases $\propto \exp(-V)$. Only gauge proposals which can suppress this volume fluctuations will work as MCMC methods in the large volume limit. In general, methods based on MD integration have a very solid advantage. Namely the fluctuations only increase proportionally to the discretization error of the numerical integration $\propto V^{1/2n}$.

4.1 Modification of the HMC

To keep the volume scaling of the MD integration for the gauge proposal, different modification of the HMC algorithms are considered.

One possible way is to change the boundary conditions. A common approach to overcome topological freezing is given by open boundary conditions in time [50]. The boundary conditions allow transitions between topological sectors and reduces the corresponding autocorrelation time to $\tau_{int}(Q) \propto a^{-2}$. This comes with the breaking of translational invariance and larger extents in the lattice time direction. Open boundaries are the standard in simulations of CLS and are used in simulations of very fine lattice spacings, see e.g. [51].

Another approach is given by generating masterfields [54, 82]. Here, the idea is to increase the physical volume such, that local topological charge fluctuations are sufficient even if the global charge is fixed. An algorithm, which is investigated to enable this masterfield simulation, is based

on stochastic molecular dynamics [56]. Another way is given by fixing the simulation to each topological sector and averaging over the different sectors during the calculation of the observables [52, 53]. Another class of algorithms, which are using coarser and smaller lattices to propose a finer lattice is given by multiscale equilibration [59–61]. However, here the coarse grid to fine grid transition requires re-thermalization of the fine grid.

Another possible way is to modify the MD integration. For example by using skewed detailed balance [63]. This approach extends the accept-reject procedure, i.e. after a rejection the MD integration is continued, not re-started, and the additional accept-reject is modified including the probability distribution form the initial, the first and second step. This can be iterated, however acceptance of the second or even higher step might be small and the procedure works likely only if the integration errors oscillate and does not grow. The extension can be done, such that the fix-point condition is fulfilled.

Another way is given by using trivializing maps. The idea is to map the theory through a variable change to a trivial theory. Here, slow modes, e.g. which couple to the topologocal charge, are mixed with high frequent modes and gauge updates decouples easily in few integration steps. However, a major task is to find an appropriated map to project between the target and trivial region. In the initial proposal, the Wilson flow was proposed [64]. Alternative approaches to design such maps are motivated by the Schwinger-Dyson equation [65] or by combination with normalizing flows [66, 67].

Another possibility is to change how different modes are integrated during the MD integration. This can be done by redefining the conjugated momenta term by using Riemannian-mannifolds or Fourier acceleration [62]. This effectively changes the integration steps for the low and high modes, by accelerating the integration in the direction of the slow modes and deaccelerating the high modes.

To tame or trigger dynamics during the integration one can add marginal terms to the Hamiltonian. An example is to add Pauli-Villars fields [68]. These fields can reduce large cut-off effects in many fermion simulation such as $N_f = 8, 12$ and make simulations also at coarser lattice spacings possible.

In a similar way additional terms can be added to the Hamiltonian, e.g. which can de-correlate modes. For example, in case of the topological charge a metapotential, which couples to the topological charge, can be added which effectively decreases topological barriers between sectors [69, 70]. The additional terms are coming with an additional weight, which can be corrected after sampling the configurations, e.g. by reweighting. While it was found [69], that autocorrelation times in case of pure gauge simulations can be minimized, the additional term comes with additional numerical costs.

Another way is to use modified integrators, such as machine learned leap frog integrators [71] to de-correlate the update. This has to be done with some care otherwise the error of the integration might recover the linear volume behaviour.

4.2 Global Corrections

An obvious idea to overcome long autocorrelation times is to introduce a gauge proposal without autocorrelations. If the distribution of the proposal is known, a combination with an accept-reject step leads to a valid MCMC algorithm, as long as the fix-point condition is satisfied.

However, as discussed, this naively leads to an ineffective algorithm at large volumes V, i.e. it scales with $\langle P_{acc} \rangle \propto e^{-V}$.

The challenge is, if a suitable proposal with short autocorrelation time is found, to control the variance of the logarithms of the ratios $r(U, U')$

$$\Delta S = \ln p(U') - \ln p(U) - \ln q(U') + \ln q(U) \ . \tag{15}$$

This can be done by restricting the phase space of the distributions. Namely, the Boltzmann-factor can be factorized into a product of distributions, each depending on a different parameter space.

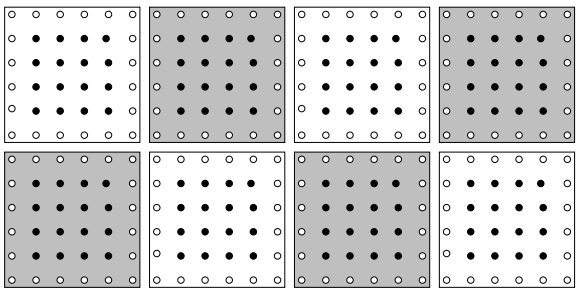

Figure 8: Domain decomposition of a two dimensional lattice into black and white blocks. All filled points contains links only located in the corresponding domains (taken from [72]).

For ultra local actions, such as the pure gauge plaquette action $\beta S_g = \beta/(2N) \sum_{x;\mu<\nu} \mathrm{Re}(1 - \mathrm{tr}\, P_{\mu,\nu}(x))$, this can be done via domain decomposition of the lattice, e.g. see Fig. 8. Then the action splits up into parts, which are independent of each other and defined within a domain and parts which are connecting domains, i.e. the global distribution splits up into a product of distributions which scales with the domain sizes. An idea is to find an transition probability, which can propose new gauge fields within a block and allows to flip the topological charge, see subsec. 4.2.2.

