卒 業 研 究 報 告

題　　目

Windows プログラミングによる
グラフ作成ソフトの開発

指　導　教　員

武田 光由 実験講師

報　告　者

鶴野　哲生

平成 １４年　　２月　　７日

高知工科大学 電子・光システム工学科

目　次

第１章　序論 1

１．１　背景 1

１．２　目的 1

第２章　仕様 3

２．１　グラフソフトの比較 3

２．２　グラフ作成ソフトの仕様 5

２．３　流れ図の説明 6

第３章　補間法 9

３．１　補間法 9

３．２　使用する補間法 9

３．２．１　ラグランジュ補間 9

３．２．２ ３次スプライン補間 10

３．２．３ 線形最小２乗法 12

３．３　補間法の比較 14

第４章　Windows プログラミング.....16

４．１　Windows プログラミング 16

４．２　API 関数..... 16

第５章　プログラムの解説とグラフ作成ソフトの使用法.....21

５．１　プログラムの解説..... 21

５．２　グラフ作成ソフトの使用法..... 25

第６章　結論.....30

参考文献.....31

謝辞.....32

付録　プログラムソース.....33

第１章　序論

１．１　背景

かつてコンピューターは研究機関や企業などでの研究・業務用として使われていた。しかし、情報化社会といわれるようになった現在、我々の生活の中でパソコンはテレビなどの家庭電化製品と同様一般に普及している。この要因には以下のようなものが挙げられる。

- OS の進化
- パソコンの低価格化
- ソフトの充実
- インターネットやゲームができるなどの用途の拡大

時代が CUI の「MS-DOS」から GUI（Graphical User Interface）の「Windows」へ移行するにともなってパソコンが一般家庭にまで浸透した。これにより一部の専門家などが使うソフトから大衆向けのソフトになり、プログラマーに要求されるプログラミングの能力に変化が生じた。

CUI の時代では処理系によって若干の違いはあるものの基本的には「C 言語」などの単一の言語だけでプログラムすることが可能であった。しかし、「Windows」の場合にはいわゆる「Windows プログラミング」が必要にりプログラミングが複雑になったが、Microsoft 社の「Visual Stdio」や Borland 社の「Delphi」などが「Windows プログラミング」を容易にすることを可能にした。

１．２　目的

本研究は科学技術分野で使用するグラフ作成ソフトの開発をすることが目的である。工学分野ではレポートや論文に実験データや理論計算などのグラフを使用するため様々なグラフソフトが利用されている。現在、世の中に普及しているグラフソフトは市販品に限らずフリーソフトにも優れたものが多くあり、使用目的や好みに応じて選ぶことができる。しかし、それらのソフトを実際に使用して機能や操作性において不満を覚えることもある。そこで、この不満点を解消した独自のグラフソフトを開発することが最終目標である。しかし、卒業研究といったごく限られた期間でこれらのソフトを凌ぐものを開発することは不可能であると考えている。そこで、本研究では、現在普及しているグラフ

ソフトの機能を参考にしながら最終目的であるグラフソフトの Ver.１を作成することを目指す。

「背景」でも記したように、グラフ作成ソフトを開発するに当たって単に測定データをグラフ化するだけではなく、ソフトの操作性向上のためにもインターフェースの充実が重要である。したがって、本研究を進めるうえで以下の準備が必要である。

1. プログラミング言語の習得
2. GUI プログラミング技術の習得

MS－DOS などのパソコンや大型計算機が主流であった時代では、プログラム言語として Basic や Fortran が広く用いられてきた。しかし、現在ではシステム開発の分野において C 言語が最も使用されている。そして、企業や研究機関、一般家庭を含めたパソコンの OS は Windows 系統のものが最も普及している。これらの理由からプログラミング言語では C 言語を選択し、開発環境は Visual Studio による Windows プログラミングを選択した。これらの技術の習得も本研究目的の一部であり、習得した技術をもとに今回作成したソフトをより完全なものに近づけて行きたいと考えている。

第２章　仕様

２．１　グラフソフトの比較

我々が普段使用するグラフには実に様々なものがあり、それらのグラフは用途により使い分けられている。また、グラフソフトも幾つかありそのグラフの機能もいろいろである。図１にグラフの例を示す。

