

SILE 上の描画環境の構築の試み

呉工業高等専門学校・一般科目 深澤 謙次 (Kenji Fukazawa)
Department of General Education,
Kure National College of Technology

1 はじめに

数学や物理学の研究者や教育者の中には、論文や教材の作成に \LaTeX を用いる者が多くいるが、 \LaTeX を使用していると、 \TeX マクロ言語を使って機能を拡張することを考えるようになる。しかし、 \TeX マクロ言語はよく使われている一般的なスクリプト言語とはかなり異なっており、この言語で \TeX マクロを作成し \LaTeX の機能を手軽に拡張することは簡単ではない。

一般的なスクリプト言語と同じようにコードを書くことによって機能を拡張することができれば、例えば、外部プログラムを利用することによって、数学の問題の解答を自動的に生成するようにすることができるはずである。その他の例は、成績一覧表を作成するなど、生データを処理した結果を用いて文書を作成する場合である。成績一覧表を作成する場合には以下のステップを踏む必要がある。

(1) 生データ (raw data) をデータ処理する

(2) 処理したデータを \LaTeX 形式に変換する

(1) を行うためには、Excel などの表計算ソフトを使うか、perl や python などのスクリプト言語を使う必要がある。その後、何らかの方法で処理した (1) のデータを (2) のように \LaTeX 形式に変換する必要があるが、もし、データの修正がある場合は、(1)、(2) を繰り返さなければならず、手間が掛かる。これらの処理をするマクロを \TeX マクロ言語を使ってプログラムすることも可能であろうが、このマクロ言語は通常の script 言語のように手軽に使えるものではないと思われる。

この状況を改善する 1 つの方法は、スクリプト言語を組み込み、この言語で書かれた組版システムを使用することである。そのような組版システムの 1 つとして、SILE がある。SILE は組み込み用のプログラミング言語 Lua で書かれており、 \LaTeX 風の独自の入力フォーマットと XML による入力フォーマットの 2 種類のフォーマットに対応しておりソースファイルを処理して、直接 PDF ファイルを出力する組版システムである。

教材の作成、特に数学教育に関する教材の作成において、正確で綺麗な挿図が必要であるが、SILE は画像を挿入する機能はあるが \LaTeX の picture 環境に相当するものが無く、 $\text{K}\epsilon\text{T}\text{pic}[1]$ のような正確できれいな表現力のある図を作成する方法が無い。そこで本論文では、SILE 上で図を作成する試みについて報告する。

2 組版システム SILE

2.1 SILE の概要

SILE は Simon Cozens 氏が開発している、プログラミング言語 Lua で書かれた組版システムである。L^AT_EX 風の独自の入力フォーマットと XML による入力フォーマットの2種類のフォーマットに対応しており、ソースファイルを処理して、L^AT_EX のような独自の DVI ファイルを作らずに直接 PDF ファイルを出力するように作られている。フリーな OpenType レイアウトエンジンである HarfBuzz を用いた OpenType レイアウト機能を備えており、日本語を含む様々な言語・用字系をサポートしている。さらに Lua のスクリプトを自分で作成することにより、機能を拡張することができる。

SILE は T_EX から、box-and-glue モデル、hyphenation アルゴリズム、line-breaking アルゴリズムの3つを継承したが、一方で、T_EX で使われているいくつかの技術を、以下のように標準的な技術で置き換えている。

<u>T_EX decisions</u>	<u>⇒ Industry standards</u>
METAFONT	⇒ TrueType fonts
DVIs	⇒ PDFs
7-bit ASCII	⇒ Unicode
macro languages	⇒ Lua language

この結果得られた SILE の特徴をまとめると、以下のようになる。

- フレームを用いた複雑なページレイアウトへの対応
- 高水準のプログラミング言語（Lua）を用いた組版エンジンの容易な拡張可能性
- XSL スタイルシートを用いずに XML からの PDF の生成
- グリッド上でのタイプセット
- Fontconfig を介したフォントアクセス