In case of the fermion determinant, decomposition into only local parts is not possible, however one can use the Schur decomposition of the determinant [72, 73]

$$\det D = \det S \prod_i \det D_i^{(blk)} \quad \text{with} \quad S = D_{w,w}^{(blk)} - D_{w,b}(D_{b,b}^{(blk)})^{-1} D_{b,w} \tag{16}$$

which factorizes the determinant in local part, given by the block determinants, and a global part, given by the Schur compliment S. By further decomposing the blocks into smaller blocks, the method becomes recursive. The decomposition can be done via asymmetric domains using red-black coloring, which leads to a decomposition in time used in a fermionic multi-level approach [74, 75], see subsec. 4.2.1 . This can be extented to a decomposition in four dimension, as discussed in [84, 85]. Alternative decompostions of the determinants are proposed, such as a complete factorization in time [76].

Another possibility to control the variance of eq. (15) is to use correlations between the distributions or the corresponding action. This can done by using parameter, e.g. via a linear parameter, which introduces shifts in the gauge coupling $\delta\beta$ [73, 77, 101]. This can be extented arbitrarily to a full parametrisation of the corresponding distribution. For example one can introduce neutral networks, which can be trained via machine learning [78], see subsec. 4.2.3.

4.2.1 Multilevel algorithms

Domain decomposition techniques are relevant for other aspects. For example they are used in MCMC approaches called multi-level algorithms. The basic idea is to make use of decorrelated and separated domains to improve the statistics of the measurements, e.g. for two domains the localized parts can be sampled independently which improves the error in the ideal case to $\propto 1/(\sqrt{N}\sqrt{N}) = 1/N$.

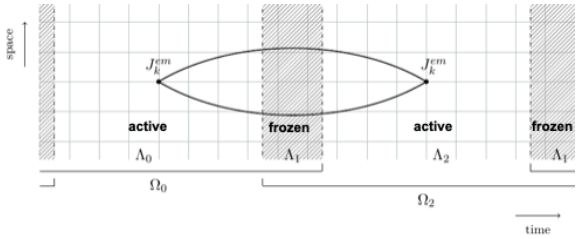

Figure 9: The figure shows the one dimensional decompostion in time used for the multilevel algorithm used in [83].

While multi-level or multi-rate sampling are well-known in case of pure gauge simulations, i.e. they are in general straightfoward to apply if the action as well as the observable can be decomposed in local terms, it becomes hard if the action as well as the observable is a non-local object. This is in general the case for fermionic contributions depending on the determinant or inverse matrix elements. However, the effective lattice action is local, a necessity for continuum limits, in the sense that separated domain decouples with the distances exponentially proportional to the lowest fermionic mode. This potentially makes multi-level algorithms effective if a suitable decomposition is found.

A successful approach is given by a one-dimensional decomposition in time [74, 75]. Here, the domain is decomposed into two *active* regions, denoted by Λ_0 and Λ_2, which are separated by two (one with open boundary conditions) *frozen* domain Λ_1, see Fig. 9. Now, we can decompose the fermion determinant (here $Q = \gamma_5 D$)

$$detQ = \frac{det(1-\omega)}{detQ_{\Lambda_1} detQ^{-1}_{\Omega_0} detQ^{-1}_{\Omega_2}} \tag{17}$$

where Q_{Λ_1} is defined on the frozen domain Λ_1, Q_{Ω_0} on the domain including Λ_0 and Λ_1 and Q_{Ω_2} on the domain including Λ_2 and Λ_1. The global term, which includes the contribution between the active domains Λ_0 and Λ_2 is given by

$$\omega = P_{\partial\Lambda_0} Q^{-1}_{\Omega_0} Q_{\Lambda_{1,2}} Q^{-1}_{\Omega_2} Q_{\Lambda_{2,0}} \tag{18}$$

with the boundary projector $P_{\partial\Lambda_0}$. The global part can be further factorized into a part which can be treated via a multi-boson approach and a global part treated via a reweighting factor, see [75]. Note, that fermionic observable can be decomposed in a similar way, see [74].

Now, we can define a MCMC method, as follows. Starting from a thermalized configuration each active domain is updated via a HMC algorithm localized to Ω_0 and Ω_2 independently from each other n_1 times and on each sample the local part of the observable is measured (note that the global correction enters here as a reweighting factor). Afterwards by including the global corrections based on eq. (18) the global lattice can be updated. At this level also shifts of the lattice are possible. This is followed by global update steps ideally performed until a new global configuration is obtained. If the procedure is performed n_0 times an effective sampling of localized modes with an statistical error $\propto 1/(n_1\sqrt{n_0})$ is obtained.

Alternatively, one can use n_0 configurations of an generated ensemble and perform the local updates on each of them. This leads to an improved error as demonstrated [83] in case of the HVP of magnetic moment of the muon on a lattice with $L = 48$, $a = 0.065$ fm at 300 MeV pions with domain sizes $\Lambda_1 = 8$ and $\Lambda_{0/2} = 40$. In case of the HVP the long distance contribution are notoriously difficult to estimate due to an exponential increase of the noise given by $\frac{\sigma^2_{G^{conn}_{u,d}}(x_0)}{[G^{conn}{}_{u,d}(x_0)]^2} \propto \frac{1}{n_0} e^{2(M_\rho - M_\pi)|x_0|}$. By using a statistics of $n_0 = 25$ configuration separated each by 48 MDUs and

$n_1 = 10$, a precision of the full HVP contribution to 1% could be achieved, see Fig. 10 for the effective error reductions.

Especially in combination with masterfield simulations multilevel approaches have the potential to change the way how observables are measured on the lattice. Despite this, still for many applications the potential is not well understood, e.g. how effective the method becomes if physical pion masses are approached, because the required distance between active domain is increased. Multilevel approaches add an additional level of complexity also on the level of implementations and parallelization. It is well suited for modular supercomputing and to overcome strong scalability issues, compare [17, 97], but an efficient (open source) implementation is so far missing. Python APIs, such as lyncs [40] or GPT [42], are potentially well-suited package, where the complexity of the multi-level algorithm can be managed on the higher level.