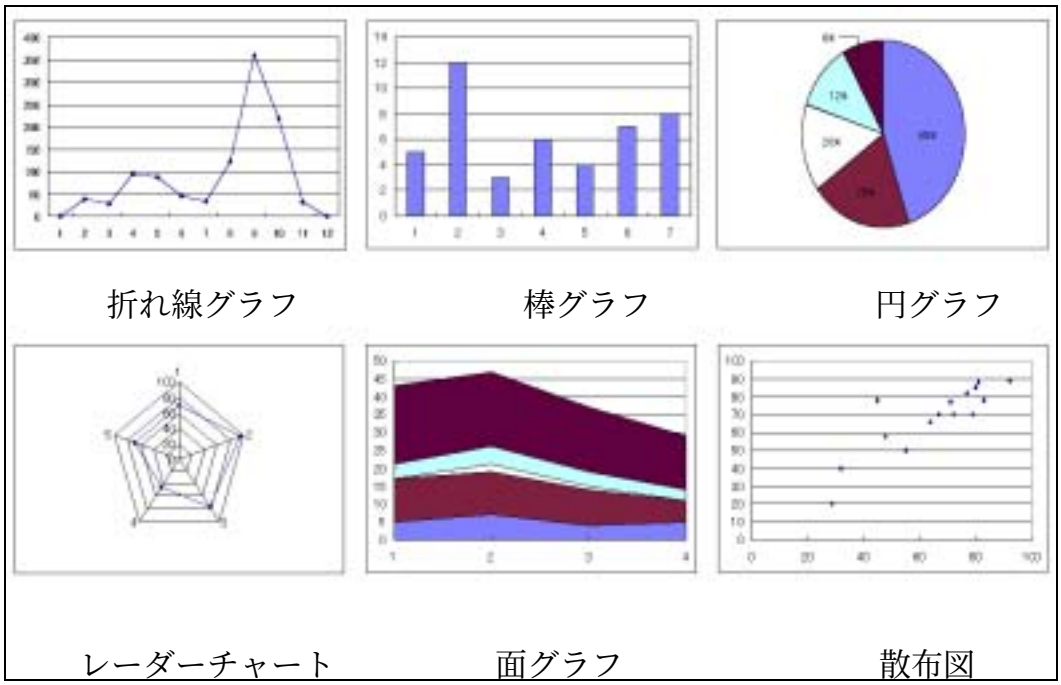

図 1

世の中には色々なグラフを描くことのできるソフトがある。以下にそのソフトについての説明をする。

1．「Excel」

これのソフトは表計算ソフトでその機能の一環としてグラフが描ける。グラフの種類は豊富である。しかし、科学計算のグラフ製作が専門のソフトではないため、実験データなどのグラフを作成するには不向きである。

2. 「Sma４」

科学の世界で広く使われている散布図を簡単に作成できる 2 次元グラフ描画ソフトである。このソフトはテキスト形式の関数やデータから自動的にグラフを作成することができ、対数プロットやスプライン補間、最小二乗フィットを使用したグラフ表示機能も備えている。また、数式変換機能もあり演算子や絶対値、平方根といった組み込み関数による数式変換もできる。

3. 「Ngraph」

このソフトは 2 次元グラフ作成ソフトである。 テキスト形式の実験データ等を読み込んでグラフ化、データ解析するほか、ドロー機能やスクリプト（マクロ）なども備えている。グラフは n 次関数・ユーザ定義関数による最小二乗フィットが可能であり、プロットデータの評価・マスク・変更もできる。さらに任意指定の数式によりデータ加工する数式変換機能もある。

4. 「gnuplot」

このソフトは対話的にグラフを描くソフトである。グラフは、関数の他に、実験データなどの離散的な数値も描くことができる。また、2 次元と 3 次元のグラフが描ける。利点としては結果を tex のプログラムとして出力できる。

5. 「EasyGraph」

このソフトは数式を入力すると、画面にそのグラフを描くソフトである。表示できるグラフは、関数、極形式、パラメータ曲線の 3 種類である。グラフ式にパラメータを使うと、パラメータを変化させながらグラフの変化の様子を観察することができる。

6. 「PLOT」

このソフトは２次元グラフを作成するソフトである。定型のテキストファイルや、Excel の表などをクリップボード経由で読み込み、X 軸と Y 軸のあるグラフを描画する。グラフはプリンターに直接出力できるほか、クリップボード経由でベクトル形式のメタファイルとして、ワープロソフトなどの外部ソフトに貼り付けることができる。

7. 「GraphCalc」

このソフトは２次元・３次元グラフを作成するソフトである。３次元グラフはマウス操作で回転させて様々な角度から見ることができる。３次元グラフは画像として保存したり印刷ができるほか、DXF 形式のファイルで保存することも可能で、多くの 3D CG モデリングソフトで読み込んで編集することが可能である。

２．２　グラフ作成ソフトの仕様

２．１で紹介したグラフソフトにある機能を参考にしながら本グラフ作成ソフトは以下に列挙する機能を付加した。

（１）グラフの設定

- 「倍率指定」・・・・・・・・・・・・グラフの倍率を指定する。
- 「軸比率」・・・・・・・・・・・・・X 軸・Y 軸の比率を変更する。
- 「中心座標」・・・・・・・・・・・・中心の座標を変更する。
- 「色設定」・・・・・・・・・・・・・グラフの色を設定する。
- 「プロット設定」・・・・・・・・・プロットの形を設定する。
- 「グラフの設定」・・・・・・・・・グラフの結び方を設定する。
- 「コメントの挿入」・・・・・・・・X 軸・Y 軸にコメントを挿入する。
- 「グラフの削除」・・・・・・・・・グラフを削除する。

（２）ユーザーインターフェース

- 「新規作成」・・・・・・・・・・・・新規にグラフを作成する。
- 「開く」・・・・・・・・・・・・・・以前保存したファイルを開く。
- 「名前を付けて保存」・・・・・・・設定を名前を付けて保存する。
- 「上書き保存する」・・・・・・・・設定を上書き保存する。
- 「画像の保存」・・・・・・・・・・グラフを画像として保存する。
- 「プリンタの設定」・・・・・・・・プリンタの設定をする。
- 「印刷」・・・・・・・・・・・・・・グラフを印刷する。

（３）その他の機能

- 「ヘルプ」・・・・・・・・・・・・・ヘルプを表示する。

- 「バージョン情報」・・・・・・・・バージョン情報を表示する。

２．３　流れ図の説明

図２は本グラフ作成ソフトの起動から終了までの一連の流れ図である。

ソフトを起動すると「初期画面」が現れる。次に数値を入力してやる必要がある。数値の入力方法には手動での入力と以前保存した設定ファイルからの読み込みの２種類がある。手動での入力をする場合は「設定」を選び、以前の設定ファイルからの読み込みをする場合には「開く」を選ぶ。

数値の入力がすむと前述のグラフ関連の設定で説明した各種「グラフの設定」をしなければならない。グラフの設定が終わるとグラフがウィンドウに表示される。現在ウィンドウに表示されているグラフを消去したい場合は「グラフの消去」を選ぶび初期画面に戻る。これを選択したことによりこれまで入力した数値やグラフの設定までは削除されない。

