National Computational Infrastructure for Lattice Gauge Theory
SciDAC-2 Closeout Report

A proposal in response to Office of Science Notice DE-FG02-06ER06-04 and Announcement Lab 06-04: Scientific Discovery through Advanced Computing.

Lead Institution: Fermilab
Batavia, IL 60510

Lead Principal Investigator and DOE Contact: Paul Mackenzie
Address: Theoretical Physics Department
Fermilab
Batavia, IL 60510
Email: mackenzie@fnal.gov
Phone: 630-840-3347

Office of Science Programs Addressed: High Energy Physics and Nuclear Physics

Office of Science Program Office Technical Contacts: Lali Chatterjee, Ted Barnes, and Randall Laviolette

Participating Institutions and Principal Investigators:

Physics:

Boston University*, Richard Brower † ‡
Brookhaven National Laboratory*, Frithjof Karsch † ‡
Columbia University, Norman Christ †
Fermi National Accelerator Laboratory*, Paul Mackenzie † ‡
Indiana University*, Steven Gottlieb ‡
Massachusetts Institute of Technology*, John Negele † ‡
Thomas Jefferson National Accelerator Facility*, David Richards †‡
University of Arizona*, Doug Toussaint ‡
University of California, Santa Barbara*, Robert Sugar † ‡
University of Utah*, Carleton DeTar ‡
University of Washington, Stephen Sharpe †

Computer Science:

DePaul University*, Massimo DiPierro ‡
Illinois Institute of Technology*, Xian-He Sun ‡
University of North Carolina*, Rob Fowler ‡
Vanderbilt University*, Abhishek Dubey ‡

* Institution submitting an application
† Project Principal Investigator, Member of the USQCD Executive Committee
‡ Institution Principal Investigator

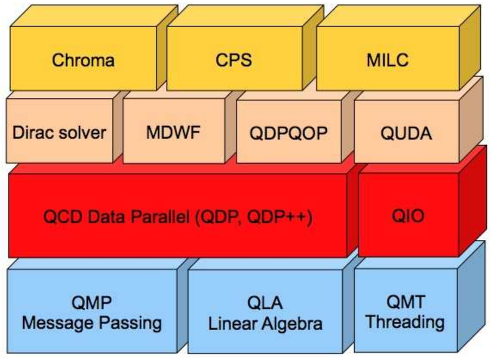

Figure 1: Levels of the QCD API

Under its SciDAC-1 and SciDAC-2 grants, the USQCD Collaboration developed software and algorithmic infrastructure for the numerical study of lattice gauge theories. This work was carried out jointly by high energy and nuclear physicists within USQCD, in collaboration with applied mathematicians and computer scientists. The software and its documentation is publicly available at the USQCD software web site *http://www.usqcd.org/usqcd-software*. The code has been widely adopted within the United States, and is used extensively abroad. It has been instrumental in our effective use of leadership class computers, and of the dedicated computers funded for USQCD through the LQCD Computing Project. The committees which review our hardware program on a yearly basis have consistently emphasized the importance of the work done under our SciDAC grants, and the need for their continuation.

1 Project SciDAC-2 Close Out Report

1.1 The QCD Applications Programming Interface

Under our SciDAC-1 and SciDAC-2 grants, the USQCD Collaboration created the QCD Applications Programming Interface (QCD API), a unified programming environment that enables its users to quickly adapt existing codes to new architectures, easily develop new codes and incorporate new algorithms, and preserve their large investment in existing codes. It has greatly facilitated the efficient use of leadership class computers and commodity clusters. The QCD API was developed as a layered structure which is implemented in a set of independent libraries. It is illustrated in Fig. 1, which shows the three levels of the API and the application codes that sit on top of them. Extensions to these libraries and the maintenance of the API code is an ongoing activity as computer architectures and algorithms change. The API is a critical software underpinning for all of our community application codes, which requires maintenance, testing, version control, documentation and distribution. During SciDAC-2, the components of the API that had been developed in SciDAC-1 were ported and optimized for new architectures (as described in this subsection) and extensions for new architectures and new algorithms were added (as described in the next subsection.)

Level 1 of the API provides the code that controls communications and the core single processor computations. To obtain high efficiency, sometime much of this layer has to be written in hardware specific assembly language; however, versions exist in C and C++ using MPI for transparent portability of all application codes.

Message Passing: QMP defines a uniform subset of MPI-like functions with extensions that (1) partition the QCD space-time lattice and map it onto the geometry of the hardware network, providing a convenient abstraction for the Level 2 data parallel API (QDP); (2) contain specialized communication routines designed to access the full hardware capabilities of computers, such as the Blue Gene line, and to aid optimization of low level protocols on cluster networks. New versions are developed as needed to accommodate changing architectures and algorithms. For example, as discussed below, hooks to combine message passing and threaded code are being added, as is the ability to work with multiple lattice geometries, which is needed for multigrid and domain decomposition algorithms.

Linear Algebra: All lattice QCD calculations make use of a set of linear algebra operations in which the basic elements are three–dimensional complex matrices, elements of the group SU(3). These operations are local to lattice sites or links, and do not involve inter-processor communications. The C implementation has about 19,000 functions generated in Perl, with a full suite of test scripts. The C++ implementation makes considerable use of templates, and so contains only a few dozen templated classes (the required specific classes are generated on demand by the compiler). For both C and C++ it is important to optimize the code for the most heavily used linear algebra modules.

Data Parallel Interface: Level 2 (QDP) contains data parallel operations that are built on QLA and QMP. QDP allows extensive overlapping of communication and computation in a single line of code. By making