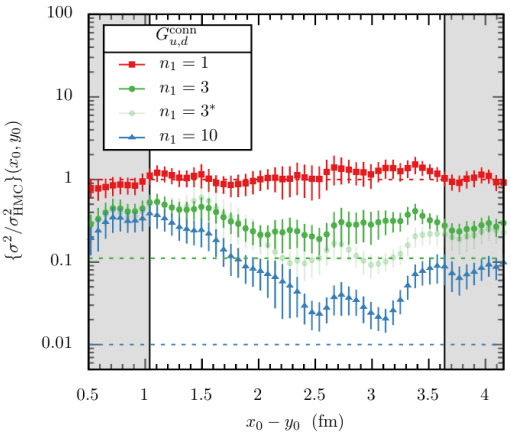

Figure 10: Reduction of the error in $\sigma^2_{G^{conn}_{u,d}}$ using n_1 multilevel steps with respect the distance $x_0 - y_0$ (taken from [83]).

4.2.2 Gauge updates without topological barriers

Let us take a closer look to gauge proposals, which can change topological sectors. As pointed out [18], topological freezing towards fine lattices is connected to the gauge group and is also present if fermions are neglected. Because of the additional costs, coming with fermions, possible gauge proposals are investigated within pure gauge, before fermions are added.

An idea for a solid gauge proposal is to minimise the number of changed variables or gauge links by allowing possible flips of the topological charge. A possible transition in two dimensional U(1)-pure gauge theory, if fermions are present also known as Schwinger model, is given by winding the fields [79–82]

$$U_\mu(x) \to U^\Omega_\mu(x) = \Omega(x) U_\mu(x) \Omega(x+\hat{\mu}) \tag{19}$$

with $\Omega^\pm(x_n) = e^{\pm\frac{\pi}{2}(\frac{n}{L_w}+r)}$ on the domain $L_w \times L_w$ with a random shift r. The proposed configuration is then accepted with the probability $P_{acc}(U \to U') = \min[1, e^{-S[U']+S[U]}]$. In combination with a HMC step after each winding step, the algorithm becomes ergodic.

This was tested in the 2D Schwinger model at a gauge coupling of $\beta = 11.25$ and compared to simulation with a HMC at fix topological charge and a *masterfield* simulation using a large volume with lattice extent $L = 8192$, see Fig. 11.

The acceptance rate of the winding proposal might break down towards very fine lattice, i.e. the phase of the transformation $\Omega^\pm(x_n)$ does not depends on the gauge coupling and likely pushes the gaugefield out of the equilibrium for very fine lattice spacings. Adaptations to implement a similar transformation in SU(3) theories, are so far unsuccessful due to a break down of the acceptance rate, see [69].

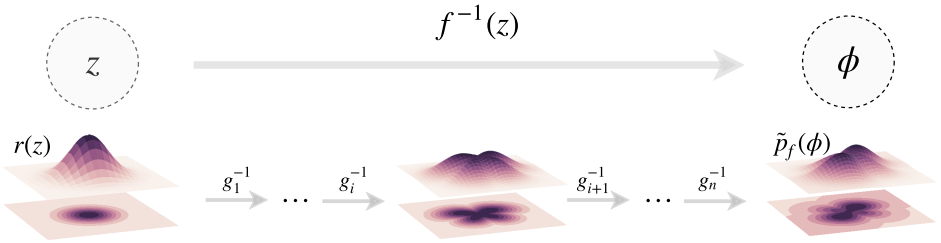

Figure 12: The figure shows the construction of the normalized flow using coupling layers g_i^{-1} in case of the ϕ^4 model (taken from [78], note that $\tilde{p}_f(\phi)$ corresponds here to $q(\phi)$).

A more promising approach for updates in SU(N) theories is given by multi-tempering algorithms [94, 95], successfully applied in four dimensions. The idea is to introduce a defect within the lattice, given by a small cube with links set to zero. The defect enables smooth transition between topological sector, similar to open boundary. To recover the target lattice $j = 0$, without a defect, a weight parameter $w_j = 1 - j/(N_{lvl} - 1)$ is multiplied with the plaquettes located at the defect, where N_{lvl} is the number of levels. Now, we can generate a Markov chain by pairing usual update, like a HMC algorithm, with transition accept-reject steps between different level. If the transition acceptance rate is tuned, which can be done by varying the number of levels, the algorithm can change topological sector smoothly, as demonstrated in [95, 96]. Note, that the computational cost are increased with the number of levels, which will increase the cost towards very fine lattices likely significantly.

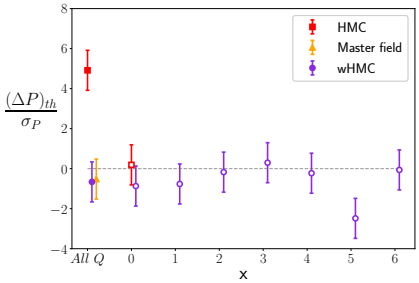

Figure 11: The figure shows the normalized difference between the measured plaquette and the analytical expectation for the different algorithms presented in [82].

4.2.3 Generative models for gauge theories

The acceptance rate of the accept-reject step eq. (2) can be also controlled by making use of correlations between $q(U)$ and $p(U)$. An idea is to use neural networks in combination with machine learning to train the corresponding correlations. A possible approach is given by generative model, which are applied to pure gauge theories using gauge equivariant flows [78, 86–88].

The idea is to use a flow map $f^{-1}(z)$ to propose new configurations with known distribution

$$q(\phi) = r(f(\phi)) \cdot \left| \det \frac{\partial f(\phi)}{\partial \phi} \right| . \tag{20}$$

with $r(z)$ a random distribution. By writing the map $f^{-1}(z)$ as a product over coupling layers

$$g^{-1}(z) = \begin{cases} \phi_a = z_a \\ \phi_b = (z_b - t_i(z_a)) \cdot e^{-s_i(z_a)} \end{cases} \tag{21}$$

which updates a subset of variables ϕ_b using only a set of frozen parameters ϕ_a, the Jacobian is block-diagonal and is simple to compute. In case of the ϕ^4 model one can use an even-odd mapping, where all variables at even points belong to one subset and the variables at the odd points to the other subset. By using gauge invariant objects, e.g. plaquettes instead of gauge links, as in- and output the coupling layers becomes gauge equivariant. Links are than updated with the corresponding plaquettes, see [86].