現在のグラフの各種設定、入力数値を保存したい場合は「上書き保存」、「名前を付けて保存」のどちらかを選ぶ。また、画像を BitMap として保存したい場合は「画像の保存」を選択する。

現在表示されているグラフを印刷する場合、「プリンタの設定」を選び用紙のサイズや印刷するむきなどを変更することができる。

グラフの設定の保存や印刷などが終わりこれとは別に新規にグラフを作成したい場合は「新規作成」を選択する。これは「グラフの消去」と違いグラフの設定や入力した数値なども消去してしまうので設定を残したい場合はあらかじめ保存しなければならない。これを選択すると初期画面に戻り以下これまでの処理を繰り返す。

新規にグラフを作成する必要がない場合はここでソフトの起動からの一連の動作は終了である。

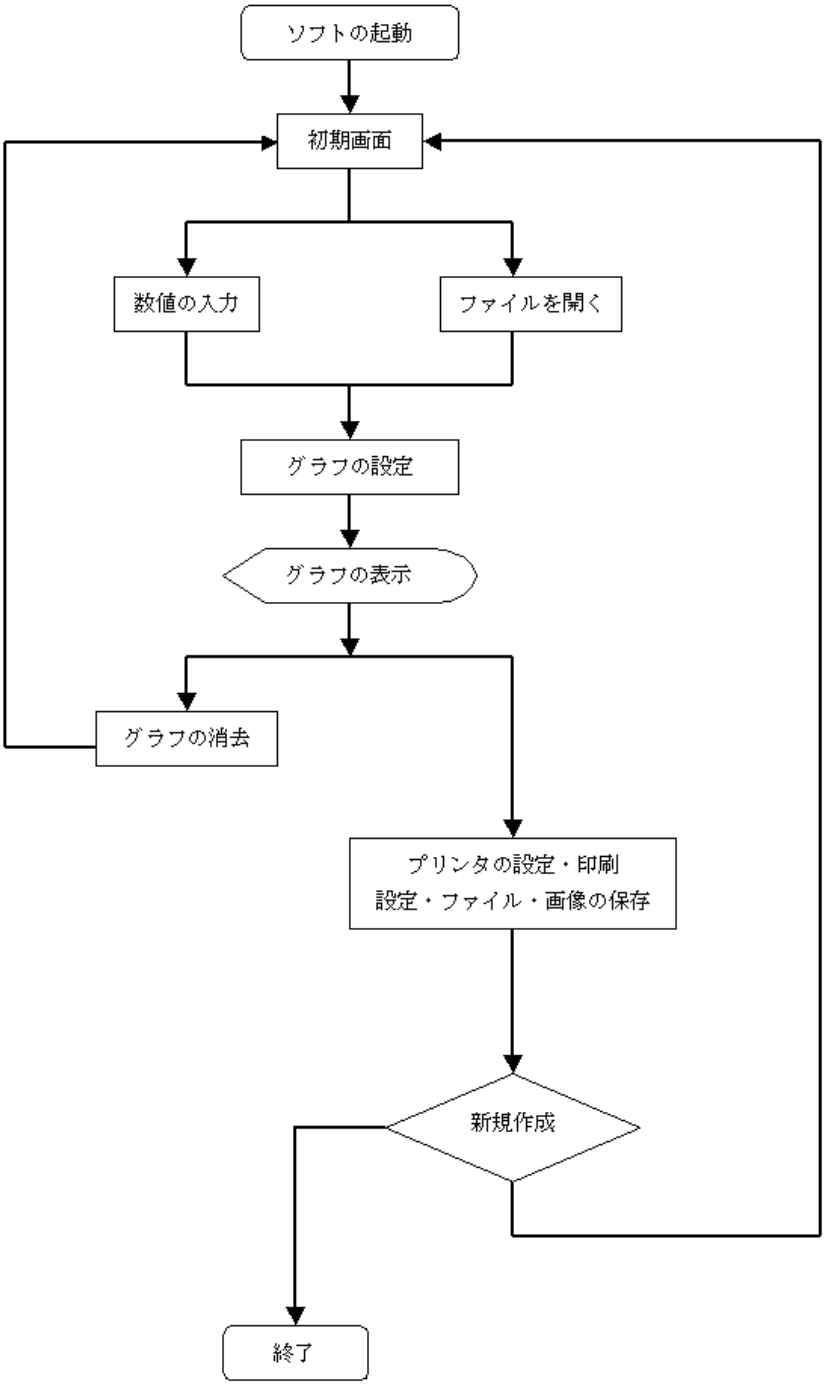

図 2

本グラフ作成ソフトの実際の画像を用いた使い方の説明は第５章で詳しく説明する。また、本グラフソフトは図１で説明した「棒グラフ」や「円グラフ」のようなグラフではなく、科学技術分野での実験データなどをグラフ化するこ

とを目的としている。そのために「折れ線グラフ」や「散布図」といったグラフを扱い測定値などのグラフを滑らかに結ぶことが必要となる。そのためには補間法が必要になり、数ある補間法の中でも「３次スプライン補間」、「ラグランジュ補間」、「線形最小２乗法」を用いてグラフを描く。補間法については次章で詳しく説明する。

第３章　補間法

３．１　補間法

補間法とは、測定値の間を合理的につなぐ方法である。最も単純な測定値のつなぎ方はおのおのの測定点を直線で結ぶものである。補間には与えられた点を通る滑らかな多項式を求める手法と、与えられた点に近い点を通る滑らかな曲線や面を求める手法とがある。後者は図形処理や最小２乗法で用いられる。

