



一般社団法人

国際数理科学協会会報

No.110/2019.4

編集委員：藤井淳一（委員長）

目次

総会議事録

決算予算表

貸借対照表

* 寄稿

総会議事録

2019年3月24日

場所：大阪教育大学 天王寺キャンパス 本館 306 号室

時間：2019年3月24日 午後3時～5時

議長：代表理事 植松康祐

出席者（代議員）：植松康祐、藤井淳一、富永雅、八木厚志、濱田悦生、地道正行、会沢成彦、道工勇、石井博昭、古澤仁

出席者（顧問）藤井正俊、長尾壽夫

総代議員数 30 名、出席者数 28 名（委任状含）、総会は成立。

【決算の概要】

- 1) 海外書籍取次業者（EBSCO）からは前年に引き続き多くの注文がありましたが、決算日までに未回収分が発生していました。（3月までに振込みがありました。）
- 2) 業務効率化を引き続き行い、全体では、100万円程度の赤字です。
- 3) 印刷費は、ほぼ予算通りの数字となっています。

【予算の概要】

- 1) Jstage を論文公開にあたり利用することになりました。過去の論文をシステム対応する人件費として予算（15万円：担当辻本氏）を計上しています。
- 2) 印刷費については、通常3回で約50万円ですが、Vol.81-3(FIM 特集号)が通常よりややページ数が多かったこと、海外の全ての著者に贈呈することから注文部数が多くなり、予算の上乗せ（+5万円）を行いました。来年度も SCMJ 誌を三回発行予定しております。

【協会活動】

- 1) SCMJ 誌の発行については、既に Vol.81-1.2.3 を発行致しました。Vol.81-3(FIM 特集号) 発行時に、著者からの細かい手直しに対応したことで、スケジュールがやや遅れました。今後調整いたします。
- 2) 投稿総数、国内・海外、採択率の報告

- 3) 論文掲載にあたっての順序は、審査終了日を基準としていますが、博士論文等で早期の掲載や掲載刊号ページ数の確定を要請される場合などイレギュラーなケースへの対応が話われました。
- 4) 従来から問題になっていた、論文ページの統一を過去に遡ってすすめています。(e-XX ページを廃止し、雑誌ページに統一)
- 5) デジタルオブジェクト識別子 DOI 利用と Jstage 公開を夏までには行います。とりあえず、電子書籍化されている Vol.76～最新号までを公開します。また、細かい手続き (DOI は、編集部で付けるか、著者に任せるか) などは今後決めていきます。
- 6) Jstage という論文公開の永続的プラットフォーム・ホーム導入により今まで SCMJ 誌で発行した全論文のウェブ上公開をこれから進めることになりました。
- 7) 2019 (3 月) までの新規会員の報告 (敬称略)
 - ① 三好博之 (京都産業大学理学部数学科教授) 専門分野: 圏論・数理論理学・理論計算機科学
 - ② 陰山真矢 (関西学院大学理工学研究科博士研究員) 専門分野: 非線形現象
 - ③ 太田家健佑 (大阪大学大学院情報科学研究科) 専門分野: 応用解析・情報科学・経済学
 - ④ 堀口正之 (神奈川大学理学部教授) 専門分野: 計画数学・統計数学

[審議事項]

- 1) 植松康祐先生 (大阪国際大学) の代表理事任期 1 年間の延長が決定いたしました。会沢成彦先生 (大阪府立大学) の監事任期 1 年間の延長が決定いたしました。
- 2) 寺岡義伸先生より前年度末での顧問辞任の申し出がありました。
- 3) 協会では保有している数学雑誌について、リストを作成しています。まだまだ途中ですが、HP に公開します。
- 4) 協会では保有している数学雑誌の内には、かなり貴重なものがあるはずなので、大学等の購入先を見つける活動も続けるべきとの意見が寄せられました。
- 5) 海外の Editor については、E*** のような会員番号を付与し、同時に PW を通知し、最新の論文を閲覧できるようにしたいと思います。数年前までは、紙雑誌を Airmail で送っていましたが、経費カットの点で今では中止していますので、従来と同じ利便性を提供することが目的です。さらに、これらを通知する段階で、他の Editor 推薦も依頼したいと思います。
- 6) 論文を SCMJ 誌に掲載するにあたって、トラブル回避の観点より「著作権譲渡同意書」の導入が話されました。当面、法律専門家が作成した本格的な契約書を導入することまでは行いませんが、論文掲載の際に、著者と「著作権に関する確認書 (覚書)」を Email にて交わす方向ではどうかと審議されました。

