タイトル	MATLABにおける文字変数の扱い方と日本語文の統計解析
著者	高井，信勝
引用	北海学園大学工学部研究報告，33：183-202
発行日	2006-02-20

＜技術ノート＞MATLABにおける文字変数の扱い方と日本語文の統計解析

高　井　信　勝*

＜Technical Report＞The Treatment of Characters in MATLAB and Its Application to the Statistical Analysis of Japanese Text

Nobukatsu TAKAI*

Abstract

In this report, the treatment of English and Japanese characters which are processed in a computer is described to make use of understanding the modern cryptography such as DES and RSA. Especially, the way by which the characters are treated by the technical computer language MATLAB is explained in detail. The histogram of Hirakana characters in Japanese text is also investigated as an example of the character analysis by a computer.

1．はじめに

インターネットを飛び交う情報は，常に危険にさらされている．悪意を持ったものがいれば容易に盗聴され，悪用される恐れがある．そのため，重要なメッセージは暗号化して送られる．このように，かつては主として軍事機密に用いられてきた暗いイメージの暗号技術は，現代では通信の安全性を確保する手段として新しい地位を占めている[1),2)]．

現代の暗号技術が過去のものと大きく異なる点は，コンピュータを用いたディジタル技術であることである．そこでは，文字は文字そのものとしては扱われずに，各々の文字に対応する数値が文字として扱われる．これを「数値的同値」といい，これを用いることで暗号技術は数学演算を駆使する技術となっている[3),4)]．そのために暗号技術を学習するときには，この数値的同値，つまり文字と数値の関係を知っていなければならない．本稿では，教育的な目的のもとで，文字と対応する数値の関係をまとめ，それが科学技術言語MATLABでどのように利用できるかを述べる[5)]．最後に，数値的同値を用いた日本語文のひらかなに関する試み的な文章

＊北海学園大学工学部電子情報工学科
＊Department of Electronics and Information Engineering, Faculty of Engineering, Hokkai-Gakuen University

解析を示す．

2．文字コード

計算機データは，周知のように，数値データであれ文字データであれ，すべてのデータは0と1によって扱われている．つまり2進数である．2進数では人が見てもその内容を理解できないので，通常，数値データは10進数に変換して表現される（数値は16進数として表現する場合もある）．そこで，文字を，決まった約束のもとで，数値に割り付けておくと，文字も2進数，10進数，場合によっては16進数で表現できる．ところが，これではあるデータが数値であるか文字データであるかが区別できないことになるので，Cやフォートランなどの言語では，あらかじめ「この変数は文字変数である」と宣言して区別し，MATLABでは変数の記述を判断することで，変数の型を自動認識する．

このように，計算機は文字を数値に置き換えて処理している．これを文字コード方式という．文字コードは,1つの文字を1つの数に読み替えるもので，その体系には以下のようなものがある[6]．

10進	文字	10進	文字	10進	文字	10進	文字	10進	文字	10進	文字	10進	文字	10進	文字
0		16		32	Space	48	0	64		80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	“	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7		23		39	‘	55	7	71	G	87	W	103	g	119	w
8	(BS)	24		40	(	56	8	72	H	88	X	104	h	120	x
9	(TAB)	25		41	)	57	9	73	I	89	Y	105	i	121	y
10	(改行)	26		42	*	58		74	J	90	Z	106	j	122	z
11		27	(Esc)	43	+	59		75	K	91	[	107	k	123	{
12		28		44	,	60		76	L	92	¥	108	l	124	|
13	(復帰)	29		45	-	61		77	M	93	]	109	m	125	}
14		30		46	.	62		78	N	94	^	110	n	126	~
15		31		47	/	63		79	O	95	_	111	o	127	(DEL)

表1 標準ASCIIコード表

A. ASCIIコード：米国規格協会（ANSI）が1962年に制定した文字コード・セットである．アルファベットや符号，数字，改行コードなどのキャラクタを８ビットでコード化したもので，数値が０～127の前半の部分を標準ASCIIといい，ISO（国際標準化機構）において国際規格の一部になっている．表１が標準ASCIIコード表で，数値が32～126の範囲に，印刷したときに現れる数字，文字，記号が割り当てられ，これ以外の数値はカーソル制御などに割り当てられている．また，アルファベットの大文字は順に65～90，小文字は97～122の数値に割り当てられているので，同じ文字の大文字と小文字は32だけ隔たっている．

B. JISコード：JIS（日本工業規格）が1978年に制定した日本語を扱うための文字集合で，2965字の第１水準，3388字の第２水準が定められている．第１水準漢字は，当用漢字表1850字と当時の人名用漢字別表120字を全て含んでいる．何度かの修正を加えて2000年に，さらに多くの漢字を表すために約4000字の第３水準／第４水準が選定されている．

C. シフトJIS：JISコードを基に作られた日本語を扱うための文字コード・セットの１つで，パソコンで広く使われている．1983年に米マイクロソフト，アスキー，日本IBM，三菱電機が共同で考案したもので，JISコードと同様２バイトで１つの文字を表すが，コード領域を移動させることで７ビットASCIIコードとの混在で必要となる切り替えコードを不要としている．MS漢字コードともよばれる．

D. Unicode：世界中の文字の多くを，統一された文字コードで表現しようとする規格で，米マイクロソフトや米アップルコンピュータを中心として制定され，ISO（国際標準化機構）で国際規格の一部として採用されており，JISでも取り入れられている．Unicodeはすべての文字を２バイトの文字コードとして表現し，Windowsなどが採用している．Unicodeを採用したアプリケーションなら，どの言語のOSでも利用できる．ただし，２バイトにすべての文字を収めるため，日本語，中国語，韓国語の漢字に関して，同じコードに異なる字体などが割り当てられているなどの問題もある．

