Development of Performance Evaluation System and Knowledge Database on Matrix Computations

- iWAPT2006 -
(12 Sep. 2006)

Shoji ITOH
(University of Tsukuba, JAPAN)

Table of Contents

Introduction

In the field of scientific and technical computation, we need to solve numerically the various equations which describe realistic problems like natural phenomena or engineering problems.

In the end their solutions often be reduced to solving linear equations of large size:

$$Ax = b$$

A : the coefficient matrix of size $n \times n$

$\boldsymbol{x}$: the solution vector of size n

$\boldsymbol{b}$: the right hand side vector of size n

It is important to solve this equation fast and accurately.

Motivation: Which algorithm should we select ?

Who evaluate these algorithms' properties and efficiencies
systematically ?

Direct methods

based on
Gaussian Elimination
LU , Cholesky
others

Iterative methods

Nonstationary iterative Methods
(Krylov Subspace meth.)

Conjugate Gradient
BiCG , CGS
Bi-CGSTAB
BiCGstab(ℓ)
GMRES , GPBiCG
others

Stationary iterative methods

Jacobi
Gauss-Seidel
SOR
others

Furthermore :

Preconditioning !
ILU, Point Jacobi,
Hybrid, I+S, SA-AMG,
others

Current status and problems

If the coefficient matrix A is symmetric positive definite, we may select the CG method or the Cholesky method.

But if the coefficient matrix is nonsymmetric, we never know which algorithm should we select ?

Especially, about Krylov subspace method:

> There is no clear best Krylov subspace method at this time, and there will never be a best overall Krylov subspace method.
>
> Templates for the solution of linear systems: building blocks for iterative methods, R. Barrett, et. al, SIAM, 1994.

Further, we never know the effect of preconditioning till getting these computing results.

Research purposes

Systematic performance evaluation and the characteristic analysis on numerical algorithms

We analyze actual characteristics and so on, by using some data analyzing methods or data evaluation methods, TQM, statistics, data mining or visualization technique.

Here, we pay attention to various data generated by several solution algorithms.

Survey chart on Iter. Num. (Grouping by Solver)

