

目 次

1	メディアと数理への導入	1
1.1	この講義の目的	1
1.2	文字コード	1
1.3	ディレクトリ構造とファイルの拡張子	2
2	ショートカットと正規表現	3
2.1	ショートカット	3
2.2	正規表現	3
3	通信の暗号化とホームページ	7
3.1	sftp によるデータの送受信	7
3.2	HyperText Markup Language	8
3.3	フォント, 画像の種類	10
3.4	画像ファイルフォーマット	10
3.5	光の三原色と色の三原色	11
3.6	画素, ドット, ピクセル	12
4	ホームページの作成	13
4.1	ホームページを作るにあたって	13
4.2	ホームページビルダー 17 を用いた作成方法	14
5	ホームページの作成 (確認)	15
5.1	正しくページが表示されない場合	15
5.2	ページが表示された場合	16
6	数式, 図形の表現 I	17
6.1	\TeX (テフ, テック), \LaTeX (ラテフ, ラテック, etc)	17
6.2	実際に使用してみる	17
6.3	$\text{\textbackslash begin\{document\}}$ と $\text{\textbackslash end\{document\}}$ の間	19
6.4	印刷	20
7	数式, 図形の表現 II	21
7.1	\TeX の命令と注意事項	21
7.2	\TeX の命令と種類	22
8	数式, 図形の表現 III	29
8.1	\TeX 図形	29
8.2	画像ファイルの貼り付け	31
9	数式, 図形の表現 IV	33
9.1	Beamer とは	33
9.2	Beamer のコマンド	34
9.3	二段組	36

10 Geogebra I	37
10.1 GeoGebra とは	37
10.2 使い方	38
11 Geogebra II	41
11.1 軌跡	41
11.2 微分, 積分	42
11.3 関数の微分と接線	42
11.4 マクローリン展開	43
11.5 おまけ	44
12 Scratch I	45
12.1 Scratch とは	45
12.2 プログラム作成	45
13 Scratch II	49
13.1 変数	49
13.2 リスト	49
13.3 ソート	50
14 Scratch III	51
14.1 フローチャート	51
15 まとめの課題	53

1 メディアと数理への導入

1.1 この講義の目的

1. 情報化社会で必要となる知識を身につけ、活用させる能力を修得すること。

コンピューターを利用したマルチメディア表現について学習し、ホームページの作成やプレゼンテーションソフトを用いた”表現”を行う。

2. 中学・高校の教員になって役立つ情報を得ること。

ソフトウェアを利用した簡単な図形処理と画像処理について学習し、実習を行う。更に、LaTeX を用いて数式・文書と図形・画像の表現を行う。

3. 2 年次以降の講義で必要となる予備知識を身につけること。

今後学習するプログラミング言語の導入として、簡単なプログラミングの仕組みについての実習を行う。

♡ 注意! この講義は実習がメインとなる。また、課題は、計算機室のパソコンからメールで提出したり、ファイル転送で行うため、ユーザ名 (OUS-ID) とパスワードが必要である。

1.2 文字コード

コンピュータで文字を扱う場合、文字そのものを扱うのではなく、文字を表す”文字コード”を用いる。例えば、1Byte で表すことが出来るものとして、ASCII コードと呼ばれる文字コード (0 から 127 まで) があり、A が 65, B が 66, ..., Z が 90 で、a が 97, b が 98, ... となっている。

	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
10	LF*	VT	FF*	CR	SO	SI	DLE	DC1	DC2	DC3
20	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
30	RS	US	SP	!	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	'	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	DEL		

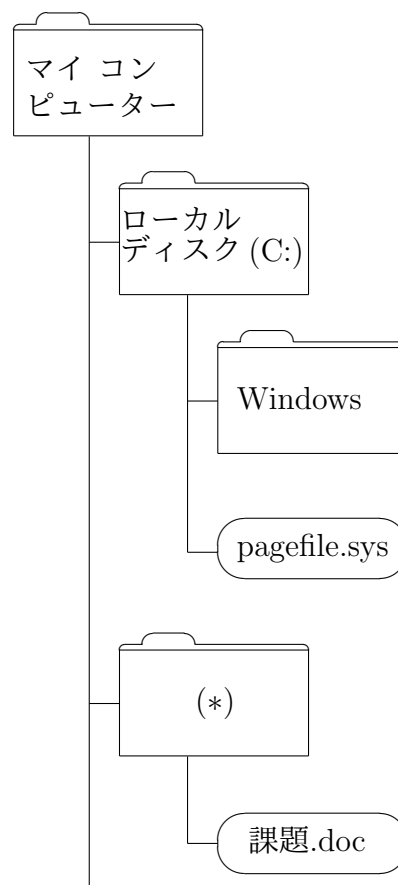
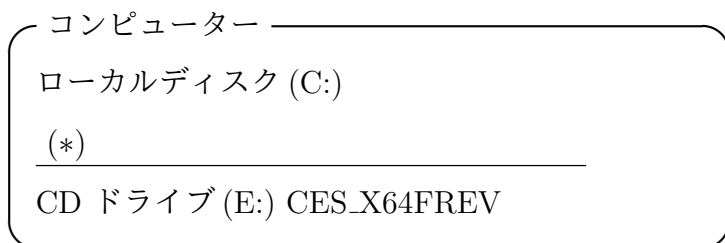
♡ 注意! 92 の \ (バックスラッシュ) は日本では ¥ が用いられる。

1.3 ディレクトリ構造とファイルの拡張子

コンピュータにおいて、ファイルの管理はディレクトリ (≡ フォルダ) の階層構造で行われている。

例えば、Windows パソコンなら右図のような階層構造をしている (一部のみ掲載)。マイコンピュータや、ローカルディスク (C:) というディレクトリがあり、それぞれのディレクトリには (サブ) ディレクトリやファイルが収納されている。

応用数学科計算機室の Windows 7 では、エクスプローラーから確認すると”コンピュータ”の下層ディレクトリとして、



があることが解る。今後、

『ローカルディスク D』, 『D ドライブ』, 『ボリューム D』

と言ったりすることがあるが、いずれもここにある (*) のことをさしている。

右の図にあるファイル pagefile.sys の . 以降をファイルの拡張子という。したがって、課題.doc の拡張子は doc である。

拡張子には”ファイルの種類を識別する”という重要な役割がある。拡張子の例として

.sys .exe .com .bat .txt .html .doc .jpg .c .JAVA .mp4 .wma .tex

などなど沢山ある。

課題 1-1. 上記の (*) に当てはまる名称を mori@xmath.ous.ac.jp 宛てにメールで送ること。その際、メールの件名は ”課題_1-1_学生番号_氏名” とし、メールの本文に (*) を記入する。

また、英数文字、記号は必ず半角文字にすること。ここの_は半角スペースを意味している。

♠ **補足** メールを送信アドレスは OUS-ID で作成したアドレスに限る。携帯やフリーメールから送信した場合、未提出となることがある。(フィルタにかからない。)

本日の課題の締め切りは、 **6 月 18 日 (火)13:00** とする。

2 ショートカットと正規表現

2.1 ショートカット

パソコンでの作業でマウスを使う作業は多い。しかし、文章を入力しているときにマウスに手を持って行くのは手間である。そこで、いくつか便利なショートカットキーを紹介しておく。

Shift + ↑ ↓ ← →	文字の”センタク”
Ctrl + a	表示されているモノ全てを”センタク”
Ctrl + c	”センタク”したものを”キオク”する
Ctrl + s	編集集中のファイルを上書き保存する
Ctrl + v	”キオク”したものを貼り付ける
Ctrl + x	”センタク”したものを削除して”キオク”する
Ctrl + z	アンドゥ(取り消し)をする
Ctrl + Tab	タブの切り替え
Alt + Tab	ウィンドウの切り替え
Alt + F4	ウィンドウを閉じる
Ctrl + r	置換のウィンドウを開く
Ctrl + f	検索のウィンドウを開く

IME(日本語入力するとき)のショートカット

Ctrl + i	カタカナ	Ctrl + u または F6	ひらがな
Ctrl + t または F10	半角英数	Ctrl + o	半角
Ctrl + p	英数	F7	全角カタカナ
F8	半角カタカナ	F9	全角英数

以上は一例である。

2.2 正規表現

正規表現を用いると検索、置換が容易に行える。詳しく説明することは避けるが、こういった方法があることを知っておけば、必要となったとき調べることが出来るであろう。

正規表現を用いて検索、置換を行う場合に特殊な文字として **メタ文字** というものがある。検索用と置換用では使用が異なるものが多いので、検索用の一部を紹介しておく。

検索用メタ文字 (の一部)

.	任意の 1 文字
*	直前のパターンの 0 回以上繰り返し (最長一致)
+	直前のパターンの 1 回以上繰り返し (最長一致)
?	直前のパターンの 0~1 回繰り返し (最長一致)
*?	直前のパターンの 0 回以上繰り返し (最短一致)
+?	直前のパターンの 1 回以上繰り返し (最短一致)
??	直前のパターンの 0~1 回繰り返し (最短一致)
A 列 B 列	の左右の文字列のいずれか (A 列または B 列)
[文字列]	文字列のいずれか 1 文字 (例: [abc] や [0-9])
(文字列)	置換後へ継承。継承する場合は ¥1 などとする。(例 2 - 2)
¥	直後のメタ文字をエスケープする (直後の文字を正規表現のメタ文字として扱わないことを指定。 例えば、() や [] なら ¥(¥) や ¥[¥] のようにする)
¥n	改行文字
¥d	すべての半角数字
¥D	半角数字以外すべて
¥w	すべての半角英数字とアンダースコア
¥W	半角英数字とアンダースコア以外すべて
¥l	半角英小文字すべて
¥L	半角英小文字以外すべて (英大文字、数字、全角文字など)
¥u	半角英大文字すべて
¥U	半角英大文字以外すべて (英小文字、数字、全角文字など)
¥n	改行文字
¥s	空白 (スペース)

♡ 注意! エディタによって正規表現を使える、使えないがあり、使える場合も形式が異なっていることがある。ここでは、今後使用する TeXworks を使用することを前提として進める。

♡ 注意! 日本語の扱いはエディタによって異なるが、正規表現では扱えないと考えておいた方がよい。

例 2-1. 正規表現を用いた置換の例 1

A regular expression, regex or regexp is, in theoretical computer science and formal language theory, a sequence of characters that define a search pattern. Usually this pattern is then used by string searching algorithms for "find" or "find and replace" operations on strings.

という文章に対して、正規表現の置換で `¥s..¥s` を `_ (__) _` に変換させると

A regular expression, regex (__) regexp is, (__) theoretical computer science and formal language theory, a sequence (__) characters that define a search pattern. Usually this pattern (__) then used (__) string searching algorithms for "find" (__) "find and replace" operations (__) strings.

となる。これは空白と空白の間に 2 文字あるものを見つけ変換している。従って、is, が変換されていないことが解る。(_ は半角スペースを意味する。)

例 2-2. 正規表現を用いた置換の例 2

one, two, three, four, five, six, seven, eight, nine, ten.

に対して、`¥s...([,.])` を `_ ___ ¥1` に変換すると

___, ___, three, four, five, ___, seven, eight, nine, ___.

となる。これは空白 '¥s' に続く 3 文字があり、その後に , または . があるものを検索している。また、() で囲むことによって置換した時に継承している。すなわち ¥1 は検索でヒットした文字が、なら , にし、検索でヒットした文字が . なら . を表示している。

例 2-3. 正規表現を用いた置換の例 3

$f(x,y)=6x+y-3$, $g(x,y)=x+y$ に対して、 $g(0,0)=-3$, $g(1,0)=3$, $g(0,1)=-2$, $g(1,1)=4$

と入力した時点で、後半は g でなく f であることに気付いた。ここで、`g¥((¥d),(¥d)¥)` を `f(¥1,¥2)` に置換すると

$f(x,y)=6x+y-3$, $g(x,y)=x+y$ に対して、 $f(0,0)=-3$, $f(1,0)=3$, $f(0,1)=-2$, $f(1,1)=4$

となる。¥d は数字なので、x や y は (すなわち $g(x,y)$ は) ヒットしない。また、() はメタ文字なので、検索するためには ¥ が必要となる。継承するものが 2 つあるので ¥1 と ¥2 を使う。

数字が 2 桁以上のものを含む場合は ¥d+ とする。したがって、上記の場合 $g(10,20)$ などはヒットしない。

課題 2-1. 以下の作業を TeXworks で作業せよ。(ショートカットと通常の置換)

(1) 以下の文字列を作成せよ。

```
abcdefghijklmnopqrstuvwxyz  
bcdefghijklmnopqrstuvwxyz  
cdefghijklmnopqrstuvwxyzab  
defghijklmnopqrstuvwxyzabc  
efghijklmnopqrstuvwxyzabcd  
fghijklmnopqrstuvwxyzabcde  
ghijklmnopqrstuvwxyzabcdef  
hijklmnopqrstuvwxyzabcdefg  
ijklmnopqrstuvwxyzabcdefgh  
jklmnopqrstuvwxyzabcdefghi  
klmnopqrstuvwxyzabcdefghij  
lmnopqrstuvwxyzabcdefghijk  
mnopqrstuvwxyzabcdefghijkl  
nopqrstuvwxyzabcdefghijklm  
opqrstuvwxyzabcdefghijlmn
```

(2) (1) で作成した文字列のうちで、i, j, k を I, J, K に変換せよ。

(3) (2) で作成した文字列のうちで、(改行を含まず) 連続する defg を DEFG に変換せよ。

課題 2-2. <https://www.xmath.ous.ac.jp/~mori/sub2.html> から kadai2-2.txt を表示 (ダウンロード) し、TeXworks にて以下の編集をせよ。

(1) 行頭にある行番号に従って並び変えよ。

(2) 同じ行番号の行は削除せよ。

(3) 行番号を [01] のように、角カッコを追加する。(正規表現を使う)

(4) カッコ 「, 」 を ” に変換せよ。(正規表現を使う)

応用課題. <https://www.xmath.ous.ac.jp/~mori/sub2.html> から kadai2-3.txt を表示 (ダウンロード) し、下のように各行で昇順になるように並べ替えよ。

```
0011111222223333333333444455555666677778888899999999  
000000111222222233334444445556666677778888888999999  
00000011112222222333334444455555566667788888888999
```

(以下略)

課題 2-1. と 2-2. が出来た人は、その画面を演習補助の先輩に見せて確認してもらい、間違いが無ければ名簿に課題番号と学生番号を記入すること。

講義外で行った人は保存しておいて、次回の講義のときに見せること。

本日の課題の締め切りは、6 月 21 日 (金) 講義終了時間とする。

3.2 HyperText Markup Language

ウェブ上の文書を記述するためのマークアップ言語である。文章の中に記述することでさまざまな機能を記述設定することができる。(Wiki ペディア より)

