今までの数学ソフトとは違うカルキングの「ライブ数式スタイル」とは

Live　Math　Interface　Style　（ライブ数式インターフェイススタイル）

国産のユニークな数式計算ソフト「カルキング」の特長

●数式計算/ドキュメント作成ソフト(Windows8/7/Vista/XP対応.32/64bit)
カルキングは、パソコン画面上のどこにでも手で書くように、数学の記法通りに数式や文書・作図・表・２Ｄ／３Ｄ関数グラフを簡単に表記・計算できる,ワープロ感覚の統合数式計算ソフトです。数式と文書が混在して使える計算ソフトです。
加えて画面の表記通りに印刷物として美しく出力できるＷＹＳＩＷＹＧ機能を,具備する数々の特長を誇っております。HTML/Tex変換で保存でき/PDF形式変換等も可能な便利なツールです。（教育学術関係/技術研究関係/その他）.

★カルキングの「ライブ数式スタイル」の入力の定義

(1)基本的ワープロ機能を具備

作図機能や、表機能、アプリケーション・オブジェクトの相互の貼り付け等を含む

(2)自然数式の直接編集が可能。（ユーザに内部表現を見せない）

自然数式の意味「ライブ数式スタイル -基本用語」を参照
PC画面上のどこにでも、数学の記法通り（手入力のように）に数式が入力でき 計算できる

(3)任意の位置に作成した数式をプロパティに基づき、数値計算や記号計算が可能

プロパティの意味「ライブ数式スタイル -基本用語」を参照（P.7）

(4)計算結果の計算式も自由な編集、コピーが可能。

(5)式の一部分のコピーも可能。

(6)計算式の任意の場所に、コピーされた数式を埋め込むことが可能。

(7)ページ境界での折り返しだけでなく右隣への伸長も可能。

★ライブ数式スタイルではない例

ほとんどの数学ソフトは、コマンド形式というユーザーインターフェースになっています。
以下では、世界的に有名なフリーのR言語の説明画面です。
基本の形は　(1) 行の初めにプロンプト記号（>）が表示されています。
(2) プロンプト記号のあとに計算した式を書きます。
(3) 答えは次の行の初めに表示されます。

R言語の説明画面　　（以下画面写真はすべてR言語）

カルキングでは

hako=10

メニューから代入操作

hako=10

メニューから計算操作

hako×2=20

メニューから計算操作

R言語の説明画面

カルキングでは

array={10,20,30}

メニューから代入操作

array={10, 20, 30}

メニューから計算操作

array×2={20, 40, 60}

メニューから計算操作

R言語の説明画面

カルキングでは

$\text{array2}=\begin{pmatrix}10 & 20 & 30\\ 20 & 40 & 50\end{pmatrix}$

メニューから代入操作

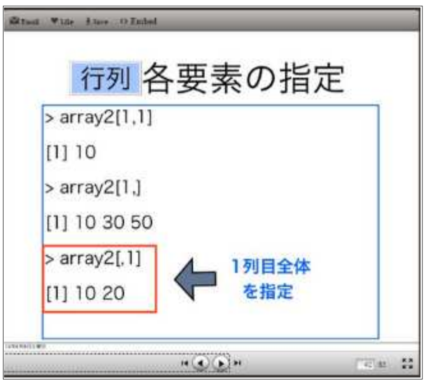
$\text{array2}=\begin{pmatrix}10 & 20 & 30\\ 20 & 40 & 50\end{pmatrix}$

メニューから計算操作

$\text{array2}_{1,1}=10$

メニューから計算操作

R言語の説明画面



カルキングでは

$\mathrm{array2}_1=(10\quad 20\quad 30)$ $\mathrm{array2}_{*,1}=\begin{pmatrix}10\\20\end{pmatrix}$

メニューから計算操作 メニューから計算操作

以上の簡単な例からも分かりますが、コマンド形式数学ソフトとは、計算に特化した計算専用ソフトと考えることができます。さらに数式の表現方法は判読可能ですが、自然数式ではありません。この資料はカルキングで作成していますが
コマンド形式の数学ソフトではとても作れません。

