DiaGraph†マニュアル

豊田章人　　萩原朋道

京都大学工学部　電気工学第二教室

1991 年 6 月 1 日改訂
(Ver. 1.13)

†Copyright by T. Hagiwara and A. Toyoda (1990)

目 次

1 カレント座標，サブカレント座標，カレント方向 **1**

2 カレント座標の保存 **2**

3 加算器 **2**

4 引出し点 **2**

5 サンプラー **3**

6 線分，矢印 **3**

7 +, − の記号 **4**

8 ラベルの座標へのカレント座標の移動 **5**

9 カレント方向の変更 **5**

10 長方形に囲まれた文字列 **5**

11 文字列の出力とサブカレント座標 **6**
11.1 引出し点，加算器に対する文字列の出力 6
11.2 サンプラーに対する文字列の出力 7
11.3 線分，矢印に対する文字列の出力 7
11.4 ラベルの座標へカレント座標を移動させた時のサブカレント座標 8
11.5 長方形に対する文字列の出力 8
11.6 サブカレント座標に関する注意点 8

12 カレント座標，サブカレント座標の移動 **9**

13 数値ラベル **10**

14 各パラメータの設定 **10**

15 LaTeX のコマンド **11**

16 コメント文 **12**

17 各コマンドの書式 **12**

18 LaTeX のソースファイルへの変換 **12**

19 エラーメッセージ **13**

20 サンプルファイル **14**

図形作成用言語 “DiaGraph” は組版清書システム LaTeX を用いてブロック線図を描くために開発されたものである．マウスなどをインターフェイスとして使う図形用ソフトと違い，DiaGraph ではコマンドを並べることにより線分，矢印，長方形，加算器などの図形のつながり方を指示し，その図形に対応する LaTeX のソースファイルを生成することができる．DiaGraph の利用には LaTeX の図形処理関係のコマンド等の知識を必ずしも要しないが，`picture` 環境，`figure` 環境の説明には一通り目を通しておくことを勧める．

以下，DiaGraph の概要ならびに各コマンドについて説明を行う．コマンド名，ラベル名の大文字，小文字は区別される．また，コマンドの引数の区切りを示す左，右の丸カッコとコンマについては，その前後に空白 (スペース，タブ) を指定しても無視される．

1　カレント座標，サブカレント座標，カレント方向

ブロック線図を手で描くとき，直前に描いた図形の終点から次の図形を描き始めることが多い．DiaGraph のコマンドによるブロック線図の記述もこのようなイメージに従っている．すなわち，直前の DiaGraph コマンドで描いた図形の終点が次のコマンドで描かれる図形の始点として自動的に選ばれる．この，次の図形の始点となる位置をカレント座標と呼ぶことにする．

ブロック線図で矢印を描いたとき，その矢印の信号が目標値を表していることを示すために矢印のそばに “r” などと書くことがある．これに対応して，DiaGraph では，各図形のそばの適当な位置にコメントを書くためのコマンドが用意されている．このコマンドによりコメントが書き出される位置は直前に描いた図形に応じて適切な位置が自動的に割当てられる．この位置のことを以下ではサブカレント座標と呼ぶことにする．

DiaGraph では，カレント座標から上下左右のいずれかの方向に向って図を描くことができる．DiaGraph コマンドには図形を描くときの方向を明示的に指示するコマンドと，そうでないコマンドがある．方向を明示的に指示しないコマンドの場合，図形の描かれる方向は直前の図形の描かれた方向と同じ方向が指定されたものとみなされる．この暗黙の方向のことをカレント方向と呼ぶことにする．方向を明示的に指示するコマンドを利用すると，カレント方向はその方向に更新される．カレント方向の値は，方向を明示的に指示しない図形コマンドだけでなく，前述のコメントを書き出すコマンドが呼ばれた場合にも参照され，コメントの書き出し方 (左詰め，右詰めなど) を定める目的に利用される．

これら 3 つの概念と以下で説明を行う DiaGraph コマンドとを用いて，部品となる図形の相対的な位置関係 (すなわち，つながり方) を指示することにより，DiaGraph は全体の図の様子ならびに大きさを認識し，その大きさの図形に対応する LaTeX ソースファイルを出力する (必要なら上下左右にマージンを付加させることも可能である)．カレント座標，サブカレント座標は，最初に描いた (部品となる) 図形の始点を基準とした相対的な位置を示すものであり，上述のように，図形のつながり方を指示したり，全体の図の大きさを DiaGraph が計算したりする上で使用されるが，これらの座標の絶対的な値そのものについては使用者が意識する必要は全くない．

