ネットワーク計算用行列工房: Matrix Workshop による性能評価システム

松 原 有 里[†] 中 田 秀 基[††] 関 口 智 嗣[††]
建 部 修 見[††] 長 嶋 雲 兵[†††]

科学技術計算における行列の果たす役割は非常に大きい. また, 行列データは多種多様な性質とサイズを有し, 特に大規模なものについては多様な行列データをユーザが自由に作成するのが困難であり, データコンテンツサービスの対象となるのに適している.

以上の認識から我々は行列を対象としたサービス「ネットワーク計算用行列工房:Matrix Workshop」の開発を行なっている. 本システムでは, サイズに自由度が与えられており, 性質の明らかな行列の提供により信頼性の高い行列演算アルゴリズム, プログラムのテストをサポートすることが可能となった. これにより, 計算機システムの評価を行なうことも可能となっている. 最後に, 本システムを用いた性能評価システムについて Linpack Benchmark を用いた評価をおこなう.

Matrix Workshop: a Matrix Generator on WWW

Yuri Matsubara,[†] Hidemoto Nakada,[††] Satoshi Sekiguchi,[††]
Osamu Tatebe[††] and Umpei Nagashima [†††]

In scientific computing, matrix data are taking important role for representing the physical structure to be understood. Thus, developing efficient algorithms with those matrices is getting more important on high performance computing systems. Difficulty resides in comparing the algorithms with different machines, different matrices and different programs. Regarding performance evaluation in high performance computing, establishing standard data set especially as matrices is strongly desired so that the input data for those algorithms are shared by each other. So, the authors have been developing the "Matrix Workshop" to demonstrate the feasibility of the system which generates matrix data set by giving the matrix structure, dimension of the matrix, and so on. Users accessing the system via WWW or http are able to archive arbitrary size of the matrices with which they understand the behavior of the algorithms developed. In this material, the design overview and preliminary application results are shown.

1. はじめに

科学技術計算においては連立１次方程式の解法として様々なアルゴリズム, プログラムの開発が行なわれている. これらの性能評価やアルゴリズムの正当性を検証することは, 計算結果に信頼性を与えるうえで大変重要となる.

この行列演算アルゴリズムのテストや比較のためには信頼のおける共通のテスト行列が必要となる. 例えば, 連立一次方程式の反復解法や固有値の解法といった行列演算アルゴリズムは, 行列の構造や対称性に依存している. アルゴリズムのテストや比較のためには, その構造や対称性をもつテスト行列が必要となる. また, 条件数や固有値などの特質がわかっている行列であることが望まれる. そのためテスト行列は, 行列成分の定義, 構造, 固有値, 条件数などの行列の性質が明らかであることが重要である.

また, 行列演算アルゴリズムが特定のサイズの行列で正しく機能することは, 可能なあらゆるサイズの行列に対して正しく機能することの保証にはならない. 例えば, 反復法ではアルゴリズムのサイズ依存性があり, あるサイズの時には収束しないということもある得る. よってテスト行列には, サイズ変更などの自由度が必要である.

このように, プログラムの実行性能は与えられたデータにも大きく依存するにも係わらず, 新たに開発されたアルゴリズム, プログラムに対しての評価, 数値実験は十分に行なわれているとはいえない. このことは, アルゴリズム, プログラムの評価を行なう際に用いられる共通の入力データがこれまで存在しなったことからも明らかである. そこで我々は, テスト行列に必要な以上の条件を満たすテスト行列の提供により, 信頼性の高いアルゴリ

† お茶の水女子大学 Ochanomizu University
†† 電子技術総合研究所 Electrotechnical Laboratory
††† 物質工学工業技術研究所 National Institute of Materials and Chemical Research

