点字教材作図システム Bplot (CUI バージョン) 解説

開発:藤芳 衛・石田 透・大澤彰子・藤芳明生発行:テストと学習環境のユニバーサルデザイン研究機構

(2016年3月7日)

連絡先: 03-6909-9954 e メール fujiyosi@udlte.or.jp URL: http://www.udlte.or.jp

点字教材作図システム Bplot (CUI バージョン) 解説

はじめに	5
(1) 点字教材作図システム Bplot (CUI バージョン)の開発目的	5
(2) 開発の背景	5
(3) Bplot の特長	6
第1章 Bplot の操作	8
1.1 ファイル構成	8
1.2 2種類の動作モード	8
1.3 対話型モード	8
1.3.0 メインメニュー	8
1.3.1 システム設定	9
1.3.2 コンソールサブメニュー	
1.3.3 テストモード	
1.3.4 デバッグモード	
1.3.5 リストサブメニュー	
1.3.6 アウトプットサブメニュー	
1.3.7 Bplot と EDEL ファイルの相互変換サブメニュー	
1.3.8 ポストスクリプトサブメニュー	
1.3.9 ヘルプサブメニュー	
1.4 フィルタモードエラー! ブックマークが定義され	•
1.4.1 ファイル出力エラー! ブックマークが定義され	
1.4.2 直接印字エラー! ブックマークが定義され	•
1.5 エラーメッセージ	
第2章 作図コマンドプログラムの作成法 (基礎編)	
2.1 作図コマンドの概要	
2.2 点の種類と点を打つコマンド	
2.3 定形的な直線系図形を描く	
2.4 不定形な直線系図形を描く	
2.5 定形的な曲線系図形を描く	
2.6 不定形な曲線系図形を描く	
2.7 図に点字を書き込む (1.47年)	
第3章 作図コマンド解説 (上級編)	
3.1 ウィンドウの設定	
3.2 ウィンドウの設定	37

3.2.1	ウィンドウの表示	37
3.2.2	b5 ウィンドウ設定コマンド	38
3.2.3	a4 ウィンドウ設定コマンド	38
3.2.4	origin 原点の設定	38
3.2.5	縦向きウィンドウ設定コマンド	41
3.2.6	横向きウィンドウ設定コマンド	41
3.2.7	裏面点字設定コマンド	42
3.2.8	window 設定コマンド	43
3.2.9	frame コマンド	43
3.3 変数	数と数式定義コマンド	46
3.3.1	変数定義コマンド	46
3.3.2	マクロ変数定義コマンド	50
3.4 点气	字印字コマンド	53
3.4.1	点字フォーマットコマンド	53
3.4.2	braille (brl) NABCC 点字印字	53
3.4.3	仮名書き点字印字	
3.4.4	印字ヘッド移動コマンド	57
3.4.5	点字ドキュメント印刷コマンド	57
	ージ送りコマンド	
	ページ送りコマンド	
3.5.2	ページ送り制御コマンド	58
3.6 点页	の大きさや点間隔設定コマンド	58
3.6.1	dot 点の大きさ指定コマンド	59
3.6.2	点線の点間隔設定	
	plot 打点コマンド	
3.6.4	point 特大点コマンド	63
3.6.5	design 破線設計コマンド	66
3.7 注彩	沢コマンド	
3.7.1	注釈行設定コマンド	
3.7.2	注釈領域設定コマンド	70
3.8 直統	泉系定形図形の作成	74
3.8.1	line 直線印字コマンド	
3.8.2	ハッチング用点線コマンド	74
3.8.3	arrow 矢印コマンド	
3.8.4	dart 矢印コマンド	75
3 8 5	应 押 軸 作 図 コマンド	76

	3.8.6	box 箱形作図コマンド	79
	3.8.7	diamond ひし形作図コマンド	79
	3.8.8	polygon 正多角形作図コマンド	79
	3.8.9	parallelogram 平行四辺形作図コマンド	79
	3.8.10	rectangle 長方形作図コマンド	80
	3.9 直	「線系不定形図形の作成	80
3.	.10 曲流	線系定形図形の作図	85
	3.10.1	circle 円の作図コマンド	85
	3.10.2	ellipse 楕円作図コマンド	85
	3.10.3	oval 楕円の弧の作図コマンド	85
	3.11	曲線系不定形図形の作成	86
3.	.12 関	数グラフの作成	94
	3.12.1	軸が y 軸に平行な放物線	94
	3.12.2	軸が x 軸に平行な放物線	97
	3.12.3	双曲線1	00
	3.12.4	直角双曲線1	03
	3.12.5	多項式のグラフ 1	06
	3.12.6	三角関数のグラフ1	09
	3.12.7	指数対数のグラフ1	.12
3.	.13 統	計グラフ1	16
	3.13.1	箱ひげ図1	.16
3.	.14 貼	り付けと回転コマンド1	21
	3.14.1	paste 貼り付けコマンド	21
3.	.15 点(の二重打ち等を防ぐコマンド1	25

はじめに

(1) 点字教材作図システム Bplot (CUI バージョン)の開発目的

点字教材作図システム Bplot (CUI バージョン)は重度視覚障害を有する盲学校の教員や 点字図書館の職員、学生や研究者及び各企業の社員等がコンピュータを活用して各種の触 読図(手で触察して把握する図)を自立的に作成可能にするために開発されたシステムです。 高解像度の点字プリンタの使用により、点字教科書をはじめ点字教材の精密な図を作成可 能にするために開発されました。

作図システム Bplot には CUI バージョンとユニバーサルデザインバージョンの 2 つのシステムがあります。Bplot (CUI バージョン)は特に重度視覚障害者が出来る限り使いやすいように設計されています。

本解説は第1章で Bplot (CUI バージョン)の操作方法を解説します。第2章で初歩的な作図コマンドプログラムの作成方法を紹介します。第3章で100種を超える作図コマンドを機能別に解説します。第4章で点字教材の種類別に作図プログラムの作成方法を詳細に解説します。

本解説に搭載されている作図プログラム例はすべて実際に印字可能です。

Bplot (CUI バージョン)と Bplot (ユニバーサルデザインバージョン)は、テストと学習環境のユニバーサルデザイン研究機構のホームページ http://www.udlte.or.jp においてあります。適宜ダウンロードしてお使い下さい。

なお、Bplot (CUI バージョン)は Windows 用と DOS 用の 2 つがあります。 コンピュータ環境に応じてお使い下さい。

(2) 開発の背景

点字教材作図システム Bplot (Braille plotter)は、大学入試センター試験の点字問題の触読図(触察して把握する図)をコンピュータを活用して重度視覚障害職員も自立的に作成可能にするため 2001 年に開発されました。従来、内外の点字教科書や点字教材の触読図の作図ソフトウェアは、晴眼者がマウス等を使用してコンピュータ画面に描画して作成するGUI (Graphical User Interface)であるため重度視覚障害職員には使用不可能でした。重度視覚障害職員は点字問題の触読図を実際に手で触察して校正する触読校正作業しか従事できませんでした。しかし、少しでもわかりやすい触読図を作成するためには、はじめから重度視覚障害職員も作成に参加することが必要とされます。

このため、グラフィックプロッタ制御コマンド方式の作図言語と作図言語の処理系を開発することにより Bplot を開発しました。エディタを使用して作図コマンドを記述すれば重度視覚障害職員も初めて CUI (Character User Interface)で作図が可能となりました。

しかし、晴眼職員には日頃使い慣れた GUI に比べ CUI は使い勝手が悪いため、2007 年に GUI にも対応した Bplot2 を開発しました。

このBplot2は作図言語を媒介としてCUIとGUIとが相互に連動することに着目し、2011

年にユニバーサルデザインの作図システム Bplot (ユニバーサルデザインバージョン)を開発しました。コンピュータ画面にエディタウィンドウとキャンバスウィンドウ及びメッセージウィンドウを備え、エディタウィンドウに CUI で作図コマンドを記述すればキャンバスウィンドウにその図が描画されます。一方、マウスやタブレット等を使用してキャンバスウィンドウに GUI で描画すればその図の作図コマンドが自動的に作成され、エディタウィンドウに加筆されます。また、発生したエラー等はメッセージウィンドウに表示されます。

Bplot (ユニバーサルデザインバージョン)の開発により、重度視覚障害者も GUI による作図が可能となりました。しかし、GUI で作図するためには描画した図を即座に見ながら作図する WYSWYG (What You See is What You Get) が求められます。確かに晴眼者にとっては GUI は WYSWYG です。だが、重度視覚障害者にとっては GUI は WYSWYG ではありません。図を作成した後に点字プリンタで打ち出さなければ確認することができません。

現在、高解像度の触知ディスプレイも研究されています。しかし、実用化は相当先のことになるものと推測されます。それに実用化されたとしても触覚の弁別域は視覚よりもかなり大きいため、GUIにより精密な作図は困難です。

このため、2015年に重度視覚障害者にとってより使い勝手の良い CUI を実現した Bplot (CUI バージョン)を開発しました。

主要な改良点は第一に各作図コマンドを構成するパラメータは従来数値で指定していました。Bplot (CUI バージョン)では変数と変数や関数を含む数式で記述可能となりました。変数の値を調整するだけで図の形やレイアウトを一斉に修正することが可能です。第二に画像データベースに蓄積された既存の画像を Bplot に読み込み修正して活用することが可能となりました。

重度視覚障害職員も図やグラフを立体コピー機等で浮き出した図にして触察しながら自立的に作図が可能となります。実際に重度視覚障害職員もセンター試験の全教科・科目の点字問題の図の作成が自立的に可能であることが実証されています。

(3) Bplot の特長

- 1) Bplot(CUI バージョン)を活用すれば重度視覚障害者も自立的に点字教材の触読図の作成が可能となります。従来、重度視覚障害者は晴眼者が作成した図を校正するだけしかできませんでした。しかし、Bplot (CUI バージョン)の開発により重度視覚障害者自身も直接作成することが可能となりました。
- 2) Bplot (CUI バージョン) と Bplot (ユニバーサルデザインバージョン) を活用すれば重度視覚障害者と晴眼者とが情報を共有し、協力して点字教材の作図作業を行う統合的職場環境を実現します。
- 3) 重度視覚障害の教員も理数系の科目等、授業で必要な点字教材を自立的に準備すること

が可能となります。実際に理療科教員の方も解剖図等点字教材の作成にすでに活用しています。

4) 学生や研究者も実験結果のグラフを実際に触読図にして触察することによりグラフの傾向をより的確に把握することが可能となります。また、触読図のコマンドプログラムを保存しておきレポートや論文の作成に活用することも可能となります。

第1章 Bplot の操作

1.1 ファイル構成

Bplot は bplot.exe と bplot.sys 及び bplot.hlp の 3 つのファイルで構成されています。 bplot.exe は Bplot の実行ファイルです。

bplot.sys は Bplot のシステム環境を保存しているファイルです。使用する点字プリンタ (ESA721、ESA600G)、プリンタ用紙のサイズ(8 インチ用紙、10 インチ用紙)、縦向き・横向き(portrait、landscape)の指定、各種ファイル名等、前回使用していたシステム環境が保存されています。

bplot.exe がバージョンの違う bplot.sys を読み込みますと正常に動作しないことがあります。その場合は \mathbf{r} オプションを付けて bplot.exe を起動して下さい。bplot.exe が初期化され、正常に動作するようになります。

念のため bplot.sys を抜いておきます。bplot.exe を 1 度実行しますと bplot.sys が自動的 に作成されます。

bplot.hlp は英文のオンラインヘルプメッセージが収められたテキストファイルです。 Bplot のメインメニューでヘルプを選択してから作図コマンド名を入力しますとこの bplot.hlp ファイルが参照されヘルプメッセージが表示されます。

1.2 2種類の動作モード

Bplot は対話型モード(interactive mode) とコマンドプロンプトのフィルタモード(filter mode)の 2 つのモードで動作します。

(1) 対話型モード

bplot.exe を起ち上げて画面のメニューに基づいて操作して下さい。起ち上げ時メインメニューが画面に表示されます。メインメニューの下に

Bplot: s c t d l o e p h q?

という1文字コマンド入力行が表示されます。

メインメニューには 10 個の 1 文字コマンドが表示されます。メニューのコマンド番号またはコマンドの頭文字 1 文字を Bplot の 1 文字コマンド入力行から入力すると対応するサブメニューに入ります。

詳しくは 1.3 節で解説します。

1.3 対話型モード

1.3.0 メインメニュー

Bplot の起ち上げ時メインメニューと 1 文字コマンド入力行が表示されます。

メインメニューには 10 個のサブメニューがあります。 メインメニューは星印(アステリスク)で囲まれています。 メインメニューは 3 行からなっています。

1 行目 1: Set 2: Console 3: Test 4: Debug

2 行目 5: List 6: Output 7: Edel 8: Postscript

3 行目 9: Help 0: Quit

メインメニューの下に1文字コマンド入力行

bplot: s c t d l o e p h q?

が表示されます。

1 文字コマンド入力行の? (クエスチョンマーク)の右にサブメニュー番号またはサブメニューの頭文字を入力するとサブメニューに入ります。

メインメニューを抜けて Bplot を終了する時は、0 または q (quit)を入力して下さい。

メインメニューの 1 行目「1: Set 2: Console 3: Test 4: Debug」のサブメニューは作 図システム Bplot の設定や作図コマンドのテストをするための 4 つのサブメニューです。

2行目の「5: List 6: Output 7: Edel 8: Postscript」のサブメニューは作図コマンドプログラムを検索したり、作図コマンドプログラムを実行して触読図を印字したり、作図ソフトウェア「エーデル」のデータファイルと Bplot コマンドプログラムファイルの相互変換をしたり、あるいは画像を通常のプリンタで印字するためにポストスクリプトファイルを作成したりするための 4 つのサブメニューです。

3行目のサブメニュー9: Help 0: Quit は作図コマンドのオンラインヘルプを見るためのサブメニューです。また、Bplot を終了するためのサブメニューです。

1.3.1 システム設定

メインメニューから1またはs(set)を選択しますと設定メニューに入ります。

setup: f p v h q?

設定メニューで p (printer)を選択しますと点字プリンタの選択メニューが表示され、次の行はプリンタの選択待ち状態となります。

現在対応している点字プリンタは、解像度が 73dpi の片面印字の ESA721 と 100dpi の両面印字の ESA600 の 2 機種です。なお、解像度は点字プリンタ用紙上 1 インチ辺りに印字可能な点の数です。通常の点字プリンタの解像度は 20dpi または 25dpi です。

設定メニューから v (view)を選択しますと現在のシステムの設定状態が表示されます。 例えば点字プリンタ ESA721 と通常 A4 サイズと呼ばれる点字用紙を使用する場合の Bplot のシステム設定状態は次のとおりです。

grid (0 599 0 790) frame (0 599 0 790) window (-10.35 10.35 -12.35 12.53) as a4 and portrait printer (ESA721) is set up. v/h ratio: 1.0000

grid は点字プリンタが持つ横方向と縦方向の印字範囲を表示します。最初の 2 つの数値は横方向の点の位置の範囲です。次の 2 つの数値は縦方向の点の位置の範囲です。

frame は現在 grid 上の印字可能な範囲を表示します。数値は grid と同様、横または縦の 点の位置の範囲を示します。例えばプリンタ用紙の上半分や横半分に、印字可能範囲を制 限すると frame の範囲は grid 範囲より狭くなります。

window は grid に対応する座標の割り付け状態を表示します。最初の 2 つの数値は横方向の座標範囲を示します。次の 2 つの数値は縦方向の座標範囲を表示します。

その他、使用する点字用紙の大きさ(A4、B5)、点字プリンタ用紙の使い方(縦向き、横向き(portrait、landscape))、使用する点字プリンタ名(ESA721、ESA600G)が表示されます。また、v/h ratio: は点字用紙上のウィンドウ座標系の縦と横の比です。例えば縦横比が1.0000 と大きく違っている場合に円を描けば、プリンタ用紙上には縦または横にひずんだ楕円形が描かれます。

1.3.2 コンソールサブメニュー

メインメニューの1行目の4つのサブメニューのうち、

2: Console 3: Test 4: Debug

は作図コマンドをテストするために用意されています。

もしこれらのサブメニューからメインメニューに戻りたい時はコマンド「bye」を入力して下さい。終了コマンド 0 または q (quit)を入力すると B plot を終了することが出来ます。メインメニューから 2 または c (console)を入力するとコンソールサブメニューに入ります。 画面に C cosole C N:と表示されます。 C console C N:の C はコンソールサブメニューの入力受け付け番号です。 コマンドを一つ入力するたびに C 1 ずつ増加します。

console N:の後に作図コマンドを入力し、エンターキーを押しますとその作図コマンドが実行され、作図コマンドが点字プリンタの印字命令に変換され出力されます。

もし出力先を変えずにそのまま実行しますとプリンタの印字命令が直接画面に出力され

ます。画面には意味の分からない文字列としてそのまま表示されます。

実際に点字プリンタに出力して印字結果を見たい場合は、

- (1)点字プリンタに電源を投入し、点字プリンタ用紙をセットしておいて下さい。
- (2) Bplot の起ち上げ時に、bplot > aux と入力し、出力先を aux (auxilial interface device) に変更しておいて下さい。aux は点字プリンタ ESA721 または ESA600G 等が使用している RS232C インタフェースの出力先です。

この後、コンソールモードで作図コマンドを入力するたびにそのコマンドが印字命令に 変換され点字プリンタで印字されます。

例 1.3.2 円を描く

円とその中心点を描いてみましょう。

circle 0 0 5 0 360 2.2

と入力して下さい。

circle は円を描くコマンド名です。円は、円の中心や半径等を指定するパラメータが 6 個 必要です。

最初の2つのパラメータは円の中心の座標(0,0)を指定しています。点字プリンタ用紙の中心の座標は(0,0)です。

次のパラメータ「5」は円の半径 5cm を指定します。長さはセンチメートルで指定して下さい。

次の 2 つのパラメータ「0」「360」は円を 0 度から 360 度まで、すなわち円の全周を描くことを指定します。円周の範囲は中心角で指定して下さい。上半分の半円の円周の指定は「0」「180」となります。

最後のパラメータ「2.2」は円周を描く点線の点と点の間隔を 2.2mm に設定します。点線の点と点の間隔はミリメートルで指定して下さい。

次に、

plot 0 0

と入力すると座標(0,0)に中心点が印字されます。

点字用紙をページ替えするためにはページ送りコマンド ff (formfeed)を入力して下さい。

console N:から test または debug コマンドを入力すると次に説明するテストモードまたはデバッグモードに移ります。

1.3.3 テストモード

テストモードは、コマンドの機能を画面で直接確認するために設けられたモードです。 作図コマンドを処理して印字命令を出力する代わりに、コマンドがどのような印字命令に 変換されたかを画面に表示します。

テストモードを抜けてメインメニューに戻るためにはコマンド bye を入力して下さい。 メインメニューから3またはt(test)を入力するとテストモードに入り test N:と表示され ます。1.1 節のコンソールモードから test と入力してテストモードに入った状態と同じです。 テストモードで表示される各点の座標はウィンドウ座標系と呼ばれる座標系です。点字 プリンタ用紙の真ん中を原点(0,0)とする実数の座標系です。

点字プリンタ ESA721 の A4 用紙の座標系はコンソールモードと同様、

window: -10.35 10.35 -12.35 12.53

です。原点は用紙の中央です。横座標は(-10.35, 10.35)、縦座標は(-12.53, 12.53)です。長さの単位はセンチメートルです。

座標(x0, y0)から座標(x1, y1)まで直線を作図するコマンドは

line x0 y0 x1 y1 pitch

です。

lin は直線の作図コマンド名です。直線コマンドは、5 つのパラメータが必要です。パラメータ x0、 y0、x1、 y1、pitch は数値で指定します。

最初の2つのパラメータは直線の始点の座標(x0, y0)を指定します。

次の2つのパラメータは直線の終点の座標(x1, y1)を指定します。

最後のパラメータ pitch は直線の点間隔をミリメートルで指定します。

テストモードで作図コマンドを実行しますと、点線の各点の作図コマンド plot x y の列として画面に表示されます。点の座標(x,y)はウィンドウ座標系の点の数値です。

テストモードにおける直線コマンド

line 0 0 2 1 2.2

の実行結果は次のようになります。

line 0.00 0.00 2.00 1.00 2.2 plot 0.000 0.000

```
plot 0.197 0.098
```

plot 0.394 0.197

plot 0.590 0.295

plot 0.787 0.394

plot 0.984 0.492

plot 1.181 0.590

plot 1.378 0.689

plot 1.574 0.787

plot 1.771 0.886

plot 1.968 0.984

始点から終点まで打てる限り点を打っていき、その結果が画面に plot コマンドで表示されます。

1.3.4 デバッグモード

メインメニューから 4 または d (debug)を入力するとデバッグモードに入ります。テストモードと同様、コンソールモードからも直接デバッグモードに入ることが出来ます。

デバッグモードとテストモードの違いは、テストモードが表示する座標はウィンドウ (window)座標系であるのに対してデバッグモードの座標はグリッド(grid)座標系です。

グリッド座標系は点字プリンタ用紙の左下を原点とする座標系です。点の印字位置が自然数で表示されます。

点字プリンタ ESA721 の A4 用紙のグリッド座標系は、

grid 0 599 0 790

です。横座標が 0~599、縦座標が 0~790 です。

1.3.5 リストサブメニュー

ファイルフォルダ内のファイル名を見るためのサブメニューです。ドライブレター、フォルダ名、ファイル名、拡張子を入力して下さい。

アステリスク及びクエスチョンマークのワイルドカードも使用可能です。

1.3.6 アウトプットサブメニュー

アウトプットサブメニューは、作図コマンドプログラムを実行して出力先を標準出力またはファイルを指定して触読図を印字するためのサブメニューです。操作は対話型になっていますので(1)~(6)の Bplot からの問に答えて下さい。入力ミス等、アウトプットサブメニューから直接メインメニューに戻りたい時はエスケープキーを押してからエンターキーを押して下さい。

(1) コマンドプログラムファイル名入力

最初の行に最も最近使用された作図コマンドプログラムのファイル名が大カッコ付きで表示されます。次の行に bplot command program name = と表示され入力待ち状態となります。

もし大カッコの中のファイル名を変更する必要がなければそのままエンターキーを押して下さい。

ドライブ名とフォルダ名は変えずにファイル名だけを変える場合は、ファイル名と拡張 子を入力して下さい。

拡張子だけを変えたい場合はピリオド付の拡張子だけを入力して下さい。

フォルダ名以下を変更したい時は、フォルダ名、ファイル名、拡張子の 3 つをすべて入力して下さい。

また、ドライブ名以下を変更したい時は、ドライブ名、フォルダ名、ファイル名、拡張 子の4つをすべて入力して下さい。

(2) 出力モード選択

出力モードメニュー(0: plotter mode 1: printer mode)が表示され、その下の行に、which mode [printer mode]?

