Research Report

KSTS/RR-15/006
April 13, 2015

The flexible incomplete LU preconditioner for large nonsymmetric linear systems

by

Takatoshi Nakamura
Takashi Nodera

Takatoshi Nakamura
School of Fundamental Science and Technology
Keio University

Takashi Nodera
Department of Mathematics
Keio University

Department of Mathematics
Faculty of Science and Technology
Keio University

KSTS/RR-15/006
April 13, 2015

The flexible incomplete LU preconditioner for large nonsymmetric linear systems

Takatoshi NAKAMURA[1] Takashi NODERA[2]

April 13, 2015

Abstract

The ILU factorization is one of the most popular preconditioners for the Krylov subspace method, alongside the GMRES. Properties of the preconditioner derived from the ILU factorization are relayed onto the dropping rules. Recently, Zhang et al. [Numer. Linear. Algebra. Appl., Vol. 19, pp. 555–569, 2011] proposed a Flexible incomplete Cholesky (IC) factorization for symmetric linear systems. This paper is a study of the extension of the IC factorization to the nonsymmetric case. The new algorithm is called the Crout version of the flexible ILU factorization, and attempts to reduce the number of nonzero elements in the preconditioner and computation time during the GMRES iterations. Numerical results show that our approach is effective and useful.

key words. ILU Factorization, preconditioner, GMRES
AMS(MOS) subject classifications. 65F10, 65K10

Contents

1 Introduction

The preconditioned iterative methods for nonsymmetric linear systems [3, 6, 10, 12] are effective procedures for solving large and sparse linear systems of equations:

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

arises from the discretization of elliptic partial differential equations. Two good preconditioners are known, such as the incomplete LU factorization(ILU) [1, 5, 9, 15] and the modified incomplete LU factorization [1], each of which makes use of an approximate factorization of the coefficient matrix into the product of a sparse lower triangular matrix L, and a sparse upper triangular matrix U. It has been observed on an empirical basis that it generates a linear system with eigenvalues that are mostly clustered near 1. The effectiveness of both techniques for nonsymmetric linear systems of equations derived from non-self-adjoint elliptic boundary value problems, has been demonstrated in many numerical experiments [4, 5, 6, 10, 15].

There are now numerous Krylov subspace methods for solving nonsymmetric linear systems of equations, e.g. the GMRES, the Bi-CGSTAB, the QMR and the IDR(s) ([11, 13, 14]). In order to be effective, these methods must be combined with a good preconditioner, and it is generally agreed that the choice of the preconditioner is even more critical than the choice of the Krylov subspace iterative methods. The GMRES [8] is useful for a thorough treatment of preconditioned iterative procedures. The preconditioning of a coefficient matrix is known as one of the methods for improving the convergence of the GMRES. The preconditioner of the ILU factorization applied to the GMRES is popular and is considered to be one of the fundamental preconditioners in the solution of large nonsymmetric linear systems of equations. The search for effective preconditioners is an active research topic in scientific computing. Several potentially successful methods of the ILU factorizations have been recently proposed [15]. The performance of the ILU factorization often is dependent on the dropping method to reduce fill-ins. There are some dropping strategies, for example, the dual dropping strategy which makes it possible to determine the sparsity of incomplete factorization preconditioners by two fill-in control parameters: (i) τ: dropping tolerance and (ii) p: the number of p largest nonzero elements in the magnitude are kept. Recently, Zhang et al. [15] proposed using parameter q to control the number of nonzero elements in preconditioner L, in the IC factorization. Their proposed scheme was called IC factorization with a multi-parameter strategy. The parallel implementation of the ILU factorization is investigated in [6, 7].

In this paper, the general framework of the dropping strategy in an ILU factorization will be proposed. Further to this, a method for overcoming the shortcomings that a calculating norm is needed to use diagonal elements, will be explored. In section 2, the most promising approach for preconditioning will be discussed. In section 3, two flexible ILU factorizations will be proposed and explored. In section 4, the results of extensive numerical experimentations will be tabulated. The conclusion follows.