３．２　使用する補間法

３．２．１　ラグランジュ補間

ラグランジュ補間とはn個の点（x_i , y_i）（$i = 0,1,\Lambda ,n-1$）が与えられたとき、$y_i = P(x_i)$（$i = 0,1,\Lambda ,n-1$）を満たす$n-1$次多項式を求め、データ点以外の点の値を求める方法である。この補間多項式は以下のように求められる。

$$f(x) = \frac{(x-x_1)(x-x_2)\Lambda\ (x-x_n)}{(x_0-x_1)(x_0-x_2)\Lambda\ (x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\Lambda\ (x-x_n)}{(x_1-x_0)(x_1-x_2)\Lambda\ (x_1-x_n)}y_1$$
$$\Lambda + \frac{(x-x_0)(x-x_1)\Lambda\ (x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\Lambda\ (x_n-x_{n-1})}y_n \qquad (３．１)$$

$$= \sum_{i=0}^{n}\left(\sum_{j=0}^{n}\frac{x-x_j}{x_i-x_j}\right)y_i \qquad (３．２)$$

（ただし　$i = j$ の場合は除く）

表２のx, yにラグランジュ補間を使ってグラフ化したものが図３のグラフである。

x	-2	-1	0	1	2
y	3	1	0	2	4

表 1

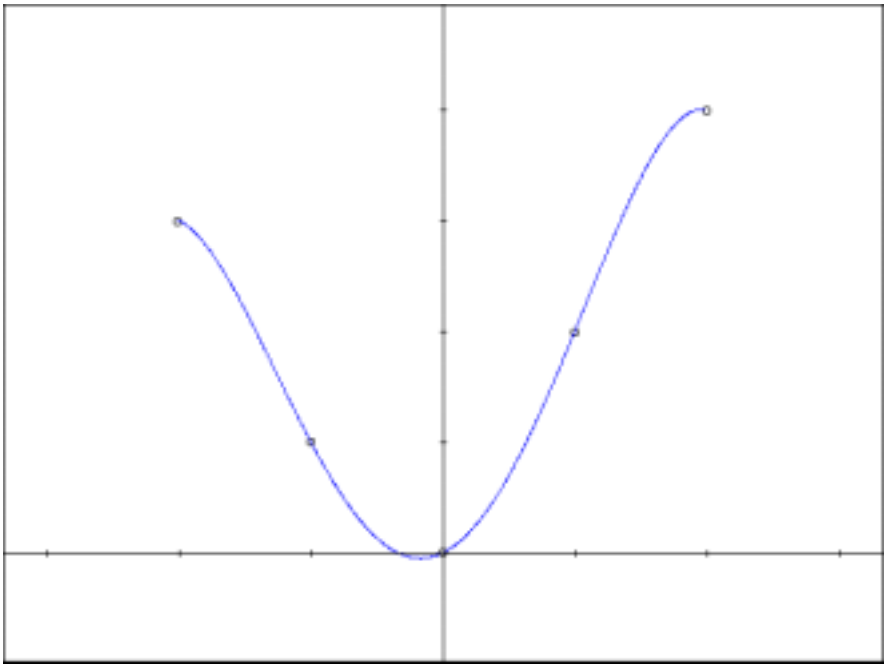

図 3

３．２．２　３次スプライン補間

３次スプライン補間とは、点 $x_0, x_1, \Lambda, x_{n-1}$ の関数値 $f(x_0), f(x_1), \Lambda, f(x_{n-1})$ と、両端における微分係数 $f'(x_1)$ と $f'(x_{n-1})$ を与えられたとき、これらの値と一致するような補間式

$$s(x) = f(x_i) + c1_i(x - x_i) + c2_i(x - x_i)^2 + c3_i(x - x_i)^3 \qquad （３．３）$$

$$x_i \leqq x \leqq x_i, \quad i = 0, 1, \Lambda, n_{i-2}$$

の係数 $c1_i, c2_i\ c3_i$ を求める方法である。次のような手順で ck_i は求められる。

各区間の長さ h_i を求める。（$i = 0, \Lambda, n-2$）

$$h_i = x_{i+1} - x_i$$

λ_i, ν_i, d_i を計算する。（$i = 1, \Lambda, n-2$）

$$\lambda_i = \frac{h_i}{h_{i-1}-h_i},\quad \nu_i = 1-\lambda_i,\quad d_i = \frac{6}{h_{i-1}+h_i}\left\{\frac{y_{i+1}-y_i}{h_i}-\frac{y_i-y_{i-1}}{h_{i-1}}\right\} \qquad (3.4)$$

$\lambda_0 = 1,\quad \nu_{n-1} = 1$とする。

d_0とd_{n-1}を計算する。

$$d_0 = \frac{6}{h_0}\left\{\frac{y_1-y_0}{h_0}-f'(x_0)\right\},\quad d_{n-1} = \frac{6}{h_{n-1}}\left\{f'(x_{n-1})-\frac{y_{n-1}-y_{n-2}}{h_{n-1}}\right\} \qquad (3.5)$$

３重対角行列

$$\begin{bmatrix} 2 & \lambda_0 & & & & \\ \nu_1 & 2 & \lambda_2 & & & \\ & \Lambda & \Lambda & & & \\ & & \nu_{n-2} & 2 & \lambda_{n-2} & \\ & & & \nu_{n-1} & 2 & \end{bmatrix}\begin{bmatrix} M_0 \\ M_1 \\ \mathrm{M} \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \mathrm{M} \\ d_{n-2} \\ d_{n-1} \end{bmatrix} \qquad (3.6)$$

を解いて、M_iを求める。

M_iを用いてck_iを求める。（$i = 0,1,\Lambda\ ,n_{i-2}$）

$$c2_i = \frac{M_i}{2},\quad c3_i = \frac{M_{i+1}-M_i}{6h_i},\quad c1_i = \frac{y_{i+1}-y_i}{h_i}-c2_i h_i - c3_i h_i^{\,2} \qquad (3.7)$$

