

## スクリプト言語仕様

## 目次

1 概要.....	3
2 スクリプト言語の文法.....	3
2.1 言語の概略構造.....	3
2.2 関数型スクリプトの文法.....	3
2.3 マクロ型スクリプトの文法.....	4
3 ユーザが関数やマクロを作成するときに使用できる関数.....	4
3.1 イベント駆動関数.....	4
3.2 セル操作関数.....	4
3.3 関数ライブラリ(SSSLIB).....	5
3.4 スタック操作関数.....	5
4 マクロ.....	5
5 シート関数.....	5
6 言語構成要素.....	17
7 維持管理方法.....	18
7.1 VBA コードファイル.....	18
7.2 VBA 開発画面.....	18
7.3 新しいエクセルファイルでスクリプトを利用できるようにするには.....	18
7.4 新しいシートを追加したら.....	18
7.5 関数ライブラリ(SSSLIB)の追加.....	18
7.6 シート関数の追加.....	19
7.7 マクロの追加.....	19
7.8 コンフィギュレーション管理.....	19

## 図表目次

図2-1	スクリプト言語の文法構造 .....	7
表2.1	演算子の優先順位.....	8
表2-2	関数ライブラリ(SSSLIB) (スクリプトで利用できる関数) .....	9
表2-3	エクセル標準関数.....	12
表3-1	イベント駆動関数 .....	12
表3-2	セル操作関数(ユーザ定義関数及びマクロで利用) .....	12
表3-3	関数ライブラリ(SSSLIB) (関数及びマクロの中で利用できる関数で、表2-2と異なる関数) ...	13
表3-4	スタック操作関数.....	14
表4-1	マクロ .....	15
表5-1	シート関数 .....	15
表5-2	セル操作関数(シート関数で利用) .....	16

## 1 概要

エクセルによる計算においては、各セルに式を定義して計算させるのがよく使われる方法であるが、計算式が複雑になると Visual Basic で関数(ユーザ定義関数)を作成してセルから呼び出して計算させたり、複数のセルを一度に計算する場合にはマクロを作成して実行させたりしている。

ところが、セルに定義した計算式(ユーザ定義関数の呼出し含む)では、当該セルのみに計算結果を返してくるだけなので、ベクトル演算やマトリクス演算を行う場合でも、その計算式を展開して各セル一つひとつに式を定義しなければならず、式の入力が煩雑な上に可読性が非常に悪いものとなっている。そのため、計算手順が複雑になるとマクロを作成することになるが、今度は汎用的にパラメータ(入力セル範囲)を実行時に指定して利用できるマクロにすると、計算過程が第三者に分かり難くなる。

そこで、ベクトル演算やマトリクス演算等の複数の演算結果を返す式をスクリプトとしてセルに定義し、それを解釈実行することで一つのセルに定義した式から複数セルに結果を返すことを可能とした。

本書はそのスクリプト言語の仕様を解説したものである。

## 2 スクリプト言語の文法

### 2.1 言語の概略構造

ユーザは、関数型とマクロ型でスクリプトを定義することができる。

スクリプト言語の文法構造を図2-1に示す。また、参考としてスクリプト言語の構成要素を6項に示す。

### 2.2 関数型スクリプトの文法

#### (1) 標準形式

ユーザは、`exe` 関数を呼出す下記の形式で、定義済みの関数を用いてセル範囲に演算結果を代入する式をシートに記述する。

```
=exe( 式表題, 代入文, 代入文・・・)
```

式表題は、当該セルに表示される文字列や数値等であり、エクセルが直接評価する計算式が記述でき、その演算結果が表示される。

代入文は、指定したセル範囲に演算結果を代入する式を記述する。複数の代入文がある場合、後ろから順に実行される。

#### (2) 拡張形式

エクセルで使用できない演算子を用いる場合、代入文全体または演算子部分を引用符で囲むと、その引用符及びその前後の不要なカンマを削除してエクセルの文法に合致した形式としてから `exe` 関数を呼出す(エクセル文法に合致しない演算子は、文字列データの形式で表現する)。

```
=exe( 式表題, “代入文, 代入文・・・”)  
=exe( 式表題, 代入文の一部, “演算子”, 代入文の一部・・・)
```

拡張スクリプトのみで用いることができる機能は、文末に[拡]を付す。

以後は、標準形式に変換後の説明とする。

#### (3) 代入文

指定したセル範囲に演算結果を代入する式を記述する。

セル範囲 = 式 : 右辺の計算結果を左辺のセル範囲に代入。  
 セル範囲 += 式 : 右辺の計算結果を左辺のセル範囲の各セルの値に加算して代入[拡]。  
 セル範囲 -= 式 : 右辺の計算結果を左辺のセル範囲の各セルの値から減算して代入[拡]。  
 セル範囲 \*= 式 : 右辺の計算結果を左辺のセル範囲の各セルの値に掛けて代入[拡]。  
 セル範囲 /= 式 : 右辺の計算結果で左辺のセル範囲の各セルの値を割って代入[拡]。  
 セル範囲 %= 式 : 右辺の計算結果で左辺のセル範囲の各セルの値を割った余りを代入[拡]。