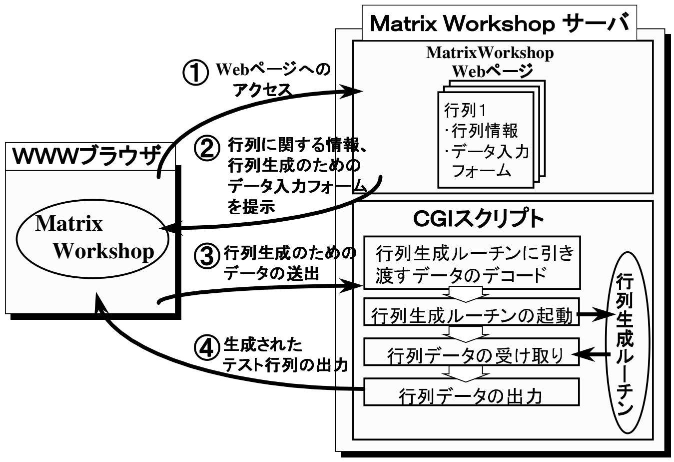

図 1　システム構成

ズム, プログラムのテストをサポートする行列生成システム, Matrix Workshop を開発した．本システムでは,

- 性質の明らかな行列を提供
- 提供する全ての行列は, 任意サイズで生成可能

ということを特徴とする．本稿では本システムの構成, 現状を述べるとともに, その応用としてネットワーク数値情報システム Ninf[3] の枠組を用いた, 行列演算アルゴリズム, プログラムおよび計算機システムの性能評価システムについて提案する．

2. Matrix Workshop

2.1 Matrix Workshop の構成

本システムはクライアント・サーバ型のシステムで, クライアントは Web ブラウザである．行列の生成と入手が, WWW から行なうことが可能なためユーザは, データを容易に入手することが可能である．Matrix Workshop のシステム構成は図 1のようになっている．

Web ページでは, 各行列の性質に関する情報 (行列の構造, 行列成分の定義, 条件数など) や, 行列を生成させるためのデータ (行列サイズや, 行列成分生成のための定数値など) 入力フォームを提供する．

行列の生成は CGI を介して行なわれる．Web 画面に提示されたデータ入力フォームにユーザは行列サイズやいくつかの定数値を入力する．これらは, Matrix Workshop サーバ (図 1 右上) のホストコンピュータ上の CGI スクリプトに渡される．入力データが, CGI スクリプトから行列生成プログラムに引き渡されることで, 行列生成プログラムが起動され, 行列が生成される．生成された行列の出力先は Web 画面かファイル (gzip あり or なし) かをユーザが行列生成の実行を掛ける前に Web 画面に提示されている選択項目により選ぶことが可能である．

行列の出力形式は Matrix Market[1] (同様にテスト行列を提供している, 2.3 で述べる.) で採用されている, Matrix Market Exchange Formats を採用した．