2.2 SILE で作成した文書の例

ここでは、SILE で作成した日本語の文書の例を示す。日本語の文書を作成する場合は、document 環境のオプションとして class=jplain を指定し、日本語のフォントを `\font[family=Takao P明朝 language=ja]` のように指定することによって、日本語の文書が作成できる。

```
\begin[papersize=a4,class=jplain]{document}
\script[src=packages/ruby]
\font[family=Takao P明朝,language=ja]%
```

```
\ruby[reading=にほん]{日本}は\ruby[reading=むかし]{昔}から「寄付しない国」として知られていました。対名目GDP比寄付金規模はイギリスの四分の一、アメリカの九分の一です。世界寄付指数によると、個人寄付はアジアの諸国の中でワースト3位です。社会の評論家によると、いくつかの理由があります。ウチとソトの違いがあって、ソトへの関心が弱いと言われています。また、日本政府と市町村が幅広い市民サービスを提供しますので、海外のように社会福祉に貢献する寄付の必要がありません。
```

```
しかし、それが全体像でしょうか？2011年から日本の寄付文化が変わっているという証拠が見えてきました。2011年といえば、東大震災でした。2001年で震災に関する寄付金が6000億円を超えましたが、それと別に災害と関係がない個人寄付が5182で、合計で1.1兆円でした。2013年度では、個人寄付が7000億円まで上がって、歴史的に法人寄付と同等でありました。日本で寄付文化が根付いているところです。
```

```
\end{document}
```

図 1: 日本語文書のための SILE ソースファイルの例

このソースファイルを SILE で以下のようにコンパイルすることによって、pdf ファイルが作成できる。

```
sile <ファイル名> (ファイルの拡張子 “.sil” は省略可)
```

図 2 から、漢字のルビが正しく振られていることがわかる。

```
日本はにほん昔むかしから「寄付しない国」として知られていました。対名目GDP比寄付金規模は分の一です。世界寄付指数によると、個人寄付はアジアの諸国の中でワースト3位での理由があります。ウチとソトの違いがあって、ソトへの関心が弱いと言われています。民サービスを提供しますので、海外のように社会福祉に貢献する寄付の必要がありま
```

```
しかし、それが全体像でしょうか？2011年から日本の寄付文化が変わっているとい  
えば、東大震災でした。2001年で震災に関する寄付金が6000億円を超えましたが  
寄付が5182で、合計で1.1兆円でした。2013年度では、個人寄付が7000億円まで上  
りました。日本で寄付文化が根付いているところです。
```

図 2: 図 1 のソースファイルから SILE で作成される pdf ファイルの一部

縦書きの日本語の文書を作成する場合は、document 環境のオプションとして layout=tate を追加し、日本語のフォントを指定する。例えば、

```
\begin[papersize=a4r,class=jplain,layout=tate]{document}  
\font[family=Takao P明朝,language=ja]
```

のように指定すればよい。図 1 を縦書きにした pdf ファイルは図 3 のようになる。

日本は昔から「寄付しない国」として知られて、
 分の一です。世界寄付指数によると個人寄付はア
 理由があります。ウチとソトの違いがあつて、ソトへ
 サービスを提供しますので、海外のように社会福祉
 しかしそれが全体像でしょうか？2011年か
 1年といえば、東大震災でした。2001年で震災
 係がない個人寄付が5182で、合計で1.1兆
 歴史的に法人寄付と同等でありました。日本で寄
 東大震災の他一つの大きな変化は、2011年
 する Charity Aid Foundation の調査によると
 税制の取り扱いです。寄付金控除がある国の個人
 この新しい環境では「あなたがたの富のあるところ
 4)がますます大事であると思います。日本の個人
 いますか？ほとんどの教会は自分の活動や経営費
 を支えている教会は非常に少ないです。たまに外界
 国を建てようという教会の文化は日本でありませ
 一般社会に比べて「喜んで与える」教会の寄付文化
 この間、私たちのハウスマスターでは献金という理