特例として、戻り値の無い関数は、「式」のみを記述する。

#### (4) 式

エクセルの式の記述に準じ、算術演算、比較演算、論理演算、関数呼出し及びそれらを組合せて式を記述する。

演算子の優先順位は、表2-1に示したとおりであり、単項演算子は指数演算子の次の順位におく。

#### (5) 関数

スクリプトで使用できる関数は表2-2に示した関数ライブラリ(SSSLIB)の他、表2-3に示したエクセル標準関数がある。

なお、上記の表に示した以外の関数が必要な場合、その関数のコードを作成したうえで、スクリプト実行関数[ExecuteFunction]の中からその関数を呼び出す改修を行うことで利用可能となる。

### 2.3 マクロ型スクリプトの文法

ユーザは、exe 関数を呼出す下記の形式で、ユーザが定義したマクロを実行する式をシートに記述する。マクロの演算結果は、マクロの中でセルに書き込むものとする。

=exe( 式表題, マクロ名, 引数, 引数・・・)

式表題は、当該セルに表示される文字列や数値等であり、エクセルが直接評価する計算式が記述でき、その演算結果が表示される。

マクロ名は、実行したいマクロ名を引用符(“)で囲んで記述する。

引数は、そのマクロに渡す引数をセル範囲または即値データ(エクセルが直接評価する計算式)で記述する。

## 3 ユーザが関数やマクロを作成するときに使用できる関数

### 3.1 イベント駆動関数

ユーザがデータシートを追加した場合、表3-1に示したイベント駆動関数を各シートに設定する必要がある。これらはユーザが直接利用することはないが、マウス操作やセル内容変更等によって発生したイベントに従って自動的に実行され、スクリプトを実行させる。

### 3.2 セル操作関数

ユーザが作成する関数の引数として入出力されるデータは、インタープリタがセル操作関数を使ってシートとの間で値を授受するので、下記のように簡潔に記述できる。

```

Function UserFunc (Va As Vector3D, Vb As Vector3D) As Vector3D
'Va      :入力、3Dベクトル
'Vb      :入力、3Dベクトル
'VserFunc :出力、3Dベクトル(ベクトルの外積を正規化したベクトル)
    UserFunc = VctNrm (VctCrp (Va, Vb))
End Function

```

マクロの引数として入出力されるデータは、その引数のセル範囲をパブリック配列 **Args** に設定してからマクロが起動されるので、**Args** 配列を参照してセル操作関数を使ってシートとの間でデータ授受を行う。シートとの間でのデータ授受は、表3-2に示した関数を用い、下記のように記述する。

```

Sub UserSub ()
'Ma :入力、3×3マトリクス
'Va :入力、3Dベクトル
'Do :出力、
    Dim L As Boolean, Ma As Matrix3x3, Va As Vector3D, Do(5)As Double
    L = GetMatrix3x3 (Ma, Args(1)) 'Ma にシート上の3×3マトリクスを取り込む
    L = GetVector3D (Va, Args(2)) 'Va にシート上の3次元ベクトルを取り込む
    Do(1)=..., Do(2)=..., Do(3)=..., Do(4)=..., Do(5)=...
    L = PutDoubleArray (Do, Args(3)) 'Do をシートに設定する
End Sub

```

### 3.3 関数ライブラリ(SSSLIB)

シート上でスクリプト文により利用できる関数ライブラリ(SSSLIB)は、表2-2に示したものが用意されており、その大部分はユーザが関数やマクロを作成するときに利用できるが、一部の関数はスクリプト用とは異なる。表3-3にスクリプト用とは異なる関数を示す。

### 3.4 スタック操作関数

スクリプト実行関数(FunctionExecution)におけるユーザ定義関数の実行は、スタックを経由してデータ授受を行う。ユーザ定義関数に引き渡すデータをスタックから取り出したり、結果をスタックに積む関数として、ユーザは表3-4に示したスタック操作関数が利用できる。

## 4 マクロ

マクロはインタープリタが最小限の面倒しか見ない。マクロの引数は配列 **Args** にセル範囲を示すデータが設定されているだけなので、ユーザがセル操作関数を使ってシートとの間でデータ授受を行う。即値データの引数はそのまま **Args** で引き渡される。参考として作成済みのマクロを表4-1に示す。

## 5 シート関数

エクセルの計算式の演算結果は、その計算式を記述している一つのセルに格納されるものに限定され、複数のセルに跨って演算結果を格納することができない。このようなシート関数として直接(関数 **exe** を用いたスクリプト文でなく)利用できる関数には下記がある。

