§ 運動方程式の数値的解法

目的 運動方程式の数値的な解法について理解を深める。

1 概要

運動方程式を解析的に解くことが困難な場合でも,現代ではコンピューターを用いて解を 数値として得ることができる。一般式は得られないが,必要な精度で時間,座標,速度を求 めることが可能である。このテーマではC言語を用いたプログラミングによって,運動方程 式の数値的な解法を試みよう。以下で取り扱う例題は,解析的に解けるものも含まれている が,数値的な解法によって容易に解の概略が分かる。

テーマの後半ではコンピューターの果たしている他の役割として "制御 "を取りあげる。 注

- このテーマでは予習の提出はありません。事前にテキストを読んできてください。
- USB メモリーやフロッピー等の記録媒体を持参してきてください。
- 実験終了時に課題を提出すること。このテーマでは,レポート提出は不要です。
- 2 C++ BuilderX を用いる
- 2.1 はじめに

ここで使用する言語は Borland 社 (http://www.borland.co.jp) の C++BuilderX Personal を用います。これは C/C++言語を 用いたプログラミング環境を提供するソフトウエアで, 統合開 発環境 (IDE) と呼ばれるものです。はじめに, デスクトップ上 にある C++BuilderX のアイコンをダブルクリックして起動し ます (図1)。



図 1: アイコン

初めての起動であっても,何らかのファ イルが読み込まれて起動します。2回目以 降は前回作成したプログラムが読み込まれ て起動します。メインメニューの[ファイ ル][プロジェクトを閉じる]で閉じてしま います(図2)。



図 2: プロジェクトを閉じる

come/Wel	come.cb	×	

図 3: 閉じるプロジェクトを選択する

この後,図3のような確認画面が現れますので,チェッックして全部消してしまいます。こうして,図4のような何のファイルも開かれていない画面になります。



図 4: 何も開かれていない画面

2.2 プロジェクトを開始する

メインメニューから [ファイル][新規]を選択すると図5の窓が表示されます。

ロジェクト ソースファイル	Web	
_		
		<u>())</u>
新規コンソール 新規 DLL	GUI	新規ライブラリ
R	アフリケーションの新想	e të at
2		
新規インボートライブラリ	cvs	
	からプロジェクトを取得	からプロジェクトを取得する
1923	A	
Classification Pre-7/1	いずたからプロペi・クト	↓[]↓ マイクフライルからプロジェクト
のプロジェクト	57185545151	X499744W0094949491
		
1		
プロジェクトグループ		

図 5: 新規コンソール

図5は作成するプログラムの種類を選ぶ画面で,ここでは新規コンソールを選びます。OK をクリックすると図6が表示されます。

· ···	コンソールアプリ ・ コンソールアプリク - プロジェクト設定	ケーション名と場所の運択 ーション用に名前とディレクトリを設定します。
class Shape	名前:	first
shapetype stype; public:	ディレクトリ(四):	ents and Settings/Administrator/cbproject/first 💌 🛄
	U709±9F8	#黒中のフロシェクトクルーフに通知 ↓

図 6: プロジェクトの作成

この図にある"プロジェクト"とはプログラム作成の作業全体を意味します。図6 でプロ ジェクトに適当な名前を付けると,ホームディレクトリーの中にフォルダーが作成されます。 ここでは Administrator というユーザーで,"first"というプロジェクトですが,この場合下 記のフォルダーが作成されます(それぞれのユーザー名に置き換えること)。

C:\Documents and Settings\Administrator\cbproject\first

図6で[次へ]をクリックし、図7のプラットフォームの選択画面を表示します。



図 7: プラットフォームの選択

Windows 用のソフトを作成するので初期設定のまま次へをクリックすると,図8の画面になります。

プロジェク	フトに追加する	ファイルを指定します。	
919-	・ドファイル:	名前 untitled	<u>説明</u> main()のある C++ ソ
-12#-	トされたファイ	(ル: ファイル名	
			追加 削號