とモード選択行が表示され入力待ち状態となります。

0: plotter mode の 0 を選択するとプロッタモード印字用ファイルを作成します。ESA721 点字プリンタの場合、プロッタのように印字ヘッドを上下左右に動かしながら印字していきます。次に説明するプリンタモードの約3倍の速さで印字が可能です。

1: printer mode の 1 を入力しますと印字する前に一旦印字データをメモリに記録してからその後、プリンタのように点字プリンタ用紙の一番上から 1 ラインずつ印字していくプリンタモードのファイルが作成されます。プロッタモードに比べ約 3 倍程度の印字時間がかかります。しかし、gap コマンド等を使用すれば表面または裏面の点の二重打ちを防止する機能が使用できます。(将来は表面と裏面間の二重打ちも防止する計画です。)

(3) 印字枚数入力

how many copy [1] = と表示され入力待ち状態となります。1 枚 \sim 100 枚までの印刷枚数を入力して下さい。

直接エンターキーを押すと大カッコの中の1すなわち、1枚の印字命令が出力されます。

(4) 出力先選択

(0: standard output 1: file)と表示され、出力先を標準アウトプットとするかファイルにするか選択出来ます。0 または 1 を入力して下さい。

(5) 出力先ファイル名入力

一つ前の(4)で(1: file)を選択しますと出力先ファイル名入力行が表示されます。 最後に使用したコマンドプログラムファイル名の拡張子を out に変えた出力ファイル名が 表示されます。次の行に output file name =と表示され、入力待ち状態となります。

出力ファイル名がそのままで良ければエンターキーを押して下さい。

変更する場合は入力して下さい。

(6) 作成確認

作成するかどうかの確認が求められますのでyまたはnを入力して下さい。直接エンターキーを押すと大カッコの中のyesとなり、出力されます。

1.3.7 Bplot と EDEL ファイルの相互変換サブメニュー

Bplot のコマンドプログラムと EDEL の作図ファイルの相互変換をするサブメニューです。Bplot と Edel ファイルの相互変換が可能となるため、EDEL から Bplot ファイルに変換すれば重度視覚障害者も変換された Bplot ファイルを CUI で修正し校正することが可能となります。一方、Bplot から EDEL ファイルに変換すれば EDEL の環境で晴眼者が変換された Bplot の画像を見ながら GUI で修正することが可能となります。

最初に EDEL ファイルから Bplot ファイルに変換するかまたは Bplot ファイルから EDEL ファイルに変換するかを選択して下さい。(0: EDEL to Bplot 1: Bplot to EDEL)と表示され、次の行に、

convert file [EDEL to Bplot]?と表示され入力待ちとなります。

そのままエンターキーを押すと大カッコの中の EDEL to Bplot 変換が選択されます。も し逆の Bplot to Edel 変換を希望する時は 1 を入力して下さい。

1.3.7.1 EDEL から Bplot への変換

(1) Edel ファイル名入力

EDEL ファイルから Bplot ファイルへの変換が選択されると、最も最近読み込まれたファイルの拡張子が EDEL の拡張子 edl に替えられたファイル名が大括弧付きで表示されます。次の行に、

read EDEL file name =と表示され入力待ちとなります。

エンターキーを押すと大カッコの中のファイルが読み込まれます。

もし変更する時はファイル名を入力して下さい。

(2) Bplot ファイル名入力

次に出力ファイル名入力行になります。今読み込んだ EDEL ファイル名の拡張子が Bplot の拡張子 bpl に替えられたファイル名が大カッコ付きで表示されます。 次の行に、

write Bplot file name =と表示され入力待ちとなります。

エンターキーを押すと大カッコの中の Bplot ファイルが出力先となります。

(3) ファイル変換確認

最後にファイル変換確認が求められます。

1.3.7.2 Bplot から EDEL ファイルへの変換

(1) Bplot ファイル名入力

最も最近読み込まれたファイル名の拡張子がBplotの拡張子bplに替えられたファイル名が大カッコ付きで表示されます。次の行に、

read Bplot file name = と Bplot ファイルの読み込み行が表示され入力待ちとなります。

(2) EDEL ファイル名入力

次に Edel ファイル名の入力行となります。

(3) ファイル変換確認

最後にファイル変換確認が求められます。

1.3.8 ポストスクリプトサブメニュー

ポストスクリプトサブメニューは Bplot 作図コマンドプログラムファイルを読み込み、ポストスクリプトファイルを作成します。ポストスクリプトファイルを作成すれば作図した画像をツールを使用してコンピュータ画面に表示したり印刷したりすることが可能となります。

操作は対話型です。必要事項を入力して下さい。入力ミス等、ポストスクリプトサブメニューから直接メインメニューに戻りたい時はエスケープキーを押してからエンターキーを押して下さい。

(1) Bplot コマンドプログラムファイル名の入力

最も最近使用されたプログラムファイル名が大括弧付きで表示され、次の行に、

bplot command program name = と表示され作図コマンドプログラム名の入力待ちとなります。

エンターキーを押すと大カッコの中のコマンドプログラム名が入力されます。

もしプログラム名を変更したい時はドライブ名、フォルダ名、プログラム名、拡張子の 4 つのうち、変える必要がある部分から後をすべて再入力して下さい。

もしドライブ名とフォルダ名を変えずにファイル名だけを変える場合は、コマンドプログラム名と拡張子を入力し直して下さい。

(2) ポストスクリプトファイル名入力

入力した作図コマンドプログラム名の拡張子を ps に変えたポストスクリプトファイル名 が大括弧付きで表示され、次の行に

postscript file name =と表示され、入力待ちとなります。

大カッコの中のポストスクリプトファイル名がそのままで良ければエンターキーを押して下さい。

もし変更する場合は再入力して下さい。

(3) 作成確認

作成するかどうかの確認が求められます。 y または n を入力して下さい。

直接エンターキーを押すと大カッコの中の yes となり、ポストスクリプトファイルが作成されます。

1.3.9 ヘルプサブメニュー

Bplot のメインメニューから h (help)を選択しますとオンラインヘルプサブメニューに入ります。画面に作図コマンド名が一覧表示され、最下行に help N:と表示されコマンド名入力待ちとなります。

ヘルプメニューのうちから目的の作図コマンド名の最初の 3 文字以上を入力しますと英文の簡単なコマンド説明が表示されます。作図コマンドの書式とコマンドの各パラメータの説明及びコマンドの機能が表示されます。

ヘルプモードを終了する時は、bye または quit を入力して下さい。 例えば plot と入力すると、

plot x y

x, y: coordinates of point plot the point

と表示されます。

1.5 エラーメッセージ

作図コマンドのエラーを発見し、修正しやすくするために Bplot は描くコマンドを実行する前に出来る限りコマンドを解析してエラーメッセージを表示するように設計されています。エラーメッセージを見てエラー箇所を修正して実行して下さい。

エラーメッセージは 4 つの部分で構成されています。エラー番号、プログラムファイル 名とエラーが発見された行番号、コマンド名、及びエラー内容です。

例 1.5 エラーメッセージ

例えば直線を描くプログラムファイルの作図コマンド名やパラメータをわざと書き直し、 エラーを発生させてみましょう。

直線を描くプログラムは、

dot 1

line 0 0 2 3 2.2

ff

プログラム中の作図コマンド dot を doq に、line コマンドの 4 番目のパラメータを 2 から b に変えて実行してみます。

 $\begin{array}{l} \operatorname{doq} 1 \\ \operatorname{line} 0 \ 0 \ 3 \ \mathrm{b} \ 2.2 \\ \mathrm{ff} \end{array}$

実行時コンピュータ画面に次のエラーメッセージが表示されます。

error 1? line.bpl line 1 "doq": unknown command error 2? line.bpl line 3 "line": undefined variable b error 3? line.bpl line 3 "line": eval 3 errors 0 warning

1番目のエラーは、プログラムファイル line.bpl の 1 行目の doq というコマンドは、見知らぬコマンド名ですというメッセージです。

2番目のエラーは、line.bpl の3行目のlineというコマンドに、定義のされていない変数b があるというメッセージです。

3番目のエラーは、line.bpl の3行目の line というコマンドに、計算できないパラメータ があるというメッセージです。

最後のメッセージは、エラーが3箇所、注意箇所が0であることを表示しています。

第2章 作図コマンドプログラムの作成法 (基礎編)

2.1 作図コマンドの概要

第2章はBplot (CUI バージョン)で作図する方法に関する解説の初級編です。

実際に Bplot で触読図を作成するためにはまず最初にコンピュータのエディタを使って作図コマンドプログラムファイルを作成することが必要です。Bplot がこの作図コマンドプログラムファイルを読み込みますと、作図コマンドが点字プリンタの印字命令に変換され、実際に触読図が印字されます。

作図コマンドは行頭に作図コマンド名を書き、その後にそのコマンドが必要とする数値 等パラメータを記述します。コマンド名と各パラメータは空白(マスあけ)で区切ります。

Bplot はコマンド名の最初の3文字だけでコマンドを識別しています。4文字以降は省略しても構いません。しかし、省略すると別の方があなたが作成した作図コマンドプログラムを読んで利用するときに把握しにくくなる場合もありますので省略しないことをお勧めします。

ただし例外はプリンタ指定コマンド名の ESA721 と ESA600G 及び印字方式指定コマンド名の plotter は最初の 3 文字では識別できないため 5 文字を入力して下さい。ESA721 と ESA600G は最初の 3 文字が同一でありまた、plotter は打点コマンド plot と識別できないためです。

基本的に作図コマンドは作図コマンドプログラムファイルに 1 行 1 コマンドを記述してください。作図プログラムを読みやすくするためにコマンド名を適宜字下げ(インデント)しても構いません。この解説書の作図例ではプログラムの流れがわかりやすいようにプログラムは字下げして表記しています。

作図コマンドは点字プリンタ用紙上に設定されたウィンドウに対して記述します。点字 用紙上の位置を横座標xと縦座標yの2つの数値で表すために座標系が定義されています。 この座標系が定義された点字用紙上の長方形の平面をウィンドウと呼びます。ウィンドウ の範囲はプリンタが印字可能な点字用紙上の長方形の範囲と対応しています。

通常(ウィンドウを変化させない限り)プリンタ用紙の中央が原点(0,0)です。横座標は右方向(X-軸方向)が正の向きです。縦座標は上方向(Y-軸方向)が正の向きです。

点字プリンタ ESA721 及び ESA600G (JTR (株))は B5 サイズ(横 8 インチないし 8.5 インチ、縦 10 インチ)または A4 サイズ(横 10 インチ縦 11 インチ)の点字用紙を使用します。なお、B5 または A4 は通称であり、世間で言う B5 または A4 サイズとは違います。

B5 サイズの点字プリンタ用紙上に定義されているウィンドウの座標系は、横座標の範囲が左から右に-8.28cm~8.28cm、縦座標の範囲が下から上に-11.48cm~11.48cm です。一方、A4 サイズの用紙のウィンドウの座標系は、横座標の範囲が-10.35cm~10.35cm 縦座標が-12.53cm~12.53cm です。

Bplot を初めて使い始める時またはプリンタ用紙の大きさを変える時に、プリンタ用紙設定コマンド b5 または a4 を作図コマンドプログラムファイルの最初の所で記述しておいて

ください。1度実行されればBplotは用紙サイズ設定を記憶しています。

2.2 点の種類と点を打つコマンド

点字プリンタ ESA721 及び ESA600 には印字する点が 3 種類用意されています。この点を替えるコマンド名は dot です。dot は点の種類を指定するパラメータが一つ必要です。0 が小点、1 が中点、2 が大点の3 種類です。

点を打つコマンド名は plot です。 plot は打つ点の座標をパラメータとして必要とします。 座標はウィンドウ上の座標です。

例 2.2.1 点種を変えて打点

点字プリンタ用紙に3種類の点を使って点を打ってみましょう。

原点 (0,0) に大点、原点から右に 1 cm 上に 1 cm いったところに中点、原点から右に 2 cm 上に 2 cm いったところに小点を打ってみます。

プログラム例の最初の3~5行はプリンタ用紙の右上隅にページ見出しを印字するためのコマンドです。また、プログラムファイルの最後のff コマンドはページを送るコマンドです。ページ見出しを印字するfmt と brl コマンド及びページ送るコマンドff は後述します。

// ex20201.bpl

fmt 40 24

brl 40-10 24 #b4#b4#a

brl 7 23 little, midle and large points

a4

dot 2

plot 0 0

dot 1

plot 1 1

dot 0

plot 2 2

ff

ex20201.bpl の印字結果

ウィンドウ上で打つ点の座標を替えれば点字用紙上にどのような点も打つことが可能です。

また、点の種類を変えれば表現力のある図の作成が可能となります。

2.3 定形的な直線系図形を描く

例 2.3.1 直線を作図

例えば原点を始点として右に 3cm 上に 2cm 言ったところの点を終点として直線(線分)を描いてみましょう。直線のコマンド名は line です。line は始点と終点の座標及び描く点線の点と点の間の間隔(ドットピッチ)の 5 つのパラメータ(引数)を指定します。プログラム中//はその後の文字列が注釈(コメント)であることを示す命令です。注釈はコマンドプログラムにメモを書き込むためのもので Bplot は読み飛ばします。

ff (formfeed)はページを送るコマンドです。

// ex20301.bpl

fmt 40 24

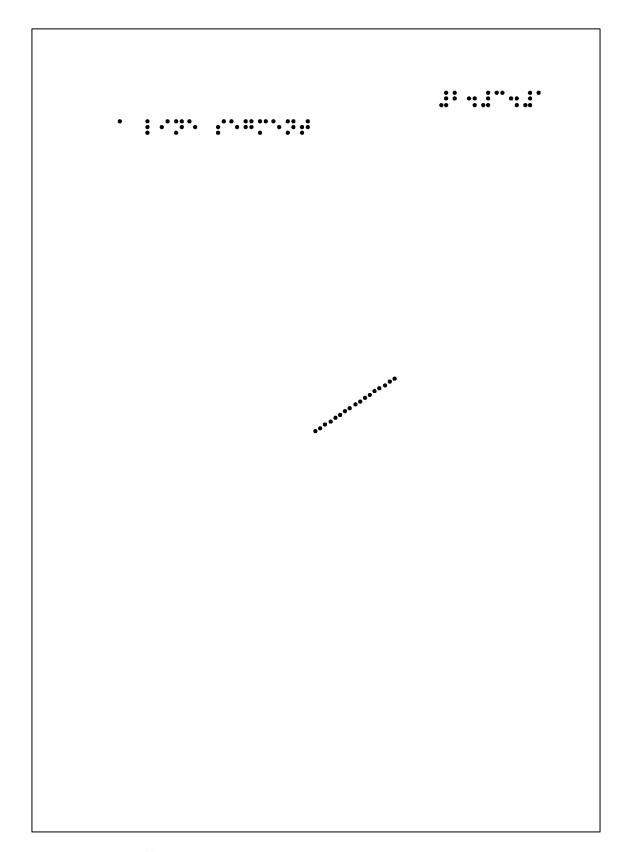
brl 40-10 24 #b4#c4#a brl 7 23 a line segment

a4

dot 1

line 0 0 3 2 2.2

ff



ex20301.bpl の印字結果

直線の始点と終点の座標を座標系の範囲内で自由に変えれば目的の直線を描くことが出来ます。また、点線の点と点の間隔(ドットピッチ)を変えれば任意の点線や実線を引くことが可能です。

Bplot のすべての作図コマンドの点間隔の長さの単位は常にミリメートル(mm)です。ウィンドウの座標系の長さの単位がどのように変わろうとも常にミリメートルです。図を拡大縮小しようとも点間隔が変わらないため点線や実線の手触りは変わらないように設計されています。

例 2.3.2 点種を変えて長方形を描く

// ex20302.bpl

fmt 40 24

brl 40-10 24 #b4#c4#b

a4

dot 0

line -3 -2 3 -2 1.8

dot 1

line 3 -2 3 2 2.2

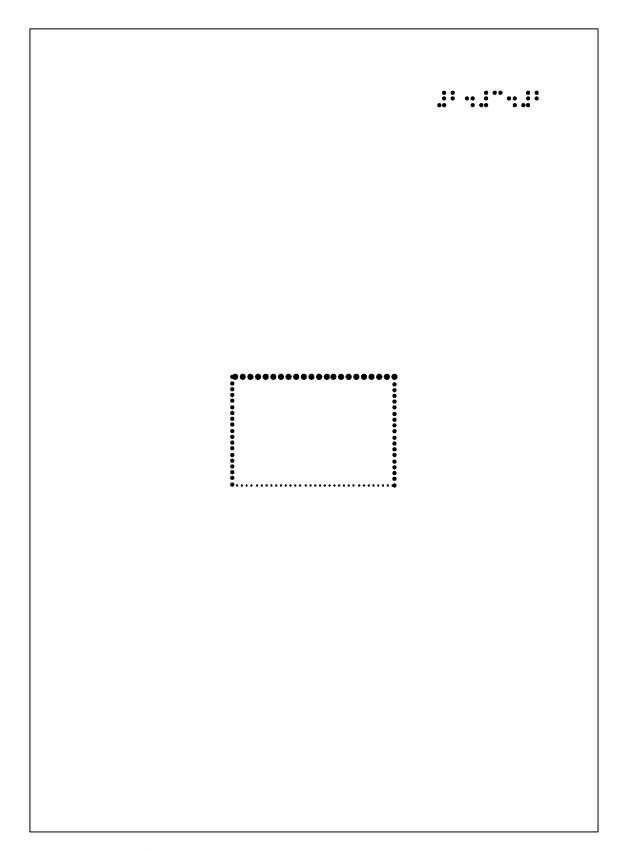
dot 2

line 3 2 -3 2 2.8

dot 1

line -3 2 -3 -2 2.2

ff



ex20302.bpl の印字結果

2.4 不定形な直線系図形を描く

直線系図形の中には、正方形や長方形、ひし形や平行四辺形等、定形的な直線系図形と 折れ線のような不定形な図形があります。折れ線作図コマンド cline は定形・不定形を問わず自由に直線系図形を描くことが出来るコマンドです。

例 2.4.1 折れ線を描く

何本かの直線を繋いで折れ線図形を描くコマンド cline (concatinated lines)が用意されています。例 2.3.2 の長方形の例では 4 本の直線を繋いで描いていました。cline を使用すればより容易に描くことが可能となります。

cline は複数行にわたるコマンドです。

1行目にコマンド名 cline と点間隔を書きます。

2 行目から始点とそれに続く折れ線の折れ目の点と終点の座標を行を変えて書きます。

最後に座標データの終了を示す「9999 0」を書いてください。Bplot は、この「9999 0」を読み込みますと座標データが終了したことを認識します。折れ線の描く点は始点から何点あっても構いません。「9999 0」を読み込むまで座標データとして読み込みます。

なお、次のプログラムの cline の直後の 2.2 は点間隔を 2.2mm に設定するパラメータです。

// ex20401.bpl

fmt 40 24

brl 40-10 24 #b4#d4#a

brl 7 23 ,fig4 box by cline command

a4

dot 1

cline 2.2

-3 -2

3 -2

32

-32

-3 -2

99990

ff

ex20401.bpl の印字結果

cline を使用すればどのような直線系の複雑な図形も容易に描くことが可能です。最後の座標データの行が終点となり、line で記述する場合のようにいちいち終点の座標データを書く必要がなくなります。

