

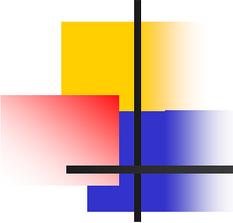
線形方程式求解アルゴリズムに 対する求解性能データに 基づいた体系的な性能比較

－ 行列・固有値問題の解法とその応用 －

第2回研究会
(2006.11.16)

伊藤祥司

(筑波大学 大学院システム情報工学研究科)



発表内容

1. イントロダクション

- モチベーション
- 本研究の目的

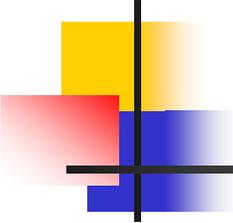
2. 本研究のアプローチ

- 性能評価の情報システム
- システム環境,
今回の評価方法
- Survey chart
(Visual Evaluation)

3. 実験結果, 考察

- Solver ? or
Preconditioning ?
- $I+S$ 前処理の効果
について

4. まとめ



はじめに

自然現象や工学現象の解析では、数値シミュレーションが行われることもしばしばであり、それらを記述する方程式は、多くの場合、大規模な線形(連立1次)方程式

$$Ax = b \quad A : n \times n \text{ 係数行列,} \\ x, b : n \text{ 次ベクトル}$$

を解くことに帰着される。

シミュレーションに要する計算時間の大半が、この計算に費やされるため、速く、正確に解くことが重要である。

〔モチベーション〕どのアルゴリズムを選択すれば良いのだろうか？

アルゴリズムを体系的に
性能比較した情報が、
十分に提供されていない！

直接解法

ガウス消去法に
基づく解法,
LU, Cholesky
others

反復解法

非定常反復法 (Krylov部分空間法)

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

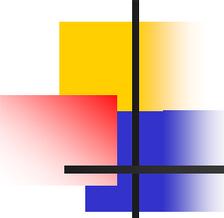
定常反復法

Jacobi
Gauss-Seidel
SOR
others

更には：

解法と併用する前処理も！

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



本研究の目的

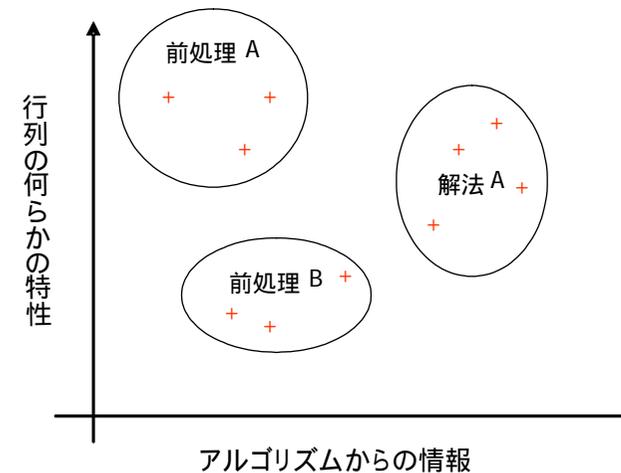
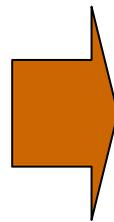
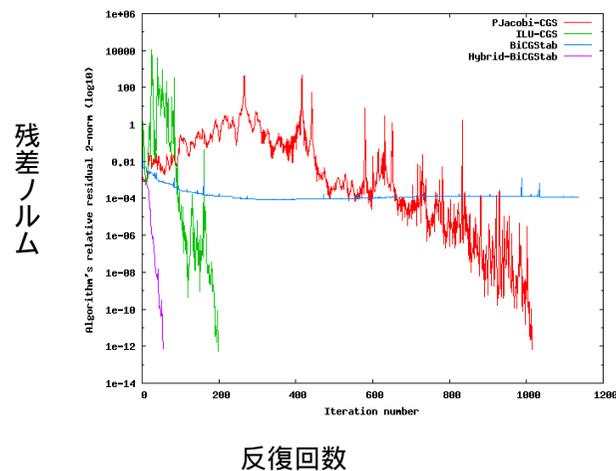
数値計算アルゴリズムの 体系的な性能評価と特性分析

求解アルゴリズムが生成する様々なデータや
解くべき問題の性質を示すデータ

その他の注目すべきファクター等に対して、

品質管理, 統計分析, データマイニングや可視化評価
などを用いて, 実際の特性や傾向などを分析する.