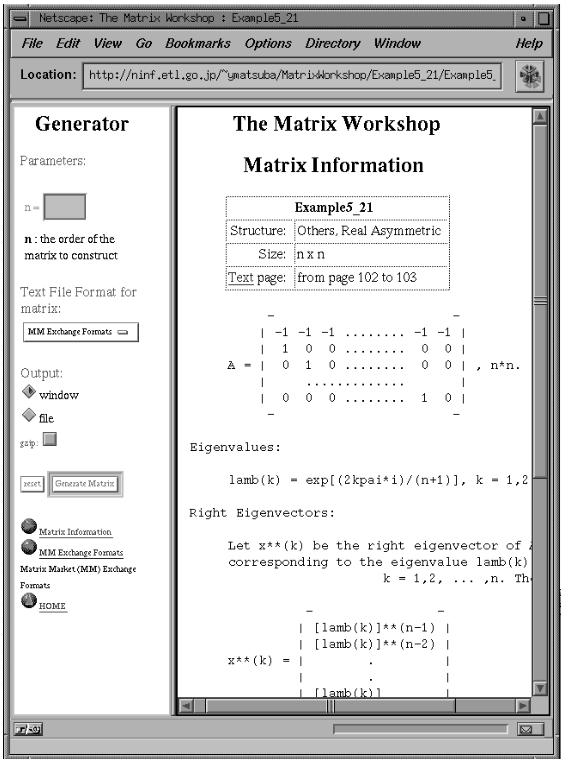

図 2　テスト行列集に対する Web 画面

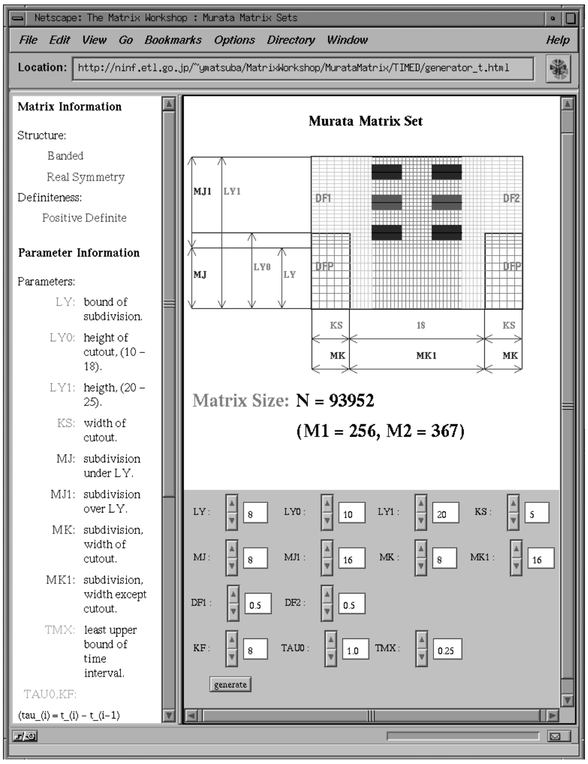

図 3　村田の問題に対する Web 画面

2.2　提供する行列

本システムでは, 行列の性質が明確な行列として条件数や固有値が解析的にわかっている行列を提供する．現在は以下の行列セットが実装されている．

“A Collection of Matrices for Testing Computational Algorithms”[2] (テスト行列集) では, 様々な種類のテスト行列を提供している．これには, サイズ固定の行列と, 任意のサイズで提供される行列とがあり, 本システムでは, 任意サイズで生成可能な行列を抜粋し, 提供する (図 2)．本システムでは行列の性質に関する情報として各行列に解析的に分かっている, 逆行列, 固有値, 条件数などの情報も提供する．提供する行列の種類は表 1のとおりである．

表 1　構造による行列の分類

Structure		symmetric	asymmetric
密行列	12	9	3
帯行列	15	13	2
三角行列	1		1
その他	6	4	2
total	34	26	8

村田らによる純拡散問題[4] では, 熱方程式の定常問題及び, 時間依存問題より導出される行列を提供する．これらの問題では, 2 次元の切欠きのある場に, 場所により段差をもたせた拡散係数を設定し, 熱拡散のシミュレーションを行なう．このモデルからは, 対称正定値 5 重対角行列が, 離散化の際に導出される．本システムで提供する行列には, この熱拡散のシミュレーションを行なう場の寸法や, 離散化の際のメッシュ数をパラメータ化することで, 行列サイズに自由度が与えられている (図 3)．

2.3　関 連 研 究

本システムと同様にテスト行列を提供するシステムとしては, NIST の R. Pozo らによって設計・開発された Matrix Market[1] がある．Matrix Market でも WWW 上でテスト行列を提供している．

提供する行列については, 数学的な基本行列に対しては Matrix Workshop, Matrix Market ともほぼ同等の機能を提供する．これらの基本行列を生成するためのシステムの違いとして, Matrix Market では Java ベースでクライアント側で生成しているが，Matrix Market では CGI を用いてサーバ側で対応していることが挙げられる．

実問題より導出される特定のテスト問題については, Matrix Market ではサイズ固定のテスト行列がテキスト形式で与えられていたり, 行列が導出されるルーチンが提供されていたりする．このように Matrix Market では実問題より導出される行列については自動生成することができない．これに対し Matrix Workshop では, 実問題より導出される行列に対してもサイズ，性質の変更が容易に対応できることを特徴としている．