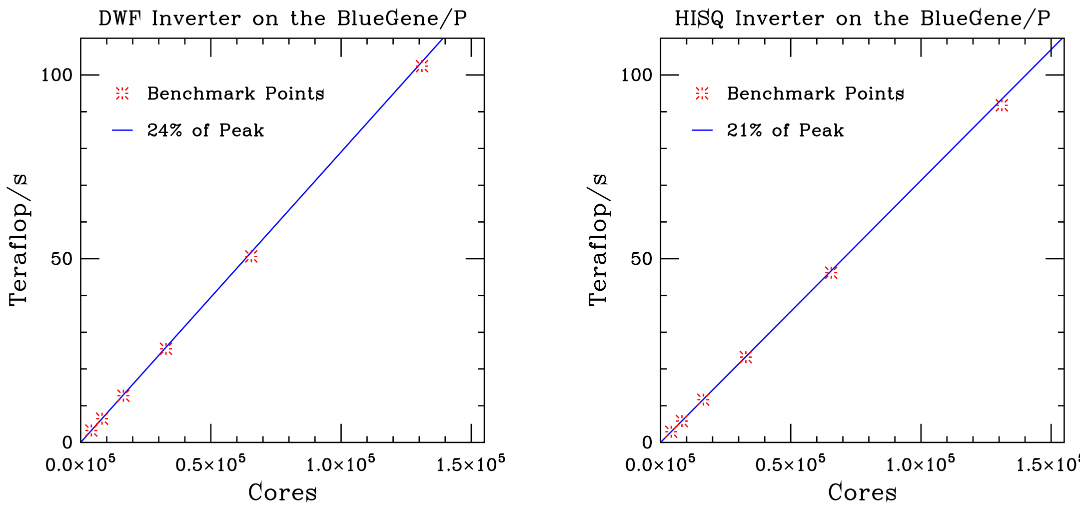

Figure 2: Performance of the Dirac solver on the Blue Gene/P in Tflops as a function of the number of cores for DWF quarks (left panel) and HISQ quarks (right panel). The red bursts are the benchmark points, and the solid blue lines indicate 24% and 21% of peak, respectively. These are weak scaling tests with the number of lattice points per core being held fixed at 6^4 for the DWF solver, and 8^4 for the HISQ solver.

use of the QMP and QLA layers, the details of communications buffers, synchronization barriers, vectorization over multiple sites on each node, etc. are hidden from the users, allowing them to focus on the physics, rather than the subtleties of parallel programming. QDP significantly accelerates the process of developing new codes and optimizing existing ones. It also lowers barriers for entry into the field by graduate students, postdoc and senior scientists from other fields.

Optimized Subroutines: Level 3 (QOP) consists of highly optimized code for a limited number of subroutines that consume a large fractions of the resources in any lattice gauge theory calculation. Most notable among these is the subroutine for the solution of the linear, sparse matrix equations involving the Dirac operator discussed in Sect. 3. To obtain the level of performance at which we aim, it is necessary to optimize these subroutines for each architecture. These routines are generally written with extensive assembly language coding, either employing hand coding or specialized tools, such as Bagel [1] and QA(0) [2], which were developed to generate optimized codes. The data mapping and cache efficiency is extensively tuned. In Fig. 2 we show the performance of the Dirac solver for DWF and HISQ quarks on the Blue Gene/P.

Data Management: QIO enables users to read and write the different types of files that arise in our work in standard formats. It supports a logical partitioning of the computer into I/O partitions with one core per partition handling I/O for the data in just that partition. Thus, in a suitable files system our codes can read and write data in parallel from/to a single file, or in any file system from/to multiple files, and these files can be flattened into one large one offline on a single processor machine. There are no unusual memory requirements for this process. By tuning the size of the I/O partitions, we can maximize the I/O bandwidth and avoid contention. In order to maximize the physics output from the very large computational resources that go into the generation of gauge configurations, we share all gauge configuration files that are created with USQCD resources. To enable this sharing we have created standards for file formats, which QIO adheres to. In addition, we are charter members of the International Lattice Data Grid (ILDG), which established a basic set of meta-data and middleware standards to enable international sharing of data [3, 4], which are also adhered to by QIO.

Application Codes: There are three large, publicly available application code suites developed by members of USQCD that take advantage of the QCD API. Chroma was built directly on QDP++, while the Columbia Physics System (CPS) and the MILC code predate the API, but incorporate key features of it. As a result, all three applications suites benefit immediately from any extensions to or optimizations of the QCD API. Among them, these suites contain all of the codes required for the QCD configuration generation and measurement campaigns we intend to carry out over the next three years. The application code suites and their documentation can also be found at the USQCD software web site.

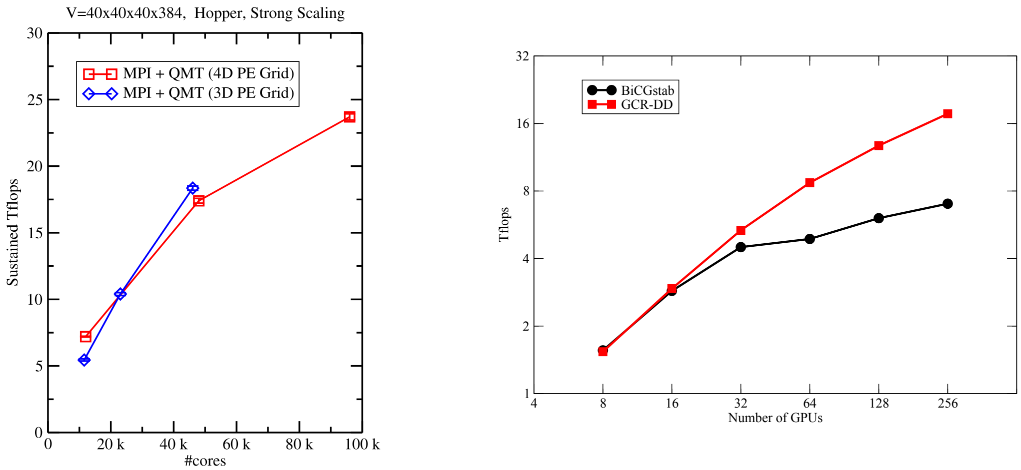

Figure 3: Strong scaling tests of the Wilson-clover inverter with threaded codes. On the left, results on the NERSC Cray XE6, Hopper, using hybrid code with MPI and the QMT library; and on the right, results on the LLNL Edge cluster for up to 256 GPUs with MPI and CUDA threads.

1.2 Recent Extensions of the QCD API

Although the QCD API and the application codes are highly portable, as we move to new computers, we typically have to upgrade the Level 1 and QOP routines. The advent of computer nodes with large numbers of cores and the use of GPU accelerators on the nodes have required that we develop threaded versions of our codes. Furthermore, we, and others in our field, regularly develop new algorithms which must be integrated into the API and the application codes. These developments require that we continually upgrade and extend the QCD API. Here we give a few highlights of this phase of our work.

