

# Scilabにおける疎行列向け高精度演算の実装と評価

吉川慧子<sup>1</sup>, 斉藤翼<sup>1</sup>, 石渡恵美子<sup>2</sup>, 長谷川秀彦<sup>3</sup>

1) 東京理科大学大学院, 2) 東京理科大学, 3) 筑波大学  
日本応用数理学会 2012年度 年会, 2012年 8月28日~9月2日, 稚内全日空ホテル

## 研究目標

誰でも、簡単に、手間なく使える  
高精度演算環境の構築

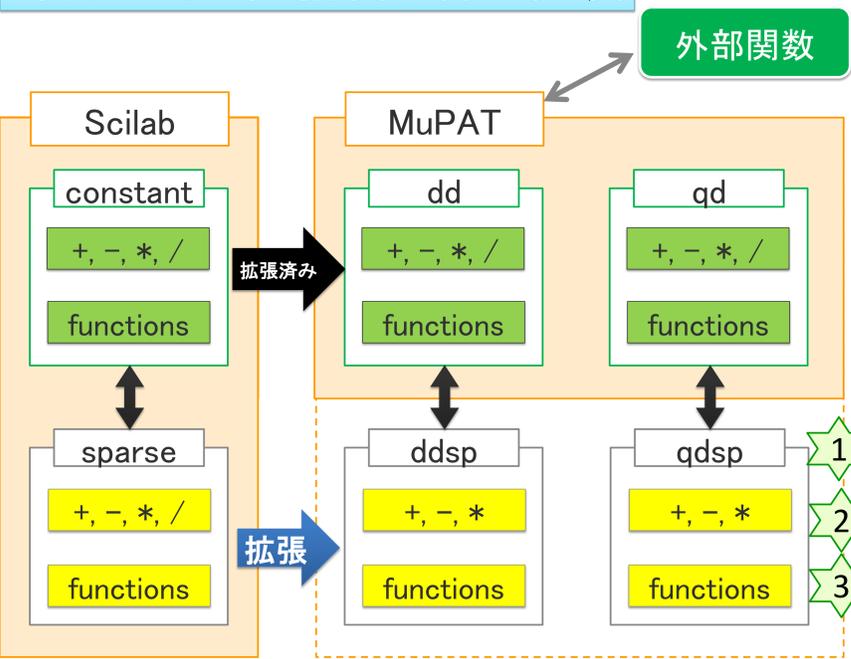
フリーの数値計算ソフトウェアScilabに  
2009年 4倍精度演算環境 QuPAT  
2011年 4・8倍精度演算環境 MuPAT  
を構築 [1,2]

## MuPAT (Multiple Precision Arithmetic Toolbox)

- ✓ 倍精度・4倍精度・8倍精度演算が同時に扱える
  - Double-Double(DD) ... 2つの倍精度数で4倍精度を表現
  - Quad-Double(QD) ... 4つの倍精度数で8倍精度を表現 [3]
- ✓ 2種類の実装を選択できる
  - Scilabのみの実装 ... OSやハードウェアに依存しない
  - C言語による外部関数を利用した実装 ... 高速な演算が可能 [2]
- ✓ 密行列データ型向けのみ対応