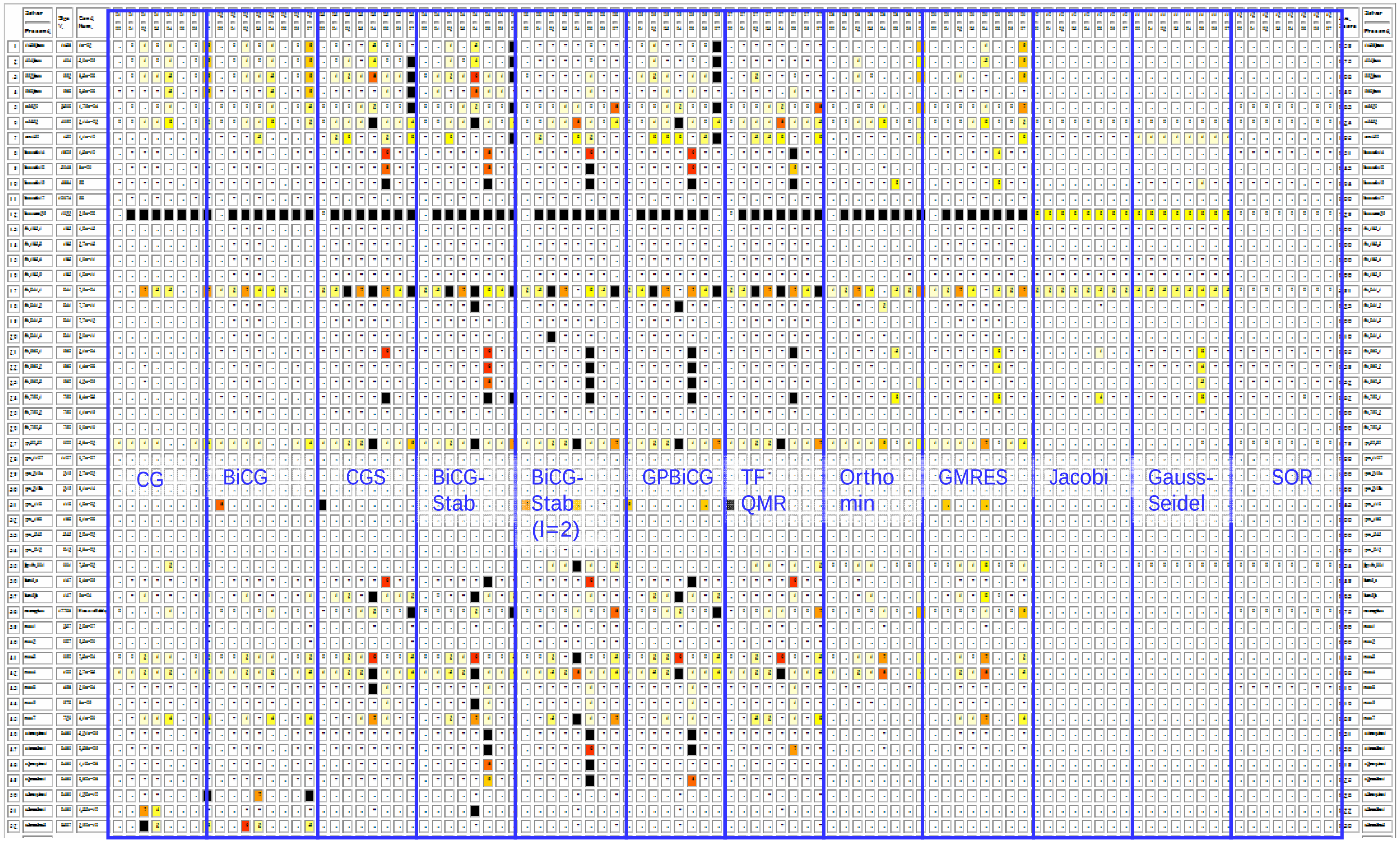
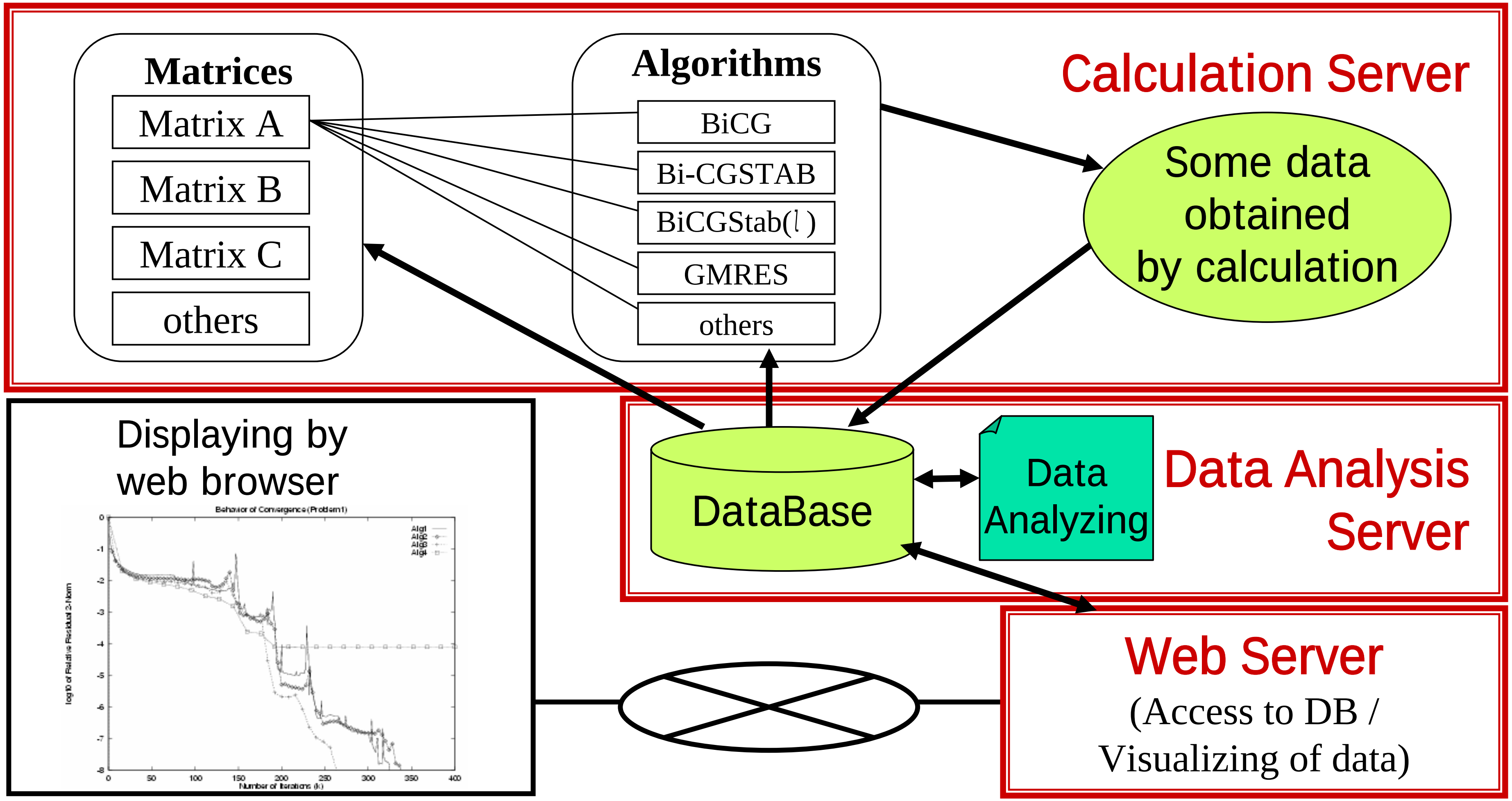


Table of Contents

Computational evaluation system (Conceptual diagram)



System environment (Calculation Server)

- **Matrices (Test problems):**

  52 kinds of matrices for linear equations from Matrix Market

  Each RHS vector is generated by *b*=A*x*, here *x*=(1.0 , … , 1.0)

- **Algorithms (Numerical Calculation Library):**

  Lis [Lis-AMG-1.0.1 seq. ver.] (H.Kotakemori, A.Nishida, H.Hasegawa)

  12 solvers × 8 preconditioning (including "no precond.")

Machine	Sun Fire V880
CPU	UltraSPARC III (900 MHz)
Memory size	16 GB
Operating System	Solaris 9
Compiler	Sun Workshop 6 (cc, f90)

Outline of Lis library

Lis (a Library of Iterative Solvers for linear systems)

- Including sequential version of the algorithms, and two of parallel versions of MPI and OpenMP.

- Supporting 11 types of storage format. (CRS, CCS, MSR, DIA, …)

Component of Lis: [Lis-AMG-1.0.1 modified ver. (almost equiv. to ver.1.0.2)]

(Some algorithms with † mark are one of features of Lis)

Iterative methods	
Non-stationary iterative methods	
	CG, BiCG, CGS, BiCGStab, BiCGStab(l), GPBiCG†, TFQMR, Orthomin(m), GMRES(m)
Stationary iterative methods	
	Jacobi, Gauss-Seidel, SOR

Preconditioning	
For non-stationary iterative methods	
	None, (Point) Jacobi, ILU(*k*), SSOR, Hybrid†, I+S†, SAINV, SAAMG†
For stationary iterative methods	
	None, I+S†

Evaluating conditions

- The maximum of iterative number:

  We set the value of matrix's size.