The neural networks s_i and t_i, e.g. which consist of convolutional kernels with few hidden layers, can be trained by minimizing the loss-function

$$L(q) = D_{KL}(q||p) - \log(Z) = \int \prod_j d\phi_j \, q(\phi)(\log(q(\phi)) + S(\phi)) \, . \tag{22}$$

This can be done by drawing random samples z and by adjusting the weights in the coupling layers. One step is done on a set of samples, also called batch, on which the loss-function can be approximated.

After a sufficient number of training steps, e.g. the effective sample size per configuration $\text{ESS} = 1/N(\sum_{i=1}^N p(\phi_i)/q(\phi_i))^2/\sum_{i=1}^N (p(\phi_i)/q(\phi_i))^2$ reaches a relative high values, the trained gauge equivarient map $f^{-1}(z)$ can be used to draw proposals for gauge configurations, which can be used in MCMC algorithms or in weighted averages. By combining the proposal with an accept-reject step $P_{acc}(U \to U') = \min[1\ , q(U)p(U')/p(U)q(U')]$ we obtain an exact algorithm, which samples configuration distributed with $p(U)$.

Generative models are successfully applied to 2D discrete lattice models, such as U(1) theory [86] or SU(3) [87]. Decorrelation between proposed gauges are included in the design, i.e. they are maximally decorrelated because each initial set is drawn from a random distribution. Larger autocorrelation times can only occur due to a small total acceptance rate.

Generative models are a promising novel way to model physics distribution and have the potential to give new insights into QCD. In contrast to other machine learning applications a combination with an accept-reject step leads to an exact algorithm.

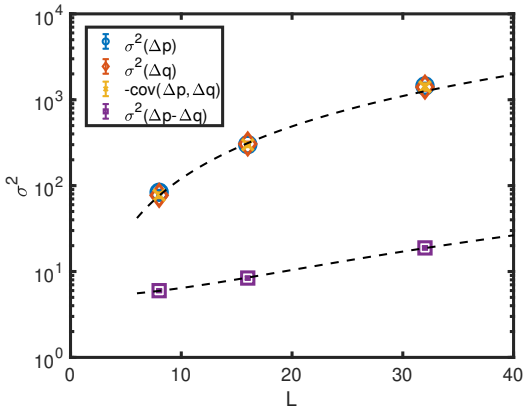

Figure 13: The variances of the log-distribution obtained by training gauge equivariant maps in the 2D U(1) in dependence of the volume is shown. The number of coupling layers are increased with the lattice extent and the training is stopped after the improvements of the optimizer deterioted.

In case of the 2D-U(1) model, the minimization of the loss function leads to a minimization of the volume fluctuations, as shown in Fig. 13, while the overall scaling is still proportional to $\propto V$. This can be understood as a fine tuning problem. Let us assume that the distribution are log-normal distributed, than it follows $\langle P_{acc} \rangle \approx 1 - \frac{\sigma}{\pi}$ for $\sigma \ll 1$ with the variance $\sigma^2 = \text{var}(\Delta p) + \text{var}(\Delta q) + 2\text{cov}(\Delta p, \Delta q)$. As shown in Fig. 13 we found that the covariance scales like the variances with $\text{var}(\Delta p) + \text{var}(\Delta q) \approx -2\text{cov}(\Delta p, \Delta q)$. While the major part of the fluctuations cancels out, the remaining part still increases with the volume. This illustrate a general problem of the approach, the scalability towards larger volumes,

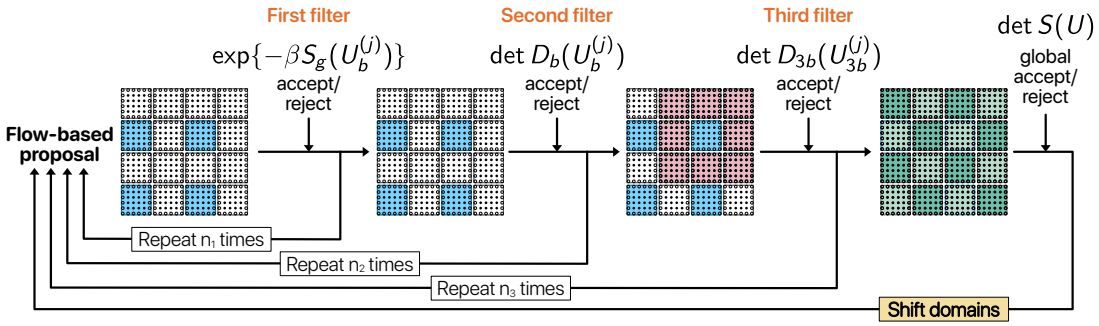

Figure 14: The steps of the 5 level flowGC algorithm, which iterates starting from the left with a flow-based proposal for the blocks highlighted in blue, followed by nested accept-reject filter steps. In the last step a global correction step is done, followed by a shift of the domains before restarting

e.g. it is not well understood how to reach larger acceptance rates for $L > 32$ in case of 2D U(1) model. A more complete discussion of the phenomenon can be found in a recent publication [92].

However this topic, how to achieve scalable generative models, is under active research and different approaches are considered. Possible research directions are given by investigation of various optimizations of the maps, e.g. by modifying neural networks or using different flows, or by modification of the flow maps, e.g. using continues flows [98].

4.2.4 Domain decomposed normalizing flows and fermions

Another possibility to overcome the volume scaling in generative models is to make use of physical properties of the theory, in particular by using localization, e.g. as discussed by factorising the action using domain decomposition.