2 Preconditioning

The preconditioner M can reduce the number of iterations, because the properties of this coefficient matrix can be improved through preconditioning [1]. One possibility is to solve

KSTS/RR-15/006
April 13, 2015

Algorithm 1 Crout Version of the ILU Factorization

1: **for** $k = 1 : n$ **do**
2: Initialize row $\boldsymbol{z}$: $z_{1:k-1} = 0, z_{k,k:n} = a_{k.k:n}$
3: **for** $\{ \, i \mid 1 \le i \le k-1$ and $l_{ki} \neq 0 \, \}$ **do**
4: $z_{k:n} = z_{k:n} - l_{ki} * u_{i,k:n}$
5: **end for**
6: Initialize column $\boldsymbol{w}$: $w_{1:k} = 0, w_{k+1:n} = a_{k+1:n,k}$
7: **for** $\{i \mid 1 \le i \le k-1$ and $u_{ik} \neq 0\}$ **do**
8: $w_{k+1:n} = z_{k+1:n} - u_{ik} * l_{k+1:n,i}$
9: **end for**
10: Apply a dropping rule to row $\boldsymbol{z}$
11: Apply a dropping rule to column $\boldsymbol{w}$
12: $u_{k,:} = \boldsymbol{z}$
13: $l_{:,k} = \boldsymbol{w}/u_{kk}, \; l_{kk} = 1$
14: **end for**

the left preconditioned system of the equation:

$$M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{b}. \tag{2}$$

In general, the preconditioning matrix M is often chosen so that $\mathrm{cond}(M^{-1}A) \ll \mathrm{cond}(A)$, where $\mathrm{cond}(Z)$ is the condition number of matrix Z. A remedy exists when the preconditioner M is available in factored form, e.g., as an incomplete LU factorization $M = LU$, where L is a lower triangular matrix and U is an upper triangular matrix.

2.1 ILU factorization

The ILU factorization is an LU factorization with reducing fill-ins. The ILU factorization factorizes coefficient matrix A as follows:

$$A = LU + R, \tag{3}$$

where L is a lower triangular matrix, U is an upper triangular matrix, and R is an error matrix. The Crout version of the ILU factorization [15] is presented in Algorithm 1. The dual dropping strategy was used in line 10 and 11. For a less complex problem, the effect of the dropping rule is not as important. For large scale problems, however, it is critically important. The number of iterations appears to be sensitive to the dropping tolerance. The basic idea of the dual dropping strategy is constituted by the following two steps:

1. Any elements of L or U whose magnitude is less than tolerance τ is dropped:

   $$|u_{ij}| \le \tau \times \|\boldsymbol{z}\| \;\Rightarrow\; u_{ij} = 0, \quad \text{or} \quad |l_{ij}| \le \tau \times \|\boldsymbol{w}\| \;\Rightarrow\; l_{ij} = 0$$

   where τ is a dropping tolerance.

2. In the k-th column of L, the number of the p largest nonzero elements in the magnitude are kept. Similarly, the number of the p largest nonzero elements in the k-th row of U, which includes the diagonal elements, are kept. This controls the total memory storage that can be used by the preconditioner.

To study parameter p, a new dropping strategy was proposed which changes p by some parameters during the computation of the preconditioner. A dynamically changed parameter q according to the magnitude of elements in the preconditioner L , where q is the number of nonzero elements kept in the corresponding column of L, was introduced for this exercise.

3 Flexible ILU factorization

Zhang [15] proposed a flexible IC factorization which changed parameter p according to the norm of the already computed elements of preconditioner L. This idea was explored to propose a flexible ILU factorization with a new norm, and this will be referred to as the n-flexible ILU. In the n-flexible ILU factorization, q, the number of nonzero elements kept in each column of L and each row of U, is determined as follows:

$$q = \begin{cases} \max\left(p_{\min},\ p + \left[c\log_{10}\dfrac{\|\boldsymbol{l}_j\|}{g_j}\right]\right), & (\|\boldsymbol{l}_j\| < g_j), \\ \min\left(p_{\max},\ p + \left[c\log_{10}\dfrac{\|\boldsymbol{l}_j\|}{g_j}\right]\right), & (\|\boldsymbol{l}_j\| \geq g_j), \end{cases} \tag{4}$$

where parameter p is selected as a basic parameter to control the number of nonzero elements in the preconditioner $p_{\min}$ and $p_{\max}$ that indicate the range of the number of nonzero elements kept in each column of L and row of U. Moreover, parameter c is a proportion value and $g_j = \left(\sum_{k=1}^{j} \|\boldsymbol{l}_k\|\right)/j$.