表２および表３は，和文コードに関係するシフトJISとUnicordにおける文字の割り当て範囲の概略である．これらでは数値が126まではシフトJISもUnicordもASCIIコードに一致している．言い換えると，どちらの文字コードも最初の部分はASCIIコードがあり，後続の和文文字の部分の配置が異なっている．

MATLABでは，欧文コードとして（半角）英数字にはASCII（表１）が用いられている．また，和文コードとしては，MATLAB Version 6 まではシフトJISコード，Version 7 以降ではUni-

cordが使われているので注意を要する．本稿では，MATLAB Version7.1を使用した．なお，ASCII文字は7ビット，和文コードは16ビットで表現されるが，MATLABではその認識は自動的になされる．

文字	割り当て数値範囲（10進）	割り当て数値範囲（16進）
半角英数字	33～126	0021～007E
半角カタカナ	161～223	00A1～00DF
記号	33089～33276	8141～81FC
全角英数字	33839～33434	824F～829A
ひらがな	33439～33521	829F～82F1
全角カタカナ	33600～33686	8340～8396
ギリシャ文字	33695～33750	839F～83D6
漢字（第1水準）	34975～39026	889F～9872
漢字（第2水準）	39071～60068	989F~EAA4

表2 シフトJISの文字割り当て範囲（一部分省略）．

文字	割り当て数値範囲（10進）	割り当て数値範囲（16進）
半角英数字	33～126	0021～007E
ギリシャ文字	913～969	0391～03C9
ひらかな	12353～12436	3041～3094
全角カタカナ	12449～12542	30A1～30FE
漢字（一部を除く）	19968～40869	4E00～9FA5
全角英数字	65281～65375	FF01～FF5F
半角カタカナ	65382～65424	FF66～FF90

表3 Unicordの文字割り当て範囲．漢字ではその範囲の中に割り当てられていない数値が点在する．

3．文字データの数値変換

表4に文字データ変換に関するMATLABコマンドを示す．文字は，**double**コマンドを用いて10進数に変換され，10進数は**char**コマンドを用いて文字に変換される．文字および10進数データは**dec2bin**コマンドを用いて2進数に変換される．つまり，10進数のXを，Xbin=**dec2bin**(X)で変換すると，Xの2進数が得られる．また，Xが文字データであれば，その文字に割り当てられているUnicord値の数値（10進数）の2進数が得られる．逆に，2進数から10進数への変換には**bin2dec**を用いる．ちなみに，コマンド名**dec2bin**はdecimal to binary（10進数から2進数へ），コマンド名**bin2dec**はbinary to decimal（2進数から10進数へ）に由来している．以

下，具体的に文字あるいは文字配列の変換例とそこにみられる留意点を述べる．

なお，表中には10進数を16進数に変換する**dec2hex**，16進数から10進数に変換する**hex2dec**，16進数から倍精度数値に変換する**hex2num**がある．これらについては，ここでは触れない．

関数	記述	説明
double	Xdec=**double**(X)	文字Xを10進数（double）に変換
bin2dec	Xdec=**bin2dec**(Xbin)	2進数データXbinを10進数データ（double）に変換
char	Xch=**char**(Xdec)	10進数Xdecを文字に変換
dec2bin	Xbin=**dec2bin**(X)	文字Xを2進数に変換．Xは文字データの配列．
	Xbin=**dec2bin**(X, N)	文字Xあるいは10進数Xdecを2進数に変換．Nを指定するとNビットの2進数に変換．あるいはまた，任意の正の整数（X＜2^52）を2進数に変換．Nを指定するとN桁の2進数を出力
dec2hex	Xhex=**dec2hex**(Xdec)	10進数Xdecを16進数に変換
hex2dec	Xdec=**hex2dec**(Xhex)	16進数Xhexを10進数に変換．Xhexは16進数を表す文字変数
hex2num	X=**hex2num**(Xhex)	16進数Xhexを倍精度数値に変換

表4　文字データ変換に関するコマンド．語源的には，dec，bin，char，hexはそれぞれdecimal（10進），binary（2進），character（文字）hexadecimal（16進）に由来している．

(1) 1文字の2進数変換

たとえば，

X='あ'

に対して，**double**コマンドを用いて，Xdec=**double**(X)を実行すると，Xdec＝12354　が出力される．これが文字「あ」のUnicord値である．また，2進数に変換するコマンド**dec2bin**をもちいて，Xbin=**dec2bin**(X)を実行すると，2進数

Xbin＝11000001000010

が得られる．この結果は，10進数XdecをXbin=**dec2bin**(Xdec)と変換することでもえられる．これは，コマンド**dec2bin**の引数が文字の場合は，その対応する10進数として扱われるからである．なお，結果の2進数データの型は文字データ（**char**）である．

2進数Xbinの文字を知るときには，最初に**bin2dec**を用いて10進数に変換し，その後で**char**コマンドを用いる．つまり，Xch=**char**(**bin2dec**(Xbin))を実行すると，結果は，文字データ

