

2016年度 Ruby開発プロジェクト助成 最終報告

2017年3月10日

岸本誠

2016年度 Ruby開発プロジェクト助成「Ruby/Numoの整備」について、2017年3月10日時点の（最終）報告である。（再）とあるのは、中間報告から（大きくは）変わっていない項目である。

1. 状況の公開について

（中間報告と基本的に同様）

GitHubの公開リポジトリ <https://github.com/metanest/numo-lapack> にて作業状況は公開している。numo-linalgとして作業開始時に存在していたコードからの、一部流用分があるが、主要部である `ext/numo/lapack/gen/tmpl/*.c` のコードは2016年12月末現在の合計で2630行であった。

2017年3月10日時点で3784行となっている。増加ぶんのほとんどはnormを実装した `norm.c` の1153行である（LAPACKに相当する機能が無かったため、wrapperではなく、このソースコード内に全機能を実装した）。

2. 中間での達成状況(再)

当初予定で中間時点までは線形代数ライブラリ（LAPACKインタフェース）の実装に注力することとしたが、ほぼ予定通り達成できている。

行列の要素の格納順に関して、`Numo::NArray`は現在において一般的なrowメジャーであるのに対し、LAPACKはFORTRANなどで伝統的なcolumnメジャーであり、変換層は組込まない原則としたことから、慣れないユーザには不親切と考え、Rubyで簡単な変換層を暫定的に実装した（将来的には本格的な実装による置換えを意図している）、`numo/linalg-compatible.rb` を追加した。

これにより `numpy.linalg` との機能を比較すると（一見似ているが、`scipy.linalg` は多くの追加機能や細かい変更などがある別物である）、次のURLで一覧できる関数のうちから、`linalg.???` となっている `linalg` の機能のうち、<https://docs.scipy.org/doc/numpy/reference/routines.linalg.html> `norm`（ノルム）、`cond`（条件数）、`tensorsolve`、`tensorinv` 以外は相当する機能が実装できている。このうち `norm` は当初の `Numo::Linalg` の `ToDo` 中にもあった機能としては唯一の未実装であるため、即急に注力する予定である。

3. 中間報告以降の達成状況

`norm`について、wrapperではなく全体を実装したため、予定外の時間を要したが、全て実装した。

予定ではその後 GNU Scientific Library (GSL) に取り掛かることとしていたが GSLはカバーする範囲がかなり広いライブラリで対象を絞れなかったため断念し、NArray および既実装部分のドキュメンテーションとテストの作業を行うこととした。

NArrayのドキュメントとして、ドキュメンテーションとテストについてドキュメントが無かったため整備した他、APIの修正にともなう変更、`narray-devel` から `ruby-numo (numo-narray)` への移行にともなう修正漏れなどの改修をNArray側をお願いした。

`numo-lapack` については、YARDドキュメントを追加中であり、近日中に全てのメソッドに記述を追加する予定である。テストはディレクトリを作成し、スタブ的クラスを作成してある。こちらに必要な内容を追加する予定である。

4. その他(再)

その他、自分以外の開発者がもし NArray にコントリビュートしようとした場合に迷うと思われる項目 (自分の疑問点を含む) についてイシューとしてまとめてあり、今後、メンターの田中先生と相談して確認する予定である。

また以上の`numo-lapack`の実装の他、行列式を求めるために総和計算が必要であったため行ったサーベイの結果、Pythonにおける `math.fsum` と、`cruby` の新規機能である `Array#sum` の間に微妙だが重要な違いがあることを確認した (`Array#sum`は「高精度」であるのに対し、`math.fsum`は (可能な限り) 「正確」である)。

関連して、正確な計算が必要である時に使われるであろうRubyの内蔵クラスである、`Bignum`と`Rational`において、最終結果を`Float`に変換する際の手続きが、正確さが失われるものであることを確認したため、改善案をプルリクエストとして提案中である。
(副作用として`rational.c`の`rb_gcd_gmp`にリークがあることを発見し、修正された (r57236))

(追加)このプルリクエストに関しては、コードは一部改良され、精度の悪化がある場合は少なくなっているが、当方が適切だと考えている手順による計算法については現在も議論中となっている。