(1)自然数式での計算操作の比較

ライブ数式スタイル

{a,b,c}={5,9,3} a,b,cの値設定

$$\frac{-b+\sqrt{b^2-4ac}}{2a}=-0.4417424305$$

コマンド形式 多くの数学ソフトの形式

```
>a <- 5                      aの値設定
>b <- 9                      bの値設定
>c <- 3                      cの値設定
>(-b+sqrt(b*b-4*a*c))/(2*a)  計算したい式
-0.4417424305                答は次の行に表示される
```

(2) ライブ数式スタイルの典型的作業例

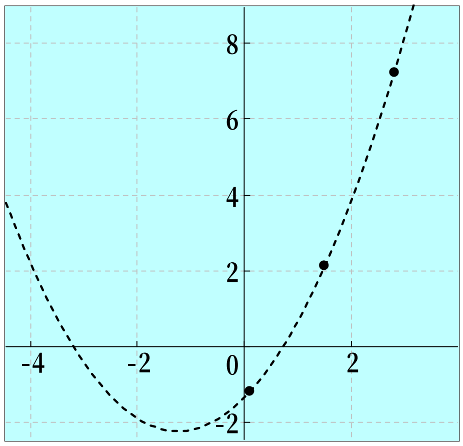
実際の仕事では、計算ばかりを行うのではなく、説明の文書を書いたり、関数グラフを表示したり必要な図を描いたりしながら、一連の作業を進めていきます。この一連の作業の連携を如何にスムーズに進めることができるかどうかが、良き数学ソフトの判断の目安になるはずです。
作業中に難しくて時間がかかるのは、解法や分かり易い説明方法を見つけることです。
多くの場合この段階は紙と鉛筆に頼ることが多いと思われます。ビジネスグラフを眺めたり関数グラフを眺めたりすることは、EXCELや数学ソフトを利用して仕事を進めることができます。
しかしいろいろな計算式を眺めながら、アイデアを考えるような時には、今までのビジネスソフトに関しては限界があり、最後は紙と鉛筆に頼ることになります。
ライブ数式スタイルはこの作業にブレークスルーをもたらそうとするものです。
まだまだ完成形にはなっていませんが、進むべき方向は既存の数学ソフトよりはるかに便利になるものと確信しています。

以下に、簡単な例を使用して、ライブ数式スタイルによる作業の進め方を示します。

背景色が黄色の長方形で囲まれた部分は、カルキングで演算したものです。

例　3点 (0.1, -1.2)　(1.5, 2.1)　(2.8,7.2)　**を通る2次式の関数を求める**

3点を通る点線のカーブの式を求める



以下のようにy(x)を関数定義する。

$y(x)=a+bx+cx^2$　　　　(1)

x={0.1, 1.5, 2.8}　**の時の記号計算を行う。**

記号計算

$y(0.1)=a+0.1b+0.01c$
$y(1.5)=a+1.5b+2.25c$
$y(2.8)=a+2.8b+7.84c$

それぞれの右辺をコピーして
連立方程式の左辺に貼り付け

a+0.1b+0.01c= -1.2
a+1.5b+2.25c= 2.1
a+2.8b+7.84c= 7.2

手動操作で作られた方程式

この3つの式を選択して、「実行」メニュー「方程式関連」ー「連立多項式」で以下のように解が求まる。

a = -1.348718
b = 1.429182
c = 0.579976

方程式の解の表示式を利用して代数代入する。

$a+bx+cx^2=-1.348718+1.429182x+0.579976x^2$

代数計算

$y(x)=-1.348718+1.429182x+0.579976x^2$

最終ゴールが得られました。

解法の改善 (2次多項式だけでなく、さらに高次の多項式も求めたい)