ラグランジュ補間でグラフ化したものと同じ表１でのx, yの点を３次スプライン補間でグラフ化したものが図４のグラフである。

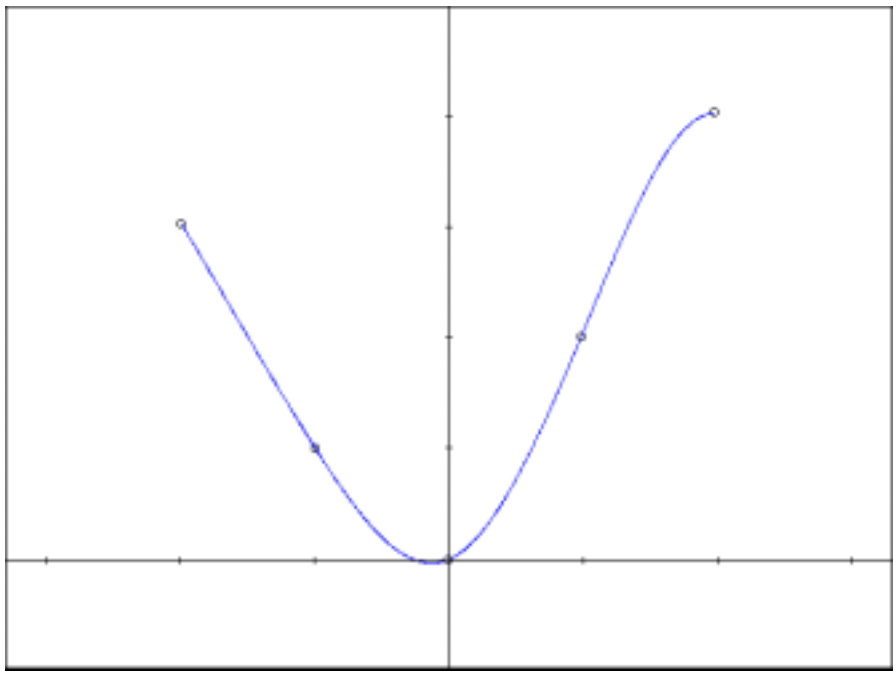

図 4

３．２．３　線形最小２乗法

線形最小２乗法は点 $(x_0, y_0), (x_1, y_1), \Lambda, (x_{n-1}, y_{n-1})$ が与えられたとき、

$$y = ax + b$$

が成立するとして、各点から直線 $y = ax + b$ に鉛直に下ろした線分の長さの２乗の和

$$S = \sum_{i=0}^{n} (y_i - ax_i - b)^2 \qquad (i = 0, 1, \Lambda, n_{i-1}) \qquad （３．８）$$

を最小にするような a, b を決定するの方法である。S をパラメータ a と b で微分して、それぞれ０とおけば S の最小値を求めることができる。

$$\frac{\partial S}{\partial a} = 2\left\{a\sum x_i^{\,2} + b\sum x_i - \sum x_i y_i\right\} = 0 \qquad （３．９）$$

$$\frac{\partial S}{\partial b} = 2\left\{a\sum x_i + \sum b - \sum y_i\right\} = 0 \qquad （３．１$$

0）
これを a と b について解くと

$$a = \frac{n\sum_i x_i y_i - \left(\sum_i x_i\right)\left(\sum_i y_i\right)}{n\sum_i x_i{}^2 - \left(\sum_i x_i\right)^2} \qquad (3.11)$$

$$b = \frac{\sum_i y_i - a\sum_i x_i}{n} \qquad (3.12)$$

となる。

表２はランダムにとった１５点の x, y の値である。これを最小２乗法を用いてすべての x, y を通過する最もよい直線で結んだものが図５のグラフである。

この直線は最小２乗法により切片が約12.48で傾きが約0.85である１次関数

$$y = 0.85x + 12.48$$

となった。

x	29	32	45	48	55	64	67	71	72	77	79	80	81	83	92
y	20	40	78	58	50	66	70	77	70	82	70	85	88	78	89

表 2

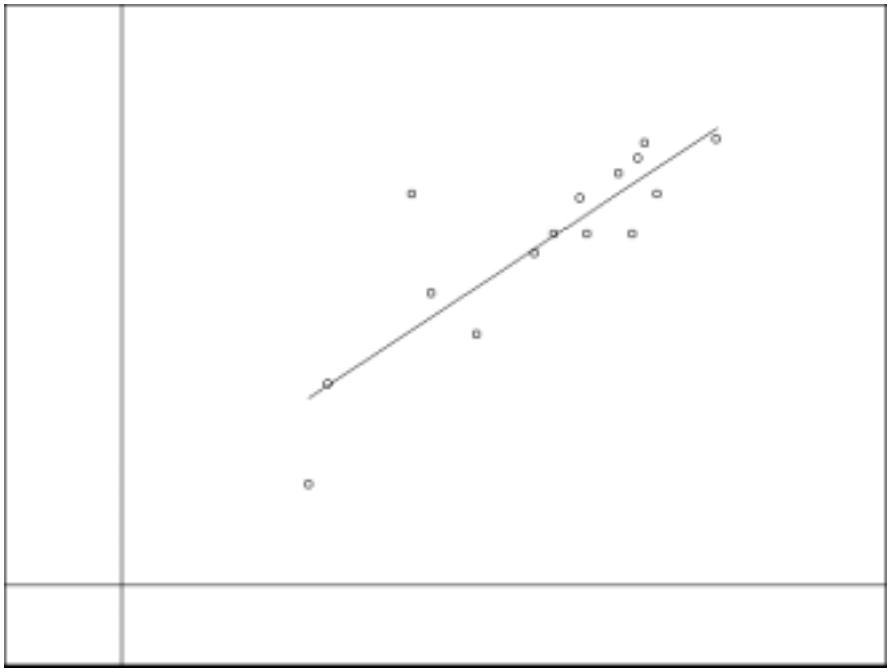

図 5

３．３　補間法の比較

図３のラグランジュ補間と図４の３次スプライン補間で描いた２つのグラフを比較するとさほど変わりないように見える。しかし、どのような測定値であっても図３、４のように同じようなグラフになるとは限らないので注意が必要である。