例 3-1. HTML 形式のソースファイルの内容

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML lang="ja">
  <HEAD>
    <META http-equiv="content-type" content="text/html; charset=UTF-8">
    <LINK rev="made" href="mailto:mail@example.com">
    <TITLE lang="en">HyperText Markup Language - Wikipedia</TITLE>
  </HEAD>
  <BODY>
    <DIV>
      <H1 lang="en">HyperText Markup Language</H1>
      <P>HTML は、<A href="http://ja.wikipedia.org/wiki/SGML">SGML</A>
        アプリケーションの一つで、ハイパーテキストを利用してワールド
(中略)
        ハイパーテキストによる HTML 間の連携が可能である。</P>
    </DIV>
  </BODY>
</HTML>
```

ホームページ (ソースファイル) を作成するとき、命令 (要素やタグ) を全て覚える必要はなく、最低限の命令のみ覚えておき、後はその都度インターネットや書籍で調べればよい。

また、最近では要素やタグを意識せずワード感覚でホームページを作成可能なソフトが多く存在している。今回は、基礎的な要素やタグのみをインターネットで調べ、編集を行う。

課題 3-1. <https://www.xmath.ous.ac.jp/~mori/sub2.html> からサンプルファイルをメモ帳に写し、kadai.html という名前で、保存する。その後、次の (1)~(4) を行うこと。

- (1) ”第 3 回課題” の文字を (左右の) 中心に配置し、フォントサイズを今より 2 大きくする。
- (2) ”今日の講義タイトル” の文字を赤色にし、”通信の暗号化とホームページ” を次の行に移す。
- (3) 理解度チェックの中に、”WinSCP の使用” の下に ”HTML” を追加する。
- (4) それぞれの ”チェック” を適当に変更し、”2019 年 1 月 1 日” を作成した日にちにちに変更する。

課題 3-2. WinSCP で ホスト 150.55.84.12 にアクセスし、新規フォルダ public_html を作成し、その中に課題 3-1. で作成したファイル kadai.html を転送すること。その後、ブラウザから <https://www.xmath.ous.ac.jp/~s00m000/kadai.html> を表示して確認する。

(s00m000 は各自の学生番号、イニシャルは付かない)

本日の課題の締め切りは、6 月 25 日 (火) 講義終了時間とする。

方法

[I] まず、 <https://www.xmath.ous.ac.jp/~mori/sub2.html> からサンプルファイルを開き、ソースコードを表示する。その内容をメモ帳に貼り付けて 課題 (1)~(4) を行った後、kadai.html という名前で保存する。

メモ帳での保存方法

[ファイル] → [名前を付けて保存]

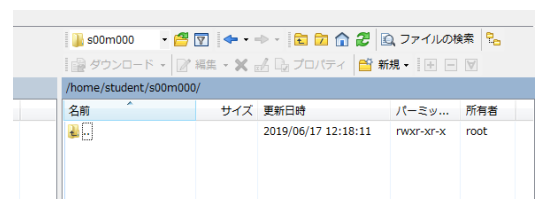
保存フォルダを選び、ウインドウ下の [ファイルの種類] をすべてのファイル (*.*) に変更し、ファイル名を kadai.html として保存する。

[II] ファイルの編集と保存が出来たら、WinSCP を起動し、 150.55.84.12 と接続する。

[III] 接続が確立すると、右側のウインドの説明が

`/home/student/s00m000/`

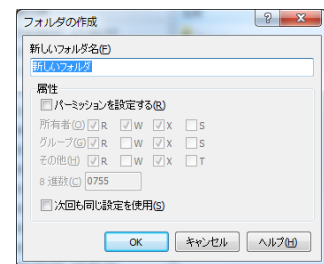
となっている。このことを確認し、右側のウインド内をクリックする。



[IV] その後、**[F7]**キーを押すと右図のようなウインドが開かれ、フォルダの作成画面になる。

ここで、”新しいフォルダ”と書かれている部分を削除し、

`public.html`



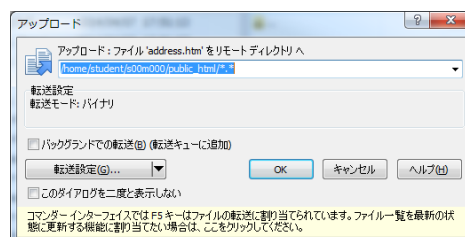
と入力し、OK ボタンを押す。

作成された public.html をダブルクリックすると、右側のウインドウの説明が

`/home/student/s00m000/public.html/`

に変わる。

[V] 左側のウインドウから、作成したファイル kadai.html を探し、選択した状態で**[F5]**キーを押すと

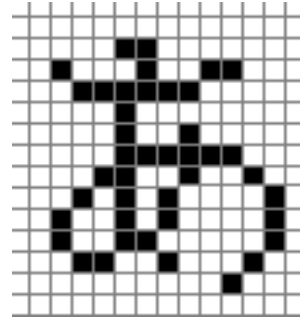


が表示されるので、OK ボタンを押す。エラーが無ければ、WinSCP を終了する。

[VI] 正しく作業が出来ていれば、ブラウザから 確認が出来る。

3.3 フォント、画像の種類

パソコン上で扱う文字は、大きく分けて2種類の文字(フォント)がある。**あ**と**あ**である。1つめは、ビットマップフォント(ラスタ形式)と呼ばれるもので、ドットの組合せで文字を表現したフォントである。コンピュータの初期には記憶容量の節約、負荷軽減などのため(予め用意された)ビットマップフォントを利用していた。このフォントはドットで表現されているため、拡大すると荒くなる欠点がある。近年のコンピュータはモニターの表示可能な(後述の)ピクセル数が増え、文字を拡大する機会が増えた(1文字に必要なドット数が増えた)ため、あまり使われなくなってきた。ただし、現在でも文字が小さい場合などはビットマップフォントを使用することも多い。



この欠点を無くすため近年使われているフォントとして、スケーラブルフォント(ベクター形式)がある。これは、文字の線の位置や形、長さなどで形を作るため、拡大や縮小をしても、ビットマップフォントのように文字が荒くなることはない。スケーラブルフォントの種類として、ストロークフォントやアウトラインフォントなどがある。

1行目の **あ** は左側がビットマップフォントで、右側がスケーラブルフォントである。

3.4 画像ファイルフォーマット

画像ファイルもフォントと同様にビットマップ画像(ラスタ形式)とジャギーのないベクター画像(ベクター形式)に分類される。ベクター画像は図形情報の集まりなので、画像を縮小、拡大、変形しても輪郭は乱れない。そのため、色数の少ない図面やイラストなどに向いているが、写真などの複雑な画像には向かない。また、表示のたびに計算を行うことによってパソコンの処理に負担がかかり、複雑な図形の場合は表示に時間がかかってしまう場合もある。

ビットマップ画像は個々の色の点を並べるので、写真や風景などの複雑な画像に適している。ただし、拡大すると小さな点が目立ってしまい画像が雑に見えるという欠点がある。

また、ビットマップ画像は画像に使われているドットのデータを全て保存するため、ファイルサイズが大きくなりがちであるが、ベクター画像はアンカー座標の数値などしか記録していないので、ファイルサイズを小さくしやすいというメリットがある。

ビットマップ画像には、可逆圧縮(無圧縮も含む)と、非可逆圧縮に大別できる。可逆圧縮(無圧縮)として身近なのはBMP(Windows bitmap)、非可逆圧縮として身近なのはJPEGやGIFがある。また、色の深度やアニメーションの可否などの違いでさまざまな形式が存在する。

♠ 補足

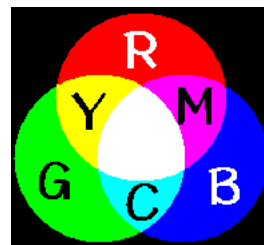
可逆圧縮とは、データを復元したときに、完全に元にもどる圧縮方法をいう。通常、意味のあるデータには、現れる符号に偏りすなわち規則性、冗長性が存在する。そのため多くの可逆圧縮では出現頻度の高いものに短い符号を、出現頻度の低いものに長い符号を与えることで、平均符号長を短くする方法がとられている。また区間的に分けてそれぞれ符号を変える、より長い符号に対して前述の処理を行うなどの方法で、可逆圧縮間でも圧縮率や展開速度に差が出る。

非可逆圧縮とは、データを復元したときに、完全には元にもどらない圧縮方法をいう。多くの非可逆圧縮では人があまり強く認識しない成分を削除することでデータを圧縮する方法がとられている。たとえば人は大きな音と小さな音を聞いた場合、小さな音をあまり認識できないし、画像に対しても小さな色の変化は認識されない。このためデータをフーリエ変換（あるいは離散コサイン変換などフーリエ変換の一種）し、高周波成分や低振幅成分を削除してしまっても、受け手に与える印象の変化に大きな差は現れない。当然削除する範囲が多ければ、元データとの差異は大きくなり、違いに気づく人も増える。画像のサイズを小さくする、動画のフレームレートを下げるなども一種の非可逆圧縮と言える。

3.5 光の三原色と色の三原色

光の三原色はR(赤 (Red))、G(緑 (Green))、B(青 (Blue))を指す。色を混ぜ合わせるにつれて、色が明るくなり、三色の色を加えていくと白になる。もちろん何も加えなければ黒である。

光の場合にはこの3つの色を使うと、ほぼすべての色が再現できる。この3色を、光の三原色という。ブラウン管、液晶モニタの発光体には、この三原色が使用されている。下の表は、光の三原色を混ぜたときにできる色の様子である。



表示色	R	G	B
BK(黒色)	0(0)	0(0)	0(0)
R(赤色)	255(FF)	0(0)	0(0)
Y(黄色)	255(FF)	255(FF)	0(0)
G(緑色)	0(0)	255(FF)	0(0)
C(シアン)	0(0)	255(FF)	255(FF)
B(青色)	0(0)	0(0)	255(FF)
M(マゼンタ)	255(FF)	0(0)	255(FF)
W(白色)	255(FF)	255(FF)	255(FF)

光の三原色に対して、色の三原色では、3つの基本色を黄色、マゼンタ、シアンとし、いろを混ぜわるにつれて、色が暗くなるように作られている。インクジェットプリンタで用いられる方法である。原理的には基本色3色を混ぜると黒になるが、実際には黒とならないため、インクジェットプリンタには黒専用のインクが備わっている。

下の写真は上が原画、中段は左から R,G,B に分解したもの、下段は中段の2つを加えたもので、左から Y,C,M に分解したものである。(モノクロ印刷では解らないと思う。)



3.6 画素, ドット, ピクセル

一般的なディスプレイでは画素(ドット, ピクセル)と呼ばれる小さな点を縦横に並べ、それらの点の輝度を別々に制御することで画面を表示する。1つの画素には赤・緑・青の3つのサブ画素(サブ・ドット)があり、それぞれの輝度を制御することで多様な色を生み出している。

従って同一の表示サイズで比較する場合、画素数が多いほど細やかで綺麗な表示が可能となる。つまり、ディスプレイ上に存在する画素数(解像度)によって表示の精細度が定められる。

近年のテレビやディスプレイなら Full-HD(1920×1200, 約 200 万画素), WQXGA(3200×2400, 約 760 万画素), 8K UHD TV(7680×4320, 約 3300 万画素) など、高解像度のものが増えている。

このような場合、ディスプレイによって(水平方向に)同じ 10cm の幅でもドットの数異なる。そこで、1 インチ(=25.4mm) 当たりの画素数を dpi(dots per inch) や ppi(pixels per inch) で表示することでディスプレイの性能を表している。

4 ホームページの作成

4.1 ホームページを作るにあたって

♣ 注意点

ホームページを作るにあたり、気をつけなければならない点はいくつかある。ホームページは、メールや SNS と違い、不特定多数の人が見るものである。まず、著作権、肖像権などは注意が必要である。メールに添付の気持ちで画像を載せてはならない。

また、個人情報の取り扱いは、自分のことだけでなく、友人など他人の情報も気軽に載せてはならない。

前回の講義で述べたように、画像を載せる場合はサイズや形式に注意しないといけない。特定の端末でしか見られない画像や、無駄にサイズの大きい画像をトップに張るのは良くない。

どうしてもそのような画像を置きたい場合は、いろんなホームページを参考にすると良い。

♣ 準備

まずは、サーバーの確認をする。ホームページのデータは **www** サーバーと言われるサーバーで管理されている。したがって、各自のパソコンで作成しただけでは一般公開されない。また、サーバーによっては機能を限定している場合も多い。例えば、今回は CGI が使用できない。そういった **www** サーバーの機能を確認しておかなければ、作成後に閲覧できないことに気づくはめになる。

次に、何を公開するか考える。前項の注意点をふまえて公開する内容を考える。もちろん **www** サーバーで利用できる範囲で考える。デザイン等は他のホームページなどを参考にすると良い。沢山のページを見ると良いページ、悪いページも解ると思う。また、ページ数も事前にある程度考えておく方がよい。ページの作り方によっては追加や削除が手間取ることもある。

♣ 方法

ホームページを作成する場合に以下のような方法でまず html ファイルを作成する。

- (i) メモ帳のようなエディタで HTML 文章を書く
- (ii) ホームページビルダーのようなソフトを使用する
- (iii) (i),(ii) の中間のようなソフト (MS word など) で組み立てる

(i) や (iii) で作った場合は、ftp ソフト等を使用して出来たファイルをサーバーに転送しなければならない。(ii) の場合は、ファイル転送も組み込まれていることが多い。なお、この講義では (ii) の方法を用いて作成するが、ファイル転送は ftp でなく sftp を使用する。

4.2 ホームページビルダー 17 を用いた作成方法

ホームページビルダーを用いる場合でも、いろいろな方法でいろいろなページを作成できる。ここでは、シンプルな作り方を紹介する。まず、ホームページビルダー 17 が起動したら

[ファイル]

→ [ベーシックテンプレートから新規作成]

→ [サンプルテンプレートから新規作成]

の順で選択する。1 ページサンプルでも、複数ページサンプルでも良いが、今回は 1 ページを選択する。

ページの種類は [PC ページ]、テーマとテンプレートはお好みで。OK ボタンを押して、ホームページを作成する。

♠ **重要** ページが完成したら [ファイル] → [名前をつけて保存] と選択。次に [保存する場所] を D ドライブの適当なフォルダに変更し¹、ファイル名を index.html に変更する。

”素材ファイルをコピーして保存”が表示されたら [保存] を選ぶ。

保存が終了したら、WinSCP で作成したファイルをサーバに転送する。

その際、必ず ファイル名が index.html になっていることを確認し、画像等のファイルも転送することを忘れないように。

課題 4-1. 4.1 節をよく読んで、自分のホームページを作成せよ。公開する内容は各自にまかせる。

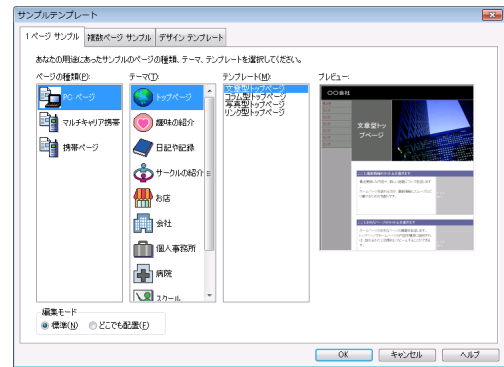
今回の課題は 7 月 26 日 17:00 以降にブラウザで私 (森) が確認する。したがって、

この 2 点を間違えると課題未提出となる。

課題 4-2. ブラウザから <https://www.xmath.ous.ac.jp/~s00m000/> を表示して確認する。
(s00m000 は各自の学生番号)

課題 4-1., 4-2. の締め切りは、**7 月 26 日 (金) 17:00** である。

次回もホームページの作成 (続き)。



¹(D:) にしなければ、次回起動時消えている!! フォルダは、新しいフォルダを作成した方が良い。

5 ホームページの作成 (確認)

5.1 正しくページが表示されない場合

[I] Not Found と表示される場合

[1] ファイル名は正しいか？

トップページのファイル名が `index.html` になっているか。

転送先のフォルダ名 (ディレクトリ名) は `public_html` になっているか。

転送先の `index.html` がフォルダになっていないか。

間違えていれば、ファイル名、フォルダ名を変更する。もしくは、一度削除してやり直す。

[2] URL を間違えていないか？

`~s00m000` の `~` と `^` を間違えていないか。学生番号の後にイニシャルを入れていないか。

[3] 転送先は正しく設定されているか？

`public_html` が見えている状態ではなく、ダブルクリックして右側のウィンドウの説明が

`/home/student/s00m000/public_html/`

になっているか確認する。

[II] 画像ファイルが表示されない場合

[1] 貼り付けで作成していないか？

画像を”張り付け”で表示させた場合、上手く転送出来ない。テンプレート以外の画像ファイルを使用する場合、必ずローカルディスクに保存し、それを参照させる形で表示させること。

[III] 転送したいファイルが WinSCP で表示されない場合

[1] 使用できない文字を使っていないか？

フォルダ名、ファイル名に使用できない文字 [機種依存文字やスペース (空白文字)]、特にスペースを入れている場合がよく見受けられる。

また、使える日本語 (全角文字) と使えない (全角文字) が有るため、アルファベットに変更して再送してみる。もちろん半角カタカナも使用しない。

[IV] なんとなくしっくりこない場合

[1] ホームページビルダーの使い方に慣れているか？

その場合は、ソースファイルを編集する。

・改行に空行がある場合、`
` を使用する。

・フォントのサイズは、`` を、`` のように相対的サイズに変える。など。

5.2 ページが表示された場合

作成したページを転送後、ブラウザから表示した場合、以下を確認する。

[1] リンク切れが無い確認する。

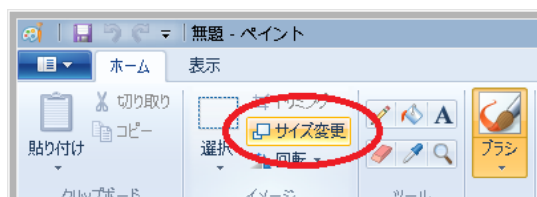
例えば [リンク] と表示されている場所をクリックしたら Not Found と表示されることがないか確認しておくこと。また、画像が表示されない場所がないか、リンク先が Not Found に成らないかも確認しておくこと。

[2] 誤字, 脱字が無い確認する。

日本語の変換のミスや、編集集中に文字を消しすぎ、消し忘れが多い。英語のつづりミスにも注意すること。

[3] 画像サイズの確認をする。

画像を表示する場合、表示で拡大縮小するのではなく、そのサイズに合った画像を作成し、貼り付けることが必要である。画像の縮小はペイントで簡単にできる。



画像の形式は適切なものを選択するように。jpeg 形式の画像をペイントで縮小し、BMP 形式でホームページで使用するのは避けるように。

[4] 再度、肖像権, 著作権など確認する。

念のため、友人に確認してもらうとよい。サーバにアップした後なら世界中の誰からでもチェックを受けられる。気になる場合は、応数計算機室の本棚に著作権に関する本があるので、そこらでもチェックするとよい。

♡ 注意! これら [1] から [4] も採点対象となる。

なお、7 月 26 日 (金)17:00 が締切で、それ以降も編集 (上書きアップロードが) 可能な状態ですが、編集しないこと。止むを得ず変更する場合は、必ず連絡をするように。

次回から L^AT_EX について 4 回の講義 (実習) を行う。

6 数式, 図形の表現 I

6.1 \TeX (テフ, テック), \LaTeX (ラテフ, ラテック, etc)

\TeX とは、数式の処理に優れる組版ソフトウェアであり、その \TeX を使ってもっと簡単に論文やレポートを作成出来るように開発されたのが \LaTeX や $\text{\LaTeX} 2_{\epsilon}$ である。

HTML 文章と同じように、命令を用いてページを作成していく。沢山の命令があるが、基本的なものさえ覚えておけば、後は必要なときに調べればよい。

例 6-1. (\LaTeX ソースファイル) —