図 8: default のソースを選ぶ

untitled にチェックを入れて,終了をクリックすると,first というプロジェクトの初期設 定が完了します(図9)。

C++BuilderX - C:/Documents and Settings/Administrator.	cbproject/first/	first.cbx											×
ファイル(E) 編集(E) 検索(S) 表示(V) ブロジェジト(P)	実行(R) チーム	(<u>A</u>) ウィザー	F(Z)	シール(D	ウィンド	500 AN7(H)							
	A		-	h in d		A Windows	•	👷 Debug Build 🔹	- 4 💀	6 -	盘 - 中	+	
📳 📴 🎒 🗒 first.cbx 🔹													
I first.cbx													
first.exe													
p united.cpp													
クラスブラウザ情報を更新しています													

図 9: 初期設定の終了

2.3 プログラムの作成

図 10 に示す untitled.cpp という文字をダブルク リックすると図 11 の画面が表示されます。以下で この図にあらわれている領域の(四角い枠の)名 称をあげます。

b		8
📑 firs	t.cbx	
10	first.exe	
	untitled.cp	р
	I	à

⊠ 10: untitled.cpp



図 11: 領域の名称

最初に簡単な C 言語のプロ グラムを作成するため,一端 ⑦の内容ペインのテキストを すべて削除します。その後,図 12のように記述して,ツール バーから[すべて保存]をク リックします(以下,必要に 応じて保存すること)。



🗷 12: Hello World

このプログラムは画面上に"Hello World"と表示して,改行するプログラムです。以下簡 単に説明すると,5行目の関数 printf が画面に描画する働きをします。このうち\n は改行 を表します。この printf を使用するために1行目で,#include <stdio.h>と記述します。 stdio.h は標準的な入出力を定義したパッケージで,その中で printf も定義されています。 このように,C では使用する機能を含んだパッケージを,プログラムの冒頭で読み込みます。 5,6行目に示すように,セミコロン(;)を用いて文の終わりを示します。

2.4 プログラムのコンパイルと実行

先に入力したプログラムをコンパイラを使用し て実行可能プログラムにします(コンパイルする という)。メインメニューの[プロジェクト][プ ロジェクト"first.cbx"のメイク]を実行すると(図 13), window 下部にメッセージペインという窓 が開きます(図14)。ここにはエラーなどのメッ セージが表示されます。エラーメッセージが有れ ば,プログラムを修正します。



図 13: プログラムのコンパイル



図 14: メッセージペイン

コンパイルが終了したら,ツールバーの [プロジェクトの実行]をクリックします (図 15)。図 16 のようにメッセージペイン に"Hello World"と表示されたらプロジェク トの完了です。

10101 -	Windows	•	Debug Build	•	E
					プロジェクトの実行

図 15:	実行
-------	----

C:\Doc	uments	and S	ettings\A	dministrat.	or\cbpr	oject∖fi	.rst\win	dows\Deb	ug_Build	d\first.e	xe
Hello	World										
(333333											
	終了口	- F 0	でプロセスは	終了しました							

図 16: 終了コード 0

エクスプローラーで作業用のフォルダーをみると図 17 のように"first.exe"が作成されてい るのが確認できます。



 \boxtimes 17: first.exe

	-	
c	- 1	
	۰.	

