2004/6　パソコンリテラシの雑誌に掲載された論文を一部割愛し、Ver8版に修正

カルキングVer8－数式処理と数式を含む文の作成とに優れたツール

1.　はじめに

昨年のソフトウエア特集の一つのテーマに，カルキングを取り上げた。しかし，カルキングは膨大な機能を持っているので，紹介できたのはほんの一部であった。カルキングはその後 Ver8 にバージョンアップされ，機能も更に向上された。バージョンアップ後も，改良が加えられ，ユーザは改良版をいつでもインターネットを通して，無料でしかも簡単にインストールできる。この特集では，二つの点に着目して,解説をしたい。一つは，充実した数式処理の機能について，もう一つは，数式エディタと一体化した作図機能を含む文章作成ソフトの機能についてである。この文章も実は，カルキングで作成した。文章作成ソフトウエアとしてカルキングを見ると，単に使いやすいだけでなく，作成した原稿がそのまま，HTMLの文書にも，AmS_LaTeX（米国数学会で採用しているTeX）にも，直ちにカルキングの中で変換できる点にある。筆者は永らくM社の文章作成ソフトウエアと，その中の数式エディタを使用してきたが，カルキングを使うことで，数式エディタを使う苦労から解放されたようである。使いやすさで付言すると，目に見える範囲で誤りが見付かるように設計されていること，及びM社の文章作成ソフトに比べ，マウスでのクリック回数が半分以下になったと思われることである。多くのC言語のコンパイラでは，全角のスペースと半角のスペースを区別している。これは見た目では区別できない。このことは学生を悩ませることになった。かなりの時間を全角のスペースと格闘することになったのである。コメントに日本文が使えるのが原因であった。カルキングはこの点について親切な対応をしている。カルキングでは，Mapleのように式の終わりに“;”をつけない。このため2行以上にわたって書かれて式が本当は1行の繋がった式か，そうでないかが見た目では分からない。また，配列の要素指定は多くのプログラム言語では，“[]”を用いて表すが本来数学の記号では，$a_{i,j,k}$ のように表したい。一方単に添え字のついた変数も同じ様に表したいことがおこる。カルキングでは，一見して同じでもその式を指定（マウスを使って囲むとその部分が青色の点線で囲まれる）することで目でみて区別できるように工夫されている。以下ページ数の都合で筆者が特に重要だと感じた点に絞って解説することにする。

2．カルキングVer8の数式処理機能

まず出来る主な機能を改めて列挙しよう。

1）代数演算

式の展開，同類項のまとめ，因数分解，Σ及びΠの計算，${}_nP_k$ 及び${}_nC_k$を含む代数計算ができる。和及び積がいずれも一つの式で簡単に書ける。例えば,

$$\sum_{k=1}^{n} k^3=\frac{1}{4}n^2+\frac{1}{2}n^3+\frac{1}{4}n^4$$

2）解析計算

極限値の計算，微分，偏微分，定積分ができる。Ver8では常微分方程式，多元多次元連立方程式も解けるようになった。また極限計算や不定積分も自由に扱えるようになった。

$$\lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x=e \qquad \lim_{x\to 0}\frac{\sin x^3}{x^3}=1$$

$\int x^3 dx = \frac{1}{4}x^4$ $\int x\sin x\, dx = -x\cos x + \sin x$

$(-1)^{-i}=23.140692632779267$ これは，e^{π}で複素数の計算ができることを示している。

3）数値計算

初等関数，ガンマ関数，ヘビーサイド関数，ベータ関数，（1種及び2種の）ベッセル関数の計算が出来る。代数方程式の解を求める，微分方程式の解を求めることなどができる。C言語などで書かれた数値計算ライブラリではせいぜい16桁の精度でしか計算できないから高精度で計算できることは有難い。例えば二つの大きな数の最大公約数を求めてみる。

$a=13^{123}\times 17^{15}$

$b=19^{77}\times 17^{10}$ とすると，

gcd(a,b)=2015993900449

となる，しかもこの計算は，この紙面でこのまま計算できる。

$17^{10}=2015993900449$

だから正しく計算されている。

a=2.96323540768230173532543463281206271979160013247915755174380410629168191451870878651443492762671098208528780543500652463918027515479281839351199060636346021$\times 10^{155}$

b=5.86835647276976041897153506804082114442412999993475139166642564459068796537888745575380907055903846768318081411$\times 10^{110}$

