社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－HPC－107（48）
2006／8／2

線形方程式求解アルゴリズムに対する体系的な性能比較について

伊藤祥司
筑波大学 大学院システム情報工学研究科

線形方程式求解アルゴリズムの体系的な性能比較を行うための，新しいアプローチによる取り組みを紹介する．本研究は，典型的な線形方程式求解アルゴリズムを用いて，数多くのテスト問題を解く際に得られる求解性能を示すデータ自体を評価する．そして，現状の研究手段として，このようなデータを生成するための情報システム，および，そこで得られたデータをビジュアルに比較評価するための Web ベースの性能表示システムについても説明する．

Development of performance evaluation system for numerical algorithms to solve linear equations

Shoji ITOH
Graduate School of Systems and Information Engineering, University of Tsukuba

A new approach to evaluate the performance for numerical algorithms to solve linear equations is proposed. In this research, data itself that shows solution performance obtained when a lot of test problems are solved by using the typical numerical algorithms to solve linear equations is evaluated. Moreover, a web-based informaion retrieval system which displays performance data of algorithms visually is explained.

1．はじめに

自然現象や工学現象の解明では，数値シミュレーションを用いた解析が盛んである．それらのシミュレーションでは，多くの場合，大規模で疎な$n \times n$ の係数行列を持つ線形方程式

$$Ax = b \tag{1}$$

を解くことに帰着される．

ところが，線形方程式の求解アルゴリズムには様々なものが存在し，対象とする問題の性質によっては，その性能が十分に発揮されないような場合や数値解が得られない場合もある[1]．従って，実際のシミュレーションにあたっては，どの求解アルゴリズムを適用したら良いか指針が欲しい．

求解アルゴリズムの性能比較を行うためには，実際のシミュレーションコードでアルゴリズムを入れ替え，計算した上で評価するしかないが，このようなことは数値シミュレーションを行う研究者にとっては，時間のロスであり極力避けたいことである．一方で，求解アルゴリズムを提案する側からは，そのような情報提供が十分になされていない．

本研究ではこのような状況を改善するための新しい試みとして，数値計算全般を対象として，実際の計算結果から得られるデータに注目する．それらの中から，求解アルゴリズムの性能や特性を評価する方法，および，体系的に評価する方式の検討に取り組んでいる[5,6].

現段階では，線形方程式向きの反復解法とそれらと併用する前処理とで構成される求解アルゴリズムに対する性能評価のデータを採取している最中である．その中で得られた諸データからデータベースを構築し，反復解法と前処理の性能比較やグラフ表示を容易にするための検索システムを開発した．本システムを用いると，一般的な数値実験用の問題に対して，線形方程式求解アルゴリズムの性能比較を瞬時に行なうことができる．

本稿の第２節では，本研究を始めるに至った説明としての従来の問題点と現状，および，長期的展望としての本研究の目的について説明する．第３節では，線形方程式求解アルゴリズムに対する性能評価のための情報システムについて説明する．第４節では，求解アルゴリズムの性能を比較評価するためのシステムにつ

いて説明する．

2. 本研究のモチベーション，および，目的

任意の線形方程式(1)に対する求解アルゴリズムの中の分類としては，ガウス消去法に基づく直接解法，SOR法などの定常反復解法，共役勾配(CG)法やBiCG法に代表される非定常反復解法が挙げられる．さらに，これらの求解効率を向上させるために併用する前処理などの技法も多く存在する．

このように，一つの方程式を解くための選択肢が多いにも関わらず，解くべき線形方程式と求解アルゴリズム（本研究では，解法および解法と併用する技法の組合せを指して求解アルゴリズムと呼んでいる）との間の相性などを示す情報が少ない．特に，係数行列が大きいサイズで非対称なときには，アルゴリズム選択には迷うのが現状である．一般的には，直接解法は，小サイズの問題で係数行列が密なときに有効であるとされ，それ以外の問題には反復解法を用いるという一般論もあるが，例外も少なくないため，実用面での適用範囲などに関する体系化された情報は無い．

そして，反復解法の収束性については，求解アルゴリズムに対して理論的な側面で様々に議論されているものもあるが，実際の数値計算を行うと，理論どおりの収束とはならない場合が多い．主な理由として，浮動小数点演算の丸め誤差や，各アルゴリズムの収束性に関する理論面からの弱さなどが考えられる．また，係数行列が正定値対称行列の場合には，行列の固有値分布や条件数に関するCG法の収束性についての議論などは十分になされている[3]が，これらの一方で，非対称行列に対する反復解法の収束性の議論で目立ったものは少ない．さらには，非定常反復解法であるKrylov部分空間法について，

> “There is no clear best Krylov subspace method at this time, and there will never be a best overall Krylov subspace method.”