評価方法の一例 (イメージ)

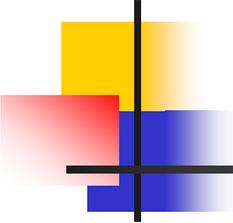


アルゴリズムの個々の性能評価:

1つの問題(係数行列)に対して、複数のアルゴリズムの収束の様子を評価してきた。

アルゴリズムの体系的な性能評価:

様々な問題の、ある特性と、アルゴリズムとの間にはどのような関係が存在するだろうか？



発表内容

1. イントロダクション

- モチベーション
- 本研究の目的

2. 本研究のアプローチ

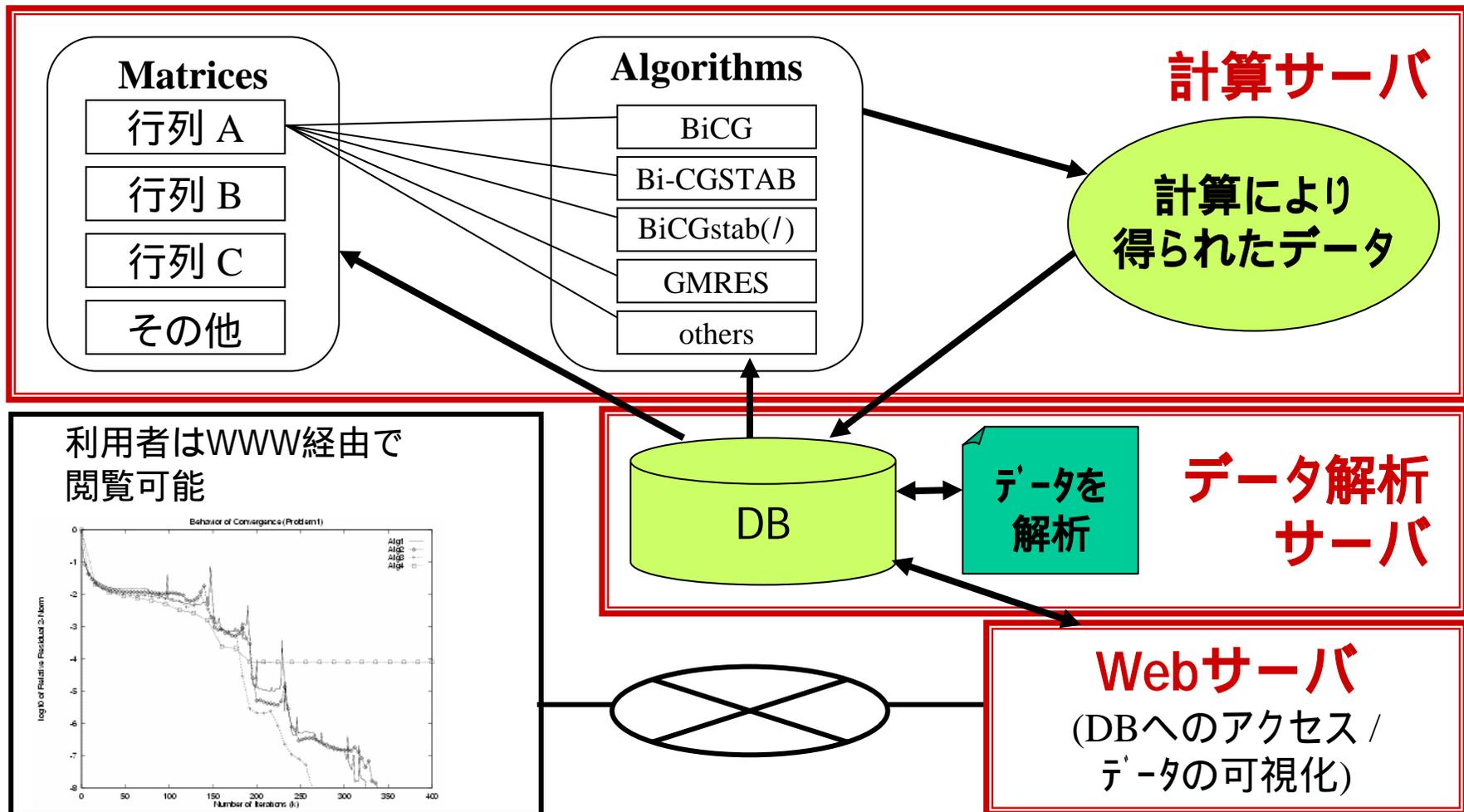
- 性能評価の情報システム
- システム環境,
今回の評価方法
- Survey chart
(Visual Evaluation)