2　カレント座標の保存

n(*label*)

このコマンドは，カレント座標に *label* という名前のラベルを付け，その値をメモリーに保存する．このラベルは他のコマンドから参照できる[1]．この種のラベルを後述の数値ラベルと区別する場合には，座標ラベルと呼ぶ．ラベルはピリオド，マイナス，等号，ドルマーク (. - = $) 及び数字では始まらず，かつスペース，タブ，左，右の丸カッコ，コンマ，等号，ドルマーク，及びダブルクォーテーション (( ) , = $ ") を含まない半角 30 文字以内の文字列であり，200 個まで定義できる．ラベルが全角文字等 (全角文字および 2 バイト系半角文字) を含むことは差し支えないが，その場合，ラベルの長さの制限においては全角文字等の 1 文字は半角文字に換算して 2 文字分とみなされる．

このコマンドを使用してもカレント座標，サブカレント座標及びカレント方向は変らない．

3　加算器

a(*label*)

このコマンドは，カレント座標を中心に半径が ¥a の値に等しい[2]加算器を描き (¥a などについては **14**を参照)，加算器の中心の座標に *label* という名前を付けて保存する．

図 1: 加算器

このコマンド使用後のカレント座標の移動はなく，カレント方向も不変である．サブカレント座標については **11.1**を参照のこと．

4　引出し点

k(*label*)

このコマンドは，カレント座標を中心に直径が ¥k の値に等しい引出し点を描き，引出し点の座標に *label* という名前を付けて保存する．

コマンド使用後，カレント座標及びカレント方向は変らない．サブカレント座標については **11.1**を参照のこと．

[1] このコマンドで定義されたラベルも，以下で述べる a, k コマンドで定義されたラベルも，区別なく参照することができる．

[2] 厳密には ¥a の値に最も近い LaTeX が出力可能な半径が選ばれる．引出し点に関する ¥k も同様である．

5　サンプラー

s($length$)

このコマンドは，図 2のようにカレント座標からカレント方向にサンプラーを描き，カレント座標を移動する ($length$ は 1，2.5 などの数値である)．つまり，カレント座標からカレント方向に $length$, カレント方向に垂直に $length/2$ だけ移動した点とカレント座標を結んだ線分を描き，カレント方向に $length$ だけカレント座標を移動する．

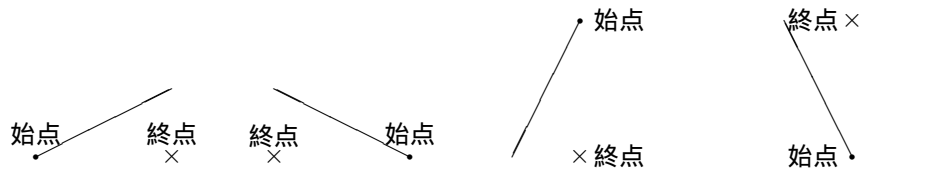

図 2: サンプラー

また

s()

と書くと，$length$ として ¥s の値が指定されたものとしてサンプラーを描く．

なお，コマンド使用後，カレント方向は不変である．サブカレント座標については **11.2** を参照のこと．

6　線分，矢印

r($length$)

このコマンドは，図 3のようにカレント座標から右に $length$ の長さの線分を描き，カレント座標を線分の終端に移動する．左，上，下に線分を引きたい場合は r の代りに，l，u，d の各コマンドを使えばよい．

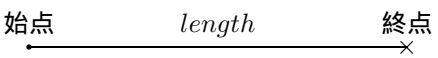

図 3: 線分 (長さを指定)

これらのコマンドにおいて，線分の終点を中心として描かれる，または描かれている加算器がある場合は，$length$ より加え合わせ点の円の半径 ¥a の値だけ短めの線を引く (この場合でも，カレント座標はこの短めの線分の終点ではなく，加算器の中心となる．すなわち，カレント座標は $length$ だけ移動する)．逆に，線分の始点を中心として描かれる，または描かれている加算器がある場合は，加算器の中心から ¥a の値だけ線分を引く方向に移動した点から $length$ より ¥a だけ短めの線を引く (したがって，カレント座標はやはり $length$ だけ動く)．