* 2018年度決算予算表
(2018年/1/1-18/12/31)

収 入			
科 目	18年度実績	18年度予算	19年度予算
前年度繰越金	-		
刊行物頒布代(書店)	376,200	400,000	400,000
刊行物頒布代(書店)海外\$より	471,750	800,000	800,000
会費			
機関会員 A(旧協力校)			
機関会員 B(交換誌)			
賛助会員(機関会員)	312,000	400,000	320,000
正会員(国内)	683,000	700,000	680,000
ページチャージ・別刷	78,640	50,000	50,000
事務所解約保証金(特別収入項目)			
設備更新積立金			
(イ)減価償却積立金取り崩し分			
(ロ)回転資金取り崩し分			
(ハ)事務機購入積立金取り崩し分			
預金利子	49		
(\$→¥:調整項目)	1,035,394	1,258,000	1,318,000
雑収入			
合 計	2,957,033	3,608,000	3,568,000
支 出			
科 目	18年度実績	18年度予算	19年度予算
通信交通輸送費(イ+ロ+ハ)	128,647	180,000	120,000
(イ)編集通信交通費(含出張費)	34,300	-	-
(ロ)査読通信費			
(ハ)抜刷等輸送費	94,347	180,000	120,000
原稿料	3,000	18,000	18,000
租税公課		-	-
印刷費	340,481	370,000	550,000
組版委託費・書籍整理費	56,220	180,000	70,000
SE委託費(辻本氏)	250,000	300,000	300,000
消耗品代		10,000	10,000
備品代(OA機器soft,本代,rental server)	47,458	50,000	50,000
人件費	1,082,200	1,450,000	1,200,000
借事務所代	813,072	810,000	810,000
電話代	76,323	70,000	70,000
振込料・手数料	9,632	10,000	10,000
電気代	30,832	40,000	40,000
保険料	10,249	10,000	10,000
税金	70,000	70,000	70,000
会費(学術団体)			
雑費	16,919	40,000	20,000
会議費	22,000		20,000
DOI会費			50,000
DOIシステム対応予算(辻本氏)			150,000
合 計	2,957,033	3,608,000	3,568,000

2018年度 貸借対照表
(18/1/1-18/12/31)

(¥)会計

借 方			貸 方		
科目	期 首	期 末	科目	期 首	期 末
固定資産(保証金)			協会活動予備資金		
流動資産	2,869,103	2,147,745	出版基盤強化積立金	500,000	500,000
(定期預金)			TOTAL INDEX 積立金	414,993	414,993
(普通預金)	2,869,103	2,147,745	IT機器積立金		
(現金)			事務所移転積立金		
安全資産ファンド	6,873,301	6,873,301	事務所機購入積立金		
			減価償却積立金		
			回転資金		
			繰越金	8,827,411	8,094,052
			未払費用		12,001
合 計	9,742,404	9,021,046	合 計	9,742,404	9,021,046

外貨会計

借 方			貸 方		
科目	期 首	期 末	科目	期 首	期 末
固定資産			協会活動予備資金	\$37,173.57	\$41,564.14
流動資産			IT機器積立金		
定期預金(★)	\$1,070.67	\$1,072.49	\$-¥準備金		
普通預金(★)	\$36,102.90	\$40,491.65	繰越金		
\$国債2(★)	\$0.00	\$0.00	合 計 \$	\$37,173.57	\$41,564.14
合 計 \$	\$37,173.57	\$41,564.14			
(ユーロ)(★)	€ 2,074.40	€ 2,074.40	(ユーロ)	€ 2,074.40	€ 2,074.40
¥マルチマネー	8,515,936	8,515,982	¥マルチマネー	¥8,515,936	8,515,982
¥普通預金	1,132,361	332,361	¥普通預金	¥1,132,361	332,361