Xch＝あ

が得られる．以上の関係を図式的に示すと，図1のようになる．

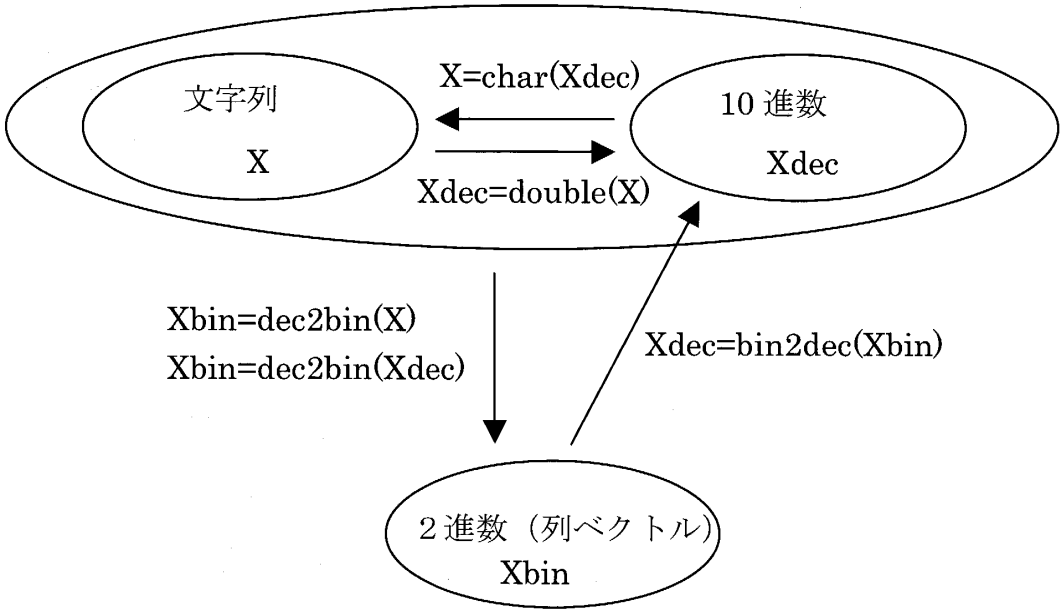

図1 MATLABでサポートされている文字列，10進数，2進数に関する変換コマンドの関係

⑵文字列の変換

つぎに，1次元行ベクトルとして与えられる文字列の変換を調べる．たとえば，

X='北海道'

を定義すると，これは1行3列の行ベクトルである．これの2進数をXbin=**dec2bin**(X)によって求めると，

Xbin＝0101001100010111
0110110101110111
1001000001010011

が得られる．これはUnicordが1文字を16ビット（2バイト）で表現されていることによる3行16列の2進数配列（0と1の文字データ配列）である．つまり，MATLABでは文字列Xが行ベクトルであっても，その2進数は16ビットが単位になった列ベクトルとして得られる．この結果，Xdec=**bin2dec**(Xbin)から得られる等価な10進数は列ベクトルとなり

Xdec＝21271
28023
36947

となる．これからXch=**char**(Xdec)を求めると

Xch＝北
海
道

となる．このように，一度２進数に変換したのちに，文字配列を復元すると，最初のXを得るにはこの転値を取らなければならないことに注意する必要がある．

⑶文字行列の変換

さらに，一例として，Xが

X＝北海道の
　　秋の山々
　　は美しい

で与えられる３行４列の行列の場合を調べてみよう．これを一度２進数に変換し，それから10進数（Xdec），およびそれに等価な文字配列（Xch）はつぎのように得られる．

Xbin		Xdec		Xch
0101001100010111		21271		北
0111100111001011		31179		秋
0011000001101111		12399		は
0110110101110111		28023		海
0011000001101110	Xdec=**bin2dec**(Xbin)	12398	Xch=**char**(Xdec)	の
0111111110001110	⇨	32654	⇨	美
1001000001010011		36947		道
0101110001110001		23665		山
0011000001010111		12375		し
0011000001101110		12398		の
0011000000000101		12293		々
0011000001000100		12356		い

これをみてわかるように，Xが行列（２次元配列）の場合には，一度２進数に変換したのちにもとの行列Xを求めることは簡単ではない．これは前述したように，文字行列Xの２進数を求めるとXの１列目の２進数が最初に得られ，つぎに２列目，３列目，…と列ごとの２進数が順次得られるからである．

結論的に，文字行列を２進数として扱ったのちに，それに等価な10進数を求めたり，もとの文字配列を求めることは避けた方がよい．文字行列の変換は，一旦文字配列を１次元配列のベクトルに変換し，⑵で述べた文字列の変換として扱わなければならない．もとの文字行列が必要であれば，結果の１次元配列（ベクトル）から２次元配列（行列）を作成する．[プログラムリスト１]はこれを実行する一例である．

［プログラムリスト１］文字行列の変換（例）

```
% hokkaido_no_aki. m
clear ;
X0＝['北海道の' ; '秋の山々' ; 'は美しい']        %文字の２次元配列の定義
[m,n]=size(X0) ;
X=[] ;
for k=1 : m ; X=[X,X0(k,1 : n)] ; end              %１次元配列（行ベクトル）に変換
Xbin=dec2bin(X) ;                                  %２進数（列ベクトル）へ変換
Xdec=bin2dec(Xbin) ;                               %10進数（列ベクトル）へ変換

Xdec2=[] ;
for k=1 : m ; x1=n*(k−1)+1 ; x2=x1+n−1 ; Xdec2=[Xdec2 ; Xdec(x1 : x2)'] ; end
Xdec2                                              %10進数行列の表示
Xch=char(Xdec2)                                    %文字行列とその表示
%End of file
```

このプログラム中で，文字行列X0が，まず行ベクトルXに変えられている．このとき，最初に空ベクトルX=[] が定義され，for文を用いてXの中に次々と文字を順に追加していく処理がなされている．その結果，for文を抜け出た後では文字列のベクトルが得られる．

後半の空ベクトルXdec2=[] は，逆に，文字データベクトルXdecから行列数値データを復元するために用いられている．このプログラムの場合には，m＝３，n＝４であるので，ベクトルが３行４列の行列に変換される．

４．テキストデータの読み込み

MATLABでは，テキストファイルの文章を読み込むコマンドとして，MATLAB Version 6 以降から，表５にある**importdata**が用意されている．

TXT=**importdata**(‘*filename.txt*’)

を実行すると，そのファイルにかかれているテキスト文章が読み込まれ，変数TXTとして定義される．しかし，TXTはセル配列であって，それがそのままテキストの文字配列であるわけではない，一般に，文章は改行で区切られた段落の集まりであり，TXTの内容は改行までを単位にした段落ごとの文章の集まりである．そのため，長さが異なるテキストデータを要素にし