という報告もある[1].

ここで，「どのアルゴリズムを選択するか」ということは，結局，アルゴリズムの性能をどのように評価するかということにも関係する．アルゴリズムを提案する側の研究者は，数学面での議論に裏付けられた反復解法の収束性や，求解精度の高さに注目する．一方，シミュレーションする側の研究者（アルゴリズムのユーザ）にとっては，自分の問題が解けることと，求解するまでの速さが重要であり，計算精度に関しての許容範囲はシミュレーション対象によっても異なる．したがって，アルゴリズムに対して評価するポイントも違うため，性能評価の手法自体も異なるはずである．

このようなことを念頭において，明確な結論に至らなくとも，せめて，求解したい問題とアルゴリズムとの間の相関などを確認したい．そこで，本研究では，求解すべき問題の特性を表すデータや，実際の計算を行うときに求解アルゴリズムが生成するデータ，計算結果のデータなどに注目する．すなわち，実際のデータに対する比較評価や特性分析などをとおしての，体系的な評価である．

本研究の目的は，数値計算の様々な分野で取り扱う求解すべき問題と，求解によって得られる様々なデータとの定性的・定量的な分析に基づく，数値計算アルゴリズムに対して体系的にサーベイ・評価するシステムを開発することである．ここで，評価システムとは，定性的･定量的なデータ分析による評価方式，および，評価を行うシステム環境としての処理系の双方を指している．現段階では，線形方程式の求解アルゴリズムに対する評価システムから手がけ始めており，求解の際に得られる様々なデータに対する比較評価のシステムを開発している．

線形方程式に関する性能分析に関連する研究については，線形解法などのパラメタを推定する，J. Dongarraらによる**SALSA**プロジェクト[9]や，線形方程式の係数行列を分析して演算量を最小化するオーダリング方法や直接解法を提供する**Grid-TLSE**などがある[4]．国内では，ITBLポータルのTIS[2]が，ユーザがサブミットした線形方程式の問題を実際に解き，性能情報などのデータを提供するサービスを行っている．これらに対し，本研究では，データ分析を用いて線形方程式や数値計算アルゴリズム自体の性能評価を行うことが大きな特徴である．

3. 線形方程式求解アルゴリズム性能評価のための情報システムの開発について

線形方程式を高速かつ高精度で解くために，様々な解法や前処理技法がこれまで提案されてきている．それらを集めたライブラリも多数提供されている中で，本研究では，まず，反復解法から着手することとし，Lis[7] (ver. lis-AMG-1.0.1) の逐次版に基づく修正ライブラリを使用した（修正後の版の性能はver. lis-AMG-1.0.2 に相当する）．Lisには，12種類の反復解法と8種類の前処理（前処理なしも数えて）が用意されており，これら全てを組合せると，単純に数えて96通りのアルゴリズムができる（現版では，一部で排他

的な組合せもある）．

線形方程式求解用アルゴリズムの開発などでは，典型的なテスト問題としてMatrix Market[8]やUF Sparse Matrix Collection[10]などが用いられている．本研究の中ではこれまでに，Matrix Marketの中から選んだテスト行列50種類程度の線形方程式に対してLisを用いて解いた．右辺項ベクトルは，解ベクトルの全要素を1.0として(1)式に代入して生成した．各反復解法に与える初期ベクトルには，零ベクトルを用いた．したがって，１つの係数行列に対して１種類の線形方程式が作られる．収束判定は，アルゴリズム中の残差ベクトルに対する相対残差ノルムが$\|r_k\|_2/\|b\|_2 \leqq 1.0\times10^{-12}$を満たしたときとした（$k$ は反復回数）．最大反復回数として行列のサイズを指定した．

これらを用いてジョブ実行し，データ採取するための計算機環境は表１のとおりである．ここでは，アルゴリズムの基本的な情報を取得するために，ジョブ実行は全て1CPUで行った．