Hybrid MPI/Threaded Code: It seems clear that in the near future computer nodes will contain large numbers of cores, and that for such machines we will need to employ a hybrid programming model in which communication between nodes is programmed in MPI or QMP, and work on nodes is performed with threaded code. At this early stage we do not believe that a "one size fits all" approach is possible, so we are experimenting with a variety of them. We have obtained early access to the Blue Gene/Q because members of our collaboration at Columbia University and Brookhaven National Laboratory, and our international collaborators at the University of Edinburgh, worked with colleagues at IBM on its design. They are well along in the development of code for domain wall fermions, and find that a hybrid MPI/OpenMP approach works well. They have also produced a highly optimized DWF solver using the Bagel tool [1], which produces assembly code for the Blue Gene/Q's PowerPC processors. Similar code for HISQ and Wilson-clover quarks will follow. For GPU accelerators, we are using CUDA threads on the GPUs combined with POSIX threads on the CPU, and MPI between nodes, while for computers with Intel and AMD multi-core processors, such as the Cray XE series, we have implemented a new threaded library, QMT. Our long range goal is to provide a single uniform data parallel interface so that the applications programmer does not need to be aware of the details of the hybrid code. In Fig. 3 we show strong scaling results for threaded code on NERSC's Cray XE6, Hopper, and on the Edge cluster at LLNL.

The QUDA GPU Library: Starting in 2008, we have explored high performance Dirac solvers in CUDA on NVIDIA GPUs [5]. This effort was initially supported by NSF funding, but has rapidly expanded into a major SciDAC project with the development of the QUDA (QCD in CUDA) library [6, 7, 8], and the rapid deployment of GPU accelerated clusters at Jefferson Laboratory and Fermilab. Our ability to respond rapidly to this new architecture demonstrates the advantage of our clear factorization of Level 3 solvers in the QCD API. At present the QUDA library has expanded to include all Dirac solvers used in QCD (Wilson-Clover, HISQ/asqtad, domain wall and twisted mass). The result has been a dramatic improvement in price/performance for a range of analysis work that is dominated by Dirac solvers. The most recent advance has been the extension of code from single to multiple GPUs. The multiple-GPU codes enables us to analyze the full set of lattices sizes generated by USQCD members with excellent weak scaling. In a

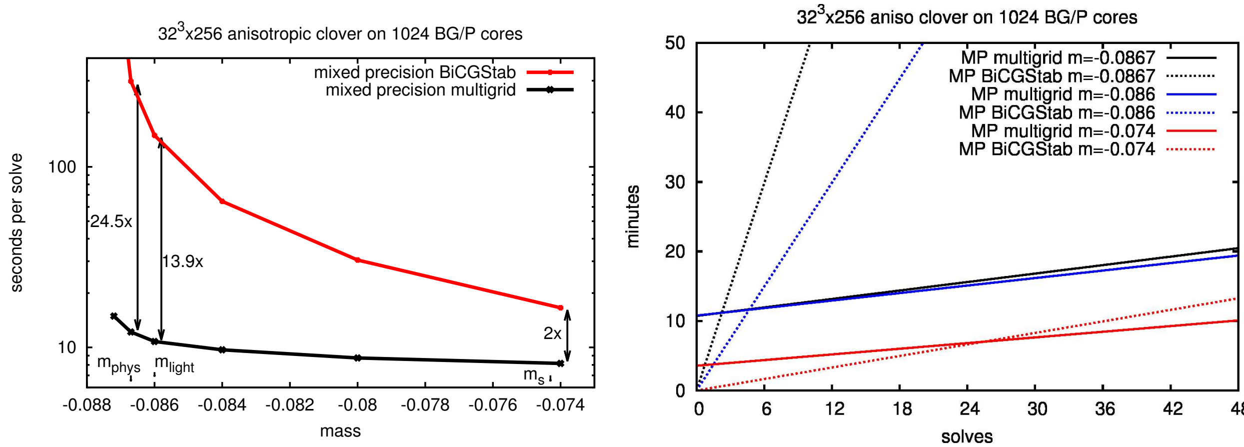

Figure 4: The left panel shows the marginal wall-clock per solve for the multigrid algorithm compared with our best BiCGStab Krylov solver on the Blue Gene/P for a $32^3 \times 256$ lattice with the Wilson-Clover Dirac operator. The right panel shows the total time including the setup for multiple solves on the same configuration by the multigrid and BiCGStab algorithms as a function of the number of solves [9].

paper presented to Super Computing 2011 we demonstrated that we have achieved good strong scaling on up to 256 GPUs for the HISQ/asqtad and Wilson-Clover solvers running on the Edge cluster at LLNL [9].

Solvers for the Dirac Operator: The solver for the lattice Dirac operator has traditionally been a dominant focus of algorithm and specialized software development because of its central role in all QCD codes. A large variety of Krylov solvers have been used with the conjugate gradient and BiCGStab being the current work horses in many production codes. Data layout to improve cache behavior and hand coded assembly kernels are commonplace. For example the Möbius [10, 11] domain wall fermion (MDWF) solver uses Morton ordering in its internal data representation [2], and the QUDA code employs specializes mappings and a novel mixed precision schemes from half (16 bits) to single (32 bits) to double (64 bits) in order to provide double precision accuracy with reduced data traffic between the processor and the memory. A new area of activity beginning to show great promise is the use of multigrid methods [12]. Lattice gauge theorists have attempted to apply multigrid methods to QCD for over twenty years [13]. In collaboration with applied mathematicians from TOPS, we have finally succeeded in formulating an adaptive multigrid solver for Wilson-clover [14]. In the left hand panel of Fig. 4, we show the speedup in the time for one additional solution provided by the multigrid solver compared with our best BiCGStab Krylov solver – nearly a 25x speed up as we move to the physical light quark mass limit. The multigrid algorithm has an overhead to construct its preconditioner, and in the right hand panel of Fig. 4 we show the number of solves with different right hand sides needed to amortize this overhead sufficiently so that the multigrid solver outperforms the BiCGStab one. In some measurement routines, such as those involving disconnected diagrams, hundreds of solves are required on each configuration, so the multigrid algorithm already offers a major improvement over BiCGStab. For the physical light quark mass, the crossover occurs for two or three solves, leading to the possibility of using multigrid in our configuration generation work. This is the beginning of a new opportunity for multi-level algorithms for other parts of our code, and will become increasingly important as the quark masses are reduced and lattice sizes increased. In this same spirit, we are exploring and implementing a variety of "deflation" and Schwarz domain decomposition methods [15].