Bplot には直線(線分)で構成される直線系の図形を描くために多数のコマンドが用意されています。定形的な直線系の図形を描くために、直線(line)、箱(box)、長方形(rectangle)、平行四辺形(paralleogram)、ひし形(diamond)、正多角形(polygon)等があります。

その他矢印を描くコマンドも arrow と dart の 2 種類が用意されています。

また、直線系の図形を自由に描くコマンド cline が用意されています。

2.5 定形的な曲線系図形を描く

例 2.5.1 円と中心点を描く

定形的な曲線系図形を描くために多数のコマンドが用意されています。円(circle)、楕円 (ellipse)、放物線(vparabola、hparabola)、双曲線(hyperbola、rhyperbola)等が用意されています。

例えば円を描いてみましょう。円のコマンドは circle です。 circle は中心点の座標、半径、描く円周の円周角の範囲及び点線の点と点と間隔(ドットピッチ)の6つのパラメータを指定します。

例えば中心点が原点 (0,0)、半径が 5cm 中心角が 0 度~360 度、点間隔が 2.2mm の円を描いてみましょう。

点を打つコマンド plot を使用して中心点を打っておきます。

// ex20501.bpl

fmt 40 24

brl 40-10 24 #b4#e4#a

a4

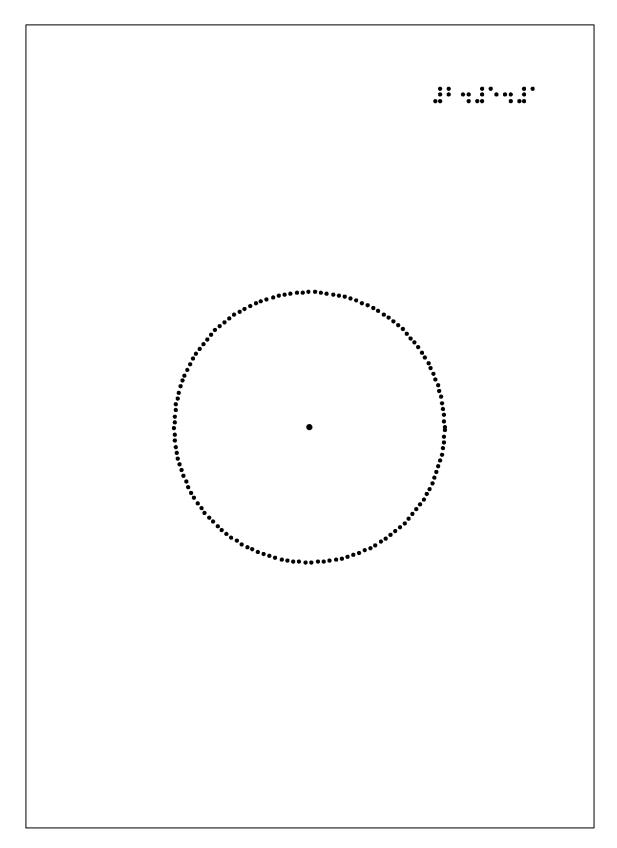
dot 1

circle 0 0 5 0 360 2.2

dot 2

plot 0 0

ff



ex20501.bpl の印字結果

円周角の範囲を変えれば半円や4分円等任意の大きさの円弧を描くことが可能です。

2.6 不定形な曲線系図形を描く

例 2.6.1 自由曲線を描く

定形的な曲線系図形だけでなく、曲線系図形を自由に描くコマンドも用意されています。 spline は曲線を自由に描くコマンドです。指定された点を通るなめらかな曲線を描くコマンドです。

spline は折れ線コマンド cline と同様、複数行にわたるコマンドです。

spline は8個のパラメータを必要とします。

最初の 2 つのパラメータで曲線を描くウィンドウ上の横方向の範囲を指定します。引き続く 2 つのパラメータで描く曲線の縦方向の範囲を指定します。一般的にはこの範囲はウィンドウ全体に取っておき、もし描く範囲を制限したい場合に範囲を狭めてください。

5 番目のパラメータは自由曲線の種類を指定します。始点と終点が繋がっていない曲線 (開曲線)を描く場合は $1\sim3$ の数値を指定してください。始点と終点がなめらかに繋がっている曲線(閉曲線)を描く場合は 4 を指定してください。

6番目と7番目のパラメータは描く曲線を横方向または縦方向に平行移動させるためのパラメータです。平行移動をさせないのであれば「0」「0」を指定してください。

最後の8番目のパラメータは点線の点間隔です。

spline コマンドの次の行から曲線が通る各点の座標を記述してください。

終点の次の行に座標データの最後を示すパラメータ「9999 0」を指定してください。

5番目のパラメータを4にして閉曲線の星形を描いてみます。4の閉曲線の場合始点(5,0)が終点ともなります。

// ex20601.bpl

```
fmt 40 24
```

brl 40-10 24 #b4#f4#a

brl 7 23 ,fig4 star by spline command

a4

dot 1

spline -10 10 -12 12 4 0 0 2.2

50

22

0.5

-22

-50

-2 -2

0 -5

2 -2

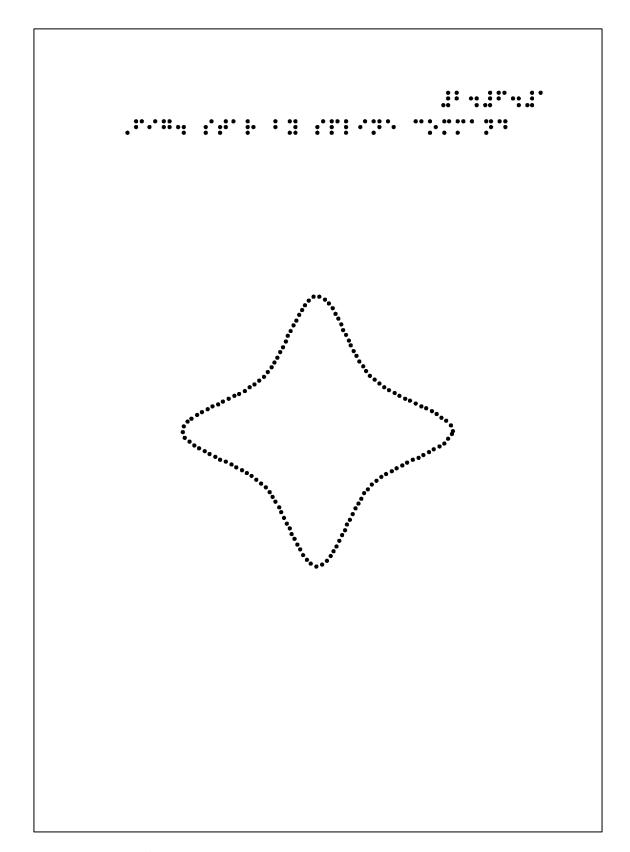
99990

ff

spline 曲線が通る点は任意でよいのでどんな曲線もこのコマンドで描くことが可能です。 ただし、隣り合う点の \mathbf{x} 座標が等しいとデータエラーとなります。その場合はどちらかの \mathbf{x} 座標を少しずらしてください。

もし曲線がうまく描けない場合は spline 曲線を指定する 5 番目のパラメータを変えて試してみてください。

地図の国境線や海岸線、グラフの曲線、漢字の線等、spline 曲線により曲線系の図形を 自由に描くことが可能です。



ex20601.bpl の印字結果

2.7 図に点字を書き込む

図には標題を付けたり、図形の頂点にアルファベットの名称を付けたり、更に化学構造式の原子の名称を書き込んだり点字を書き込む必要があります。この場合には点字を書き込む2つのコマンド braille と kana が用意されています。

braille は brl と略して使用可能です。brl は最初の 2 つのパラメータで書き入れる点字の 1 マス目の 1 の点の位置の座標を指定します。その後に 1 マス空けて書き込む点字を書きます。点字は NABCC (North American Braille Computer Code)で記述してください。

kana は書き込む点字の1マス目の位置の座標とその後に半角カタカナ及び英数字で記述してください。Bplot がそれを点字に変換して印字します。

例 2.7.1 ひし形の頂点に名称を書く

ひし形の作図コマンドは diamond です。最初の 2 つのパラメータはひし形の中心(重心) の位置の座標を書きます。

- 3番目のパラメータは中心から左右の頂点までの距離です。
- 4番目のパラメータは中心から上下の頂点までの距離です。
- 5番目のパラメータは点間隔です。

// ex20701.bpl

fmt 40 24

brl 40-10 24 #b4#g4#a brl 7 23 ,fig4 diamond

a4

dot 1

dia 0 0 3 2 2.2

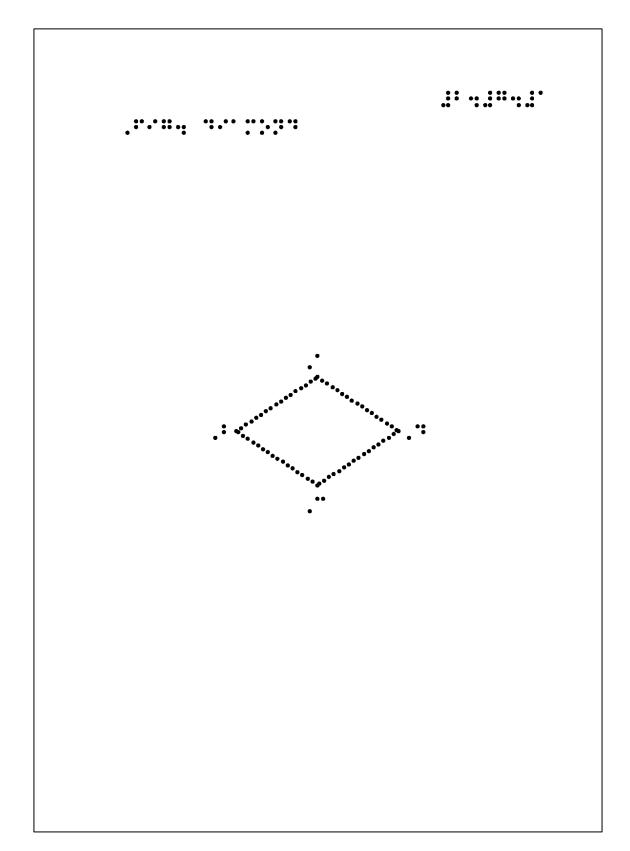
brl -0.5 2.8, a

brl -4 0.2 ,b

brl -0.5 -2.5,c

brl 3.2 0.2,d

ff



ex20701.bpl の印字結果

第3章 作図コマンド解説 (上級編)

本章は、作図コマンド別に作図コマンドプログラムの作成方法と作図例を解説します。 作図例のコマンドプログラムは examples フォルダに入っています。Bplot の作図プログ ラムを作成する際に必要箇所をコピーしてお使い下さい。印字可能であることが確認され ています。

なお、地図や模式図、流れ図や各種のグラフ等、点字教材の作成方法は教材の種類別に 第4章で解説します。

Bplot には英文のオンラインヘルプが用意されています。Bplot のメインメニューからヘルプサブメニューに入り、コマンドメニューをご覧いただきその中から関連のあるコマンド名を選びコマンド名の頭の 3 文字ないし 5 文字を入力するとそのコマンドのヘルプメッセージが表示されます。

ヘルプメッセージは、1行目にコマンドの書式が表示されます。その下の行に3マスずつ字下げして解説が表示されています。簡単なコマンドのパラメータの説明とコマンドの機能が表示されています。

エディタを使用して CUI で作図コマンドプログラムファイルを作成してから Bplot を使用して印字して下さい。印字にはアウトプットサブメニューに入って下さい。

ヘルプのコマンドメニューは次のとおりです。

menu (Initial 5 or 3 chars are effective for each command)

bye help menu test debug bplot step nostep wait

log error output postscript

view b5 a4 origin portrait landscape mirror window frame

variable define undefine

esa721 esa600g plotter printer

// /* */

alarm version radian degree calc system

dot plot point design

fmt braille kana ff formfeed move document

line hatch arrow dart

box diamond polygon parallelogram rectangle

circle ellipse oval

cline spline xaxis yaxis

vparabola hparabola hyperbola rhyperbola

f1 f2 f3 f4 f5 sin cos tan

normal bwplot weibull pow exp l10 ln

rotate paste rpaste include

gap scrape erase

explain a command (return by 'bye').

3.1 ウィンドウの設定

第2章の2.1節で簡単に紹介したウィンドウの概念を詳細に説明すると共にウィンドウの 設定コマンドについて解説します。

点字教材の触読図を作成するためには、まず最初に、頭の中の触読図のイメージを Bplot の作図コマンドを使用してウィンドウ(window)と呼ばれる長方形の平面上に描いて下さい。この描かれたウィンドウ上の図は Bplot により自動的に点字プリンタ用紙上の長方形の平面上に変換され実際に印字されます。

ウィンドウは座標系が定義された長方形の仮想平面です。このウィンドウ座標系を使用 して作図コマンドを記述して下さい。

一方、点字プリンタが実際に印字する点字用紙上の長方形の平面にも座標系が定義されています。この長方形の実平面の座標系はグリッド(grid)座標系と呼びます。Bplot はウィンドウ座標系の仮想平面の作図コマンドをグリッド座標系の点字用紙上の実平面に変換して印字しています。

例えば、高解像度点字グラフィックプリンタ ESA721 で B5 サイズまたは A4 サイズの点字用紙に印字する場合、Bplot の初期設定時(デフォルト)のウィンドウ座標系とグリッド座標系の範囲は次のとおりです。

なお、フレーム(frame)座標系はグリッド座標系の印字可能な範囲を制限した場合の座標 範囲です。次の例では印字可能範囲が制限されていないためフレーム座標系はグリッド座 標系と同じになります。

b5:

grid(0 479 0 724)

frame(0 479 0 724)

window(-8.28 8.28 -11.48 11.48)

As b5 and portrait, printer (ESA721)

V/H ratio: 1.0007

a4:

grid(0 599 0 790)

frame(0.599.0.790)

window(-10.35 10.35 -12.53 12.53)

As a4 and portrait, printer (ESA721)

V/H ratio: 1.0000

グリッド・フレーム・ウィンドウのカッコの中の最初の 2 つの数値が横座標(x) 座標(x) の範囲です。次の 2 つの数値が縦座標(y) 座標(x) の範囲です。

点字プリンタ ESA721 は B5 サイズの点字用紙に横 480 点、縦 725 点打つことが可能です。A4 サイズの点字用紙に横 600 点、縦 791 点打つことが可能です。

グリッド座標系はこの実際に点を打つことが可能な点字用紙上の位置を示しています。2 つの自然数(0、1、2、3、……)でx 座標とy 座標が表されています。原点(0,0)は左下隅です。

一方ウィンドウは、平面上の位置が \mathbf{x} 座標も \mathbf{y} 座標も実数で定まる座標系です。初期設定時のウィンドウ座標系は点字用紙の中央を原点(0.00,0.00)とする座標系です。 \mathbf{x} 座標も \mathbf{y} 座標も左下隅が最も小さい値となります。しかし、実際に点を打つことが可能な位置はグリッド座標系の 2 つの自然数で定まる位置です。

ウィンドウの状態を見るためには Bplot のメインメニューから set サブメニューに入り v (view)を入力すると表示されます。

また、テストモードで view と入力すると画面に表示されます。

3.2 ウィンドウの設定

ウィンドウ座標系は自由に変えることが可能です。ウィンドウ座標系設定コマンドには 次の9つがあります。

view ウィンドウの状態を表示するコマンド

b5 B5 サイズの点字用紙に対応したウィンドウ設定コマンド

a4 A4 サイズの点字用紙に対応したウィンドウ設定コマンド

origin 原点の移動と図の拡大縮小コマンド

pootrait 点字用紙の縦向きに対応したウィンドウ座標系設定コマンド

landscape 点字用紙の横向きに対応したウィンドウ座標系設定コマンド

mirror 裏面点字印字設定コマンド

window ウィンドウ座標系設定コマンド

frame フレーム座標系設定コマンド

3.2.1 ウィンドウの表示

<書式>

view (with no parameter)

<解説>

view コマンドはウィンドウの現在の設定状態を表示させるためのコマンドです。ウィンドウは変化しません。

表示される内容は Bplot の設定サブメニューの v (view)と同じです。

3.2.2 b5 ウィンドウ設定コマンド

<書式>

b5 (with no parameter)

<解説>

B5 サイズの点字用紙に対応したウィンドウ座標系を設定します。原点は点字用紙の中央となります。また、グリッド及びフレーム座標系も設定しなおされます。

例 3.2.2 b5

b5 コマンドが実行された後の Bplot の設定状態です。

b5:

grid(0 479 0 724)

frame(0 479 0 724)

window(-8.28 8.28 -11.48 11.48)

As b5 and portrait, printer (ESA721)

V/H ratio: 1.0007

3.2.3 a4 ウィンドウ設定コマンド

<書式>

a4 (with no parameter)

<解説>

A4 サイズの点字用紙に対応したウィンドウ座標系を設定します。原点は点字用紙の中央となります。また、グリッド及びフレーム座標系も設定し直されます。

例 3.2.3 a4

a4 コマンドが実行された後の Bplot の設定状態です。

a4:

grid(0 599 0 790)

frame(0 599 0 790)

window(-10.35 10.35 -12.53 12.53)

As a4 and portrait, printer (ESA721)

V/H ratio: 1.0000

3.2.4 origin 原点の設定

<書式>

origin x0 y0 biasx biasy

<解説>

図を平行移動させたりまたは拡大縮小させたりするコマンドです。origin には 4 つのパラメータが必要です。

現在のウィンドウ座標系の(x0, y0)に新しい座標系の原点(0, 0)を移動させます。 また、x 座標を biasx 倍、y 座標を biasy 倍拡大また縮小させたりします。biasx と biasy が共に 1 より小さいとき図形は縮小しますがウィンドウは逆に拡大したことになります。

例 3.2.4 origin コマンド

原点を中心とする円を描いた後、origin コマンドを使用して、上に 7cm 移動させたところに半径を 1/2 に縮小した円を描いてみます。更に、下に 7cm 移動させたところに半径を 1.5 倍に拡大した円を描いて見ます。

// ex30204.bpl

fmt 40 24

brl 40-10 24 #c4#b4#d brl 7 23 ,fig4 example of origin

a4

dot 1

circle $0\ 0\ 2\ 0\ 360\ 2.2$

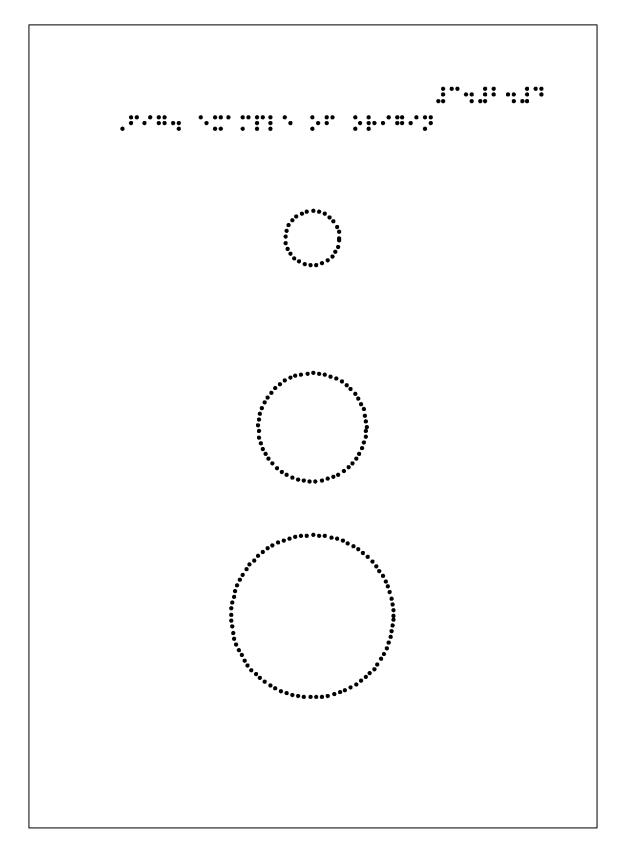
origin 0 7 0.5 0.5 circle 0 0 2 0 360 2.2

a4

origin 0 -7 1.5 1.5

circle 0 0 2 0 360 2.2

ff



ex30204.bpl の印字結果

プログラムには最初の 5 行にプログラムの名称を表示したりページ番号を印字したりするためのコマンドを加筆しています。1 行目の//は注釈行(コメント行)です。プログラム名を注釈として表示しています。印字には関係がありません。

3~5 行目はプログラムのページ番号やタイトルを印字するためのコマンド行です。詳細な解説は後述します。

9 行目、12 行目、16 行目の circle は原点 (0,0) を中心として半径 2 の円を 0 度から 360 度まで点線の点間隔 2.2mm で描く円のコマンドです。

11 行目の origin コマンドの最初の 2 つのパラメータ「0」「7」は現在の座標 (0,7) に新しい原点 (0,0) を移動させます。その結果、原点が 7cm 上に上がりますので円も 7cm 上がって描かれます。また、次の 2 つのパラメータ「0.5」「0.5」は図の x 方向の長さと y 方向の長さをそれぞれ 1/2 に縮小します。その結果、上に 7cm 上がったところにもとの円の 1/2 の半径の小さな円が描かれます。