3 C 言語の基本的なルールとグラフ作成の方法

3.1 幾つかの約束事

最初に簡単な計算を試みます。最初,以下のように内容ペインに記述します。

```
#include <stdio.h>
#include <math.h>
double x,y;
int main()
{
    x=2;
    y=sqrt(x);
    printf("%f のルートは %f です。 \n",x,y);
    return (0);
}
```

1行目の#include <stdio.h>は標準的な入出力関数が定義されているパッケージです。

表 1: 定義されている関数

sin(x)	:	x の 正弦	log(x)	:	自然対数
cos(x)	:	x の 余弦	log10(x)	:	常用対数
tan(x)	:	x の 正接	sqrt(x)	:	平方根
exp(x)	:	指数関数	fabs(x)	:	絶対値

#include <math.h>はルート などの数学関数が定義されている パッケージです。定義されている 関数には他に表1のようなものが あります。

プログラム中の double x,y; は変数 x と y が倍精度の実数型

であることを宣言しています。このように C 言語では実数と整数を区別して取り扱い,また 変数を使用する前に変数の型を宣言する必要があります。変数の型には表2のようなものが あります。

表 2: 変数の型

int : 整数型 float : 単精度実数型	double : 倍精度実数型
--------------------------	-----------------

int main() 以降がプログラムの本体です。 x=2 は代入文といわれ, 変数 x に 2 を与え ます。次行の y=sqrt(x) も代入文ですが, この場合は x のルートを計算した値を y に代入し ます。次行の printf 文は画面に計算結果を表示します。printf の働きをみるために, ここ でプログラムをコンパイルして実行すると, メッセージペインに下記のように表示されます。

2.000000 のルートは 1.414214 です。

printf 文中にある最初の%f は x を倍精度の実数として表示せよという指示をあらわして います。整数を表示するときには%d を用います。最後の行の return(0) は,オペレーティ ングシステムに終了のメッセージを伝えるものです。

3.2 繰り返しと分岐

コンピューターは繰り返し計算が得意です。ここでは,単純な計算を繰り返し実行してみ ます。まず,割り算のあまりを求めるプログラムを書いてみます。以下のように内容ペイン に記入します。これは,17を3で割った場合のあまりを求めるものです。

```
#include <stdio.h>
#include <math.h>
int a,b,c;
int main()
{
    a=17;
    b=3;
    c = a % b;
        printf("a = %d, b = %d, c = %d \n",a,b,c);
return (0);
}
```

% はあまりを求める演算子です。変数 c にあまりが代入されます。コンパイルして,実行すると,以下のように表示されます。

a = 17, b = 3, c = 2

次に,101から120までの数を,3で割って,そのあまりを求めてみます。同じ計算を繰 り返し実行するために,ここではforループを用います。forループの形式は,下記のよう になっています。{}に囲まれた命令を,終了条件が満たされるまで繰り返し実行します。

for(初期文; 終了条件; 繰り返し文) {実行する命令文}

for ループの働きを確認するために,以下の内容を内容ペインに記入します。これは数値 を繰り返し画面に書くプログラムです。

```
#include <stdio.h>
#include <math.h>
int i;
int main()
{
for(i=101; i<=120; i=i+1){
    printf("i = %d \n",i);
}
return (0);
}</pre>
```

コンパイルして実行すると,メッセージペインに101から120までの数が描かれます。プログラム中のiは制御用の変数です。i=101から始まって1ずつ増加し,120になるまでループ内の命令が繰り返されます。i=i+1によって,iは一回ごとに1増加します。

次に,あまりを求める演算を繰り返し実行します。内容ペインに以下のように記入します。

```
#include <stdio.h>
#include <math.h>
int i,a;
int main()
{
for(i=101; i<=120; i=i+1){
    a= i % 3;
    printf("i = %d あまり = %d \n",i,a);
}
return (0);
}</pre>
```

コンパイルして実行すると,メッセージペインに下記のような結果が表れます。

```
i = 101 あまり = 2
i = 102 あまり = 0
i = 103 あまり = 1
i = 104 あまり = 2
(以下,省略)
```