このような大きな数の最大公約数をユークリッドの互除法で求めるには，整数の割り算で余りを求める必要がある。大きな数の割り算の余りを求めるのは，大変である。これが苦も無く求められるのは素晴らしい。カルキングではスクリプトの形を使えば，どんなプログラムでも書ける。しかも，高精度で行える。例えばカルキングでは，代数方程式を入力し，実行－方程式－一元多項式と選択すると解が直ちに得られる。

現在では，係数が実数である必要があるが，次に挙げるように，スクリプトを使ってプログラムを作れば，複素数でも同じに扱える。

sin(0.12)=0.11971220728891936

π=3.14159265358979

e=2.71828182845905

Pi=3.14159265358979323846264338

x={0,0,0}

```
f( x,n)
  wf=1
  ( for k = 1 to n step 1 )
    wf=wf×x+c_k
  return wf
```

方程式の最高次の係数を1とし，配列cに係数を降冪順にいれる。

$x^3+x^2-3x+2=0$

の解をカルキングで求めたものは次のとおり。

x = -2.511547141694531984

x = 0.75577357084726604 + 0.47447677800732685i

x = 0.75577357084726604 - 0.47447677800732685i

c={1,-3,2}

```
ini( n,r)
  ( for k = 1 to n step 1 )
    x_k=r(cos(2Pik/n)+i×sin(2Pik/n))
  return 0
```

```
durand( f,m,n,mm )
  ini(n,mm)
  ( for k = 1 to m step 1 )
    ( for s = 1 to n step 1 )
      wduran=1
      ( for t= 1 to n step 1 )
        {wduran=wduran×(x_s-x_t)   s≠t
        {wduran=wduran             s=t
      x_s=x_s-f(x_s,n)/wduran
      u=(n+1)(k-m+1)+s
      {Sheet1_1,u=x_s   k>m-2
      {u=u              k<m-1
  return 0
```

パラメータは左から方程式の左辺の値を計算する関数名，繰り返し回数，方程式の次数，解を含む円の半径の値（予測値，係数の絶対値の最大値＋1）

durand(f,10,3,4)=0

0.75577357084198238 + 0.47447677802422744i
-2.5115471416945248 + 6.9988624072047344×10^-15^i
0.75577357084726593 - 0.47447677800732702i
0.75577357084726604 + 0.47447677800732674i
-2.5115471416945319
0.75577357084726604 - 0.47447677800732685i

複素数を係数とする方程式の場合の例

$x^3 + (1+i)x^2 + (-3+5i)x + 2 - 3i = 0$

c={1+i,-3+5i,2-3i}

```
durand1( f,m,n,mm )
ini(n,mm)
( for k = 1 to m step 1 )
 ( for s = 1 to n step 1 )
  wduran=1
   ( for t= 1 to n step 1 )
   {wduran=wduran×(xs-xt)   s≠t
   {wduran=wduran           s=t
  xs=xs-f(xs,n)/wduran
  u=(n+1)(k-m+1)+s
  {Sheet2_{1,u}=xs   k>m-2
  {u=u               k<m-1
return 0
```

durand1(f,15,3,6)=0

出力をSheet2にするためだけの変更をした。もし変更しないで実行すると，上の表に重ね書

y={{0.563107578717864015391015132566247842787392397888108407947278＋ 0.1247831277460520454185464520387783043729341707816270616436 71i},{-2.7970540543068116894312213706791691869165257269076005610745 2+ 0.665428875226452026667446538311570436407968195740207084969947i},{1.233946475588947674040206238112921344129133329019492153127 24 -1.790212002972504072085992990350348740780902366521834146613 62i}}

きされる。 このように，係数は複素数でも同じに解ける。ほんの少しプログラムの修正と追加をし，実数の対で複素数を表すと，1000桁の精度で計算するプログラムに変えることが出来る。計算時間も僅かである。ここで結果だけ載せる。但し見やすいように複素数の形にし精度は60桁で計算したもの。1000桁でも構わない。

-2.7970540543068116 + 0.66542887522645189i
0.56310757871786399 + 0.12478312774605201i
1.2339464755889475 - 1.7902120029725042i
-2.7970540543068116 + 0.665428875226452i
0.56310757871786399 + 0.12478312774605201i
1.2339464755889478 - 1.7902120029725042i

カルキングVer8では，プログラムを別のファイルにし，パラメータだけを入力すれば計算する機能が追加された。これは，自分で作ったプログラムをライブラリ化し，それを組み込み関数と同様に使えるようにする機能である。