同様に、15 行目の origin コマンドにより 7cm 下がったところに半径が 1.5 倍になった円が描かれます。

なお、14 行目の a4 コマンドはウィンドウの座標系を初期設定の状態に戻しています。座標系を移動させた上に更に移動させると移動後の座標系がわかりにくいためです。混乱しない自信があれば a4 コマンドを挿入する必要はありません。

3.2.5 縦向きウィンドウ設定コマンド

<書式>

portrait (with no parameter)

<解説>

ウィンドウ及びグリッドとフレームの座標系を点字用紙の縦向きに対応した座標系に設定します。Bplot の初期設定時は portrait です。

横向き landscape から縦向きに変更する場合に portrait コマンドを使用してください。

3.2.6 横向きウィンドウ設定コマンド

<書式>

landscape (with no parameter)

<解説>

ウィンドウ及びグリッドとフレーム座標系を点字用紙の横向きに対応した座標系に変更します。横向き landscape の場合点字用紙のミシン目のある側が下になるように設定されます。その結果、点字用紙のミシン目のない側が上となりバインダーで上を綴じることが出来ます。

横向き landscape にする場合はプログラムファイルの最初の所で宣言して下さい。

例 3.2.6 landscape

landscape コマンドにより縦向き向き字用紙に印字する状態から横置きに変更した場合 の Bplot の設定状態を次に示します。view コマンドで表示したものです。**例 3.2.2 b5** と **例 3.2.3 a4** と比較してみて下さい。

b5:

grid(0 724 0 479)

frame(0 724 0 479)

window(-11.48 11.48 -8.28 8.28)

As b5 and landscape, printer (ESA721)

V/H ratio: 0.9994

a4:

grid(0 790 0 599)

frame(0 790 0 599)

window(-12.53 12.53 -10.35 10.35)

As a4 and landscape, printer (ESA721)

V/H ratio: 1.0001

3.2.7 裏面点字設定コマンド

<書式>

mirror flag dx

<解説>

裏面点字印字設定コマンド mirror コマンドにより点字用紙の裏面をあたかも表面に対応したウィンドウに対するように作図することが可能となります。作図コマンドの始点や終点等の座標のx座標が左右裏返されて実現しています。表面のウィンドウに対する作図コマンドを記述すれば点字用紙を裏返してセットしても適切な裏面点字が印字されます。

mirror コマンドには2つのパラメータが必要です。

最初のパラメータ「flag」は裏面点字の印字の開始と終了を宣言するフラグです。フラグが1の場合、Bplotは点字用紙を裏返して点字プリンタにセットするように警告します。点字用紙を裏返してセットしてからエンターキーを押して下さい。裏面に対応した点字が印字されます。

フラグが 0 の場合、裏面点字の印字が終了となります。Bplot は点字用紙を表面に戻して セットするように警告します。セットし直してエンターキーを押すと表面印字に戻ります。 2番目のパラメータ「dx」は表面と裏面のずれを調整するものです。ミリメートルで設定 して下さい。点字プリンタの紙おさえの位置によってずれは異なります。印字する前に 1 度表面と裏面の横方向のずれをミリメートル単位で測定しておいて下さい。例えば mirror コマンドを使用して表面と裏面に原点を印字した後、x 方向のずれを計測してみて下さい。 紙おさえの位置を変えない限りずれは変わりません。

3.2.8 window 設定コマンド

<書式>

window x0 x1 y0 y1

<解説>

window コマンドはウィンドウを任意に設定することが可能です。ウィンドウを変えるとウィンドウの縦横比が変わりますので注意する必要があります。 ウィンドウの縦横比を変えたくない場合は origin コマンド(3.2.1 節)を使う方が簡単です。

window コマンドには 4 つのパラメータが必要です。「x0」でウィンドウの x 座標の左端を、「x1」で x 座標の右端を指定します。同様に、「y0」でウィンドウの y 座標の一番下の端を、「y1」で y 座標の一番上の端を指定します。

図を描画するウィンドウの座標系は仮想空間ですので長方形であれば大きさは任意です。 世界地図のように緯度と経度で表現される大きな座標系であっても、メモリの単位がミリ メートルでもセンチメートルでもキロメートルでも自由です。

window コマンドにより原点を移動させる、すなわち、図を平行移動させたり、図を拡大縮小させたり自由に変形することが出来ます。

3.2.9 frame コマンド

<書式>

frame fx0 fx1 fy0 fy1 wx0 wx1 wy0 wy1

<解説>

frame コマンドは点字プリンタが印字可能なグリッドの範囲を制限するために使用します。しかし、使う機会はあまりないと思います。

パラメータは8つ必要です。そのうち、最初の4つでグリッドの範囲を制限します。「fx0」と「fx1」によりグリッドのx座標の範囲を制限します。同様に、「fy0」と「fy1」によりグリッドのx座標の範囲を制限します。

後の 4 つのパラメータ「wx0」「wx1」「wy0」「wy1」でウィンドウ座標系を設定して下さい。

例 3.2.9 frame コマンドの例

例えば点字用紙の右上 1/4 だけを印字可能にしたい場合は次のようになります。

// ex30209.bpl

fmt 40 24

brl 40-10 24 #c4#b4#i brl 7 23 ,fig4 first quadrant of window

a4

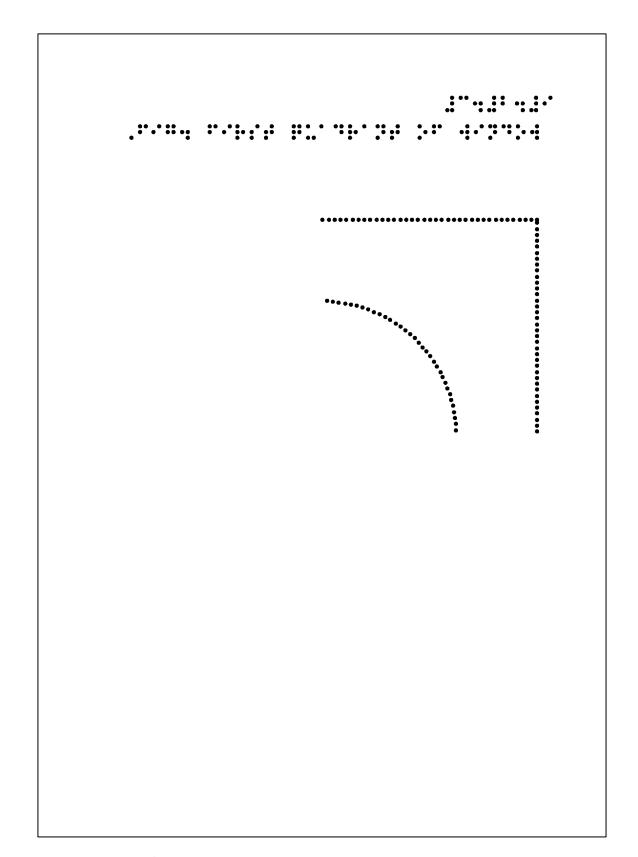
frame 300 599 400 790 -10.35 10.35 -12.53 12.53

dot 1

circle 0 0 6 0 360 2.2 box 0 0 8 8 2.2

ff

frame により印字範囲が第 1 象限に制限されています。原点を中心に半径 6 で描いた円と中心から左右上下の辺までの距離が 8 の箱形の第 1 象限の部分だけが印字されています。なお、circle と box コマンドは後述します。



ex30209.bpl の印字結果

3.3 変数と数式定義コマンド

作図コマンドのパラメータを変数やその数式で記述することが可能です。ウィンドウに作図コマンドを記述するためには始点や終点等座標をウィンドウ座標系で記述する必要があります。origin コマンド等ウィンドウ設定コマンド(3.2節)でウィンドウを操作しますと作図コマンドのパラメータの座標等もそれに合わせて計算し直す必要が出てきます。パラメータが変数や関数を含む数式で記述できれば、座標等、計算の煩わしさが解消され、計算がすべて自動化されます。

変数には通常の変数とマクロ変数が用意されています。

3.3.1 変数定義コマンド

<書式>

variable = expression

<解説>

変数定義コマンドは CUI で作図コマンドを記述する際に非常に有効です。変数を使用して作図コマンドを記述すれば変数の値を変えることにより図形を自由に配置しなおしたり、点線の点間隔等同じ値を取るパラメータを一度に変更し直したりすることが可能となります。

変数定義コマンドは、変数名の後に等号記号を書きその後に数値または数式を書いて下さい。

数式には数値だけでなく、定義された変数名や関数が使用可能です。関数電卓のような機能が実現されています。

変数名は英字で始まる英数字の文字列です。アンダーラインも含めることが出来ます。 しかし、Bplotで定義されているコマンド名は変数名には使用できません。

数式には四則演算記号、べき乗記号、カッコ等が使用できます。

また、次の関数も使用可能です。変数をxとすると、

pqrt(x) ルート

exp(x) 自然対数の底eのべき乗

log(x) 自然対数

log10(x) 常用対数

sin(x) サイン

cos(x) コサイン

tan(x) タンジェント

dx(x) 点字用紙の x 軸方向の距離 (ミリメートル)

dy(x) 点字用紙の y 軸方向の距離 (ミリメートル)

例 3.3.1 変数と関数の使用

例えば変数と三角関数を含む数式を使用すれば星形を簡単に描くことが可能となります。

```
ex30301.bpl
fmt 40 24
   brl 40-10 24 #c4#c4#a
   brl 7 23 ,fig4 a star by cline
a4
1 = 5
m = 2
r = 360/10
pitch = 2.2
x0 = 1*\cos(90+0*r)
y0 = 1*\sin(90+0*r)
x1 = m*\cos(90+1*r)
y1 = m*\sin(90+1*r)
x2 = 1*\cos(90+2*r)
y2 = 1*\sin(90+2*r)
x3 = m*\cos(90+3*r)
y3 = m*\sin(90+3*r)
x4 = 1*\cos(90+4*r)
y4 = 1*\sin(90+4*r)
x5 = m*\cos(90+5*r)
y5 = m*\sin(90+5*r)
x6 = 1*\cos(90+6*r)
y6 = 1*\sin(90+6*r)
x7 = m*\cos(90+7*r)
y7 = m*\sin(90+7*r)
x8 = 1*\cos(90+8*r)
v8 = 1*\sin(90+8*r)
x9 = m*\cos(90+9*r)
y9 = m*\sin(90+9*r)
x10 = l*cos(90+10*r)
```

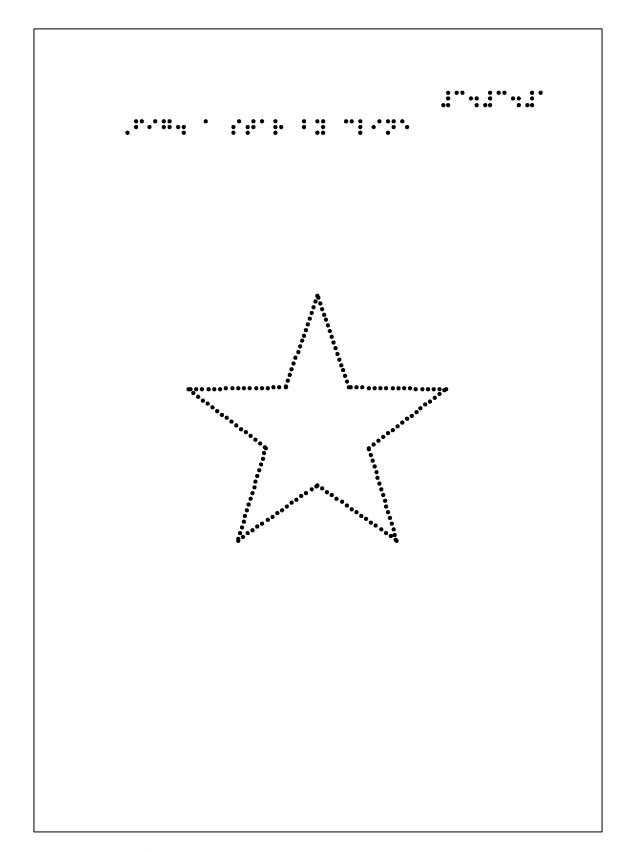
y10 = l*sin(90+10*r)

dot 1

cline pitch

- x0 y0
- x1 y1
- x2 y2
- x3 y3
- x4 y4
- x5 y5
- x6 y6
- x7 y7
- x8 y8
- x9 y9
- x10 y10
- 9999 0

ff



ex30301.bpl の印字結果

3.3.2 マクロ変数定義コマンド

<書式>

define variable string undefine variable

<解説>

マクロ変数も定義可能です。

define コマンドは作図プログラム中のマクロ変数名 variable を文字列 string で置き換えます。作図コマンドの一つまたは複数のパラメータを一つのマクロ変数名で置き換える際に使用します。文字列にはスペースを含むことも出来ます。

なお、マクロ変数はパラメータとして使用可能ですけれども、パラメータの数式中の一部として使用することを出来ません。

undefine コマンドは使用しなくなったマクロ変数を解放する際に使用します。

例 3.3.2 define コマンド例

プログラム ex30301.bpl を define 文を使用して cline コマンドの最後のパラメータ「99990」をマクロ変数 EndofData でかき変えて読みやすくしてみます。

// ex30302.bpl

fmt 40 24

brl 40-10 24 #c4#c4#b

brl 7 23 ,fig4 example of define command

a4

1 = 5

m = 2

r = 360/10

pitch = 2.2

define EndofData 9999 0

x0 = l*cos(90+0*r)

 $v0 = 1*\sin(90+0*r)$

 $x1 = m*\cos(90+1*r)$

 $v1 = m*\sin(90+1*r)$

 $x2 = 1*\cos(90+2*r)$

```
y2 = l*sin(90+2*r)
x3 = m*\cos(90+3*r)
y3 = m*\sin(90+3*r)
x4 = 1*\cos(90+4*r)
y4 = l*\sin(90+4*r)
x5 = m*\cos(90+5*r)
y5 = m*\sin(90+5*r)
x6 = 1*\cos(90+6*r)
y6 = 1*\sin(90+6*r)
x7 = m*\cos(90+7*r)
y7 = m*\sin(90+7*r)
x8 = 1*\cos(90+8*r)
y8 = 1*\sin(90 + 8*r)
x9 = m*\cos(90+9*r)
y9 = m*\sin(90+9*r)
x10 = l*cos(90+10*r)
y10 = l*sin(90+10*r)
dot 1
cline pitch
   x0 y0
   x1 y1
   x2 y2
   x3 y3
   x4 y4
   x5 y5
   x6 y6
```

ff

x7 y7 x8 y8 x9 y9 x10 y10 EndofData

ex30302.bpl

3.4 点字印字コマンド

図に標題や名称及び注釈等を点字で印刷するために次の 5 つのコマンドが用意されています。

fmt 点字フォーマットコマンド

braille 点字印字コマンド

kana 仮名書きの点字印字コマンド

move 印字ヘッド移動コマンド

document 点字文書印刷コマンド

3.4.1 点字フォーマットコマンド

<書式>

fmt col row

<解説>

fmt はページ番号や標題等、点字用紙の適切な位置に点字を割り付けるフォーマットコマンドです。「col」は1行のコラム数(マス数)、「row」は1ページの行数です。

A4 用紙に片面書きで 1 行 40 マス、1 ページ 24 行の点字を割り付けるフォーマットコマンドは

fmt 40 24

となります。

B5 用紙の片面が来 1 行 32 マス、1 ページ 22 行の割り付けフォーマットコマンドは fmt 32 22

です。

点字の位置は一番左から 1 マス目、2 マス目、3 マス目……となります。また、行数は B5 用紙の場合、一番上から 22 行目、21 行目、……1 行目となります。A4 用紙の場合、一番上から 24 行目、23 行目、23 行目、……1 行目となります。

3.4.2 braille (brl) NABCC 点字印字

<書式>

braille x y string

<解説>

braille は座標(x, y)から NABCC (North American Braille Computer Code)の点字文字列 string を印字するコマンドです。コマンド名 braille は brl と略しても構いません。

ページ見出しやページ番号、図の標題や注等は、フォーマットコマンド fmt により通常の点字割り付けにフォーマットしておけば簡単に点字を配置することが可能となります。

図に点字を書き入れる場合、点字文字列の最初の点字の1の点の位置がbrl コマンドの座標 (col, row) となります。

brl の点字文字列は NABCC で書いて下さい。

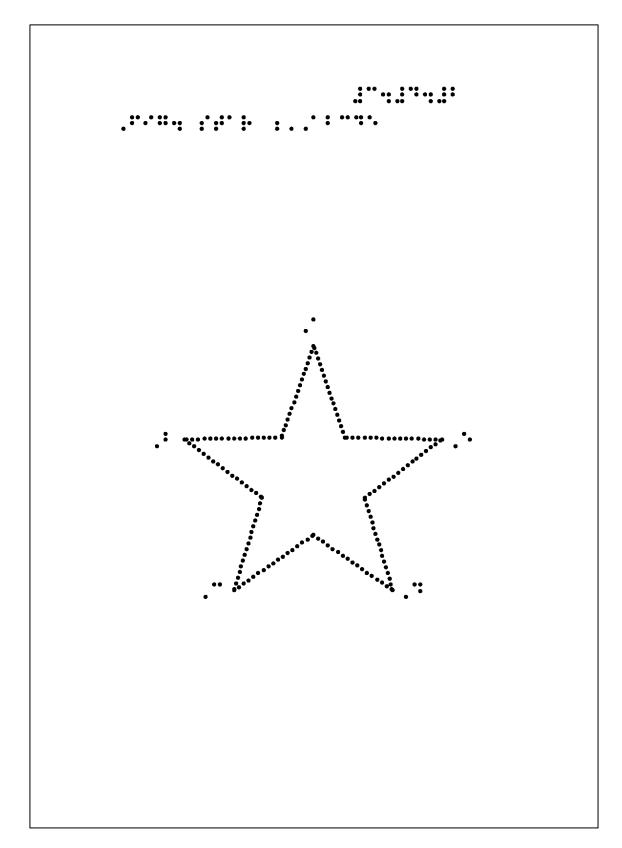
例 3.4.2 brl 点字印字

brl コマンドを使用して 3.3.2 の ex30302.bpl の折れ線の星の形状の図にページ見出しと 図の標題及び各項点に A、B、C、D、E の名称を付けてみます。

// ex30402.bplfmt 40 24 brl 40-10 24 #c4#d4#b brl 7 23 ,fig4 star ;,,abcde a4define EndofData 9999 0 1 = 5m = 2r = 360/10pitch = 2.2 $x0 = 1*\cos(90+0*r)$ $y0 = 1*\sin(90+0*r)$ $x1 = m*\cos(90+1*r)$ $y1 = m*\sin(90+1*r)$ $x2 = 1*\cos(90+2*r)$ $y2 = 1*\sin(90+2*r)$ $x3 = m*\cos(90+3*r)$ $y3 = m*\sin(90+3*r)$ $x4 = 1*\cos(90+4*r)$ $y4 = 1*\sin(90+4*r)$ $x5 = m*\cos(90+5*r)$ $y5 = m*\sin(90+5*r)$ $x6 = 1*\cos(90+6*r)$ $y6 = 1*\sin(90+6*r)$ $x7 = m*\cos(90+7*r)$ $y7 = m*\sin(90+7*r)$

```
x8 = 1*\cos(90 + 8*r)
y8 = 1*\sin(90+8*r)
x9 = m*\cos(90+9*r)
y9 = m*\sin(90+9*r)
x10 = l*cos(90+10*r)
y10 = l*sin(90+10*r)
origin 0 -2 1 1
dot 1
cline pitch
   x0 y0
   x1 y1
   x2 y2
   x3y3
   x4 y4
   x5 y5
   x6 y6
   x7 y7
   x8 y8
   x9 y9
   x10 y10
   EndofData
///
   brl x0-dx(5) y0+dy(10) ,a
   brl x2-dx(12.5) y2+dy(2),b
   brl x4-dx(12.5) y4+dy(2),c
   brl x6+dx(3) y6+dy(2),d
   brl x8+dx(3) y8+dy(2),e
```

ff



ex30402.bpl の印字結果

4 行目の brl は一番上の行 24 行目にページ番号を 40-10 マス目から印字するコマンドです。40-10 は、1 行 40 マスのうちから行末の 10 マスをページ番号に割り当て、ページ番号を 30 マス目から印字させるものです。

31 行目以降の brl は頂点に A、B、C、D、E の名称を印字するコマンドです。

xとyの座標の数式中の関数 dx, dy は拡張点の位置と点字が重ならないようにするためです。頂点の位置から点字 1 マス目の位置をミリメートル単位で離して印字して下さい。 日本の点字は通常マス間隔が 5mm、点と点の間隔が 2mm として計算して下さい。

関数 dx、dy を使用すれば、ウィンドウの座標がどのように変わろうと点字の1マス目の1の点の位置をミリメートル単位で調整することが可能です。

3.4.3 仮名書き点字印字

<書式>

kana col row string

<解説>

英数仮名の半角文字列の string を点字に変換して座標 (col, row) から点字で印字します。 仮名から点字に変換するところを除けば brl コマンドと同じです。