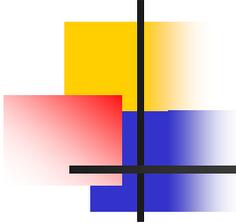
3. 実験結果, 考察

- Solver ? or
Preconditioning ?
- $I+S$ 前処理の効果
について

4. まとめ

情報システム (概念図)





システム環境(計算サーバ)

- Matrices (Test problems):

線形方程式向けの52種類の行列 (Matrix Marketの行列)

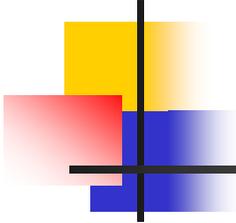
各々のRHSは $b=Ax$, $x=(1.0, \dots, 1.0)$ を用いて生成した

- Algorithms (Numerical Calculation Library):

Lis [Lis-AMG-1.0.1 seq. ver.] (小武守, 西田, 長谷川: CREST SSI prj.)

12 解法 × 8 前処理 (“前処理無し”も含める)

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

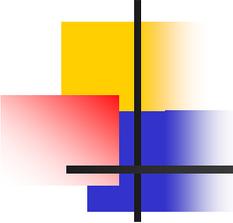
Lis (a Library of Iterative Solvers for linear systems)

- 線形方程式求解アルゴリズムの逐次版, 並列版 (MPI, OpenMP).
- 11種類のデータ形式をサポート. (CRS, CCS, MSR, DIA, ...)

Component of Lis: [Lis-AMG-1.0.1 modified ver. (ver.1.0.2とほぼ同じ)]

(† Lis の特徴: 国内の研究成果)

反復解法		前処理	
非定常反復解法		非定常反復解法向き	
	CG, BiCG, CGS, BiCGStab, BiCGStab(/), GPBiCG[†] , TFQMR, Orthomin(m), GMRES(m)		None, (Point) Jacobi, ILU(<i>k</i>), SSOR, Hybrid[†] , I+S[†] , SAINV, SA-AMG[†]
定常反復解法		定常反復解法向き	
	Jacobi, Gauss-Seidel, SOR		None, I+S[†]



評価条件

- 最大反復回数:

行列のサイズを設定

- 反復法の初期ベクトル:

$$x_0 = \mathbf{0}$$

- 収束判定条件:

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

- ベクトル r : アルゴリズム中の残差ベクトル ,

$$r_{k+1} = r_k - \alpha_k A p_k \quad (\text{CG法の例})$$

- 数値解が有効かどうかの判定 : 真の残差ベクトル

$$\hat{r} = b - A\hat{x}, \quad \hat{x} : \text{数値解}$$

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

Survey chart

(Grouping by solver)

The prime grouping item is solver .
 The fast order of cpu time to all algorithms.
 Criterion of true residual norm is 1.0e-08 .
 Matrix type is any .