The idea is to train normalizing flows for local gauge updates within a domain. This can be done by restricting the flow map and its corresponding coupling layers to variables within the domain. As shown in the 2D-U(1) model [91], the flow can be trained in a similar fashion to the periodic case [88]. By freezing boundary terms, updates within a domain are decoupled from the other domains and each domain can be updated independently, i.e. updates are independent from the global volume. Note that to overcome topological freezing the local updates need to enable topological transition.

To generate configurations with dynamical fermions the fermion determinant has to be included. This can be done by using accept-reject steps, similar to [73]. The general idea is to filter out UV fluctuations step by step using nested hierachical filter steps including correlation between different parts. Let us write the Boltzmann factor as a product over factorized distributions $p(U) = \prod_j^n P_j(U)$ with the distribution of the jth step

$$\rho_j(U) = P_0(U, \alpha_i^{(0)}) P_1(U, \alpha_i^{(1)}) \dots P_j(U, \alpha_i^{(j)}) \,. \tag{23}$$

The terms in $p(U)$ can be ordered with respect to the their range, i.e. P_0 contains ultra local actions parts like the pure gauge action, P_1 contains local action but short range interactions like block determinants while P_n includes the global Schur complement, which contains long range

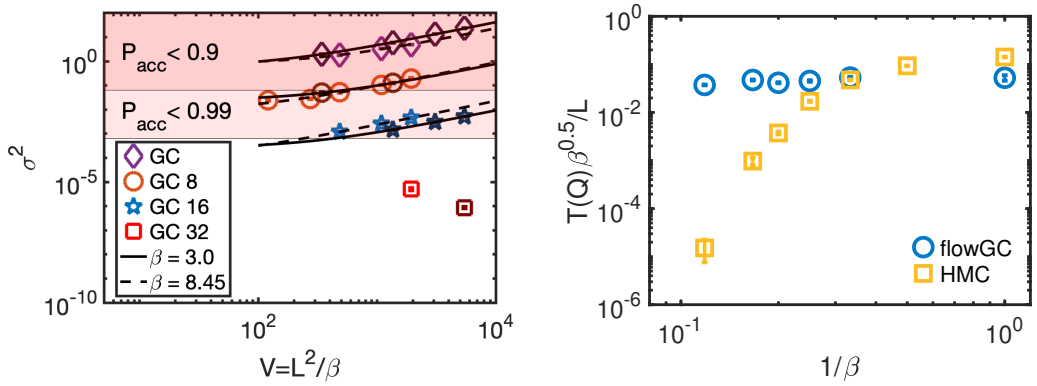

Figure 15: The acceptance rate of the global correction step is shown at constant rate of update links for different distances between active domains is shown in the left figure. On the right side the tunneling rate of the global steps of the flowGC is shown at a constant line of physics with $m_{PS}\sqrt{\beta} = 4$ and $L/\sqrt{\beta} \sim 40$.

interactions. Now, the accept-reject step of the jth-step is given by

$$P^j_{acc}(U \to U') = \min\left[1\,, \frac{\rho_{j-1}(U)\rho_j(U')}{\rho_{j-1}(U')\rho_j(U)}\right]\,. \tag{24}$$

The introduced parameters $\alpha_i^{(j)}$ can be tuned such that the acceptance rate $\langle P^j_{acc}\rangle$ starting from $j = n, \ldots, 0$ is maximized.

A MCMC chain is then given by nested hierachical filter steps with gauge updates on domains giving by flow proposals. A 4-level approach, denoted as flowGC, is illustrated in Fig. 14 and is given by

0. Flow proposal to generate N_0 samples within each active block with lattice extent $l = 8$.
1. Accept/reject step over the N_0 samples using the pure gauge action of the active blocks as target probability and keeping the final accepted configuration.
2. Calculation of the determinant of the block operator $D(U_j)$ with $L_b = n \cdot l$ with $n \in 1, 2, \ldots$ and accept/reject. Repeat, starting from step 0., until a sufficient number of links are updated.
3. Calculation of the determinant of the extended $3L_b \times 3L_b$ Dirac operator and perform an accept/reject step. Repeat, starting from step 0., until a sufficient number of domains are updated.
4. Calculation of Schur complement term performing a global accept/reject step correcting to the target probability.

This is sufficient to highly suppress fermionic fluctuations, if the active domains are separated by a distance of $d = 32$ very high acceptance rate were obtained, see left panel of Fig. 15 where the domain extent l is set to the distance of the active domains d to keep the number of active links fixed at different separations d and lattice volumes [91]. By using a domain size of $l = 8$ the flowGC can sample sufficiently topological sectors, as depicted on the right panel of Fig. 15, in case of the 2D Schwinger model at a constant line of physics with $m_{PS}\sqrt{\beta} = 4$ and $L/\sqrt{\beta} \sim 40$ and updates of 16% of the links.

Towards larger and more complex systems, the second and third filters become potentially a bottleneck by likely developing low acceptance rates. This effect could be mild down by flow updates which also takes into account factors of the factorised fermion weight.

4.2.5 Flows with fermions

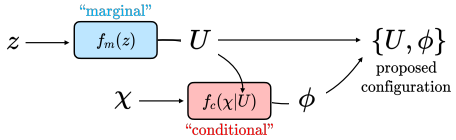

Figure 16: The diagram shows the procedure for training flows for pseudofermion and gaugefields (taken from[90]).

The gauge equivariant flow models in subsec. 4.2.3 were originally developed for pure gauge systems. Including the fermion weight increased drastically the computational effort, i.e. requires frequently computation of the determinant during the training of the weights. Rewriting the fermion determinant via the pseudofermions integral will decrease computational load to the inverse of the matrix, however, if treated as a stochastic estimate similar to reweighting [3], will add large stochastic fluctuations. In general this could be tamed by similar methods which increase the accept-reject steps of eq. (2).