図 3: SILE で作成される縦書きの日本語文書の一部

SILE は Lua 言語で実装されているので、Lua でスクリプトを書くことによって、自分が望むようにカスタマイズすることができる。以下は Lua スクリプトを用いた簡単な例である。

```
\begin{script}
  for i=1,10 do
    SILE.typesetter:typeset(i .. " x " .. i .. " = " .. i*i .. ". ")
    SILE.typesetter:leaveHmode()
  end
\end{script}
```

1 x 1 = 1.
 2 x 2 = 4.
 3 x 3 = 9.
 4 x 4 = 16.
 5 x 5 = 25.
 6 x 6 = 36.
 7 x 7 = 49.
 8 x 8 = 64.
 9 x 9 = 81.
 10 x 10 = 100.

図 4: Lua スクリプトを利用した出力例

3 SILE 上での描画方法

本研究論文では、SILE 上での描画方法として、以下の 3つの方法を検討する。

- (1) “Lua Graphics Toolkit” (Lua の graph モジュール) を利用する方法
- (2) “Asymptote” (Vector Graphics Language) を利用する方法
- (3) “libtexpdf” (T_EX の dvipdfmx から抽出した PDF library) を利用する方法

3.1 “Lua Graphics Toolkit” を利用する方法

SILE は Lua 言語を用いて実装されているので、Lua の graph モジュールである Lua Graphics Toolkit を利用して図を作成することができる。以下は、Lua Graphics Toolkit を利用して関数のグラフを表示する例である。

```
\begin{script}
graph = require "graph"
function f(x) return math.sin(x) * x end
p = graph.plot("y = sin(x) * x")
line = graph.fxline(f, 0, 25)
p:addline(line, "red")
p:show()
p:save("/tmp/lgm")
os.execute("ppmtjpeg /tmp/lgm.ppm > ./fig/lua-graph-module.jpg")
\end{script}

\begin{center}