表１　データ採取のための計算機環境

計算機	Sun Fire v880
プロセッサ	UltraSPARC Ⅲ
メモリサイズ	16GB
OS	Solaris 9
コンパイラ	Sun Workshop 6 (cc , f90)

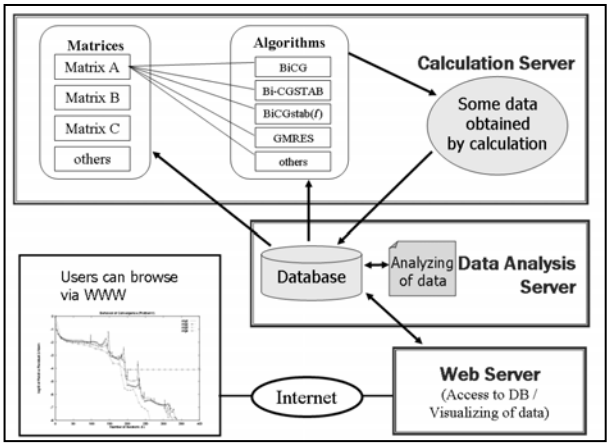

図１　アルゴリズム性能評価のための情報システム（概念図）

図１は，求解アルゴリズムの性能データ採取と，データベース構築，そして，データ検索を行うシステムの概念図を表している．図中の矢印は，ファイルや，分析データの大まかな流れを表している．図１では，各サーバの処理概要を表現しているだけであるが，ここで最も重要なことは，Calculation Serverの計算環境，すなわち，プロセッサ，メモリサイズ，コンパイラなどを常に同じ条件にした上で，ジョブ実行してデータ採取することである．

本節ではCalculation Serverの部分について説明している．図中の“Matrices”でMatrix Marketのテスト行列を用い，“Algorithms”でLisのライブラリに収められている全ての求解アルゴリズムを用いている．

現状では，独立したData Analysis Serverは用意されておらず，Calculation Serverで得られたデータをWeb Serverに転送しているが，今後予定している研究では，係数行列の固有値，行列サイズなどの情報や，求解アルゴリズムが生成する情報など，データベースに蓄積された情報を解析するためのサーバを用意する．

4.　求解性能を比較評価するためのシステムについて

図１のDatabase(DB)に蓄積された様々なデータに対して，統計分析やデータマイニングの手法などを用いて求解性能の傾向を分析することが本研究のメインテーマであるが，現段階ではまず，DB内のデータを表示してアルゴリズムの性能比較を容易にするためのシステムを作成した．

ここで，線形方程式と求解アルゴリズムとの相性を評価するためには，どのアルゴリズムが収束したか，あるいは，高速に解が得られたかということを評価すること以外にも，後述する「見かけ上の収束」に終わってしまうアルゴリズムや，解が得られないアルゴリズムなどに着目することにも意義がある．

4.1 データベースに格納された情報

DBには，反復解法のアルゴリズムの中で用いられている残差ベクトルのノルムの履歴情報と，数値解($\hat{x}$)が得られるまでに要した反復回数，CPU時間および数値解を用いて評価した真の残差

$$r = b - A\hat{x} \tag{2}$$

のノルムの情報が，各行列とアルゴリズムの全ての組み合わせについて格納されている．

4.2 アルゴリズムの求解性能の情報検索システム

これらのデータの中から必要な情報を検索するためのWebサーバサイドのアプリケーションは，PHPとGnuplotを用いて作成した．図２は，検索する際にパラメタ入力する画面イメージである．ここで，検索したい問題の組み合わせをメニューから選択する[6]．本小