```
%documentclass[dvipdfmx,a4j,12pt]{jarticle}
%ここがプリアンブル
%begin{document}
{\bf 第1回 集合の基礎、数学で使う記号}
%begin{flushright}
学生番号 \underline{\hspace{40mm}} 氏名 \underline{\hspace{50mm}}
%end{flushright}
{\bf 問題 1-1.} 以下の集合 $A, B, C$ に対して、(1) から (4) に答えよ。

(略)

%end{document}
```

例 6-2. (例 6-1 を変換して表示した例.) —

第 1 回 集合の基礎、数学で使う記号

学生番号_____ 氏名_____

問題 1-1. 以下の集合 A, B, C に対して、(1) から (4) に答えよ。

(略)

\TeX では細かい設定を プリアンブル (`%documentclass` と `%begin{document}` の間) に指定出来るが、慣れるまでは `%begin{document}` と `%end{document}` の間の部分のみを編集していく。

6.2 実際に使用してみる

\TeX は話を聞くより実際に使用した方が習得出来る。(細かいことは後述。)

\TeX では作成したソースファイルを DVI ファイルや PDF ファイルに変換して、初めて文章、数式や表などが表示される。この講義 (実習) では、TeXworks というランチャー兼エディタを使用して \TeX のファイル作成, PDF の表示を行う。

デスクトップ上のTeXworksを起動すると、黒い画面(コマンドプロンプト)が表示されるが、少し待つと起動する。何も入力されていない新規のファイルが表示されたら、

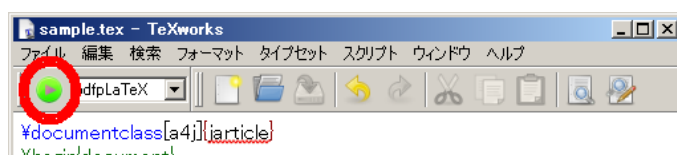
```
%documentclass[a4j]{jarticle}
%begin{document}
吾輩は猫である。名前は真田奈衣。
```


```
どこで生れたかとうんを見当がつかぬ。
何でも薄暗いじめじめした所で
ニャーニャー泣いていた事だけは記憶している。
吾輩はここで始めて人間というものを見た。
%end{document}
```

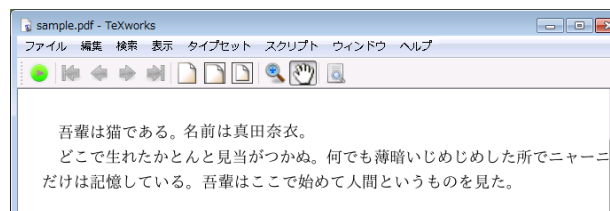
と入力する。入力が終わったら、[ファイル] から [名前を付けて保存] を選び、適切なフォルダを選択する。フォルダを選んだら、ファイル名を入力する。


その際、ファイル名(フォルダ名)は 日本語, 空白入り, 記号は避けた方が良いでしょう²、sample.tex や HandM01.tex などとする。


[保存] 後、左上のアイコン(下図の丸で囲っているアイコン)をクリックする。



アイコンが  に変わり、少し待ち、エラーが無ければ元に戻り、右側に pdf ファイルが表示される。



アイコンが  のままならエラーメッセージが

ログの表示に表示されている。エラーの場合は、 をクリックして変換を中止し、エラーメッセージを元に修正し、再度アイコンをクリックする。

エラーとして多いもの

よく見かけるミスは、全角と半角の入力ミスと $\$$, $\{$, $\}$ の数が対応していないミスの2通り。例えば、 $\$$ で囲むべき $y=x^2+3x\$$ が $y=x^2+3x$ や $y=x^2+3x \$$ となっていた場合、

```
! Missing $ inserted.
<inserted text>
$
```

1.13

のようなエラーメッセージがログの表示に表示される。

²使用可能なものと不可能なものがある。なので使わない方がよい

6.3 $\begin{document}$ と $\end{document}$ の間

★文字入力 何もせず文字を入力したとき、日本語や English はそのまま表示される。ただし、機種依存文字は使えない。例えば、全角入力した I は表示されない。I は半角大文字の I を使う。

太文字 日本語 や イタリック English は命令が必要で、太文字なら $\textbf{日本語}$ と入力する。

★記号入力 そのままでは入力できない記号もある。例えば、\$ や & などは命令文に使われるため、表示には一手間かかる。

★数式 I 数式の変数などはそのまま x と入力するとフォントが文中のものになるので、数式を表す命令 (\$ で囲む、 x) を使うことによって、 x と表示される。ちなみに、 x^2 は、 x^2 と入力し、 $\frac{1}{2}$ は $\frac{1}{2}$ と入力する。その他、 π や θ は、 π や θ と入力する。

また数式を中央に表示する場合は $\$$ で囲む³。例えば、 $y = x^2 + 3$ を数式立てにする場合、 $y=x^2+3$ とする。(以下のようにになる。)

$$y = x^2 + 3$$

♡ 注意! 文字と数式の扱いに注意すること! た例えば、 $x+y$ でなく、 $x+y$ とする。

また、数式として繋げて書く場合、例えば $2x - \frac{1}{2}y^2$ を、 $2x - \frac{1}{2}y^2$ のように書くのではなく、 $2x - \frac{1}{2}y^2$ とすること。

★改行 改行も一工夫必要で、例えば L^AT_EX ソースファイル内に

岡山県岡山市北区理大町 1 - 1
岡山理科大学

と書いても

岡山県岡山市北区理大町 1 - 1 岡山理科大学

と表示される。改行したい場合は、空行を入れる。ただし、1 つ以上入れても空行は出来ない。したがって、

岡山県岡山市北区理大町 1 - 1

岡山理科大学

とすると改行され

岡山県岡山市北区理大町 1 - 1

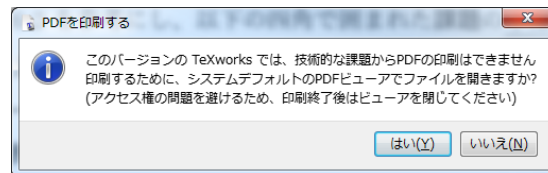
岡山理科大学

となる。空行を入りたい場合は \vspace 等を使う。

³別行立て数式、数式立てと言う。今後は数式立てと呼び、\$ で囲むことを数式モードと呼ぶことにする

6.4 印刷

出来た PDF ファイルを印刷する場合は、PDF ファイルが表示されている (右側) ウィンドウの [ファイル] から、[PDF を印刷する] を選ぶ。



そのときに、右図のようなメッセージが出るので [OK] を押す。

その後、PDF ビューアーが開かれるので、[ファイル] から [印刷] を選択し印刷が出来る。

課題 6-1. 例 6-1, 例 6-2 を参考にし、以下の四角で囲まれた課題の中身が表示されるような \LaTeX ファイルを作成せよ。ただし、改行位置については気にしなくてよい。

課題

第 6 回 数式, 図形の表現 I

学生番号 _____ 氏名 _____ 得点 _____

問題 1-1. 整式 $P(x) = ax^3 + bx^2 - 8x - 7$ が $x + 1$ で割り切れ, $x - 2$ で割ったときの余りが -3 となるように、定数 a, b の値を定めよ。

問題 1-2. 次の等式を証明せよ。

$$\frac{1}{1-x} + \frac{1}{1-y} = 1 + \frac{1-xy}{(1-x)(1-y)}$$

問題 1-3. 円周率 π の連分数展開は以下であることを示せ。

$$\pi = 3 + \frac{1}{7 + \frac{1}{15 + \dots}}$$

課題が出来た人は印刷し、学生番号と氏名を手書きで記入して提出すること。講義の後に作成した人は、次回の講義 (7月5日) のとき提出すること。

課題 6-1. の締め切りは、**7月5日 (金) の講義終了時間**である。

◇ 余談 応用数学科の卒業研究では、この \LaTeX を用いて卒業論文を書くゼミが多い。また、慣れると数式や記号の入力が楽になるので教員を目指す場合、是非習得して欲しい。

\LaTeX はフリーソフトなので自由にインターネット上からダウンロードして使用可能であるが、インストールが少々大変なので、CD-ROM が付いた書籍などを購入して使うことを薦める。

また、命令文などインターネット上で調べることも可能であるが、卒業論文に必要と思われるものはこの講義で紹介する予定である。

7 数式、図形の表現 II

7.1 T_EX の命令と注意事項

まず、今後必要となる以下のパッケージをプリアンブルに指定する。

```

\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{arydshln}
\usepackage{ascmac}

```

★章, 節 文章を書くとき、章の見出し、節の見出し、本文 ... のようなレイアウトで書く。T_EX では章や節を `\section{***}` や `\subsection{***}` で記述することによって左上のように、前後のスペース、フォントサイズの調整と番号付けを行ってくれる。このことは、章や節の追加や削除を行ったとき、自分で番号の付け替えを行わなくて良いことを意味する。

★注釈 % 以降の記述は注釈として解釈され、本文に影響を与えない。

★空白 空白を入れるには半角スペースを入れれば1つ空白が出来る。ただし、1つ以上入れても1つの空白しか入らない。例えば、”学生番号(), 氏名()” と入力しても、変換結果は ”学生番号(), 氏名()” と表示される。

1つの空白以上の空白が必要な場合は `\hspace` を使う。例えば、`\hspace{20mm}` は 2cm の空白()をあけることを意味する。同様に、`\vspace{10mm}` を使うと

のように、行をあけることも出来る。ただし、`\vspace` の前に空行を挟むこと。

また、改行後に空白が入る(この行の先頭のように)ことを避けたい場合は `\noindent` を使う。

```
\noindent
```

また、

とすると、`\noindent` の前の空白が無くなる。この場合も `\noindent` の前に空行を挟むこと。

★地の文と命令文 T_EX の命令文は `\` で始まることが多い。たとえば、L^AT_EX と表示するには、`\LaTeX` と入力している。このとき、`\LaTeX` は命令文で、それ以外は地の文である。これを、

”このとき、`\LaTeX` は命令文で、”

と続けて書くと、`\` 以降どこまでが命令文で、どこからが地の文か解らなくなる。(特に英文ならなおさら。)そこで、命令文の終わりにはスペースを1つ以上入れなければならない。

♡ 注意! 後ろに空白を必要としない命令もある。例えば、`\$` や `\&` など。これらは空白を入れると \$ 30 のようにスペースが出来て不格好になる。

7.2 T_EX の命令と種類

i) 単独で命令となるもの

例えば、先の L^AT_EX や $y = x^2 + 3$ ($\$y=x^2+3\$$) など個々で意味をなす命令となっている。また、太文字にする `\bf` も単独で意味をなすが、それ以降すべてを太文字にする命令なので、`{ }` で囲み範囲指定をする。このように`{\bf 範囲指定}`する。

ii) 引数を伴い命令となるもの

分数を表す `\frac{ }{ }` や下線を引く `\underline{ }` などは `{ }` 内にある引数に対して命令を行う。たとえば、ここに下線を引く。は、`\underline{ここ}` に下線を引く。

iii) `\begin` と `\end` で命令となるもの

例 6-1 にあった `\begin{flushright}~\end{flushright}` や、中央寄せの `\begin{center}~\end{center}` などがそうである。

★数式 II `$$` で囲む以外に数式立てする方法として `equation` 環境, `eqnarray` 環境を使用する方法がある。前者は単一行、後者は複数行の場合に使う。どちらも式番号を付けられる。例えば