- Initial vector: $x_0 = \mathbf{0}$

- Converging criterion: $\|r_k\|_2 / \|b\|_2 \leq 1.0 \times 10^{-12}$

- Here, the vector $\boldsymbol{r}$ is the residual vector in the algorithm. For example,

$$r_{k+1} = r_k - \alpha_k A p_k$$

  on the Conjugate Gradient method.

- Evaluation by the true residual vector

$$\hat{r} = b - A\hat{x}, \quad \hat{x}$$: Numerical solution

$$\|\hat{r}\|_2 / \|b\|_2 \leq 1.0 \times 10^{-8}$$

Outline of the iterative method to solve a linear equations and its convergence property

As a representative of the Krylov Subspace method:
The Conjugate Gradient (CG) method

An initial vector : $\boldsymbol{x_0}$

$$p_0 = r_0 = b - Ax_0$$

$$\text{for } k = 0,1,\cdots \text{ until } \|r_k\| \le \varepsilon \|b\|$$

$$\alpha_k = (p_k, r_k)/(p_k, Ap_k)$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k Ap_k$$

← **The residual vector**

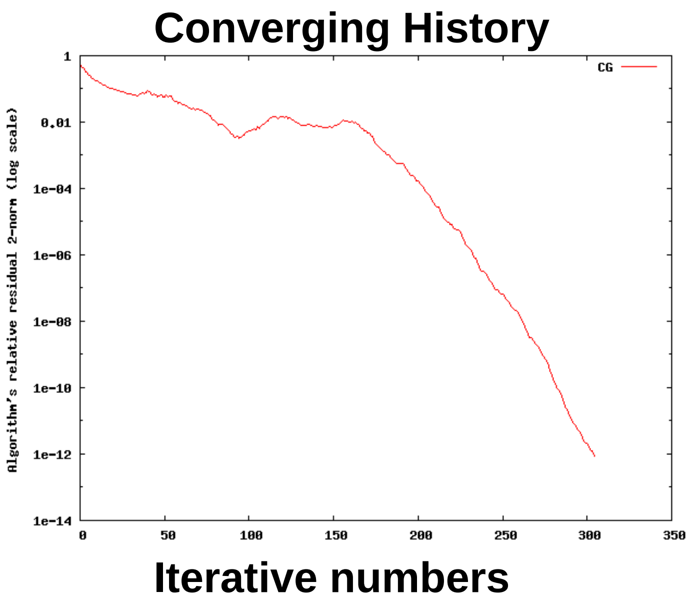
$$\beta_k = -(r_{k+1}, Ap_k)/(p_k, Ap_k)$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