また

r(*label*)

のように長さ *length* のかわりに加算器などのラベル *label* を指定すると，図 4のように，*label* の座標がカレント座標を始点とする指定した向きの線分の延長上にあれば， *label* の座標まで線分を引き (ただし， その位置を中心とする加算器があるときは上と同様の処理がなされる)，その他の場合は *label* の座標と x 座標が一致する位置まで右に線分を引く[3] ．

図 4: 線分 (ラベルを指定)

また，矢印を引くためには r，l，u，d のかわりに R，L，U，D を使えばよい．

なお，このコマンド使用後，カレント方向は線分，矢印が引かれた方向と同じになる．サブカレント座標については **11.3**を参照のこと．

7 $+$, $-$ の記号

p
m

これらのコマンドは，カレント座標からカレント方向と逆の方向に ¥p の値, カレント方向に向って左に ¥v の値だけ移動した位置に $+$(m の場合は $-$) の記号を付ける．例えば

```
R(4)
a(adder)
p
```

としたときは図 5のような図が描ける．

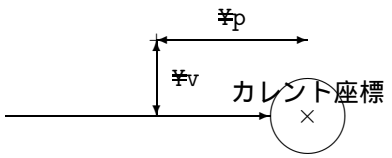

図 5: $+$ の記号

このコマンド使用後，カレント座標，サブカレント座標及びカレント方向は変らない．

[3] l の場合は x 座標が一致するまで左に，u, d の場合は y 座標が一致するまで上あるいは下に線分を引く．また，r 指定にもかかわらず，実際にはラベルで指定された座標がカレント座標よりも左にある場合などは Direction specification error となる．

8　ラベルの座標へのカレント座標の移動

g(*label*)

このコマンドは，n, a, k のいずれかのコマンドで *label* の名前で保存した座標へカレント座標を移す．

カレント方向は変らない．サブカレント座標については **11.4**を参照のこと．

9　カレント方向の変更

t(*dir*)

カレント方向は，線分や矢印を引けば，その時点でそれらを引いた方向にかわるが，それ以外の時にカレント方向を変えたい場合は上のコマンドを使う．方向の指定は，右にしたいとき *dir* に r を，左にしたいとき *dir* に l を，上にしたいとき *dir* に u を，下にしたいとき *dir* に d を書けばよい．なお，カレント方向の初期値は右となっている．

このコマンドを使用してもカレント座標，サブカレント座標は変らない．

10　長方形に囲まれた文字列

b(*xlength*, *ylength*, "*comment*")

このコマンドは，図 6のようにカレント方向に左右が *xlength*，上下が *ylength* の長方形を描き，その中央に *comment* という文字列を書く (この文字列には '$ G(s) $' など

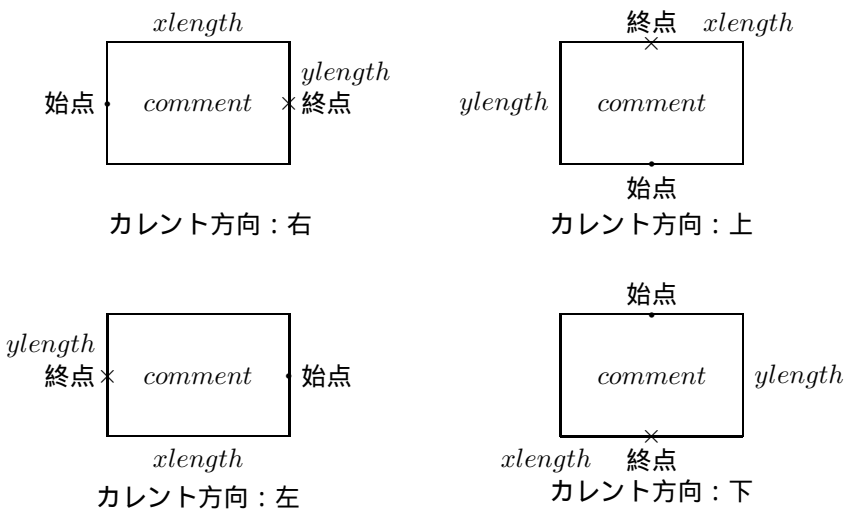

図 6: 長方形に囲まれた文字列

のように LaTeX のコマンドを含めることも可能であり[4]，漢字も含められるが，‘"’ を含むことは許されない)．$xlength, ylength$ は数値，"はダブルクォーテーションである．また，b のかわりに B を使えば，¥d の値に等しい間隔の破線で長方形が描ける．