```

作成される図は次のようになる。

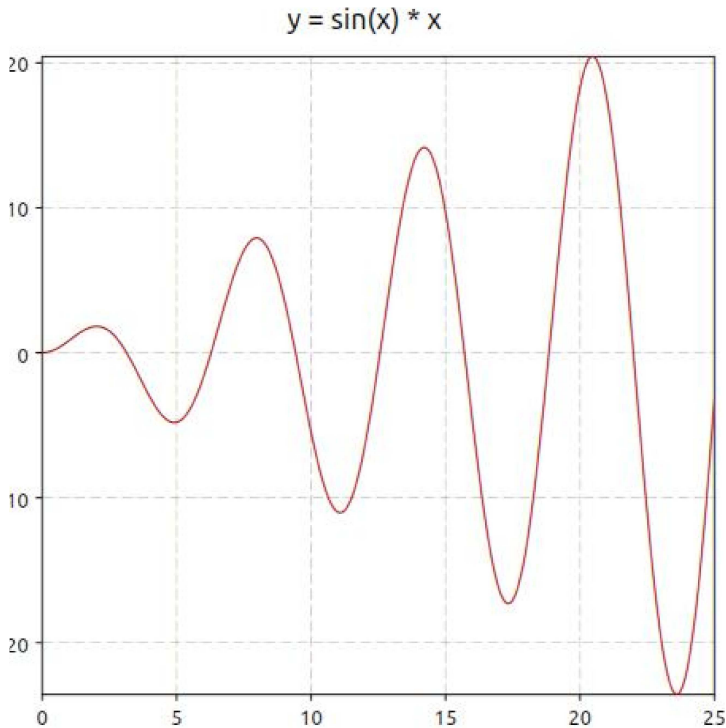


図 5: Lua Graphics Toolkit を利用した関数のグラフの例

3.2 “Asymptote” を利用する方法

Asymptote は metapost をベースに設計された強力なベクターグラフィック言語であり、様々な図を作成することができる。SILE 上で Asymptote で作成した図を表示させられるが、以下の例のように、SILE のソースファイルの中に Asymptote のソースを書いて図を表示させることもできる。この例では、Lua スクリプトとして実行するようにしているが、Asymptote 用の環境を定義して、その中にコードを書くようにすることもできる。

```
\begin{script}
asy_script="import geometry;"
asy_script=asy_script.."label("\$\displaystyle\frac{a}{\sin A} =
\frac{b}{\sin B} = "
asy_script=asy_script.." \frac{c}{\sin C} = 2R$",
truepoint(S), \S);"
f = io.open( "/tmp/theorem_sin.asy", "w")
f:write(asy_script)
f:close()
```

```
os.execute("asy -f pdf -o fig/theorem_sin.pdf /tmp/theorem_sin.asy")
\end{script}

\begin{center}
\img[src=./fig/theorem_sin.pdf]
\end{center}
```

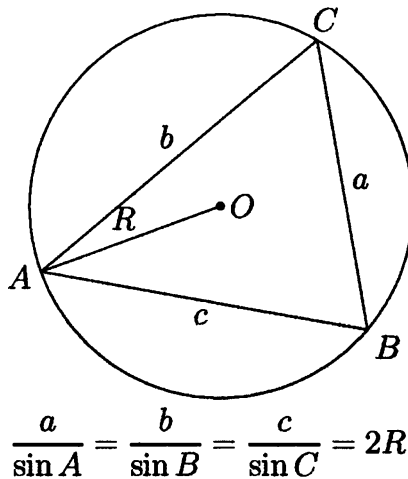


図 6: Asymptote を利用した図の例

3.3 “libtexpdf” を利用する方法

DVI ファイルを PDF ファイルに変換するプログラム `dvipdfmx` を元に作成された `libtexpdf` という PDF ライブラリがあり、この中に以下のような関数が含まれている。

`texpdf_dev_moveto`, `texpdf_dev_rmoveto`, `texpdf_dev_closepath`, `texpdf_dev_lineto`,
`texpdf_dev_rlineto`, `texpdf_dev_curveto`, `texpdf_dev_vcurveto`, `texpdf_dev_ycurveto`,
`texpdf_dev_rcurveto`, `texpdf_dev_arc`, `texpdf_dev_arcn`,

これらの関数を利用すれば、SILE で作成される文書の中に図を挿入することができる。

4 まとめと考察

本研究論文では、SILE 上で描画する 3 つの方法について検討した。

- (1) “Lua Graphics Toolkit” (Lua の `graph` モジュール) を利用する方法
- (2) “Asymptote” (Vector Graphics Language) を利用する方法
- (3) “libtexpdf” (TeX の `dvipdfmx` から抽出した PDF library) を利用する方法

(1) の方法で関数のグラフを表示することはできるが、 x 方向と y 方向のスケールを同じにする方法がなく、正確な図を描く必要がある場合には向いていない。(2) の方法は 3D 画像も含めて、ほとんどどんな図も作成できるが、複雑な図を描く場合は、SILE 上でコードを書くのではなく、Asymptote で作成しておく方がよい。(3) の方法で図を描くことはできるが、 $\text{K}\epsilon\text{T}\pi\text{c}$ のような正確できれいな図を作成するには、そのためのクラスを定義する必要がある。

SILE は Lua 言語が使われているので、工夫次第でいろいろなことができ、大きな可能性を秘めている。例えば、e learning system STACK 上で学生が問題を解く場合、問題のページ・解答と解説のページ・誤答の場合の原因を解消するためのページなど、複数のページが表示されるが、学生がそれらのページを保存したい場合、現状では、それぞれのページが表示されているときに画面を保存するしかないが、SILE を利用することで、1つの PDF ファイルとして保存させられると考えられる。SILE は、今後の発展が期待できる有望な組版ソフトである。

謝辞

本研究は、京都大学数理解析研究所共同事業「数学ソフトウェアとその効果的教育利用に関する研究」による成果である。

参考文献

- [1] $\text{C}\text{A}\text{S}\text{T}\epsilon\text{X}$ 応用研究会 (編), 「 $\text{K}\epsilon\text{T}\pi\text{c}$ で楽々 $\text{T}\epsilon\text{X}$ グラフ」, イーテキスト研究所, 2011.