```
\begin{equation}
y=x^2 - 4x + 3
\end{equation}
```

$$y = x^2 - 4x + 3 \quad (1)$$

```
\begin{eqnarray}
y&=&x^2 - 4x + 3\\
&=&(x-3)(x-1)
\end{eqnarray}
```

% = の前後に & を入れることによって次行と = 揃えする。

$$y = x^2 - 4x + 3 \quad (2)$$

$$= (x - 3)(x - 1) \quad (3)$$

```
\begin{eqnarray*}
y&=&x^2 - 4x + 3\\
&=&(x-2)^2-1
\end{eqnarray*}
```

$$y = x^2 - 4x + 3$$

$$= (x - 2)^2 - 1$$

である。式番号は自動的に通し番号になり、式番号をつけたくない場合は `\begin{ }`, `\end{ }` の `eqnarray` や `equation` の最後に `*` を付けて `eqnarray*` や `equation*` とする。

♡ 注意! 以下は数式モードか、数式立てもしくは数式 II で上げた命令内で入力すること。

★カッコ カッコのうち、(), [] はそのまま表示される。{ } は ¥{ ¥} と入力する。 $(\frac{\sqrt{2}}{2}, 1)$ のようなとき、() の高さを調節するにはそれぞれの () の前に ¥left または ¥right を付けくわえて ¥left(¥right) と入力すると、 $(\frac{\sqrt{2}}{2}, 1)$ のように調整してくれる。

★分数、連分数 分数は $\frac{\text{分子}}{\text{分母}}$ を用いる。例えば、 $\frac{1}{2}$ は $\frac{1}{2}$ とかく。このとき、分子と分母の順番に 注意すること。連分数は、分子や分母の $\{ \}$ 内にまた分数を入れればよい。例えば、 $\frac{1}{1+\frac{1}{1+\frac{1}{1+}}}$ は $\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+}}}}$ となる。

★上付き, 下付き 上付きは \sim で下付きは $_$ である。例えば、 x^n や x_0 は、 $x^{\sim n}$ や $x_{\sim 0}$ である。また、 $\{a_i\}_{i=1}^{10}$ は $\forall \{a_{\sim i}\}_{i=1}^{\sim 10}$ である。上付き, 下付きは、 $x_{\sim 0}^{\sim 2}$ など $\{ \}$ 内が1文字なら $\{ \}$ を省略($x_{\sim 0}^{\sim 2}$)できる。

★三角関数, 指数関数, 対数関数, n 乗根

\sin, \cos, \tan は $\text{\texttt{\$sin}}, \text{\texttt{\$cos}}, \text{\texttt{\$tan}}$ と書き、 \exp も $\text{\texttt{\$exp}}$ と書く。このとき $\text{\texttt{\$}}$ を忘れると \sin や \exp となる。 e^x と書く場合は、 e の書体を e とするため、 $\text{\texttt{\$mathrm{\texttt{e}}}}^{\text{\texttt{x}}}$ と書く。

平方根は`¥sqrt`を使う。例えば、 $\sqrt{x^2 + y^2}$ は`¥sqrt{x^2 + y^2}`と書き、 n 乗根は`¥sqrt[n]`を使い、 $\sqrt[3]{-8}$ は`¥sqrt[3]{-8}`と書く。

★総和, 総乗 総和と総乗はそれぞれ $\sum_{i=1}^n$, $\prod_{i=1}^n$ を使う。

たとえば、 $c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$ は `c_{ij}=\sum_{k=1}^na_{ik}b_{kj}` と書く。同じ命令文でも $\$$ で囲めば文中(数式モード)と異なり、

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}$$

となる。

★極限, 微分, 積分極限は $\lim_{x \rightarrow 0} f(x)$ と書くと、 $\lim_{x \rightarrow 0} f(x)$ と表示される。微分は $f'(x)$ と入力して $f'(x)$ と表示したり、分数を用いて $\frac{dy}{dx}$ と入力することによって $\frac{dy}{dx}$ と表示させたりする。

積分は $\int_{\text{下付}}^{\text{上付}} f(x)dx$ を使う。 $\int_0^1 f(x)dx$ ならば、 $\int_0^1 f(x)dx$ と入力する。

これらは、数式立てにすると

$$\lim_{x \rightarrow 0} f(x), \quad \frac{dy}{dx}, \quad \int_0^1 f(x) dx$$

と表示される。ちなみに、 ∞ は `¥infty` と書く。

★ベクトル, 行列ベクトルは`\vec`を使い、`\vec{a}`と書くと \vec{a} と表示される。ただし、変数が多いときは \overrightarrow{AB} では寂しいので、`\overrightarrow{AB}`と書いて \overrightarrow{AB} と表示させる。

行列の表示にはいくつかの方法があるが、汎用性の高い `array` 環境を使う⁴。使用 방법은以下の例を参考するとよい。

array 環境使用例

出力	コマンド	コメント
$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	<code>\left \begin{array}{cc} a & b \\ c & d \end{array}\right </code>	<code>\begin{array}</code> の後の <code>{cc}</code> は c (中央) で2列の配列を意味する。 <code>\</code> は改行、行の終わり
$\left(\begin{array}{cc c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{array} \right)$	<code>\left(\begin{array}{cc:c} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{array}\right)</code>	<code>{cc:c}</code> は2列目と3列目の間に点線を入れることを表す。
$\begin{array}{c cc} & a & b \\ \hline c & ac & bc \\ de & ade & bde \end{array}$	<code>\begin{array}{ c rl} & a & b \\ \hline c & ac & bc \\ de & ade & bde \end{array}</code>	<code>{ c rl}</code> は2列目と3列目の間以外に縦線を入れることを表し、1列目は中央、2列目は右、3列目は左に揃えている。 <code>\hline</code> は横線を入れる。
$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$	<code>\left[\begin{array}{ccc} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{array}\right]</code>	\cdots , \vdots , \ddots を出力するには <code>\cdots</code> , <code>\vdots</code> , <code>\ddots</code> を使う。

◇ 余談 行列や行列式を扱う場合は `pmatrix` や `vmatrix` を使うことも多い。汎用性はない。

★枠, 表 表の作成には `array` 環境以外に `table` 環境がある。こちらの使用は数式環境ではない。

単位	評価	点数	GPA
認定	S	100～90	4
	A	89～80	3
	B	79～70	2
	C	69～60	1
不認定	D	59～0	0
	E	未受験, 未資格	0

```

\begin{table}[htb]
\begin{tabular}{|l|c|c||r|}\hline
単位 & 評価 & 点数 & GPA \\\hline
認定 & S & 100～90 & 4 \\\cline{2-4}
& A & 89～80 & 3 \\\cline{2-4}
& B & 79～70 & 2 \\\cline{2-4}
& C & 69～60 & 1 \\\hline
不認定 & D & 59～0 & 0 \\\cline{2-4}
& E & 未受験, 未資格 & 0 \\\hline
\end{tabular}
\end{table}

```

ここではこの `table` 環境について 紹介のみに留め、詳しく触れないことにする。

⁴この `array` 環境も数式モード、数式立てで使う。

★その他 (ギリシャ文字) 数式モード、数式立てで使う。

大文字	コマンド	小文字	コマンド	大文字	コマンド	小文字	コマンド
A	<code>A</code>	α	<code>\alpha</code>	N	<code>N</code>	ν	<code>\nu</code>
B	<code>B</code>	β	<code>\beta</code>	Ξ	<code>\Xi</code>	ξ	<code>\xi</code>
Γ	<code>\Gamma</code>	γ	<code>\gamma</code>	O	<code>O</code>	o	<code>o</code>
Δ	<code>\Delta</code>	δ	<code>\delta</code>	Π	<code>\Pi</code>	π	<code>\pi</code>
E	<code>E</code>	ϵ	<code>\epsilon</code>	P	<code>P</code>	ρ	<code>\rho</code>
Z	<code>Z</code>	ζ	<code>\zeta</code>	Σ	<code>\Sigma</code>	σ	<code>\sigma</code>
H	<code>H</code>	η	<code>\eta</code>	T	<code>T</code>	τ	<code>\tau</code>
Θ	<code>\Theta</code>	θ	<code>\theta</code>	Υ	<code>\Upsilon</code>	υ	<code>\upsilon</code>
I	<code>I</code>	ι	<code>\iota</code>	Φ	<code>\Phi</code>	ϕ	<code>\phi</code>
K	<code>K</code>	κ	<code>\kappa</code>	X	<code>X</code>	χ	<code>\chi</code>
Λ	<code>\Lambda</code>	λ	<code>\lambda</code>	Ψ	<code>\Psi</code>	ψ	<code>\psi</code>
M	<code>M</code>	μ	<code>\mu</code>	Ω	<code>\Omega</code>	ω	<code>\omega</code>

一部のギリシャ文字の小文字には変体文字がある。

小文字	変体文字	コマンド	小文字	変体文字	コマンド
ϵ	ε	<code>\varepsilon</code>	ρ	ϱ	<code>\varrho</code>
θ	ϑ	<code>\vartheta</code>	σ	ς	<code>\varsigma</code>
π	ϖ	<code>\varpi</code>	ϕ	φ	<code>\varphi</code>

★その他 (記号 1) 数式モード、数式立てで使用する記号の例

出力	コマンド	出力	コマンド	出力	コマンド	出力	コマンド
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\leq	<code>\leqq</code>	\geq	<code>\geqq</code>
\because	<code>\because</code>	\therefore	<code>\therefore</code>	\pm	<code>\pm</code>	\times	<code>\times</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\supseteq	<code>\supseteq</code>	\subseteq	<code>\subseteq</code>
\cap	<code>\cap</code>	\cup	<code>\cup</code>	\subset	<code>\subset</code>	\supset	<code>\supset</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>	\rightarrow	<code>\rightarrow</code>	\leftarrow	<code>\leftarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftarrow	<code>\Leftarrow</code>
\rightarrow	<code>\rightarrow</code>	\mapsto	<code>\mapsto</code>	\neq	<code>\neq</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\circ	<code>\circ</code>	\bullet	<code>\bullet</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>
\emptyset	<code>\emptyset</code>	\bot	<code>\bot</code>	\angle	<code>\angle</code>	\triangle	<code>\triangle</code>

★その他 (記号 2) 通常の入力で使用する記号の例

出力	コマンド	出力	コマンド	出力	コマンド	出力	コマンド
$\#$	<code>\#</code>	$\$$	<code>\\$</code>	$\%$	<code>\%</code>	$\&$	<code>\&</code>
$-$	<code>_</code>	$\{$	<code>\{</code>	$\}$	<code>\}</code>	\S	<code>\S</code>
\pounds	<code>\pounds</code>	\oe	<code>\oe</code>	\oslash	<code>\oslash</code>	\O	<code>\O</code>
\TeX	<code>\TeX</code>	\LaTeX	<code>\LaTeX</code>	$\LaTeX 2_{\epsilon}$	<code>\LaTeXe</code>	\circ	<code>\circ</code>

★フォントサイズ フォントのサイズは { } で囲み、`\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize` (標準), `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge` で指定する。

例えば、`{\Large ABC}` とすれば ABC となる。それぞれのサイズは以下ようになる。

ABC, ABC, ABC, ABC, ABC, ABC, ABC, ABC, ABC, ABC

★囲み 文字などを囲む場合には、以下のコマンドを使う。このとき、`[1]` は”イチ”ではなく”エル”で、`left` を意味している。

タイトルを付ける場合

```
\begin{itembox}[1]{タイトルを付ける場合}
内容 1¥¥
内容 2
\end{itembox}
```

または

```
\begin{screen}
内容 1¥¥
内容 2
\end{screen}
```

のように、タイトルを付ける場合と付けない場合で分かれる。

♠ 補足 `screen` を `shadebox` もしくは `boxnote` に変更すると枠の形状が変わる。

shadebox にした場合
内容

boxnote にした場合
内容

課題 7-1. A4 用紙に課題プリントのような出力となるように \TeX ソースファイルを作成せよ。ただし、最後の S00M000 氏 名 は各自の学生番号, 氏名に変えること。

課題が出来た人は、メールの件名 (題名) を ”課題 7-1. 学生番号 氏名” とし、ソースファイルの 内容 を本文に貼り付け `mori@xmath.ous.ac.jp` まで送ること。件名が異なっていたり、添付ファイルで送った場合は採点されないこともある。

課題 7-1. の締め切りは、**7 月 9 日 (火)** の講義終了時間である。

課題プリント

三角関数の加法定理

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta \quad (1)$$

$$\sin(\alpha - \beta) = \sin \alpha \cos \beta - \cos \alpha \sin \beta \quad (2)$$

$$\cos(\alpha + \beta) = \cos \alpha \cos \beta - \sin \alpha \sin \beta \quad (3)$$

$$\cos(\alpha - \beta) = \cos \alpha \cos \beta + \sin \alpha \sin \beta \quad (4)$$

定義 1.1 f を開区間 I 上の関数とする。 f が $x_0 \in I$ で微分可能であるとは

$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

が有限値として存在するときをいう。

定理 1.1

$\{f_n\}_{n=1}^{\infty}$ を $[a, b]$ 上の関数列で、 $[a, b]$ 上の関数に一様収束しており、各 f_n は有界可積分関数であるとする。このとき、極限関数も有界可積分であって

$$\lim_{n \rightarrow \infty} \int_a^b f_n(x) dx = \int_a^b \lim_{n \rightarrow \infty} f_n(x) dx$$

が成り立つ。とくにベキ級数 $\sum_{n=0}^{\infty} a_n(x - x_0)^n$ に対しては、その収束区間において

$$\int \sum_{n=0}^{\infty} a_n(x - x_0)^n dx = \sum_{n=0}^{\infty} \frac{a_n}{n+1} (x - x_0)^{n+1} + C$$

が成立する。

定義 1.2 (正則変換、正則行列)

1 次変換 f

$$f : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

において、 $ad - bc \neq 0$ のとき、 f は逆変換を持ち、その逆変換 f^{-1} は

$$f^{-1} : \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

となる。このように表現行列に逆変換が存在する 1 次変換 f を正則変換といい、正則変換の表現行列を正則行列という。

以上の作成を L^AT_EX 2_ε で行いました。 S00M000 氏 名

8 数式、図形の表現 III

8.1 T_EX 図形

T_EX で図を描くには `picture` 環境や `METAPOST` などが使われてきた。この講義ではより高度なグラフを描くことができる `TikZ`(ていくす) を用いる。いろいろな例を見ながら説明する。

ちなみに、`TikZ` のマニュアルは

<http://ftp.yz.yamagata-u.ac.jp/pub/CTAN/graphics/pgf/base/doc/pgfmanual.pdf>

にあるが、1200 ページを超える量があり、目的のコマンドを探すのは難しいかもしれない。

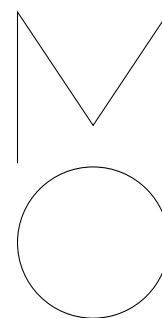
そこで、<https://www.ctan.org/pkg/visualtikz> にある `The document itself` を見るのが良いかもしれない。こちらは、コマンドと結果が表示されている。

例 8-1 (`TikZ` の例 1). `TikZ` を用いたサンプルファイル (左) と、実行結果 (右) を以下に示す。