また, Matrix Workshop では行列の生成を CGI を介して行なっている．そのため URL を指定してそのサイトの情報を取得できる API (Application Programming Interface) を用いれば, プログラム中から自由に行列データを取り込んで利用することが可能である．また Matrix Workshop で提供される行列に対し, プログラム中からでも行列サイズの変更が容易である．しかし, Matrix Market では Java で行列を生成させているためそれが出来ない．

3. Matrix Workshop と Ninf を用いた性能評価システム

3.1　概　　要

ネットワーク数値情報システム Ninf[3] では広域ネットワーク上に分散された計算資源や情報資源へのアクセスを容易に実現することにより, ハイパフォーマンスコンピューティング (高精度かつ高信頼性のある計算をより高速で高性能・高品質に実現すること) を支援することを目標とする．Ninf はクライアント・サーバ型のシステムであり, サーバを広域ネットワーク上に複数設定する．遠隔地のユーザは簡単な API を通じて Ninf のライブラリ関数を呼び出すことで, 高性能 / 高品質のライブラリを使用することができる．

また, Ninf のクライアント API である Ninf_call は遠

隔ライブラリを呼び出す際に，その引数としてURLを記述することができるようになっている．Ninfのこの機能により, Matrix WorkshopやMatrix Marketにおいて提供されるデータをNinfのライブラリサービスで提供されるプログラムに投入し，特定の計算機を指定しながら計算を続行することができる．この機能を利用して，「プログラム」「計算機」と「行列データ」を組み合わせて性能評価を行うことができる．

例えば, 新たにルーチンを作成した作成者はルーチンをサーバ管理者に登録してもらい, Ninfサーバ上で実行可能にすることで他の高速 / 高性能なルーチンと性能を比較することが可能となる. また, 様々なサーバ (計算機) 上で計算することが可能となるため, 計算機の性能評価も可能となる. さらに, Matrix WorkshopやMatrix Marketで提供される共通なテスト問題を用いることで, すでに他のプログラム作成者によって行なわれている評価と比較することも可能となる.

このような考えに基づき, 自動的な性能評価システムを構築した. 本稿では上記の予備評価として, 行列演算ルーチンに対しMatrix WorkshopとNinfの枠組を用いた評価を行なった. 評価にはLinpack Benchmarkのガウスの消去法を用い, 問題を変えサイズを変えて, Ninfサーバ上で実行した時の演算時間などの評価をする.

3.2 NinfとMatrix Workshopの連携

前述した通り, Matrix Workshopでは行列をCGIを介して生成させている そのためWebページの入力フォームに入力されたデータはURLのsearch partにエンコードされて，CGIスクリプトに渡される．

Ninfのクライアントは現在,C, Fortran, Java, Lispなどの言語で実装されている. Ninfサービスを利用するには以下のように, クライアントプログラム中よりNinfクライアントAPIである, Ninf_callを呼び出す.

Ninf_call(”dmmul”, n, A, B, C);

このようなNinf_callの記述に対し, 計算の入力としてURLで指定されたサイトにある情報を用いる際の例を以下に示す.

```
char *buffer;
buffer = "http://ninf.etl.go.jp/MatrixW
orkshop/cgi-bin/exec_matgen.cgi?Arg-1-n
=n&Exname=Example4_16&Content-Type=x-ma
th%%2Fmatrixmarket"
Ninf_call("dmmul", buffer, buffer, C);
```

このように引数としてURLを指定する事で, そのサイトのデータが計算の入力として用いられる. 上記の例では, 計算の入力となる第2, 3引数は行列要素が格納された配列である. そのため, URLのサイトからの入力データが適当な形式にパーズされて行列要素としてNinfサーバ上で実行されるルーチン (Ninf Executable, この場合dmmul) に引き渡されなければならない.