To include fermions within the flows several approaches where investigated, see [93]. A possible way is to enable fermion contributions by sampling pseudofermions [90]. The general idea is to factorize the distribution into

$$p(U,\phi) = p(U)p(\phi|U) \quad \text{with} \quad p(U) \propto \det DD^\dagger(U) e^{-S_g(U)} \quad \text{and} \quad p(\phi|U) \propto \frac{e^{-S_{pf}(U,\phi,\phi^\dagger)}}{\det DD^\dagger(U)} \tag{25}$$

where $p(U)$ is denoted as the marginal distribution and $p(\phi|U)$ the conditional distribution, see Fig. 16. For the training of the networks the distributions are splited accordingly with $q(U,\phi) = q(U)q(\phi|U)$. Now, the idea is to train first the marginal to generate a set of configurations $\{U\}$. In a second step the pseudofermion map $f_c(\chi|U)$ is trained on the constant set $\{U\}$. The flow maps for the pseudofermions require a new design of maps, i.e. gauge covariance of the networks can be implemented by using parallel transporter to approximate $D(U)$.

To further improve the sampling IR/UV-filtering techniques, such as even-odd reductions or Hasenbusch-mass preconditioning, can be introduced, i.e as depicted in Fig. 17.

First results for full QCD in four dimensions with two mass-degenerated fermions using the discussed procedures are presented in [90]. Here, the normalizing flows were used in case of a simulation for a lattice with extent $L = 4$ and a mass value of $\kappa = 1$ at a coupling constant $\beta = 1.0$.

Obviously additional steps are needed in order to apply normalizing flows at the production level in full QCD calculations. In general this requires a better understanding on how to scale up the systems. Sampling directly pseudofermion fields can be used to improve other applications, e.g. in stochastic estimation of the fermions in reweighting [3].

5. Conclusion

Advances of algorithms for dynamical fermions in the last decades enable simulation of ensembles at physical pion masses. To take the next steps towards larger volumes with $L = 8$ fm,

computing time and software package to utilize the pre-exa and exascale machine are available although scalability on novel architectures is currently missing.

On the other hand efficient algorithm to unfreeze topology are under development but not available for large lattices, except for using open boundary conditions. Due to that reaching lattices of size $L = 128$ at fine lattice spacings of $a = 0.04$ fm need additional algorithmic advances. Several ideas are investigated where possible solutions could be given by a combination of update steps, which allow transitions between topological sectors, followed by HMC updates.

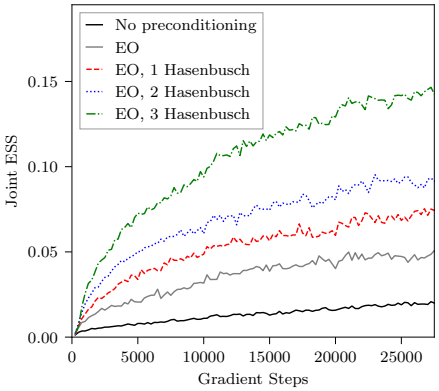

Figure 17: The figure shows the effect of using Hasenbusch mass preconditioning in case of generating pseudofermions in case of ESS (taken from [90]).

To make continuous advances flexible software solutions are required, in order to develop but also deploy algorithms using efficiently state-of-art HPC systems. In general high level packages, which can utilize modularity of systems and deal with flexible parallelizations, can be based on python APIs, such as GPT [42] or lyncs [40, 41], which already enable access to highly optimized lattice kernels of the packages grid and QUDA, respectively. This can be very useful for next steps in combining machine learning approaches with lattice QCD algorithms.

To conclude, the community is very active in investigating very diverse approaches and ideas, which will push the frontiers of our understanding of fundamental physics in the future.

Acknowledgments. The author gratefully thanks the committee of the 39th International Symposium on Lattice Field Theory for the honour to give a review on algorithms for dynamical fermions. The author thanks for all the effort given to support this work, in particular, special thanks goes to C. Alexandrou, S. Bacchio, F. Knechtli and G. Koutsou. The author thanks all members of the ETMC for an enjoyable collaboration. J.F. received financial support by the German Research Foundation (DFG) research unit FOR5269 "Future methods for studying confined gluons in QCD", by the PRACE Sixth Implementation Phase (PRACE-6IP) program (grant agreement No. 823767) and by the EuroHPC-JU project EuroCC (grant agreement No. 951740) of the European Commission. Results were obtain on various Supercomputers like Juwels-Booster at JSC, Hawk at HLRS, SuperMUC at LRZ, Frontera at TACC.

References

[1] M. Creutz, Phys. Rev. D **38** (1988), 1228-1238 doi:10.1103/PhysRevD.38.1228

[2] M. Luscher, [arXiv:1002.4232 [hep-lat]].

[3] J. Finkenrath, F. Knechtli and B. Leder, Nucl. Phys. B **877** (2013), 441-456 [erratum: Nucl. Phys. B **880** (2014), 574-575] [arXiv:1306.3962 [hep-lat]].

[4] S. Aoki *et al.* [CP-PACS], Phys. Rev. D **67** (2003), 034503 [arXiv:hep-lat/0206009 [hep-lat]].

[5] S. Duane, A. D. Kennedy, B. J. Pendleton and D. Roweth, Phys. Lett. B **195** (1987), 216-222

[6] S. A. Gottlieb, W. Liu, D. Toussaint, et al.,Phys. Rev. D **35** (1987), 2531-2542

[7] M. Hasenbusch, Phys. Lett. B **519** (2001), 177-182 [arXiv:hep-lat/0107019 [hep-lat]].

[8] M. A. Clark and A. D. Kennedy, Phys. Rev. Lett. **98** (2007), 051601 [arXiv:hep-lat/0608015]

[9] T. Lippert, Lect. Notes Comput. Sci. **15** (2000), 166-177 [arXiv:hep-lat/0007029 [hep-lat]].

[10] Y. C. Chen *et al.* [TWQCD], Phys. Lett. B **738** (2014), 55-60 [arXiv:1403.1683 [hep-lat]].