- (1) 単一データの演算結果を出力する任意の関数(一般のユーザ定義関数、エクセル標準関数)。
- (2) 表2-2の関数ライブラリ(SSSLIB)のうち、計算結果(関数の戻り値)が単一データの関数について、マトリクスやベクトルのように複合データを扱う関数を、セル範囲からデータを取り込んでから呼び出すドライバで包み込んで、シート関数として利用できるようなした関数。
- (3) 表2-2の関数ライブラリ(SSSLIB)のうち、全ての引数が単一セルを参照しており、計算結果(関数の戻り値)が単一データの関数は、そのままシート関数として利用できる。

上記の(1)項の関数として予め用意してある関数及び、(2)項と(3)項の関数を表5-1に示す。

シート関数においてシートから引数データを取得するには表5-2に示したセル操作関数を用い、下記のよう記述する。

```
Function sVctInp (rVa As Range, rVb As Range) As Double
'Va      :入力、3Dベクトル
'Vb      :入力、3Dベクトル
'sVctInp :出力、ベクトルの内積
    Dim L As Boolean, Va As Vector3D, Vb As Vector3D
    L = GetVector3DRange (Va, rVa)   '引数のベクトル Va をシートから取得する
    L = GetVector3DRange (Vb, rVb)  '引数のベクトル Vb をシートから取得する
    sVctInp= VctInp (Vb, Va)        '関数の戻り値は単一のデータでなければならない
End Function
```

任意の構造をしたデータをシートから取得するには、エクセルVBAの文法に従ってコードを記述することになるが、module1 のセル操作関数が参考となる。

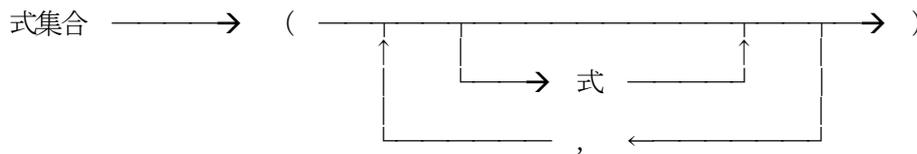
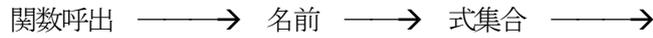
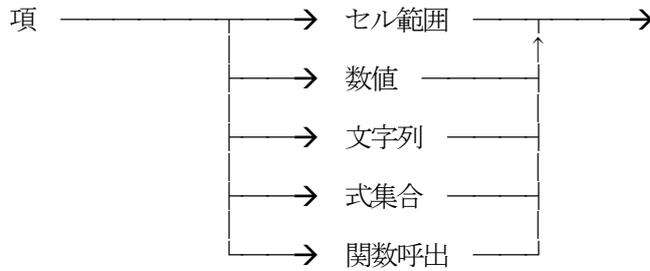
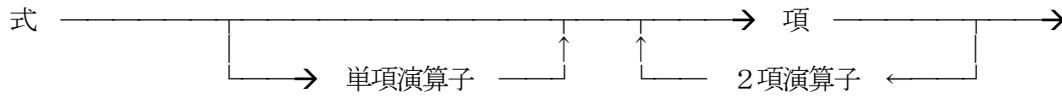
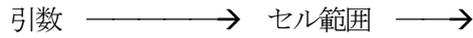
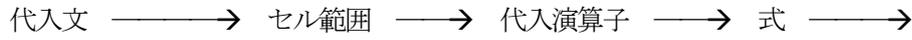
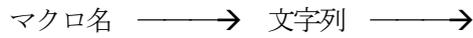
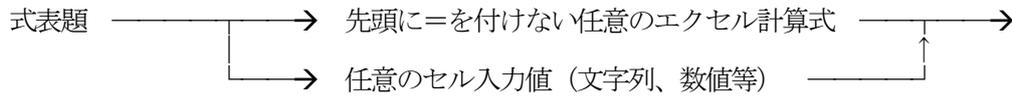
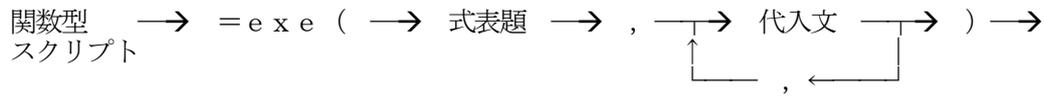