4）線形計算

行列の演算，連立一次方程式の求解，行列式の計算ができる。固有値及び固有ベクトルの計算が出来る。

5）グラフ作成

2次元，3次元のグラフが描ける。グラフの拡大，縮小が出来る。

3次元グラフはマウスを使い，図形の立体的な回転も出来る。中学高校の数学教科書に出てくる図形は殆ど出来る。更に，化学式の構造式を簡単に表示できるプログラムも用意されている。葡萄糖の構造式，ニトロベンゼンの構造式を計算機で図示する作業は大変である。それが吃驚するほど簡単にできることを後に示す。

3.1 カルキングによる入力

3.1.1 カルキングによる数式を含む入力法

数式を含む文を作成するとき，使いたい記号

に積分記号，積和の記号Σ，Π，数式記号(±≠≦≧≡∉¬∀∃∋など)，α，β，γなどのギリシャ文字,*a,b,c*などの数式に使う文字などがある。筆者はM社の文章作成ソフト及びそれに内蔵されている数式エディタを今まで使用してきたが，幾つかの難点があった。その中で一番困ったのは，1ページの数式を入力するとハードディスクに格納できなくなることである。“メモリが足りないので使用中の他のプログラムを終了するようにしてください”と表示されて解決策が無くなる。止むをえず数行入力する毎に格納し，入力の仕事を終了し，再度プロプログラムを立ち上げて入力することにした。メモリは主メモリ1Gバイト，ハードディスク80Gバイトあるので，メッセージの意味は良く分からない。

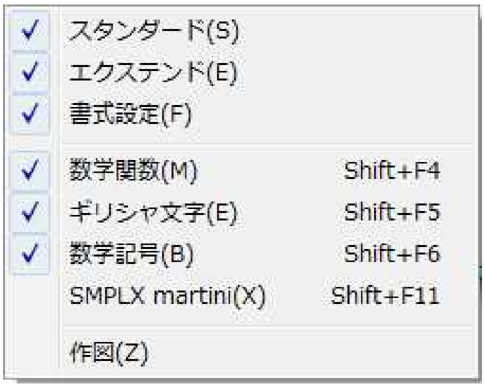


図3.1　表示－ツールバー

図3.4　数学関数及び数学記号

そこで今回は，文章も式も全てカルキングで作成することにする。カルキンクの画面には，積分記号などの記号が表示されている。表示－ツールバー－数学記号を選ぶと，数学記号の一覧表が画面に表示される。いづれの記号もそこをクリックすれば，指定した場所(ポインタの指しているところ)に入力される。これはM社のものに比べ手間が格段に少なく早く入力できて有難い。一々数式入力と日本文入力を切り替えなくて済むこと，式の中にスペースを入力することも，文を上下左右に微小に動かすことも簡単である。数学関数の記号は，定積分，不定積分，和，積，極限，順列，組み合わせ，2項係数，ベータ関数，ガンマ関数，転置行列・逆行列演算，ベッセル関数，ガウス記号，複素数演算記号，ルジャンドル関数等を表す。

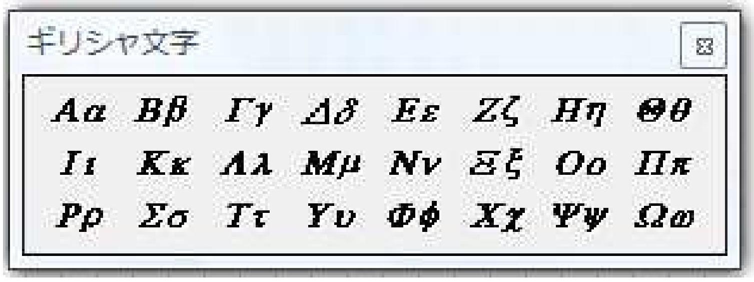


図3.2　ギリシャ文字

これらの記号は，式の中に書け演算もそのままできるが，それらの式を含む文としても書ける。数学記号も同じである。Unicodeでは，数式の中に書いて，演算子として実行できるものとできないものがある。実行出来ないものは，単に演算子の記号として文中に使えるものである。その区別は色分けでされている。また，筆記体のアルファベットの書体はなかなか気に入ったものが無く困っていたが，シンプレックスではこのフォントを作ってしまわれた。*l* の字はlでは分かり難く大変であったが解決した。作られたフォントは次のとおりである。