```

%documentclass[dvipdfmx,a4j]{jarticle}
%usepackage{tikz}
%begin{document}
%tikz%draw (0,0)--(0,2)--(1,0.5)--(2,2)--(2,0);
%tikz%draw (0,0) circle (1);
%end{document}

```



`TikZ` を用いて図を作成する場合、まずプリアンブルに

```
%usepackage{tikz}
```

を追加する (2 行目)。そして、図を描きたい場所でそれぞれのコマンドを入力する。

この例ではまず、座標 $(0,0)$, $(0,2)$, $(1,0)$, $(2,2)$, $(2,0)$ を結ぶ線を描いている。次に、 $(0,0)$ を中心とする半径 1 の円を描いている。

$(0,0)$ の位置が 2 つの図で異なるのは、それぞれ描いた図に合わせて図形領域が確保されるからである。合わせて描きたい場合は、次項 ★ の方法を利用する。

例 8-2 (`TikZ` の例 2). `TikZ` を用いたサンプルファイル (左) と、実行結果 (右) を以下に示す。

```

%documentclass[dvipdfmx,a4j]{jarticle}
%usepackage{tikz}
%begin{document}
%tikz%draw[line width=2pt,rounded corners=12pt,->]
(0,0)--(1,1)--(2,0)--(3,1)--(4,0);
%end{document}

```



(4 行目) 線の幅を 2pt、角を 12pt 分丸くし、線の終わりを矢印 `->` にしている。

線の両端 (一方) を矢印にする方法は、`->` 以外に、`<-`, `<->`, `->>` などがある。

★ 複数の図を1つの領域に描く場合、`\begin{tikzpicture}` と `\end{tikzpicture}` で囲む。

例 8-3 以下は TikZ 部分のみ (左) と、それを含んだ実行結果 (右) を以下に示す。

```
\begin{tikzpicture}[x=20mm,y=20mm,line width=1pt]
```

```
\draw[->] (0,1)--(1,1);
```

```
\draw[-latex] (0,1.5)--(1,1.5);
```

```
\draw[-stealth] (0,2)--(1,2);
```

```
\end{tikzpicture}
```

```
\begin{tikzpicture}
```

```
\draw (0,0) circle(0.5) node {$\pi r^2$};
```

```
\fill[blue!10](2,0) ellipse (1 and 0.5);
```

```
\draw (2,0) ellipse (1 and 0.5);
```

```
\draw (2,0) node {\hbox{\tate 楕円}};
```

```
\draw (-1,-1) rectangle (3.5,1);
```

```
\end{tikzpicture}
```

```
\begin{tikzpicture}[x=1mm,y=1mm]
```

```
\fill[gray] (0,0) circle (1)
```

```
(10,10) circle (1)
```

```
(20,10) circle (1)
```

```
(20,0) circle (1);
```

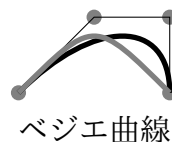
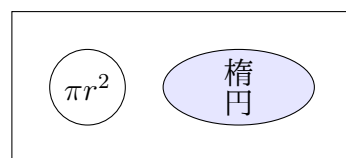
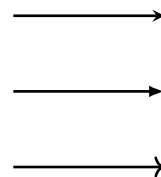
```
\draw(0,0)--(10,10)--(20,10)--(20,0);
```

```
\draw[line width=2pt] (0,0) .. controls (10,10) and (20,10) .. (20,0);
```

```
\draw[line width=2pt,gray] (0,0) .. controls (10,10) .. (20,0);
```

```
\draw (10,-5) node {\hbox{ベジエ曲線}};
```

```
\end{tikzpicture}
```



同じような線を沢山描きたい場合は、`\foreach` が便利である。

```
\foreach \変数 in {a,b,...,c} object
```

これは、変数を a から c まで、増分 $b-a$ で変化させ、`object` を実行させるコマンドとなる。

例 8-4 (foreach の使用例). 実際の命令文と実行結果を以下に示す。(TikZ 部分のみ)

```
\begin{tikzpicture}[x=1mm,y=1mm]
```

```
\draw (0,0)--(100,0);
```

```
\foreach \x in {0,...,100} \draw(\x,0)--(\x,3);
```

```
\foreach \x in {0,5,...,100} \draw(\x,0)--(\x,5);
```

```
\foreach \x in {0,10,...,100} \draw(\x,0)--(\x,7);
```

```
\end{tikzpicture}
```

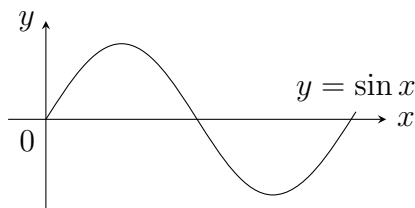


3 行目は変数 x を 0 から 100 まで変化させ、 $(x, 0)$ から $(x, 3)$ まで直線を描いている。

4 行目は変数 x を 0 から 100 まで増分 5 で変化させ、 $(x, 0)$ から $(x, 5)$ まで直線を描いている。

5 行目は変数 x を 0 から 100 まで増分 10 で変化させ、 $(x, 0)$ から $(x, 7)$ まで直線を描いている。

簡単な関数のグラフも描くことが出来る。その他、詳しいことはマニュアル等を参照する。



8.2 画像ファイルの貼り付け

画像を貼り付ける場合は、画像を貼り付けたい辺りに

```
\begin{figure}[位置指定]
  \includegraphics[オプション]{ファイル名}
\end{figure}
```

と入力する。”辺り”と言ったのは、画像の貼り付け場所を [位置指定] で以下の4つから指定するためである。複数書く場合、書く順は関係なく、優先順位は h,t,b,p の順である。

h : 記述した場所に表を出力 , t : ページの上端に表を出力

b : ページの下端に表を出力 , p : 表専用のページを用意して出力

[オプション] には

bb : 画像サイズの情報 , width : 表示幅 , height : 表示高さ

scale : 拡大縮小率 , angle : 回転角 , clip : はみ出しを切り取る

などがある。オプションを複数使う場合は、, で区切る。

bb の指定方法は、画面左下の座標、右上の座標で指定する。例えば、画像サイズが 640×480 の画像なら、

```
[bb=0 0 640 480]
```

とする。この講義では必ず指定すること。

表示幅, 表示高さは縦横の比率を変えて拡大や縮小したい場合や表示サイズを指定したい場合 など に使用する。例えば、画像サイズが 640×480 の画像を縦方向だけ半分にしたい場合は、

```
[bb=0 0 640 480, width=640pt,height=240pt]
```

とする。縦横の比を保って拡大(縮小)したい場合は scale を使う。例えば 30% にしたい場合は、

```
[bb=0 0 640 480, scale=0.3]
```

とする。

画像を回転させたい場合は `angle` (と `origin`) を使う。`angle = n` で回転角 n を指定し (反時計方向回りを+)、回転の中心は `origin=oz` で指定する。`oz` は図の 中心 [c], 左上 [tl], 右上 [tr], 左下 [bl], 右下 [br] のいずれかの ([] 内の) 値を入れる。[デフォルトは bl]

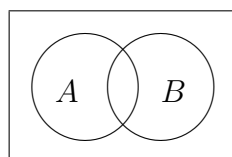
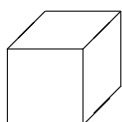
ファイル名の場所には、ファイル形式が EPS, BMP, JPEG などの画像のファイル名を書く。

たとえば、画像サイズが 640x480 の画像を 30% に縮小し、画像の中心を中心とし、30 度回転させて張り付ける場合のコマンドが以下となる。

```
\begin{figure}[htbp]
  \includegraphics[bb=0 0 640 480, scale=0.3,angle=30, origin=c]{ous.bmp}
\end{figure}
```

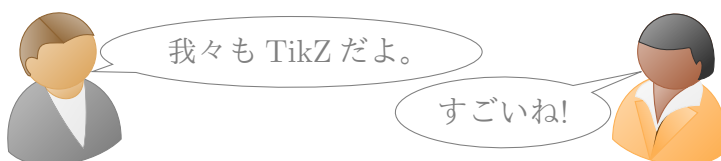


課題 8-1. 以下の図が出力となるように \TeX ソースファイルを作成せよ。ただし、横に並べる必要はなく、大きさや線の太さもそれぞれ異なって構わない。



課題 8-2. 好きな図を考え、その図が出力となるように課題 8-1. の \TeX ソースファイルの続きに作成せよ。その際、作成した 3 つの図が 1 枚の A4 用紙にまとまるようにすること。

課題 8-2. が出来た人は印刷し、用紙に学生番号と氏名を書いて提出すること。課題 8-2. の締め切りは、7 月 16 日 (火) 講義終了時間である。



9 数式、図形の表現IV

9.1 Beamer とは

L^AT_EX でプレゼンテーションを行う場合、以前は slides や seminar のクラスが使われていた。最近では、powerdot や Beamer などのクラスがよく使われている。ここでは、Beamer を使用したプレゼンテーションの説明を行う。まずは例を述べ、そのあと個別に説明を行う。

例 9-1 (Beamer の使用例) 説明のために、行頭に行番号を付けているが、実際は入力しない。

```

1 \documentclass[dvipdfmx,cjk,12pt]{beamer}
2 \usetheme{Antibes}
3 \title{Beamer によるプレゼンテーション}
4 \author{応用 数学}
5 \institute[岡山理科大学]
6
7 \begin{document}
8   \begin{frame}{}
9     \titlepage
10  \end{frame}
11  \begin{frame}{まずは}
12    最初のページ。
13  \end{frame}
14  \begin{frame}[t]{その次}
15    2 ページ目
16  \end{frame}
17 \end{document}

```

まず、1 行目はいつもの呪文。

2 行目はメイン・テーマの指定。省略可能で、省略すると default となる。プリアンブルにはメイン・テーマの他に カラー, フォント, インナー, アウターのテーマも指定できる。

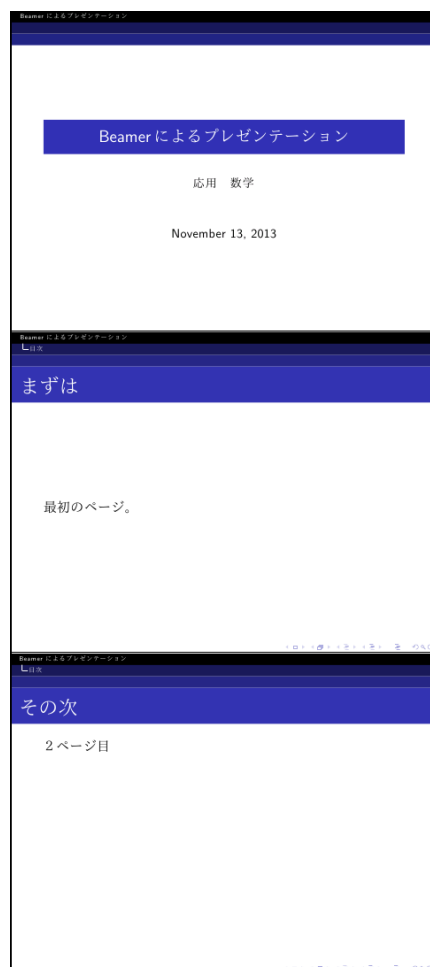
3,4,5 行目は発表タイトル、著者名、所属等である。

6 行目 (`\begin{document}` の前) に空行が必要。

8,9,10 行が 1 ページ目 (タイトル)、11,12,13 行が 2 ページ目、14,15,16 行が 3 ページ目となる。

11 行目や 14 行目の `{frame}` の後の `{ }` 内は各ページのタイトルを示す。また、14 行目の `{frame}` の後の `[t]` は、フレーム内のどの位置 `[top, center, bottom]` (上詰め, 中央, 下詰め) に表示するかを指定している。省略すると `[c]` となる。

右図は、それぞれのページの実行画面である。



9.2 Beamer のコマンド

フレーム

Beamer では、`\begin{document}` と `\end{document}` の中に `frame` を作り、(基本的に) その `frame` 内を 1 ページとする。`frame` 環境の中では、通常の \LaTeX のコマンドを使用する。

[フレーム]

```
\begin{frame}[options]{はじめに}
  プリアンプルの部分は、例 9-1 を参照。
  .....
\end{frame}
```

この `options` にはさまざまなオプションが用意されているが、`t,c,b` を良く使う。

♠ 補足 別ファイルで作成した \LaTeX のソースファイルを読み込むことも出来る。たとえば、`sin.tex` を別ファイルで作成し、読み込むには

```
\input{filename.tex}
```

とする。命令文の行数の多い図や表など もしくは、何度も使うような命令文は、別のファイルに保存しておくと便利である。

ブロック

\LaTeX で普通に使用していた囲みも使えるが、Beamer ではブロックが使える。

基本的なブロック囲みは `\begin{block}` と `\end{block}` で囲む。

[ブロック]

```
\begin{block}{基本ブロック}
  基本的なブロック囲みは
\end{block}
```

基本ブロック

基本的なブロック囲みは

`\pause`

1 つのフレーム内で、表示を一時停止する場合に `\pause` を使う。スペースキーを押すことによって、`\pause` 以下を表示する。

例 9-2 (`\pause` の使用例)

```
\begin{frame}{}
  まず、この部分は表示される。
```

```
\pause
```

この部分は、スペースキーが押されるまで表示されない。

```
\end{frame}
```

オーバーレイ

`\pause` よりも表示の順番を細かく設定する方法として、`itemize` を使用する。

[オーバーレイ](説明のため行番号付き)

```
1 \begin{itemize}
2   \item<1-> その 1
3   \item<2-3> その 2
4   \item<3> その 3
5   \item<4> 2,3 を消して、その 4
6   \item<1,4> その 1 とその 4 のとき表示
7 \end{itemize}
```

上の例では、

2 行目の `\item<1->` は 1 回目以降、表示される。

3 行目の `\item<2-3>` は 2 回目と 3 回目に表示される。

4 行目の `\item<3>` は 3 回目のみ表示される。

5 行目の `\item<4>` は 4 回目のみ表示される。

6 行目の `\item<1,4>` は 1 回目と 4 回目に表示される。

色付け

\LaTeX と同様に文字に色付けをすることが出来るが、Beamer では標準で `red`, `blue`, `green`, `cyan`, `magenta`, `yellow`, `black`, `darkgray`, `gray`, `lightgray`, `orange`, `violet`, `purple`, `brown` などがある。

使用方法は