Solver	Size V.	Cond Num.	01	01	01	01	01	01	01	01	01	02	02	02	02	02	02	02
Precond.			00	01	02	03	04	05	06	07	00	01	02	03	04	05	06	
1138_bus	1138	1e+02		3		3	1		1	6		1	5	2	1		1	

Solver	01: CG	02: BICG	03: CGS	04: BICGStab	05: BICGStab(=2)	06: GPBICG	07: TFGMR	08: Orthomin	09: GMRES	10: Jacobi	11: GaussSeidel	12: SOR
Precond.	00:none	01: PJacobi	02: ILU	03: SSOR	04: Hybrid	05: I+S	06: SAINV	07: SAAMG				

Matrices

Solver	Size V.	Cond. Num.	01	01	01	01	01	01	01	01	02	02	02	02	02	02	02	03	03	03	03	03	03	03	04	04	04	04	04	04	04	05	05	05	05	05	05	05	06	06	06	06	06	06					
Precond.			00	01	02	03	04	05	06	07	00	01	02	03	04	05	06	07	00	01	02	03	04	05	06	07	00	01	02	03	04	05	06	07	00	01	02	03	04	05	06	07	00	01	02	03	04	05	06
1 1138_bus	1138	1e+02	.	3		3	1	.	1	6	.	1	5	2	1	.	1	3	.	1	*	*	1	1	1	*	.	.	7	.	1	.	5	.	*	*	*	*	1	*	*	.	1	4	*	*	1	0	
2 494_bus	494	3.9e+06	.	5		6	1	.	2	4	.	2	5	3	1	.	1	2	.	2	4	*	2	2	1	4	.	.	7	2	2	.	4	.	*	*	*	*	2	*	*	.	3	*	*	1	.		
3 662_bus	662	8.3e+05	.	7		8	2	.	2	4	.	3	5	4	1	.	1	3	.	4	7	5	3	4	1	4	1	4	7	6	3	4	1	4	1	*	*	*	*	4	*	*	.	3	4	3	*	2	1
4 685_bus	685	5.3e+05	*	*	*	*	4	.	*		*	*	*	*	2	.	*	6	.	*	*	*	*	6	*	8	.	7	*	*	5	6	2	*	*	*	*	*	6	*	*	.	*	*	5	*	4	*	
5 add20	2395	1.76e+04	.	7	.	7	2	.	1		2	5	2	4	1	.	1	4	3	8	4	9	2	6	1	9	2	2	3	5	1	2	1	9	1	4	2	6	1	3	1	7	1	3	2	4	1	3	1
6 add32	4960	2.14e+02	7	9	8		7	.	0	4	4	5	4	5	3	.	0	2	7	8	7	9	7	8	0	4	5	4	7	8	7	7	0	4	5	6	6	7	4	6	0	3	4	5	4	5	3	5	0
7 arc130	130	1.1e+10	*	*	*	4	.	.	.	*	7	4	*	*	5	*	1	*	*	4	*	*	*	*	1	*	6	*	*	2	4	*	*	*	*	3	6	1	*	1
8 bcstkt14	1806	1.3e+10	.	*	*	*	.	.	*	*	.	*	*	*	.	.	*	*	*	*	*	*	9	*	*	.	*	*	*	*	*	9	*	*	.	*	*	*	*	9	*	*	.	*	*	*	*	7	*

Algorithms (Combination of Solver and Preconditioning)

チャート内の配色

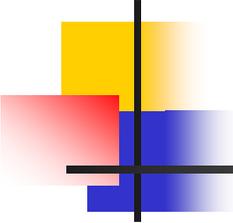
各行列から生成される各々の線形方程式に対して:

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

$$\text{Score} = \frac{\text{最速のアルゴリズム}}{\text{他のアルゴリズム}} * 10$$

(収束までの所要反復回数
やCPU時間について)

