

# Rにおける任意精度計算パッケージRmpenv

中野 純司 モデリング研究系 教授

## はじめに

現代の計算機では、実数計算は浮動小数点計算（通常、倍精度計算）で行われているので、桁落ちなどの数値計算上の誤差を避けることはできない。この影響を軽減するためには任意精度計算を行えばよい。

フリーの統計解析ソフトウェアRにおいても、この数値計算の誤差が問題になることがある。われわれは任意精度によるR上の計算環境（実数複素数の四則計算及び基本的な数学関数）のパッケージRmpenvを構築してきたが、今回Rに標準で実装されている計算子%\*(行列積)等の任意精度による行列計算機能や逆行列を求める関数を実装した。

行列の計算にはMPACK (<http://mplapack.sourceforge.net/>) を用いてこれをRから呼び出すよう機能を追加した。さらに統計利用者が容易に任意精度計算を利用できるような工夫を行った。

## 任意精度の数の表示

行列等も取り扱う上で、Rの標準書式とほぼ同等に任意精度の数値が識別可能なように表示できるようにした。

```
> library(Rmpenv) # デフォルトの仮数部は256bit
precision(mp.prec) was set in 256bit.
precision(mp.digits) was set in 39.
number of significant digits in decimal is 77.063679
```

次のパッケージを付け加えます: 'Rmpenv'

以下のオブジェクトは 'package:base' からマスクされています:

```
%*%, atan2, crossprod, log10, log2, matrix, outer, rcond, rep_len,
tcrossprod
```

```
> mp_prec(128) # 仮数部のビット数を128に
precision(mp.prec) was set in 128bit.
precision(mp.digits) was set in 19.
number of significant digits in decimal is 38.531839
> Const(pi)
[1] 3.141592653589793238
> # options(digits=??) 相当の機能
> options(mp.digits=38) # 128bitは10進数では38桁の精度
> Const(pi) # 表示桁数及び丸め処理も行う
[1] 3.1415926535897932384626433832795028842
> mpreal(1:5) # 任意精度の整数
[1] 1 2 3 4 5
> mpreal(1:5)/2 # 任意精度の有限小数
[1] 0.5 1.0 1.5 2.0 2.5
> mpreal(1:5)/3 # 任意精度の循環小数
[1] 0.3333333333333333 0.6666666666666667 1.0000000000000000
[4] 1.3333333333333333 1.6666666666666667
```

## 任意精度数と倍精度数の同時利用

任意精度を利用する上で最も陥りやすい「倍精度→任意精度」の単純変換による精度の喪失を引き起こさないように、関数 mp を作成した。これはRのインタプリタが式をパースした時点で実数および複素数が評価されてしまう（文字列として扱われない）点を、自動的に文字列に置換する事によって任意精度数として扱えるようにする関数である。

```
> library(Rmpenv)
> options(digits=22)
> 0.05 * 100 # 実数による通常の演算
[1] 5
> mpreal(0.05)*100 # 任意精度を使うが、倍精度の値を与えている
[1] 5.0000000000000002775557561562891351059
> mp(0.05*100) # 簡易関数'mp'による置換した実行
[1] 5
> mp(0.05*100, .debug=TRUE) # 置換した内容の確認(数値を文字列に再置換)
mpreal("0.05") * mpreal("100")
```

## 任意精度による計算例

任意精度の実数のみのパッケージとしてよく知られているRmpfrにも行列積は実装されているが、その実装はRで書かれているので非常に低速である。Rmpenvはそれよりかなりスピードアップされている。ただしもちろん、通常の倍精度での計算速度よりはかなり遅い。

### • 行列積

Rmpfr (CRANに登録されている既存のパッケージ)

```
> library(Rmpfr) # CRANにある既存の任意精度パッケージ
> N<-512
> A<-matrix(mpfr(runif(N^2),100),N,N) # 倍精度の乱数を100bitの任意精度に変換
> system.time(A%*%A) # R内での行列積
ユーザ システム 経過
20124.520 36.128 20193.861
```

Rmpenv

```
> library(Rmpenv)
> invisible(mp_prec(100)) # 仮数部の精度を100bitに
> N<-512
> A<-matrix(mpreal(runif(N^2)),N,N) # 倍精度を100bitの任意精度に
> system.time(A%*%A) # mpackのL3BLAS Rgemmを呼び出す
ユーザ システム 経過
32.388 0.004 32.447
```

通常の倍精度

```
> N<-512
> A<-matrix(runif(N^2),N,N)
> system.time(A%*%A) # 通常の倍精度による行列積
ユーザ システム 経過
0.008 0.000 0.008
```

### • 逆行列

Rmpenv

```
> library(Rmpenv)
> hilbert <- function(n) mp({ i <- 1:n; 1 / outer(i - 1, i, "+") })
> h32<-hilbert(32) # nが32のヒルベルト行列の作成
> rcond(h32) # 条件数の逆数を求める
[1] 7.36861126014253321049079118102104558063e-48
> sh32 <- solve(h32) # 逆行列を求める
> all(round(sh32 %*% h32, 15)==diag(32)) # 単位行列が求まったかの確認
[1] TRUE
```

通常の倍精度

```
> hilbert <- function(n) { i <- 1:n; 1 / outer(i - 1, i, "+") }
> h32<-hilbert(32) # nが32のヒルベルト行列の作成
> rcond(h32) # 条件数の逆数を求める
[1] 1.354346e-19
> sh32 <- solve(h32) # 逆行列を求める
solve.default(h32) でエラー:
システムは数値的に特異です: 条件数の逆数 = 1.35435e-19
```

悪条件の逆行列に遭遇しても任意精度であれば計算が可能である。また関数mpによって式の可読性は失われてはいない。

## おわりに

このパッケージ (Rmpenv) は現在も開発中なため関数名や仕様は今後大きく変わる可能性がある。現在はQR分解を開発中であり、固有値分解、特異値分解などの線形計算も利用できる任意精度パッケージを目指している。本研究は中間栄治氏（株式会社COM-ONE）との共同研究である。