```
{\color{上記の色}色付けしたい文字}
```

であり、例えば

```
{\color{red}赤色の文字}
```

とする。最初と最後の `{ }` を忘れると、それ以降の文字が全て色付けされる。

その他

上記以外にもさまざまな機能があり、慣れてくると非常に便利なソフトである。

しかし、あくまでも pdf ファイルに変換するため、pdf ファイルで出来ないことは出来ない。
また、追加機能も多いため組み合わせると正しく動かないコマンドも多い。

(私が使う上では不自由はしていない。)

どう使うかにおいて、想像力と工夫が必要である。

9.3 二段組

ページの中央で分割し、二段組にすることが出来る。試験問題などで使われることがある。

表現とメディアの数理 定期試験

学生番号 _____ 氏名 _____

問題 1. 64MiB の容量をもつ USB メモリに、日本語が何文字保存できるか答えよ。 ただし、メディアフォーマット等に必要な領域は 0 とする。	問題 2. 次のアドレスをドットアドレスで表せ。 (1) 01000100 10000000 11111111 00000100 (2) 11110000 00111010 00011000 01111111
--	--

二段組をする場合は、プリアンプルに `\usepackage{multicol}` を書き、二段組をしたい場所に

`\begin{multicols}{2}`

二段組にしたい内容 (左側)

`\columnbreak`

二段組にしたい内容 (右側)

`\end{multicols}`

とする。この命令文の前に次のコマンドによって、左右の間隔や区切り線の太さを指定できる。

`\setlength{\columnsep}{10.0pt}` で左右コラムの間隔の指定

`\setlength{\columnseprule}{0.4pt}` で区切り線の太さの指定

♠ 補足 なお、1 ページ全体や 1 ソース全体を二段組にする場合は、`\twocolumn` ~ `\onecolumn` を用いたり、`twocolumn` クラスを使う。

課題 9-1. (1) 例 9-1 を参考に、プレゼンテーション用の L^AT_EX ソースファイルを作成せよ。

1 ページ目はタイトルがくること。

2 ページ目には大学生生活や卒業研究、教職についてなど何かコメントを上詰めで書くこと。

3 ページ目には `itemize` を用いて、1,2,3,4,5,6,7 に対して、1 回目は 1 のみ、2 回目は偶数のみ、3 回目は奇数のみ、4 回目は素数のみ、5 回目は全てが表示されるページを作ること。

4 ページ目にはブロックを用いて何か定義か定理を挿入すること。

(2) (1) で作成したものに対して、メイン・テーマ (例 9-1 の 2 行目) をネットで調べ、Antibes 以外のものに変更し、`\author{応用 数学}` の ”応用 数学” は各自の学生番号, 氏名に変更せよ。

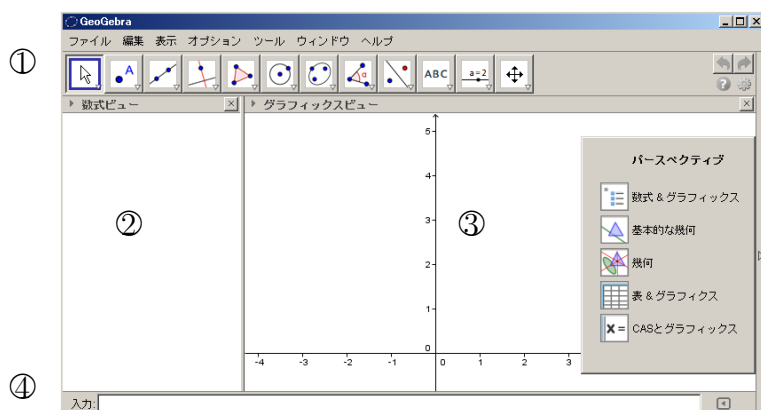
課題が出来た人は、メールの題名を ”課題 9-1. 学生番号 氏名” とし、ソースファイルを本文に貼り付けて `mori@xmath.ous.ac.jp` まで送ること。

課題 9-1. の締め切りは、7 月 19 日 (金) 講義終了時間である。

10 Geogebra I

10.1 GeoGebra とは


GeoGebra = Geometry (幾何) + Algebra (代数) を意味し、対話的なグラフィックスで直感的に操作が出来、代数, 幾何, 解析を統合的に扱える数学ソフトウェアである。起動画面上の



① を作図ツール (群)、④を入力テキストフィールドと呼び、図形や数式を入力するために使用する。②を代数ウィンドウ、③をドローイングパッドと呼び、入力した図形や数式を代数的、視覚的に表示する。

右側に表示された パースペクティブ はデフォルトの ”数式&グラフィックス” を使用する。

・ 作図ツール

作図ツールを用いれば、ドローイングパッドにマウス操作で入力出来る。作図ツールには点や 直線を描くツールのほか、中点や二等分線を入力するツールも含まれる。各ツールは、右下のやじるし  をクリックすると、そのツール群に含まれるものが表示される。

・ 入力テキストフィールド

入力テキストフィールドを用いれば、座標, 方程式, コマンドや関数を直接入力することが出来る。また、右端には記号や真偽値演算が選択できるようになっている。入力するコマンドは英語が多いが、補完機能があるので、最初の2,3文字を入力すると、選択が可能となる。

数式の入力には少し注意が必要である。まず、 x^2 は $x^{\wedge}2$ と入力し、以下 x^3 は $x^{\wedge}3$ と入力する。また、掛け算 $a \times b$ は $a*b$ で、割算 $\frac{a}{b}$ は a/b と入力する。

掛け算*は省略可能な場合もある。2a などは可能だが、ab は ab とい変数か、 $a \times b$ か解らないため、省略しない方がよい。(場合で異なる。) コマンドの大文字、小文字は区別されないが、変数は区別されるので注意が必要である。例) $A=(1,1)$ は点 $A(1,1)$ となり、 $a=(1,1)$ は $(1,1)$ ベクトルとなる。

10.2 使い方

GeoGebra には入力テキストフィールドへの入力と、作画ツールを用いてドローイングパッドに入力する 2 つの方法がある。多くの図形 (コマンド) はどちらでも入力が可能で、どちらの場合も、代数ウィンドウに代数的な表示、ドローイングパッドに視覚的な表示がされる。

ただし、どちらか一方でしか入力できないものもあるため注意が必要である。

・入力テキストフィールドを用いた入力

まずは入力テキストフィールドでの入力を行う。

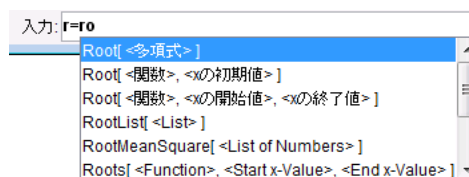
例 10-1. 3 次関数 $f(x) = x^3 - 3x^2 - x + 1$ に対して $f(x) = 0$ の解、 $y = f(x)$ の極値と変曲点を求めてみる。入力テキストフィールドに

```
f(x)=x^3-3x^2-x+1
r=root[f]                # 根
s=extremum[f]            # 極値
t=inflectionpoint[f]     # 変曲点
```

と入力する。(# 以降はコメントである。入力しない。)

3 次関数のグラフと、 x 軸との交点、極大、極小の点、変曲点が表示される。

[補完機能] 例えば `s=root[f]` を入力している途中で、候補が表示される。この中から目的のものを選択すれば、続きを打つ必要がなくなる。() なども補完される。

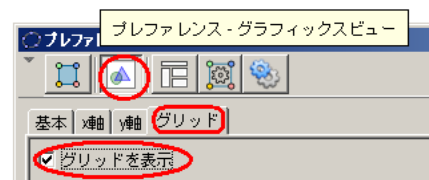


例 9-1 の後、 $f(x)$ の関数を変えてみる。入力テキストフィールドに

```
f(x)=x^3-3x^2
```

と入力する。3 次関数のグラフが変わり、それに伴って各点も変化する。このように、他のオブジェクトに従属するオブジェクトは、主のオブジェクトを変更すると自動的に変更される。

[グリッド] 場合によっては、座標軸があると便利かもしれない。ツールバーの [表示] から [レイアウト] を選び、プレファレンス-グラフィックビューの "グリッド" から、"グリッドを表示" にチェックを付けると方眼用紙のようになる。



♠ 補足 マウスのホイールを使うとズームイン、ズームアウトが可能である。また、座標軸をスライドさせるには、作画ツールの右端にある "グラフィックビューの移動" を使う。

♠ 補足 グラフが沢山表示されて見辛くなったら、代数ウィンドウで表示させたくないオブジェクトの○印をクリックすると表示の ON/OFF が切り替えられる。

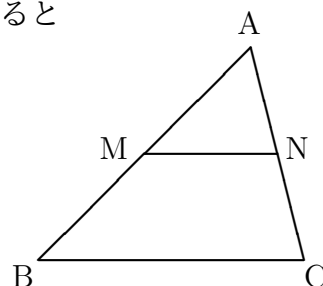
・作画ツール群を用いた入力

作画ツールを用いた入力に慣れるために次の定理を確認する。


中点連結定理

△ABCの2辺AB, ACの中点をそれぞれM, Nとすると

$$MN \parallel BC, \quad MN = \frac{1}{2}BC$$

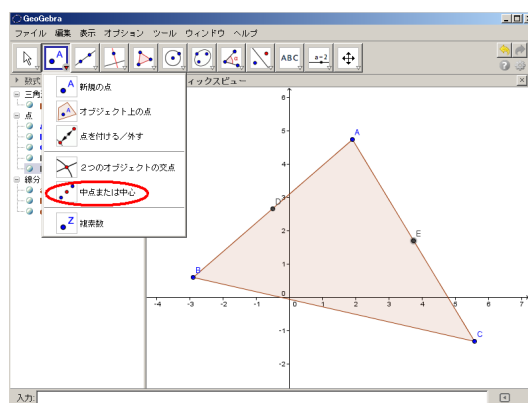


まず、“ファイル”から“新規”を選び、新たなドローイングパッドで三角形ABCを作画する。

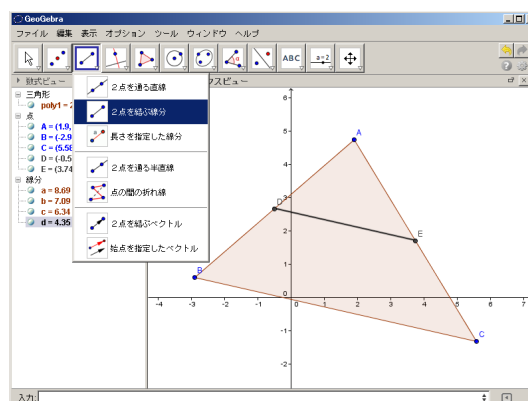
作画ツールから“多角形”モード  を選び、ドローイングパッド上で異なる3ヶ所をクリックし、3点A, B, Cを作成する。このとき、3点A, B, Cは1直線上になく、少し離して選ぶ。

その後、三角形を完成させるため最初の場所(点A)をクリックする。点Aと異なる点をクリックし、点Dが作画されないように気をつける。⁵

三角形が出来たら、辺AB, ACの中点を作画するために“中点または中心”(右図参照)を選ぶ。その後、点Aと点Bをクリックすると中点Dが作画される。同様に点Aと点Cをクリックする。



点Eが作画されたら、出来た2点D, Eを結ぶ線分を作画する。先ほどのアイコンの右側にあるツールから、“2点を結ぶ線分”モードを選択し、点Dと点Eをクリックする。

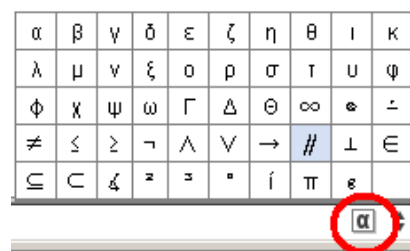


⁵間違えた場合は消したいオブジェクトを右クリックして削除することが出来る。最初から行う場合は、“ファイル”から“新規”を選ぶ。

線分 DE が描けたら入力テキストフィールドを使う。このとき、平行を意味する真偽値演算 \parallel は入力テキストフィールドの右端、 α タブから選択して入力する。

キー入力の $/$ を 2 つでは正しく動かない。

このことに注意して、入力フィールドに



$$\text{平行} = a \parallel d$$

と入力する。代数ウィンドウに 平行=true と表示されたはずである。これは線分 a と d が平行なとき true と表示され、平行でないとき false と表示される⁶。次に

$$\text{長さ} = a \dot{=} 2 * d$$

と入力する。等しいことを意味する真偽値演算 $\dot{=}$ もタブから選択する。これは、a と 2d が等しければ true と表示することを意味している。

平行も長さも true と表示されたら、作画ツールの”移動”モード (左端の矢印のついたアイコン) を選び、点 A, B, C の 1 点をドラッグすると三角形の形が変わるが、平行と長さは常に true を表示している。これで、どのような三角形でも中点連結定理が成り立つことが確認できた。

例題 10-1. 三角形の 5 心の定理を確認せよ。

ユークリッド空間内の三角形の 5 心

- ・ 外心 三角形の 3 辺の垂直二等分線は 1 点で交わる。
- ・ 重心 三角形の辺の中点と向かい側の頂点を結ぶ線分 (3 つ) は 1 点で交わる。
- ・ 垂心 三角形の 3 つの頂点から対辺に下ろした垂線は 1 点で交わる。
- ・ 内心 三角形の 3 つの内角の二等分線の交点は 1 点で交わる。
- ・ 傍心 三角形の 1 つの内角と他の 2 つの外角の二等分線 (3 つ) は 1 点で交わる。

課題 10. まず、三角形を 1 つ作図する。その三角形に対して、定規とコンパスで作図する方法で外接円と内接円を作図せよ。

♠ **補足** 定規とコンパスで描けることをしっかり理解すること。

課題 10. が出来た人は、ファイルに保存し、メールに添付ファイルとして mori@xmath.ous.ac.jp まで送ること。メールの件名はいつも通りの約束で ”課題 10. 学生番号 氏名” とすること。

課題 10. の締め切りは、**7 月 19(金) 講義終了時間**とする。

⁶true と表示されないときは、=などが全角になってないか、点 D ,E の取り方などを確認する。

11 Geogebra II

11.1 軌跡

GeoGebra では軌跡を表すことが出来る。そこで次の例題をを考える。

例題. 点 $A(4,4)$ に対して、点 P が円 $x^2 + y^2 = 4$ の周上を動くとき、次の点の軌跡を求めよ。

- (1) 線分 AP の中点 M
- (2) 線分 AP を $1:2$ に内分する点 Q

ここでは、問題文の点や円を正しく入力するために入力テキストフィールドを使う。

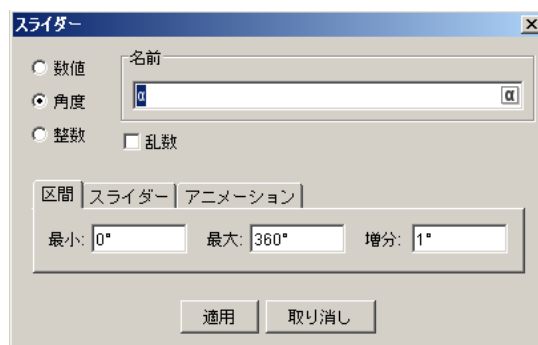
まず、入力テキストフィールドに

$$A=(4,4)$$

$$x^2+y^2=4$$

と入力する。次に、作画ツール から ”スライダー” を選択する。選択後、ドローイングパッド内をクリックするとスライダーの設定画面が表示される。

ここで、”数値” に入っているチェックを ”角度” に変えて ”適用” を押す。



次に、入力テキストフィールドで

$$P=(2\cos(\alpha), 2\sin(\alpha))$$

と入力する。 α は $//$ と同様に α タブから選択する。続けて

`segment [A,P]`

線分

`M=midpoint [A,P]`

中点

`Q=2*A/3+P/3`

AP を $1:2$ に内分する点

と入力する。入力が終わりましたら、代数ウィンドウ内の α を右クリックし、アニメーションにチェックを付けると軌跡が表示される⁷。

課題 11-1. 次の軌跡を表示させ、それぞれの方程式、座標を目視で考察し解答せよ。

(1) 点 $A(-2, 0)$, $B(2, 0)$ と円 $x^2 + y^2 = 4$ がある。点 C がこの円上を動くとき、3点 A, B, C が三角形を成したときの重心の軌跡を表示し、軌跡の方程式のおおよそを答えよ。

(2) 放物線 $y = x^2 + 2(a - 2)x + a$ の頂点の軌跡を表示し、その軌跡の描く曲線と x 軸、 y 軸との交点を答えよ。

⁷残像を残したい場合は、代数ウィンドウ内の M または Q を右クリックして、[残像表示] にチェックを入れる。

11.2 微分, 積分

GeoGebra は関数の微分, 積分も表すことが出来る。入力テキストフィールドに