図2-1 スクリプト言語の文法構造

表2.1 演算子の優先順位

分類		演算子	順位	機能
算術演算子 (指数演算子)		^	9	指数演算
単項演算子	算術演算子	+	8	単項演算子(正符号)
		-	8	単項演算子(負符号)
	論理演算子	not	8	単項演算子(否定)
		~~	8	同上[拡]
		~	8	同上[拡]
算術演算子		++	7	1つカウントアップ[拡]
		--	7	1つカウントダウン[拡]
		>>	6	右シフト[拡]
		<<	6	左シフト[拡]
		/	5	除算
		*	5	乗算
		¥	5	整数除算
		mod	5	余り
		%	5	余り[拡]
		-	4	減算
		+	4	加算
		&	3	文字列の連結
		比較演算子		Like
Is	オブジェクトの比較(使用不可)			
<>	等しくない			
>	より大きい			
>=	以上			
==	等しい[拡]			
=	等しい			
<=	以下			
<	より小さい			
論理演算子		not	1	論理否定
		and		論理積
		or		論理和
		xor		排他的論理和
		eqv		論理等価演算
		imp		論理包含演算
		~		否定[拡]
		&		および[拡]
				または[拡]
		~~		否定[拡]
		&&		および[拡]
				または[拡]

(注) 優先順位は数値が大きいほど高い。

表2-2 関数ライブラリ(SSSLIB) (スクリプトで利用できる関数)

No	名 称	機 能
1	Actand (s,c)	逆正接 $\tan^{-1}(s/c)$ [0~360deg]
2	AngMat (a,ix)	多段階の回転を表す方向余弦マトリクスの計算
3	AngQtn (a,ix)	多段階の回転を表す四元数の計算
4	Arcosd (c)	逆余弦 $\cos^{-1}(c)$ [0~180deg]
5	Arsind (s)	逆正弦 $\sin^{-1}(s)$ [-90~90deg]
6	Artand (s,c)	逆正接 $\tan^{-1}(s/c)$ [-180~180deg]
7	AryCpy (a(),ia ,b(),ib,n)	1次元配列のコピー
8	Azimuth (r,v)	方位角
9	BinDgt (Bn,n)	2進数を各ビット列に分解する
10	CarGeo (r)	カルテシアン座標系から測地座標系へ変換
11	Cd7Wgs (Rc,Vc)	C-7系からWGS84系へ変換
12	Cosd (Ang)	角度の単位(度)に対する余弦
13	DcmQtn (a)	方向余弦マトリクスから四元数へ変換
14	DcmRav (a)	方向余弦マトリクスから回転角ベクトルへ変換
15	Deg (a)	ラジアンから度へ単位変換
16	DIP (r,v,beta)	大気中落下点(DIP)の計算
17	DIPr (r,v,beta)	大気中落下点(DIP)の地心半径ベクトル
18	DIPv (r,v,beta)	大気中落下点(DIP)の慣性速度ベクトル
19	DIPx (Rv,Vv,beta,RetR,RetV )	大気中落下点(DIP)の計算(DIP 基本関数)
20	DrvAng (a,w,ix)	回転角(オイラー角)の微係数
21	DrvDcm (d,w)	方向余弦マトリクスの微係数
22	DrvQtn (q,w)	四元数の微係数
23	DrvRav (a,w)	回転角ベクトルの微係数
24	er ()	地球離心率
25	GeoCar (lat,lng,alt)	測地座標系からカルテシアン座標系へ変換
26	GeoDes (lat1,lon1,lat2,lon2)	測地線に沿った距離の計算
27	GeoGra (d)	データ型「GeoGraphicQuantities」の生成
28	GeoPos (t,f)	地理関連諸量の計算
29	GeoXyz (p)	地理データから慣性座標系の飛行状態量の計算
30	Gmst (t)	グリニッジ平均恒星時(平均春分点からグリニッジまでの赤経)
31	Gmu ()	地球重力定数( $m^3/s^2$ )
32	GrmSmt (a)	グラム・シュミットの正規直交化
33	GrnLng (d,t)	グリニッジ平均恒星時(平均春分点からグリニッジまでの赤経)
34	GrvAcc (r)	地球重力加速度
35	IIP (r,v)	真空中瞬時落下点の計算
36	IIPr (r,v,A)	真空中瞬時落下点(IIP)の地心半径ベクトル
37	IIPv (r,v,A)	真空中瞬時落下点(IIP)の速度ベクトル

表2-2 関数ライブラリ(SSSLIB) (スクリプトで利用できる関数)