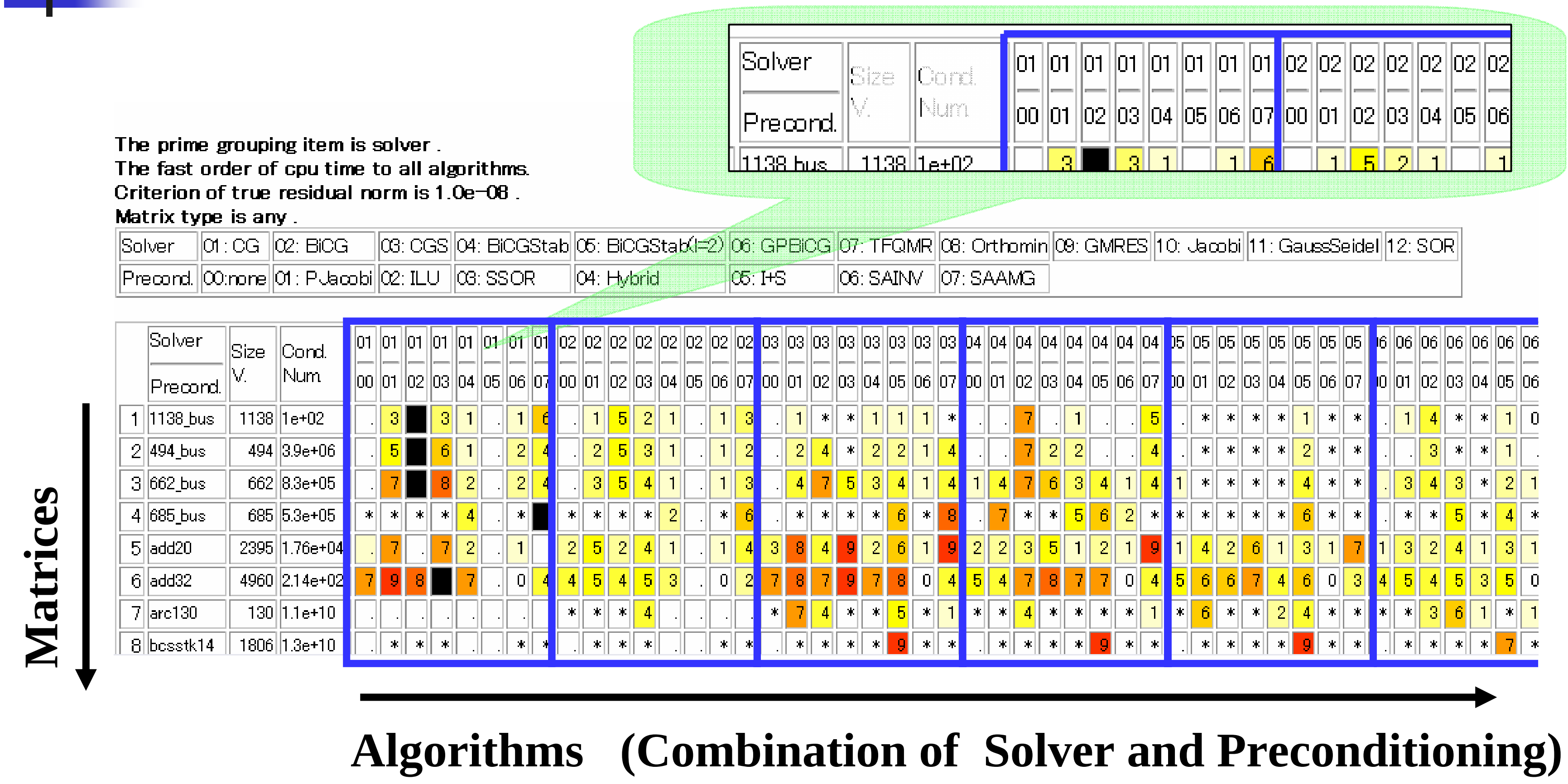
end

These iterative algorithms can converge on the solution with repeating.

$$\log_{10}\left(\frac{\|r_k\|}{\|r_0\|}\right)$$



Survey chart



NOTE: Some blue square frames may be causing a slight difference by the limit of the performance of the drawing software.

Explanatory notes

Score	0	1	2	3	4	5	6	7	8	9	10
Color											

$$\text{Score} = \frac{\text{The fastest algorithm}}{\text{Other algorithm}} * 10$$

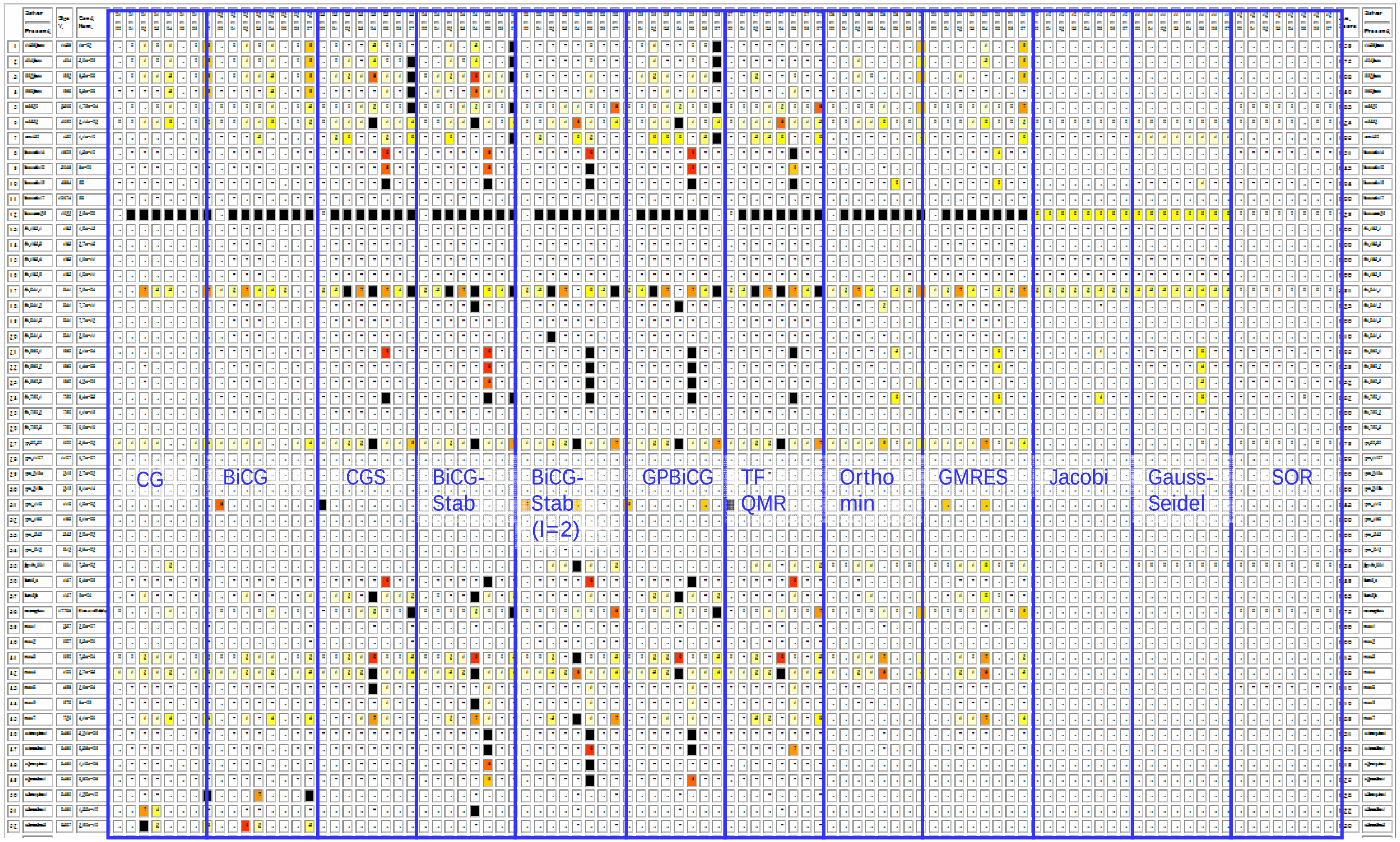
(About CPU time or the iterative number till convergence)

“*” means “Not converged by judging the true residual” (converged by judging the algorithm’s residual)