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

日本文と式が混じった文章を入力しようとすると，フォントの切り替えが煩わしい。ところがカルキングは素晴らしい方法で解決している。最初それに気づかなかったが，マニュアルに例をあげて説明してあった。文中の変えたいフォントを，マウスの右ボタンで指定してクリックすると図3.3のように表示されるので，“書式の引用”を選べばフォントの変更が出来る。では，実際に数式を含む文をカルキングで入力してみる。

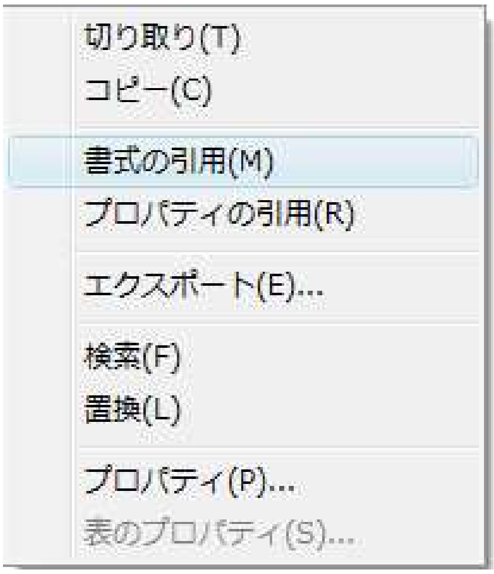


図3.3　書式の引用

次のπは無理数であることの証明はI,Nivenによって与えられたものである。

πは無理数である

補題1. $f(x)=\dfrac{x^n(q-px)^n}{n!}$　ここで，n,p,qは，整数とおく。

$F(x)=\sum_{j=0}^{n}(-1)^j f^{(2j)}(x)$とおくと，$F(0)$ 及び $F\left(\dfrac{q}{p}\right)$はともに整数となる。