節で説明するシステムはWeb上でも公開しており，次のURL（暫定）からアクセスできる．

http://mma.cs.tsukuba.ac.jp/~itosho/sesna

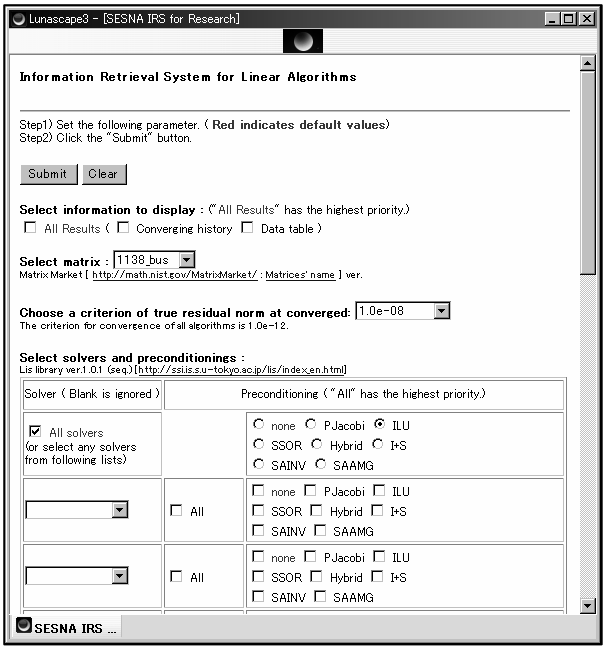

図２ 検索情報の設定画面

図２の例では，線形方程式の係数行列で1138_busという行列を選択し，全ての解法 (All solvers) に対してILU前処理を施したアルゴリズムを選択しているところである．また，数値計算により求解された数値解の精度をあらためて検証するために，(2)式で表される真の残差のノルム値を評価する．図２では，10^{-8}以下の精度を許容範囲としている（「Choose a criterion of true residual norm at converged:」の設定箇所にて，1.0e-08と設定している）．

このように設定して検索実行したときの表示が図３である．図３前半のグラフは，図２で選択されたアルゴリズムにより求解した際に得られた相対残差ノルムの収束の振る舞いを表示しており，グラフの横軸は反復回数，縦軸は相対残差ノルムでありlog10スケール表示である．実際のシステムではカラー表示されており，各アルゴリズムは個々に異なる色で区別されている．

図３後半の「Data Table」の項目は表２のとおりである．

Statusの内容は，真の残差の値から「収束した」(conv.)か，アルゴリズム中の残差では収束しているが，(2)式により真の残差で評価すると収束していない，いわゆる「見かけ上の収束」(no conv.)か，「全く収束していない」(「max. itr.（反復回数の上限に到達）」あるいは「brk. dwn.（ブレークダウン）」) かを表示する．
見かけ上の収束（図３ではNo. 3, 5, 7 のアルゴリズム）

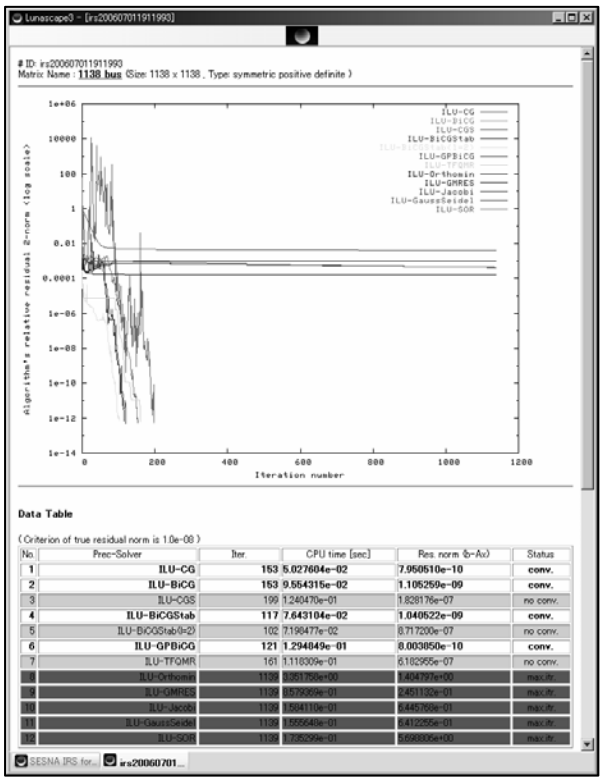

図３ 検索結果の表示画面(収束グラフ)

表２ Data Table中の項目の説明

No.	選択したアルゴリズムに付した番号
Prec-Solver	アルゴリズム名（前処理−解法名）
Iter.	収束までの所要反復回数
CPU time	収束までのCPU時間[sec]
Res. norm	式(2)で表される真の残差のノルム
Status	求解の状況を表示する