No	名 称	機 能
38	IIPx (r,v,A, RetR,RetV)	真空中瞬時落下点(IIP)の計算(IIP 基本関数)
39	LatCtG (g)	地心緯度から測地緯度へ変換
40	LatGtC (g)	測地緯度から地心緯度へ変換
41	LchPad ()	射点位置データ
42	LclPos (a)	地表面位置ベクトルの計算
43	LeapSec (mjd)	閏秒の検索
44	Lneqsl (a,n,m)	線形方程式を解く
45	LokAng (t,F,Rs)	ルックアングル計算
46	LsqBep (x,y)	最小自乗法による最適な次数での多項式近似係数を求める
47	LsqPol (x,y,nP)	最小自乗法による多項式近似係数を求める
48	MatAng (v,p,q)	ある視点の座標系から見た2つの方向余弦マトリクス間の回転角(回転角ベクトル)
49	MatAry (m)	3×3マトリクスを1次元配列に変換
50	MatRtn (a,b,ix)	マトリクスの回転
51	Mjd ()	修正ユリウス日の計算
52	Mx31Mp (a,b)	マトリクスとベクトルの乗算
53	Mx33 (d11,d21,d31 ,d12,d22,d32 ,d13,d23,d33)	9要素から3×3マトリクスの生成
54	Mx33Dt (a)	マトリクスのデターミナント
55	Mx33Iv (a)	逆マトリクス
56	Mx33Mp (a,b)	マトリクスの乗算
57	Mx33Tp (a)	マトリクスの転置
58	Mx33V3 (vx,vy,vz)	ベクトルから3×3マトリクスの生成
59	OdrAng (a,ix)	回転角(回転順) [OdrAng.x] <sub>j</sub> ・[OdrAng.y] <sub>j</sub> ・[OdrAng.z] <sub>k</sub> =A
60	OrbElm (t,F)	軌道要素
61	OrbXyz (t,O)	軌道要素から慣性座標系位置及び速度の計算
62	PolyEx (c,n,a)	多項式展開
63	Qtn (c,sx,sy,sz)	四元数の生成
64	QtnAng (v,p,q)	ある視点の座標系から見た2つの四元数間の回転角(回転角ベクトル)
65	QtnAry (q)	四元数を1次元配列に変換
66	QtnCnj (a)	共役四元数
67	QtnDcm (a)	四元数から方向余弦マトリクスへ変換
68	QtnRav (q)	四元数から回転角ベクトルへ変換
69	QtnRtn (a,b,ix)	四元数の回転
70	QtqqMp (a,b)	四元数の乗算

表2-2 関数ライブラリ(SSSLIB) (スクリプトで利用できる関数)

No	名 称	機 能
71	QtqvMp (a,b)	四元数とベクトルの乗算
72	quickSort (sD,odr)	クイック・ソート
73	Rad (a)	度からラジアンへ単位変換
74	RavDcm (a)	回転角ベクトルから方向余弦マトリクスへ変換
75	RavQtn (a)	回転角ベクトルから四元数へ変換
76	Rer ()	地球赤道半径
77	RtnAng (a,ix)	回転角(オイラー角) $[RtnAng.i]_j \cdot [RtnAng.j]_j \cdot [RtnAng.k]_k = A$
78	RtnMat (a,ix)	回転マトリクス $RtnMat=[A]_{ix}$
79	RtnQtn (a,ix)	回転を表す四元数
80	SetPad (Lat,Lng,Alt,Az)	射点位置の設定
81	SetRef (ID ,tl)	慣性座標系基準の設定
82	SetTime (day,time)	日時の設定
83	Sind (a)	角度の単位(度)に対する正弦
84	Site (Lat,Lng,Alt,Az)	データ型「SiteLocation」の生成
85	Sqrt (a)	平方根の計算
86	StaVar (r,v,Qib)	データ型「StateVariables」の生成
87	StdRnd ()	正規乱数の発生
88	Vct3D (x,y,z)	3次元ベクトルの生成
89	VctAdd (a,b)	ベクトルの加算
90	VctAng (a,b)	2つのベクトルの間の角度
91	VctAry (v)	3次元ベクトルを1次元配列に変換
92	VctCrp (a,b)	ベクトルの外積
93	VctInp (a,b)	ベクトルの内積
94	VctMag (a)	ベクトルの絶対値
95	VctNrm (a)	ベクトルの正規化
96	VctRtn (v,a,ix)	ベクトルの回転 $VctRtn=[A]_{ix} \cdot V$
97	VctSmp (v,s)	ベクトルとスカラーの積
98	VctSub (a,b)	ベクトルの減算
99	Wer ()	地球自転レート(deg/s)
100	WgsCd7 (Rw,Vw)	WGS84系からC-7系へ変換

表2-3 エクセル標準関数

No	名 称	機 能
1	if (条件,値1,値2)	条件が真のとき値1、偽のとき値2
2	columns (セル範囲)	セル範囲に含まれる列数
3	exp (a)	指数関数
4	rows (セル範囲)	セル範囲に含まれる行数
5	cos (角度)	角度(ラジアン)の余弦
6	sin (角度)	角度(ラジアン)の正弦
7	pi ()	円周率 $\pi$ の値

スクリプトで使用する必要最小限の関数を組み込んでいる。

表3-1 イベント駆動関数

No	名 称	機 能
1	Worksheet_BeforeDoubleClick	セルのダブルクリック時に、セル範囲が伝達される
2	Worksheet_SelectionChange	セルの選択変更(クリック or ドラッグ)時に、セル範囲及び範囲が伝達される
3	Worksheet_Calculate	シート上で関数実行イベントの発生が通知される

表3-2 セル操作関数(ユーザ定義関数及びマクロで利用)