続いて,上記の演算の中で,ちょうど3で割り切れる場合を抜き出しましょう。このため ここでは if 文を利用します。if 文の形式は下記のようになっています。

if(条件文)(実行する命令文);

if 文は,()の中の条件が成り立つならば,その後に続く()の中の命令を実行します。 このif 文の機能を利用して,3の倍数だけを抜き出します。内容ペインに下記のように記入 します。

```
#include <stdio.h>
#include <math.h>
int i,a;
int main()
{
for(i=101; i<=120; i=i+1){
    a= i % 3;
    if(a == 0)(printf("i = %d あまり = %d \n",i,a));
}
return (0);
}</pre>
```

```
11
```

条件文が a == 0 となっていることに,注意してください。代入文とは異なって, = を 2 個用いています。コンパイルして実行すると,下記のような結果がメッセージペインに表れます。

 i = 102
 あまり = 0

 i = 105
 あまり = 0

 i = 108
 あまり = 0

 i = 111
 あまり = 0

 i = 114
 あまり = 0

 i = 117
 あまり = 0

 i = 120
 あまり = 0

このように, if 文や, for 文を用いることによって, プログラムの動きを制御することが出来ます。

3.3 課題1

以下の働きをするプログラムを作成しなさい。それぞれ,プログラムリストを印刷して提 出してください。

- a. 1 から 100 までの整数の和を求めるプログラム。 ヒント:和を保存するための変数を用意する。
- b. 1 から 100 までの奇数の和を求めるプログラム。
- c. 28 の約数を求め表示するプログラム。 注意:28 自身は約数に含めません。
- d. 28 の約数の和を求めるプログラム。 また,496 の約数の和を求めるプログラム(プログラムリストは何れかを提出)。

3.4 配列を用いる

運動方程式を数値的に解く場合には,下記のような一定時間ごとの座標値を求めることに なります。

 $x(t_0), x(t_1), x(t_2), x(t_3) \cdots$

このような一連のデータを取り扱う場合には,配列という,変数に添え字を付けたものを 利用します。10個の配列であれば,下記のように宣言します。

double x[10];

上記で,要素が10個の配列を宣言していますが,注意すべきことは,添え字が0から始まる点です。そのためx[0]からx[9]までが利用できます。x[10]は,この宣言文では定義されていません。

配列を使用する例題として,三角関数の計算を繰り返し実行してみます。ここでは繰り返しの制御文として for ループを用います。下記のように内容ペインに記入してください。

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159
double x[10],t[10];
int main()
{
   for(int i=0; i<=9; i+=1){
       t[i]=2.0*PI*(i/9.0);
       x[i]=sin(t[i]);
       printf("i= %d , x = %f \n",i,x[i]);
   }
   return (0);
}</pre>
```

#define PI 3.14159 は PI を 3.14159 で置き換えるマクロです。コンパイルして実行す るとメッセージペインに下記のように表示されます。

i= 0 , x = 0.00000 i= 1 , x = 0.642787 i= 2 , x = 0.984808 i= 3 , x = 0.866026 i= 4 , x = 0.342022 i= 5 , x = -0.342017 i= 6 , x = -0.866024 i= 7 , x = -0.984808 i= 8 , x = -0.642791 i= 9 , x = -0.000005

3.5 グラフを書く

数値だけでは,計算が上手く行ったのか分かりませんので,結果をグラフで表してみましょう。C言語にはグラフ描画用の命令がないので,ここでは"GLIBW32 ver1.36"というライブ ラリーを利用します(作成 Yamada. K 氏, http://www.asahi-net.or.jp/~uc3k-ymd/)。こ れをプログラムの冒頭で読み込みます。内容ペインに下記のように記入してください。