```
derivative[(x-1)^3]          # 微分
```

と入力すると、 $(x-1)^3$ の微分が求まる。 $\sin x$ や $\log x$ の微分

```
derivative[sin(x)]
```

```
derivative[log(x)]
```

も関数のグラフとして表示させることが出来る。積分も微分と同様に

```
integral[(x-1)^3]
```

と入力すると、 $(x-1)^3$ の積分が求まる。 $\int \sin(x)dx$, $\int \log x dx$ も同様に次のように入力する。

```
integral[sin(x)]
```

```
integral[log(x)]
```

積分の場合、区間を入力することで定積分を求めることが出来る。例えば、 $\int_0^\pi 2 \sin(x)dx$ は

```
integral[2sin(x),0,Pi]
```

と入力する。

11.3 関数の微分と接線

関数 $f(x) = \sin(x)$ のグラフ、その導関数、グラフ上の点における接線を表示する。まず

```
f(x)=sin(x)
```

```
A=point[f]
```

グラフ f 上の点

```
t=tangent[A,f]
```

点 A における f の接線

```
s=slope[t]
```

t の傾き

とテキスト入力フィールドに入力する。

ここで、ドローイングパッド上に表示された点 A をドラッグして左右に動かすと、関数 $f(x) = \sin(x)$ の各点における接線の傾きが表示される。各点での傾きを見るため、

```
B=(x(A),s)
```

$x(A)$ は点 A の x 座標の値

とテキスト入力フィールドに入力し、先ほどと同様に点 A を左右に動かす⁸。

B の動き (s の変化 = 各点での傾きの変化) を見るには、B に残像をつければ解りやすい。この B は、もちろん $(x, f'(x))$ だから

```
derivative[f]
```

で表示される。

⁸この点 A は代数ウィンドウ内の A を右クリックして [アニメーション オン] にチェックをつけると、アニメーションさせることも可能である。

11.4 マクローリン展開

スライダー (パラメーター) を使うことによって、関数の冪級数の近似値を目視で求めてみる。
ここでは、マクローリン展開を考えることにする。すなわち、

$$\begin{aligned} f(x) &= f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \cdots \\ &=: a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots \end{aligned}$$

の $a_0, a_1, a_2, a_3, \dots$ を考える。

例題 . 関数 $f(x) = \frac{1}{1-x/2}$ のマクローリン展開の x^3 の項までを GeoGebra を用いて推測せよ。

まず、入力テキストフィールドに

$$f(x)=1/(1-0.5*x)$$

と入力する。 a_0 は明らかに 1 なので、

$$g(x)=1$$

とする ($g(x)$ が求めたい近似式)。次に、

$$t(x)=f(x)-g(x)$$

と入力し、プロパティからこの関数の色 (例えば 青) を変る。次に、スライダー (a) を作成し

$$h(x)=a*x$$

とする。 a をアニメーションさせることにより、青色 ($t(x)$) のグラフと黒色 ($h(x)$) のグラフを観察すると a の値が 0.5 の辺りで一番近似されていることが解る (のではないか?)。解りにくければ、スライダー a のプロパティで、区間を 最小 0, 最大 1, 増分 0.01 とすれば解りやすくなるかもしれない。さらに、アニメーション の速度を 0.1 に変更すればさらに解りやすいだろう。

そこで、 $g(x)$ を $g(x)=1+0.5*x$ に変更し、 $h(x)$ を $h(x)=a*x^2$ に変更する。

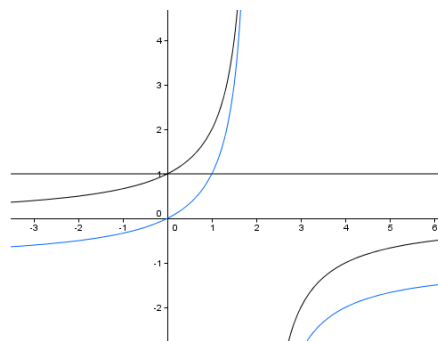
先ほどと同様に、青色 ($t(x)$) のグラフと黒色 ($h(x)$) のグラフを観察すると、(難しくなってくるが、第一象限と第二象限をみて) だいたい 0.25 だと解る。

同様のことをもう一度行くと、3 次までの近似が $g(x) = 1 + 0.5x + 0.25x^2 + 0.125x^3$ だと推測できる。もちろん推測なので、別の答えが出るのも良い。

課題 11-2. $y = \log(x+1)$ のマクローリン展開の係数の近似を GeoGebra で目視し 3 次近似まで推測せよ。また、その近似を用いて $\log 2, \log 3$ の値を推測せよ。

課題が出来た人は、答えをメールの本文に書き、mori@xmath.ous.ac.jp 宛てに送ること。メールの件名はいつもの通りの約束で ”課題 11-1. 11-2. 学生番号 氏名” とすること。

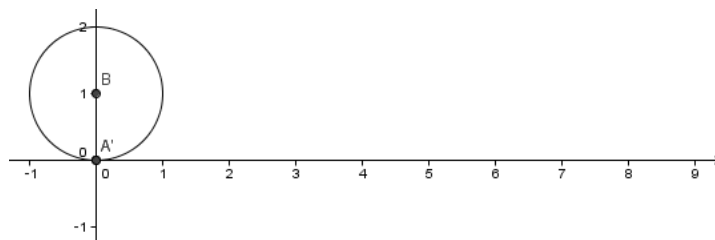
課題 11-1, 11-2 の締め切りは、**7 月 23 日 (火) 講義終了時間**である。



11.5 おまけ

・サイクロイド

軌跡の1つであるが、サイクロイドという曲線がある。円が直線上を転がるとき、円の定点の軌跡



が描く曲線である。ちなみに、サイクロイドは $x = r(u - \sin u)$, $y = r(u - \cos u)$ で表わされる。

この動きを表示させるために、以下をテキスト入力フィールドに入力する。