Matrix Workshopでは, 生成された行列要素はMatrix Market[1)] でも採用されている, Matrix Market Exchange Formatsを採用している. このフォーマットは行列データの交換 (やり取り) が容易なASCIIファイルフォーマットである. このフォーマットは, 実数, 複素数, 整数, patternの要素からなる行列に対して定義されている. 疎行列に対しては非ゼロ要素のみが記述され, また対称行列に対しては下三角 (もしくは上三角) 行列の要素のみが記述されるなど, データ量を削減する工夫がされている. 形式としては, ヘッダ行, コメント行, 行列データからなっている. 以下にMatrix Workshopで提供する実対称帯行列である, Example4_16の例を示す.

```
%%Matrixmarket matrix coordinate real \
     symmetric
%Example4_16: on page 75 of Text
%Structure: Banded, Real Symmetric
12 12    33
 1  1    5.00000000e+00
 2  1   -4.00000000e+00
 3  1    1.00000000e+00
        :
        :
```

%%で始まる行がヘッダ行である. ヘッダ行でその行列が密行列か疎行列かや, 実行列か複素行列か, 対称か非対称かが示されている. また, %で始まる行がコメント行である. ヘッダ行, コメント行以下が行列データとなっている. この例のように疎行列である場合は, 行列データの第一行目にその行列の行サイズ, 列サイズ, 非ゼロ要素数が示され, 二行目から一行ごとに行列の一成分が列番号, 行番号とともに列単位による順で出力される.

NinfサーバはMatrix Market Exchange Formatsの入力データに対し, 行列要素を密行列形式にパーズし実行ルーチンに引き渡す. また行列をメモリへ格納する際の順序のコンベンションが, CとFortranでは異なる. そのため行列要素を実行ルーチンに引き渡す際, 列単位で引き渡すか行単位で引き渡すかということが問題となる. Ninfでは, どちらの形式で要素を格納するかをNinf IDL中の記述によって判別している. Ninf IDLはNinfで提供されるライブラリのインターフェース情報を記述するための言語で, 先の行列積の例では以下の様になる.

```
Define dmmul(IN int n, IN double A[n][n],
       IN double B[n][n], IN double c[n][n])
Required "dmmul.o"
Language "Fortran"
Calls mmul(n,A,B,C);
```

行列の格納形式としてはデフォルトでは, 行単位による格納となるが, 上記IDLの記述のように,

```
Language "Fortran"
```

と指定することで, 列単位による格納となる.

Ninf_callのAPIとしては, 非同期呼び出し

Ninf_call_async, 並列に実行するトランザクション区間の指定 Ninf_begin_transaction, Ninf_end_transaction 等が用意されている.

4. 評　　価

4.1 評価環境

Matrix Workshop と Ninf を用いた性能評価システムの予備評価として連立 1 次方程式の解法を問題を変え, サイズを変えて LAN 上のワークステーションクラスタでの並列実行を評価した.

評価環境 (図 4) としては, 電総研の DEC Alpha (333MHz) の 32 台のワークステーションクラスタを用いた. 各ワークステーションは, 100Base/TX Ethernet スイッチで結合されている. クラスタのホストとして 1 台の Alpha ワークステーションクラスタがある.

Ninf サーバを各 32 台のワークステーションで実行し, クライアントとしてホストのワークステーションを用いた. また Matrix Workshop サーバとしては, 同じく電総研の SS-20 を用いた. Web サーバのサーバソフトは, Apache のバージョン 1.2.4 である.

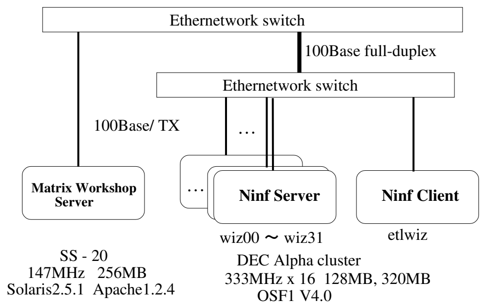

図 4　評価環境

4.2 Matrix Workshop の基本性能

Matrix Workshop の基本性能として

(1) overhead: Matrix Workshop サーバを呼び出すのにかかる時間
(2) generate: 行列の生成にかかる時間
(3) transfer: 生成された行列データを返す時間

を調べた.

