# 4. Large-scale Parallel Numerical Computing Technology Research Team

## 4.1. Team members

Toshiyuki Imamura (Team Leader)

Yoshio Okamoto (Visiting Researcher)

Yukiko Hayakawa (Assistant)

## 4.2. Research Activities

The Large-scale Parallel Numerical Computing Technology Research Team conducts research and development of a large-scale, highly parallel and high performance numerical software library for the K computer. Simulation programs require various numerical algorithms for the solution of linear systems, eigenvalue problems, fast Fourier transforms, and non-linear equations. In order to take advantage of the full potential of the K computer, we must select algorithms and develop a numerical software library based on the concepts of high parallelism, high performance, high precision, resiliency, and scalability.

Our primary mission of this project is to develop a highly parallelized and scalable numerical library on the K computer system, namely KMATHLIB. It comprises several components such as for solving

1) System of linear equations,
2) Eigenvalue problems,
3) Singular value decomposition,
4) Fast Fourier transforms, and
5) Nonlinear equations.

The K-specific topics are also our challenging works as follows;

a) Tofu interconnect,
b) Parallel I/O,
c) Fault detection (soft-error), and
d) Higher accuracy computing.

We are going to complete this project through close collaboration among computational science (simulation), computer science (hardware and software) and numerical mathematics. Our final goal is to establish a fundamental technique to develop numerical software libraries for next generation supercomputer systems based on strong cooperation within AICS.

## 4.3. Research Results and Achievements

In this report, we mainly focus on three topics, 1) development of KMATHLIB, 2) collaborative work with JAEA, and 3) CREST project. Other activities done in FY2012-2013 can be referred from the future plans and the publication list.

### 4.3.1. Development of KMATHLIB

**Benchmarking OSS (Open Source Software) packages**

We recognize that numerical library is an important tool to support simulation users when they develop their own practical codes on the K computer. Usage of the K computer is spread over a wide range of spectral from hundred to ten-thousand nodes or up to the whole system. Thus, to support such users widely is a very challenging work.

On the FY2012-2013, we researched and examined the availability of OSS (Open Source Software) packages, some of which have been ported to the K computer. To complete this work, we developed a benchmark suite at first. By using this benchmark suite and the standard profiling tool on the K computer, we measured several performance metrics, for example parallel efficiency, parallel speedup, and communication overhead. In order to utilize the computational potential of the K computer, we, especially, would like to know the parallel behavior defined by such as parallel efficiency, load balance and communication overhead. Figure 1 is an example of the benchmark suite. It illustrates the parallel performance of `pdgesv' and `pdhseqr' of ScaLAPACK on the K computer.

The benchmarking on this FY2012-2013 covers ScaLAPACK (pdgesv and pdhseqr), PETSc (GMRES and JD), and FFTW. We already run the benchmark code from 2^4=16 nodes (=128 cores) to 2^15=32768 nodes (=262144 cores). Currently, the benchmark is in the analyzing phase for the obtained results. Detail will be revealed in the FY2013-2014.

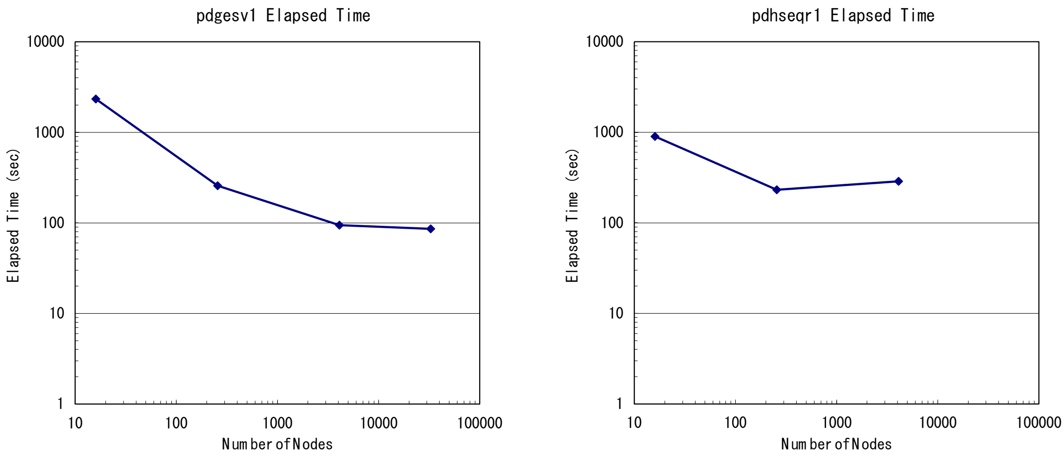


Figure 1: Parallel performance of the ScaLAPACK on the K computer in a strong scaling (Left: pdgesv, Right: pdhseqr)

**Enhancement of the existing OSS**

We are also concerned with the demand of a lot of users. There are hundreds of numerical routines of the K computer, which are not implemented but demanded from users strongly. We designed a layer structure of KMATHLIB as shown in Figure 2.