```
#include <stdio.h>
#include <math.h>
#include "glibw32.h"
#define PI 3.14159
double x[10],t[10];
int main()
  for(int i=0; i<=9; i+=1){
    t[i]=2.0*PI*(i/9.0);</pre>
        x[i]=sin(t[i]);
        printf("i= \%d , x = \%f \n",i,x[i]);
ginit(450,350,WHITE);
GRAPH g;
char strl[]="Sin 関数";
textout(10,5,strl,BLACK,1,1);
g.window(0,-1.2,2*PI,1.2);
g.view(20,20,420,320);
g.axis(0.2,0.1);
g.setcolor(RED):
for( int i = 0; i<=9; i+=1){
    g.fcircle(t[i],x[i],0.04);</pre>
}
savebmp("myfile.bmp"); // gend()の直前に書くこと!
gend();
return (0);
}
```

ginit(450,350,WHITE)以降がグラフを書く部分です。以下簡単に説明をします。

ginit(450,350,WHITE);	サイズ 450x350 ピクセルのグラフ窓を用意
GRAPH g;	GRAPH オブジェクトのインスタンス化
char strl[]="Sin 関数"; textout(10,5,strl,BLACK,1,1);	窓に文字を書く
g.window(0,-1.2,2*PI,1.2);	イメージする論理座標を決める
g.view(20,20,420,320);	(左下隅) 描画するスクリーン座標の範囲を決める (左上隅と右下隅の値)

g.axis(0.2,0.1);	縦軸,横軸の座標軸の目盛りを設定
g.setcolor(RED);	描画の色を変更
<pre>for(int i = 0; i<=9; i+=1){ g.fcircle(t[i],x[i],0.04); }</pre>	円を描いて塗りつぶす。0.04 は円の半径
<pre>savebmp("myfile.bmp");</pre>	グラフをファイルとして保存
gend();	描画終了

上記で,描画するグラフのサイズを決定している数値の意味を,具体的に図示したものが, 図 18 です。ginit(450,350,WHITE)でグラフ用の window のサイズを決定します。この数 値はピクセル(ドット)を基準にしています。g.view(20,20,420,320)は用意した window のうちで,座標(20,20)と(420,320)で指定される長方形の内側をグラフの描画範囲とするこ とを示します。ただし,この場合,座標(0,0)は window の左上隅にあり,縦軸の座標値の増 加する向きは画面下方に向いていて,通常と逆向きです。次に,この描画枠に"論理座標"を 割り当てます。g.window(0,-1.2,2*PI,1.2)は描画枠の左下隅を論理座標値の(0,-1.2) に,右上隅を(2*PI,1.2)に割り当てます。座標軸の取り方は通常の場合と同じ増加方向で す。グラフを書くときには,この論理座標値を基準にして描きます。



図 18: 数値の意味

この他の機能として,画面に点を打つg.pset(x,y)等があります。詳細はデスクトップ上 にある glibw32の" 関数仕様 HLP"を参照してください。

コンパイルをする前にglibw32.libを下 記の手順でプロジェクトに追加します。プロ ジェクトペインで右クリックして[ファイル の追加]を選び(図19),図20で下記のファ イルを選択します(作業用のフォルダーには ありません)。

C:\CBuilderX\lib\glibw32.lib



図 19: ライブラリーの追加

ତ "first.cbx" (C)	追加				2
Fィレクトリ(D):	dil 🛄			💌 📴 🛍 💆 (🐼 • 😰
デスクトッ デスクトッ ホーム 「 プロジェク サンブル		ant jar audio castor-0.9.5.jar cbuilder jar cloudscape jar EccpressoAll jar ext flextm.jar JarManifestFix jar javac.jar		bcbatl.lib bfc42.lib bfc42d.lib bfcs42d.lib cw32.lib cw32.lib cw32lib cw32lib glibw32.lib glibw32.lib glibw32.lib anafxcwd.lib	
	ファイル名(N): ファイルタイプ(T	glibw32.lib): パイナリビルドファイル	/ (*.a; *.build;	*.dll; *.lib; *.res)	
			ОК	キャンセル	ヘルフ

図 20: ライブラリーの選択

追加されたライブラリーファイルはプロジェクトペインに現れます(図 21)。



図 21: 追加されたライブラリー

これでライブラリーを利用する準備が整ったの で,プログラムをコンパイルします。ただしプロ グラムの実行はツールバーの [プロジェクトのデ バッグ] をクリックします (図 22)。コンソール窓 (黒い画面) とグラフ画面が表示されます (図 23)。

-	A Windows	+	👫 Debug Build 🔹	•	- 6	- 付		۲
							7トのデバッ	<u>, 4</u>

図 22: プロジェクトのデバッグ

これと同時に下記のファイルが作成されます。administrator はそれぞれのユーザー名に, first は各プログラム名に置き換えてください。この bmp ファイルがグラフ画像です。

 $\texttt{C:\Documents}$ and <code>Settings</code>

\Administrator\cbproject\first\windows\Debug_Build\myfile.bmp



図 23: グラフの表示

3.6 課題2

以下の問いに答えなさい。プログラムリストと結果のグラフを提出すること。

a. 下記の式のグラフを $(0 \le x \le 2\pi)$ の範囲で書きなさい。

 $f(x) = \sin(4x)\exp(-x)$

b. 下記の式のグラフを $(-2\pi \le x \le 2\pi)$ の範囲で書きなさい。

 $f(x) = \sin(4|x|)\exp(-|x|)$

4 重力場における運動

運動方程式を数値的に解く例として,重力場における上方に投げあげられた物体の運動を 考えてみましょう (図 24)。鉛直上方に投げあげるとして,時間をt,高さをx,速度をv で 表し,初期条件としてt = 0 s で速度を v_0 ,高さを0 m とします。また重力加速度はgです。

よって時刻 $t + \Delta t$ での速度は,時刻 t での速度に速度変化分 図 24: $v \ge v\Delta t$ を加えて,

 $v(t + \Delta t) = v(t) + (-g\Delta t)$

また Δt 間の移動距離 Δx は

 $\Delta x\simeq v\Delta t$

時刻 $t + \Delta t$ の位置座標は,時刻 t での座標 x(t) に上記の移動距離を加えて,

 $x(t + \Delta t) = x(t) + v\Delta t$

この計算を繰り返し行うことによって,位置座標の時間的推移を求めることができる。以 上をプログラミングで表現してみましょう。

4.1 最大値を求める

下記のプログラムが,上方に投げあげられた物体の運動を求めるものです。その中で使用 している変数は t[i], v[i], x[i] がそれぞれ時間,速度,座標を表しています。ひとつの ステップの時間間隔を 0.01 s として 4 s まで計算しています。初期条件は t = 0 で x[0] = 0, v[0] = 9.8 としています。以下先のステップの速度と座標を用いながら,0.01 s 後の速度 と座標を計算しています。以下のように内容ペインに記入してください。