たセル配列として扱われている．

セル配列の要素，つまり各段落の文章取り出すためには表５のコマンド**deal**を用いる．テキスト全体の段落をT1, T2, T3, ...と定義して取り出すときには

[T1, T2, T3, ...]=**deal**(TXT{ : })

と記述する．ここで，｛:｝はすべてのセル配列要素を意味する．これによってTXTのすべての要素TXT{ : }が，順に，複数の出力変数[T1, T2, T3, ...]に格納される．つまり，T1は１番目の段落，T2は２番目の段落…ということになる．また，n番目の段落（要素）だけを取り出すときには，

T=deal(TXT{n})

と記述する．

関数	記述	説明
importdata	TXT=**importdata**(*'filename.txt'*)	ファイル名 *filename*のテキストファイルからデータを読み込む記述．TXTは段落をセル要素とするセル配列．
deal	[T1, T2, T3,...]=**deal**(TXT{ : })	セル配列TXTをT1, T2, T3...に分解．
	T=**deal**(TXT{n})	n番目の段落をTに格納．

表５　テキストファイルからのデータ読み込みに用いるコマンド．

［プログラムリスト２］テキストファイルを読み込むプログラム

```
% ReadingText. m
clear；close all
TXT=importdata('司馬遼太郎(1).txt')；　%テキストファイルの読み込み　⇒　cell配列
T1=deal(TXT{1})；
T2=deal(TXT{2})；
T3=deal(TXT{3})；
L＝40；
text1＝fnc_TextMatrix(T1, L)；　　%段落１
text2＝fnc_TextMatrix(T2, L)；　　%段落２
text3＝fnc_TextMatrix(T3, L)；　　%段落３
text=[text1 ; text2 ; text3]　　%段落の結合と表示
%End of file
```

［プログラムリスト２］は，テキストファイルから文字データを読み込むプログラム例であ

る．このプログラムでは，テキストファイル「司馬遼太郎(1).*txt*」が最初にセル配列TXTとして読み出されている．このファイルの内容はつぎの文章１である．

文章１（司馬遼太郎の文章）

司馬遼太郎「歴史の世界から」，p.338　『自分の作品について』より

小説を書くという作業は，自分自身の中に普遍的人間が厳然として住んでいて，それがいかに奇妙な心理や行動を表現しようとも，本来普遍性からはずれることがないという，いわば証明不要な公理のようなものを信じる以外に書けるものではない．と，そのように思うのは，私のかぼそい小説的教養がそう私に思わせるだけで，私自身は不幸なことにそういう作家ではない．私はどうにも，自分自身の生活や行動や心理については，それを記録して他人に見せるだけの話題価値を見出せない人間なのである．たとえば私は自分の本の裏に自分の略歴が出ているだけですぐに目を伏せたくなるほどに不愉快だし，自分の本名を活字でみるだけでも，羞恥と説明しがたい憎悪に似た感情をもってしまう．そういう人間が，たとえ座談の場合でも自分自身の内面もしくは外面を語らなければならないというような表現行動に，とても参加できない．そういう意味では，私には作家であることの本来的なものが欠けているように思う．

そうでありながら，人間と人間関係を見たり考えたり感じたりすることがたまらなく好きだということと，どういうぐあいに噛みあっているのか，自分でもよくわからない．あるいは，自分以外のすべての人間と人間現象に関心があると図式的にいってしまうと，それで済むことかと思ったりする．しかしそうでもなさそうにも思える．

この文章は,1行目のヘッダーの部分の１番目の段落と，「小説を書く作業は，…」に続く２番目の段落，そして「そうでありながら…」以降の３つの段落からなっている．プログラムでは，おのおのの段落を**deal**コマンドを用いてT1, T2, T3として取り出している．これらは１行からなる文字配列で，これをコマンドウインドウに表示すると，配列が長すぎてコマンドウインドウに納まらないことがしばしば起きる．このような場合には,1行の長さを希望の長さに指定して２次元の文字行列に変換するとよい．

行ベクトルを指定した列数（文字数）からなる行列に変換すると，段落の文章全体をコマンドウインドウに表示できるとともに，複数の段落を結合した行列を作成することもできる.［プログラムリスト２］の後半では，指定する文字数の列からなる行列に変換するファンクションMファイル**fnc_TextMatrics**を用い，最後に３つの段落が結合されている．このファンクションMファイル**fnc_TextMatrics**を［プログラムリスト３］に示す．

［プログラムリスト３］行ベクトルを列数Lの行列に変換するファンクションMファイル

```
% fnc_TextMatrix. m
%行ベクトルテキストをM行L列の行列に変換する関数Mファイル
%----------------------------------関数の宣言----------------------------------
function  X=fnc_TextMatrix(T,L) ;        %入力の引数        T：行ベクトルのテキスト
                                         %   L：変換後の１行の文字数
%  ------------------------------------------------------------------------------
%１行をL文字に変換（最終行を除いた処理）
sizeT=length(T) ;                        %Tの全長（文字数）
X=[];                                    %処理データを格納する行列  ⇒  出力変数
m=0 ;
while 1
    if (m+1)*L>sizeT ; break ;           %最終行の検出
    else
    X=[X ; T(m*L+1 : (m+1)*L)] ;         %m行目にL文字格納
    m=m+1 ;
    end
end
% ----------------------ここまでで，最終行を除いた文字行列が完成 ----------------------
%最終行処理
Xend=ones(1,L)*12288 ;                   %12288（3000hex）は全角スペース
Xend=char(Xend) ;                        %最終行を空白文字に初期化
s=sizeT-m*L ;                            %最終行の文字数
Xend(1 : s)=T(m*L+1 : sizeT) ;           %最終行文字に置き換え
X=[X ; Xend] ;                           %最終行を追加（出力データ）
% End of file
```