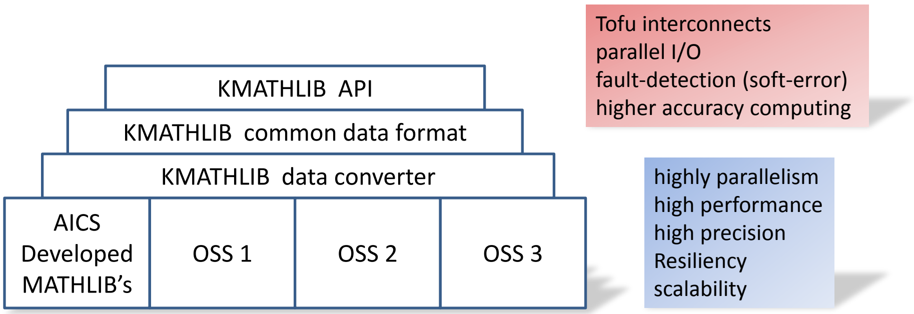


Figure 2: Schematics of the structure of KMATHLIB

We take two ways to develop the KMATHLIB; one is to modify the existing OSS packages, and another is to develop our own solvers to be plugged in the KMATHLIB. On this FY2012-2013, we have modified some of OSS packages and developed following three numerical routines;

- The generalized eigenvalues solver taking advantage of Eigen-K,
- 3D-decomposition FFT routine, and
- Distributed random number generator.

These numerical solvers are quite important tools for scientific simulation codes being studied in AICS.

a) *The generalized eigenvalue solver (KMATH_EIGEN_GEV)* is employed by a lot of simulation code in the material science field. To solve the Schrödinger equation, it yields a nonlinear eigenvalue problem. Then, we modify the non-linear problem to a generalized eigenvalue problem with an iteration scheme, called SCF (Self-Consistent Field calculation). Since, KMATH_EIGEN_GEV is called the innermost loop, thus, it is required to perform very fast. We enhanced our own developed standard eigenvalue solver Eigen-K [1], and implemented it on the K computer. Figure 3 is the parallel performance of the KMATH_EIGEN_GEV routine of the K computer system. It shows the good parallel performance when the number of processes, P (=nodes, here) is composed of large two numbers, for example P is the power of two. The parallel efficiency defined by 'elapsed time using one node' / ('elapsed time using P nodes' * P) achieves approximately 0.236 (=243.3/16.06*64) when the matrix dimension is 10000 and the number of nodes P=64. Compared with the existing solver, pdsygvx in ScaLAPACK, KMATH_EIGEN_GEV outperforms pdsygvx approximately 1.5 times by the elapsed time when the matrix dimension is 10000 and P=64 (KMATH_EIGEN_GEV→16.06[sec], pdsygvx (NB=32) →23.29[sec]). This implies a big advantage of KMATH_EIGEN_GEV.

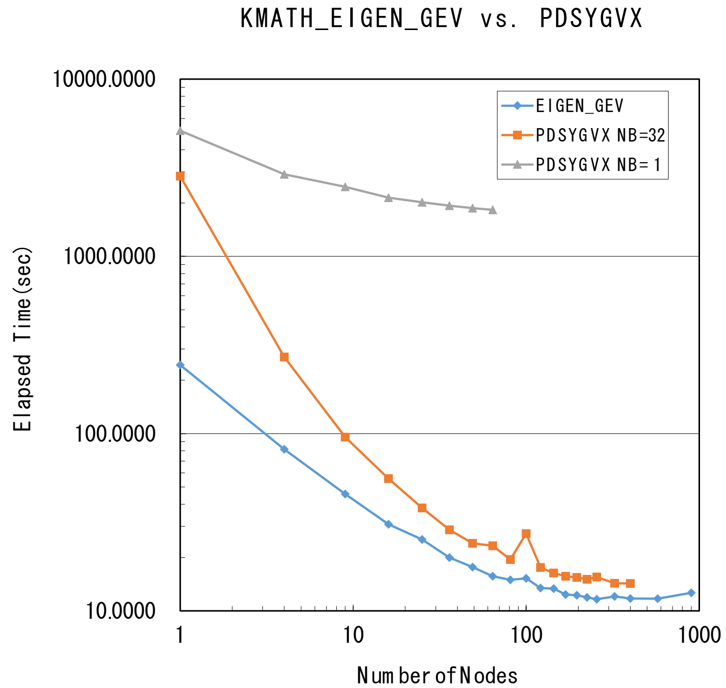


Figure 3: Parallel performance of KMATH_EIGEN_GEV on the K computer in a strong scaling