に該当するデータのバックグラウンドは黄色で表示しており，全く収束していないもの（図３では，No. 8以後のアルゴリズムが該当）に対しては，該当データのバックグラウンドは赤色で表示している．

また，図３に続く画面では，CPU時間を示す棒グラフも表示されるが，本稿では割愛している．

4.3 アルゴリズム全体に対する求解性能比較システム

ここでは，１種類の線形方程式に対して，どのアルゴリズムが有効であるかを評価する方法について説明する．現状では，最も単純な方法として，Lisに用意された全アルゴリズムに対し，解に収束するまでの所要反復回数，あるいは，CPU時間の大小を比較している．

ただし，本小節で説明するシステムは，本研究向けに用意したものであり，一般には公開していない．

4.1節で説明したデータに対して，係数行列と真の残

差のノルム値を指定する．次に，全アルゴリズムによる計算結果のデータに対して，CPU時間，あるいは，収束までの所要反復回数の昇順にソートする．そして，4.2節の方法と同じく，「収束した」「見かけ上の収束」「全く収束していない」を判定して画面に表示する．

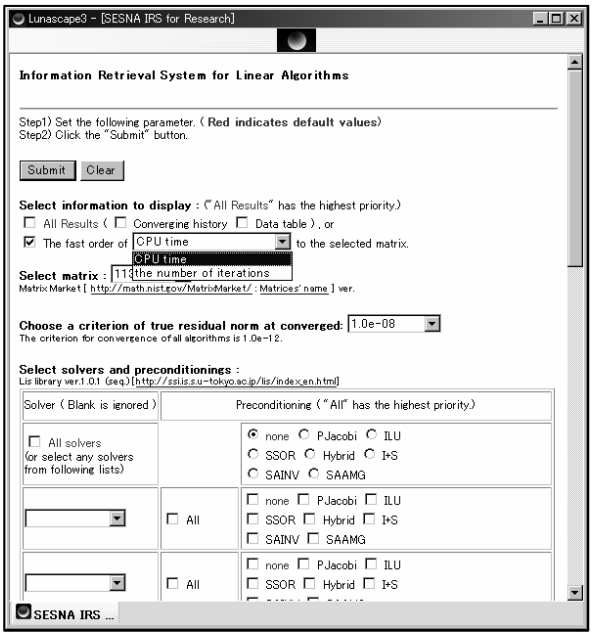

図４ 性能比較システムの設定画面

図４の設定画面では，図２の場合と同様に，1138_bus を選択し，真の残差ノルムの判定基準として10^{-8}を指定し，ソート比較する項目にはCPU時間を設定した．その結果の表示画面が，図５，６である．これら両図は本来連続した画面であるが，説明するのに冗長なレコードは省略している．

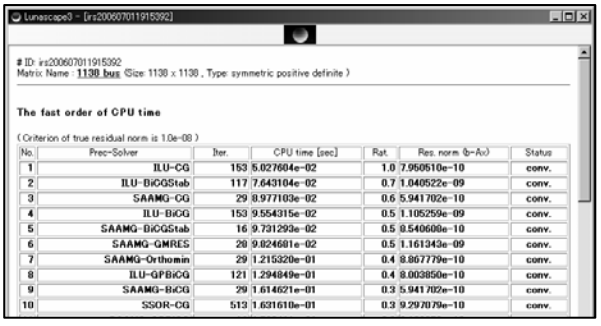

図５ CPU時間でソートした結果の表示（収束した場合）

図５は，「収束した」アルゴリズムをCPU時間の昇順に並べてたところである．図３のData Tableと異なる情報として，「Rat.」という項目を追加している．これは，最速（CPU時間最短，あるいは，収束までの反復回数が最小）のアルゴリズムとそれ以外のアルゴリズムとの性能を相対的に比較するためのファクターである．すなわち，

$$\text{Rat} = \text{最短のCPU時間} \div \text{それ以外のアルゴリズムのCPU時間} \qquad (3)$$

により算出される値を，相対的な性能を示す指標として定義した．

図６中央の薄いグレーのレコードは「見かけ上の収束」のアルゴリズムである．実際の画面では，黄色のバックグランドで表示される．図６下部の濃いグレーはのレコードは「全く収束していない」アルゴリズムの一覧であり，実際の画面では赤色のバックグランドで表示される．