The nonzero elements of L were compared with the nonzero elements of $\widetilde{L}$, where L was generated by a fixed ILU factorization and $\widetilde{L}$ is generated by n-flexible ILU factorization, respectively:

$$\begin{aligned} \mathrm{nnz}(L) &\approx np, \\ \mathrm{nnz}(\widetilde{L}) &\approx np + \sum_{j=1}^{n}\left[c\log_{10}\frac{\|\boldsymbol{l}_j\|}{g_j}\right]. \end{aligned}$$

This results in the following relation:

$$\sum_{j=1}^{n}\left[c\log_{10}\frac{\|\boldsymbol{l}_j\|}{g_j}\right] < 0 \Rightarrow \mathrm{nnz}(L) > \mathrm{nnz}(\widetilde{L}). \tag{5}$$

The next step was to consider the logarithmic function $f(x) = \log_{10} x$. This function satisfied the following relation: (i) $f'(x)$ is a monotonic decreasing function, and (ii) $f(1) = 0$. From these properties, it was not difficult to prove the following inequality:

$$\log_{10}(1+d) + \log_{10}(1-d) < 0 \quad (0 < d < 1). \tag{6}$$

Assuming that $\|\boldsymbol{l}_j\|/g_j$ is a symmetric distribution to 1, $\sum[c\log_{10}\|\boldsymbol{l}_j\|/g_j] < 0$ and $\mathrm{nnz}(L) > \mathrm{nnz}(\widetilde{L})$, were obtained. The upper matrix U also satisfied the same relation. It was concluded that the n-flexible ILU factorization reduced the nonzero elements of the preconditioner. The n-flexible ILU factorization is characterized by the shortcoming that it needed to calculate the norm during each iteration and as a result, increased the computation time. To overcome this issue, diagonal elements were used instead of $\|\boldsymbol{l}_j\|/g_j$ and a diagonal flexible ILU factorization was proposed called the d-flexible ILU. The d-flexible ILU factorization determined the number of nonzero elements as follows:

$$q = \begin{cases} \max\left(p_{\min},\ p + \left[c \log_{10} \frac{|d_j|}{\widetilde{g}_j}\right]\right), & (|d_j| < \widetilde{g}_j), \\ \min\left(p_{\max},\ p + \left[c \log_{10} \frac{|d_j|}{\widetilde{g}_j}\right]\right), & (|d_j| \geq \widetilde{g}_j), \end{cases} \tag{7}$$

where $\widetilde{g}_j = \left(\sum_{k=1}^{j} |d_k|\right)/j$. In the next section, it will be verified that the d-flexible ILU factorization is suitable for practical use.

4 Numerical experiments

Numerical experiments were implemented, based on ITSOL packages [11]. In this section, the numerical results were used to compare the following methods for solving two examples: the d-flexible ILU, the d-fixed ILU (ILUC) and the n-flexible ILU. All numerical experiments were done on the DELL Precision T1700 with 3.50GHz and a 16GB main memory, using C language. In these experiments, $\boldsymbol{x}_0 = 0$ for an initial approximate solution, and solution $\boldsymbol{x}_i$ is considered to have converged if the norm of the residual, $\|\boldsymbol{r}_i\| = \|\boldsymbol{b} - A\boldsymbol{x}\|$, satisfied the following convergence criterion:

$$\|\boldsymbol{r}_i\|/\|\boldsymbol{r}_0\| < 1.0 \times 10^{-12} \tag{8}$$

where $\boldsymbol{r}_i$ is the residual vector of the i-th iteration. We denoted the computation time of the factorization as CPT, the computation time of GMRES as CGT, the total computing time as the Total, the rational of nonzero elements of L and U to nonzero elements of original coefficient matrix A as $\mathrm{nz}(LU)/\mathrm{nz}(A)$, and the iterations of GMRES as Its. These result were used to illustrate the efficiency of the flexible ILU preconditioning.

4.1 Example 1

The first matrix problem arising from the finite difference discretization of the boundary value problem of the two-dimensional partial differential equation with Dirichlet boundary conditions in [4] is calculated as follows:

$$\begin{aligned} -\Delta u + D\left\{(y - \frac{1}{3})u_x + (x - \frac{1}{3})(x - \frac{2}{3})u_y\right\} - 43\pi^2 u \\ = G(x, y) \text{ on } \Omega = [0, 1]^2, \end{aligned} \tag{9}$$

where $u(x, y) = 1 + xy$ on $\partial\Omega$. The operator was discretized using a five point centered finite difference scheme to discretize on a uniform grid with a mesh spacing $h = 1/128$ in either direction. The parameters were set as follows: $p = 15$, $p_{\min} = p - 0.2p$, $p_{\max} = p + 0.2p$ and $c = 8$. Table 1 shows that in each example, the d-flexible ILU factorization reduces the nonzero elements of the preconditioning matrix without increasing total computation time. The results show that the d-flexible ILU factorization reduces memory usage efficiency.

