

# AppleScript/Smile2 による数値解析プログラミング教育について

箕原辰夫\*<sup>1</sup>

Email: minohara@cuc.ac.jp

\*1: 千葉商科大学政策情報学部政策情報学科

◎Key Words AppleScript, Smile2, 数値解析, プログラミング教育

## 1. はじめに

慶應義塾大学湘南藤沢キャンパスで開設されている「スクリプト言語プログラミング基礎」という科目の秋学期のクラスにおいて、この2年間、AppleScript<sup>(1)</sup>とSmile2<sup>(2)</sup>を利用した数値解析のプログラミングの仕方について触れてきた。AppleScript 自体は、HyperTalk に続く、だいぶ古い言語ではあるが、macOS 上での定型処理の自動化では、未だに現役で使われている。Smile2 の開発環境上で AppleScript と Smile2 で用意されているライブラリと組み合わせることによって、Python で numpy, scipy, matplotlib を利用した数値解析と同等の環境を構築することができる。また、Smile2 自体は独自のグラフィックスライブラリを持っており、それを利用した GUI のプログラムも作成することが可能になっている。実際に授業中で実践した内容に基づき、Smile2 を使ってどのようなプログラミングが可能になるかを紹介する。

## 2. Smile/Smile2 について

### 2.1 Smile/Smile2 のリリース

Smile の最初の版は、AppleScript のリリースにあわせて、1995 年、フランスの SatImage<sup>(2)</sup> という開発会社から出されている。Smile は、SMI Limited Edition の頭文字をとったもので、SMI は、Scriptable Measurements on Images の頭文字をとったものとなっている。Smile の AppleScript 開発環境の上に、2D, 3D グラフィックスおよびグラフのプロットなどを含む、グラフィックによる視覚化を行なったもの SmileLab という製品になっている。Smile は、AppleScript の言語のバージョンが 2.0 以降になったことを受けて作り直しており、64bit 版の Smile2 がベータ版としてリリースされている。SatImage のホームページでは、まだ 32bit 版の Smile という名前のままになっている。現在は、SatImage によるベータ版の Smile2 のサポートは、個人ベースになっている。1990 年代の Smile 開発当初は有料のパッケージであったが、現在は Smile および Smile2 共に無料で macOS 上にフリーにダウンロードすることができる。なお、Smile2 のバージョンは、2024 年 6 月現在で、beta-4.0.0 版(232)となっており、ユーザからバグなどの報告がなされると、ビルドを新しくしている。実際に、2022 年の段階で、行列演算でのコマンドと Suite の組み合わせでエラーが生じてい

たので、連絡したら、すぐに修正した版をリリースしてくれた経緯がある。Smile2 は最新の macOS でも稼働する。

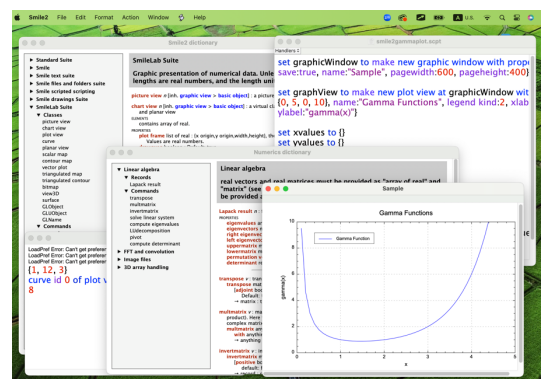


図1 Smile2 の稼働画面

### 2.2 SatImageOSAX

AppleScript は、AppleScript 2.4 ぐらいまでは、OSAX (Open Scripting Architecture eXtensions) という形で、サードパーティーがコマンドを拡張して使うことができた環境が用意されていた。しかしながら、それ 2.4 以降の版では、OSAX として標準の Scripting Additions 以外は使えないようになってしまった。それまで、SatImage は、SatImageOSAX という名称で、Smile で用意されているコマンド群 (Suite と呼ばれる) の一部を、無償で開放してきた。SatImageOSAX は、AppleScript に必要不可欠なものだが、それが使用できなくなってしまったので、現在は米国の Late Night Software 社の Script Debugger<sup>(3)</sup> のコミュニティにおいて、移植されたアプリケーション版 (SatImageOSAX.app) が用意されており、AppleScript から直接利用することができるが、これは配列や行列関係の Suite の一部コマンドが使えない状態になっている。このバグに連絡はしたが、修正して貰えていない。授業では、その部分は Smile2 で実習を行なうことにして、それ以外の Suite のコマンドを利用することにした。