b) *3D-decomposition FFT routine (KMATH_ FFT3D)* is an innovative enhancement of massively parallel computing in the case that the number of nodes exceeds ten thousand. Current major parallel FFT packages adopt 2D-decomposition data layout, so-called pencil-decomposition. However, it is apparent that the number of parallelism is bounded by the number of grids to be decomposed. When the grid size is assumed to be 100x100x100, 2D-decomposition cannot handle the data on (or map the data into) more than 10000 nodes.

We mainly developed a 3D-decomposition (cubic-decomposition) template program, where user can easily embed and select OSS FFT subroutine such as FFTW [2], FFTE [3], and etc. The call of MPI_Alltoall routines among MPI subcommunications are encapsulated in the implementation.

c) *Random number generator (KMATH_ Random)* is also one of the important numerical tools, as the Monte Carlo simulation plays big role in the scientific simulation codes. Mersenne twister algorithm invented by Prof. Makoto Matsumoto, Hiroshima University, is known as extremely good random number generator, since it yields a very long period and performs faster. We developed user interface KMATH_Random for the distributed random number generator using dSFMT [4] as a random number engine. KMATH_Random maintains a random seed corresponding to the MPI communicator. Therefore, user can easily generate multiple series of random numbers, flexibly. In addition, the random number period is selectable via an environment variable. For example, in case that “export KMATH_RAND_RANGE=1000” is specified, we obtain a very long period of 2^1000.

[1] Eigen-K: http://ccse.jaea.go.jp/ja/download/eigenk.html

[2] FFTW: http://www.fftw.org/

[3] FFTE: http://www.ffte.jp/

[4] dSFMT: http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/index-jp.html

### 4.3.2. High precision mathematical library

We have ported a high precision version of PDSYEVD, namely QPEigenK, with the double-double floating point arithmetic data format (hereafter, DD) on a Fujitsu FX10 system which uses a successor microprocessor installed in the K computer. This study is done via joint work with Dr. Susumu Yamada and Dr. Masahiko Machida, Japan Atomic Energy Agency. Figure 4 shows the largest absolute residual error (defined by max |Ax-wx|) on the DD format are plotted with varying matrix dimension. This suggests that the DD version eigenvalue solver offers finer accuracy (approximately 16 digits accurate) than the original SD (single-double) version. Even though the DD format requires additional floating point number operations rather than the SD format, its high accuracy is promising a qualitative improvement large-scale and complex simulations.

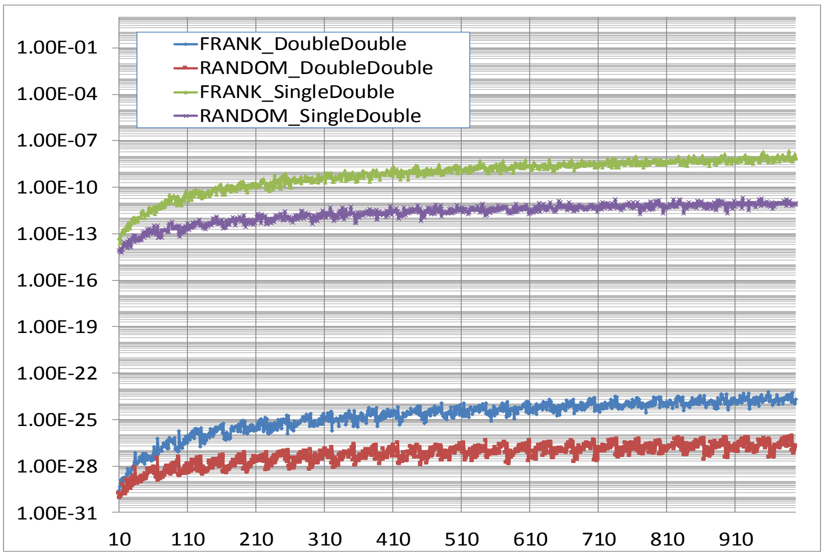


Figure 4: Accuracy test of QPEigenK on a Fujitsu FX10

(Absolute Residual Error max|Ax-wx|_oo)