数理科学推進基金会計

借 方			貸 方		
科目	期 首	期 末	科目	期 首	期 末
清水基金	1,000,000	1,000,000	ISMS受賞基金	1,000,000	1,000,000
功力基金	100,000	100,000	国際研究交流基金	1,737,510	1,737,510
石原	2,000,000	2,000,000	通信費	0	0
その他	538,580	538,580	交通費	0	0
			繰越金	901,070	901,070
合 計	3,638,580	3,638,580	合 計	3,638,580	3,638,580

★印は、為替相場変動リスクあり

* 寄稿

Qt ツールとしての TeXworks

藤井 淳一 (大阪教育大学 教育協働学科 理数情報講座)

1. はじめに

最近の Windows L^AT_EX 環境は、TeXworks を使われる方が多いと思います。しかしこれの元になった MacTeX の TeXShop と違って、カスタマイズは付属のファイルに触らないとできず、また少し癖があるので参考までに気づいたことなどを書き留めておきたいと思います。基本的な設定については、学生用に書いた <http://ex.osaka-kyoiku.ac.jp/~fujii/EX/texworksMANU.html> を見ていただくことにして (ただし、「エンコーディング」部分は、今までの Shift-JIS 系のファイルを使わない方は、UTF-8 でお使いください)、あまり書いていないような Qt 関連部分を中心に書いてみたいと思います。マクロや、色付けの参考にしていただけると幸いです。

Qt (キ्यूト) は GUI を含む汎用ツールですが¹⁾、あまり日本では知られておらず大きな書店で探しても数冊しか見つからないレベルで、ましてや CSS について書いてあるものはほとんどありません。もともと Javascript と CSS (スタイルシート) は、実行部分とデザインという役割分担がありましたが、それを大胆にプログラムで取り入れたものに、Visual C# の WPF (cf. [1, 2]) と呼ばれるものがあります。実行部分は従来の C# コーディングで書き、デザイン部分は HTML・CSS 型の XAML でという役割分担 (もちろんそれぞれでも他方の役割はある程度可能) で連携もうまくできていて画期的なシステムでした (しかし、3D の部分が弱く、3D ツールの Unity に押されて今ではすたれてしまい、C# といえば、Unity になってしまいました)。なぜか Visual C++ にはそのフレームワークがないので、代わりに C++ 自身で開発された同様のシステムが Qt です。Javascript 系の実行部分と、CSS 系のデザイン部分で成り立っており、TeXworks はこれをベースに作られて、デザインやマクロに利用されています。今後利用されていくかもしれませんが、なかなか一般人にはハードルが高いのでは、せつかくの平易なシステムが宝の持ち腐れです。例によって、GitHub の技術者集団 (Qt project, Qt company) が作っている途中なので、まだバグも多くてマニュアルも整備されているとはいいがたく、実際いまだに日本語漢字変換の色表示バグは治っていません。しかし、基本部分はしっかり作られていて、将来的には有望なシステムと思われれます。ここでは、TeXworks のカスタマイズ用の部分のみに限って、その利用の具合を見てみたいと思います。

2. TeXworks Qt Scripting [4]

基本プログラム部分は Javascript です。TeXworks との連携の主なものは、以下の TW オブジェクトのプロパティで値のやり取りができます (詳細は [4] 参照) :

```
TW.target   編集集中の TeX ソース
  .selection   その選択部分文字列
  .cursorPosition()   カーソル位置
  .selectRange(start, end)   指定部分の選択
  .insertText(文字列)   「文字列」をカーソル位置挿入
TW.system(コマンド文字列)   シェル実行
TW.script.fileName   編集集中のファイル名フルパス
```

¹⁾ Qt Quick という開発環境があるようで、それに関する本が中心です。

例として、選択部分の大文字小文字切り替えマクロ `toggleUL.js` は以下のようになります。このような実行可能なファイル名で（拡張子 `js` は Javascript ファイル名です）保存したのち、TeXworks のフォルダ以下の `\twdata\scripts` もしくはその中のフォルダのいずれかに置きます：

```
// TeXworksScript
// Title: Toggle upper/lower case
// Shortcut: Ctrl+Shift+U
// Description: Toggle upper/lower case by the top character
// Author: JF
// Version: 0.3
// Date: 2016-02-11
// Script-Type: standalone
// Context: TeXDocument

var txt = TW.target.selection;
var len = txt.length;
var pos = TW.target.selectionStart;
var uc = txt.toUpperCase();
var lc = txt.toLowerCase();
if (txt==uc) { TW.target.insertText(lc);} else{ TW.target.insertText(uc);}
TW.target.selectRange(pos, len);
```