No	名 称	機 能
1	SelectTargetRange	選択されているセル範囲を取得し、GetTargetRange 関数に伝達する(イベント駆動関数 No.1、No.2 から呼び出される)
2	GetTargetRange	演算対象セルの選択範囲を取得する(セル範囲をマニュアル設定する場合に使用)。
-	CXXXArray	任意型の配列データを倍精度型の配列データに変換する
3	CDBlArray	倍精度型
4	CIntArray	整数型(2バイト長)
5	CLngArray	整数型(4バイト長)
-	GetXXXXXX	シートから XXXXXX 型データを取得する関数(関数またはマクロから呼び出される)
6	GetArray	任意型の配列データ
7	GetGeoGraphicQuantities	地理関連諸量
8	GetInertialFrame	慣性座標系規定データ
9	GetMatrix3x3	マトリクス(3x3)
10	GetOrbitalElements	軌道要素
11	GetQuaternion	四元数

表3-2 セル操作関数(ユーザ定義関数及びマクロで利用)

No	名 称	機 能
12	GetTable	汎用テーブルデータ
13	GetSiteLocation	地上局位置データ
14	GetTrajectoryStateVariables	飛行状態量
15	GetVariant	任意型のデータ
16	GetVector3D	3次元ベクトル
17	GetVectorND	N次元ベクトル
-	PutXXXXXX	シートへ XXXXXX 型データを格納する関数(マクロ(Sub)のみから呼び出される)
18	PutArray	任意型の配列データ
19	PutGeoGraphicQuantities	地理関連諸量
20	PutMatrix3x3	マトリクス(3x3)
21	PutOrbitalElements	軌道要素
22	PutQuaternion	四元数
23	PutTrajectoryStateVariables	飛行状態量
24	PutVariant	任意型のデータ
25	PutVector3D	3次元ベクトル
26	PutViewQuantities	視野情報
27	PutArrayRange	関数引数で範囲指定された任意の型のデータをシートに設定する

表3-3 関数ライブラリ(SSSLIB) (関数及びマクロの中で利用できる関数で、表2-2と異なる関数)

No	名 称	機 能
-	スクリプトで利用できない関数	
1	Derivs ()	微係数評価
2	GeoPos (F)	地理関連諸量の計算
3	GeoXyz (p)	地理データから慣性座標系の飛行状態量の計算
4	GrnLng (d,T)	グリニッジ平均恒星時(春分点からのグリニッジの経度を求める)
5	IntIni (n)	積分制御の初期化
6	LokAng (F,Rs)	ルックアングル計算
7	RkgInt ()	ルンゲクッタ・ギル法による数値積分
8	TblInt (F,ai)	汎用テーブルの補間
-	スクリプト用と引数や戻り値が異なる関数	
9	BinDgt (Bn,Bd,n)	2進数を各ビット列に分解する
10	C7fWgs (Rc,Vc,Vw)	地球固定座標系でC-7系からWGS84系へ変換
11	Lneqsl (a,m,n)	線形方程式を解く

表3-3 関数ライブラリ(SSSLIB) (関数及びマクロの中で利用できる関数で、表2-2と異なる関数)

No	名 称	機 能
12	LsqBep (x,y,nD,nY,c,nP)	最小自乗法による最適な次数での多項式近似係数を求める
13	LsqPol (x,y,nD,nY,c,nP)	最小自乗法による多項式近似係数を求める
14	WgsC7f (Rw,Vw,Vc)	地球固定座標系でWGS84系からC-7系へ変換

表3-4 スタック操作関数

No	名 称	機 能
--	PushXXXXXX	データの格納
1	Array	任意型の配列データ
2	Data	任意型データ
3	Date	日付型データ
4	DblArray	倍精度型の配列データ
5	Ggq	地理関連諸量型データ
6	IntArray	整数型の配列データ
7	Mat	3X3マトリクス型データ
8	Orb	軌道要素型データ
9	Qtn	四元数型データ
10	Time	時刻型データ
11	Tsv	飛行状態量型データ
12	Site	地上局位置型データ
13	Vct	3次元ベクトル型データ
14	View	視野情報型データ
--	PopXXXXXX	データの取出し
15	PopArray	任意型の配列データ
16	PopData	任意型データ
17	PopDate	日付型データ
18	PopDblArray	倍精度型の配列データ
19	PopGgq	地理関連諸量型データ
20	PopIntArray	整数型の配列データ
21	PopMat	3X3マトリクス型データ
22	PopOrb	軌道要素型データ
23	PopQtn	四元数型データ
24	PopTime	時刻型データ
25	PopTsv	飛行状態量型データ
26	PopSite	地上局位置型データ
27	PopVct	3次元ベクトル型データ
28	PopView	視野情報型データ