ファンクションMファイルでは，一般に，最初に関数を

function Xmozi=**fnc_TextMatrix**(T,L) ;

のように宣言する．この記述は，関数名 **fnc_TextMatrix**の入力の引数としてT（入力テキスト），L（１行の文字数）を与え，後続の実行で得られるXmoziを出力する処理である．なお，このプログラムファイル名は，関数名と同名にするとよい．

処理の内容は，無限ループの while文を用いて，テキストTの最初のL文字を空行列Xの１行

目に，つぎに（L+1)番目からのL文字を２行目に，(2L+1)番目からのL文字を３行目に…と格納していくものである．ただ，最終行の処理に注意が必要である．読み込まれたテキストの文字数は様々であるので，最終文字がどの位置で終わるかはテキスト次第であり，その位置がどこでも出力は行列として閉じたもの（長方形）でなければならない．そのため，最終行（Xend）は最初に全角の空白スペース（Unicord 12288）を与え，最終行の文字列をその最初の位置に上書きする処理を行っている．このようにして未知の文字数の段落文を，指定した列数Lの行列に変換している．行数は，結果として自動的に定まる．

5．文章解析：テキスト中のひらかなの頻度分布

英文の文章に使われるアルファベットの頻度は，調べる文章によっても異なるが，多くの調査でe, t, a, o, i… の順であることが知られている．そして，この頻度分布が暗号解読に利用されたりする．筆者は和文の文章でひらかながどのような頻度で用いられているかに興味をもった．この章では，上述の文字変数の扱い方を適用して３人の作家によるエッセィ風の文章に現れるのひらかなの頻度分布を定量的に調べた．

文字	10進	16進	文字	10進	16進	文字	10進	16進	文字	10進	16進
／	12352	3040	さ	12373	3055	な	12394	306A	み	12415	307F
ぁ	12353	3041	ざ	12374	3056	に	12395	306B	む	12416	3080
あ	12354	3042	し	12375	3057	ぬ	12396	306C	め	12417	3081
ぃ	12355	3043	じ	12376	3058	ね	12397	306D	も	12418	3082
い	12356	3044	す	12377	3059	の	12398	306E	ゃ	12419	3083
ぅ	12357	3045	ず	12378	305A	は	12399	306F	や	12420	3084
う	12358	3046	せ	12379	305B	ば	12400	3070	ゅ	12421	3085
ぇ	12359	3047	ぜ	12380	305C	ぱ	12401	3071	ゆ	12422	3086
え	12360	3048	そ	12381	305D	ひ	12402	3072	ょ	12423	3087
ぉ	12361	3049	ぞ	12382	305E	び	12403	3073	よ	12424	3088
お	12362	304A	た	12383	305F	ぴ	12404	3074	ら	12425	3089
か	12363	304B	だ	12384	3060	ふ	12405	3075	り	12426	308A
が	12364	304C	ち	12385	3061	ぶ	12406	3076	る	12427	308B
き	12365	304D	ぢ	12386	3062	ぷ	12407	3077	れ	12428	308C
ぎ	12366	304E	っ	12387	3063	へ	12408	3078	ろ	12429	308D
く	12367	304F	つ	12388	3064	べ	12409	3079	ゎ	12430	308E
ぐ	12368	3050	づ	12389	3065	ぺ	12410	307A	わ	12431	308F
け	12369	3051	て	12390	3066	ほ	12411	307B	ゐ	12432	3090
げ	12370	3052	で	12391	3067	ぼ	12412	307C	ゑ	12433	3091
こ	12371	3053	と	12392	3068	ぽ	12413	307D	を	12434	3092
ご	12372	3054	ど	12393	3069	ま	12414	307E	ん	12435	3093

表６　10進数－16進数とひらかなの対応表（Unicord）

文章はいずれも２つの段落からなるものであるが，それに筆者が冒頭部分にヘッダーを加えたので，各々のテキストファイルは３段落からなっている．したがって，解析の際には，第１段落を取り除いておこなった．

解析の手順は，以下の５ステップを経る．すなわち，

ステップ１：まず，複数の段落からなるテキスト文書を読み込み，全文を１行のベクトルとして定義する．

ステップ２：つぎに，その中からひらかなだけを抽出する．ひらかなの抽出は，Unicordでは表６に示すように10進数に対応しているので，10進数で12353から12435までの文字を取り出す．

ステップ３：10進数からなるひらかなの文字群のヒストグラムを求め，表示する．

ステップ４：頻度の大きい順にランキング付けをおこなう．

ステップ５：結果を表示する．

［プログラムリスト４］は，ステップ１～３を実行する．これが主プログラムで，全文からひらかなだけが抽出され頻度分布が得られる．引き続く［プログラムリスト５］はヒストグラムの結果からひらかなの使用度数のランキングを求め，結果をコマンドウインドウに表示するプログラムである．このプログラムでは，同時に全文字数に対するひらかなの割合が示される．

［プログラムリスト４］ひらがなの使用頻度を調べるプログラム

```
% Mozi_Hindo. m
%文字データを読み，使われているひらかなの頻度をしらべる
clear；close all
% Step 1 ----------テキスト文書の読み込み----------
disp('') ; disp('>>>Step1')
TXT=importdata('司馬遼太郎(1).txt') ;          %テキストファイルの読み込み⇒cell配列
NT=length(TXT) ;
T=[] ;
T2=deal(TXT{2}) ; T3=deal(TXT{3}) ;          %cell配列の分解　⇒　行ベクトル
T=[T2,T3] ;
% Step 2 ----------ひらかなの抽出（第２段落と第３段落）----------
disp('') ; disp('>>>Step2')
```

```
N=length(T) ;                          %Tの全長（文字数）
Number_of_Mozi=N
aiueo=[];                              %ひらがなを格納する空行列
for k=1 : N
    if T(k)>=12353&T(k)<=12435 ; aiueo=[aiueo, T(k)] ; end
                                       %「あ（12353)」から「ん（12435)」までを格納
end
Hira=fnc_TextMatrix(aiueo,40)          %ひらがなの表示
% Step 3 ----------ひらかなのヒストグラム（頻度分布） ----------
disp('')
disp('>>>Step 3 ：See Figure 1 ')
Hira_dec=double(aiueo);                %かな文字を10進数に変換
x＝12353：12435；
hist(Hira_dec, x) ;                    %かな文字のヒストグラムとその表示⇒ Figure 1
```