```
#include <stdio.h>
#include <math.h>
#include "glibw32.h"
#define SIZE 401
double t[SIZE],tmax=-1000000,tmin=1000000;
double v[SIZE];
double x[SIZE];
double x[SIZE],xmax=-1000000,xmin=1000000;
double dt=0.01,g=9.80;
int main()
{
  t[0]=0.0;
  x[0]=0.0;
  v[0]=9.80;
  for(int i=1; i<=(SIZE-1); i+=1){</pre>
        t[i]=dt*i;
        v[i]=v[i-1]+(-g)*dt;
        x[i]=x[i-1]+v[i]*dt;
  }
for(int i=0; i<=SIZE-1; i+=1){
    if(t[i]>tmax)(tmax=t[i]);
if(t[i]<tmin)(tmin=t[i]);</pre>
if(x[i]>xmax)(xmax=x[i]);
if(x[i]<xmin)(xmin=x[i]);</pre>
7
printf("tmax = %f tmin = %f \n",tmax,tmin);
printf("xmax = %f xmin = %f \n", xmax, xmin);
ginit(450,350,WHITE);
GRAPH g;
char strl[]="自由落下";
textout(10,5,strl,BLACK,1,1);
g.window(0,1.2*xmin,1.1*tmax,1.2*xmax);
g.view(20,20,420,320);
g.axis(tmax/10,(xmax-xmin)/10);
g.setcolor(RED);
for(int i = 0; i<=SIZE-1; i+=1){</pre>
     g.pset(t[i],x[i]);
}
savebmp("myfile.bmp"); // gend()の直前に書くこと!
gend();
return (0);
```

グラフを作図するには時刻と座標の最大値最小値を知る必要があります。このためここでは if 文を利用しています。