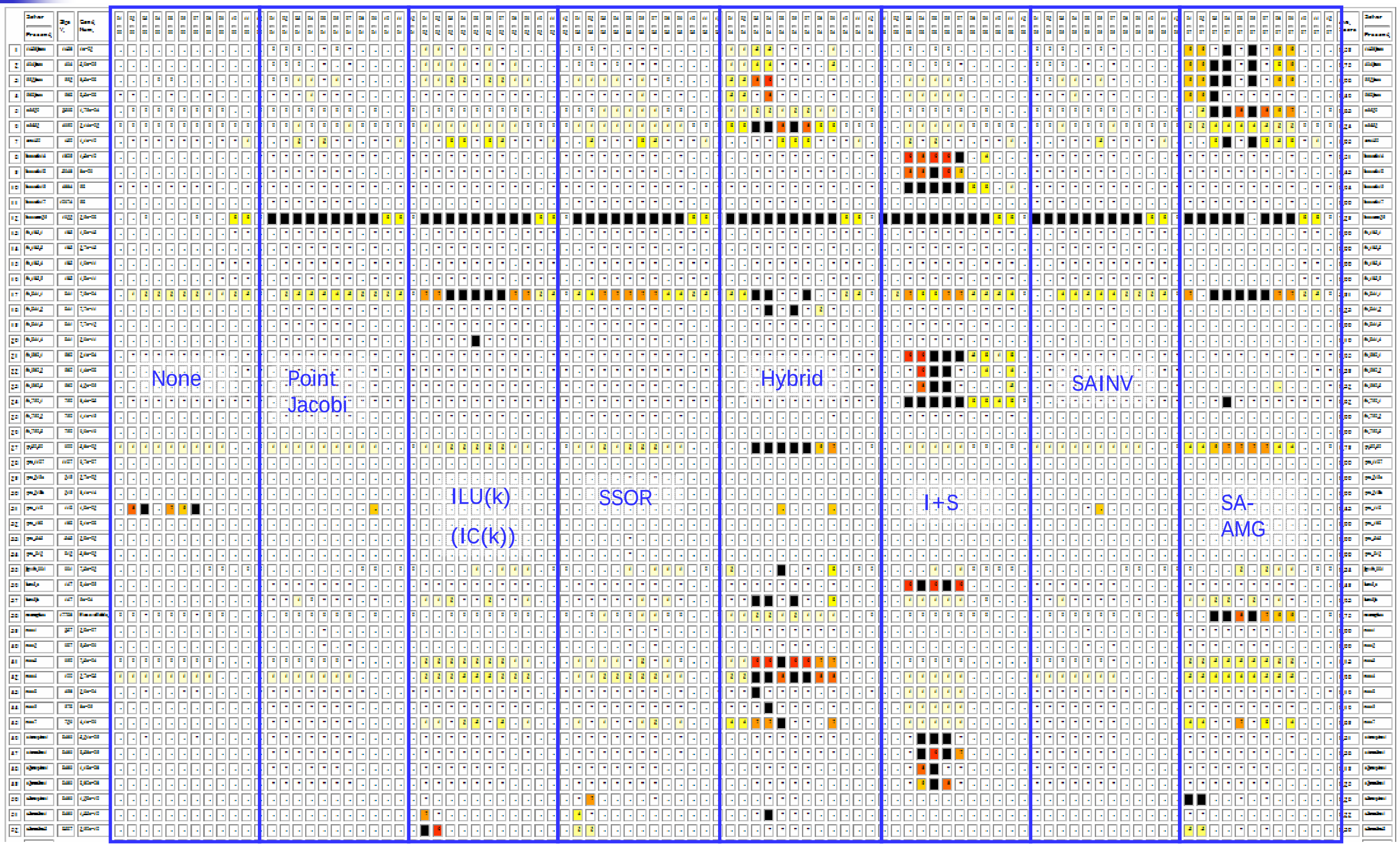
“.” means “Reached at the maximum of the iterative number, or breakdown” (Not converged by judging the algorithm’s residual)

Table of Contents

Survey chart on Iter. Num. (Grouping by Solver)



Survey chart on Iter. Num. (Grouping by Preconditioning)