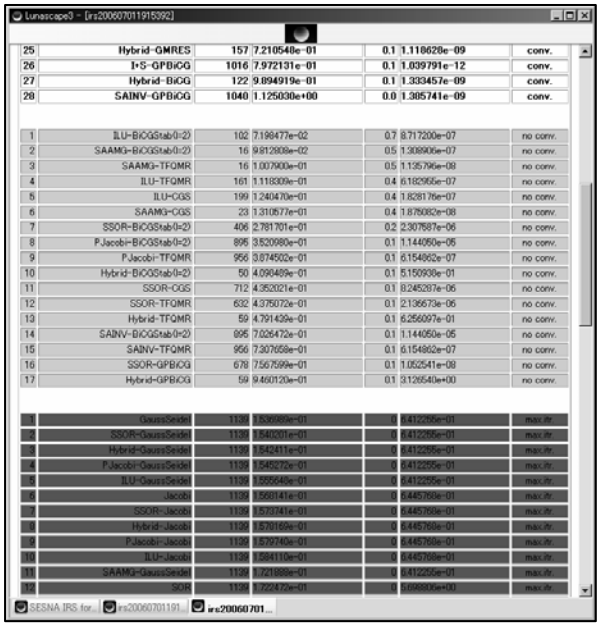

図６ 真の残差ノルムによる判定（収束しなかった場合）

5. まとめと今後の課題

本稿では，線形方程式に対する求解アルゴリズムの例を挙げて，求解アルゴリズムの開発とそれらの利用における現状および問題点について述べた．それら問題点の解決に向けてのアプローチの１つとして，多くのテスト問題を解いて得られるデータに着目し，求解アルゴリズムの性能評価を行う方法について説明した．さらに，その具体的なデータ収集の実現手段として構築した情報システムについて紹介した．

その情報システムのDBに格納された，アルゴリズムの性能を示すデータ，つまり，反復解法の収束性，数値解に対する真の残差ノルム，CPU時間，収束までの

所要反復回数を比較項目として，以下に示す評価方法について説明した．

1) 選択された１種類の線形方程式に対する，任意のアルゴリズムの収束の振る舞いの表示と，真の残差ノルムによる計算精度の判定を行う．
2) 選択された１種類の線形方程式に対する全アルゴリズムの性能一覧を表示し，CPU時間，あるいは，収束までの所要反復回数を昇順にソートし，真の残差ノルムによる計算精度を判定した上で，判定結果のグループごとにリストアップする．

本システムが表示するデータの内，反復回数と収束グラフはアルゴリズムの性能と特性とを示し，CPU時間はLisとしてプログラミングされた求解アルゴリズムの性能も示している．

現在は，用意された全てのテスト問題の線形方程式と全てのアルゴリズムの性能一覧を表示するシステムを作成中であり，そこでの評価方式には(3)式の指標を利用する．

参考文献

[1] Barrett, R., et. al., *Templates for the solution of linear systems: Building Blocks for Iterative Methods*, SIAM, 1994. （邦訳）長谷川里美，長谷川秀彦，藤野清次：反復法 Templates，朝倉書店，1996.

[2] Fukui, Y. and Hasegawa, H., Test of Iterative Solvers on ITBL, In proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region (HPC Asia 2005), pp. 422-425, 2005. TiS project: http://amazon.slis.tsukuba.ac.jp/TiS/

[3] Golub, G. and Van Loan, C. F., *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore and London, 1996.

[4] Grid-TLSE project, http://www.enseeiht.fr/lima/tlse/

[5] Itoh, S. and Hasegawa, H., A Plan to Develop an Evaluating System for Numerical Algorithms, The 2nd International Conference on Scientific Computing and Partial Differential Equations & The First East Asia SIAM Symposium, HKBU, Hong-Kong, Dec., 2005.

[6] 伊藤祥司，線形方程式求解アルゴリズムの性能評価と性能情報データベースの構築，日本計算工学会 第 11 回講演会，大阪，６月，2006.

[7] Kotakemori, H., Hasegawa, H. and Nishida, A., 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. http://ssi.is.s.u-tokyo.ac.jp/lis/

[8] Matrix Market, http://math.nist.gov/MatrixMarket/

[9] SALSA Project, http://icl.cs.utk.edu/salsa/

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