3.4.4 印字ヘッド移動コマンド

<書式>

move x y

<解説>

印字ヘッドを座標 (x,y) に移動させるコマンドです。次の document コマンドと共に使用します。

fmt 点字割り付けフォーマットまたは B5 や A4 等のウィンドウコマンドの座標系で指定して印字へッドを移動させます。

3.4.5 点字ドキュメント印刷コマンド

<書式>

document filename

<解説>

Bplot は filename で指定された NABCC ファイルを読み込み印刷します。ファイルの内容は B5 用紙なら 1 行 32 マス、A4 用紙なら 1 行 40 マスというようにフォーマットしておく必要があります。

文字列の印字開始位置は印字ヘッドが現在ある位置です。印字ヘッドが点字用紙の一番 左上にある場合は通常の点字印刷をし始めます。

必要があれば前節の印字ヘッド移動コマンド move でヘッドを動かして印刷して下さい。

3.5 ページ送りコマンド

図を印字した後、プリンタ用紙を 1 ページ送るために 2 つのページ送りコマンド ff と formfeed が用意されています。

ff はページを送るコマンドです。

formfeed はページ送りを制御するコマンドです。

3.5.1 ページ送りコマンド

<書式>

ff (with no parameter)

<解説>

作図コマンドプログラムの最後にページ送りコマンド ff を記述して下さい。ページが送られ新しい点字用紙がセットされます。本解説のプログラム例の最後には常に ff が記述されています。

3.5.2 ページ送り制御コマンド

<書式>

formfeed flag

<解説>

ページ送り制御コマンド formfeed はページ送りを制御します。 パラメータ flag は 0 または 1 を取ります。

flag が 0 の場合はページ送りコマンド ff を無効にします。後述の部品貼り付けコマンド paste 及び回転貼り付けコマンド rpaste を使用して ff を含む部品プログラムを貼り付けて 1 枚の図を作成することが可能となります。もし ff コマンドが有効ならば貼り付けた部品プログラムがそのつどページが送られ部品がバラバラになってしまいます。

flag が 1 の場合、プリンタにページ送りを指令すると共に、その後の ff コマンドを有効にします。

部品等子どものプログラムファイルを親のプログラムファイルに貼り付けて 1 枚の図を作成するためには親のプログラムの最初に formfeed 0 を記述して下さい。また、最後にformfeed 1 を指定して下さい。作成された図がページ送りされて出てきます。

3.6 点の大きさや点間隔設定コマンド

点の大きさや点線の点間隔を設定することが可能です。点の大きさや点線の点間隔及び破線を設計することにより触読しやすい図を作成することが出来ます。関連するコマンドやパラメータが5つあります。

dot 点の大きさ指定コマンド

dotpitch 点線の点間隔を指定するパラメータ

plot 打点コマンド

point 特大点打点コマンド

design 破線設計コマンド

3.6.1 dot 点の大きさ指定コマンド

< 去去>

dot dotsize

<解説>

高解像度点字プリンタ ESA721 及び ESA600G には 3 種類の打点ピンを備えています。 dot コマンドは 3 種類の打点ピンを切り替えて点や点線及び実線等を印字します。

パラメータ dotsize は、0 が小点、1 が中点、2 が大点を指定します。1 の点は通常の点字の点と同じ大きさです。

3.6.2 点線の点間隔設定

<書式>

dotpitch

<解説>

直線系や曲線系の点線や実線の作図コマンドの最後のパラメータにより点線の点間隔を 設定することが出来ます。点間隔はミリメートル単位で指定して下さい。指定された点間 隔で点線や実線が印字されます。拡大や縮小等ウィンドウの座標系がどのように変わろう とも点線や実線の点間隔はミリメートル単位で変わりません。

例 3.6.2 dot と点間隔

次の表のように 3 種類の点種と点間隔を変えることによりいろいろな触感覚の点線や実線を引くことが可能となります。直線 line コマンドを使用して種々の直線を引いてみます。

	点種	線種	点間隔
1本目	小点	実線	1.8mm
2本目	小点	実線	2.2mm
3本目	小点	点線	5.0mm
4本目	中点	実線	2.2mm
5 本目	中点	実線	2.8mm
6本目	中点	点線	5.0mm
7本目	大点	実線	2.8mm
8本目	大点	実線	3.4mm
9本目	大点	点線	5.0mm

次のプログラム ex30602.bpl の 9 本の点線の始点の x 座標と終点の x 座標は変わらないため変数 x0 と x1 を変数定義コマンドで定義しています。また、点線は横線であり、始点と終点の y 座標も一致しますので変数 y で定義しています。次の点線を引くたびに y 座標を計算し直しています。

// ex30602.bpl

fmt 40 24 $brl\ 40\mbox{-}10\ 24\ \#c4\mbox{-}14\mbox{-}4 b \\ brl\ 7\ 23\ ,fig4\ dotted\ and\ solid\ lines$

a4

x0 = -5

x1 = 5

y0 = 10

dy = 2

dot 0

y = y0

line x0 y x1 y 1.8

y = y0-1*dy

line x0 y x1 y 2.2

y = y0-2*dy

line x0 y x1 y 5.0

dot 1

y = y0-3*dy

line x0 y x1 y 2.2

y = y0-4*dy

line x0 y x1 y 2.8

y = y0-5*dy

line x0 y x1 y 5.0

dot 2

y = y0-6*dy

line x0 y x1 y 2.8 y = y0-7*dyline x0 y x1 y 3.4 y = y0-8*dyline x0 y x1 y 5.0

ff

このように点種と点間隔を変えればバラエティーに富む線種を実現することが可能となります。

ただし、点間隔は、小点が 1.8mm 中点が 2.0mm 大点が 2.8mm 程度が最小として下さい。それ以下にしますと点が繋がって紙が切れる恐れがあります。

••••••
• • • • • • • • • • • • • • • • • • • •
•••••••••
••••••
• • • • • • • • • • • • • • • •

ex30602.bpl の印字結果

3.6.3 plot 打点コマンド

<書式>

plot x y

<解説>

plot は座標 (x, y) に打点するコマンドです。点の大きさは打点コマンドの記述の前に点の大きさ設定コマンド dot で指定しておいて下さい。

3.6.4 point 特大点コマンド

<書式>

point x y

<解説>

point は、 $\det 2$ の大点よりも大きい点を座標(\mathbf{x} , \mathbf{y})に打つコマンドです。ひし形に中点で 4 点打った上に中央に大点を打って実現しています。

点字プリンタの精度の関係で若干ひしゃげることがあります。

例 3.6.4 plot と point による打点

plot コマンドで小点・中点・大点を打ち更に point で特大点を放射状に打ってみます。

// ex30604.bpl

fmt 40 24

brl 40-10 24 #c4#f4#d

brl 7 23 ,fig4 four kinds of dots

a4

dot 0

r = 2

plot r*cos(90) r*sin(90)

plot r*cos(150) r*sin(150)

plot r*cos(210) r*sin(210)

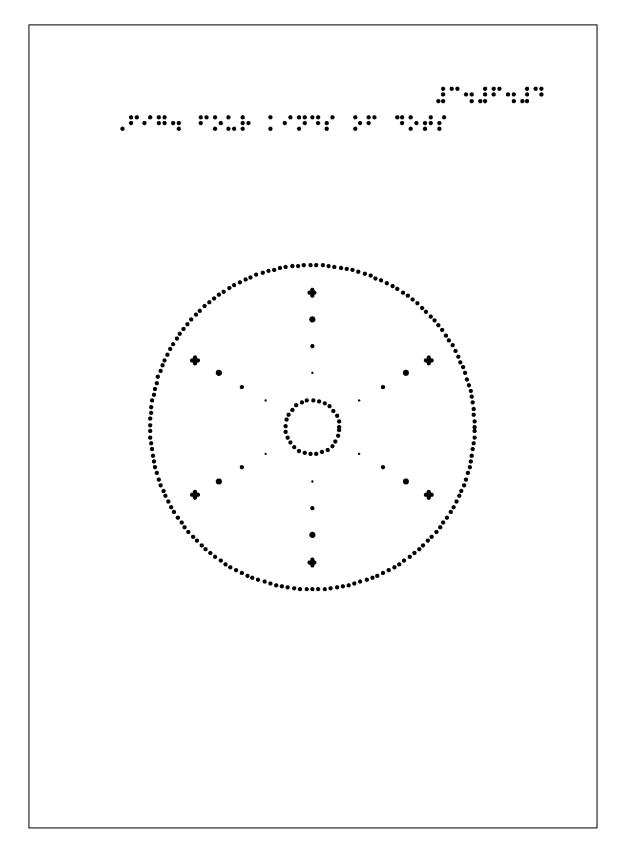
plot r*cos(270) r*sin(270)

plot r*cos(330) r*sin(330)

plot r*cos(390) r*sin(390)

```
dot 1
r = 3
  plot r*cos(90) r*sin(90)
  plot r*cos(150) r*sin(150)
  plot r*cos(210) r*sin(210)
  plot r*cos(270) r*sin(270)
  plot r*cos(330) r*sin(330)
  plot r*cos(390) r*sin(390)
dot 2
r = 4
  plot r*cos(90) r*sin(90)
  plot r*cos(150) r*sin(150)
  plot r*cos(210) r*sin(210)
  plot r*cos(270) r*sin(270)
  plot r*cos(330) r*sin(330)
  plot r*cos(390) r*sin(390)
r = 5
  point r*cos(90) r*sin(90)
  point r*cos(150) r*sin(150)
  point r*cos(210) r*sin(210)
  point r*cos(270) r*sin(270)
  point r*cos(330) r*sin(330)
  point r*cos(390) r*sin(390)
dot 1
   circle 0\ 0\ 1\ 0\ 360\ 2.2
   circle 0\ 0\ 6\ 0\ 360\ 2.2
```

ff



ex30604.bpl の印字結果

3.6.5 design 破線設計コマンド

<書式>

design flag seg0 gap0 seg1 gap1

<解説>

design は破線を設計するコマンドです。点線や実線に加えて破線の使用が可能となります。パラメータは4つです。

パラメータ「flag」は3つの値を取ります。「1」は、直線系及び曲線系のすべての点線を design で設計された破線で印字する開始指令を有効にします。「0」は破線設計指令を無効 にします。また、「2」は破線設計の指令を保存しながら実行を一時的に保留します。「3」は保留された設計指令を有効にします。

「flag」の後の4つのパラメータは破線を構成する2つの点パターンを定義します。2つの点パターンを繰り返すことにより破線の印字を実現しています。

「seg0」と「gap0」は破線の最初の点パターンを定義します。何点打ってその後何点飛ばすかを指定します。

「seg1」と「gap1」は 2 番目の点パターンを定義します。何点打ってその後何点飛ばすかを指定します。

この2つの点パターンを繰り返しては線が印字されます。

例 3.6.5 破線デザイン例

小点と中点を使用して破線を印字してみます。基本の点間隔は、小点が 1.8mm 中点が 2.0mm とします。

点パターンは5種類です。

- 1. 2点打って1点休む
- 2. 2点打って2点休む
- 3. 2点打って2点休み1点打って2点休む
- 4. 3点打って2点休む
- 3点打って2点休み、1点打って2点休むです。

// ex30605.bpl

 $\mathrm{fmt}\ 40\ 24$

brl 40-10 24 #c4#f4#e

brl 7 23 ,fig4 examples of broken lines

a4

x0 = -6

x1 = 6

y0 = 9

dy = 2

dot 0

pitch = 1.8

y = y0

 $\operatorname{design} 1\ 2\ 1\ 2\ 1$

line x0 y x1 y pitch

y = y-dy

 $\operatorname{design} 1 \ 2 \ 2 \ 2 \ 2$

line x0 y x1 y pitch

y = y-dy

 $\operatorname{design} 1 \ 2 \ 2 \ 1 \ 2$

line x0 y x1 y pitch

y = y-dy

 $\operatorname{design} 1\ 3\ 2\ 3\ 2$

line x0 y x1 y pitch

y = y-dy

 $\operatorname{design} 1\ 3\ 2\ 1\ 2$

line x0 y x1 y pitch

dot 1

pitch = 2.0

y = y-dy

 $\operatorname{design} 1\ 2\ 1\ 2\ 1$

line x0 y x1 y pitch

y = y - dy

design 1 2 2 2 2

line x0 y x1 y pitch

y = y - dy

design 1 2 2 1 2
 line x0 y x1 y pitch
y = y-dy
design 1 3 2 3 2
 line x0 y x1 y pitch
y = y-dy
design 1 3 2 1 2

line x0 y x1 y pitch

 $\operatorname{design} 0 \ 0 \ 0 \ 0 \ 0$

ff

•• •• •• •• •• •• •• •• •• •• •• •• ••
•• •• •• •• •• •• •• •• •• •• ••
•• • •• • • • • • • • • • • • • •
••• ••• ••• ••• ••• ••• ••• •••
••• • ••• • ••• • ••• • ••• • •••

ex30605.bpl の印字結果

3.7 注釈コマンド

注釈コマンドには2つの役割があります。第1は、プログラムファイルに標題を付けたりアルゴリズム(算法)を説明したり変数に注を付ける等、プログラムファイルに適宜注釈(コメント)を付けることです。第2は、作図プログラムの印字テストを実施する際、テストに無関係な領域を注釈にすることにより印字時間を節約したり、エラーの発生箇所を発見しやすくしたり効率的にプログラムテストを実施するためてす。

注釈行または注釈領域は Bplot により処理されることを無く、図の印字には関係しません。

注釈には2種類の注釈コマンドがあります。

// 注釈行設定コマンド

/**/ 注釈領域設定コマンド

3.7.1 注釈行設定コマンド

<書式>

//

<解説>

プログラムファイルの行頭にスラッシュ記号を二つ以上連続して書くとその行が注釈となります。また、作図コマンドの行のオワリに空白をおいてからスラッシュ記号を二つ以上連続して書くと行末までが注釈となります。スラッシュを二つ以上書かずに注釈を書きますとエラーが発生します。

braille (brl)、kana、define コマンドの後の文字列に注釈文をつけることはできません。標題や変数や注等、プログラムに出来る限り丁寧に注釈を付けておけば他者にとってプログラムのアルゴリズムの理解が容易になるだけでなく、自己にとっても後から読み返したときに容易にアルゴリズムを思い出すことが出来ます。

3.7.2 注釈領域設定コマンド

<書式>

/* ··· */

<解説>

行頭にスラッシュとアステリスク記号を書いた行から行頭にアステリスクとスラッシュ記号を書いた行までの行全体を注釈とするコマンドです。注釈領域設定コマンドによって数行をまとめて注釈行として必要な注釈を書いておくことが可能となります。また、プログラムファイルの印字テストにおいてテストに無関係な領域を注釈化(コメントアウト)し、テストを効率的に実施することも可能となります。

例 3.7.2 コメントアウトの例

次のプログラム ex30702.bpl は五芒星の図のプログラム ex30402.bpl に対する注釈コマンドの例です。1行目はプログラム名を注釈行コマンドを使用してテストした例です。1行目はプログラム名の注釈行です。

17 行目 \sim 32 行目は注釈領域コマンドの例です。五芒星の星形の印字を注釈化(コメントアウト)し、頂点の点字が正確であるかどうかをテストした例です。頂点の記号 $A\sim$ E だけが印字されています。

```
ex30702.bpl
fmt 40 24
   brl 40-16 24 #c4#g4#b
   brl 7 23 ,fig4 example of commentout
a4
define eod 9999 0
1 = 5
m = 2
r = 360/10
pitch = 2.2
origin 0 -2 1 1
/*
dot 1
cline pitch
   1*\cos(90+0*r) 1*\sin(90+0*r)
   m*\cos(90+1*r) m*\sin(90+1*r)
   1*\cos(90+2*r) 1*\sin(90+2*r)
   m*\cos(90+3*r) m*\sin(90+3*r)
   1*\cos(90+4*r) 1*\sin(90+4*r)
   m*\cos(90+5*r) m*\sin(90+5*r)
   1*\cos(90+6*r) 1*\sin(90+6*r)
   m*\cos(90+7*r) m*\sin(90+7*r)
   1*\cos(90+8*r) 1*\sin(90+8*r)
```

```
\begin{array}{c} m*\cos(90+9*r) \ m*\sin(90+9*r) \\ l*\cos(90+10*r) \ l*\sin(90+10*r) \\ eod \\ */\\ ///\\ \\ brl \ l*\cos(90+0*r) \cdot dx(5) \ l*\sin(90+0*r) + dy(10) \ ,a \\ brl \ l*\cos(90+2*r) \cdot dx(12.5) \ l*\sin(90+2*r) + dy(2) \ ,b \\ brl \ l*\cos(90+4*r) \cdot dx(12.5) \ l*\sin(90+4*r) + dy(2) \ ,c \\ brl \ l*\cos(90+6*r) + dx(3) \ l*\sin(90+6*r) + dy(2) \ ,d \\ brl \ l*\cos(90+8*r) + dx(3) \ l*\sin(90+8*r) + dy(2) \ ,e \\ \end{array}
```

ff

ex30702.bpl の印字結果

3.8 直線系定形図形の作成

流れ図やグラフ等を作成するため必要とされる直線系の定形的な図形作図コマンドが 11 種類用意されています。直線や矢印及び座標軸を描くコマンドが 6 種類、箱形やひし形等直線で囲まれた図形が 5 種類用意されています。

line 直線印字コマンド

hatch ハッチング用点線印字コマンド

arrow 始点と終点を指定する矢印印字コマンド

dart 始点と方向と長さを指定する矢印印字コマンド

xaxis x軸を印字するコマンド

yaxis y軸を印字するコマンド

box 箱形印字コマンド

diamod ひし形印字コマンド

polygon 正多角形印字コマンド

parallelogram 平行四辺形印字コマンド

rectangle 長方形印字コマンド

3.8.1 line 直線印字コマンド

<書式>

line x0 y0 x1 y1 pitch

<解説>

line は、始点の座標(x0, y0)と終点の座標(x1, y1)及び点線の点間隔 pitch を指定して任意の直線(線分)を描くコマンドです。始点と終点の座標はウィンドウの任意の点を指定できます。点字プリンタの印字ヘッドが効率的に動けるように、始点と終点を上手に指定すれば図を高速に印字することが可能となります。

3.8.2 ハッチング用点線コマンド

<書式>

hatch x0 y0 x1 y1 unit

<解説>

hatch は始点の座標(x0, y0)と終点の座標(x1, y1)及び点線の点間隔 unit をウィンドウ座標系で指定する点線コマンドです。line コマンドのようにウィンドウ座標系が拡大や縮小によりどのように変化しようとも点間隔が変わらないのに対し hatch コマンドの点間隔 unit はウィンドウ座標系の拡大や縮小に伴って同じように変化します。図形の内部を一定の点パターンで埋めるハッチングに使用すれば点パターンの形や点の数を変えずにウィンドウの座標系の変化に応じて変化させることが可能となります。

3.8.3 arrow 矢印コマンド

<書式>

arrow x0 y0 x1 y1 head_angle head_length head_type pitch

<解説>

arrow は始点の座標(x0, y0)と終点の座標(x1, y1)を指定して矢印を描くコマンドです。

「head_angle」と「head_length」及び「head_type」の3つのパラメータにより矢尻の形状を指定することが可能です。「head_angle」と「head_length」により矢尻の角度と長さを指定できます。「head_type」により4種類の矢尻の形状を指定できます。0は左右対称の矢尻となります。1は左肩側だけの矢尻となります。2は反対に右肩側だけの矢尻となります。3は矢尻のない矢印となります。

矢尻の角度「head_angle」は初期設定時(デフォルトの場合)度数で指定して下さい。もし 弧度法で指定する場合は後述の radian コマンドを発行しておいて下さい。弧度法から度数 指定に戻すためには degree コマンドを発行して下さい。

pitch は点線の点間隔です。

なお、破線設計コマンド design によって点線が破線に変えられても矢尻はもとの点間隔で描かれます。一般に矢尻は短いため破線に変えると矢尻として読み取りにくくなるためです。

3.8.4 dart 矢印コマンド

< 去書>

dart x0 y0 angle length head_angle head_length head_type pitch

<解説>

dart は始点の座標(x0, y0)と矢の方向 angle 及び矢の長さ length を指定して矢印を描くコマンドです。

「head_angle」と「head_length」及び「head_type」の3つのパラメータにより arrow コマンドと同じく 4 種類の矢尻の形状を指定することが可能です。「head_angle」と「head_length」により矢尻の角度と長さを指定できます。「head_type」により 4 種類の矢尻の形状を選択することが可能です。0 は左右対称の矢尻です。1 は左肩側だけの矢尻です。2 は右肩側だけの矢尻です。3 は矢尻のない矢印です。