Observation - 1

From these results, we can know that

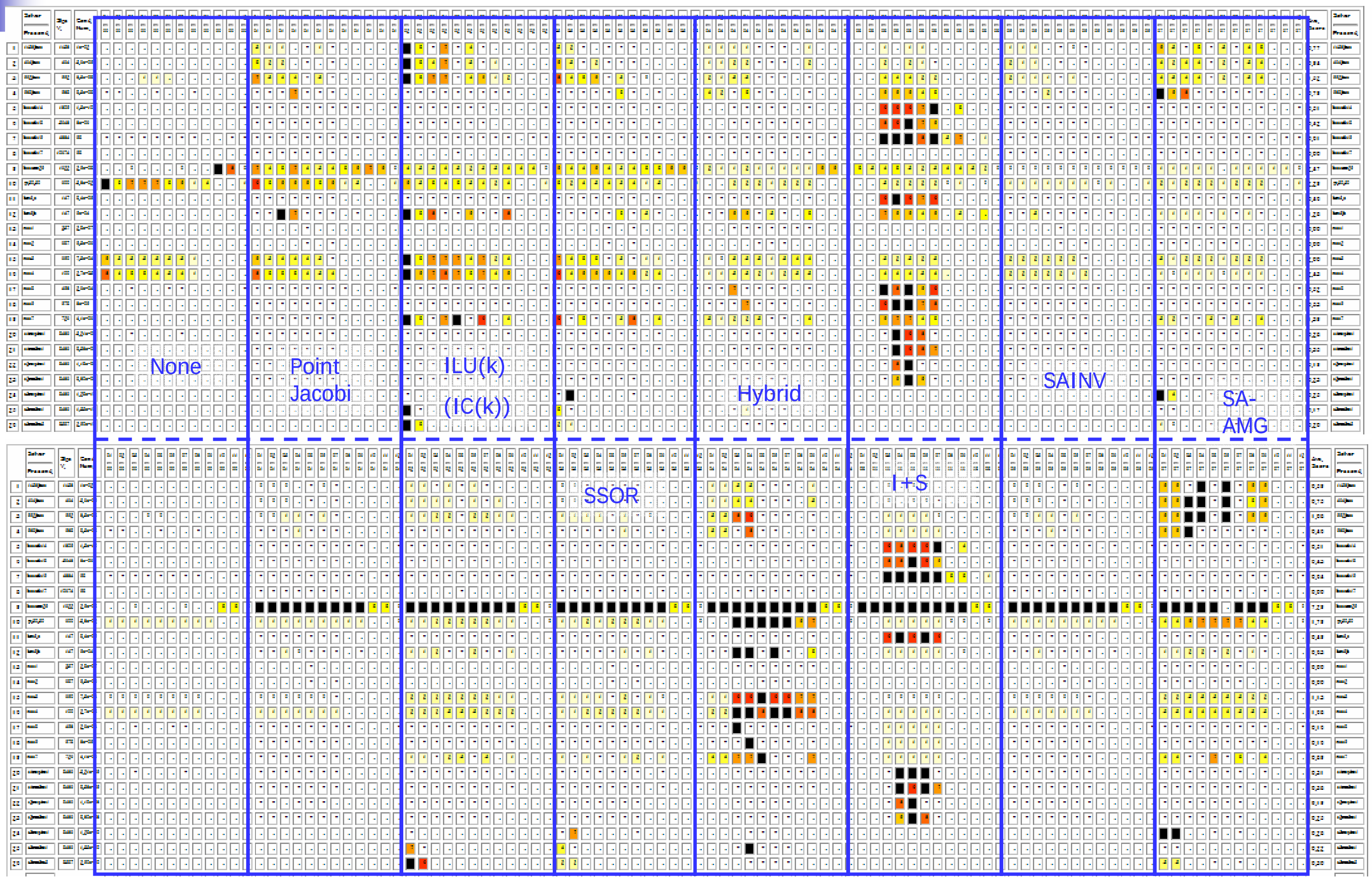
preconditioning
is more effective than
solver

to convergence of iterative solutions.

For Symmetric systems

CPU time

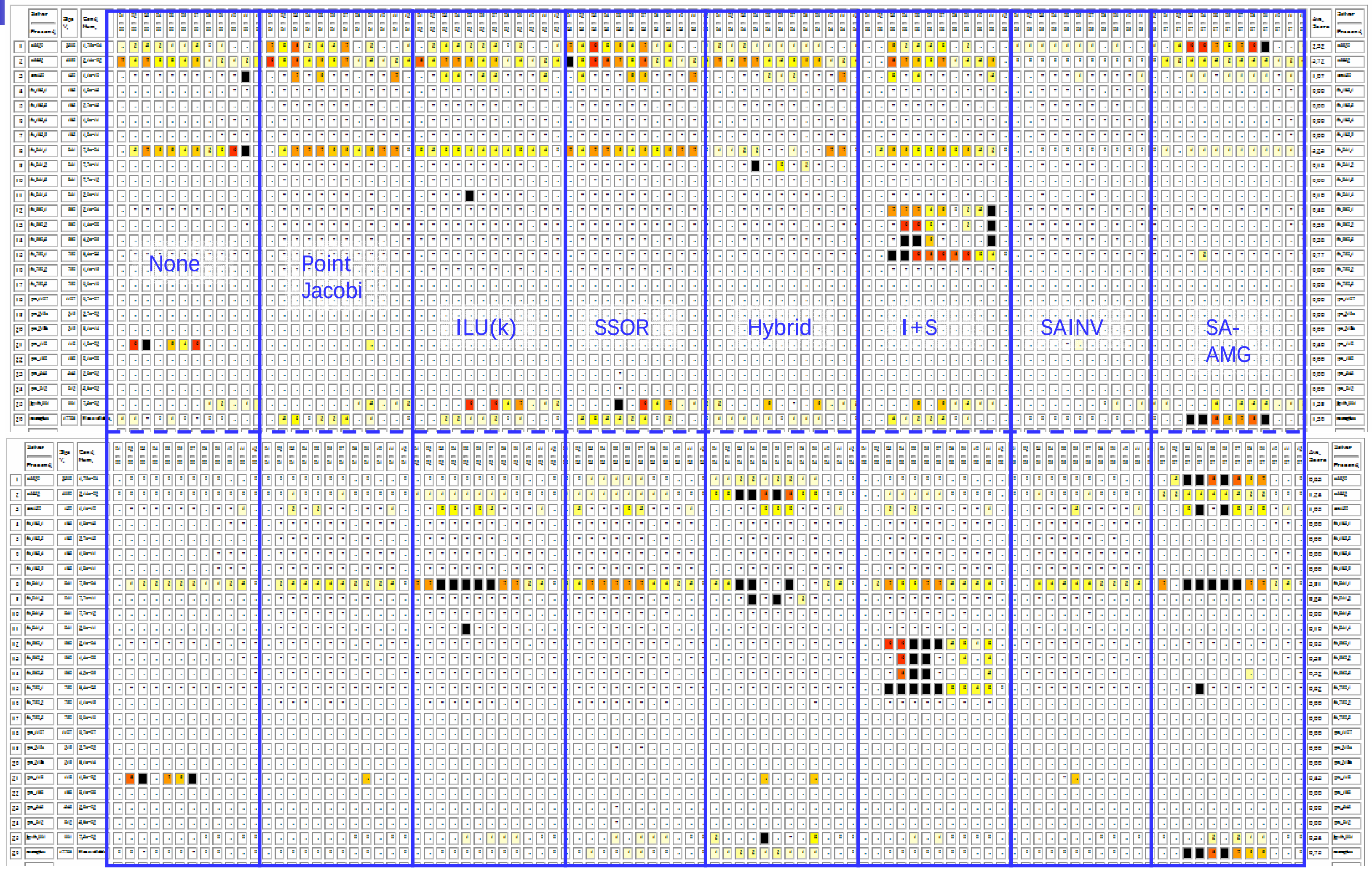
Iter. Num.