### 2.3 Smile2 の Suite 群

Smile2 の SmileLab における Suite 群は、上記の SatImageOSAX 群の Suite と合わせて、様々なコマンドを内包する Suite が用意されている。以下のリストで、大文字・小文字の使い方は、Smile2 のコマンド辞書にあわせている。

- SatImageOSAX の Suite 群
  - SatImage Text Additions
  - SatImage File Additions
  - Mathematical Functions
  - Arrays
  - Arrays and List Utilities
- Smile Suite 群
  - Smile
  - Smile text suite
  - Smile files and folder suite
  - Smile scripted scripting
  - Smile drawings Suite
  - SmileLab Suite
  - Smile over IP
  - Smile IO devices
  - Web Suite
- Graphics Library 群
  - Prologue
  - Context
  - Paths
  - Text
  - Image
  - Layers
  - References
  - Miscellaneous
- Numerics 群
  - Linear algebra
  - FFT and convolution
  - Image files
  - 3D array handling
- Geometry 群
  - Geometry suite
  - Transformations
  - Constructions
  - Triangles
  - Drawings
- XMLLib 群
  - SatImage XML DOM
  - XML XPath, XSLT
  - XML Documents
  - XML Validation
  - SatImage Property List Additions
  - SatImage Pool Additions
  - XNF Additions

授業においては、この中の SatImageOSAX 群と Numerics 群、および Smile 群の中の SmileLab Suite を用いて、数値計算、数値解析関係の内容を扱ったが、今後は一般の 2D グラフィックスについても、扱いたいと考えている。

### 3. Smile2 を用いた数値計算・数値解析

#### 3.1 配列の計算

AppleScript は、Python の用に演算子に対しての拡張を

記述できないため、すべてコマンドで記述しなければならない。Python の numpy の array に対する要素ごとの加減乗除は、演算子ではなく、Arrays の Suite の中にある、addlist, sublist, multlist, divlist を使って行なう。以下は、array を作って、各要素に cos, sin を適用した後、スカラ一積を求める例になっている。s2 の各要素には、a の各要素 n の  $2\sin(n)\cos(n)$  の値が格納されている。

```
set a to createarray 10000 range {0, 2*pi}
set c to (cos a)
set s to (sin a)
set cs to multlist c with s
set s2 to multlist cs with 2
```

リストに対しての統計計算は、statlist コマンドによって、最大値・最小値（およびそのインデックス）・総和・平均・分散・標準偏差が入った statrecord という名前のレコードに求めることができる。

```
set a to {0, 1, 2, 3, 4, 5, 6}
log (statlist a)
-- {maximum:6.0, maximum index:7,
   minimum:0.0, minimum index:1,
   sum:21.0, mean:3.0, variance:4.0, stdev:2.0}
```

また、乱数によるリストの初期化とヒストグラムの作成も行なうことができる。

```
set a to randomarray 100 range {0, 10}
set b to histogram a samples 5 minimum 0 maximum 10
log (b as list of real)
-- {{1.0, 3.0, 5.0, 7.0, 9.0}, {25.0, 20.0, 18.0, 16.0, 21.0}}
```

-- 最初のリストが各区間の中央値、次のリストが個数

#### 3.2 線形代数（行列）の計算

行列について、要素ごとの和差積商を求めるには、リストの addlist, sublist, multlist, divlist がそのまま使える。加えて、行列の積を求める multimatrix や、転置行列を求める transpose、逆行列を求める invertmatrix のコマンドがある。

```
set x to {3, 1, 2, 2, 1, 7, 4, 5, -1}
set m to {class:matrix, ncols:3, nrows:3, array of real:x}
set tm to transpose m
set im to invertmatrix m
set ident to multmatrix m with im -- 単位行列になる
```