Improved Hybrid Monte Carlo Evolution: Besides the Dirac solvers, the other major consumer of floating point operations in lattice field theory codes is the symplectic integrator for the molecular dynamics equations that arise in the hybrid Monte Carlo algorithms used to generate gauge ensembles. Over the period of the SciDAC grants, a major advance has been the development of the Rational Hybrid Monte Carlo (RHMC) [16], which is implemented in all of our major application codes. It typically results in a two to four times speedup in the generation of gauge configurations. An even higher order symplectic Force Gradient integrator has been designed [17], which promises further improvements in the next generation of gauge configurations on very large lattices.

A Appendices

A.1 References Cited

[1] P. Boyle, *The Bagel web site*, *http:www2.ph.ed.ac.uk/ paboyle/bagel/Bagel.html*.

[2] A. Pochinsky, *Writing efficient QCD code made simpler: QA(0)*, PoS **LATTICE2008**, 040 (2008).

[3] M. G. Beckett, B. Joo, C. M. Maynard, D. Pleiter, O. Tatebe and T. Yoshie, *Building the International Lattice Data Grid*, Comput. Phys. Commun. **182** (2011) 1208 [arXiv:0910.1692 [hep-lat]].

[4] See International Data Lattice Grid (ILDG) website: *http://www.usqcd.org/ildg/*

[5] K. Barros, R. Babich, R. Brower, M. A. Clark and C. Rebbi, *Blasting through lattice calculations using CUDA*, PoS **LATTICE2008**, 045 (2008) [arXiv:0810.5365 [hep-lat]].

[6] M. A. Clark, R. Babich, K. Barros, R. C. Brower and C. Rebbi, *Solving Lattice QCD systems of equations using mixed precision solvers on GPUs*, Comput. Phys. Commun. **181**, 1517 (2010) [arXiv:0911.3191 [hep-lat]].

[7] S. Gottlieb, G. Shi, A. Torok and V. Kindratenko, *QUDA programming for staggered quarks*, PoS **LATTICE2010**, 026 (2010).

[8] R. Babich, M. A. Clark and B. Joo, *Parallelizing the QUDA Library for Multi-GPU Calculations in Lattice Quantum Chromodynamics*, arXiv:1011.0024 [hep-lat].

[9] R. Babich, M. A. Clark, B. Joó, G. Shi, R. C. Brower and S. Gottlieb, Proceedings of the 2011 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11), 11 (2011), arXiv:1109.2935.

[10] R. C. Brower, H. Neff and K. Orginos, *Moebius fermions: Improved domain wall chiral fermions*, Nucl. Phys. Proc. Suppl. **140** (2005) 686 [arXiv:hep-lat/0409118].

[11] R. C. Brower, H. Neff and K. Orginos, *Moebius fermions*, Nucl. Phys. Proc. Suppl. **153**, 191 (2006) [arXiv:hep-lat/0511031].

[12] J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn and C. Rebbi, *Adaptive Multigrid Algorithm for Lattice QCD*, Phys. Rev. Lett. **100**, 041601 (2008) [arXiv:0707.4018 [hep-lat]].

[13] R. Babich, J. Brannick, R. C. Brower, M. A. Clark, T. A. Manteuffel, S. F. McCormick J. C. Osborn,and C. Rebbi, *Adaptive multigrid algorithm for the lattice Wilson-Dirac operator*, Phys. Rev. Lett. **105**, 201602 (2010) [arXiv:1005.3043 [hep-lat]].

[14] J. C. Osborn, R. Babich, J. Brannick, R. C. Brower, M. A. Clark, S. D. Cohen and C. Rebbi, *Multigrid solver for clover fermions*, PoS **LATTICE2010**, 037 (2010) [arXiv:1011.2775 [hep-lat]].

[15] M. Luscher, *Solution of the Dirac equation in lattice QCD using a domain decomposition method*, Comput. Phys. Commun. **156**, 209 (2004) [arXiv:hep-lat/0310048].

[16] M.A. Clark, A.D. Kennedy and Z. Sroczynski, *Exact 2+1 flavour RHMC simulations*, Nucl. Phys. (Proc. Suppl.) **140**, 835 (2005); M.A. Clark, Ph. de Forcrand and A.D. Kennedy, *Algorithm Shootout: R versus RHMC*, PoS LAT2005, 115 (2005).

[17] A. D. Kennedy, M. A. Clark and P. J. Silva, *Force Gradient Integrators*, PoS **LAT2009**, 021 (2009) [arXiv:0910.2950 [hep-lat]].

DOE Final Report
Project: DE-FC02-06ER41441

Massimo Di Pierro

April 6, 2012

```
DOE Award Number: DE-FC02-06ER41441
Name of Recipient: Massimo Di Pierro
Principal Investigator: Massimo Di Pierro
Consortium Name: USQCD Collaboration
```

1 License

2 Executive Summary

Our research project is about the development of visualization tools for Lattice QCD. We developed various tools by extending existing libraries, adding new algorithms, exposing new APIs, and creating web interfaces (including the new NERSC gauge connection web site). Our tools cover the full stack of operations from automating download of data, to generating VTK files (topological charge, plaquette, Polyakov lines, quark and meson propagtors, currents), to turning the VTK files into images, movies, and web pages. Some of the tools have their own web interfaces. Some Lattice QCD visualization have been created in the past but, to our knowledge, our tools are the only ones of their kind since they are general purpose, customizable, and relatively easy to use. We believe they will be valuable to physicists working in the field. They can be used to better teach Lattice QCD concepts to new graduate students; they can be used to observe the changes in topological charge density and detect possible sources of bias in computations; they can be used to observe the convergence of the algorithms at a local level and determine possible problems; they can be used to probe heavy-light mesons with currents and determine their spatial distribution; they can be used to detect corrupted gauge configurations. There are some indirect results of this grant that will benefit a broader audience than Lattice QCD physicists and explained below.

3 Achievements and Comparison of Goals

The goal of this research project is the creation of a visualization toolkit that could be used to aid physicists in the analysis of data from lattice QCD computations. This goal has been successfully achieved as descibed below.