表4-1 マクロ

No	名 称	機 能
1	DataSmoothing	データを平滑化する
2	Demo_AttitudeIntegrations	姿勢パラメータ積分の演習(四元数、方向余弦マトリクス、オイラー角)
3	Demo_RandomNumber	正規分布の演習(正規分布をする2つの確率変数の和、差、積の分布)
4	TableAndIntegTest	汎用関数テーブルと積分関数のテスト
5	TableTest	汎用テーブルのテスト

表5-1 シート関数

No	名 称	機 能
-	SSSLIB 関数のドライバ ドライバによりシート関数として利用できる SSSLIB 関数(エクセル標準関数と同じ用法) (戻り値が単一のデータで、引数が複数個(構造型)のデータを含む関数)	
1	sAzimth (a,b)	方位角を求める
2	sGrnLng (a,b)	グリニッジ平均恒星時(春分点からのグリニッジの経度を求める)
3	sMx33Dt (a)	マトリクスのデターミナント
4	sPolyEx (C,n,a)	多項式展開
5	sVctAng (a,b)	2つのベクトルの間の角度
6	sVctInp (a,b)	ベクトルの内積
7	sVctMag (a)	ベクトルの絶対値
-	ドライバなしでシート関数として利用できる SSSLIB 関数(Module3 の関数があるまま利用できる) (全部の引数及び戻り値がそれぞれ単一のデータから成る関数)(エクセル標準関数と同じ用法)	
1	Actand (s,c)	逆正接 $\tan^{-1}(s/c)$ [0~360deg]
2	Arcosd (c)	逆余弦 $\cos^{-1}(c)$ [0~180deg]
3	Arsind (s)	逆正弦 $\sin^{-1}(s)$ [-90~90deg]
4	Artand (s,c)	逆正接 $\tan^{-1}(s/c)$ [-180~180deg]
5	Cosd (Ang)	角度(度)の余弦
6	Deg (a)	ラジアンから度へ単位変換
7	er ()	地球離心率
8	GeoDis (lat1,lon1,lat2,lon2)	測地線に沿った距離の計算
9	Gmu ()	地球重力定数( $m^3/s^2$ )
10	Rad (a)	度からラジアンへ単位変換
11	Sind (a)	角度(度)の正弦
12	Sqrt (a)	平方根の計算
13	StdRnd ()	標準正規乱数の発生

表5-1 シート関数

No	名 称	機 能
14	Wer ()	地球自転レート(deg/s)

表5-2 セル操作関数(シート関数で利用)

No	名 称	機 能
-	GetXXXXXXXXRange	関数引数で範囲指定された XXXXXX 型データをシートから取得する関数(シート関数から呼び出される)
1	GetArrayRange	関数引数で範囲指定された任意型の配列データ
2	GetMatrix3x3Range	関数引数で範囲指定されたマトリクス(3x3)
3	GetQuaternionRange	関数引数で範囲指定された四元数
4	GetVariantRange	関数引数で範囲指定された任意型のデータ
5	GetVector3DRange	関数引数で範囲指定された3次元ベクトル
6	GetVectorNDRange	関数引数で範囲指定されたN次元ベクトル

## 6 言語構成要素

Visual Basicで作成したスクリプト言語の構成要素を下記の記号を用いて示す。

- | : または(選択肢)を表す。
- 2重下線付き** : 既定の文字列または終端記号を表す。
- [ ] : カッコ内の要素がゼロ回または1回現れる。
- { } : カッコ内の要素がゼロ回または1回以上繰り返し現れる。
- ( ) : カッコ内の要素が1回現れる。

スクリプト言語の構成要素は下記のとおりである。

- スクリプト := 関数呼出
- 関数呼出 := 名前 式集合
- 式集合 := ( { , 式 } )
- 式 := 代入式 | 演算式
- 代入式 := セル範囲 代入演算子 演算式
- 演算式 := 項 | 単項演算子 項 | 式 { 2項演算子 項 }
- 項 := 要素 | 式集合 | 関数呼出
- 要素 := セル範囲 | 数値 | 文字列
- セル範囲 := 名前 | 名前    名前
  
- 字句要素 := 名前 | 数値 | 文字列 | 演算子 | 区切り記号
- 名前 := 英字 { 英数字 }
- 英字 := a~z | \_ | \$ | @
- 英数字 := 英字 | 数字
- 数字 := 0~9
- 数値 := 整数 | 実数 | 16進数
- 整数 := 数字 { 数字 }
- 実数 := 整数    整数 | 整数 [    整数 ] ( e | d ) [ + | - ] 整数
- 16進数 := 0x ( 数字 | a~f ) { 数字 | a~f }
- 演算子 := 代入演算子 | 単項演算子 | 2項演算子
- 代入演算子 := = | += | -= | \*= | /= | %=
- 単項演算子 := + | - | ~ | ~~ | not
- 2項演算子 := ^ | ++ | -- | << | >> | \* | / | % | ¥ | + | - | & | | | < | <= | = | == | >= | > | <> | && | || | and | or | xor | imp | like
- 区切り記号 := , | ; | :