x	-5	-4	-3	-2	-1	0	1	2	3	4	5
y	0.19	0.29	0.5	1	2.5	5	2.5	1	0.5	0.29	0.19

表 3

表３は$y = \frac{5}{(1+x^2)}$の式を（$-5 \leqq x \leqq 5$）の範囲であらわしたものである。この範囲でラグランジュ補間と３次スプライン補間でグラフ化したものが図６である。

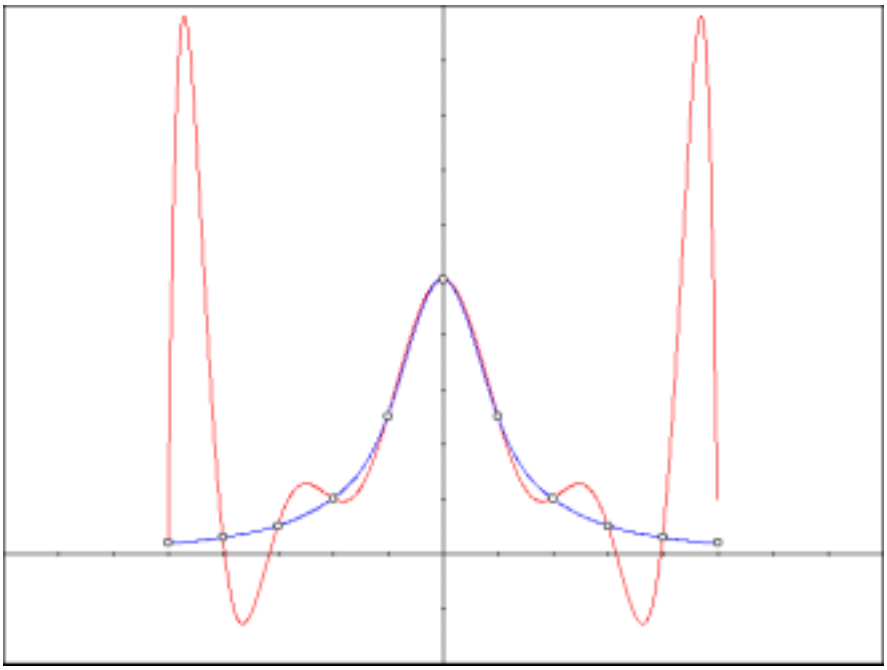

図 6

このグラフは図３、４のグラフと比較するとラグランジュ補間と３次スプライン補間で求めた補間点の値の違いが顕著に見られる。大きく角のように振動しているグラフがラグランジュ補間のグラフである。

先に述べたようにラグランジュ補間は $n-1$次多項式でる。これは計算量は少なくてすむが，高次式で、無理にすべての点を通ろうとするため振動しやすい欠点を有している。

それに対して，３次スプライン補間は隣接する３点を１次及び１次導関数が通るような３次式で補間するため多項式補間よりも振動しにくい。そのため変曲点も表現できる優れた性質をもつ補間法であるために実験データの処理に有効である。これらのことからラグランジュ補間の用途としては理論的に「測定値がＮ次関数」になる場合に使用する。

第４章　Windows プログラミング

４．１　Windows プログラミング

Windows プログラミングとは Microsoft 社の Windows 系統の OS で動作するプログラムを開発することを言う。最も基本的な Window は図 7－１のような構成になっており境界線内のクライアント領域の部分でアプリケーションの多くの出力が表示される。

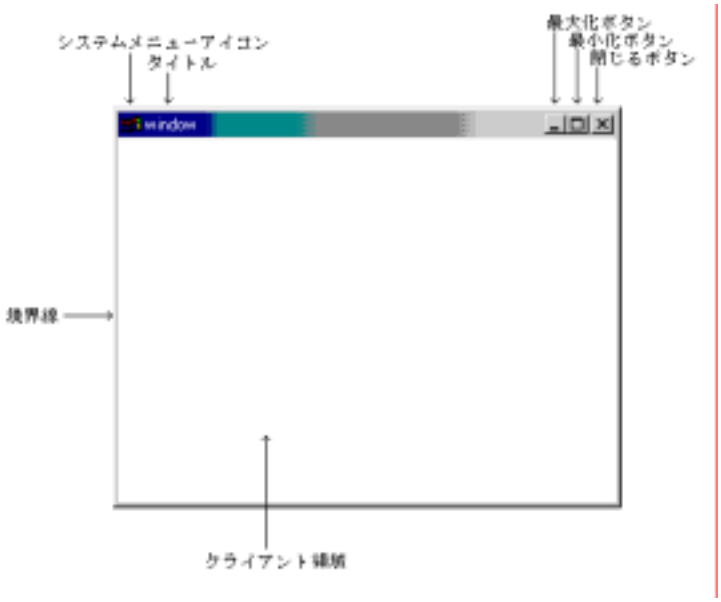

図７ - １

図７ - ２

４．２　API 関数

Win32 API とは Microsoft 社がアプリケーション開発のために定義したプログラミングインターフェイスのセットのことである。具体的には、関数、構造体（型)、定数などから構成されている。Win32 の「32」とは 32bit のことを意味しており、Windows3.0 以前の OS では Win16 で 16bit であった。