Links to code, images and video can be found at:

`http://latticeqcd.org`

The original grant proposal stated:

> Lattice QCD computations comprise multiple steps, creating very large datasets, but the final result is typically encompassed in a small set of numbers with the analysis performed in an automated way. While an automated procedure may be beneficial in efficiency, the ability to visualize the data being analyzed is important both as an aid to the analysis, and as a means of acquiring insight into the physics. [...]
>
> Crucial to the success of the graphics-visualization initiative will be a close collaboration between physicists to devise and interpret visualization of physically important quantities, and computer scientists to provide the appropriate visualization toolbox. Questions that visualization might address are many: can we understand how flux-tube formation observed with infinitely heavy quarks extends to hadrons where one or more of the quarks is light; what is the distribution of charge within a nucleon; can we display the distribution of spin and magnetism within a hadron? In the longer term, can we visualize the interactions of hadrons? Currently, no general-purpose package is available tailored to the display of lattice data. Thus a software package will be developed with a general GUI capable of reading a set of four-dimensional lattice quantities, and taking their ensemble average; performing a projection into a real four-dimensional vector; interpolating the 4-D vector into a continuous four-dimensional field; taking

three-dimensional slices of a four-dimensional field; displaying the data using density plots, iso-surfaces, and 2-D projections; and displaying the evolution of data, both in simulation time for four-dimensional quantities, and as the evolution of three- and two-dimensional slices in the remaining coordinates. The software will support two types of plug-ins: type-1 plug-ins that perform specific physics measurements and output a real 4-D vector, and type-2 plug-ins that take the interpolated 3-D field and generate specific types of plots.

Most of the research underlying this project will consist of identifying a set of physical measurements suitable to be implemented as type-1 plug-ins. The visualization techniques for the type-2 plug-ins are very similar to standard techniques used for representation of 3-D geophysical data and, when possible, we will incorporate existing libraries into the development of our plug-ins.

The system will be developed in C++ and take advantage of existing graphics and visualization libraries such the Trolltech QT libraries and the Visualization Tool Kit (VTK) library. The plug-ins will be callable from C or C++ code conforming to the QCD API, and will form another component of our Level 4 QCD Toolbox. The system will be capable of reading datasets in the SciDAC/ILDG format and the MILC format.

[...]

DePaul University will lead the design and development of a visualization tool for lattice QCD. Work will be done in collaboration with physicists involved in the project and with computer scientists at the University of North Carolina. The goals for the first year of the project are to identify and catalog the types of datasets to be visualized, identify appropriate smoothing and visualization algorithms, and develop a prototype interface. In subsequent years, plugins will be developed to read in the various types of datasets produced in lattice QCD simulations, and tools for manipulating the data in increasingly sophisticated ways will be created. A total of 1.08 FTE per year is budgeted for this effort.

A QCD physics toolbox will be constructed which will contain sharable software building blocks for inclusion in application codes, performance analysis and visualization tools, and software for automation of physics work flow. New software tools will be created for managing the large data sets generated in lattice QCD simulations, and for sharing them through the International Lattice Data Grid consortium.

Our basic toolkit consists of two parts. The first part has been implemented in the form of C++ libraries which are now included within the FermiQCD toolkit which is part of the USQCD Software Suite. These API allow the project of arbitrary fields into 3D and 4D scalar

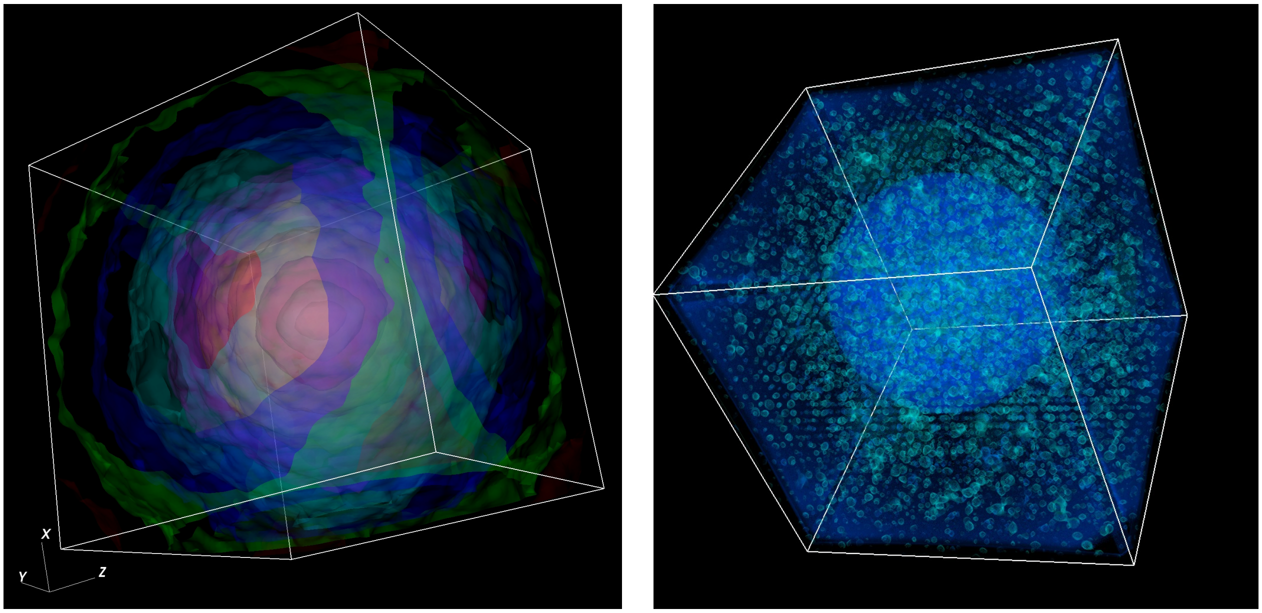

Figure 1: Example Images. The left one shows iso-surfaces for a quark wave function. The right one shows the density of HMC hits in presence of a lump of topological charge.

fields with can be visualized using open source toolkits like VisIt, Paraview, and MayaVi. The toolkit includes algorithms to project topological charge density, plaquette, Polyakov line components, quark propgators, meson propagators, and current insertions. It provides an API to create other custom operators and quark contractions and project/visualize them too. We also modified the minimum residue and the stabilized bi-conjugate inverter so that it is possible to observe the spatial effect of this algorithms and visualize their convergence for arbitrary sources. Our choice of the FermiQCD toolkit is motivated by the critical need to be able to perform visualizations for arbitrary $SU(N)$ gauge groups. FermiQCD is the only lattice QCD code, part of the USQCD Software Suite, that at this time supports abitrary gauge groups.

The second part of our toolkit consists of a collection of Python programs that interface with the C++ programs and make the system more accessible to scientists by implementing a typical workflow. Specifically we developed 7 different programs.

The first program (as required by the original grant proposal) has the ability to convert MILC/ILDG data (as well other data formats) into the format required for visualization. On top of that, as described below, the same program has the ability to download gauge configurations from the NERSC repository, which is the largest public repository of gauge configurations in the United States.