矢の方向「angle」と矢尻の角度「head_angle」は初期設定時(デフォルトの場合)度数で指定して下さい。もし弧度法で指定する場合は後述の radian コマンドを発行しておいて下さい。弧度法から度数指定に戻すためには degree コマンドを発行して下さい。

pitch は矢印の点線の点間隔です。

なお、破線設計コマンド design によって点線が破線に変えられても矢尻はもとの点間隔で描かれます。一般に矢尻は短いため破線に変えると矢尻として読み取りにくくなるため

です。

3.8.5 座標軸作図コマンド

各種のグラフを作成するために 1 コマンドで x 軸または y 軸の座標軸を描くコマンドが 用意されています。

<書式>

xaxis x0 x1 unit pic pitch yaxis y0 y1 unit pic pitch

<解説>

座標軸作図コマンドによりウィンドウの座標系のx軸及びy軸上に座標軸を各々1コマンドで描くことが出来ます。

xaxis は x 座標軸を、yaxis は y 座標軸を描くコマンド名です。各コマンドは 5 つのパラメータを必要とします。

x0 と x1 または y0 と y1 は座標軸を描く x 軸または y 軸上の始点と終点の座標を指定します。

unit は座標軸に付ける目盛りの間隔を指定します。

pic は座標軸上の目盛りの長さを指定します。整数値で指定して下さい。pic の値が目盛りの点線の点の数になります。pic の正・負により pic の方向が変わります。pic が正ならば x 座標軸は上向きに、y 座標軸は右向きに目盛りが付けられます。一方、pic が負ならば x 座標軸は下向きに、y 座標軸は左向きに目盛りが付けられます。

pitch は点線の点間隔です。

例 3.8.5 座標軸の作図例

プリンタ用紙の中心が原点であるとき、x 軸上に-10cm ~ 10 cm、y 軸上に-10cm ~ 10 cm の範囲に座標軸を引いてみます。目盛りの間隔 unit は 1cm とします。また、目盛りは 2 点とします。x 軸の上向き、y 軸の右向きに目盛りを付けます。

// ex30805.bpl

fmt 40 24

brl 40-10 24 #c4#h4#e

brl 7 23 ,fig4 x axis and y axis

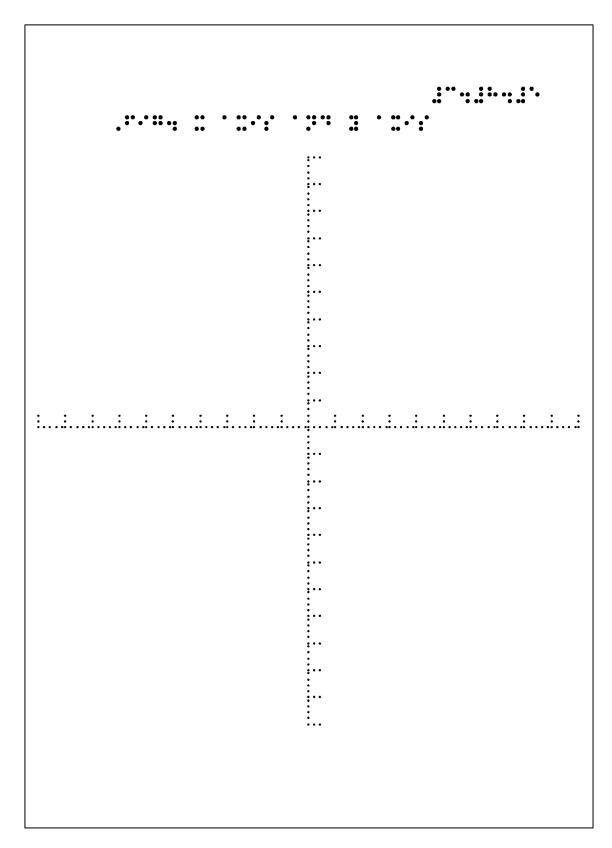
a4

unit = 1

```
pic = 2
pitch = 2.2

dot 0
    xaxis -10 10 unit pic pitch
    yaxis -10 10 unit pic pitch
```

ff



ex30805.bpl の印字結果

3.8.6 box 箱形作図コマンド

<書式>

box x0 y0 wide hight pitch

<解説>

box は箱の中心(重心)の座標(x0, y0)を指定して箱を描くコマンドです。流れ図の処理カードの作図等に使用します。

「wide」は箱の中心から左右の辺までの距離(長さ)です。

「hight」は箱の中心から上下の辺までの距離(長さ)です。

「pitch」は箱を描く点線の点間隔です。

3.8.7 diamond ひし形作図コマンド

<書式>

diamond x0 y0 wide hight pitch

<解説>

diamod はひし形の中心(重心)の座標(x0,y0)を指定してひし形を描くコマンドです。

「wide」は中心から左右の頂点までの距離(長さ)です。

「hight」は中心から上下の頂点までの距離(長さ)です。

「pitch」は点線の点間隔です。

3.8.8 polygon 正多角形作図コマンド

<書式>

polygon n x0 y0 r pitch

<解説>

polygon は正三角形や正五角形等、正多角形を1コマンドで描くコマンドです。1つの頂点が図形の中心の真上に来る左右対称な正多角形が描かれます。

「n」は正多角形の頂点の数です。

「x0」と「y0」は中心の座標(x0, y0)です。

「r」は中心から頂点までの距離(長さ)です。

「pitch」は点線の点間隔です。

3.8.9 parallelogram 平行四辺形作図コマンド

<書式>

parallelogram x0 y0 wide hight theta pitch

<解説>

parallelogram は流れ図の入出力カード等平行四辺形を描くコマンドです。パラメータは 箱形 box コマンドより一つ多い 6 個必要です。 「x0」と「y0」は平行四辺形の中心(重心)の座標(x0, y0)です。「wide」は中心から左右の辺までの水平距離(長さ)です。

「hight」は中心から上下の辺までの距離(長さ)です。

「theta」は左右の辺が底辺に対する角度です。初期設定時(デフォルトの場合)、角度は度数で指定して下さい。弧度法で指定する場合は後述の radian コマンドを発行しておいて下さい。弧度法による角度指定から度数指定に戻すには degree コマンドを使用します。

「pitch」は点線の点間隔です。

3.8.10 rectangle 長方形作図コマンド

<書式>

rectangle x0 y0 x1 y1 pitch

<解説>

rectangle は box コマンドと同様、長方形を描くコマンドです。箱形作図コマンド box とは異なり、rectangle コマンドは長方形の対角線の両端の座標を指定します。

[x0] と [y0] は長方形の対角線の一方の頂点の座標 (x0, y0) です。

「x1」と「y1」は長方形の対角線の他方の頂点の座標(x1, y1)です。

「pitch」は点線の点間隔です。

3.9 直線系不定形図形の作成

直線系の不定形な折れ線図形の作図コマンド cline (concatinated lines)が用意されています。前節 3.8 節の直線系の作図コマンドは、箱形やひし形等定形的な図形の作図コマンドに対し、折れ線作図コマンド cline は定形・不定形にかかわらず自由に折れ線を作図することが可能です。始点から終点まで屈曲点の座標を書き並べることによって折れ線を効率的に描くことが出来ます。直線コマンド line を繋いで描くよりも簡単に書くことができます。折れ線グラフや不定形な直線系の図形を描くのに最適です。

<書式>

cline pitch

x0 y0

x1 ya

..

••

99990

<解説>

折れ線作図コマンド cline は複数行からなるコマンドです。

1 行目に作図コマンド名 cline と点線の点間隔「pitch」を書きます。

2 行目以降に始点、屈曲点、終点の座標を各々行を変えて書きます。最後に、折れ線の座標データの終了を示す「9999 0」を書きます。

cline は「99990」を読み込みますと折れ線の座標データを読み込んだものと判断し、作図を開始します。

例 3.3.1、例 3.3.2、例 3.7.2 の星の形は cline により描かれています。

例 3.9.1 1 から n までの自然数の和を求める流れ図

3.8 節及び 3.9 節の作図コマンドを使用して 1 から n までの自然数の和を求める流れ図を作成してみます。ただし、「はじめ」と「おわり」を示す終端カードは楕円形です。楕円形の作図コマンドは次節の 3.10 節で解説します。

流れ図は次の $1.\sim7.$ の流れとなります。

1. 流れは楕円形の「はじめ」の開始カードから始まります。

流れ線の矢印は次の2.に向かいます。

2. 次に「n を入力する」平行四辺形の入出力カードに入ります。

流れ線の矢印は3.に向かいます。

- 3. i と s に初期値 0 を代入する $\lceil i=0 \rceil$ s=0 」の箱形の処理カードに入ります。 流れ線は下の 4 に向かいます。
- 4. $i \ b \ s \ b \ 1$ だけ加算する $[i=i+1 \ s=s+1]$ の箱形の処理カードに入ります。 流れ線は下の 5.のひし形に向かいます。
- 5. 次にi が入力n に到達したかどうかを判断するひし形の $\lceil i \ge n \rfloor$ の判断カードに入り、y (yes) とn (no) に分岐します。

流れ線は、もしy、すなわち、i がn に到達していれば次の6.の平行四辺形の入出力カードに進みます。

もしn、すなわち、iがnに到達していなければ上の4.の処理カードに戻ります。

6. 「sを出力する」平行四辺形の入出力カードに進みます。

流れ線は下の7.の終了カードに向かいます。

7. 流れは楕円形の「おわり」のカードに入り終了となります。

// ex30900.bpl

fmt 40 24

brl 40-10 24 #c4#i4#j

brl 7 23 ,fig4 flow chart of sum of numbers

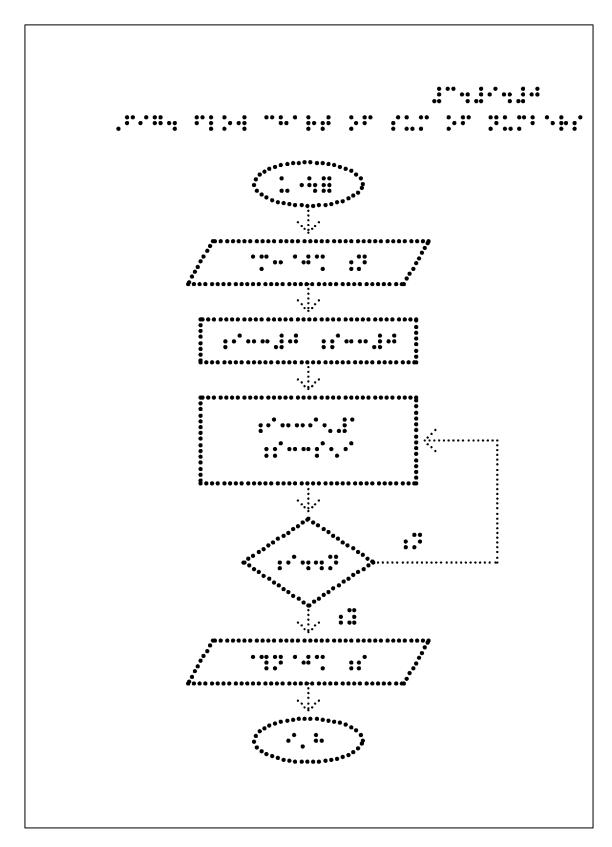
a4

```
define eod 9999 0
pitch1 = 2.2
pitch0 = 1.8
x0 = 0
y0 = 9
w = 4
h = 0.8
1 = 1.3
///
dot 1
brl x0-dx(10) y0+dy(2) u"|=
   ellipse x<br/>0 y<br/>00.5\ensuremath{^*w}h0360 pitch<br/>1
y1 = y0-h-l-h
brl x0-dx(22.5) y1+dy(2) `m3`j% ;n
   paralleogram x0 y1 w h 60 pitch1
y2 = y1-h-l-h
brl x0-dx(32.5) y2+dy(2) ;i33#j ;s33#j
   box x0 y2 w h pitch1
y3 = y2-h-l-2*h
brl x0-dx(20) y3+dy(8) ;i33i5#a
brl x0-dx(17.5) y3-dy(1) ;s33s5i
   box x0 y3 w 2*h pitch1
y4 = y3-2*h-l-2*h
brl x0-dx(12.5) y4+dy(2) ;i44n
   dia x0 y4 0.6*w 2*h pitch1
y5 = y4-2*h-l-h
brl x0-dx(22.5) y5+dy(2) `?n`j% ;s
```

```
paralleogram x0 y5 w h 60 pitch1
y6 = y5-h-l-h
brl x0-dx(7.5) y6+dy(2) i'h
   ellipse x0 y6 0.5*w h 0.360 pitch1
brl x0+dx(10) y4-2*h-dy(2) ;y
brl x0+0.6*w+dx(10) y4+dy(10) ;n
///
dot 0
theta = 45
length = 0.6
12 = 7
arrow x0 y0-h x0 y0-h-l+dy(3) theta length 0 pitch0
arrow x0 y1-h x0 y1-h-l+dy(3) theta length 0 pitch0
arrow x0 y2-h x0 y2-h-l+dy(3) theta length 0 pitch0
arrow x0 y3-2*h x0 y3-2*h-l+dy(3) theta length 0 pitch0
cline pitch0
   x0+0.6*w y4
   12 y4
   12 y3
   eod
arrow 12 y3 x0+w+dx(3) y3 theta length 0 pitch0
arrow x0 y4-2*h x0 y4-2*h-l+dy(3) theta length 0 pitch0
arrow x0 y5-h x0 y5-h-l+dy(3) theta length 0 pitch0
```

ff

83



ex30900.bpl の印字結果

3.10 曲線系定形図形の作図

曲線系の定形図形作図コマンドが3種用意してあります。 円形と楕円形及び楕円の弧を描くコマンドです。

circle 円の作図 ellipse 楕円の作図 oval 楕円の作図

3.10.1 circle 円の作図コマンド

<書式>

circle x0 y0 r theta0 theta1 pitch

<解説>

中心の座標(x0, y0)と半径「r」及び描く円弧の中心角の範囲「theta0」と「theta1」を指定して円または円弧を描きます。最後のパラメータ「pitch」は描く点線の点間隔を指定します。

中心角「theta0」と「theta1」は度でも弧度でも指定できます。デフォルトは度です。 弧度で指定する場合は、角度を弧度にするコマンド radian をプログラムの前の部分で書い ておいて下さい。もし度に戻す場合は degree コマンドを書いておいて下さい。

点間隔「pitch」が正の場合は正の方向(逆時計回り)に円弧を描きます。負の場合は負の方向(時計回り)に円弧を描きます。デフォルトは逆時計回りです。

3.10.2 ellipse 楕円作図コマンド

<書式>

ellipse x0 y0 r0 r1 theta0 theta1 pitch

<解説>

楕円は楕円の中心(重心)の座標(x0, y0)、楕円形の x 軸方向の半径「r0」と y 軸方向の半径 [r1] を指定して描きます。

描く楕円の弧の範囲は弧に対する中心角の範囲「theta0」と「theta1」で指定して下さい。角度は度でも弧度でも指定可能です。指定方法は前節 3.10.1 の円の指定方法と同じです。

「pitch」は楕円の点線の点間隔です。pitch が正の場合は弧を逆時計回りに描きます。負の場合は、弧を時計回りに描いていきます。

3.10.3 oval 楕円の弧の作図コマンド

前節 3.10.2 の楕円コマンド ellipse とは異なり、楕円の中心座標と描く弧の始点と終点の座標を指定して楕円の弧を描くことが出来ます。

<書式>

oval x0 y0 x1 y1 x2 y2 pitch

<解説>

「x0」「y0」は楕円の中心座標(x0, y0)です。

「x1」「y1」は楕円形の弧の始点の座標(x1, y1)です。「x2」「y2」は楕円形の弧の終点の座標(x2, y2)です。

「pitch」は点線の点間隔です。

3.11 曲線系不定形図形の作成

spline(スプライン)関数を使用してなめらかな曲線図形を描く作図コマンドが使用可能です。spline 関数は始点から終点まで各点をなめらかな曲線で結ぶ自由曲線の作図コマンドです。特に曲線グラフを描く際に最適です。

図形には始点と終点が異なる開いた図形(開図形)と線がすべて繋がっている閉じた図形 (閉図形) があります。spline 関数は、開図形及び閉図形を描くことが可能です。

<書式>

spline x0 x1 y0 y1 function px py pitch

 $x_0 y_0$

 $x_1 y_1$

 $x_2 y_2$

••

99990

<解説>

自由曲線作図コマンド spline は複数行からなるコマンドです。

1行目にコマンド名 spline に続けて8個のパラメータが並びます。

最初の 4 個のパラメータは図形を描く範囲を定義します。図形の定義域(x 座標の範囲)を [x0] と [x1] によって、図形の値域(y 座標の範囲)を [y0] と [y1] によって記述します。 その後の [x1] は使用する spline 関数の番号です。開図形は x1 を指定して下さい。 閉図形は x1 を指定して下さい。

その後に描かれる図形をx軸方法及びy軸方法に平行移動させるための値「px」と「py」を記述して下さい。

最後に点線の点間隔 pitch を指定して下さい。

2 行目以降、始点から終点までの点の座標を行を変えて記述して下さい。

最後の行の「9999 0」は作図データが終了したことを spline 作図コマンドに通知する

ためのものです。spline は「9999 0」を読み込みますと曲線が通る点の座標の列が終了したと判断して曲線を作図し始めます。

spline コマンドの 1 行目の最初の 4 つのパラメータにより曲線の表示領域を自由に変えることが可能です。

5番目のパラメータ「function」は描く曲線によって選択して下さい。 $1\sim3$ は開曲線を描く際に使用します。1から3のうち、曲線が適切に描かれるものを選んで下さい。4は閉曲線を描く関数番号です。閉曲線の場合、終点と始点は自動的に結ばれますので始点と同じ座標の終点の座標は書く必要はありません。

例 3.11.1 曲線で描かれた星

spline 関数を使用して例 3.3.1 の折れ線で描かれた星をなめらかな曲線で結ばれた星に描きかえてみます。

spline コマンドの図形の定義域を指定する 4 つのパラメータとデータの終わりを示す記号はマクロ変数 range と eod (end of data)で定義しています。

// ex31101.bpl

fmt 40 24

brl 40-10 24 #c4#aa4#a

brl 7 23 ,fig4 star of smooth line

a4

1 = 6

m = 3

r = 360/10

pitch = 2.2

 $x0 = 1*\cos(90+0*r)$

 $y0 = 1*\sin(90+0*r)$

 $x1 = m*\cos(90+1*r)$

 $y1 = m*\sin(90+1*r)$

 $x2 = 1*\cos(90+2*r)$

 $y2 = 1*\sin(90+2*r)$

x3 = m*cos(90+3*r)

 $v3 = m*\sin(90+3*r)$

```
x4 = 1*\cos(90+4*r)
y4 = 1*\sin(90+4*r)
x5 = m*\cos(90+5*r)
y5 = m*\sin(90+5*r)
x6 = 1*\cos(90+6*r)
y6 = 1*\sin(90+6*r)
x7 = m*\cos(90+7*r)
y7 = m*\sin(90+7*r)
x8 = 1*\cos(90+8*r)
y8 = 1*\sin(90+8*r)
x9 = m*\cos(90+9*r)
y9 = m*\sin(90+9*r)
x10 = l*cos(90+10*r)
y10 = l*sin(90+10*r)
define range -10 10 -12 12
define EndofData 9999 0
func = 4
dot 1
spline range func 0 0 pitch
   x0 y0
   x1 y1
   x2 y2
   x3y3
   x4 y4
   x5 y5
   x6 y6
   x7 y7
   x8 y8
   x9 y9
   EndofData
```

ff



ex31101.bpl の印字結果

例 3.11.2 曲線グラフの作成

spline コマンドを使用して実験で得られたテストデータから曲線グラフを作成してみます。

グラフの横軸は解答終了率です。解答終了率は、テストが始まってから解答を終了した 被験者の割合です。

縦軸は、通常条件の被験者の解答所要時間に対する通常条件または擬似弱視条件の被験者の解答所要時間の倍率です。

グラフは解答終了率の増加と共に解答所要時間の倍率がどのように変化するかを表した ものです。

テストデータの x 座標の解答終了率は $0.1\sim0.9$ ですのでデフォルトのウィンドウの x 座標の範囲- $10\sim10$ では大きすぎます。window コマンドを使用してウィンドウを変えています。window の x 座標の範囲は- $0.12\sim1.4$ 、y 座標の範囲は- $0.6\sim5$ に取ってあります。マクロ変数 range で定義し、window コマンドと spline コマンドで参照しています。

通常条件に対する通常条件の解答所要時間の倍率のグラフは常に 1 ですのでグラフは直線コマンド line で描くことが出来ます。

一方、通常条件に対する擬似弱視条件の解答所要時間の倍率のグラフは、テストデータを spline コマンドでなめらかに結んで描かれています。ほぼ2程度です。擬似弱視条件の場合、通常条件の2倍程度の解答所要時間を必要とすることが知られます。

グラフの名称や座標軸の目盛り等の数値は、window 座標系に関係なくミリメートル単位で指定できる dx と dy 関数を使用して brl コマンドの x と y 座標を調整して表示しています。