- “*” “真の残差の判定では収束していない”
(ただし, アルゴリズム中の残差では収束している)
- “.” “最大反復回数に到達しても収束しない, または,
ブレークダウンで反復が停止した”
(要するに, アルゴリズム中の残差でも収束していない)



発表内容

1. イントロダクション

- モチベーション
- 本研究の目的

2. 本研究のアプローチ

- 性能評価の情報システム
- システム環境,
今回の評価方法
- Survey chart
(Visual Evaluation)

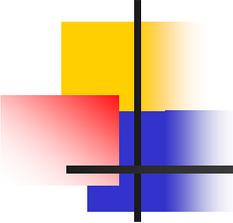
3. 実験結果, 考察

- Solver ? or
Preconditioning ?
- *I+S* 前処理の効果
について

4. まとめ

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

Solver	Test Case	Iter. Num.	Solver	Test Case	Iter. Num.
CG	CGS	...	Ortho min
BICG	GMRES
CGS	Jacobi
BiCG-Stab	Gauss-Seidel
BiCG-Stab(-2)	SOR
GPBICG			
TF			
OMR			

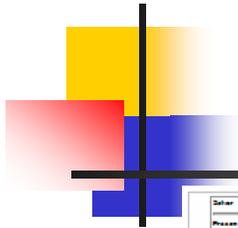


考察(その1)