正方行列の行列式を求める compute determinant、固有値（複素数のものも含む）および各固有値に対応した固有ベクトルを求める compute eigenvalues、LU 分解を行な

う LUdecomposition などのコマンドが用意されている。

```
set mat to {class:matrix, ncols:3, nrows:3, array of real:
  ({7, 6, 9}, {-7, 2, 2}, {4, 3, 2})} as array of real
log (compute determinant mat)
set eugenrec to compute eigenvalues mat with eugenvecs
log eigenvalues of eugenrec
log eigenvectors of eugenrec
set lapack to LUdecomposition mat
log uppermatrix of lapack
log lowermatrix of lapack
```

連立一次方程式の解を求める、すなわち、 $AX=B$  (ただし  $X$  は正方行列、および  $B$  は列ベクトル) となるような行ベクトル  $A$  を求める solve linear system コマンドも用意されている。

```
set mat to {class:matrix, ncols:3, nrows:3, array of real:
  ({1, 3, 4}, {-5, 2, 2}, {4, 3, 2})} as array of real
set vecb to {5, 2, 4} as array of real
log (solve linear system mat RHS vecb)
```

### 3.3 高速フーリエ変換・補間・曲線回帰の計算

高速フーリエ変換は、音声データなどに適用させることができる fft1d コマンドと、画像データなどに適用させることができる fft2d コマンドが用意されている。両コマンド共に逆フーリエ変換を行なう inverse オプションが用意されている。

```
set x to createarray 144 range {-pi, pi}
set y to randomarray 144 range {-0.75, 1}
set ys to runningsum y
set {kreal, kimag} to fft1d ys
```

補間を線形補間およびスプライン補間のどちらかを選択して行なえる (通常はスプライン補間) interpolate コマンドが用意されている。

```
set x to runningsum (randomarray 10)
set y to runningsum x
set xsample to {1, 2, 3, 4, 5, 6} as array of real
set yvalues to interpolate {x, y} at xsample
log yvalues as list of real
```

多項式による曲線回帰 (curve fitting) を行なう fitpolynomial コマンドが用意されている。多項式の次数を degree オプションで指定することができる。

```
set x to runningsum (randomarray 10)
set y to runningsum x
```

```
set fitrec to fitpolynomial {x, y} degree 3 number
formatting "%.3g"
log (fit string of fitrec)
--"2.23-2.58*x+2.07*x^2-0.113*x^3" 乱数なので
--実行ごとに毎回変わるが、3次多項式で記述
```

その他に畳み込み (convolution) を行なう filter コマンド (2次元)、filterarray (1次元)、convolve コマンド、および correlate コマンドが用意されている。また、3D array handling の Suite には、3次元配列 (テンソル) に関してのコマンドも用意されている。

### 3.4 グラフのプロット

グラフのプロットは、Python の matplotlib と同様のグラフ作成が、Smile drawings Suite と SmileLab Suite の中に、クラスやコマンドが用意されている。x座標とy座標をリストで指定した折れ線グラフは、以下のように記述することができる。

```
set w to make new graphic window
set v to make new plot view at w
set c to make new curve at v
set xdata of c to {1, 2, 3, 4, 5}
set ydata of c to {1, 4, 9, 16, 25}
draw w
```

この方法で作成するよりも簡便に以下の QuickCurve 関数を使って作成することができる。この関数は、QuickPlotLib (内部的には Smile2 の Class Scripts の Context additions にある) ライブラリとして収められており、3D のグラフも含む様々なユーティリティ関数が収められている。上記と同じ内容をこの関数を利用して、記述すると以下の1行になる。

```
QuickCurve({1, 2, 3, 4, 5}, {1, 4, 9, 16, 25}, 0)
```

最後の 0 (または null で指定する) は、トップの plot view を作って表示する指定で、このパラメータで、包含される plot view をすることもできる。それを使えば、以下のように複数のグラフを1つの plot view の中に描画することができる。

```
set w to make new graphic window
set v to make new plot view at w
set a to createarray 10000 range {0, 2 * pi}
set {c, s} to {(cos a), (sin a)}
QuickCurve(a, c, v)
QuickCurve(a, s, v)
```