Win32 API には多数の関数が定義されているが、ここではグラフ作成ソフトを開発するにあたって使用した主な関数のみここでは説明していく。

HANDLE	オブジェクトを識別するための番号に使用される 32bit 整数
HWND	ウィンドウを識別するために使用するハンドル（ウィンドウハンドル）
BYTE	8bit の符号無し整数
WORD	16bit の符号無し整数
DWORD	32bit の符号無し整数
UNIT	unsigned int と同じ
LONG	long と同じ
BOOL	真か偽かを判断するのに使われる 32bit の整数
HINSTANSE	アプリケーションを識別するハンドル
LPSTR	文字列への 32bit ポインタ
LPCSTR	定数文字列への 32bit ポインタ

表 4 - 1

コンピュータには、メモリと呼ばれる記憶領域があり 1byte（8bit）単位に区切られ、その領域に大きさを指定して、データを確保する。それをデータ型と呼び Windows の基本的なデータ型は、基本的には C 言語と同じであるが使用目的により変数の型を細かく定めてる。データ型は、”windows.h”にマクロ定義されている。表 4-1 はよく使うデータ型の一部を列挙しそれぞれの解説をしたものである。

ダイアログボックス関連
GetDlgItem(HWND *hDlg*, int *nIDDlgItem*);
・指定された ID を持つコントロールのウィンドウハンドルを取得する。
GetDlgItemInt(HWND *hdlg*, int *nIDDlgItem*, BOOL **lpTranslated*, BOOL *bSigned*);
・ダイアログボックス内に指定されたコントロールのテキストを、整数値に変換する。
SetDlgItemText(HWND *hdlg*, int *nIDDlgItem*, LPCTSTR *lpString*);
・ダイアログボックス内のコントロールのテキストを設定する。
SetDlgItemInt(HWND *hdlg*, int *nIDDlgItem*, UNIT *uValue*, BOOL *bSigned*);

・指定された整数値を文字列に変換して、ダイアログボックス内のコントロールテキストとして設定する。
SendDlgItemMessage(HWND *hdlg*, int *nIDDlgItem*, UNIT *Msg*, WPARAM *wParam*, LPARAM *lParam*);
・ダイアログボックス内の指定されたコントロールにメッセージを送る。
EndDialog(HWND *hDlg*, int *nResult*);
・モーダルダイアログボックスを破棄し、処理を終了させる。

表4 - 2

我々がソフトを使うときにWindowsがユーザーからの情報を必要とするときがある。その時にユーザーが入力した情報をWindowsに送るための物がダイアログボックスである。
「ダイアログボックス」とはコントロールをより簡単にそして有効に使うために特化したウインドウの事で図6-2の様なものである。
ダイアログボックスの中には「エディットボックス」、「コンボボックス」、「リストボックス」、「ボタン」などがあり、これらに数値や文字列の入力、あるいは選択することによりWindows側に必要な情報を送ることができる。表4-2に上げたものがダイアログボックスを使用するときに主に必要API関数である。

デバイスコンテキスト関連
CreateDC(LPCTSTR *lpszDriver*, LPCTSTR *lpszDevice*, LPCTSTR *lpszOutput*, CONST DEVMODE **lpInitDate*);
・指定されたデバイスのデバイスコンテキストを作成する。
DeleteDC(HDC *hdc*);
・デバイスコンテキストを削除する。
DeleteObject(HGDIOBJ *hObject*);
・ペン、ブラシ、ビットマップなどのオブジェクトを削除する。
ReleaceDC(HWND *hWnd*, HDC *hDC*);
・デバイスコンテキストを開放する。
SelectObject(HDC *hdc*, HGDIBJ *hgdiobj*);
・デバイスコンテキストの指定のオブジェクトを選択する。

表4 - 3

クライアント領域に何らかの図形もしくは文字列を表示する場合にはどのような色の文字を表示するかどのような図形を描くかなどの処理をしなければならない。そこで必要になるのが表 4-3,4 にあるような API 関数である。これらの API 関数を使用することでクライアント領域に自由に図形や文字列を表示することができるようになる。

「デバイス コンテキスト」とは、グラフィック オブジェクトとそれに関連する属性、出力に影響するグラフィック モードを定義する構造体のことである。多くの Win32 構造体とは異なり、アプリケーションはデバイス コンテキストを直接アクセスすることはなく、さまざまな関数を呼び出して間接的に構造体を操作しなければならない。

描画関連
BeginPaint(HWND *hwnd*, LPPAINTSTRUCT *lpPaint*);
・WM_PAINT メッセージに対する描画を開始する。
EndPaint(HWND *hDlg*, int *nResult*);
・モーダルダイアログボックスを終了する。
CreateBitmap(int *nWidth*, int *nHeight*, UINT *cPlanes*, UINT *cBitsPerPel*, // number of bits required to identify a color CONST VOID **lpvBits*);
・ビットマップを作成する。
BitBlt(HDC *hdcDest*, int *nXDest*, int *nYDest*, int *nWidth*, int *hHeight*, HDC *hdcSrc*, int *nXSrc*, int *nYSrc*, DWORD *dwRop*);
・ビットブロック転送を行う。
PatBlt(HDC *hdcDest*, int *nXLeft*, int *nYLeft*, int *nWidth*, int *hHeight*, DWORD *dwRop*);
・ブラシで長方形の範囲でペイントする。
FillRect(HDC *hDC*, CONST RECT **lprc*, HBRUSH *hbr*);
・長方形を塗りつぶす。
CreateFont(int *nHeight*, int *nWidth*, int *nEscapement*, int *nOrientation*, int *fnWeight*, DWORD *fdwItalic*, DWORD *fdwUnderline*, DWORD *fdwStrikeOut*, DWORD*fdwCharSet*, WORD *fdwOutputPrecision*, DWORD *fdwClipPrecision*, DWORD *fdwQuality*, DWORD *fdwPitchAndFamily*, LPCTSTR *lpszFace*);
・フォントを作成する。
TextOut(HDC *hdc*, int *nXStart*, int *nYStart*, LPCTSTR *lpString*, int

cbString);
・文字列を描画する。
LineTo(HDC *hdc*, int *nXEnd*, int *nYEnd*,);
・カレントポジションから指定の点までの線分を描画する。
MoveToEx(HDC *hdc*, int *X*, int *Y*, LPPTONT *lpPoint*);
・カレントポジションを更新する。
CreatePen(int *fnPenStyle*, int *nWidth*, COLORREF *crColor*);
・ペンを作成する。