コマンド使用後，カレント座標は図 6の $\times$ の位置に移るが，カレント方向は不変である．サブカレント座標に関しては **11.5**を参照のこと．

11　文字列の出力とサブカレント座標

c("*comment*")

このコマンドでは，直前に描いた図形のサブカレント座標に $comment$ という文字列を出力する．" はダブルクォーテーションである．以下に，サブカレント座標が更新されるコマンド (**12**も参照のこと) およびそのコマンドの直後に c コマンドを使用した場合に文字列 $comment$ が出力される位置について説明する．

なお，このコマンド (c コマンド) を実行してもカレント座標，サブカレント座標，カレント方向は変化しない．

11.1　引出し点，加算器に対する文字列の出力

引出し点，加算器を描いたときのサブカレント座標は，カレント座標からカレント方向に向って，左に ¥c だけ移動した位置となる．そして，¥c≥0 の場合は，カレント方向が右のときに文字列の下端中央がサブカレント座標と一致，左のときに文字列の上端中央がサブカレント座標と一致，上のときに文字列の右端中央がサブカレント座標と一致，下のときに文字列の左端中央がサブカレント座標と一致するように文字列が出力される．¥c<0 の場合は，カレント方向が右のときに文字列の上端中央がサブカレント座標と一致，左のときに文字列の下端中央がサブカレント座標と一致，上のときに文字列の左端中央がサブカレント座標と一致，下のときに文字列の右端中央がサブカレント座標と一致するように文字列が出力される．引出し点の場合について，¥c≥0 のときの文字列の出力位置を示すと，図 7のようになる．ただし，$\times$ はサブカレント座標を表す．

図 7: 引出し点に対する文字列の出力 (¥c≥0)

[4]ただし，特殊な (‘fragile な’) コマンドの場合には ¥protect コマンドを併用することが必要となる可能性がある．詳しくは LaTeX の説明書等の ¥protect コマンドの項を参照されたい．

11.2　サンプラーに対する文字列の出力

サンプラーを描いたときのサブカレント座標と文字列の出力位置の関係を示すと図 8, 9 のようになる．ただし，× はサブカレント座標を表す．

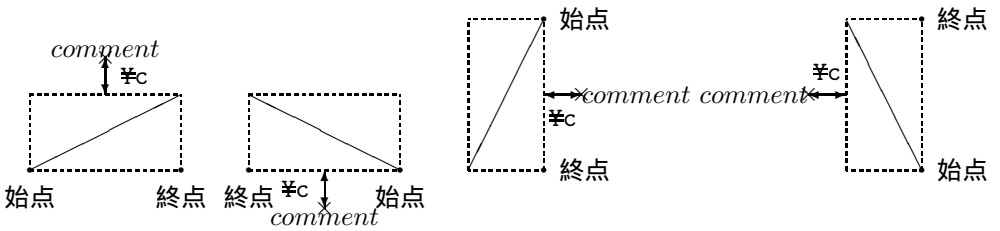

図 8: サンプラーに対する文字列の出力 (¥c≥0)

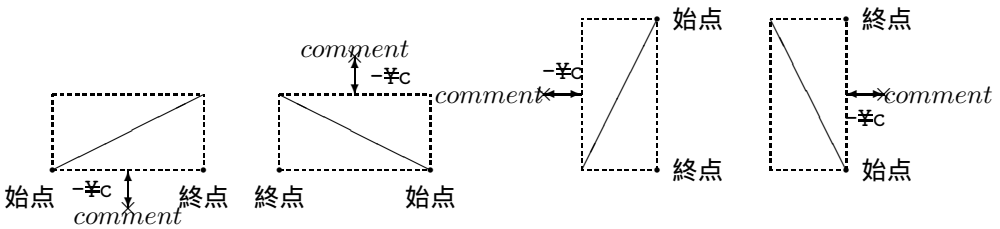

図 9: サンプラーに対する文字列の出力 (¥c<0)

11.3　線分，矢印に対する文字列の出力

線分，矢印を描いたときのサブカレント座標は，描いた線分，矢印の中点[5]からカレント方向に向って左に ¥c だけ移動した位置となる．文字列の出力位置とサブカレント座標との関係については **11.1**と同じである．矢印について実際の出力位置を示すと図 10のようになる．ただし， × はサブカレント座標を表す．