MuPATの  
高精度演算を  
疎行列データ型にも  
対応させたい

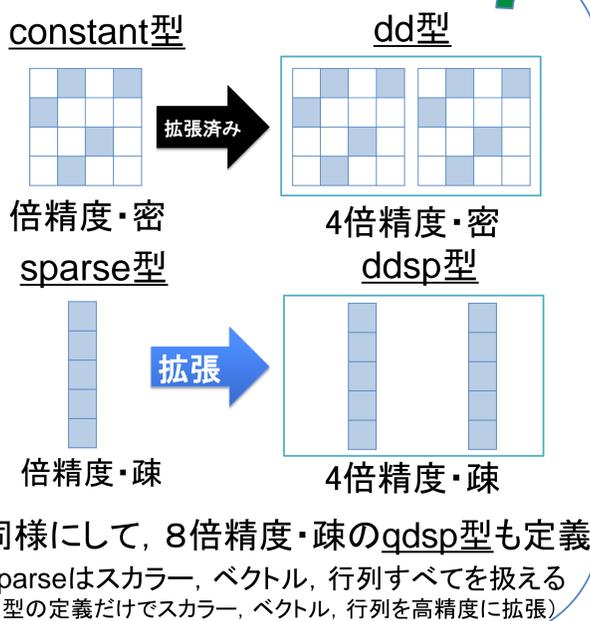
## 疎行列向け高精度演算の実装



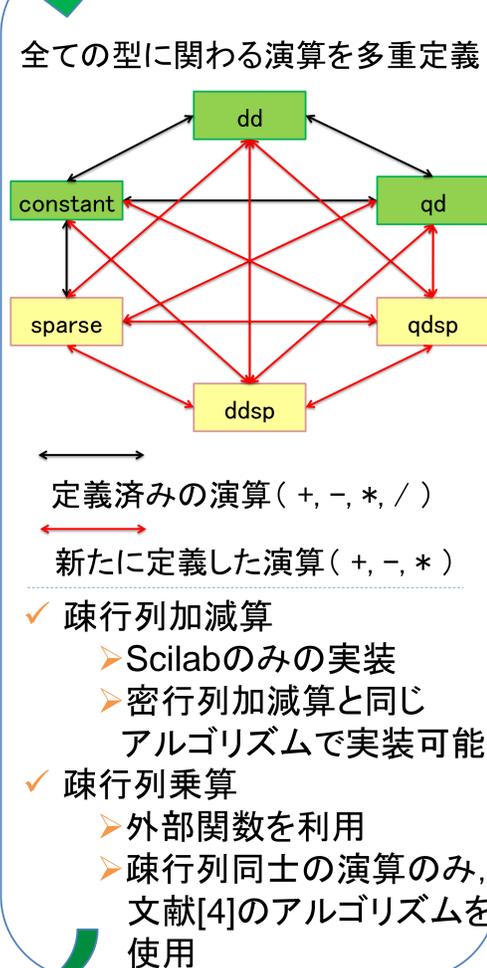
### 本実装の特徴

- ✓ 共通の演算子  $+, -, *$  や関数が利用可能  
→ 高精度演算への切り替えが簡単
- ✓ 6つのデータ型が同時に利用でき  
混合精度演算も可能
- ✓ 疎行列向け関数もほぼ網羅

### 1 型の定義



### 2 演算の定義



### 3 関数の定義

- ✓ sparse用の関数をddsp, qdspに拡張  
full 【密行列への変換】  
nnz 【非零要素数の抽出】...
- ✓  $A'$ 【転置】,  $A(i,j)$ 【成分の抜き出し・代入】等は6つのデータ型で同じ記述で扱える

## 評価: 行列演算

行列ベクトル積, 行列加算・乗算にかかる実行時間と  
使用メモリ量をqd(密データ型)とqdsp(疎データ型)で比較

➢ 使用した行列

右の性質を持つ  
1000次元の  
乱数疎行列

	非零要素率	使用メモリ量(MB)	
		qd	qdsp
A	1%	32	0.5
B	5%	32	2.4
C	10%	32	4.8

➢ 結果

	実行時間(秒)		計算結果の 使用メモリ量(MB)	
	qd	qdsp	qd	qdsp
Ax	4.44	0.14		
Bx	4.40	0.67		
Cx	4.49	1.36		
A+B	2.73	0.77	32	2.8
B+C	2.73	2.08	32	6.9
C+A	2.73	1.46	32	5.2
A*B	4000.81	9.00	32	18
B*C	3982.18	59.90	32	47
C*A	3964.51	16.33	32	30

x: 1000次元乱数ベクトル(qd型)

## 評価: CG法

```
CG法のScilabソースコード
17 A = symmtx2sp(path,mtxname); [m,n]=size(A);
18 x = ones(m,1);
19 b = A * x;
20
21 x = qdzeros(n,1); r = qdzeros(m,1);
22
23 r = b-A*x; p = r(:,1);
24 norm_r = norm(r); norm_b = norm(b);
25 eps = 1.0e-10; eb = eps * norm_b;
26
27 for co=1:n
28     r_1 = r * r;
29     pAp = p * (A * p);
30     alpha = r_1 / pAp;
31     x = x + alpha * p;
32     r = r - alpha * (A * p);
33     norm_r = norm(r);
34     if norm_r <= eb then
35         break;
36     end
37     r_2 = r * r;
38     beta1 = r_2 / r_1;
39     p = r + beta1 * r;
40 end
```

— の型の定義  
の書き換えのみ  
で精度の変更  
が可能

CG法に8倍精度の密行列演算と疎行列演算を  
適用し実行時間と使用メモリを比較

➢ 使用した行列

No.	行列	次元	非零要素率
1	nos4	100	3.47%
2	nos3	960	0.91%
3	bcsstk16	4884	0.62%
4	crystm02	13,965	0.17%

➢ 結果

精度	行列No.	実行時間(秒)		使用メモリ(MB)	
		1	2	1	2
密・8倍精度	1	0.9		0.09	
	2	262.9		7.4	
	3	8515.3		191.3	
	4	メモリーオーバーで計測できず			
疎・8倍精度	1	1/35	0.3	0.1	1/49
	2		14.6	0.5	
	3		241.1	3.8	
	4		66.8	5.0	

メモリ使用量・演算時間の削減を実現

密ではできなかった行列も計算可能に!

[1] T. Saito, E. Ishiwata and H. Hasegawa, Development of quadruple precision arithmetic toolbox QuPAT on scilab, ICCSA2010, Proceedings Part II, (2010).

[2] S. Kikkawa, T. Saito, E. Ishiwata and H. Hasegawa, Development and acceleration of multiple precision arithmetic toolbox MuPAT for Scilab, submitted to JSIAM Letters.

[3] D. H. Bailey, QD (C++ / Fortran-90 double-double and quad-double package), Available at <http://crd.lbl.gov/~dhbailey/mpdist/>

[4] Timothy A. Davis, Direct Methods for Sparse Linear Systems, SIAM, Philadelphia (2006).