Table 1: Example 1 - Numerical results of the boundary value problem

Preconditioner	Dh	CPT (sec)	CIT (sec)	Total (sec)	Its	nz(LU)/nz(A)
ILUC	2^4	0.760	1.990	2.750	35	7.549
n-Flexible		0.970	1.700	2.670	25	8.690
d-Flexible		0.640	2.020	2.660	38	6.624
ILUC	2^3	0.720	2.740	3.460	45	7.883
n-Flexible		0.960	2.240	3.200	34	9.266
d-Flexible		0.620	2.680	3.300	47	6.988
ILUC	2^2	0.710	3.910	4.620	60	8.034
n-Flexible		0.970	3.100	4.070	44	9.707
d-Flexible		0.620	3.980	4.600	66	7.399
ILUC	2^1	0.660	10.610	11.270	172	7.776
n-Flexible		0.980	4.310	5.290	58	9.826
d-Flexible		0.620	10.530	11.150	175	7.441
ILUC	2^0	0.590	27.750	28.340	473	7.282
n-Flexible		0.980	7.780	8.760	107	9.865
d-Flexible		0.590	27.730	28.320	473	7.261
ILUC	2^{-1}	0.560	–	–	–	7.009
n-Flexible		0.970	30.280	31.250	413	9.860
d-Flexible		0.560	–	–	–	7.009

4.2 Example 2

The next test problem studied was the Poisson3Db, which is a computational fluid dynamics problem from the University of Florida Matrix Collection [3]. This problem had a 85623×85623 real nonsymmetric matrix and the nonzero elements of this coefficient matrix were 2374949. The nonzero pattern of this matrix is shown in Figure 1. The parameters were set as follows: $p = 70$, $p_{\min} = p - 0.2p$, $p_{\max} = p + 0.2p$ and $c = 8$. Table 2 shows that the d-flexible ILU factorization was faster than other preconditioned methods and its preconditioner had the least nonzero elements. Judging from the computation time vs behavior of residual norm in Figure 2, these were similar to other methods. Figure 3 and Figure 4 show the distribution of $\|\boldsymbol{l}_j\|/g_j$ for the n-Flexible ILU factorization and the distribution of $|d_j|/\widetilde{g}_j$ for the d-Flexible ILU factorization, respectively. These figures suggest that the d-Flexible ILU factorization is the best method for reducing the number of nonzero elements of the fill-in.

In summary, based on the data in Table 1 and 2, it can be concluded that for these experiments, the proposed scheme of the d-flexible ILU preconditioner appears to be superior to other schemes in memory requirements especially in terms of the number of nonzero elements of the preconditioner. Furthermore, for many of these experiments, the GMRES with a d-flexible ILU preconditioner needed less total computation time compared to the ILUC and the n-flexible preconditioner with some exceptions. In Table 2, the results of the experiments are tabulated showing that the proposed scheme executes better or analogous in total time to the solution.

Table 2: Example 2 - Numerical results of the Poisson3Db problem

Preconditioner	p	CPT (sec)	CIT (sec)	Total (sec)	Its	nz(LU)/nz(A)
ILUC	70	3.470	2.500	5.970	38	4.449
n-Flexible		4.800	2.620	7.420	37	5.031
d-Flexible		2.950	2.870	5.820	46	3.965
ILUC	75	3.660	2.580	6.240	38	4.639
n-Flexible		5.260	2.670	7.930	36	5.385
d-Flexible		3.190	2.530	5.720	40	4.161
ILUC	80	3.870	2.660	6.530	38	4.815
n-Flexible		5.710	2.870	8.580	37	5.739
d-Flexible		3.390	2.480	5.870	38	4.346

5 Conclusion

The dropping strategy is integral for the ILU factorization to generate an efficient and reliable preconditioned matrix. The numerical experiments show that the proposed d-flexible ILU factorization, is able to reduce certain nonzero elements of the preconditioner as well as the total computation time. The results also suggest that the GMRES with the d-flexible ILU factorization converges faster than a GMRES with the other classical ILU factorization.

It can be concluded that d-flexible ILU factorization is a practical and effective method for solving large sparse sets of nonsymmetric linear systems of equations.