収束性に対する効果は、

解法の違いよりも
前処理の違いによる
影響の方が大きい

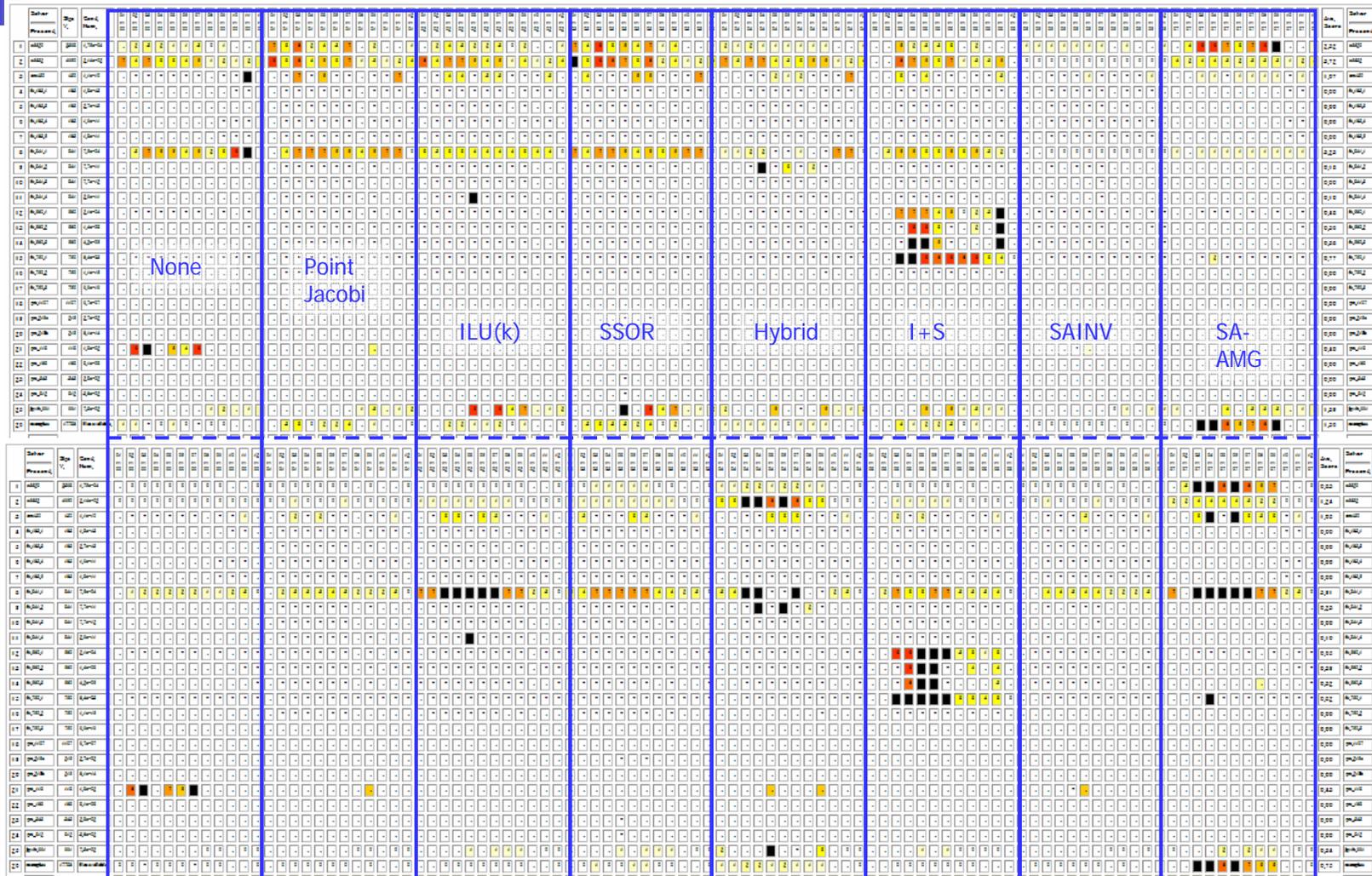
ことが確認できる(本実験の条件下で)。

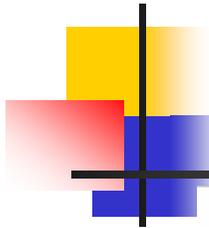


For NonSymmetric systems

CPU time

Iter. Num.