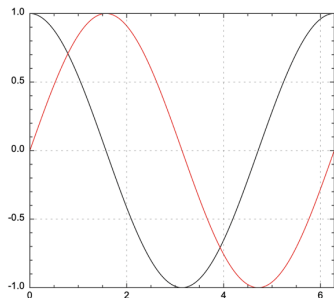


図2 プロットされたグラフ

## 4. Smile2 でのグラフィックス描画・他

### 4.1 幾何図形の描画

Smile2 の Graphics Library の Suite 群においては、2次元ベクトルの演算も含む、幾何図形の描画用のクラスやコマンドが収められている。これは、簡易描画関数のライブラリになっており、例えば、以下のスクリプトは、塗り潰された円を表示する。

```
set w to BeginFigure(0)
set {pagewidth of w, pageheight of w} to {500, 500}
SetFillColor({0, 1, 1}) -- {R, G, B, A}
SetPenColor({1, 0.5, 0.7, 1})
CirclePath({250, 250}, 50)
DrawPath("fill-stroke")
EndFigure()
```

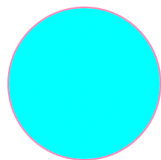


図3 描画された内容

### 4.2 GUI コンポーネント

Smile の中では、ボタンなどの独自のコンポーネントを持つダイアログを作成し、ボタンが押されたら、付随したスクリプト中の対応するコールバック関数が呼ばれて、処理をする GUI(Graphic User Interface)を作成する機能がある。しかしながら、Smile2 ではまだ未実装になっている。

### 4.3 XML による DOM の操作

Smile2 は、XML による DOM (Document Object Model) のスクリプトを組むことができる。以下は、SatImage の XML に関してのチュートリアルファイルを読み込んで、XML の DOM を解析し、<tr>のタグの中にかかれたテキストをログに表示するものである。このスクリプトは、SatimageOSAX.app に対しても実行することができる。Smile2 においては、XMLLib の Suite 群の中のクラスや

コマンドを利用して記述されている。

```
set xf to XMLOpen "http://www.satimage.fr/" &
"software/en/smile/xml/xml_tutorial.html"
encoding «class utf8»
set theroot to XMLRoot xf
set theBody to XMLChild theroot index 2
set theTables to XMLFind theBody
name "table" with all occurrences
repeat with tb in theTables
set tableRecords to XMLFind tb
name "tr" with all occurrences
repeat with tr in tableRecords
log (XMLGetText tr)
end repeat
end repeat
XMLClose xf
```

このように、Smile2 および SatimageOSAX では、Python の Beautiful Soap のように Web から、HTML や XML データを読み込み、その内容をスクリプトに取り込むことが可能になっている。

## 5. おわりに

Python2 がリリースされて、numpy, scipy, matplotlib が出てくるまでは、個人的に SmileLab/Smile を利用してきた経緯がある。数値計算・数値解析が Python で記述できるようになって、だいふ楽になったが、AppleScript でも同じことができるということを、これからのプログラミング教育で伝えていこうと考えている。Python も元はそうだが、AppleScript は、余分な記号を極力廃したプログラムを記述でき、C 言語系列のプログラミング言語の思想と対局にあるように思える。1行から記述できるインタープリタをベースに、数値計算や解析が、英語のように記述できる魅力は捨てがたいものがある。Smile2 はまだベータ版であり、Smile で実装されていた機能の一部は未実装になっているが、今後の精力的な移行のための開発と処理の安定化に期待したい。

## 参考文献

- (1) AppleScript Language Guide, Apple Computer, <https://developer.apple.com/library/archive/documentation/AppleScript/Conceptual/AppleScriptLangGuide/>, (2024年6月閲覧).
- (2) Smile, Smile2, and SmileLab, SatImage, <http://www.satimage.fr/software/en/index.html>, (2024年6月閲覧).
- (3) Script Debugger, Late Night Software, <https://latenightsw.com> (2024年6月閲覧).