

Ruby で Maple を動かすためのインターフェースの開発

情報科学科 西谷研究室 3528 村瀬愛理

1 研究目的

Ruby では数値計算のライブラリ開発が遅れており、Ruby 上では高等な関数 (素数を求めたり、最小公倍数を求めるなど) を使った数式処理を行うのが難しい。また、Ruby 以外の数式処理ソフトウェアを別に立ち上げて別々で作業したり慣れない別の言語を勉強し直したりするよりも、Ruby のみでプログラミングする方が開発速度の格段の向上が期待できる。そこで本研究では、Maple を Ruby 上で呼び出し、Maple に高等な関数や桁数の大きな数値を用いた計算をさせて、その結果を Ruby が取得するインターフェースライブラリの開発を目的とする。

2 手法

2.1 Maple とは

Maple は、1980 年にカナダ・ウォータールー大学で生まれた数式処理技術をコアテクノロジーとして持つ科学・技術・工学・数学 (STEM : Science, Technology, Engineering and Mathematics) に関する統合的計算環境である [1]。特徴として、たくさんの数学関数が用意されていること、大きな桁数の計算が可能であること、グラフの描画が簡単であり、かつ 3 次元のグラフの描画にも対応していることなどが挙げられる。数式を入力するだけで簡単に解を得ることができることから、多くの場で用いられている。

2.2 Maple 関数の類型化、出力の切り替え

今回、数多く存在する Maple の数学関数の中から整数論と行列に関するものを選抜し実装した後、入出力に関して類型化し、それぞれの出力に応じて wrapper を作った。

3 基本動作

入力された値の次の素数を出力する `nextprime` を用いて説明する。

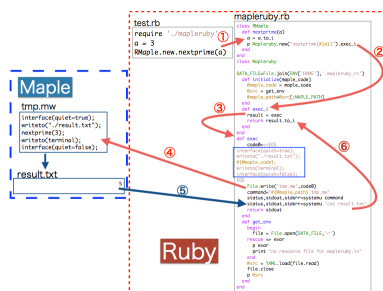


図1 mapleruby の基本動作。

1. `mapleruby` を `require` した上で使いたい関数を使う。
`RMaple.new.hogehoge` の `hogehoge` に使いたい関数名を入れる。
2. `RMaple` クラス内の `nextprime` 関数が呼び出される。その後、`Mapleruby` クラスの `exec.i` 関数へ”`nextprime(3)`”が出力される。この出力された文字列がそのまま Maple での計算に使われる。
3. 出力された文字列をさらに `exec` 関数へ出力する。
4. 青四角内の内容を Maple へと出力する。この時 `#{@maple_code}`; となっている部分に先ほどの”`nextprime(3)`”が入る。青四角の内容が Maple に出力され実行されることで得られた答えが `result.txt` に出力されるようになっている。
5. `result.txt` に出力された内容を Ruby 側で受け取り、`exec.i` に再び返す。
6. 返された値を `to.i` することで `int` 型に直して解を出力する。

4 本研究による成果

本研究で得られた成果をまとめると以下の通りである。

- RSA 暗号化における計算が、Ruby 単体で行うよりも約 100 倍の数まで扱えるようになった。
- 3 行 3 列の行列について逆行列や固有値、固有ベクトルを求めることができた。

従来 Maple と Ruby を行き来しなければならなかったような計算でも、`mapleruby` を用いることで Ruby 上のみで完結させられるようになった。

5 総括

Ruby だけで桁数の大きい計算や複雑な数学関数を必要とする計算を完結させられることが可能になった。今後は新たな関数の追加や、Maple の特徴である綺麗なグラフ描画ができることを生かしたグラフの出力を可能にするなどの課題が挙げられる。

参考文献

1. 「Maple(メイプル)とは」, サイバネット, <http://www.cybernet.co.jp/maple/product/maple/about.html>, 2017/02/01 アクセス。