また評価を行なうのに用いた問題の種類, 構造などを表 2に示した.

表 2　評価を行なうのに用いた問題

問題	構造	対称性	非ゼロ要素数
Hilbert	密	対称	nxn
Example3_9	密	非対称	nxn
Example5_21	疎	非対称	nx2-1

基本性能の計測には, 4.1 で示した評価環境と同様な環境で測定するため, Ninf サーバとする Alpha クラスタのうちの 1 台より Matrix Workshop の URL を発行し, Matrix Workshop サーバからの応答時間を調べた. 表 2で示した問題を用いた時の Matrix Workshop の基本性能は表 3のようになった.

表 3　Matrix Workshop の基本性能 [秒]

overhead		
0.2		
generate		
サイズ	1000	2000
Hilbert	18.0	1:08.55
Example3_9	36.48	2:17.65
Example5_21	0.82	2.30
transfer		
サイズ	1000	2000
Hilbert	0.31	1.63
Example3_9	0.52	1.68
Example5_21	.0	.0

今回使用した問題は行列要素が簡単な数式で定義された行列であるため, 行列の生成時間は密行列でサイズが 2000 と大きくなっても数分以内であった.

4.3 Alpha クラスタによる同時分割処理

実験には, Linpack Benchmark のガウスの消去法を用い, 問題サイズを 1 から 2048 まで変化させた時の演算時間を測った. 使用した問題は表 2の通りである.

問題の分割法としては, サイズ順に 32 台の各プロセッサに並列に問題を割り当てる. 各プロセッサに割り当てられた計算結果がすべて回収された後, 再び並列実行する.

図 5に, 密行列として Hilbert Matrix で実行した結果を示す. 図 5は, 問題サイズに対する Linpack の演算時間を示している. ソルバが直接解法であるため, 問題サイズに対して安定した演算時間を示している. 同様の実験を表 2で示した行列でも行なった. 図 6に Hilbert Matrix で Linpack を実行した時の演算時間に対する, Example3_9 および Example5_21 で実行した時の演算時間の相対差を示す. サイズが小さい領域では実行時間が小さいため, 測定誤差による変動の影響が大きいが, 演算時間は問題が変わってもほぼ同様の結果となっていることが分かる.

表 4に, サイズに対して逐次に演算を行なった時の総演算時間, Ninf サーバ 32 台で並列に実行した時の総演算時間 (逐次の結果を単純に 32 で割った時間), また問題ごとに, 分割処理による計測にかかった総計測時間を示す. 総演算時間は問題を変えてもほぼ同等なため, Hilbert Matrix での総演算時間を示した.

Ninf サーバ上のでソルバの演算時間に対し, 総計測時間は問題により大幅にばらつきがある.