### 4.3.3. CREST project

We also studied the high performance and scalable eigenvalue solver supported by the national grant, CREST JST [1] (Program code is “Development of System Software Technologies for post-Peta Scale High Performance Computing”). So far, we have developed the Eigen-K, and performance evaluation and analysis of Eigen-K was carried out on the K computer system. Eigen-K shows good performance scalability and it outperforms representative numerical libraries such as ScaLAPACK [2] and ELPA [3]. Figure 5 presents parallel performance in strong scaling. We examined that Eigen-K and ELPA show a similar tendency of parallel performance while the absolute performance of Eigen-K is two to three times higher. Eigen-K is actually adopted in KMATH_EIGEN_GEV, which is already reported. We can say that performance improvement of KMATH_EIGEN_GEV comes from the big potential of Eigen-K. Through the performance analysis, we recognized that Eigen-K has a severe performance bottleneck in the Divide and Conquer routine which is derived from ScaLAPACK. In future work, we should modify it and remove this performance bottleneck.

[1] http://www.postpeta.jst.go.jp/en/

[2] ScaLAPACK homepage, http://www.netlib.org/scalapack

[3] T. Auckenthaler, V. Blum, H.-J. Bungartz, T. Huckle, R. Johanni, L. Kraemer, B. Lang, H. Lederer, P. R. Willems: Parallel solution of partial symmetric eigenvalue problems from electronic structure calculations, Parallel Computing, Vol. 27, Issue 12, p. 783-794 (2011)

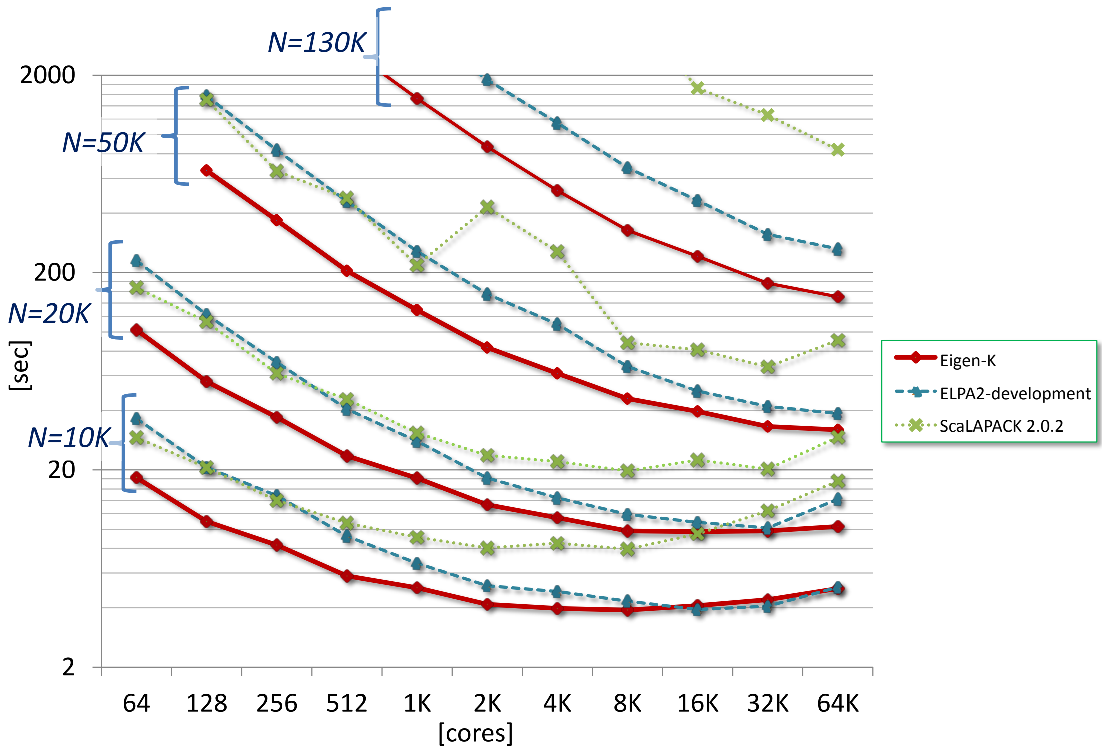


Figure 5: Eigen-K, strong scaling on the K computer

### 4.4. Schedule and Future Plan

We are developing a software package KMATHLIB which integrates several numerical libraries, such as QPEigenK and Eigen-K. In particular, KMATHLIB will be a prominent result of our primary mission. We plan to release the first version of KMATHLIB at the end of FY2013-2014 as Open Source Software. In addition, the remaining benchmark analysis should be done in the FY2013-2014, and the results will be reflected in the second version of KMATHLIB (the FY2014-2015 or later). We would like to accelerate to integrate OSS packages on the K computer and make it a helpful tool on development of scientific simulation codes.