The second program is the main interface to the C++ algorithms. The program provides command line options to run physical algorithms such as the computation of the topological change density, plaquette, Polyakov, lines, quark propagators, meson propagators, currents, 4-quark operators. The program downloads, compiles and runs the requested algorithms. Each algorithm provides the option to save the computation steps in the VTK format for visualization. Not all the C++ algorithm are accessible via this interface and some requires explicit programming. Yet the provided code and documentation should be a sufficient example for the scientists to write their own customized code for other particular cases.

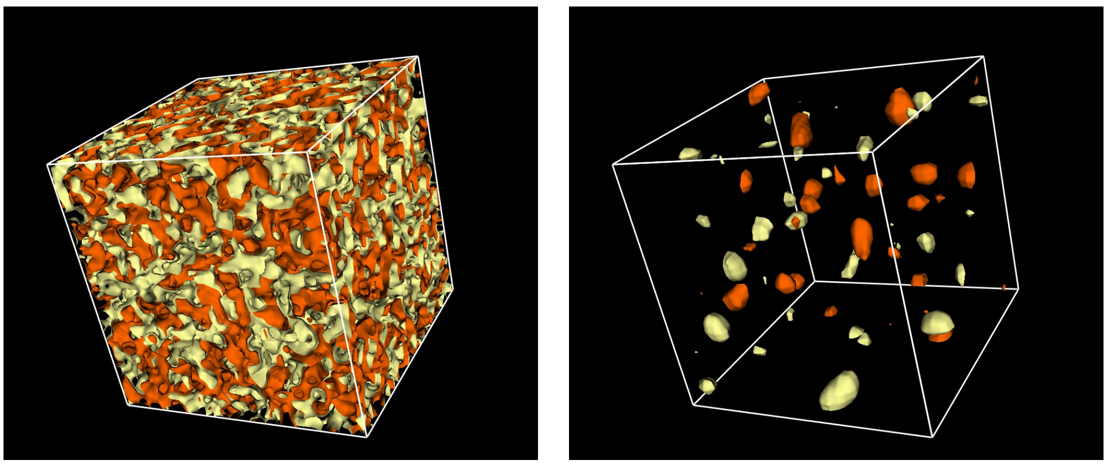

Figure 2: Example Images: They show the visualization of the topological charge at different cooling steps.

The thrid program perform the tasks of extracting information from the VTK files generated by the previous step, resampling them (to achieve better resolution in visualization), interpolating them (to make smooth visualizations and movies) and generating VisIt scripts. The VisIt visualization toolkit was developer by the Lawrence Livermore National Laboratory and it is a critical part of our workflow. It can be accessed via a GUI or programmatically via Python. Using a GUI to automate a workflow and process many files at once is not practical. It must be done programmatically. Our program generates VisIt scritps using meta-programming techniques so that QCD phycisicts do not have to code. By running these scripts scientists can directly obtain images and movies without programming.

The fourth program we develop allows to convert 3D VTK files into interactive web pages. The program opens the VTK files, identifies optimal tresholds for the iso-surfaces, computes said iso-surfaces and generates a 3D representation of the polygons in JavaScript. The output consists of static HTML files which embed the 3D objects. They can be visualized with any browser and the viewer can rotate them with the mouse. While VisIt is a general purpose tool is very powerful which allows many more maniputaions of the data, the possibility of generating 3D interactive web pages opens the possibility for scientists to view the data without installing VisIt and to publish the data on the web for other people to see.

The other programs we created are beyond the scope of the original grant proposal but we felt they were necessary and part of a broader interpretation of the conept of visualization. In fact, not all visualizations are spatial visualizations. There is other information that is important to visualize, which often takes the forms of simple 2D plots but often is not looked at because of the extra work involved in doing so. Examples are autocorrelations between physical measurements on different gauge configurations, moving averages, distributions of bootstrap samples. Our other programs serve this purpose. One of the programs read the output of typical QCD algorithms, extract the numerical results for each gauge configuration, and combines them into a bootstrap analysis. The user can specify the expression to bootstrap using command line arguments without programming. The code generates CSV files storing data for the intermediate steps of the computation. The other two programs can read those CSV files and generate plots and fits from them automatically. CSV files can also be read by many third party analysis and visualization programs.

Consider for example the computation of a four-quark matrix element. It involves the computation of a two-point and a three-point correlation funciton for each gauge configuration and their bootstrap analysis. Our programs can perform this analysis in the standard way but they also generate autocorrelation plots for each two-point and three-point correlation function, moving averages for the ratio on different time-slices, and bin the distribution of the bootstrap samples.

Those programs have been documented in long technical document attached to this report [1].

The main obstacle to this research has been accessibility. Visualization is indeed useful not but the way QCD physicists normally approach lattice QCD computations. We have therefore put lots of extra work in making our programs accessible by creating web interfaces that could simplify the task. Another obstacle is that no computing time was allocated to this project. While this did not prevent us from achieving the task of developeing the tools, it did not allow us to move beyond the original stated goal and utilize the tools for obtaining more abituious scientific results which would have been computing intensive.

Nevertheless we have used our toolkit to produce scientific results. Specifically we collaborated with Chulwoo Jung at Columbia University, Mike Clark and Richard Brower at Boston Universty and looked at the autocorrelation of topological charge density over short HMC trajectories [15]. It is a well known problem that global topological charge has a long autocorrelation. We found that the local topological charge instead has very short autocorrelation and therefore there is no measurable bias in production gauge configurations.

We also were able to observe the effect of a single instanton on a quark propagator and how its presence gives mass to the quark by increasing the exponential fall off of the propagator [1].