以下に各問題に対して, サイズ 1000 と 2000 で単独に (分割処理しないで) 計測した結果を示す. クライアン

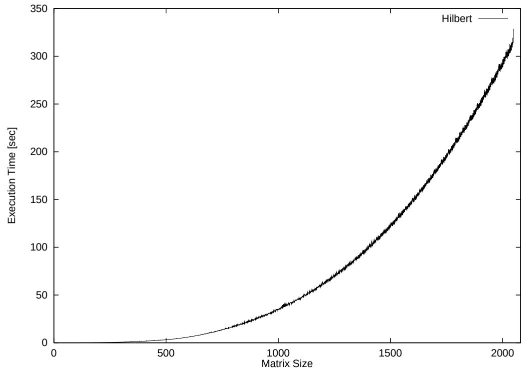

図 5　Hilbert Matrix を用いたときの演算時間

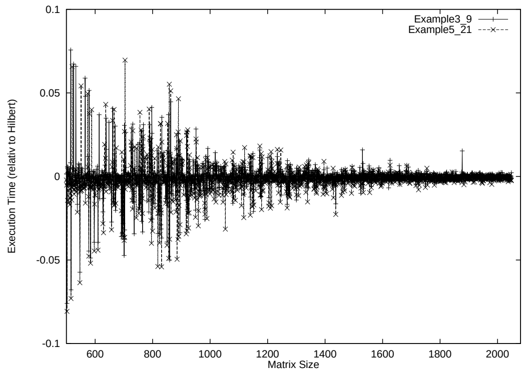

図 6　Hilbert Matrix に対する演算時間の相対差

表 4　総演算時間及び，総計測時間 [分]

総演算時間	
逐次処理	2651
並列処理	82
総計測時間	
Hilbert	512
Example3_9	899
Example5_21	136

トは, Alpha クラスタのホストのワークステーション, Ninf サーバには Alpha クラスタの内の 1 台を用いた.

表 5　各サイズにおける総計測時間 [秒]

サイズ	1000	2000
Hilbert	56	395
Example3_9	74	446

サイズ 2000 であっても計測時間の差は 50 秒程度であり, 表 4のような差が生じたのは Matrix Workshop を提供している Web サーバの安定性に問題があるためと思われる. 実際, Web サーバプログラムのエラーログに原因不明なエラーが残っていたため, 現在解析中である.

4.4　考　　察

今回の実験では, 一斉に 32 タスクを Ninf サーバに投げるため Ninf サーバから Matrix Workshop への行列生成の要求が集中し並列効果を下げている, ということが言える. 今回のように様々なサイズの行列に対するデータが欲しい場合には, Ninf において非同期呼び出しを実現する Ninf_call_async を用い, 並列実行することで計測時間の短縮が見込めるが, Matrix Workshop サーバへの負荷の集中ということを考えると計算が終ったサーバに対し随時計算要求を投げる, という枠組を考える必要がある.

5. まとめと今後の課題

本稿では, データコンテンツの提供サービスとして科学技術計算で特に重要となる行列を対象としたサービス「ネットワーク計算用行列工房: Matrix Workshop」の概説を行なうとともに, Matrix Workshop とネットワークコンピューティングの基盤システム Ninf を用いた性能評価システム実現のための予備評価として, 行列演算ルーチンを用いた評価を行なった.

Matrix Workshop では, アルゴリズム, プログラムをテストするのに適した様々な行列を提供し, それらの行列を任意サイズで提供することを可能とした. 特に大規模な行列については, 様々なデータをユーザが自由に作成するのが困難であるが, 本システムでは反復解法のテストに有効な実問題より導出される大規模疎行列の生成を自動化したことが特徴である.

今後の課題, 取り組みとしては,

- 計算機，プログラム，データの選択ができるような GUI の作成
- 固有値の分布や行列の性質を何らかの意味で表現するような項目の選定と構築，大量に CPU を必要としてもバックグラウンドで計算を行う試み
- 固有値等の計算と結果のデータベース化
- 性能評価への応用
- 実プログラム / アルゴリズムの評価

などが上げられる.

参 考 文 献

1) R.Boisvert, R.Pozo, K.Remington, R.Barrett, and J. J. Dongarra. http://math.nist.gov/MatrixMarket.
2) R. T. Gegory and D. L. Karney. *A Collection of Matrices for Testing Conputational Algorithms*. WILEY-INTERSCIENCE, 1969.
3) M. Sato, H. Nakada, S. Sekiguchi, S. Matsuoka, U. Nagashima, and H. Takagi. Ninf: A network based information library for a global world-wide computing infrastructure. In *Proceedings of HPCN'97 (LNCS-1225)*, pages 491–502, 1997.
4) 村田 健朗, 名取 亮, and 唐木 幸比古. 大型数値シミュレーション. 岩波書店, 1990.