```
if(x[i]>xmax)(xmax=x[i]);
```

上記の if 文は, x[i] が現在の xmax よりも大きいならば, xmax を, その x[i] で置き換 えます。これを, for ループを用いて, i の値を変更しながら繰り返すことによって, 座標 の最大値を xmax にセットします。

コンパイルして実行すると下記のグラフが得られます (glibw32.lib を忘れないこと)。



図 25: 自由落下

4.2 課題3

野球のボールを 20 秒間自由落下させる場合を考える (初速ゼロ)。この場合ボールには重 力の他に速度の 2 乗に比例する力が働く。この比例係数を $\mu = 0.006125$ とするとき速度の 時間変化は下記の式で表される。このときの速度と時間の関係のグラフを書きなさい (プロ グラムリストと結果のグラフを提出すること)。また速度の大きさの最大値はいくらになりま すか。

$$\frac{dv}{dt} = -g + \mu v^2$$

5 プログラミングによる機器の制御

以下では、プログラミングによる機器の自動制御を取りあげます。カノープス株式会社製の USBit を用いてラジコンカーをコントロールしてみます (図 26 の左にあるものが USBit)。



🗵 26: USBit

5.1 1秒間の直進

新規にプロジェクトを作成し,内容ペインに下記のように記入します。このプログラムは 車を1秒間直進させて停止させるものです。

USB 接続の送信装置を動作させるために,プロジェクトペインに下記のライブラリーを追加します。

C:\CBuilderX\lib\USBitLib_OMF.lib

#include "USBitLib.h"は送信機を制御する命令の宣言文です。命令には下記のものがあ ります。

USBit_SetCommand(hUSBit, USBIT_CMD_BACKLEFT, 0) ; 左後方 USBit_SetCommand(hUSBit, USBIT_CMD_BACKWARD, 0) ; 後進 USBit_SetCommand(hUSBit, USBIT_CMD_BACKRIGHT, 0) ; 右後方 USBit_SetCommand(hUSBit, USBIT_CMD_STOP, 0) 停止 USBit_SetCommand(hUSBit, USBIT_CMD_FORLEFT, 0) 左前方 USBit_SetCommand(hUSBit, USBIT_CMD_FORWARD, 0) 前進 ; USBit_SetCommand(hUSBit, USBIT_CMD_FORRIGHT, 0) 右前方 :

main 以下最初の4行は送信機を初期化する命令です(このまま記入してください)。 時間の計測にはGetTickCount()関数を使用します。#include <windows.h>で,この関 数が定義されているパッケージを読み込んでいます。時間はミリ秒単位の整数値として得られ ます。計測した時間を代入する変数tick,tick0はlong int(大きな整数)としてデータ型 を指定しています。1秒間という時間の間隔を作り出すために,while()ループを用います。

while (条件文){命令文}

これは()内の条件式が成立するまで,続く{}内の命令を繰り返し実行するもので,先のプログラムでは,時間の初期値tick0から1000msec経過するまで時間計測を繰り返します(tick0とtickとの差が1000になるまで)。

プログラムをコンパイルした後,コンピューターの USB 端子に USBit が接続されている のを確認してラジコンカーを走らせて見てください。最初はラジコンカーは充電されていま せんので,備え付けのファイルを参照して充電してください。

5.2 課題4

用意してある障害を回って8の字走行をするプログラムを作成し,プログラムリストを提 出すること。



図 27:8 の字走行