[11] G. Bali, R. Bignell, et al. PoS **LATTICE2022** (2022), 203 [arXiv:2212.10138 [hep-lat]].

[12] N. Husung, P. Marquard and R. Sommer, Eur. Phys. J. C **80** (2020) no.3, 200

[13] N. Husung, [arXiv:2212.09626 [hep-lat]].

[14] M. Cè, T. Harris, H. B. Meyer, A. Toniato and C. Török, JHEP **12** (2021), 215

[15] S. Borsanyi, Z. Fodor, J. N. Guenther, *et al.*, Nature **593** (2021) no.7857, 51-55

[16] P. A. Boyle, B. Chakraborty, C. T. H. Davies, *et al.* [arXiv:2205.15373 [hep-lat]].

[17] P. Boyle, D. Bollweg, R. Brower, *et al.* [arXiv:2204.00039 [hep-lat]].

[18] S. Schaefer *et al.* [ALPHA], Nucl. Phys. B **845** (2011), 93-119 [arXiv:1009.5228 [hep-lat]].

[19] C. Bernard *et al.* [MILC], Phys. Rev. D **97** (2018) no.7, 074502 [arXiv:1707.05430 [hep-lat]].

[20] A. Yamaguchi, P. Boyle, *et al.* PoS **LATTICE2021** (2022), 035 [arXiv:2203.06777 [hep-lat]].

[21] P. A. Boyle, G. Cossu, A. Yamaguchi and A. Portelli, PoS **LATTICE2015** (2016), 023

[22] M. A. Clark, R. Babich, K. Barros, *et al.* Comput. Phys. Commun. **181** (2010), 1517-1528

[23] K. Clark, **LATTICE2022** (2022), Software development and Machines / 430

[24] T. Aoyama, I. Kanamori, K. Kanaya, et al., PoS **LATTICE2022** (2023), 284

[25] D. Richtmann, N. Meyer and T. Wettig, [arXiv:2211.13719 [hep-lat]].

[26] M. A. Clark, A. Strelchenko, et al., Comput. Phys. Commun. **233** (2018), 29-40

[27] M. Luscher, JHEP **12** (2007), 011 [arXiv:0710.5417 [hep-lat]].

[28] M. Luscher, JHEP **07** (2007), 081 [arXiv:0706.2298 [hep-lat]].

[29] R. Babich, J. Brannick, R. C. Brower, *et al.* Phys. Rev. Lett. **105** (2010), 201602

[30] A. Frommer, K. Kahl, S. Krieg, *et al.* SIAM J. Sci. Comput. **36** (2014), A1581-A1608

[31] C. Alexandrou, S. Bacchio, J. Finkenrath, *et al.* Phys. Rev. D **94** (2016) no.11, 114509

[32] R. C. Brower, M. A. Clark, *et al.* Phys. Rev. D **97** (2018) no.11, 114513

[33] V. Ayyar, R. Brower, M. A. Clark, et al.,[arXiv:2212.12559 [hep-lat]].

[34] B. Joo, D. D. Kalamkar et al., (Springer International Publishing, Cham, 2016) pp. 415–427.

[35] R. C. Brower, M. A. Clark, D. Howarth *et al.* Phys. Rev. D **102** (2020) no.9, 094517

[36] P. Boyle and A. Yamaguchi, [arXiv:2103.05034 [hep-lat]].

[37] S. Yamamoto *et al.* [PRACE], PoS **LATTICE2021** (2022), 536 [arXiv:2201.03872 [hep-lat]].

[38] J. Espinoza-Valverde, A. Frommer, G. Ramirez-Hidalgo *et al.* [arXiv:2205.09104 [math.NA]].

[39] B. Kostrzewa, S. Bacchio, J. Finkenrath, M. Garofalo, *et al.* [arXiv:2212.06635 [hep-lat]].

[40] S. Bacchio, **LATTICE2022** (2022), Software development and Machines / 304

[41] S. Yamamoto, S. Bacchio and J. Finkenrath, PoS **LATTICE2022** (2023), 346

[42] C. Lehner, **LATTICE2022** (2022), Software development and Machines / 78

[43] J. C. Sexton and D. H. Weingarten, Nucl. Phys. B **380** (1992), 665-677

[44] Omelyan, Mryglod, and Folk, Phys. Rev. Lett. 86(5), 898. (2001).

[45] T. Takaishi and P. de Forcrand, Phys. Rev. E **73** (2006), 036706 [arXiv:hep-lat/0505020]

[46] A. D. Kennedy, P. J. Silva and M. A. Clark, Phys. Rev. D **87** (2013) no.3, 034511

[47] P. de Forcrand and L. Keegan, Phys. Rev. E **98** (2018) no.4, 043306

[48] L. Keegan, BlockCG. https://github.com/lkeegan/blockCG

[49] H. Yin and R. D. Mawhinney, PoS **LATTICE2011** (2011), 051 [arXiv:1111.5059 [hep-lat]].

[50] M. Luscher and S. Schaefer, JHEP **07** (2011), 036 [arXiv:1105.4749 [hep-lat]].

[51] S. Cali, K. Eckert, J. Heitger, F. Knechtli and T. Korzec, Eur. Phys. J. C **81** (2021) no.8, 733

[52] R. Brower, S. Chandrasekharan, J. W. Negele and U. J. Wiese, Phys. Lett. B **560** (2003), 64-74

[53] C. Czaban and M. Wagner, [arXiv:1310.5258 [hep-lat]].

[54] M. Lüscher, EPJ Web Conf. **175** (2018), 01002 [arXiv:1707.09758 [hep-lat]].