[プログラムリスト５]（続き）ひらがなの使用頻度を調べるプログラム

```
% Step 4 ----------頻度ランキング ----------
disp('') ; disp('>>>Step4')
[hindo, mozi]=hist(Hira_dec, x) ;   %hindo：頻出度数　mozi：かな文字（10進数）
Hindo0=sort(hindo) ;                %頻度の小さい順のソート
Hindo0=fliplr(Hindo0) ;             %頻度の大きい順の並び替え：　ysorted(1)が最頻度
%ランキング順の文字の検出
kmax=length(x) ;
Hindo=Hindo0(1 : kmax) ;
Mozi=[];                            %ランキング順の文字を入れる空行列
for kk=1 : kmax
for k=1 : length(hindo)
  if hindo(k)==Hindo(kk) ; Mozi=[Mozi, mozi(k)] ; hindo(k)=hindo(k)+max(hindo) ; end
                                    %Hindoは降順頻度
                                    %Moziは降順頻度に対応する文字の数値
end
```

```
end
% Step 5 ----------結果の表示（文字）----------
disp('') ; disp('>>>Step5')
CMozi=char(Mozi) ;                          %頻度の大きい順の文字の並び
disp('') ; disp('     頻度順位の表示')
disp(' ----------')
for k=1 : 30 ;                              %ランキング表示は30位まで
    disp(sprintf('%g   %s   %g',k,CMozi(k),Hindo(k)))
end
disp('----------') ; disp('')
Number_of_Hirakana=length(aiueo)            %ひらがなの個数
Ratio_of_Hirakana=length(aiueo)/length(T)   %ひらがなの割合
% End of file
```

ここでは，前節で示した文章１（司馬遼太郎の文章）のほかに，つぎの文章２（山田太一の文章）と文章３（三浦綾子の文章）を試みに調べた．これらは以下の文章である．

文章２（山田太一の文章）

山田太一著「いつもの雑踏いつもの場所で」，p.35,『他人の生活』より

私は主としてテレビドラマのシナリオを書いて暮らしている．すると人に「夜型でしょう」とよくいわれる．昼間は寝ていて，夜中に煙草のけむりがたちこめる部屋で，本にうずもれてペンを走らせているのだろう，などと想像されてしまうのである．夜中の二時に電話がかかって来て「いつお電話をしたらいいか迷いましてね．とにかく朝は駄目だし，昼はと考えると，なにをしてらっしゃるか見当がつかない．今頃なら確実に起きてらっしっると思って」といわれて，まったく人間が他の人間の生活を想像する時，いかに通念に支配されていることか，と改めておどろくのだが，これはまあ極端な例にしても，自分以外の人の現実というものは，分からないものである．看護婦さんの生活を思い浮かべても，実になんともテレビドラマに出て来る看護婦さんの姿ぐらいしか頭に浮かばない．そして，現実の看護婦さんたちは，テレビドラマの中の看護婦さんを見て「あんなのないわよねえ」などといっているに相違ないのだから，私も人のことはまったくいえない．

私は朝型で早起きで，煙草はすわず，仕事場に本を積み上げたりはしないしペンは使っ

たことがない.

文章3（三浦綾子の文章）

三浦綾子著「わが青春に出会った本」, p.50,『詩と私』より

「若い日に尊敬できる詩人を持つことのできる人は幸せである」という意味の言葉を知ったのは，何という外国小説の中であったろうか．私はこの言葉を知った時，詩人とはそれほどにも一人の人生に大きくかかわる存在なのかと，驚きを持った覚えがある．そしてまた，この言葉が，尊敬できる教師でもなく，宗教家でもなく，画家でもなく，作家でもないところに驚いたのである．詩人がいかに畏敬すべき存在であるかを，私はこの言葉に教えられた．以来，私は今に至るまで，詩人に対して全く別世界の人間に対するような尊敬を抱いて来た．むろんその尊敬の内容は，年齢と共に変わってきたが．しかし，この言葉は日本の一般社会には通用しないことを，後で知るようになった．日本人には，収入の多寡で人を評価する妙な価値観を持っている者が多いのだ．

私が初めて詩というものに接したのはいつの頃だろう．童詩というのがある．読むのではなく，うたって覚える童謡もある．私が最初に覚えたのは，多分その童謡でもなく，童詩でもなく，子守歌であったような気がする．

文章	全文字数	ひらかなの文字数	ひらかなの割合
文章1（司馬遼太郎）	569	367	64.5%
文章2（山田太一）	484	311	64.3%
文章3（三浦綾子）	448	271	60.5%

表7 3つの文章の文字数とひらかなの割合

まず，3つの文章の文字数とひらかなの割合を表7に示す．これにみられるように，全文中で使われるひらかなの割合はいずれの文章とも60%～65%ほどであった．解析に用いた文字数は少ないが，エッセィ風の文章ではこの程度のものかもしれない．

［プログラムリスト4］を用いて得られた3つの文章のヒストグラムを図2～図4に示す．いずれも横軸は文字に対応するUnicord値で，縦軸は度数である．ただ，分かりやすくするために度数の大きい文字を図中に書き加えてある．