/// ex31102.bpl

fmt 40 24

brl 40-10 24 #c4#aa4#b brl 7 23 ,fig4 ratio of completion time brl 9 22 for regular and low vision

a4

define range -0.12 1.4 -0.6 5 define eod 9999 0

pitch0 = 2

pitch1 = 2.2

```
pitch2 = 4
window range
dot 0
   xaxis 0\ 1\ 0.1\ 2 pitch0
   yaxis 0 3.8 1 2 pitch0
dot 1
     normal condition of the nondisabled
   line 0.10 1 0.90 1 pitch1
///
     pseudo low vision of the nondisabled
spline range 3 0 0 pitch1
    0.10\ 1.91
    0.20\ 1.86
    0.30\ 1.84
    0.40 1.84
    0.50\ 1.86
    0.60\ 1.88
    0.70\ 1.91
    0.80\ 1.96
    0.90\ 2.05
    eod
///
brl 0.90+dx(5) 1+dy(2) regular
brl 0.90+dx(5) 2.05+dy(2) low vision
///
   brl 0.10-dx(5) -dy(10) #j.a
   brl 0.50-dx(5) -dy(10) #j.e
```

brl 0.90-dx(5) -dy(10) #j.i

```
brl 1-dx(80) -dy(20) completion ratio
```

```
brl -dx(15) dy(2) #j
brl -dx(15) 1+dy(2) #a
brl -dx(15) 2+dy(2) #b
brl -dx(15) 3+dy(2) #c
brl -dx(15) 4+dy(2) ratio of completion time
```

///

ff

ex31102.bpl の印字結果

3.12 関数グラフの作成

関数のグラフを作成するために作図コマンドが 19 種類用意されています。放物線や双曲線等 2 次曲線が 4 種、多項式曲線が 5 種、指数対数関数が 5 種、三角関数が 3 種、その他統計関連の関数が 2 種計 19 種が作成可能です。

作図コマンドのパラメータを指定することによって任意のグラフを 1 コマンドで作成可能です。

一般に関数のグラフの作図コマンドはコマンド名に続けて関数が定義されている領域を 定義します。1番目と2番目のパラメータにより関数の定義域(x座標の範囲)を指定しま す。3番目と4番目のパラメータにより関数が定義されている値域(y座標の範囲)を指定 します。この定義域と値域を任意に指定することにより関数の必要な部分を作図すること が可能です。

関数の定義領域に続く 1 つまたは複数個のパラメータにより関数を定義する数式が表現されます。

後ろから 3 番目と 2 番目のパラメータ「px」と「py」によりグラフの位置が決定されます。「px」はグラフを x 軸方向に平行移動させます。また、「py」はグラフを y 軸方向に平行移動させます。

最後のパラメータ「pitch」は点線の点間隔をミリメートル単位で指定します。

現在この 19 種の関数のグラフと 3.11 節の自由曲線によるグラフにより任意の関数のグラフを作成することが出来ます。

今後、統計に関連するグラフを拡張していく計画です。

3.12.1 軸が y 軸に平行な放物線

軸がy軸に平行な放物線のグラフを任意に描くことが可能です。

<書式>

vparabola x0 x1 y0 y1 a px py pitch

<解説>

vparabola (vertical parabola)は、軸が y 軸に平行な放物線のグラフを任意に描く作図コマンドのコマンド名です。vparabola は 8 個のパラメータを必要とします。

「x0」と「x1」は放物線グラフの定義域を指定します。

「y0」と「y1」は放物線のグラフの値域を指定します。

「a」と「px」及び「py」により放物線を定義する数式が表現されます。

$y = a * (x - px)^2 + py$

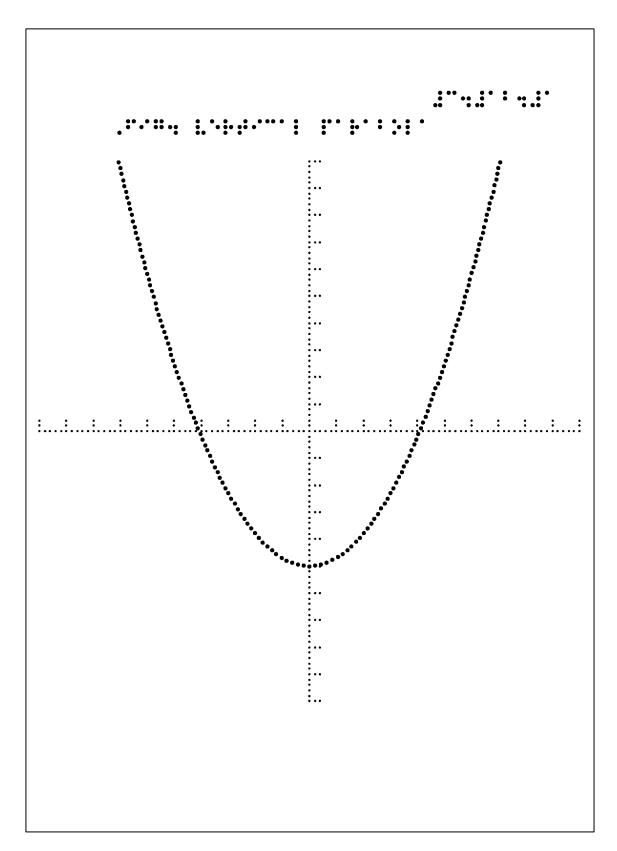
「a」が正であれば下に凸の放物線が、負の場合は上に凸の放物線が描かれます。

「px」と「py」は放物線を x 軸方向と y 軸方向に平行移動させるためのパラメータです。「pitch」は点線の点間隔です。

例 3.12.1 下に凸の放物線

制限することが出来ます。

```
//
   ex31201.bpl
fmt 40 24
  brl 40-10 24 #c4#ab4#a
  brl 7 23 ,fig4 vertical parabola
a4
define xrange -10 10
define yrange -10 10
a = 0.3
px = 0
py = -5
unit = 1
pic = 2
pitch0 = 2
pitch1 = 2.2
dot 0
  xaxis xrange unit pic pitch0
  yaxis yrange unit pic pitch0
dot 1
  vparabola xrange yrange a px py pitch1
ff
  下に凸の放物線の係数 a は 0.3 に設定しています。
  関数の定義域 x0~y1 の 4 つのパラメータの値を指定すれば放物線のグラフの描く範囲を
```



ex31201.bpl の印字結果

3.12.2 軸が x 軸に平行な放物線

<書式>

hparabola x0 x1 y0 y1 a px py pitch

<解説>

hparabola (horizontal parabola)は、軸が x 軸に平行な放物線のグラフを任意に描くことが可能です。hparabola は 8 個のパラメータを必要とします。

「x0」と「x1」は放物線グラフの定義域を指定します。

「y0」と「y1」は放物線のグラフの値域を指定します。

「a」と「px」及び「py」により放物線を定義する数式が表現されます。

$$y = \pm \sqrt{\frac{x - px}{a}} + py$$

「a」が正であれば左に凸の放物線が、負の場合は右に凸の放物線が描かれます。

「px」と「py」は放物線を x 軸方向と y 軸方向に平行移動させるためのパラメータです。「pitch」は点線の点間隔です。

例 3.12.2 左に凸の放物線

// ex31202.bpl

fmt 40 24

brl 40-10 24 #c4#ab4#b

brl 7 23 ,fig4 horizontal parabola

a4

define xrange -10 10

define yrange -10 10

a = 0.3

px = -5

py = 0

unit = 1

pic = 2

pitch0 = 2

pitch1 = 2.2

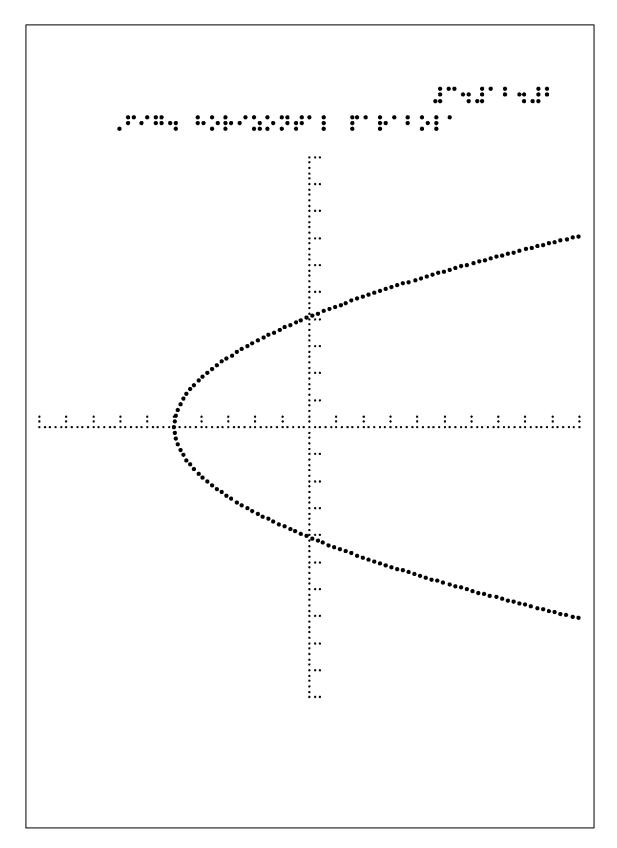
dot 0

xaxis xrange unit pic pitch0 yaxis yrange unit pic pitch0

dot 1

hparabola xrange yrange a px py pitch1

ff



ex31202.bpl の印字結果

3.12.3 双曲線

<書式>

hyperbola x0 x1 y0 y1 a b px py pitch

<解説>

hypebola は、双曲線を 1 コマンドで作図することが可能です。9 個のパラメータを必要とします。

「x0」と「x1」は双曲線の定義域を指定します。

「y0」と「y1」は値域を指定します。

 $\lceil a \rfloor$ と $\lceil b \rfloor$ 及び $\lceil px \rfloor$ と $\lceil py \rfloor$ は双曲線を定義する数式を表現します。

$$y = \pm \frac{b}{a} * \sqrt{(x - px)^2 - a^2} + py$$

「px」と「py」は双曲線を平行移動させるためのパラメータです。 「pitch」は点線の点間隔です。

// ex31203.bpl

fmt 40 24

brl 40-10 24 #c4#ab4#c brl 7 23 ,fig4 hyperbola

a4

define xrange -10 10

define yrange -10 10

a = 2

b = 2

px = 0

py = 0

unit = 1

pic = 2

pitch0 = 2

pitch1 = 2.2

dot 0

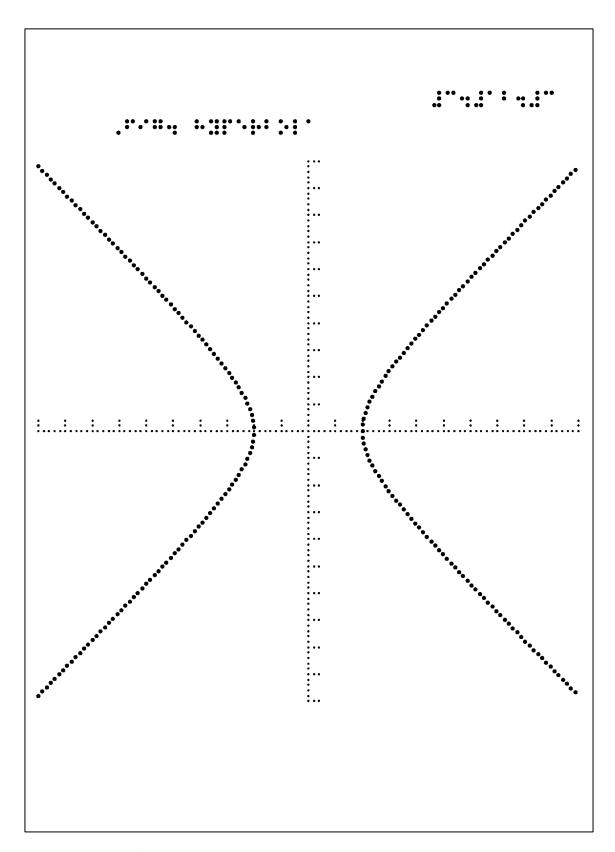
xaxis xrange unit pic pitch0 yaxis yrange unit pic pitch0

dot 1

hyperbola xrange yrange a b px py pitch1

ff

関数の定義域を制限すれば関数の任意の領域を作図することが可能です。例えば、第 1 象限と第 4 象限に制限するのであれば、x0 と x1 は 0 と 10 に設定して下さい。



ex31203.bpl の印字結果

3.12.4 直角双曲線

<書式>

rhypebola x0 x1 y0 y1 k px py pitch

<解説>

直角双曲線 rhyperbola (right-angled hypeboa)も 1 コマンドで作図可能です。パラメータは 7 つです。

「x0」と「x1」により定義域を指定して下さい。

「y0」と「y1」により値域をしてして下さい。

「k」は直角双曲線の計数です。

「k」と「px」及び「py」により、直角双曲線を定義する数式を表現します。

$$y = \frac{k}{x - px} + py$$

「px」と「py」をグラフを平行移動させるパラメータです。

「pitch」は点線の点間隔です。

例 3.12.4 直角双曲線

// ex31204.bpl

fmt 40 24

brl 40-10 24 #c4#ab4#d

brl 7 23 ,fig4 right-angled hyperbola

a4

define xrange -10 10

define yrange -10 10

k = 2

px = 0

py = 0

unit = 1

pic = 2

pitch0 = 2

pitch1 = 2.2

dot 0

xaxis xrange unit pic pitch0 yaxis yrange unit pic pitch0

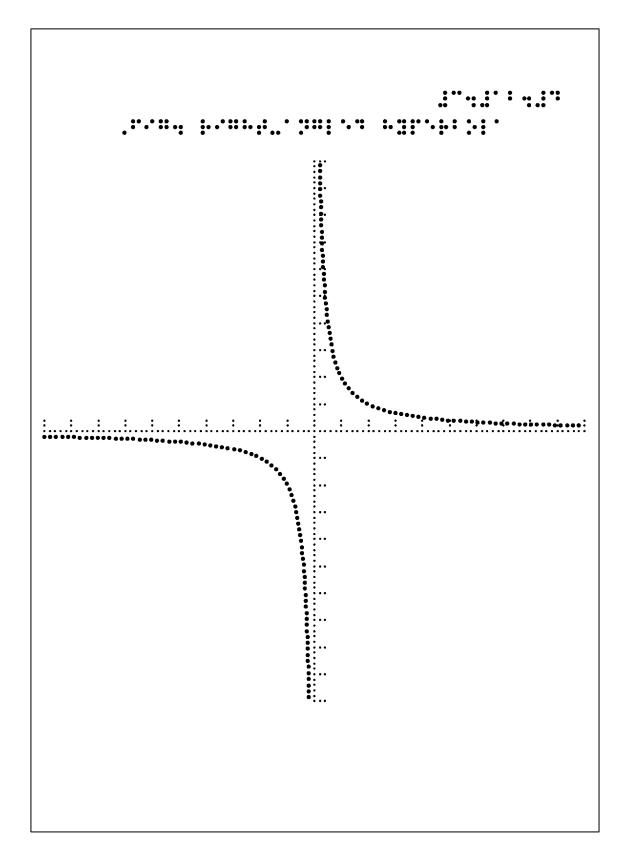
dot 1

rhyperbola xrange yrange k px py pitch1

ff

定義域を指定する 4 つのパラメータを適宜変えて目的のグラフを作成して下さい。例えば、第 1 象限または第 4 象限の直角双曲線だけを描きたい場合は定義域(x 座標の範囲)を 0 から正の範囲に制限して下さい。

また、「y0」と「y1」を制限して目的の値域の直角双曲線を描いて下さい。



ex31204.bpl の印字結果

3.12.5 多項式のグラフ

1 次から 5 次までの多項式のグラフを作成するコマンド(f1, f2, f3, f4, f5)が用意されています。

<書式>

f1 x0 x1 y0 y1 a b pitch

f2 x0 x1 y0 y1 a b c pitch

f3 x0 x1 y0 y1 a b c d pitch

f4 x0 x1 y0 y1 a b c d e pitch

f5 x0 x1 y0 y1 a b c d e f pitch

<解説>

この5つのコマンドは1次から5次までの多項式のグラフを作成するコマンドです。

「x0」と「x1」は描くグラフの定義域(x 座標の範囲)です。また、「y0」と「y1」はグラフの値域(y 座標の範囲)です。

「a」「b」 …… は多項式の係数です。

各多項式の係数と定義式は次のとおりです。

f1(x) = ax + b

 $f2(x) = ax^2 + bx + c$

 $f3(x) = ax^3 + bx^2 + cx + d$

 $f4(x) = ax^4 + bx^3 + cx^2 + dx + e$

 $f5(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$

(注意) 現バージョンの多項式のグラフには平行移動させる「px」と「py」と言う 2 つのパラメータは含まれていません。しかし、新しいバージョンでは他の関数のグラフと同様、平行移動させるパラメータを導入する計画です。

例 3.12.5 3 次の多項式のグラフ

多項式のグラフの作成例として3次の多項式のグラフを紹介します。

// ex31205.bpl

fmt 40 24

brl 40-10 24 #c4#ab4#e

brl 7 23 ,fig4 function of 3 degree

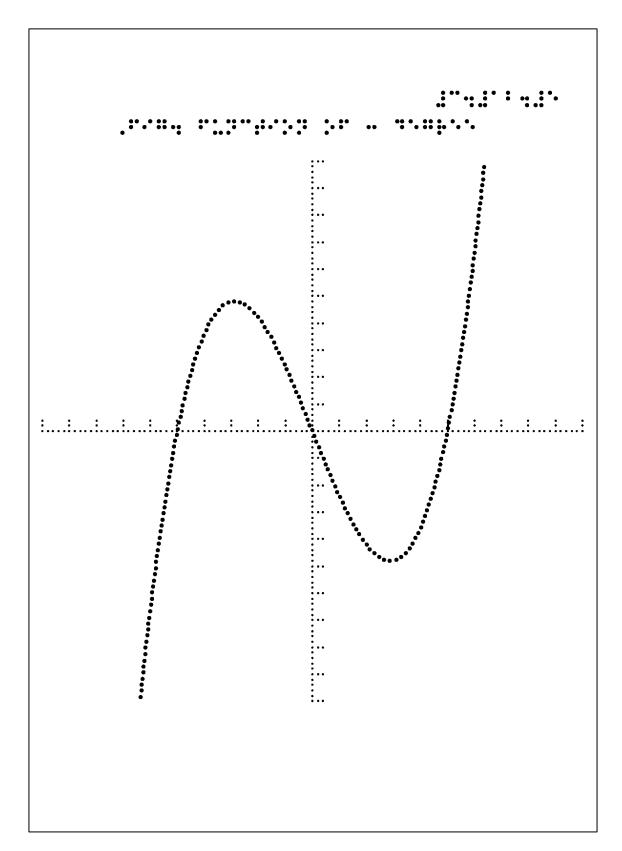
a4

define xrange -10 10

```
define yrange -10 10
a = 0.1
b = 0
c = -2.5
d = 0
unit = 1
pic = 2
pitch0 = 2
pitch1 = 2.2

dot 0
    xaxis xrange unit pic pitch0
    yaxis yrange unit pic pitch0

dot 1
    f3 xrange yrange a b c d pitch1
```



ex31205.bpl の印字結果

3.12.6 三角関数のグラフ

「サイン・コサイン・タンジェント、丸い頭に角が立つ」とデカンショ節で旧制中学や 旧制高校の学生に歌われた三角関数です。Bplotには sine、cosine、tangent の 3 つの三角 関数のグラフを作成するコマンドが用意されております。

<書式>

sin x0 x1 y0 y1 a b px py pitch cos x0 x1 y0 y1 a b px py pitch tan x0 x1 y0 y1 a b px py pitch

<解説>

サイン・コサイン・タンジェントの3つの三角関数の任意のグラフが各々1コマンドで作成可能です。コマンドの書式はコマンド名を除きパラメータは同様です。

「x0」と「x1」でグラフの定義域(x 座標の範囲)を指定します。同様に、「y0」と「y1」でグラフの値域(y 座標の範囲)を指定します。

「a」と「b」及び「px」と「py」によって三角関数の定義式を指定します。

$$\sin(x) = a * \sin(b(x - px)) + py$$

$$\cos(x) = a * \cos(b(x - px)) + py$$

$$\tan(x) = a * \tan(b(x - px)) + py$$

「px」と「py」はグラフを平行移動させるためのパラメータです。「px」はグラフを x 方向に平行移動させます。「py」はグラフを y 軸方向に平行移動させます。