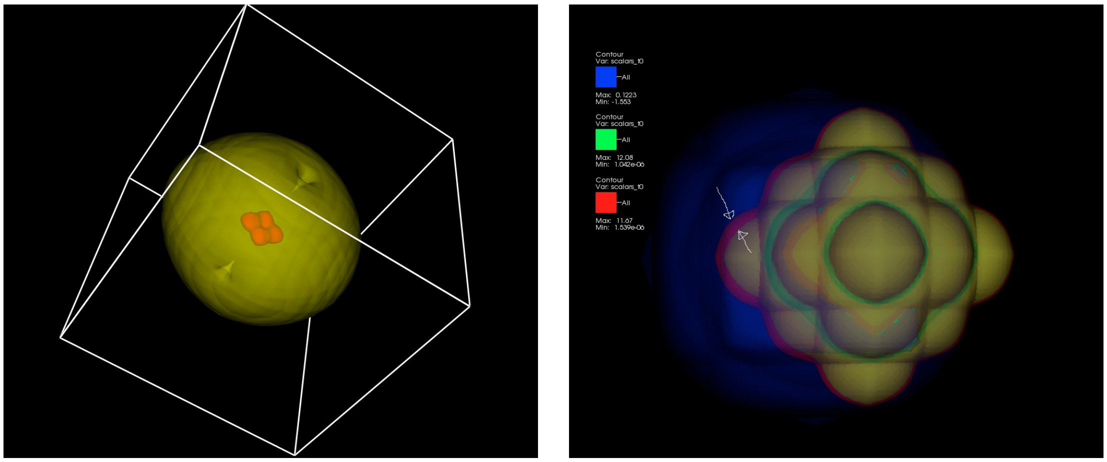

Figure 3: Example Images: left left image shows the density of a current inserted between to meson operators. The right image shows the shriking of a quark propagator (red) in presence of a localized topological charge (blue).

Anyway, that is beyond the scope of the current grant and more visualizations will be done in the future as computing time becomes available.

In our original proposal we stated that our tools would have a GUI based on the Qt toolkit. In the very early stages of our project we have revised that decision. On the one side Nokia, owner of Qt, decided to cut support for the library. On the other site it became evident that desktop GUI have became an obsolete technology giving way to modern web based interfaces. We have therefore put lots of extra work in this direction and we have created three web applications.

The first web application (nersc) [3] was developed in collaboration with the National Energy Research Scientic Center (NERSC) to replace their previous interface to the lattice QCD archive known as “gauge connection”. The new system allows searching for gauge configuration, visualize statistics, and collaborate online by editing metadata in a wiki format. The system also allows batch downloads of the data using the program we described above. We have used our tookit to process many ensembles from the NERSC archive and generated movies of the topological charge density. This program is designed to be very general purpose and it can be used by communities other than Lattice QCD to publish their data online.

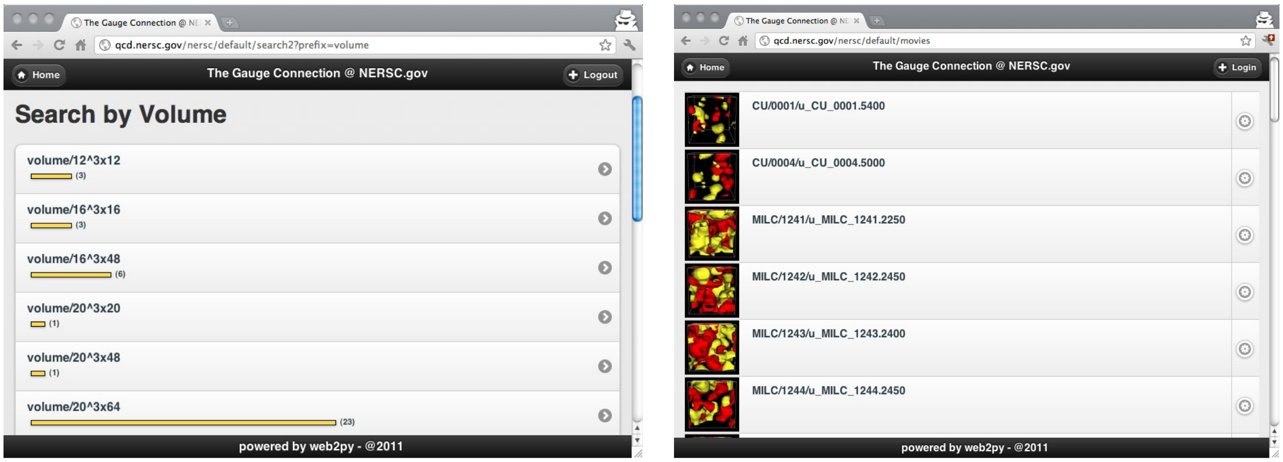

Figure 4: Screenshots from the NERSC “gauge connection” web site.

The second web application (vis) [6, 8] provides an interface to the algorithms and allows upload of gauge configurations and schedule visualization algorithms to run in backgroud. The system provides a web interface to the Portable Batch System (PBS) and to VisIt and. It can schedule both computations and visualizations. The results are displayed in a web page.

The third web application (mc4qcd) [7, 9] is an interface to our analysis and plotting tools. It allows users to upload the log file result of a physics runs, to extract data from it using pattern matching, and to perform bootstrap analysis. The reults of the analysis are stored and published online together with the replative plots (including autocorrelations, moving averages, distribution of bootstrap samples, and fits). Scientists in a group can track results and collaborate online by sharing data and comments.

In order to develop these tools, in the eraly stages of the project we have developed a set of libraries for creating online scientific applications called web2py. This project is not part of the goal of this grant but it turned out to be an important and necessary component to replace the obsolete GUI concept with the modern web based paradigm. This project took a life on its own and found applications beyound this physics project. It was released open source and it is now used by thousands of users and businesses worldwide. It won the Bossie Award for “best open source software development tool” in 2011 and the InfoWorld Best Technology of the Year Award in 2012. This provides an example of unexpected broader impact of DOE funded research. Although we consider this very important and we are proud of the result, since it falls outside the scope of the original grant, we omitted references to it in the rest of this report. Yet references are available upon request.

Figure 5: Screenshots from the VIS and the MC4QCD web applications respectively.

In 2010 we also contributed organize the 6th High End Visualization workshop in Obergugl, Austria.

Some of the visualization created with our tools were used for the opening video for the Lattice 2011 conference: `http://vimeo.com/25242353`

3.1 Summary of Code Created

The code written as part of this grant is published in the following repositories:

- `http://code.google.com/p/qcdutils/` It is the main toolkit, documented in [1].
- `http://code.google.com/p/fermiqcd/` This is a pre-existing C++ library for lattice QCD computations part of the USQCD Software Suite. The core visualization algorithms have been included here and distributed together. They are accessible via QCDUTILS.
- `http://code.google.com/p/nersc-data-publisher/` This is code behind the new NERSC “gauge connection” web interface. Develped in collaboration with James Hetrick (University of Pacific) and David Skinner (NERSC).
- `http://code.google.com/p/qcdvis/` This is a web interface to the visualization algorithms.