補題2. $\pi=\dfrac{q}{p}$　とおくと，

$0<F(0)+F(\pi)=\int_0^{\pi} f(x)\sin x\, dx<$

$\int_0^{\pi}\dfrac{q^n x^n}{n!}dx<\dfrac{(nq)^{n+1}}{(n+1)!}$

補題3. 式2で$n\to\infty$とすると矛盾が生まれる。

$f^{(k)}(0)=\left\{((-1)k-n\ {}_nC_{k-n}q^{2n-k}p^{k-n})k!\right\}/n!$

補題1の証明

$f(x)$はxの$2n$次の多項式で$n-1$次以下の係数は0となる。従って，

$f^{(k)}(k)(0)=0$ が$0\le k<n$に対して成り立つ。

また $n\le k\le 2n$　に対して

$f^{(k)}(0)=\left\{((-1)k-n\ {}_nC_{k-n}q^{2n-k}p^{k-n})k!\right\}/n!$

となる。

となるので$f^{(k)}(0)$の値は整数となる。

$f\left(\dfrac{q}{p}-x\right)=f(x)$となるので，$x=\dfrac{q}{p}$に対して

$f^{(k)}\left(\dfrac{q}{p}\right)=f^{(k)}(0)\times(-1)^k$が成り立つ。

従って，

$f^{(k)}\left(\dfrac{q}{p}\right)=0$が$0\le k<n$に対して成り立つ。

また$n\le k\le 2n$に対して，

$f^{(k)}\left(\dfrac{q}{p}\right)=\{$(整数)$\times k!)/n!$となるので

$f^{(k)}\left(\dfrac{q}{p}\right)$の値は整数となる。従って，

$F(0)$ 及び$F\left(\dfrac{q}{p}\right)$はともに整数となる。

補題2の証明

$\pi=\dfrac{q}{p}$ とおく。ここで q,p は共に自然数とする。$f(x)$ は x の$2n$次の多項式だから，

$f^{(2n+2)}(x)\equiv 0$となるので，

$F''(x)+F(x)=f(x)+(-1)^n f^{(2n+2)}(x)=f(x)$

$(F'(x)\sin x-F(x)\cos x)'=F''(x)\sin x+F(x)\sin x=f(x)\sin x$

$0<x<\pi$ で$\sin x\times f(x)>0$ となるので，

$0<\int_0^{\pi} f(x)\sin x\,dx=[F'(x)\sin x-F(x)\cos x]_0^{\pi}=F(0)+F(\pi)$

$\int_0^a f(x)dx\mathrm{c}=\sin x+\cos 2x$

また

$0<x<\pi$ で$0<\sin x<1,(q-px)^n<q^n$となるので,

$f(x)\sin x\leq\frac{q^n x^n}{n!}$となる。

従って, $\int_0^\pi f(x)\sin x\,dx<\int_0^\pi \frac{q^n x^n}{n!}dx=$

$\frac{q^n \pi^{n+1}}{(n+1)!}<\frac{(\pi q)^{n+1}}{(n+1)!}$ が得られる。

補題3の証明

式2で, $\frac{(\pi q)^{n+1}}{(n+1)!}$は, nを十分大きくとれば

1より小さく出来る。従って,

$0<F(0)+F(\pi)<1$とできる。ところが,

$F(0)+F(\frac{q}{p})$は整数であった。これは矛盾である。
QED

式のフォントをSMPLXフォントにし, 日本文は例えば, MSP明朝フォントにしたいとき, とりあえず直接入力で式を入力する。例えば次のように入力する。ついで, この式全体をマウスを使って指定する（式全体が青色の点線で囲まれる）。次に, フォントを示している欄をクリックし, SMPLX MARTINI を選ぶと次のように文字全体がSMPLXフォントの文字に変わる。

$$\int_0^a f(x)dx=\sin x+\cos 2x$$

M社の文章作成ソフトより随分早く作成できた。何より途中でメモリに格納できなくなって苦労して作成した文章が無駄になることが無かった。カルキングの凄いことはこれに留まらなかった。文の途中で1文字だけ, aのように字体を変えて入力する(Ctr-Shift-mを押して入力）ことや, 文字列を数式で置き換えることが出来る。

そして筆者は, カルキングの優れたところは, 単に数式処理, 2次元及び3次元の図形表示機能をもつだけでなく, カルキングで作成した文書全体をHTMLの文書に変換できること, 更にAmS_LaTeXの書類に変換できることにあると考える。米国数学会を始めとする外国の学会, 国内では電子情報通信学会など多くの学会で, 投稿論文を所定のTexで書くように定めているところがある。従って, これからの文章作成ソフトウエアは, Texに自動変換できることが必須であろう。作成したこの原稿全体をTexで作成した原稿に変換するには, ファイルーエクスポートを選ぶ。ついで, エクスポートの形式をTex形式にし, エクスポート先（変換後のデータの格納場所）をクリックすると変換を完了する。

図3.4 エクスポート

図3.5 Tex形式を選ぶ

この原稿全体を載せるにはページ数が無駄になるので，次の短い例についてのせる。HTMLの書類に変換するには，エクスポートの形式をHTMLの形式に選べばよい。

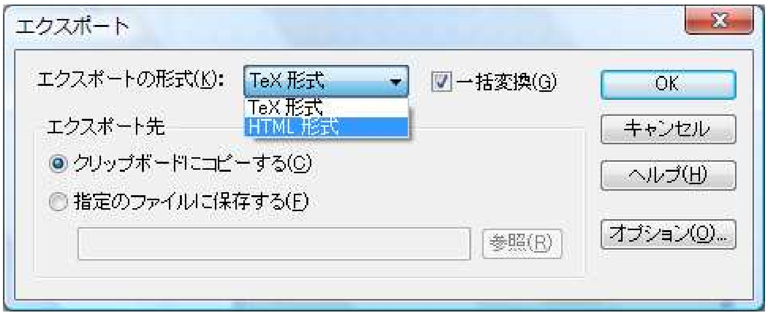


図3.6　Html 形式を選ぶ

変換例文

πは無理数である

補題1．$f(x) = \dfrac{x^n(q\text{-}px)^n}{n!}$　ここで，n,p,qは，整数とおく。

$F(x)=\sum_{j=0}^{n}(-1)^j f^{(2j)}(x)$とおくと，$F(0)$ 及び$F(\dfrac{q}{p})$はともに整数となる。

補題２．$\pi=\dfrac{q}{p}$とおくと,

$$0<F(0)+F(\pi)=\int_0^{\pi} f(x)\sin x\,dx<\int_0^{\pi}\frac{q^n x^n}{n!}dx<\frac{(nq)^{n+1}}{(n+1)!}$$