[55] D. Albandea, P. Hernández, A. Ramos *et al.* Eur. Phys. J. C **81** (2021) no.10, 873

[56] P. Fritzsch, **LATTICE2022** (2022), Plenaries / 54

[57] C. Urbach, K. Jansen, A. Shindler and U. Wenger, Comput. Phys. Commun. **174** (2006), 87-98

[58] D. Shcherbakov, M. Ehrhardt, *et al.* Commun. Comput. Phys. **21** (2017) no.4, 1141-1153

[59] W. Detmold and M. G. Endres, Phys. Rev. D **94** (2016) no.11, 114502 [arXiv:1605.09650]

[60] W. Detmold and M. G. Endres, Phys. Rev. D **97** (2018) no.7, 074507 [arXiv:1801.06132]

[61] J. Tu and R. Mawhinney, EPJ Web Conf. **175** (2018), 02006

[62] T. Nguyen, P. Boyle, *et al.* PoS **LATTICE2021** (2022), 582 [arXiv:2112.04556 [hep-lat]].

[63] J. Pinto Barros and M. Krstic Marinkovic, **LATTICE2022** (2022), Algorithms / 436

[64] M. Lüscher, Commun. Math. Phys. **293** (2010), 899-919 [arXiv:0907.5491 [hep-lat]].

[65] P. Boyle, T. Izubuchi et al.,PoS **LATTICE2022** (2023), 229 [arXiv:2212.11387 [hep-lat]].

[66] D. Albandea, L. del Debbio, P. Hernández, R. Kenway, *et al.* [arXiv:2211.12806 [hep-lat]].

[67] S. Foreman, T. Izubuchi, L. Jin, *et al.* PoS **LATTICE2021** (2022), 073 [arXiv:2112.01586]

[68] A. Hasenfratz, Y. Shamir and B. Svetitsky, Phys. Rev. D **104** (2021) no.7, 074509

[69] T. Eichhorn, C. Hoelbling, P. Rouenhoff and L. Varnhorst, [arXiv:2210.11453 [hep-lat]].

[70] T. Eichhorn and C. Hoelbling, PoS **LATTICE2021** (2022), 573 [arXiv:2112.05188 [hep-lat]].

[71] S. Foreman, X. Y. Jin and J. C. Osborn, PoS **LATTICE2021** (2022), 508 [arXiv:2112.01582]

[72] M. Luscher, Comput. Phys. Commun. **165** (2005), 199-220 [arXiv:hep-lat/0409106 [hep-lat]].

[73] J. Finkenrath, F. Knechtli and B. Leder, Comput. Phys. Commun. **184** (2013), 1522-1534

[74] M. Cè, L. Giusti and S. Schaefer, Phys. Rev. D **93** (2016) no.9, 094507

[75] M. Cè, L. Giusti and S. Schaefer, Phys. Rev. D **95** (2017) no.3, 034503

[76] U. Wenger and P. Buehlmann, Algorithms / 413

[77] A. C. Irving and J. C. Sexton, Phys. Rev. D **55** (1997), 5456-5473 [arXiv:hep-lat/9608145]

[78] M. S. Albergo, G. Kanwar and P. E. Shanahan, Phys. Rev. D **100** (2019) no.3, 034515

[79] J. Smit and J. C. Vink, Nucl. Phys. B **286** (1987), 485-508 doi:10.1016/0550-3213(87)90451-2

[80] D. B. Leinweber, A. G. Williams, J. b. Zhang and F. X. Lee, Phys. Lett. B **585** (2004), 187-191

[81] S. Durr, Phys. Rev. D **85** (2012), 114503 [arXiv:1203.2560 [hep-lat]].

[82] D. Albandea, P. Hernández, A. Ramos *et al.* Eur. Phys. J. C **81** (2021) no.10, 873

[83] M. Dalla Brida, L. Giusti, T. Harris and M. Pepe, Phys. Lett. B **816** (2021), 136191

[84] L. Giusti and M. Saccardi, Phys. Lett. B **829** (2022), 137103

[85] M. Saccardi and L. Giusti, [arXiv:2211.06902 [hep-lat]].

[86] G. Kanwar, M. S. Albergo, D. Boyda, *et al.* Phys. Rev. Lett. **125** (2020) no.12, 121601

[87] D. Boyda, G. Kanwar, *et al.* Phys. Rev. D **103** (2021) no.7, 074504 [arXiv:2008.05456]

[88] M. S. Albergo, D. Boyda, *et al.* [arXiv:2101.08176 [hep-lat]].

[89] R. Abbott, M. S. Albergo, *et al.* [arXiv:2208.03832 [hep-lat]].

[90] R. Abbott, M. S. Albergo, D. Boyda, *et al.* Phys. Rev. D **106** (2022) no.7, 074506

[91] J. Finkenrath, [arXiv:2201.02216 [hep-lat]].

[92] R. Abbott, M. S. Albergo, A. Botev, *et al.* [arXiv:2211.07541 [hep-lat]].

[93] M. S. Albergo, G. Kanwar, S. Racanière, *et al.* Phys. Rev. D **104** (2021) no.11, 114507

[94] M. Hasenbusch, Phys. Rev. D **96** (2017) no.5, 054504 [arXiv:1706.04443]

[95] C. Bonanno, C. Bonati and M. D'Elia, JHEP **03** (2021), 111 [arXiv:2012.14000 [hep-lat]].

[96] C. Bonanno, M. D'Elia, B. Lucini and D. Vadacchino, Phys. Lett. B **833** (2022), 137281

[97] P. A. Boyle, D. Bollweg, C. Kelly and A. Yamaguchi, PoS **LATTICE2021** (2022), 470

[98] S. Bacchio, P. Kessel, S. Schaefer and L. Vaitl, [arXiv:2212.08469 [hep-lat]].

[99] J. Finkenrath, C. Alexandrou, *et al.* PoS **LATTICE2021** (2022), 284 [arXiv:2201.02551]

[100] C. Alexandrou, S. Bacchio, *et al.* Phys. Rev. D **98** (2018) no.5, 054518 [arXiv:1807.00495]

[101] M. Hasenbusch, Phys. Rev. D **97** (2018) no.11, 114512 [arXiv:1805.03560 [hep-lat]].