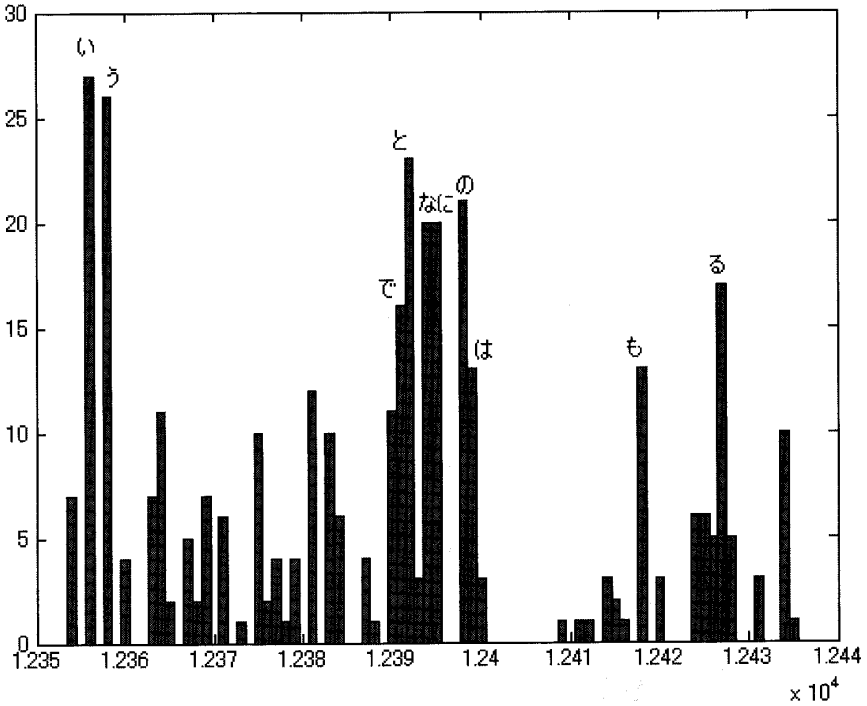

図2　文章1（司馬遼太郎）のひらかなのヒストグラム.

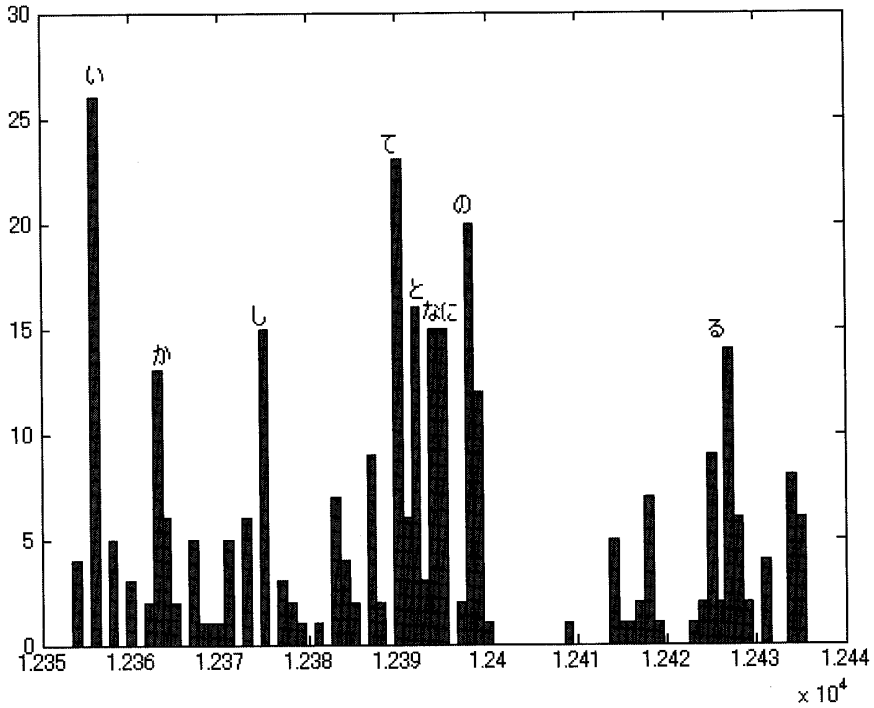

図3　文章2（山田太一）のひらかなのヒストグラム.

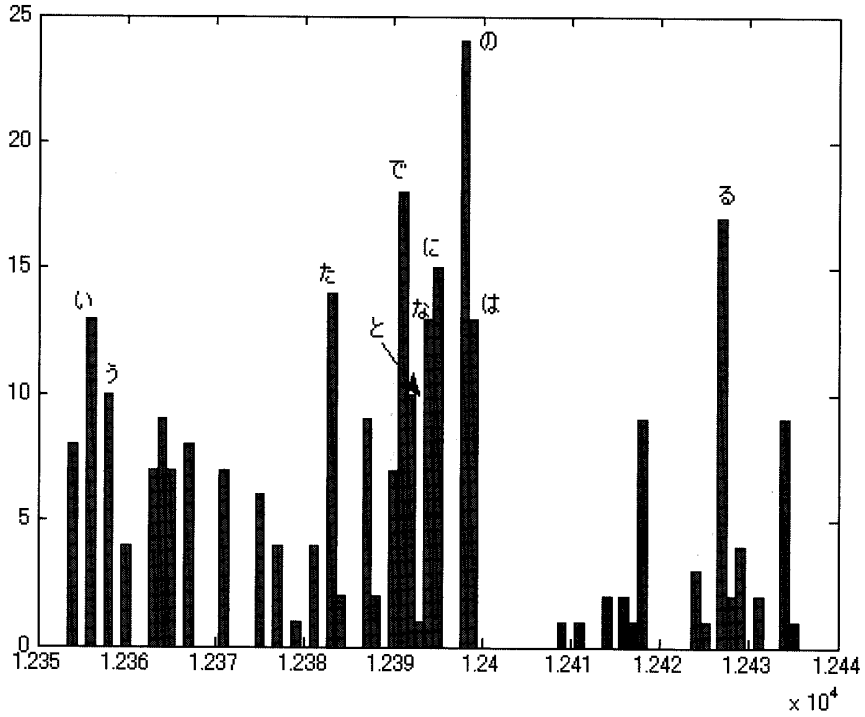

図4　文章3（三浦綾子）のひらかなのヒストグラム.