となる。

補題３．式２で$n\rightarrow\infty$とすると矛盾が生まれる。

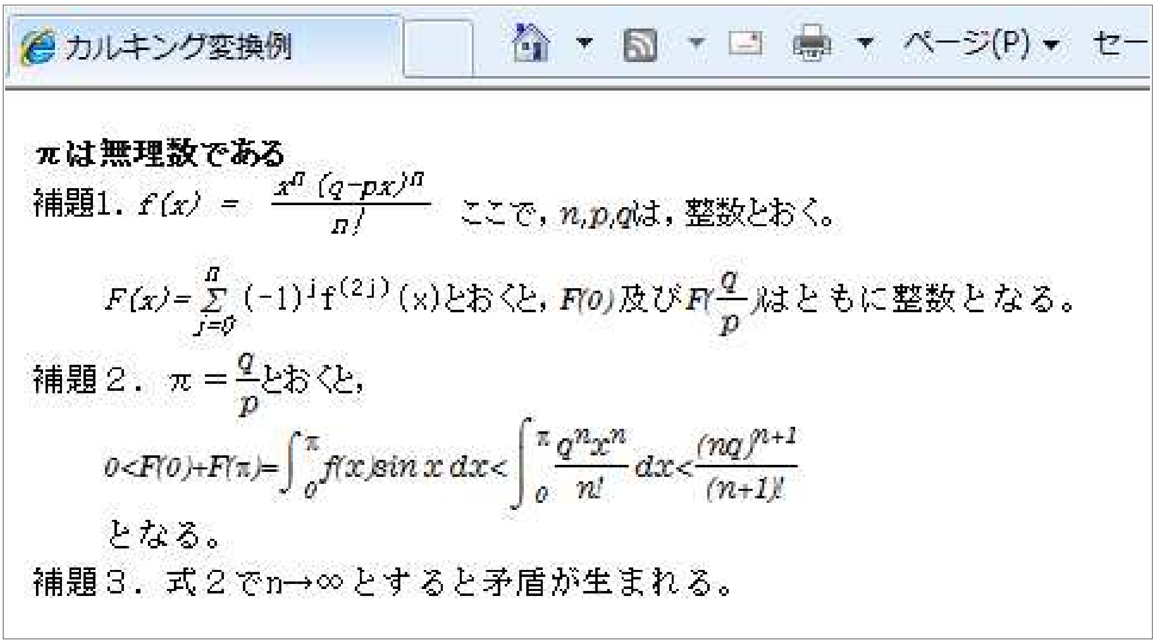


図３．７　Html文を戻したもの

次にTexに変換したものを載せる。

```
{$¥boldsymbol{π¥/}$¥textbf{は無理数である¥/},¥newline ,ここで$, ¥mathit{¥mathit{n¥/}¥mathit{¥/}¥/}¥mathit{p¥/}¥mathit{¥/}¥mathit{q¥/}$は$, $整数とおく。,¥newline ,補題$1. ¥mathit{f(x) =  ¥/}¥frac{¥mathit{x¥/}^{¥mathit{n¥/}}¥mathit{(q-px)¥/}^{¥mathit{n¥/}}}{¥mathit{n!¥/}}$,¥newline ,$¥mathit{F(x)=¥/}¥sum_{¥mathit{j=0¥/}}^{¥mathit{n¥/}} (-1)^{j}f^{(2j)}(x)$とおくと$, ¥mathit{¥/}¥mathit{F(0) ¥/}$及び$¥mathit{F(¥/}¥frac{¥mathit{q¥/}}{¥mathit{p¥/}}¥mathit{)¥/}$はともに整数となる。$,¥newline ,補題$2. ¥pi =¥frac{q}{p}$とおくと$, $,¥newline ,$0<F(0)+F(¥pi )=¥int_{0}^{¥pi } f(x)¥sin x¥dx<¥int_{0}^{¥pi } ¥frac{q^{n}x^{n}}{n!}¥dx<¥frac{(nq)^{n+1}}{(n+1)!}  $となる。$,¥newline ,補題$3. $式$2$で$n$$¥rightarrow ¥infty $とすると矛盾が生まれる。}
```