最初のコメント部分も必要らしいです。TeXworks でマクロが使えるように「スクリプト」「TeXworks スクリプト」「スクリプト一覧の更新」をしても Title が書いて無ければ反映しません。また、使われていないキー操作として Shortcut をコメント文として書けば、ショートカットキーとして動作します。最後の文は、動作上不要ですが、同じ部分を引き続き操作可能なように、選択部分を保持しています。

`system` 命令を使った例として、外部ファイルを動作させて、過渡期は頻繁に使う（他人のファイルはコードが違って文字化けすることが多い）であろう Shift-JIS と UTF-8 間の変換を試みましょう。使うプログラムは [nkf32.exe](#) で、このファイルを実行可能な path（TeX インストールフォルダの bin フォルダなど）に置いておきます。私は基本的に過去のファイルをよく見るので、Shift-JIS 設定になっていて、UTF-8 ファイルをもらおうと文字化けします。その際に次のマクロ `CodeConv.js` を実行すれば、ファイルが（Shift-JIS コードになって）、末尾に「-s」が付いたファイル名の別ファイルになって、そのファイルを読み込めば文字化けが解消します：

```
// TeXworksScript
// Title: Convert to SJIS
// Shortcut: Ctrl+ALT+Home
// Description: convert to SJIS file + -s.tex
// Author: JF
// Version: 0.1
// Date: 2019
// Script-Type: standalone
// Context: TeXDocument
```

```
nname=TW.target.fileName.replace('.tex','-s.tex');
TW.system('cmd /c nkf32 < "'+TW.target.fileName + '" > "'+nname+'"', true);
```

その逆の人は、次の Shift-JIS から UTF-8 に変換する CodeREConv.js を使ってください。今度は末尾に「-u」が付加されたファイル名のファイルに変換されます:

```
// TeXworksScript
// Title: Convert to UTF8
// Shortcut: Ctrl+ALT+END
// Description: convert back to UTF8 file + -u.tex
// Author: JF
// Version: 0.1
// Date: 2019
// Script-Type: standalone
// Context: TeXDocument

nname=TW.target.fileName.replace('.tex','-u.tex');
TW.system('cmd /c nkf32 -w8 < "'+TW.target.fileName + '" > "'+nname+'"', true);
```

3. 色のカスタマイズにむけて

ここでは、本格的な色指定の前に Qt に無関係な色指定ファイルの話をしておきます。TeXShop と違って、キーワードの色付けは、TeXworks フォルダ以下の、`\twdata\configuration\syntax-patterns.txt` を書き換えることとなります。基本正規表現で書かれているので、知らない人はおまじないように見えるかもしれませんが、その部分はあまり触る必要がないのでここでは詳しくは述べません。色指定を反映させるには、

「編集」「設定」「エディタ」「構文の色付け」 で、「LaTeX」を選択

する必要があります。後は、上記ファイル内の [LaTeX] の部分に手を入れれば、変化します。例えばよくあるのは、記号や括弧対応を太字に見やすくしたいということですが、元の部分は、

```
# special characters
darkred          N          [ $#^_& ]
```

とある部分の 2 行目です。`$#^_{}&` の文字部分に「darkred」の色付けを指定するようになっています。ここの後半部分に丸括弧も加え、色を「blue」にし、太字にするには、2 行目を

```
blue;B          N          [ $#^_( )& ]
```

などのようにすればいいわけです。次の指定の # LaTeX environments は `\begin{} \end{}` の色指定ですが、太字のほうが良ければ、色名の後に ;B を入れればいいこととなります。見やすい色にすれば、おのずと効率が上がると思います。

ただ私はあまり目がよくないので、Windows や Mac では標準の白のライト背景はつらいです。最近はやりの sublime text, atom や Visual studio code などのエディタは、DOS 時代に戻ったかの如く「ダーク系」になり、目に優しくなりました。TeXShop もダーク系が用意されています。当然 TeXworks もそ