図 10: 矢印に対する文字列の出力 (¥c≥0)

[5]線分，矢印の始点または終点に加算器があり，線分の長さがコマンドの引数に指定した長さより短く調整された場合は，この短くなった線分，矢印の中点とする．ラベルを引数とした場合も同様である．

11.4　ラベルの座標へカレント座標を移動させた時のサブカレント座標

g コマンドを利用してラベルの座標にカレント座標を移動させたときには，サブカレント座標もそれにつれて自動的に移動する．サブカレント座標の位置は，移動後のカレント座標から ¥c の値だけカレント方向に向かって左に動いた点となる (**11.1**の場合と同様)．c コマンドによる文字列の出力位置とサブカレント座標との関係も **11.1**の場合と同様である．

11.5　長方形に対する文字列の出力

長方形を描いたときのサブカレント座標は，始点と終点を結んだ線分の中点から，カレント方向に向って， ¥c≥0 のときは左，¥c<0 のときは右の辺に対して垂線を引いて，辺との交点から |¥c| だけ長方形の外側の位置になる．文字列の出力位置とサブカレント座標との関係は **11.1**と同じである．実線の長方形について，¥c≥0 のときの文字列の出力位置を示すと，図 11のようになる．ただし，× はサブカレント座標を表す．

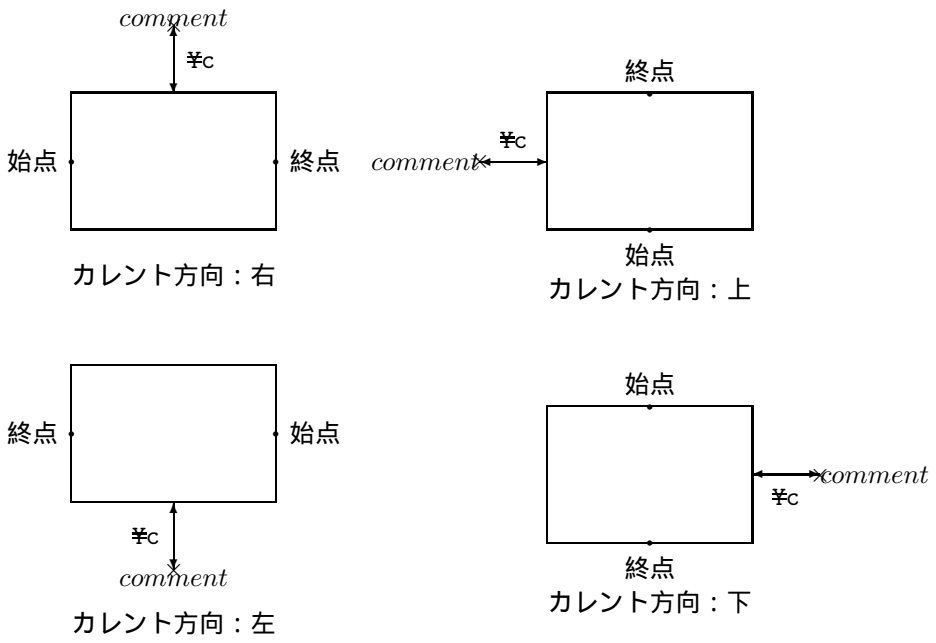

図 11: 長方形に対する文字列の出力 (¥c≥0)

11.6　サブカレント座標に関する注意点

11.1～**11.5**において，¥c の値は各コマンド (c コマンドのことではなく，そこで説明しているサブカレント座標の値が更新される a, k, s, r などのコマンド) を実行したときの値がとられる．したがって，これらのコマンドの実行後，実際に c コマンドで文字列を

出力するまでの間で ¥c の値を変更しても，一部の使用者が期待するであろう形では動作しないので注意しなければならない．例えば，¥c=0.2 となっている状態で

```
r(1)
¥c=0.5
c("$ x $")
d(1)
c("$ y $")
```

とした場合を考える．このとき，コマンド c("$ x $") の実行時に出力される文字列 '$x$' の基準位置を与えるサブカレント座標は，この c コマンドより以前で最後に実行されたサブカレント座標を更新するコマンドである r(1) が実行された時点で既に定まってしまっている．この r コマンドが実行されたときの ¥c の値はまだ 0.2 のままであるため，結局，¥c=0.5 の効果は c("$ x $") の出力結果には現れず，その効果は c("$ y $") の部分ではじめて現れることになる．c("$ x $") に対する出力結果を ¥c=0.5 に対応する位置に得るためには

```
¥c=0.5
r(1)
c("$ x $")
d(1)
c("$ y $")
```