図3.8変換表の枠組み

“編集－置換”を選ぶと，黄色の5行2列の枠が表示される。左欄に変換したい文字列を，右欄にそれを置き換えたい文字列を記入する。

x	$sin\ x$
u	$log\ x$
v	$\int_0^t f(x)dx+g(t)$

図3.9　変換表

図3.10　変換表の作成

次にその例を示す。

$$y=x^2+2x+3$$

$$z=5u^2+4u$$

$$w=\int_0^v f(x)dx+v$$

$$\downarrow$$

$$y=sin\ x^2+2sin\ x+3$$

$$z=5log\ x^2+4log\ x$$

$$w=\int_0^{\int_0^t f(x)dx+g(t)} f(x)dx+\int_0^t f(x)dx+g(t)$$

カルキングには編集の際の嬉しい機能の一つに，上下左右に対し文字および文章を微小な単位でずらせるものがある。行間が上に詰まり過ぎと思うときは，下に気の済むまで下げられるし，式の間に微小なスペースを入れたいときなどに便利である。この機能は単に“ある”というのでなく，使いやすいのである。右辺の第1項の積分の積分範囲を少し上に移してみよう。ここでは，目に見えるよう大きく動かしてみるが，実際には目に見えないような単位で動かせるようになっている。まずその動かしたい部分をマウスを使って，指定するとその部分が青色の点線で囲まれる。ついで矢印キー（通常キーボードの右についている）を押すと移動する。

$$w=\int_0^{\boxed{\int_0^t f(x)dx+g(t)}} f(x)dx+\int_0^t f(x)dx+g(t)$$

図3.11　微調節

$$w=\int_0^{\int_0^t f(x)dx+g(t)} \quad f(x)dx+\int_0^t f(x)dx+g(t)$$

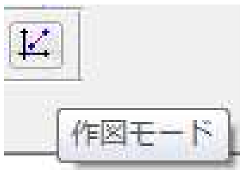
3.2　図形の作成

中学高校などの数学の問題を作成する際，三角形及び内接，外接する円を書きたいことがよくある。手書きのときは，定規とコンパスで描くが，計算機上で作るときは，案外面倒なものである。まして，綺麗な図形にしようとすると大変である。カルキングでは，解析幾何による式を，図形にすることも出来るが，作画機能だけで作成することもできる。ここでの方法はサポートセンターに教わった誰にもできるものである。話題は三角形の作図に関するものである。

3.2.1　三辺の長さを与え，三角形及び外接円を描く。

三辺の長さ7,8，9の三角形を描き，ついで外接円を描く。

図3.12　作図モード



オプション－作図モードを選ぶか，左図のアイコンをクリックすると作図モードに変わる。カルキングでは、殆どの機能が一つのボタンで選べるように工夫されている。

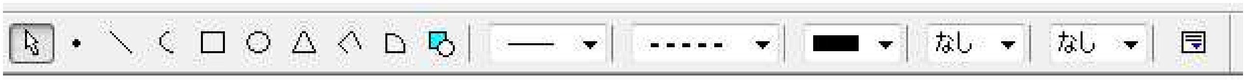
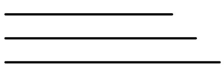
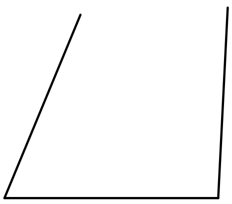


図3.13　作図モードでのアイコン



与えられた長さの三辺



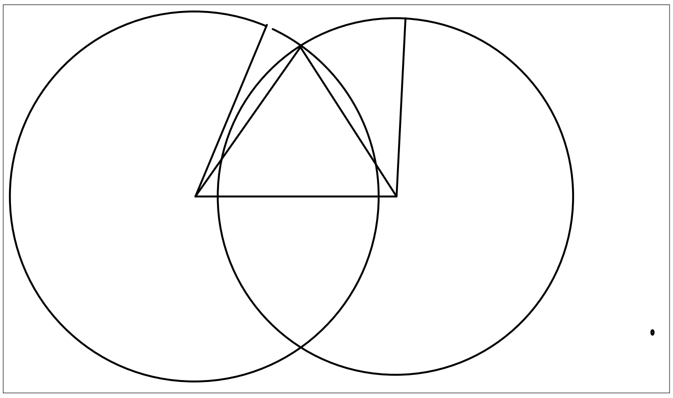


図3.17　決められた長さの円二つ

図3．14　作図オプションの選択

左から2番目'線を描くツール'の斜線のアイコンを選び，グリッド（格子点）7個，8個，9個分，三つの直線を描く。次いで斜線のアイコンを選び，グリッド（格子点）7個，8個，9個分，3つの直線を描く。次にコピーし，回転させ移動する。円弧を描くアイコンをクリックし，円の中心を指定して与えられた2つの円を描く。円の交点CとA，及びBを結ぶ。ついで円弧全てを消去する。辺がばらばらにならないように，3辺を指定して，グループ化のモードを選び固定する。円弧及び作業用の辺を消す。円弧ツールを指定し，作図オプションメニューを選び"円弧の種類-円弧の両端指定"を選ぶ。