うしたいところですが、満足のいくものにするには結構苦労しましたので、次章で色付けの方法の一つを紹介します。

4. QtCSS による色のカスタマイズ

画面の背景色や、部品の色等を変えるのは今のところ少し苦労しないと無理なようです。後で CSS ファイルの例を挙げますが、仮に user.css という名前のファイルとし、実行ファイル TeXworks.exe と同じところ（私的には C:\w32tex\share\texworks）にあるとします。そこに、以下のバッチファイル（TeXW.bat など、拡張子が bat なら何でもよい）をメモ帳などテキストエディタで作成し、デスクトップのショートカットもこれにしてください（ $\text{T}_{\text{E}}\text{X}$ の仕様上、円記号は \ になっています）：

```
cd c:\w32tex\share\texworks
start TeXworks.exe -stylesheet=./twdata/user.css %1
```

これで準備はできました。tex ソースの関連付けは、このショートカットにしてください。しかし、ダーク系にするには文字の色が通常ダーク系になっているので、前章のファイルの「LaTeX」部分を、明るい色に変える必要があります。私の場合次のように変えました：

```
[LaTeX]
# special characters
yellow;B N [${#^_{}&()}]
# LaTeX environments
lightgreen N \\\(?:begin|end)\s*\{[~]*\}
# LaTeX packages
lightblue N \\\usepackage\s*(?:\[~]*\)\s*\{[~]*\}
# control sequences
pink N \\\(?:[A-Za-z@]+|.)
# comments
gray Y %.*
```

それでいよいよいろんな背景や部品の色付けですが（cf. [3]）、私の例は現在以下のような QtCss ファイル（ファイル名 user.css で、/twdata 内に置く）になっています。詳しいことは資料不足でまだまだ不明ですが、コメント文で大体想像できるようにしました：

```
QWidget { /* メニュー基本設定 */
    background: #222222; alternate-background: #424242; color: #DDEEDD;}
QComboBox { /* ドロップダウン既存選択部品 */
    border: 1px solid gray; padding: 1px 18px 1px 3px;
    border-radius: 4px; color:yellow; background:#000022;}
QComboBox::drop-down { /* ドロップダウン時メニュー */
    subcontrol-origin: padding; subcontrol-position: top right;
    width: 15px; border-left:1px solid #AAAAAA; }
QScrollBar:vertical { background:#121212; width:12px; margin: 10px 0 10px 0;}
QScrollBar::handle:vertical { /* ドラッグ時 */
    background: #BBBBBB; min-height: 20px;}
QScrollBar::add-page:vertical,QScrollBar::sub-page:vertical{background: #333333;
```



```

/* バー背景の下と上. なければドットのみ */ }
QScrollBar:horizontal { background: #121212; height: 10px;
    margin: 0 10px 0 10px; color:cyan;}
QPushButton { /* 設定時の確認ボタン */
    background: #333333; padding: 4px; color:cyan; border:1.5px outset #888888;}
QTabBar::tab{ /* 出力窓・設定窓のタブ */
    color:yellow; background:#442200; width:100px; }
QTabBar::tab:selected{background:#884400;border:1px outset #888888;/* 選択時 */}
QTabWidget::pane{color:yellow;background:#000000;}
QLabel{ /* 部品文 (下部バー項目内、設定用文字) 基本色 */
    color:#FFFF00; background:#333333;}
QLineEdit{background-color:#000044;solid;border:0.5px solid gray;}/* 検索窓文字列入力 */}
QMenuBar{color:#EEFF8F; /* 上部文字メニュー */}
QMenuBar::item{background:#000020;color:#EEEEFF; /* メニュー項目 */}
QMenuBar::item:pressed, QMenuBar::item:selected { /* 項目選択時 */
    background: #663333;}
QMenu::item:selected {background: #666666; /* メニュー選択時 */ }
QTextEdit{color:#DDFFDD;background-color:#000000; /* 本文基本色。最後に */ }

```

これで以下のような見栄えになりました (縮めてあります) :