というように，先に ¥c=0.5 の状態に変更した上でサブカレント座標を更新するコマンド (この例では r(1) コマンド) を実行しなければならない．

12　カレント座標，サブカレント座標の移動

x(len)

このコマンドは，カレント座標を x 軸の正の方向に len ($len < 0$ のときは負の方向に $|len|$) だけ移動する．また，

x($label$)

のように len のかわりにラベル $label$ を指定すると x 座標が $label$ のそれと等しくなる位置まで，カレント座標が x 軸に平行に移動する．カレント座標を y 軸の方向に移動するときは x コマンドのかわりに y コマンドを使えばよい．これらのコマンド実行後はサブカレント座標も移動し，カレント座標と同じ値に設定される．

また，カレント座標を移動させずにサブカレント座標のみを x, y 軸の方向に移動するときは x, y コマンドの代わりに X, Y コマンドを使えばよい．$label$ を指定した場合も同様である．

これらのコマンドはカレント方向を変化させない．

13　数値ラベル

$*numlabel*=*length*

このコマンドはある長さを名前(ラベル)で参照できるようにするものであり，*numlabel* はラベル，*length* は数値である．例えば，

```
$arrowlength=2
R($arrowlength)
x(-$arrowlength)
```

は

```
R(2)
x(-2)
```

と等価である(\$*numlabel* の値が負のときも-\$*numlabel* のような指定は可能である)．このようなラベルを前述の座標ラベルと区別する場合には数値ラベルと呼ぶ．数値ラベルは必ず\$で始めなければならない．この先頭の\$を除いた部分のラベル名に関する規則(使用可能な文字，長さの制限)，及びラベルの個数の上限は座標ラベルの場合と全く同じである．数値ラベルから先頭の\$を除いた部分と同じ名前の座標ラベルが用いられることは差し支えない．

数値ラベルの利用により，複数の(部品となる)図形の長さを同時に変更することが容易に行える．

このコマンドを実行しても，カレント座標，サブカレント座標，カレント方向は変化しない．

14　各パラメータの設定

¥a (加算器の半径), ¥k (引出し点の直径), ¥c (文字列の出力位置), ¥p, ¥v (+, − の出力位置), ¥d (破線の間隔), ¥s (サンプラーの長さ) のデフォルト値は，それぞれ，0.15，0.08，0.2，0.3，0.2，0.1，0.6 となっている．

いま，加算器の半径を 0.3 に変えたいときは

```
¥a=0.3
```

とすればよい．また，デフォルト値に戻したいときは

```
¥a
```

とすればよい．他のパラメータについても同様である．

上にあげたパラメータの他にも，DiaGraph には ¥l, ¥r, ¥t, ¥b, というパラメータが用意してある．¥l は図の左端からのレフトマージンの値，¥r は図の右端からのライトマージンの値，¥t は図の上端からのトップマージンの値，¥b は図の下端からのボトムマージン

の値を表す．デフォルトはすべて 0 で，値を変更したいときは，¥a に対する場合と同様に行えばよい[6]．

これらのコマンドを実行してもカレント座標，サブカレント座標，カレント方向は変化しない．

なお，これらの ¥* のコマンドのみを記述したファイル (コンフィギュレーションファイルと呼ぶ) を diagraph.cnf という名前で作成しておくとデフォルト値は上に示した値ではなくこのファイルで指定された値となる．コンフィギュレーションファイル diagraph.cnf のサーチパスは以下の順序である．

1. カレントディレクトリの diagraph.cnf
2. 1. になければ環境変数 DIAGRAPH で指定されたディレクトリの diagraph.cnf
   ただし，MS-DOS においては，ディレクトリ名が全角文字等 (全角文字および 2 バイト系半角文字) で終わる場合には環境変数の設定時にディレクトリ名の最後に '¥' を添えておくこと．

   ```
   set DIAGRAPH=figtable
   set DIAGRAPH=¥図表¥
   set DIAGRAPH=A:.
       (MS-DOS で A: ドライブのカレントディレクトリを指定するとき)
   ```