Future studies are necessary for investigating and determining specific parameters, and finding matrices which can optimize the use of the d-flexible ILU factorization.

References

[1] M. Benzi, "*Preconditioning techniques for large linear systems*," J. of Comp. Physics, Vol. 182, pp. 418–477, 2002. http://dx.doi.org/10.1006/jcph.2002.7176

[2] Chih-Jen, L. and More, J. J., "*Incomplete Cholesky factorization with limited memory*," SIAM,J. on Sci. Comp., Vol. 1, pp. 24–45, 1999. http://dx.doi.org/10.1137/S1064827597327334

[3] Davis, T., University of Florida Sparse Matrix Collection. http://www.cise.ufl.edu/research/sparse/matrices/, 2005.

[4] Joubert, W., "*Lanczos methods for the solution of nonsymmetric systems of linear equations*," SIAM, J. on Matrix Anal. Appl., Vol. 13, pp. 926–943, 1992. http://dx.doi.org/10.1137/0613056

[5] Mayer, J., "*Alternating weighted dropping strategies for ILUTP*," SIAM, J. on Sci. Comp., Vol. 4, pp. 1424–1437, 2006. http://dx.doi.org/10.1137/030602022

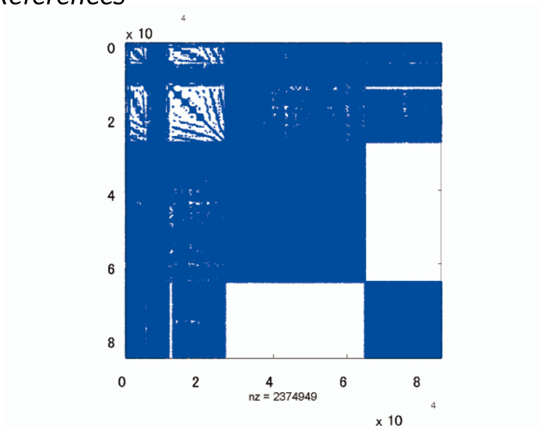

Figure 1: Example 2 - Number of nonzero elements of Poisson3Db's matrix

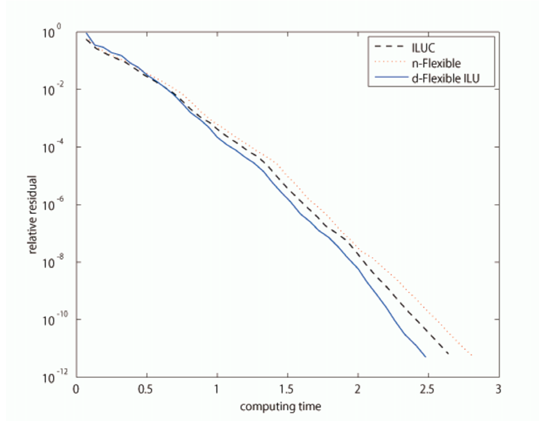

Figure 2: Example 2 - Convergence behavior of residual norm vs computation time, $p = 80$

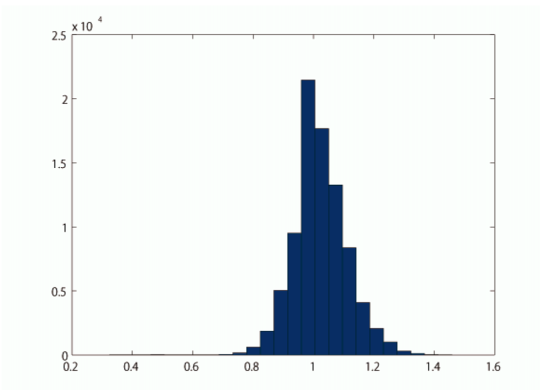

Figure 3: Example 2 - Distribution of $\|\boldsymbol{l}_j\|/g_j$ for n-Flexible ILU

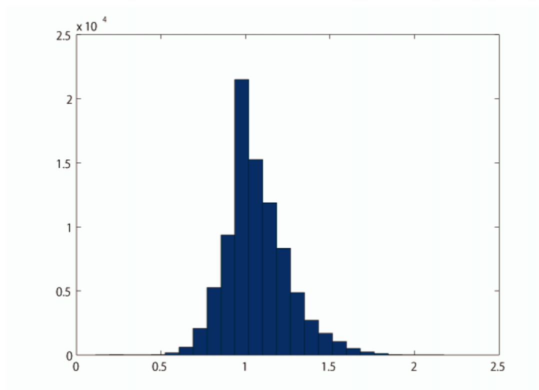

Figure 4: Example 2 - Distribution of $|d_j|/\widetilde{g}_j$ for d-Flexible ILU

[6] Moriya, K. and Nodera, T., "*Parallelization of IUL decomposition for elliptic boundary value problem of PDE on AP3000*," Springer, LNCS Vol. 1615, pp. 344–353, 1999. http://link.springer.com/book/10.1007/BFb0094901