- `http://https://launchpad.net/qcdmc` This is a web interface to our analsys and plotting tools.

3.2 Published Web Sites

- `http://latticeqcd.org` This is the main front-end where we have published links to the code and some of our visualizations. More will be published as our tools is put into production. This site includes web interfaces to VIS and MC4QCD.
- `http://qcd.nersc.gov` This is the new NERSC "gauge connection" archive (developed in collaboration with NERSC). It also stores some videos created using our tools.
- `http://tests.web2py.com/ildg` This is a new proposed web site for the International Lattice Data Grid. It provides an interface for searching lattiec QCD data in a visual way.

3.3 Fostered Collaborations

During this research project we have collaborated with Prof. James Hetrick from the University of Pacific and David Skinner at NERSC to re-build the new "gauge connection" web site.

We collaborated with Prof. Werner Berger from the Center for Computation and Technology, Louisiana State University and together we organized the 6th High End Visualization Workshop.

We utilized the VisIt software created by the Lawrence Livermore National Laboratory. Although we did not interact directly with the authors we interacted inderectly by using various online resources generated for that project.

Finally we interacted with the rest of the USQCD and the ILDG collaborations, from which we received constant feedback and suggestions.

3.4 Personnel

This grant has funded the PI and some of the following graduate students, who contributed to this research:

- Nate Wilson
- Yaoquan Zhong
- Brian Schinazi

- Tony Garcia
- Chris Baron
- Vincent Havery

3.5 Published Papers

In the following bibliography we list all the papers published by the PI and supported directly by this research grant. We omitted papers and books published by the PI on other topics not directly related to the grant scope.

References

[1] M. Di Pierro "qcdutils" arXiv:1202.4813 [hep-lat]

[2] M. Di Pierro "Improving Non-Linear Fits" arXiv:1202.0988 [cs.NA]

[3] M. Di Pierro, J. Hetrick, S. Cholia, D. Skinner "Making QCD Lattice Data Accessible and Organized through Advanced Web Interfaces" PoS **LAT2011** (2011) arXiv:1112.2193 [hep-lat]

[4] E. T. Neil *et al.* [for the Fermilab Lattice Collaboration and for the MILC Collaboration], "B and D meson decay constants from 2+1 flavor improved staggered arXiv:1112.3978 [hep-lat]

[5] A. Bazavov *et al.* [Fermilab Lattice and MILC Collaborations], "B- and D-meson decay constants from three-flavor lattice QCD," arXiv:1112.3051 [hep-lat]

[6] M. Di Pierro, Y. Zhong and B. Schinazi, "Vis: Online analysis tool for lattice QCD," PoS **LATTICE2010**, 326 (2010).

[7] M. Di Pierro and Y. Zhong, "Analysis and visualization tools for Lattice QCD," PoS **LAT2009**, 038 (2009).

[8] M. Di Pierro, "Vis: QCD workflow and visualization tool" in Proceedings of the 6th High End Visualization Workshop, Dec 8th - 12th, (2010), Obergurgl (Austria), Lehmanns Media, ISBN 978-3-86541-361-1

[9] M. Di Pierro, Y. Zhong and B. Schinazi, "mc4qcd: Online Analysis Tool for Lattice QCD," PoS A **CAT2010**, 054 (2010) arXiv:1005.3353 [hep-lat]

[10] C. Bernard *et al.* [Fermilab Lattice Collaboration and MILC Collaboration], "Tuning Fermilab Heavy Quarks in 2+1 Flavor Lattice QCD with Application to Phys. Rev. D **83**, 034503 (2011) arXiv:1003.1937 [hep-lat]

[11] A. Bazavov *et al.* [Fermilab Lattice and MILC Collaborations], "The Ds and D+ Leptonic Decay Constants from Lattice QCD," PoS **LAT2009**, 249 (2009) arXiv:0912.5221 [hep-lat]

[12] T. Burch *et al.*, "Quarkonium mass splittings in three-flavor lattice QCD," Phys. Rev. D **81**, 034508 (2010) arXiv:0912.2701 [hep-lat]

[13] J. A. Bailey *et al.* [Fermilab Lattice Collaboration and MILC Collaboration], "Progress on charm semileptonic form factors from 2+1 flavor lattice QCD," PoS **LAT2009**, 250 (2009) arXiv:0912.0214 [hep-lat]

[14] T. Burch *et al.*, "Quarkonium mass splittings with Fermilab heavy quarks and 2+1 flavors of PoS **LAT2009**, 115 (2009) arXiv:0911.0361 [hep-lat]

[15] M. Di Pierro *et al.*, "Visualization as a tool for understanding QCD evolution algorithms," J. Phys. Conf. Ser. **180**, 012068 (2009).

[16] C. Bernard *et al.*, "Visualization of semileptonic form factors from lattice QCD," Phys. Rev. D **80**, 034026 (2009) arXiv:0906.2498 [hep-lat]

[17] C. Bernard *et al.*, "B and D Meson Decay Constants," PoS **LATTICE2008**, 278 (2008) arXiv:0904.1895 [hep-lat]

[18] J. A. Bailey *et al.*, "The B —¿ pi l nu semileptonic form factor from three-flavor lattice QCD: Phys. Rev. D **79**, 054507 (2009) arXiv:0811.3640 [hep-lat]

[19] C. Bernard *et al.*, "The Anti-B $\to$ D* l anti-nu form factor at zero recoil from three-flavor Phys. Rev. D **79**, 014506 (2009) arXiv:0808.2519 [hep-lat]

[20] M. Di Pierro, "A Visualization Toolkit for Lattice Quantum Chromodynamics", in Proceedings of the 4th High End Visualization Workshop, June 18th - 22th, (2007), Obergurgl (Austria), Lehmanns Media, ISBN 9783865412164

[21] C. Bernard *et al.* [Fermilab Lattice, MILC and HPQCD Collaborations], "The decay constants f(B) and f(D+) from three-flavor lattice QCD," PoS **LAT2007**, 370 (2007).

[22] R. Todd Evans, E. Gamiz, A. X. El-Khadra and M. Di Pierro, "A Determination of the B0(s) and B0(d) mixing parameters in 2+1 lattice PoS **LAT2007**, 354 (2007) arXiv:0710.2880 [hep-lat]

[23] M. Di Pierro, "Visualization for Lattice QCD" PoS **LAT2007**, 331 (2007) `http://pos.sissa.it/archive/conferences/042/031/LATTICE2007_031.pdf`