For NonSymmetric systems

CPU time

Iter. Num.



Outline of the *I*+*S* preconditioning

Originally the ***I*+*S*** preconditioning was proposed for Jacobi method or Gauss-Seidel method by A.D.Gunawardena, S.K.Jain and L.Snyder (1991)

Coefficient matrix :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & a_{n-1n} \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

Then the matrix ***S*** :

$$S = \begin{bmatrix} 0 & -a_{12} & 0 & \cdots & 0 \\ 0 & 0 & -a_{23} & \cdots & \vdots \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & -a_{n-1n} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$Ax = b$$

$$(I + S^{(m)})Ax = (I + S^{(m)})b$$

I : Unit matrix

m=1**: original,** m=3**: Lis default**

In Japan, several variations of the ***I*+*S*** have been proposed by the group of prof. H.Niki, T.Kohno, H.Kotakemori, et.al.. of Okayama University of Science. And they have studied to apply this preconditioning to Krylov SP solvers.

Observation - 2

Effect of the $I+S$ preconditioning to symmetric system:

In general, even if matrix A is symmetric, matrix $(I+S)A$ is not symmetric.

In usual, if matrix A is symmetric, then the conjugate gradient (CG) method is thought as the best in the Krylov subspace methods.

But these results show that some combinations of the $I+S$ prec. and solver to the symmetric systems solve faster than other algorithms based on the CG method.

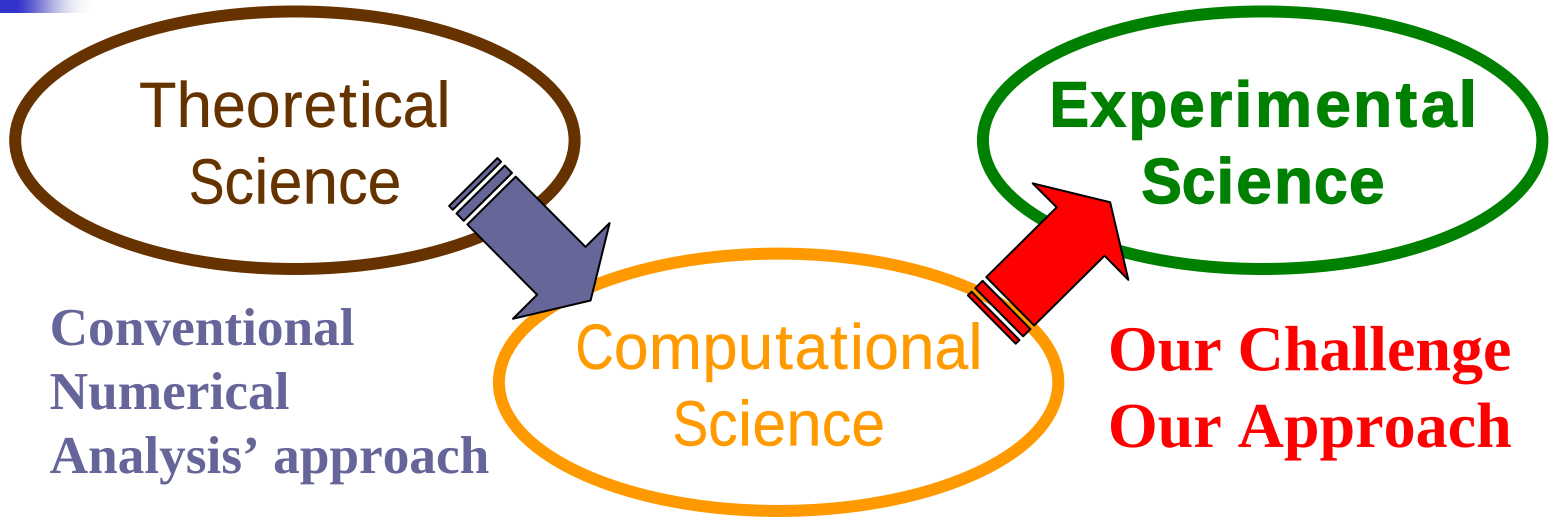
Concluding remarks

1) We have proposed a performance evaluation system of the numerical algorithm to solve linear equations. By using this system, we can analyze the data obtained in solving systematically.

2) Our survey chart shows that preconditioning is more effective in convergence than solver.

3) Especially, the $I+S$ preconditioning is effective not only in nonsymmetric system but in symmetric system, under this computational system environment and these evaluation conditions.