3. 1., 2. のいずれのディレクトリにもなければ，本プログラム diagraph の存在するディレクトリ (起動ディレクトリ) の diagraph.cnf
   (ただし，3. は MS-DOS かつ Ver. 3.0 以上の場合に限る)

15 LaTeX のコマンド

```
%¥put("x,"y){……}
```

LaTeX のコマンドを直接使いたいときは，LaTeX のコマンドの前に%の記号を付ければよい．また，そのコマンド中に"x, "y という文字列を使うとその部分それぞれはカレント座標の x 座標, y 座標に置換えられる．" はダブルクォーテーションであり，" と x または y の間に空白 (スペースおよびタブ) を入れてはならない．

このコマンドを使用しても，カレント座標，サブカレント座標及びカレント方向は不変である．

[6] 図の上下左右の端とは，(1) c コマンドで書いた文字列，(2) LaTeX のコマンドを直接使用して得た図形，文字列，を除いた部分の端をいう．Diagraph の出力する LaTeX ソースファイルは，この残りの部分にマージンを付加した大きさを占めるものとして LaTeX に処理される．したがって，上の (1),(2) の部分がここでいう図の上下左右の端よりもはみ出している場合には (このはみ出した部分が他の文章などと重ならないように) これをカバーするようにマージンを設定することが望ましい．また，たとえば左マージンを大きくとることにより図全体の位置を右に平行移動させる，という使い方もできる．

16　コメント文

%%·····

%%で始まる行は%を 1 つ取り除いた後，そのまま LaTeX のソースファイルに出力されるので LaTeX ソースファイルにおけるコメント行となる．同様に，上述のすべてのコマンドにおいて，その後ろに%%で始まるコメントを付加することができる．ただし，この場合にはそのコメントは LaTeX のソースファイルには出力されない．コマンドの途中 (ただしダブルクォーテーションで囲まれた部分を除く) で%%を使うことは許されない．

¥%·····

一方，¥%で始まる行は DiaGraph のソースファイルのみに対するコメント行であり，LaTeX のソースファイルにはなんら影響を与えない．すなわち，このコメントは LaTeX のソースファイルには出力されない．

これらのコマンドによりカレント座標，サブカレント座標，カレント方向は変化しない．

17　各コマンドの書式

上に説明したコマンドを DiaGraph のソースファイルに書くときは，1 行に

[␣␣···␣]1 つのコマンド [␣␣···␣][%%·····] CR

あるいは

[␣␣···␣]%%····· CR

[␣␣···␣]¥%····· CR

の形で書くこと．ただし，␣␣···␣ はスペースまたはタブからなる空白，····· はコメント文字列，CR は改行キーの入力を表す．また，[と] で囲まれた部分は省略可能な部分を表す．つまり，1 行に 2 つ以上のコマンドを書いてはいけない．また，1 つのコマンドを 2 行以上にわたって書くこともできない．さらに，コマンドの 1 行は 256 文字以内でなければならない．

なお，コマンド，コメントの直前に空白 (␣␣···␣ で表される部分) がある場合，その空白は LaTeX のソースファイルにもそのまま出力される．とくに，上記の書式以外に，空白のみからなる行を書くことは許されている．

18　LaTeX のソースファイルへの変換

DiaGraph のソースファイルから LaTeX のソースファイルへの変換手順は次のとおりである．

1. 上に説明した各コマンドを使用して DiaGraph のソースファイルを作る．

2. diagraph␣[-h]␣[パス指定] ファイル名 [. 拡張子] と入力する．

ただし，[と] で囲まれた部分は省略可能である (拡張子のデフォルトは grp である)．1., 2. の操作を行えば，DiaGraph のソースファイルと同名の tex という拡張子のついた LaTeX のソースファイルがカレントディレクトリにできる．もし，そのような名前の.tex ファイルが既にカレントディレクトリに存在するときは，そのファイルのバックアップを拡張子を org として作成する．(ただし，この名前の.org ファイルが既に存在しても，そのファイルは無条件に上書されるので注意すること．)

2. でオプション -h を指定した場合，デバック情報つきで LaTeX のソースファイルが作成される．すなわち，カレント座標が変更されるコマンドまたは LaTeX のコマンドが実行されるたびに，そのコマンドの DiaGraph ソースファイル中での行番号とそのコマンドの実行後のカレント方向が，(更新後の) カレント座標の位置に表示される．この情報により，部品となる図形の DiaGraph ソースファイル中での位置を容易に知ることができる．