```
r=1
t=0
A=(t,0)
B=(t,r)
A'=rotate[A,-t/r,B]      # B を中心に A を角度 -t/r 回転した点
circle[B,A]
```

ここで、代数ウィンドウ内の”数値”欄にある $t=0$ を右クリックし、プロパティを開き、スライダーを最小 -10 、最大 10 として閉じる。再び $t=0$ を右クリックし、”アニメーション オン”をチェックすると円が転がる。定点の軌跡を表示するには、代数ウィンドウ内の A' を右クリックして、[残像表示] にチェックを入れる。

・正五角形の作図

2点 A, B を作図し、線分 AB を1辺とする正五角形を作図する。円の書き方を工夫し、円が多くなり見難くなったら、不要になったオブジェの表示を OFF にして見やすくする。

- (1) 線分 AB の垂直二等分線 l を引き、 AB との交点を C とする。
- (2) 点 C を中心に半径 AB の円⁹を描き、直線 l との交点の一方を D とする。¹⁰
- (3) 2点 A, D を通る直線 m を引き、点 D を中心に AB の半分の長さ ($a/2$) の円を描き、直線 m との2つの交点に対し、 ADE の順となるように交点 E をとる。
- (4) 点 A を中心に半径 AE の円を描き、直線 l との交点を F とする。
- (5) 点 F が正五角形の一番上の頂点であるが、点 F , 点 A , 点 B を中心に半径 AB の円 $O(F)$, $O(A)$, $O(B)$ を描き、円 $O(F)$, $O(A)$ の交点を G とし、円 $O(F)$, $O(B)$ の交点を H とする。

注) G, H はそれぞれ2点あるが、五角形 $ABHFG$ が正五角形になるように選ぶ。

以上の作図で五角形 $ABHFG$ は正五角形となる。ちなみに、点 A または B を移動させても、正五角形は保たれている。

⁹線分 AB を作成し、長さが a だとする。”中心と半径で決まる円”を選び、中心を選んだ後 半径を a と入力。

¹⁰これ以降の点はすべて直線 AB に対し D 側に点を取るとする。

12 Scratch I

12.1 Scratch とは

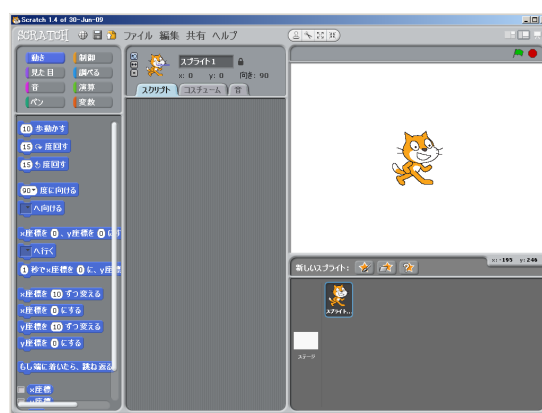
初心者が最初に正しい構文の書き方を覚えることなく結果を得られるプログラミング言語学習環境である (ウィキペディア より)。

2年生の講義ではC言語を用いてプログラミングを行う。プログラミングを行うとき、最初に必要なのはアルゴリズム (すなわち答えまでの手順) を考えることである¹¹。実は、この作業が苦手な人が多いようなので、難解なプログラミングの構文やフローチャートなど意識せず、プログラミングが出来る Scratch を紹介する。

Scratch を起動すると右の画面が表示される。

左側はブロックパレットがあり、ここにはプログラムを作るために必要なフラグメント (ブロック) がある。

真ん中には (ブロックを組み立てる) スクリプトエリアがあり、右側がステージとスプライトのリスト (アクションエリア) がある。




プログラムの作成は、ブロックパレットから命令や制御文をスクリプトエリアにドラッグし、繋げていく。

12.2 プログラム作成

実際に簡単なプログラムを作成して、動作の仕組みを理解する。

まず、ブロックパネル内の **動き** から、**10 歩動かす** を選択し、スクリプトエリアにドラッグする。ドラッグした **10 歩動かす** をダブルクリックすると、実際にどのような動きか確認できる。



プログラムの開始動作を確実にするために、**制御** を選択し、**がクリックされたとき** をスクリプトエリアにドラッグし、これら 2 つを **10 歩動かす** のように繋げる。

繋がったら右上の  の旗をクリックする。クリックすると scratch cat が 10 歩進む。


¹¹この点は、数学の証明問題と似ていると思う。

これだけでは解りにくいので、再び **制御** 内から **ずっと** を選択し、スクリプトエリア

にドラッグする。そのとき、**がクリックされたとき** **ずっと** **10 歩動かす** となるようにドラッグする。

繋がたら先ほどと同様に右上の  の旗をクリックする。今度は scratch cat が右端まで行き、止まる。そのとき、スクリプトエリア内のブロックが白枠で囲まれた状態になっている。これは、プログラムが動き続けていることを意味している。そこで、プログラムを停止するために  の **●** をクリックする。すると、白枠が無くなり停止したことを表している。

次に **動き** から、**もし端に着いたら、跳ね返る** を選択し、**10 歩動かす** となるようにドラッグする。

再び右上の  の旗をクリックすると、今度は右端まで行くと折り返して左端まで行き、また折り返して右端へと向かう。停止するには、先ほどと同様である。

♠ 補足 このときの折り返し方は、 の左側 3 種類から選択できる。

このように Scratch ではブロックを繋ぎ合わせることでプログラムを作成することが出来る。もちろん C 言語や Java のように複雑なプログラムは組めないが、アルゴリズムの学習には十分な命令 (ブロック) があるので、各自で試してほしい。

また、作成したプログラムは [ファイル] から [保存する] で保存できる。



保存する際は、左の欄から [コンピューター] を選択し、D: を選ぶ。さらに、フォルダを作成しておくと、後で解りやすい。なお、保存されたファイルの拡張子は sb である。

課題 12. 右のプログラムは、スペースキーを押すと
(雑ではあるが) ジャンプするプログラムである。

このプログラムを参考に、スペースキーを押すと、ジャンプし着地後反対方向へ進むプログラムを作成せよ。

また、キー [e] が押されたら動きを終了するプログラムを付け加えよ。

♠ 補足 ステージの座標は中心が (0, 0) で、 x 軸が -240 から 240、 y 軸が -180 から 180 のサイズになっている。

・入力, 出力

今度はキーボードからデータを入力し、表示するプログラムを考える。

例 12-1. まず、キーボードから入力したデータを扱えるようにする。ブロックパネル内の [調べる] から、**あなたの名前は何ですか? と聞いて待つ** を選択し、スクリプトエリアにドラッグする。

♠ 補足 ドラッグしたブロック内の "あなたの名前は何ですか?" は変更できる。ただし、日本語入力の場合、直接入力ではなく 左上に表示される形式 となっている。

入力された答えを扱うために、**答え** もスクリプトエリアにドラッグする。さらに、ブロックパネルの **見た目** から **こんにちは! と 2 秒言う** もスクリプトエリアにドラッグする。**答え** を

こんにちは! と 2 秒言う の "こんにちは!" の辺りにドラッグし、


ようにプログラムする。

このプログラムの実行結果の確認は、各自に委ねる。

・制御

制御の内、プログラムの流れの中で、条件によって分岐を行う代表的な 2 つを紹介する。

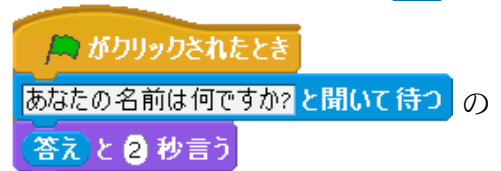
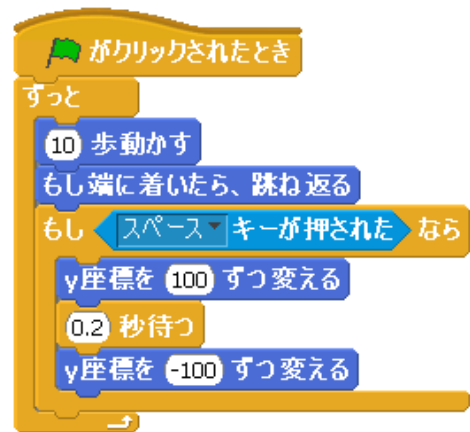
右のブロックの上のものは、条件  を満たしているなら、その間にあるブロックを実行することを意味している。

下のブロックは、条件  が満たされていないとき、その間のブロックを実行し、さらにその条件が満たされるまで繰り返す。

ここでいう "条件" とは [演算] にある  **<**  **かつ**  **ではない** などである。

これらの条件分岐は良く使われるので、しっかり活用して欲しい。

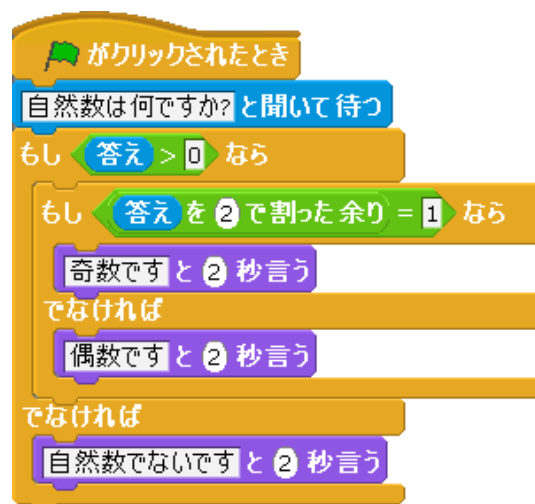
制御を使ったプログラムの例を見ながら考えてみる。



例 12-2. キーボードから入力されたデータが、奇数か偶数か表示するプログラムを考える。入力された値が、奇数か偶数か判断するには、

”もし～なら” - ”☆する” - ”でなければ★する”という条件分岐を用いる。

右のプログラム例では、それを囲うように、負の数や文字が入力された場合、メッセージを表示するようにしている。ただし、小数の場合は上手く処理できない。



課題 12. が出来た人は、メールの題名を ”課題 12. 学生番号 氏名” とし、プログラムファイル (拡張子がsbのもの) を添付し、 mori@xmath.ous.ac.jp までメールを送ること。

課題 12. の締め切りは、7月26日(金) 講義終了時間である。

♠ 補足 初めに表示されているキャラクター (スクラッチキャット) を変更したり、追加することも出来る。また、背景を変更したり、書くことも容易である。



♡ 注意! Scratch 1.4 の日本語版にはバグがあり、各自でインストールするときは必ず修正パッチをあてること。また、自宅等、ネット環境の整ったところで行う場合はwebアプリとなった2.0以降が良い。

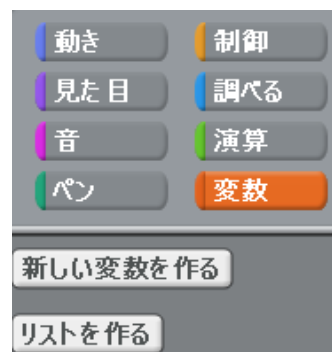
iPhone,iPad アプリではPyonkeeという類似のソフトが有り、Windows 10, macOS 10.13 なら Scratch デスクトップが使える。

13 Scratch II

13.1 変数

前回の例12-1.では1つの変数(入力された値)だけを扱っていたが、実際のプログラムを作るときは複数の変数が必要となる。今回は複数の変数を用いたプログラムを考える。

ブロックパネル内の**変数**を選択すると、右のような画面になる。ここで、**新しい変数を作る**を選ぶと、変数名?と聞かれるので、変数名(例えば x)を入力する。入力すると変数が作成されるので、後は**答え**と同様に扱えばよい。



また、変数を増やして、**演算**を行うことも出来る。

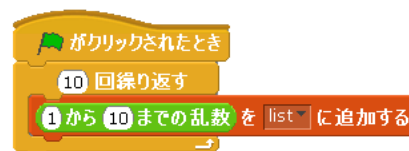
♡ **注意!** 演算で、数値の代入を(y の値を x に代入)するとき、

= を用いてはいけない。このブロックは比較のとき使うもので、代入は **x を y にする** を用いなければならない。また、 x に $3y$ を代入するなら、 **$3 * y$** を用いればよい。

13.2 リスト

変数をたくさん使ったプログラムを考える場合は、リスト(配列)を使う。リストは変数を束ねたようなもので、 a, b のような変数に対して、 $a[1], a[2]$ のようにして使うことが出来る。

例として、リストに1から10までの乱数が10個代入されるスクリプトを作成する。



まず、変数の欄から**[リストを作る]**を選ぶ。リストの名前で使う文字は英数とする。(今回の例では `list` としている。)

次に、**[制御]** から **10 回繰り返す** を選び、その中に**[変数]**の **なにかを list に追加する**

を入れる。さらに、**なにか**の場所に**[演算]**の **1から10までの乱数** を入れる。

これにより、代入された数は右のように確認することが出来る。

このリストに入っているデータを使う場合、**list の 1 番目** を使えばよい。

また、リストから1つの値を消したい場合は、**1 番目を list から削除する** を使い、全て消す場合は **すべて 番目を list から削除する** とする。



例 12-1. 10 個の乱数データを作成、表示される場合は次のようにすればよい。

- (1) リストの中にデータが有る場合もあるため、まずリスト内から全てのデータを消す。
- (2) 1 から 10 までの乱数を発生させ、リストの中に順に格納して行く。
- (3) i という変数を作り、1 にしておく。
- (4) リストの i 番目を 1 秒間表示させ、 i に 1 を加えて繰り返し、 i が 11 になったら止める。



13.3 ソート

ソートとは、与えられたデータを一定の規則に従って並べることである。ソートの方法はデータの構造やデータ数によってさまざまな方法が存在する (興味のある人はネット検索で)。いろんなソート方法を試してみるのが良いが、どの方法でも注意しなければならないことがある。

♡ 注意! リスト内の i 番目と j 番目を入れ替える場合、どちらかを別の変数に一度代入しておかなければならない。



(list の 1 番目と 2 番目を入れ替える方法. 一度 変数 i に list の 1 番目のデータを代入している。)

課題 13. 乱数で 1 から 100 までの数 20 個をリストとして作成し、小さい順に並べ替えるプログラムを作成せよ。

ただし、開始は右のようにリストから全てのデータを削除するようにして開始させること。データを削除せずに行うと、前回のものに 加えて 20 個のデータが作成される。



課題 13. が出来た人は、その画面を教員または演習補助の先輩に見せて動作確認をしてもらい、OK を貰うこと。講義外で行った人は保存しておいて、次回の講義のときに見せること。

課題 13. の締め切りは、7 月 30 日 (火) 講義終了時間である。

14 Scratch III

14.1 フローチャート

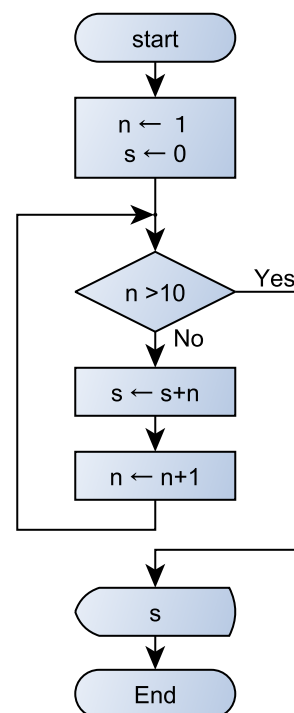
フローチャートとは作業の流れを図で表したものである。

各ステップを様々な形の箱で表し、それらの間の流れを矢印 (または実線) で繋ぐことでアルゴリズムやプロセスを表現する。

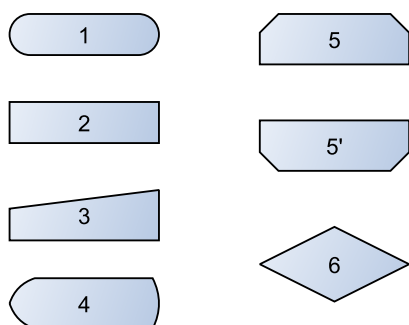
フローチャートを用いることで作成したいプログラムの大まかな流れを図示し、プログラムを組む際に役立てられる。また、他者へ説明する場合にもプログラムを見せるより視覚的で解りやすい。フローチャートはプログラミングのみならず、いろいろな分野で用いられている。

右のフローチャートは1から10までの和を求め、表示する流れを表したものである。

♡ 注意! フローチャートの描き方にはいくつかの流儀があり、細かい点で異なっている。例えば、start の直後の→のように、明らかに進む方向が解るものは実線で書く場合がある。



・ パーツ (様々な形の箱)



フローチャートに使われるパーツ (箱) のうち、よく目にするものを紹介しておく。

右の1は、手続きの始まりや終わりを意味する。

2は処理や命令

3は手動 (主にキーボード) からの入力

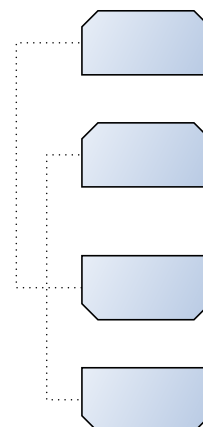
4は画面への出力

5,5' はペアーで使い、ループの開始と終了で、6は条件分岐である。

♡ 注意! 5, 5' を用いてループを考える場合、右のようなループは出来ないことに注意する。

また、6の条件分岐は成立つか否かの二択で用いることが多い。

(Switchのような場合は別)



例 14-1. 例えば入力された数が素数か合成数か判断するフローチャートは右のようになる。ただし、右のフローチャートはScratch用に書いたものなので、標準的なフローチャートの記法とは異なる。

♠ 補足 2つのエンドのうち、右上のエンドは強制終了のエンドである。

課題 14-0. 授業アンケートを記入すること。

課題 14-1. (1) ”コラッツの予想”をフローチャートを用いて考えよ。ただし、課題としてフローチャートを提出しなくてよい。

コラッツの予想

- 1) 自然数 n を考える。
- 2) n が偶数ならば n を 2 で割る。
 n が奇数なら 3 倍して 1 加える。
- 3) 2) の結果 (を新たな n とし、 n) が 1 ならば終わる。
 $(n$ が) 2 以上なら再度 2) を行う。

以上の方法でどんな自然数 n も必ず 1 になる。
(という予想)

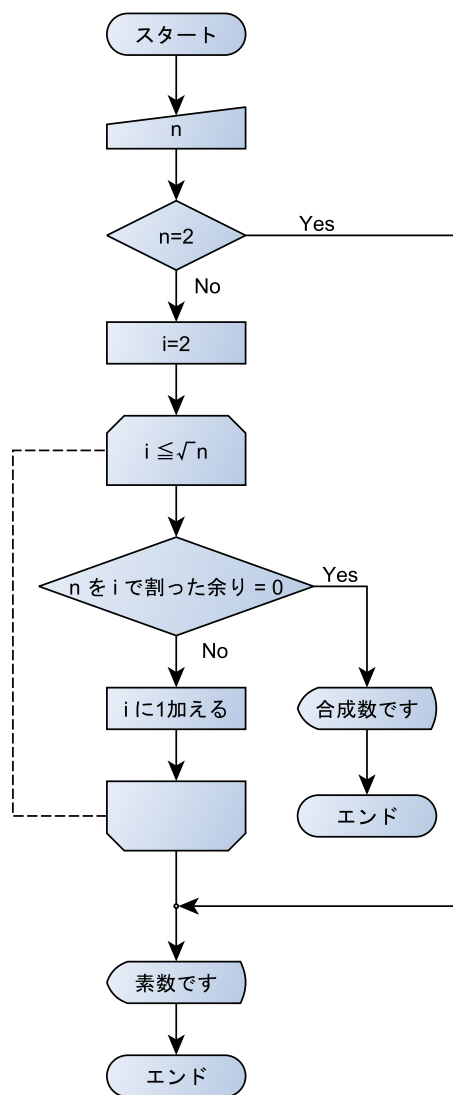
(2) そのフローチャートを元に、Scratch で入力された数が 1 になることを確認せよ。その際、それぞれの結果を 2 秒ずつ表示させるようにすること。
例えば、3 が入力されたとすると

$3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

の順で 1 になるので、10, 5, 16, 8, 4, 2, 1 をそれぞれ 2 秒表示させる。

課題が出来た人は SA の先輩に確認してもらうこと。講義時間外に行った場合は、次回の講義のときに確認してもらうこと。

課題 14-1 の締め切りは、8 月 2 日 (金) 講義終了時間 である。



15 まとめの課題

以下の課題を行い、課題 15-1. の提出は、メールの題名を ”課題 15-1(x). 学生番号 氏名” とし (x は 1 or 2)、プログラムファイルを添付し、`mori@xmath.ous.ac.jp` までメールを送ること。

また、課題 15-2. は、完成した pdf ファイルを印刷し、ホッチキス止めをするか、各紙に学生番号氏名を書いて提出すること。

課題 15-1,15-2. の締め切りは、**8 月 2 日 (金) 講義終了時間**である。

課題 15-1. 以下の (1), (2) のいずれかのプログラムを Scratch で作成せよ。

(1) 例 12-2. 例 14-1. を参考に、2 から 1000 までの数を 1 つ入力し (自然数が入力されることは仮定し)、その数が素数なら ”素数”、合成数なら ”合成数” と表示するプログラム。

(2) 乱数で 1 から 100 までの数 20 個をリストとして作成し、その中の数のうち、最大値と最小値 および、それら 20 個の数の平均値を表示するプログラム。

課題 15-2. 次項のサンプルを参考に、TeX を用いて、問題と解答を作成せよ。ただし、問題と解答は別の紙に出力すること。

問題はこの講義に関らず、情報, 数学以外の問題でも良い。

(サンプル)

第 15 回 まとめの課題

学生番号 SabMxyz 氏名 森 義之

問題 1. f, g を n 次以下の多項式とし、 p_1, \dots, p_{n+1} を $n+1$ 個の相異なる実数とする。このとき、任意の i に対して、 $f(p_i) = g(p_i)$ が成り立つならば、 $f(x) = g(x)$ であることを示せ。

問題 2. A 駅を発車した特急列車は 20 分、急行列車は 25 分、普通列車は 30 分で B 駅に到着する。特急列車は毎時 0 分、急行列車は毎時 20 分と 40 分、普通列車は毎時 x 分、 $x+20$ 分、 $x+40$ 分に A 駅を発車する ($0 < x < 20$)。乗客は時刻表を知らないものとし、A 駅に到着後、最も早く発車する列車に乗るものとする。

このとき、次の問いに答えなさい。ただし、(1), (2) は答えのみでよい。

(1) 乗客が特急列車、急行列車に乗る確率をそれぞれ求めなさい。

(2) 乗客が A 駅に到着してから、普通列車に乗車し B 駅に到着するまでの平均時間を求めなさい。

(3) 乗客が A 駅に到着してから、B 駅に到着するまでの時間の期待値を最小にする整数 x を求めなさい。

(別紙)

第 15 回 まとめの課題 (解答)

解 1. $F(x) = f(x) - g(x)$ とおくと、 $F(x)$ は $i = 1, 2, \dots, n+1$ に対して $F(p_i) = 0$ が成り立つ。 $F(x)$ も n 次以下の多項式なので、

$$F(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

とおくことができる。条件より $F(p_i) = 0$ なので、代入し $n+1$ 個の式を並べて a_i に関する連立方程式を考える。

このときの係数行列はヴァンデルモンドの行列 (の変形) であり、 $i \neq j$ に対して $p_i \neq p_j$ より $p_j - p_i \neq 0$ となる。よって係数行列の行列式の値は 0 ではない。

このことより、係数行列は正則行列で、逆行列をもつため $a_n, a_{n-1}, \dots, a_1, a_0$ は自明な解しかない。よって、 $F(x) = 0$ となるので、 $f(x) - g(x) = 0$ すなわち、 $f(x) = g(x)$ である。

解 2.