連立方程式を行列の形で表現すると改善の方法が見えてきます。

連立方程式

a+0.1b+0.01c= -1.2
a+1.5b+2.25c= 2.1
a+2.8b+7.84c= 7.2

行列形式の連立方程式

$$\begin{pmatrix} 1 & 0.1 & 0.01 \\ 1 & 1.5 & 2.25 \\ 1 & 2.8 & 7.84 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} -1.2 \\ 2.1 \\ 7.2 \end{pmatrix}$$

この方程式は3点の座標(0.1, -1.2)　(1.5, 2.1)　(2.8,7.2)のx座標値、y座標値をまとめたものから構成されている。これらは配列で表現するとは以下のようになる。

$x=\{0.1, 1.5, 2.8\}$ 代入

$y=\{-1.2, 2.1, 7.2\}$ 代入

行列の作成法

配列xの0乗、1乗、2乗を計算で求める。

$x^0=\{1, 1, 1\}$

$x^1=\{0.1, 1.5, 2.8\}$

$x^2=\{0.01, 2.25, 7.84\}$

これらはカルキングのM演算子を利用して計算すれば次の一つの式にまとまる。

$$M\{x^0,x^1,x^2\}=\begin{pmatrix} 1 & 1 & 1 \\ 0.1 & 1.5 & 2.8 \\ 0.01 & 2.25 & 7.84 \end{pmatrix}$$

求まった行列の転置行列を計算する。

$$\begin{pmatrix} 1 & 1 & 1 \\ 0.1 & 1.5 & 2.8 \\ 0.01 & 2.25 & 7.84 \end{pmatrix}^T = \begin{pmatrix} 1 & 0.1 & 0.01 \\ 1 & 1.5 & 2.25 \\ 1 & 2.8 & 7.84 \end{pmatrix}$$

上記の二つの行列は、以下の1つの式で求まる。

$$(M\{x^0,x^1,x^2\})^T = \begin{pmatrix} 1 & 0.1 & 0.01 \\ 1 & 1.5 & 2.25 \\ 1 & 2.8 & 7.84 \end{pmatrix}$$

このような簡潔な数式計算はカルキングの真髄を示している。

この方法の利点はn点を通るn+1次の多項式を求める解法への一般化が容易にできることである。

$$(M\{x^0,x^1,x^2,x^3,...\})^T =$$

スクリプトにまとめる

上記の計算スタイルは、カルキングを利用したときの典型的な作業形態です。
以上の段階で作業は終了ですが、通常このような作業を自動化するためのプログラム機能があります。カルキングにもスクリプトというプログラミング機能があり、今までの手続きを一般化してプログラムにすることができます。
この作成の詳細は割愛しますが、プログラムを以下に示します。1つ強調しておきたいことがあります。カルキングのプログラムは一般的に知られているプログラミング言語よりもはるかに簡潔に記述できることです。

関数名がget_polynomialの関数は以下のようになります。

get_polynomial(x,y)
var m,n,v,ar,ans,ans_c,t
$n=\|x\|$
return "input data error1" $n\neq\|y\|$
$ar=\mathop{\mathfrak{R}}\limits_{k=0}^{n-1} x^k$
$m=(M\ ar)^T$
return "no answer" $\det(m)=0$
$v=M\mathop{\mathfrak{R}}\limits_{k=1}^{n}\{y_k\}$
$ans=m^{-1}v$
$ans\_c=\mathop{\mathfrak{R}}\limits_{k=1}^{n}\ll ans_k\gg$
return |make_series("x", n-1 ,ans_c)|

一般的プログラミング言語では記述
できない自然数式が多用されている。
これにより簡潔な表現が可能となっている。
変数宣言も最も抽象化された形です。

make_series関数は係数を与えて多項式を
生成するシステム関数です。ここではn-1次の
xに関する多項式を生成します。

get_polynomial関数の利用法

x={0.1, 1.5, 2.8} 代入定義

y={-1.2, 2.1, 7.2} 代入定義

get_polynomial(x,y)=$-1.34872+1.42918x+0.579976x^2$ 計算

以上のようにget_polynomial関数の出力結果は自然数式になります！