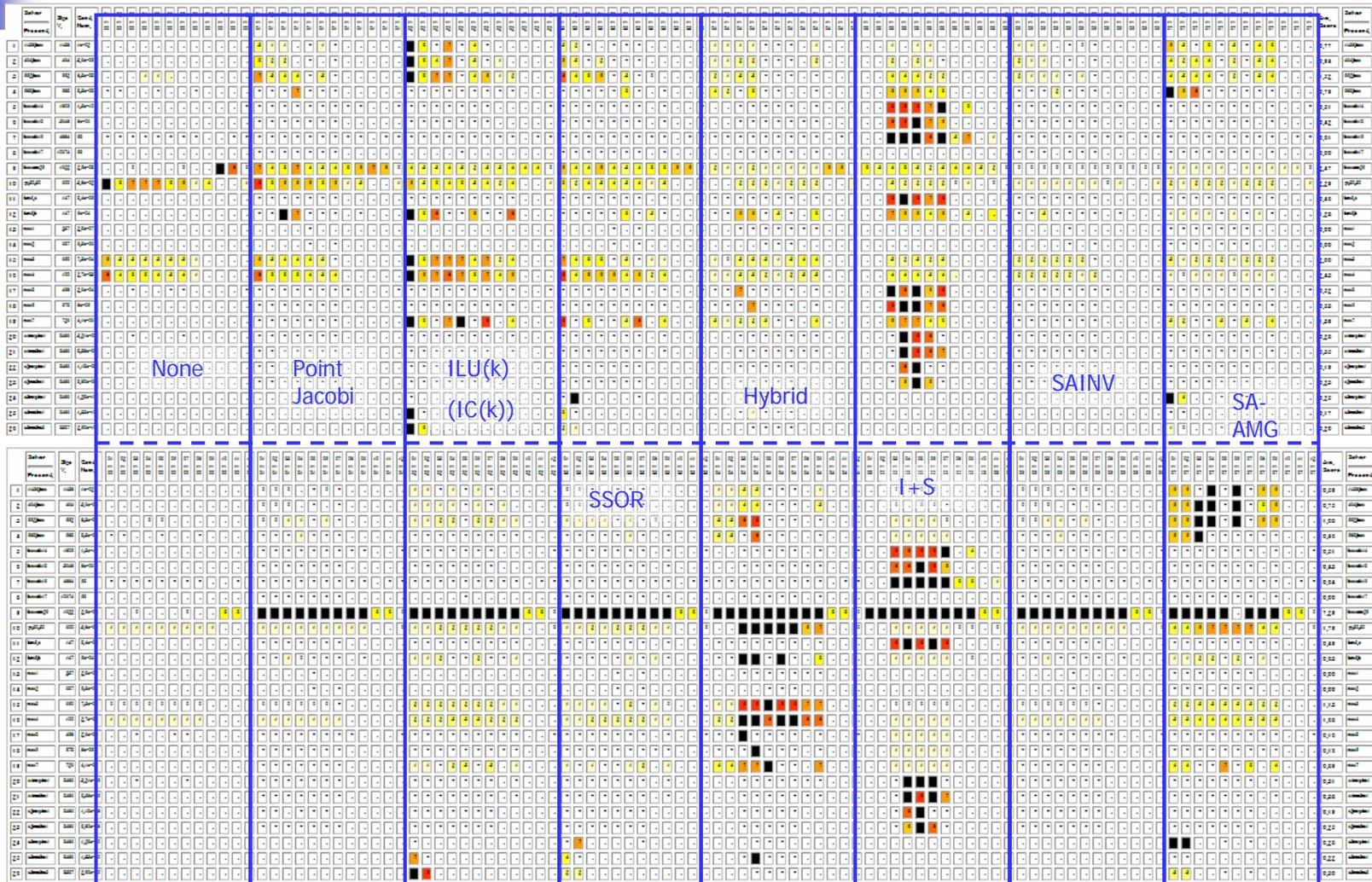




For Symmetric systems

CPU time

Iter. Num.



$I+S$ 前処理

$I+S$ 前処理 : Jacobi法 Gauss-Seidel法向けに開発された
A.D.Gunawardena, S.K.Jain and L.Snyder (1991)

係数行列 :

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

行列 S :

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

m 本の副対角要素

$$Ax = b$$

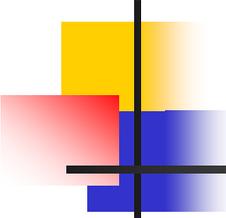
$$(I + S^{(m)})Ax = (I + S^{(m)})b$$

I : 単位行列

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

岡山理科大: 仁木, 河野, 小武守, 他

- ・ $I+S$ 前処理に対する数々の改良
- ・Krylov部分空間法に対する適用



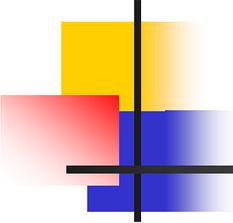
考察(その2)

対称行列に対する $I+S$ 前処理の効果:

一般に, 行列 A が対称であっても,
前処理 $I+S$ を施した系は非対称である.

通常, 係数行列 A が対称であるとき, Krylov部分空間法の中で共役勾配(CG)法がより良い(対称性を保持)と考えられている.

しかし, 本研究の結果からは, 対称系に対しても, CG法に基づくアルゴリズムより, $I+S$ 前処理とCG法以外の解法の組合せの方が速く収束する事例も確認できる.



まとめ

- 1) 線形方程式の求解アルゴリズムに対する体系的な性能評価を行うためのシステムを提案した。
性能情報検索システム (暫定URL: データ, 機能の一部を公開)
<http://mma.cs.tsukuba.ac.jp/~itosho/sesna/>
- 2) Lisを用いた体系的な性能比較の結果から, 反復解法の収束に対する影響は, 解法よりも前処理の方が大きいことが確認された.
- 3) 特に, $I+S$ 前処理は, 非対称系だけでなく対称系に対しても効果があることが確認された.