We still have several key topics not mentioned in this report but to be investigated in the K computer. “Fault tolerance” or “resilience” is one of them. It is well known that FT (fault tolerance) is a big issue not only for the K computer but general peta-scale computer systems. From the viewpoint of the numerical library, we would like to establish the algorithmic fault detection mechanism and its framework, then develop a new type of numerical library taking advantage of such a resilience-aware automatic recovery feature.

## 4.5. Publication, Presentation and Deliverables

(1) Journal Papers

Currently, not published

(2) Conference Papers

1. T. Imamura, S. Yamada, and M. Machida, "A High Performance SYMV Kernel on a Fermi-core GPU", High Performance Computing for Computational Science - VECPAR 2012, Lecture Note in Computer Science (LNCS) 7851, pp.59-71, 2013.

(3) Invited Talks

-None

(4) Posters and presentations

1. T. Imamura, S. Yamada, M. Machida, "Preliminary Report for a High Precision Distributed Memory Parallel Eigenvalue Solver", The International Conference for High Performance Computing, Networking, Storage and Analysis (SC12), poster presentation, Salt Lake City, US. Nov. 2012.
2. Y. Idomura, M. Nakata, S.Yamada, T. Imamura, T. Watanabe, M. Machida, M. Nunami, H. Inoue, S. Tsutsumi, I. Miyoshi, N. Shida, "Communication Overlap Techniques for Improved Strong Scaling of Gyrokinetic Eulerian Code Beyond 100k Cores on K-Computer", The International Conference for High Performance Computing, Networking, Storage and Analysis (SC12), poster presentation, Salt Lake City, US. Nov. 2012.
3. T. Imamura, S. Yamada, M. Machida, "Eigen-K: high performance eigenvalue solver for symmetric matrices developed for K computer", 7th International Workshop on. Parallel Matrix Algorithms and Applications (PMAA2012), June 2012, London UK. June 2012.
4. T. Imamura, "Performance Auto-Tuning in Memory-Bound CUDA-BLAS Kernel", 2013 @square HPSC, Conference on Advanced Topics and Auto Tuning in High Performance Scientific Computing, National Taiwan University, Taipei, Taiwan, 27-29 March 2013.
5. T. Imamura, "Beyond Peta-scale Computing from the Viewpoint of Numerical Libraries", The 3rd AICS International Symposium, Computer and Computational Science for Exascale Computing, RIKEN AICS, Kobe, Japan, 28 Feb.-1 March 2013.
6. T. Imamura, T. Utsumi, X. Lin, S. Yamada, and M. Machida, "Performance Tuning for the SYMV kernel on multiple GPU generations, Fermi and Kepler", IPSJ SIG Technical Reports, Vol.2012-HPC-138, No.8, pp.1-7, 2013-02-14. (Japanese).
7. T. Imamura, "KMATHLIB: development of a numerical library on the K computer", The 5-th HPCI Strategic Program Joint research exchange meeting, RIKEN AICS, Kobe, Japan, Jan.

2013.

8. S. Yamada, N. Sasa, T. Imamura, and M. Machida "Quadrature Precision Basic Linear Algebra Subprograms and Its Applications", IPSJ SIG Technical Reports, Vol.2012-HPC-137, No.23, pp.1-6, Dec. 2012. (Japanese)
9. T. Imamura, "Development of a numerical library on the K computer", Workshop on Applied and Computational Mathematics in Industry, Tokyo, Japan, Dec. 2012.
10. T. Imamura, S. Yamada, and M. Machida, "Prospective figure of an Eigensolver for a Dense matrix in an emerging post peta-scale computing Era", JSIAM annual meeting 2012, Wakkanai, Japan, Aug. 2012.
11. S. Yamada, Y. Idomura, T. Imamura, and M. Machida, "Convergence property of Krylov subspace methods for system of linear equations on fusion plasma simulation code GT5D", JSIAM annual meeting 2012, Wakkanai, Japan, Aug. 2012.
12. T. Imamura, T. Yoshida, R. Tamura, H. Kondo, S. Yamada, and M. Machida, "High Performance eigenvalue solver accelerated with an auto-tunner mechanism concering with multicore communication hiding", IPSJ SIG Technical Reports, Vol.2012-HPC-135, No.19, pp.1-8, July 2012. (Japanese)
13. T. Imamura, S. Yamada, and M. Machida, "Eigen-Exa: development of the eigensolver for dense matrices on a post peta-scale computing environment", JSCES annual meeting 2012, Kyoto, Japan, May 2012.

(5) Patents and Deliverables

Currently None