「pitch」は点線の点間隔です。

例 3.12.6 サイン曲線とコサイン曲線

三角関数のグラフの例としてサイン曲線とコサイン曲線を 1 ページの上下に作図してみます。

// ex31206.bpl

fmt 40 24

brl 40-10 24 #c4#ab4#f

brl 7 23 ,fig4 sine and cosine curves

a4

pitch0 = 2.0

pitch1 = 2.2

```
radian
```

```
window -2.2*pi 2.2*pi -12.53 12.53
```

dot 0

yaxis -10 10 1 2 pitch0

/// delineate a sine curve

origin 0 5 1 1

brl -1.5*pi 3.5 sine curve

 $\mathrm{dot}\ 0$

xaxis -2.2*pi 2.2*pi pi/4 2 pitch0

dot 1

sin -2*pi 2*pi -5 5 2.5 1 0 0 pitch1

/// delineate a cosine curve

origin 0 -10 1 1

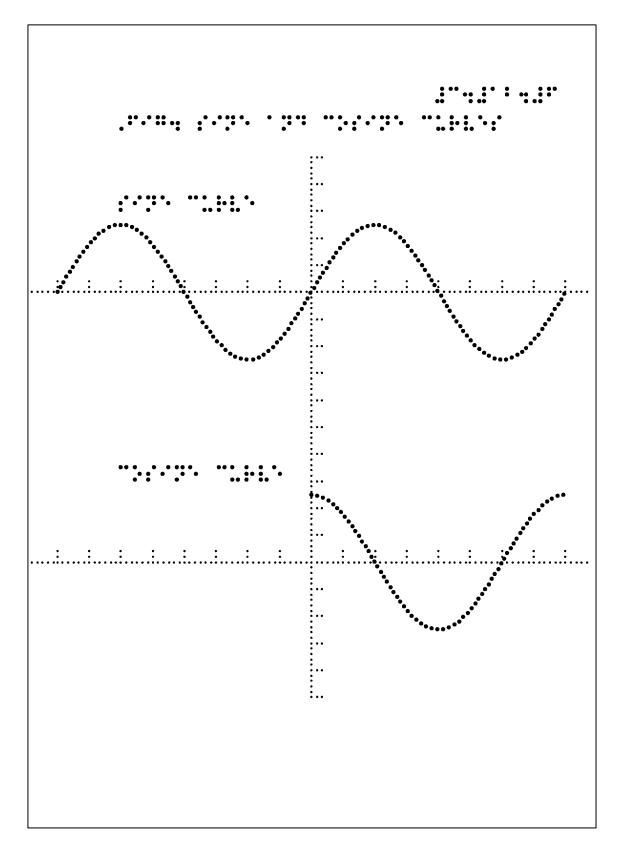
brl -1.5*pi 3.5 cosine curve

dot (

xaxis -2.2*pi 2.2*pi pi/4 2 pitch0

dot 1

 \cos 0 2*pi -5 5 2.5 1 0 0 pitch
1



ex31206.bpl の印字結果

三角関数のグラフを作成するには角度の単位は度よりも弧度の方が適しているため radian コマンドを使用しています。弧度法の単位円周率 π は定数 pi として Bplot で定義されています。

ウィンドウも x 座標の単位は弧度法の方が適しているため pi を使って window コマンドで定義しています。x 座標の範囲は+-2*pi よりもやや広い、 $-2.2*pi\sim2.2*pi$ に設定しています。一方、y 座標の範囲はデフォルトの $-12.53\sim12.53$ に設定しています。

まず、yaxis コマンドにより y 軸を描いています。

次にサイン曲線とコサイン曲線を用紙の上下に配置するため origin コマンドを使用しています。サイン曲線を用紙の上部に描くため origin 0.511 コマンドにより原点を (0,5) に移動させています。

その後、x 座標軸とサイン曲線を描いています。サイン曲線は-2*pi から 2*pi まで 2 波長 分描いています。

次にコサイン曲線を用紙の下部に描くため origin 0-10 1 1 コマンドにより原点を下方 10cm に移動させています。

その後、x 座標軸とコサイン曲線を描いています。コサイン曲線の範囲は $0\sim2*pi$ に設定して 1 波長分描いています。

このように三角関数の定義域を変えることにより任意のグラフを描くことが可能です。

3.12.7 指数対数のグラフ

指数対数の関数のグラフが 4 種類作成可能です。

pow (power) a^x

exp (exponential) 指数

110 (regular logarithm) 常用対数

ln (natural logarithm) 自然対数

<書式>

pow x0 x1 y0 y1 k a b px py pitch exp x0 x1 y0 y1 a b px py pitch l10 x0 x1 y0 y1 a b px py pitch ln x0 x1 y0 y1 a b px py pitch

<解説>

4種類の指数対数関数のグラフが作成可能です。パラメータの設定方法は関数のグラフのコマンドとほぼ共通です。

最初の4つのパラメータはグラフの定義領域を設定します。「x0」と「x1」はグラフの定義域(x 座標の範囲)です。「y0」と「y1」はグラフの値域(y 座標の範囲)です。

5番目のパラメータから2ないし3個のパラメータはそれに続くパラメータ「px」と「pv」

により関数の定義式が設定されます。

$$pow(k, x) = a * k^{b(x-px)} + py$$

$$exp(x) = a * exp(b(x - px)) + py$$

$$l10(x) = a * log_{10} b(x - px) + py$$

$$ln(x) = a * log_e b(x - px) + py$$

「px」と「py」はグラフを平行移動させるパラメータです。「px」はx 方向に、「py」はy 方向に平行移動させます。

「pitch」は点線の点間隔です。

例 3.12.7 指数と対数曲線

指数関数 exp と対数関数 ln のグラフを作成してみます。

最初にx軸とy軸を描きます。

次に指数関数と自然対数のグラフを作成します。対数関数は定義域 (x) 座標の範囲)は正ですので (x) 座標の範囲は (x) の(x) に設定しています。

// ex31207.bpl

fmt 40 24

brl 40-10 24 #c4#ab4#g brl 7 23 ,fig4 exp and log curves

a4

define xrange -10 10

define yrange -10 10

px = 0

py = 0

unit = 1

pic = 2

pitch0 = 2

pitch1 = 2.2

a = 1

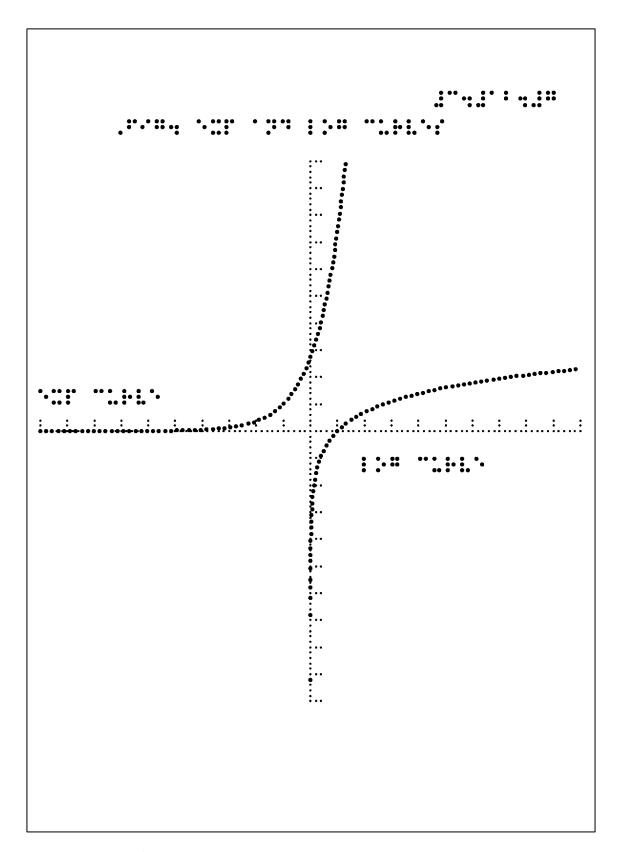
b = 1

dot 0

xaxis xrange unit pic pitch0

yaxis yrange unit pic pitch0

brl -10 1.5 exp curve brl 2 -1 log curve



ex31207.bpl の印字結果

3.13 統計グラフ

統計的分布に関するグラフの作図コマンドが、現在次の3種類用意されています。今後、 更に充実させていく計画です。

詳細は第4章で解説します。

bwplot ボックスアンドウィスカープロット

weibull ワイブル分布関数

normal 正規分布確率密度関数

3.13.1 箱ひげ図

箱ひげ図はデータの分布を表すグラフです。探索的データ解析法として開発されたグラフです。

箱ひげ図ではデータの分布を 5 つの代表値で要約します。データを小さい順に並べて中央の値が「中央値」です。異常に小さいデータとか異常に大きいデータとかを異常値としてデータ分布から外した後、最小値を「下隣接値」、最大値を「上隣接値」と呼びます。

「下隣接値」と「中央値」の間の中央値を「下ヒンジ」と呼びます。同様に、「中央値」と「上隣接値」の間の中央値を「上ヒンジ」と呼びます。

この「下隣接値」「下ヒンジ」「中央値」「上ヒンジ」「上隣接値」の五数要約値によって 分布を表します。

後述の例 3.13.1 の箱ひげ図に見るように長方形の箱の両側から左右にひげが伸びた形を しています。

箱の左端の位置が「下ヒンジ」、箱の右端の位置が「上ヒンジ」になるように箱を描きます。箱の中に「中央値」の位置に線を引きます。箱の左端から「下隣接値」まで線を引いてひげを書きます。同様に、箱の右端の位置から「上隣接値」まで線を引いてひげを書きます。

箱ひげ図によってデータの分布の概要を把握することが出来ます。

<書式>

bwplot x1 x2 x3 x4 x5 hight width pic pitch

<解説>

bwplot (Box and Wisker plot)は箱ひげ図を作成するコマンドです。パラメータは9つです。

箱ひげ図は縦書きにも横書きにも作成できます。通常複数の箱ひげ図を並べて比較する ため箱ひげ図を横書きにし、上から下に並べて比較したほうが触読しやすいようです。例 3.13.1 では横書きにしています。横書きで解説します。

x1、x2、y3、x4、x5 はデータの「下隣接値」「下ヒンジ」「中央値」「上ヒンジ」「上隣接値」の五数要約値です。

hight は箱ひげ図を描くウィンドウ上の高さです。

width は箱の幅です。横書きの場合は縦の幅です。

pic はひげの先端を示す線の長さです。正確にはひげの先端から線の上下の端までの長さです。

pitch は点線の点間隔です。

例 3.13.1 国語の得点の箱ひげ図

国語のテストの得点分布から箱ひげ図を作成してみます。通常の問題と音声問題の得点を比較する実験です。

被験者群は、点字使用者群(bl: braille user group)・弱視者群(lv: low vision group)・健常者群(nd: nondisabled group)の3群です。

テスト問題は、肉声の音声問題(ntr: natural voice)・合成音声の音声問題(syn: synthesized voice)・点字問題(brl: braille)・文字と音声のマルチモーダル問題(mul: multimodal)・通常問題(prt: print)・文書構造表音声問題(dia: document diagram)です。

得点は 100 点満点です。window コマンドによりウィンドウの x 座標の範囲を $0\sim105$ に拡大しています。

詳細は第4章で解説します。

/// ex31301.bpl

fmt 40 24

brl 40-10 24 #c4#ac4#a

brl 7 23 ,fig4 ,box and ,wisker ,plot

a4

window 0 108 -10.53 14.53

/// print a front side of sheet

dot 1

pitch = 2

width = 0.8

pic = 0.5

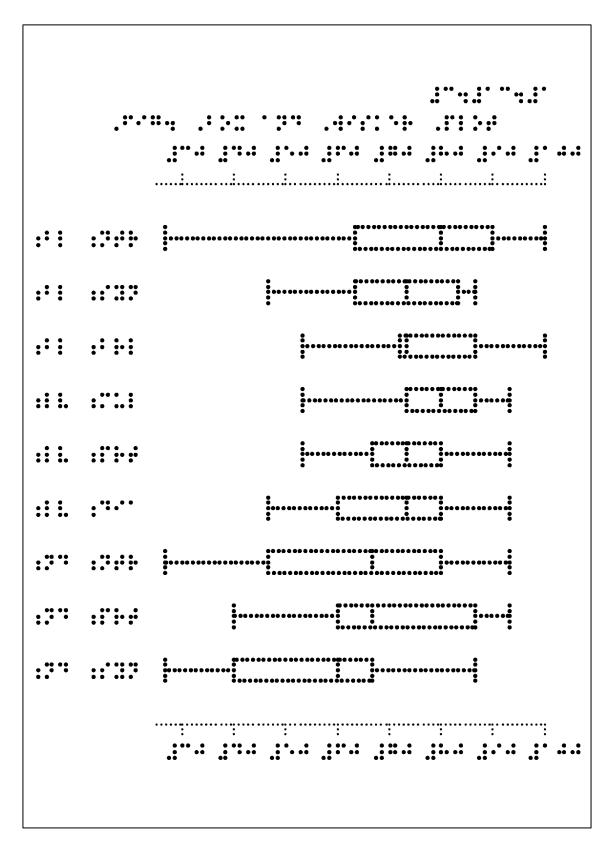
x0 = 1

hight = 9

```
kana x0 hight+dy(2) bl ntr
   bwplot 26.67 63.335 80 90 100 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) bl syn
   bwplot 46.67 63.335 73.33 83.335 86.67 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) bl brl
   bwplot 53.33 72.00 73.33 86.67 100 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) lv mul
   bwplot 53.33 73.33 80 86.67 93.33 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) lv prt
   bwplot 53.33 66.67 73.33 80 93.33 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) lv dia
   bwplot 46.67 60 73.33 80 93.33 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) nd ntr
   bwplot 26.67 46.67 66.67 80 93.33 hight width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) nd prt
   bwplot 40 60 66.67 86.67 93.33 -5 width pic pitch
hight = hight - 2
   kana x0 hight+dy(2) nd syn
   bwplot 26.67 40 60 66.67 86.67 hight width pic pitch
origin 0 11 1 1
dot 0
pitch = 1.8
   xaxis 25 100 10 2 pitch
y2 = dy(13.5)
   kana 30-dx(5) y2 30
   kana 40-dx(5) y2 40
   kana 50-dx(5) y2 50
```

kana 60-dx(5) y2 60

```
kana 70-dx(5) y2 70
   kana 80-dx(5) y2 80
   kana 90-dx(5) y2 90
   kana 100-dx(5) y2 100
origin 0 -20 1 1
   xaxis 25 100 10 -2 pitch
y3 = -dy(8)
   kana 30-dx(5) y3 30
   kana 40-dx(5) y3 40
   kana 50-dx(5) y3 50
   kana 60-dx(5) y3 60
   kana 70-dx(5) y3 70
   kana 80-dx(5) y3 80
   kana 90-dx(5) y3 90
   kana 100-dx(5) y3 100
/// print a back side of sheet
mirror 13
y0 = dy(5)
y1 = 20 - dy(5)
dot 2
pitch = 5
   line 30 y1 30 y0 pitch
   line 40 y0 40 y1 pitch
   line 50 y1 50 y0 pitch
   line 60 \text{ y} 0 \text{ } 60 \text{ y} 1 pitch
   line 70 \text{ y} 1 \text{ } 70 \text{ y} 0 pitch
   line 80 \text{ y} 0 \text{ } 80 \text{ y} 1 pitch
   line 90 y1 90 y0 pitch
   line 100 y0 100 y1 pitch
mirror 0 0
```



ex31301.bpl の印字結果

箱ひげ図から点字被験者群は、点字問題と肉声または合成音声の音声問題の得点分布間に差がないことが認められます。統計的にも有意差は認められませんでした。

3.14 貼り付けと回転コマンド

実験装置の模式図や地図や文字の字形等、図を構成する作図部品を組み合わせて 1 枚の図を作成するために個々の作図部品を貼り付けたり、回転させたり、回転して貼り付けたりするコマンドが用意されています。

paste 図に作図部品を貼り付けるコマンド

rotate 作図部品を回転させるコマンド

rpaste (rotated paste) 作図部品を回転して貼り付けるコマンド

3.14.1 paste 貼り付けコマンド

模式図を構成するビーカーや試験管等作図部品を貼り付けて 1 枚の図を作成することが可能です。また、地図や文字の字形の画像データを画像データベースから取り出して図に貼り付けることが可能です。

<書式>

paste filename x0 y0 bias_x bias_y

<解説>

貼り付けコマンド paste は作図部品を貼り付け先の図の任意の位置に任意の大きさで貼り付けることが可能です。作図部品はあらかじめ Bplot で作成しておきます。パラメータは5つです。

最初のパラメータ filename は作図部品の入ったファイル名です。

次の2つのパラメータx0とy0 は貼り付け先の座標 (x0,y0)を指定します。作図部品の原点が貼り付け先の座標 (x0,y0) になるように貼り付けられます。

次の2つのパラメータは作図部品の拡大縮小率を指定します。 $bias_x$ は作図部品のx軸 方向の拡大縮小率です。 $bias_y$ はy軸方向の拡大縮小率です。

例 3.14.1 UDLTE のロゴマーク

Bplotを開発しているテストと学習環境のユニバーサルデザイン研究機構 UDLTE のロゴマークを作成してみます。

UDLTE のロゴマークは、例 3.11.1 の曲線で描かれた星形の中にアルファベット大文字の字形 UDL が貼り付けられています。また、星形の左に T と E が大きく描かれています。

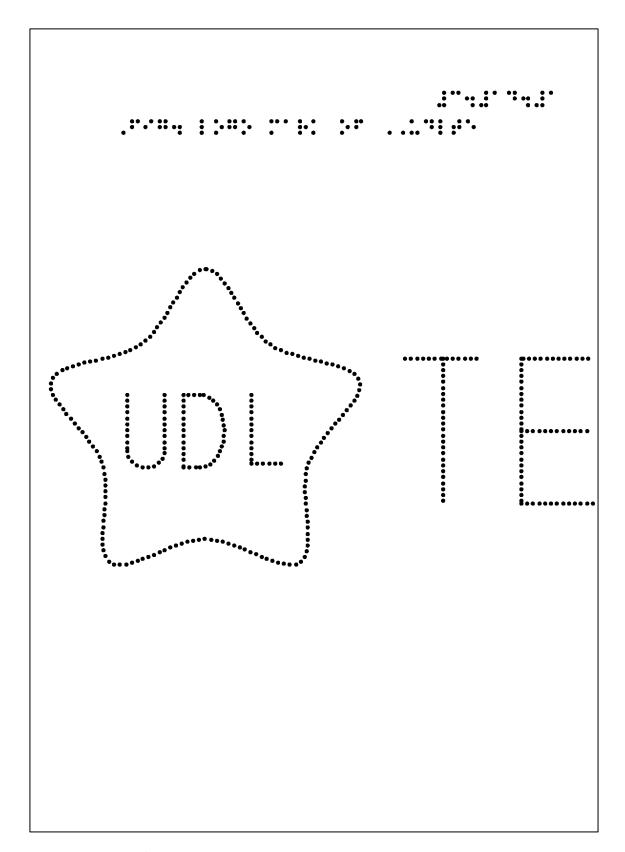
// ex31401.bpl

fmt 40 24

```
brl 40-10 24 #c4#ad4#a
   brl 7 23 ,fig4 logo mark of "udlte
a4
1 = 6
m = 4
r = 360/10
pitch = 2.2
x0 = 1*\cos(90+0*r)
y0 = 1*\sin(90+0*r)
x1 = m*\cos(90+1*r)
y1 = m*\sin(90+1*r)
x2 = 1*\cos(90+2*r)
y2 = 1*\sin(90+2*r)
x3 = m*\cos(90+3*r)
y3 = m*\sin(90+3*r)
x4 = 1*\cos(90+4*r)
y4 = l*\sin(90+4*r)
x5 = m*\cos(90+5*r)
y5 = m*\sin(90+5*r)
x6 = 1*\cos(90+6*r)
y6 = 1*\sin(90+6*r)
x7 = m*\cos(90+7*r)
y7 = m*\sin(90+7*r)
x8 = 1*\cos(90+8*r)
y8 = 1*\sin(90+8*r)
x9 = m*\cos(90+9*r)
y9 = m*\sin(90+9*r)
x10 = l*cos(90+10*r)
y10 = l*sin(90+10*r)
define range -10 10 -12 12
define EndofData 9999 0
```

func = 4

```
origin -4 0 1 1
dot 1
spline range func 0\ 0 pitch
   x0 y0
   x1 y1
   x2\;y2
   x3 y3
   x4 y4
   x5 y5
   x6 y6
   x7 y7
   x8 y8
   x9 y9
   EndofData
///
w = 2.2
h = 0
s1 = 0.1
t1 = 0.15
s2 = 0.25
t2 = 0.25
   paste upper\Psiu.bpl -w h s1 t1
   paste upper\$d.bpl 0 h s1 t1
   paste upper¥l.bpl w h s1 t1
   paste upper¥t.bpl 4*w h s2 t2
   paste upper¥e.txt 6*w h s2 t2
```



ex31401.bpl の印字結果

3.15 点の二重打ち等を防ぐコマンド

出来る限り図を見やすくしたり、後から修正したりするため次のコマンドが用意されています。各コマンドは図形の各点をプリンタ用紙に印字する前に一旦その点の座標をメモリバッファ上に記録し、メモリバッファを修正してから印字することにより実現しています。

gap は点の二重打ちを防ぐコマンドです。メモリバッファに打つ点の座標を記録するときにその座標の近傍に打たれた点の記録があるか無いかを調べ記録があればその点の記録を消してから新しい点を記録していきます。または、打つ点の近傍に打たれた点の記録があれば新しい点を打たないようにします。

scrape は指定された点を中心に円形の領域の点を消去します。 erase は指定された点を中心に長方形の領域を消去します。 詳細は第4章で解説します。