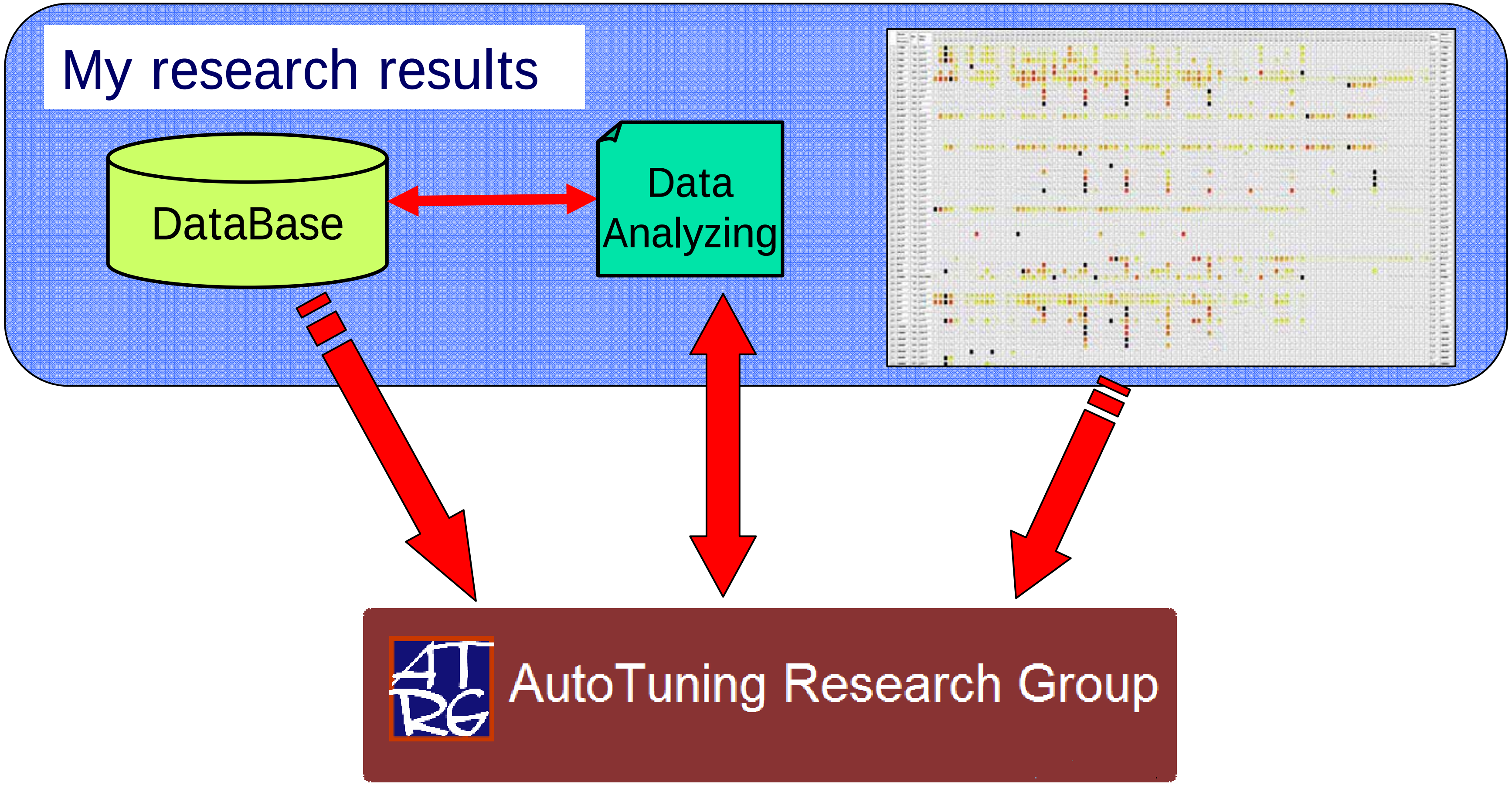
Our research fields



In the past, the solution algorithms which were proposed from the aspect of the theory have been verified by the numerical experiments. These results are merely simulation to the theory.

Our research is one of experimental sciences that analyzes obtained data. It evaluates the value itself of the data obtained by calculation.

The role of my research to ATRG after this



References

About described algorithms and Lis library :

[1] Kuniyoshi Abe, Shao-Liang Zhang, Hidehiko Hasegawa and Ryutaro Himeno, A SOR-base variable preconditioned GCR method, Trans. Japan Soc. Indust. Appl. Math., 11(4), pp. 157-170, 2001 (in Japanese).

[2] Robert Bridson and Wei-Pai Tang, Refining an approximate inverse. J. Comput. Appl. Math., Vol. 123, pp. 293-306, 2000.

[3] Akihiro Fujii, Akira Nishida and Yoshio Oyanagi, A parallel AMG algorithm based on domain decomposition, IPSJ trans. ACS, 44 SIG6(ACS1), pp. 1-8, 2003 (in Japanese).

[4] A.D. Gunawardena, S.K. Jain and L.Snyder, Modified Iterative Methods for Consistent Linear Systems, Linear Algebra Appl., Vol.154-156, pp.123-143, 1991.

[5] Toshiyuki Kohno, Hisashi Kotakemori and Hiroshi Niki, Improving the Modified Gauss-Seidel Method for Z-matrices. Linear Algebra and its Applications, Vol. 267, pp. 113-123, 1997.

[6] Hisashi Kotakemori, Hidehiko Hasegawa, Tamito Kajiyama, Akira Nukada, Reiji Suda, and Akira Nishida, Performance evaluation of parallel sparse matrix-vector products on SGI Altix3700, Proc. of the first international workshop on OpenMP (IWOMP2005), June 2005, to appear.

[7] Hisashi Kotakemori, Hidehiko Hasegawa and Akira Nishida, Performance Evaluation of a Parallel Iterative Method Library using OpenMP, In proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region (HPC Asia 2005), pp.432-436, 2005.

[8] Lis: Iterative method library, http://ssi.is.s.u-tokyo.ac.jp/index_en.html

[9] S.-L. Zhang, GPBi-CG: Generalized Product-type Methods Based on Bi-CG for Solving Nonsymmetric Linear Systems, SIAM J. Sci. Comput., Vol.18 pp.537-551, 1997.