The screenshot shows the TeXworks editor window titled 'TexWorkQt.tex - TeXworks'. The menu bar includes 'ファイル', '編集', '検索', 'フォーマット', 'タイプセット', 'スクリプト', 'ウインドウ', and 'ヘルプ'. The toolbar contains icons for file operations and editing. The main text area displays a Qt CSS file with the following content:

```

245 [LaTeX]
246 # special characters
247 yellow;B N  [${#^_}!&O]
248 # LaTeX environments
249 lightgreen N  ¥¥(?;begin|end)¥s*¥{[^]}*¥
250 # LaTeX packages
251 lightblue N  ¥¥usepackage¥s*(?¥{[^]}*¥)¥s*¥{[^]}*¥
252 # control sequences
253 pink N  ¥¥(?;[A-Za-z@]+)
254 # comments
255 gray Y  %.*
256 ¥end{verbatim}
257 ¥end[quote]
258 それでいよいよいろいろな背景や部品の色付けですが(cf. ¥cite[H]), 私の例は
    現在以下のような QtCss ファイルになっています。詳しいことは資料不足でまだ
    まだ不明ですが、コメント文で大体想像できるようにしました:
259 ¥color{red!50!black}
260 ¥begin[Verbatim][commandchars=&¥{¥}]
261 QWidget { &col[blue]/* メニュー基本設定 */
262     background: #222222; alternate-background: #424242; color:
    #DDEEDD;}
263 QComboBox { &col[blue]/* ドロップダウン既存選択部品 */
264     border: 1px solid gray; padding: 1px 18px 1px 3px;
265     border-radius: 4px; color:yellow; background:#000022;}
266 QComboBox:drop-down {&col[blue]/* ドロップダウン時メニュー */}
267     subcontrol-origin: padding; subcontrol-position: top right;
268     width: 15px; border-left:1px solid #AAAAAA; }

```

The status bar at the bottom shows 'CRLF', 'Shift_JIS', and '全351行の301行目、8文字目'.

5. おわりに

最後になりましたが、 \LaTeX で画像を使用するとき、基本的には `ebb.exe` など、画像サイズのデータ BB ファイル を作る必要があります、時々 BB ファイルがないと文句が出ることがあります。この処理があまり親切でないようなので、個人的には処理用の実行ファイルを作っていますが、困らせてはいませんでしょうか。そこでこの機会に、Qt マクロで、

「画像ファイル名を選択して実行すれば、対応する BB ファイルを作成するマクロ」

`makeBB.js` を紹介して稿を閉じます²⁾ :

```
// TeXworksScript
// Title: Make BB
// Shortcut: Ctrl+Alt+E
// Description: make BB file by ebb
// Author: JF
// Version: 0.1
// Date: 2019-03-04
// Script-Type: standalone
// Context: TeXDocument

TW.system('cmd /c cd "'
  +TW.target.fileName.substring(0,TW.target.fileName.lastIndexOf('/')')
  +' " & ebb '+TW.target.selection, true);
```

参考文献

- [1] WPF を使った C # プログラミング, <http://ex.osaka-kyoiku.ac.jp/~fujii/WPF/>
- [2] WPF アプリケーション入門講座一覧, <http://ex.osaka-kyoiku.ac.jp/~fujii/WPFhp/>
- [3] HatenaBlog, TeXworks の背景色・文字色を変える
<https://tasusu.hatenablog.com/entry/20111106/1320588877/>
- [4] TeXworks Scripting (QtScript) and related,
<http://twscript.paulanorman.com/docs/TeXWorksQtScripting.pdf>

²⁾ あまり書かれていないので、Qt とは無関係な注意も記しておきます。 \LaTeX 専用エディタでありながら、ウムラウトを指定するダブルコーテーションがおかしいので反応しません。引用符を変換する設定になっているからで、「フォーマット」「引用符の形式」で「なし」を選択してください。また、TAB を使った補完命令も、例えば、`\twdata\completion\tw-latex.txt` に指定されていますが、「複数行揃え」の環境 `multline` を、`multiline` と誤って指定されているため動作しません。全然直す気配がないので、ファイルを書き直す必要があります。GitHub 集団のページですでにそうなっているので、彼らが間違っているようです。なぜか TeXShop では間違っていないです。