表8は，ヒストグラム分布からえられた20位までの頻出ランキングである．以下，この結果を踏まえて，筆者なりの分析を述べる．まず,3者の文章とも，「の」，「に」，「は」，「が」という助詞が上位にランクしているのは，膠着語としての日本語文の共通の特徴とおもわれる．作者の文体なり，記述のクセは助詞以外の文字に現れている．

文章1（司馬遼太郎）では「い」，「う」，「と」が1位から3位を占めている．これは文中に「～という」表現が数多く用いられていることによるであろう．また，他の2つの文章では20位までにランクされていない「そ」が11位にみられる．これは，「その」，「それ」あるいは「そういう」という連体語が好んで用いられていることによるとおもう．このように，文章1では前の語句や文を受ける表現が特徴的である．

文章2（山田太一）では，文章1と同様に「い」が最上位にランクされている．しかし，「う」がランクインしていないことを考えると，文章1での「い」の使い方とは異なり，「～いる」，「～ない」という表現が使われていることによる．文章2の明瞭な特徴は，他では低ランクの「て」が2位にあることである．これは，文中に「～して」，「～されて」という口語的な表現が使われていることによると思われる．もう一つは，促音文字の「っ」が11位にあることである．促音は，話し言葉に多く現れるので，これら2つの特徴を考え合わすと，文体が口語調であることは頻度分布から推論できる．

文章3（三浦綾子）でも，促音「っ」が多用されていてこれも文体が口語調であることを示

唆する．もうひとつの特徴は，「た」が上位にランクしていることであって，これは文章が過去形のセンテンスを多く含んでいることによる反映と考えられる．このことと併せて，この文章のもうひとつの特徴が，図4のヒストグラム分布にみられる．これを他の2つのヒストグラム分布と比べてみると，相対的に，た行とな行の文字がよく使われていることがわかる．これは，過去形を含む文章であることと，「でもなく」という表現が繰り返し用いられていることによるとおもわれる．

順位	文章1（司馬遼太郎）		文章2（山田太一）		文章3（三浦綾子）	
1	い	27	い	26	の	24
2	う	26	て	23	で	18
3	と	23	の	20	る	17
4	の	21	と	16	に	15
5	な	20	し	15	た	14
6	に	20	な	15	い	13
7	る	17	に	15	な	13
8	で	16	る	14	は	13
9	は	13	か	13	う	10
10	も	13	は	12	と	10
11	そ	12	っ	9	が	9
12	が	11	ら	9	っ	9
13	て	11	を	8	も	9
14	し	10	た	7	を	9
15	た	10	も	7	あ	8
16	を	10	が	6	く	8
17	あ	7	さ	6	か	7
18	か	7	で	6	き	7
19	け	7	れ	6	こ	7
20	こ	6	ん	6	て	7

表8　頻度の20位までのランキング．数字は度数．

6．おわりに

現代の暗号技術の主流となっているDES暗号やRSA暗号といえども，ある文字を他の文字に置き換えるという意味では，古代から知られているシーザー暗号と変わらない．そして，これらを理解するためには，計算機において文字がどのように扱われているかを知らなければならない．また，このような換字方式の暗号技術とは別に，筆者は最近，文字配列を画像として取り扱う新しい暗号技術を提案した[7),8)]．これにおいてもまた，文字データがどのようにして画像データとして扱えるかを理解する必要がある．

本稿では，このような観点から，暗号技術を理解するために必要な文字変数に関して，計算

機における扱い方，特にMATLABでの利用法を記述した．また，直接には暗号技術に関係しないが，文字を数値的同値として扱うことで，日本語文章のひらかな文字のヒストグラム分布を求めることによる文章解析をおこなった．その結果から作者の文体の特徴をを探ることができることを示した．ただし，扱った文章は，わずか十数行ほどの短いものであり，現段階では試み的な文章解析にすぎない．しかし，ここでは文字データを計算機に取り込んで処理する手法が確認されたので，今後，種々の文体に関して長大な文字数からなる文章解析を行うことが興味深く残されている．

本稿は，北海学園大学ハイテクリサーチセンター事業の研究プロジェクト「視覚・画像・音声・言語情報処理の高度化と知的計測制御技術への応用」の研究に必要な文字の取り扱いについて記述した．

【参考文献】

1）セアラ・フラナリー，デイヴィッド・フラナリー著（亀井よし子訳）：「16歳のセアラが挑んだ世界最強の暗号」，（NHK出版，2001）．
2）サイモン・シン著（青木薫訳）：「暗号解読」，（新潮社　2001）．
3）結城　浩著：「暗号技術入門」，（ソフトバンク・パブリッシング2003）．
4）ブルース・シュナイアー著（山形浩生監訳）：「暗号技術大全」，（ソフトバンク・パブリッシング2004）．
5）高井　信勝：「MATLAB入門」(増補版)，(工学社,2002)．
6）日経BPデジタル大辞典（2000－2001年版）．
7）高井　信勝：ディジタルホログラフィとその暗号化技術への応用，光技術コンタクト，第42巻，第6号，283－291（2004）．
8）高井信勝：拡散型ディジタルホログラフィにおける再生像の誤差評価，北海学園大学工学部研究報告，第31号，pp.87－100（2004）．