## 7 維持管理方法

### 7.1 VBA コードファイル

event	イベント駆動関数
Module1	セル操作関数
Module2	インタープリタ
Module3	関数ライブラリ(SSSLIB)
Module4	シート関数
Module5	マクロ

### 7.2 VBA 開発画面

エクセルの[開発]タブを開き、ツールバー左端の[Visual Basic]またはツールバー中ほどの[コード表示]を選択すると、VBA 画面が表示される。VBA 画面は左側に縦長のプロジェクト・エクスプローラー領域、その右側にコード領域がある。シートやモジュールが選択されていないとコード領域は表示されず、背景色になっている。

### 7.3 新しいエクセルファイルでスクリプトを利用できるようにするには

#### (1) イベント駆動関数の挿入

プロジェクト・エクスプローラー領域でシート名をダブルクリックすると、コード領域にコードが表示される。新しいシートでは空白となっている。ここにイベント駆動関数 event ファイルの内容をコピー&ペーストする。event ファイルはテキスト形式なので、「ノート」で開くことができる。

#### (2) 関数ライブラリのインポート

VBA 画面のメニューバーで[ファイル]を開き、[ファイルのインポート]を選択して、Module1 ファイルから Module5 ファイルまで順次、インポートする。これらはプロジェクト・エクスプローラー領域に標準モジュールとして読み込まれ、Module 名が表示される。

メニューバーの代わりに、プロジェクト・エクスプローラー領域で右クリックして[ファイルのインポート]を選択することもできる。

Module のコードを表示するには、プロジェクト・エクスプローラー領域で Module 名をダブルクリックする。

### 7.4 新しいシートを追加したら

イベント駆動関数は各シート毎に必要なので、上記 7.3- (1)項の操作を行う。

### 7.5 関数ライブラリ(SSSLIB)の追加

#### (1) コードの作成

スクリプトで使用する関数は Module3 に定義する。コードの挿入位置は関数名が昇順に並ぶようにするが、一部機能を別関数に分割する場合は近接して配置する。コードには機能概要を 1 行程度で示し、引数の説明や戻り値の説明および要すれば使用条件などを記述する。

#### (2) 関数名リスト

Module3 の先頭に記述されている関数名リストに昇順に追記する。

#### (3) インタープリタへの追加

インタプリタが呼び出せるよう、**Module2** の関数[ExecuteFunction]に呼び出しコードを追加する。追加位置は関数名の昇順とする。引数は全てPushPop スタックに積まれているので、スタックからPopして関数に渡す。結果はPushしておく。詳細は他の定義済み関数を参考に。

## 7.6 シート関数の追加

**Module3** に定義した関数のうち、戻り値が単一の関数はドライバで包むことで、エクセル標準関数と同様にシートから直接呼び出せるようにする。

### (1) コードの作成

ドライバは原関数名の先頭に”s”を付加し、**Module4** に定義する。例えば、**Module3** の原関数が”Vctmag”の場合、”sVctmag”とする。

コードの挿入位置は関数名が昇順に並ぶようにするが、一部機能を別関数に分割する場合は近接して配置する。コードには機能概要を1行程度で示し、引数の説明や戻り値の説明および要すれば使用条件などを記述する。

### (2) 関数名リスト

**Module4** の先頭に記述されている関数名リストに昇順に追記する。

### (3) インタプリタへの追加は不要。

## 7.7 マクロの追加

汎用性の無い、単一用途のコードはマクロとして作成するのが便利である。

### (1) コードの作成

マクロは **Module5** に定義する。

コードの挿入位置はマクロ名が昇順に並ぶようにするが、一部機能を別マクロ・関数に分割する場合は近接して配置する。コードには機能概要、引数の説明や戻り値の説明および要すれば使用条件などを記述する。

引数は配列 Args にセル範囲や即値データとして渡される。セル範囲は入力データ領域や出力データ領域を示す。詳細は定義済みマクロを参考に。

### (2) マクロ名リスト

**Module5** の先頭に記述されているマクロ名リストに昇順に追記する。

### (3) インタプリタへの追加は不要。

## 7.8 コンフィギュレーション管理

各モジュール(**Module1**~**Module5**)の先頭にコンフィギュレーション識別番号を付与しているので、モジュールに変更を加えたら番号を変えて識別できるようにし、その下に変更者、変更日及び変更内容を記述して変更履歴を残す。

識別番号は、新しい関数・マクロを追加したら整数部を+1し、既存のものに変更を加えただけなら小数部を+1する。小数部は1桁なので、細かい変更でも10回目には整数部に桁上げする。

新しい関数・マクロの変更管理の開始時期は、その検証が十分行われてリリースする時点からとする。この時点では新しい識別番号を付与し、新しい関数・マクロを追加した旨、変更履歴に残す。