なお，MS-DOS の場合，以下の 2 点に注意すること．

- 上の手順 2. において DiaGraph のソースファイルをパス名とともに指定するときは，LaTeX と同様，'¥' のかわりに '/' をディレクトリの区切り記号として使うこと．
- 上述のファイルバックアップ (拡張子 org のファイルの作成) が実行されるためには，COMMAND.COM を環境変数 COMSPEC で指定しておかなければならない．

  ```
  set COMSPEC=A:¥COMMAND.COM
  ```

  このとき，COMSPEC で指定されたディレクトリ (上の例では A: ドライブのルートディレクトリ) に COMMAND.COM が存在していなければならないことはもちろんである．

DiaGraph では，LaTeX の picture 環境の部分のみを生成する．したがって，その前後に必要な LaTeX のコマンドを記述したファイルは各自で作成しなければならない (DiaGraph の生成した LaTeX ソースファイルを ¥input コマンドなどを利用して読込むようにすれば便利である)．特に，DiaGraph のデフォルトの長さは (コンフィギュレーションファイル diagraph.cnf を使って変更しない限り) cm を長さの単位とした場合に適当となるように考えられており，LaTeX のデフォルトの単位 pt を用いた場合には短すぎるので，¥setlength{¥unitlength}{1cm} などの指定を行うこと (**20**参照)．

19 エラーメッセージ

以下に DiaGraph のエラーメッセージ (アルファベット順) とその説明を示しておく．

- Configuration file error
  コンフィギュレーションファイル diagraph.cnf に記述の誤りがある．

- Direction specification error
  方向の指定に矛盾がある．(たとえば，r(*label*) において *label* の座標がカレント座標より左寄りにあるときなど)

- Length specification error
  長さの指定方法に誤りがある．(正の値のみが許される場合に負の値を指定したときなど)

- Memory for coordinate labels exhausted
  座標ラベルの個数が上限を越えている．

- Memory for numerical labels exhausted
  数値ラベルの個数が上限を越えている．

- Multiply defined label '$\cdots$'
  ラベル $\cdots$ が重複して定義されている．

- Syntax error
  コマンドの記述に誤りがある．

- Undefined label '$\cdots$'
  ラベル $\cdots$ が定義されていない．

- Warning: Redeclaration of length '\$$\cdots$'
  数値ラベル \$$\cdots$ が再定義された (誤って同じ数値ラベル名を二度使用したのか，意識的に再定義したのか区別できないため警告を出している．したがって，一概にエラーとは限らない．意識的に再定義したものとみなして処理される .)

20　サンプルファイル

以下のファイルの内容を使い，まず DiaGraph で example.tex を作り，次に LaTeX で demo.dvi を作ってみてください．

demo.tex の内容

```
¥documentstyle[12pt]{article}
¥begin{document}
¥setlength{¥unitlength}{1cm}
¥begin{figure}[h]
    ¥centering
    ¥input{example}
    ¥caption{Feedback system¥label{fig}}
¥end{figure}
¥end{document}
```

example.grp の内容 (コロンより右は説明で，実際には不要)

```
¥%                          : Comment for DiaGraph source file
¥% Example of
¥%   DiaGraph Source File
¥%

%% Unity Feedback System    : Comment for LaTeX source file
% ¥typeout{}                : Execute LaTeX command
% ¥typeout{Hello!}
% ¥typeout{}

¥c=0.4                      : Position of the text generated by c command
R(1)                        : Right arrow with length 1
c("$ r $")                  : Write 'r' over the arrow
p                           : Write a + mark beside the arrow
a(add)                      : Draw an adder, and label its center as "add"
R(1)
c("$ e $")
b(2,1,"$ C(s) $")           : Draw a box of size 2 × 1 and C(s) in it
c("Controller")             : Write 'Controller' over the box
R(1)
c("$ u $")
b(2,1,"$ P(s) $")
c("Plant")
r(1)                        : Draw a line of length 1 to the right
k(knot)                     : Draw a knot labeled "knot"
c("$ y $")
R(1)
g(knot)                     : Move current coordinate to the coord. of "knot"
d(2)
l(add)                      : Draw a line to the left up to the x-coord. of "add"
U(add)                      : Draw upward arrow up to the y-coord. of "add"
m                           : Write a − mark beside the arrow
```