底辺ABを通り，頂点Cを通る円弧を，円弧の両端指定モードで描く。

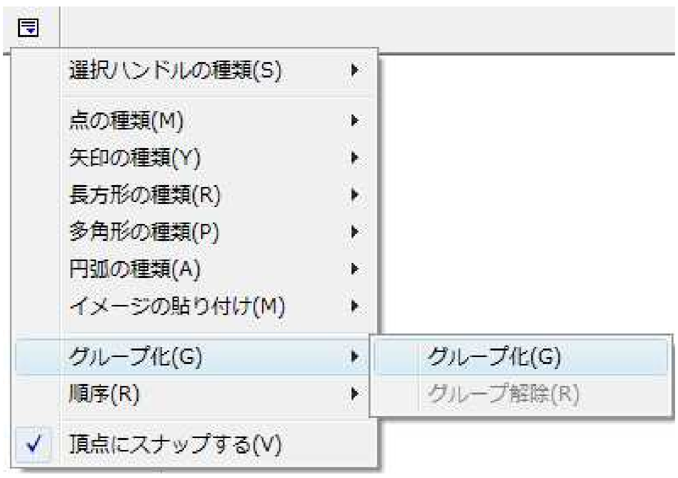
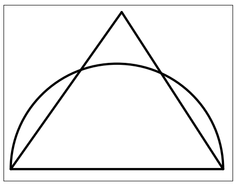


図3．18　図のグループ化

1) 

円弧を広げ，三角形の頂点を通るようにする。

2) 頂点を通るように調節

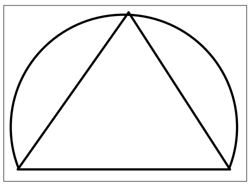
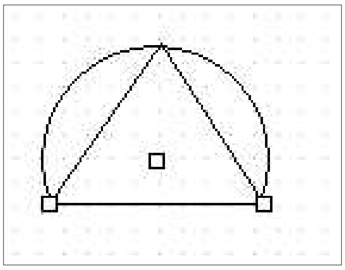


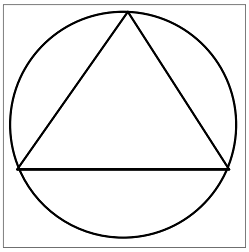
図3.19　円弧の種類の選択

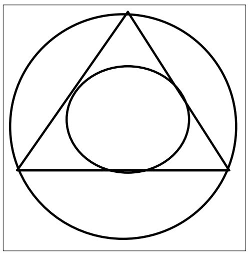
3) 円の中心が格子点(グリッド)にくるように微調節

円弧の中心がグリットの位置に来るように矢印キーを使って微調節する。円弧を消し，中心の点を元に頂点を通る円を画く。



4) 求めた中心をもとに円を画く。





三角形の二つの角について，角の2等分線を引き，その交点を求めるとその点がが内接円の中心である。コンパスを使う要領で画けばよい。

4．おわりに

2003年6月のソフトウエア特集で，紹介できなかったこと及びカルキングVer8で新たに追加された機能のうち，筆者が便利だと思うものの内の幾つかを解説した。項目の選び方は，筆者の独断によるものである。使ってみて数式処理機能の基本機能についてはMathematica及びMapleの機能とくらべ遜色がない。何よりカルキングの優れているところは，カルキングの中で，文書及び数式の混ざった文がそのまま書けることである。文字の前後左右の添え字が自由に書け，前後左右のスペースを微調節できるのは素晴らしい。その他、数式、作図部品を登録しておき任意の箇所で貼り付ける機能がある。定型的な数式、化学構造式の作成等に使用される。 プログラム言語の機能も素晴らしく，構造化プログラミングが意識せずにでき，構造的に見やすく，分かりやすいプログラムが自然に出来上がる。その上，複素数の計算を含む多重精度の計算ができる。C言語で精度がでなくて苦労した人には朗報である。現在は1000桁までの計算であるが，10万桁の精度で計算できるようにするのに，プログラムの変更は些細なことであり，パソコンの演算速度，主記憶装置の容量も増えてきているので，次のバージョンでは拡張されるかも知れない。なお本原稿はカルキングで記述され、これをZipで圧縮されたものはわずか155KBであった。なお，この紹介を書くにあたり、サポートセンターには一方ならず使い方について教えて頂いた。分かってみれば、マニュアルに殆ど親切に書かれていたが、マニュアルを読まないで使ったのが原因である。お礼申しあげたい。

参照文献

カルキングVer8　ユーザーズガイド:株式会社シンプレックス，2007.8