[7] Nodera, T. and Tsuno, N., "*The parallelization of incomplete LU factorization on AP1000*," Springer, LNCS, Vol. 1470, pp. 788–792, 1998. http://link.springer.com/book/10.1007/BFb0057834/page/4

[8] Saad, Y. and Schultz, M. H., "*GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*," SIAM J. Sci. Stat. Comput., Vol. 7, pp. 856–869, 1986. `http://dx.doi.org/10.1137/0907058`

[9] Saad, Y., "*ILUT: A dual threshold incomplete LU factorization*," Numer. Linear Algebra Appl., Vol. 4, pp. 387–402, 1994. `http://dx.doi.org/10.1002/nla.1680010405`

[10] N. Li, N., Saad, Y. and Chow, E., "*Crout versions of ILU for general sparse matrices,*" SIAM J. Sci. Stat. Comput., Vol. 25 pp. 716–728, 2003.
http://dx.doi.org/10.1137/1064827502405094

[11] Saad, Y., "Iterative methods for sparse linear systems," 2nd edition, SIAM, 2003.
http://dx.doi.org/10.1137/1.9780898718003

[12] Saad, Y., ITSOL collections. http://www-users.cs.umn.edu/˜saad/software/ITSOL/.

[13] Sonneveld, P. and Van Gijzen, M. B., "*IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations,*" SIAM J. on Sci. Comput., Vol. 31, pp. 1035–1062, 2008.
http://dx.doi.org/:10.1137/070685804

[14] Van Gijzen, M. B. and Sonneveld, P., "*Algorithm 913: An elegant IDR(s) variant that efficiently exploits biorthogonality properties,*" ACM Transactions on Mathematical Software, Vol. 38, pp. 5:1–5:19, 2011.
http://dx.doi.org/10.1145/2049662.2049667

[15] Zhang, Y., Huang, T. Z. , Jing, Y. F. and Li, L., "*Flexible incomplete Cholesky factorization with multi-parameters to control the number of nonzero elements in preconditioners,*" Numer. Linear. Algebra. Appl., Vol. 19, pp. 555–569, 2011.
http://dx.doi.org/10.1002/nla.784

Author addresses

1. **Takatoshi NAKAMURA**, School of Fundamental Science and Technology,Graduate School of Science and Technology, Keio University, JAPAN.
mailto:takatoshi@z8.keio.jp

2. **Takashi NODERA**, Department of Mathematics,Faculty of Science and Technology, Keio University, JAPAN.
mailto:nodera@math.keio.ac.jp

Department of Mathematics
Faculty of Science and Technology
Keio University

Research Report

2014

[14/001] A. Larraín-Hubach, Y. Maeda, S. Rosenberg, F. Torres-Ardila,
Equivariant, strong and leading order characteristic classes associated to fibrations,
KSTS/RR-14/001, January 6, 2014

[14/002] Ryoichi Suzuki,
A Clark-Ocone type formula under change of measure for canonical Lévy processes,
KSTS/RR-14/002, March 12, 2014

2015

[15/001] Shiro Ishikawa,
Linguistic interpretation of quantum mechanics: Quantum Language,
KSTS/RR-15/001, January 22, 2015

[15/002] Takuji Arai, Ryoichi Suzuki,
Local risk-minimization for Lévy markets,
KSTS/RR-15/002, February 2, 2015

[15/003] Kazuma Teramoto, Takashi Nodera,
Lanczos type method for computing PageRank,
KSTS/RR-15/003, March 9, 2015

[15/004] Yoichi Matsuo, Takashi Nodera,
Block symplectic Gram-Schmidt method,
KSTS/RR-15/004, March 9, 2015

[15/005] Yuto Yokota, Takashi Nodera,
The L-BFGS method for nonlinear GMRES acceleration,
KSTS/RR-15/005, March 9, 2015

[15/006] Takatoshi Nakamura, Takashi Nodera,
The flexible incomplete LU preconditioner for large nonsymmetric linear systems,
KSTS/RR-15/006, April 13, 2015