表4 - 4

第５章　プログラムの解説とグラフ作成ソフトの使用法

５．１　プログラムの解説

ここでは第３章で説明した「ラグランジュ補間」、「３次スプライン補間」、「線形最小２乗法」の３つのプログラムの解説をする。

ラグランジュ補間

```
double lagrange
(double nx[][50], double ny[][50], int n[], double x, int a)   ---------- ①
{
      int i, j;
      double fx = 0, sum = 0;

         for (i = 0; i < n[a]; i++) {
               fx = ny[a][i];
               for (j = 0; j < n[a]; j++) {
                    if (i != j)
                         fx = fx ＊ (x - nx[a][j]) /                    ②
                                   (nx[a][i] - nx[a][j]);
               }
           sum += fx;
         }
      return (sum);   ------------------------------------------------------------ ③

}
```

① lagrange 関数に引数として測定点の２次元配列 nx[ ][50] と ny[ ][50]、測定点の合計の要素 n[ ]，補間点の間隔 x，測定点の配列の要素 a を渡す。
② 式（３．２）の処理。条件 $j \neq i$ の場合を除く。
③ y の値を返す。

３次スプライン補間

```
double spline
(double nx[][50], double ny[][50], int n[], double x, int a)   ---------- ①
{
    int i;
    double p, s;
    double h[1000], d[1000], f[1000], e[1000], u[1000], m[1000], end[2],
           c1[1000], c2[1000], c3[1000];

    for (i = 0; i < n[a] - 1; i++)
    h[i] = nx[a][i + 1] - nx[a][i];

    f[0] = 1;
    e[n[a] - 1] = 1;                                                    ②

    for (i = 1; i < n[a] - 1; i++) {
        f[i] = h[i] / (h[i - 1] + h[i]);
        e[i] = 1 - f[i];
        u[i] = 6 / (h[i - 1] + h[i])  ＊  ((ny[a][i + 1] - ny[a][i]) /
                    h[i] - (ny[a][i] - ny[a][i - 1]) / h[i - 1]);
        }

    for (i = 0; i < n[a]; i++)
        d[i] = 2;

    end[0] = 0;                                                         ③
    end[1] = 0;
    u[0] = 6 / h[0]  ＊  ((ny[i][1] - ny[i][0]) / h[0] - end[0]);
    u[n[a] - 1] = 6 / h[n[a] - 2]    ＊
    (end[1] - (ny[a][n[a] - 1] - ny[a][n[a] - 2]) / h[n[a] - 2]);

    for (i = 0; i < n[a] - 1; i++) {
        p = e[i + 1] / d[i];
        d[i + 1] -= p  ＊  f[i];
        u[i + 1] -= p  ＊  u[i];                                        ④
    }
```

```
    u[n[a] - 1] /= d[n[a] -1];
     for (i = n[a] - 2; i >= 0; i--) {
          u[i] = (u[i] - f[i] ＊ u[i + 1]) / d[i];
     }                                                              ④
     for (i = 0; i < n[a]; i++) {
          m[i] = u[i];
     }

     for (i = 0; i < n[a] - 1; i++) {
          c2[i] = m[i] / 2;
          c3[i] = (m[i+1] - m[i]) / (6 ＊ h[i]);
          c1[i] = (ny[a][i+1] - ny[a][i]) / h[i] - c2[i] ＊
  h[i] - c3[i] ＊ h[i] ＊ h[i];
          if ((x >= nx[a][i]) && (x <= nx[a][i + 1])) {             ⑤
               s = ny[a][i] + c1[i] ＊ (x - nx[a][i]) + c2[i] ＊
                    (x - nx[a][i]) ＊ (x - nx[a][i]) +
                    c3[i] ＊ (x - nx[a][i]) ＊ (x - nx[a][i]) ＊
                   (x - nx[a][i]);
          }
     }

     return (s); ---------------------------------------------------------------- ⑥

}
```

① spline 関数に引数として測定点の２次元配列 nx[][50]と ny[][50]、測定点の合計の要素 n[],補間点の間隔 x,測定点の配列の要素 a を渡す。

② 式（３．４）の処理。

③ 式（３．５）の処理。

④ 式（３．６）の処理。３重対角行列を処理している。

⑤ 式（３．３）の係数$c1_i, c2_i\ c3_i$を求めている。

⑥ 式（３．３）に⑤で求めた係数を代入し、yの値を返す。

線形最小２乗法

```
double lsq1(double nx[][50], double ny[][50], int n[], int c)   ---------- ①
{
double    a, b, A=0, B=0, C=0, D=0;
int i;

 for (i = 0; i < n[c]; i++) {                                              ┐
A += nx[c][i] ＊ ny[c][i];                                                 │
B += nx[c][i];                                                             │  ②
C += ny[c][i];                                                             │
D += nx[c][i] ＊ nx[c][i];                                                 │
}                                                                          ┘

a = (n[c] ＊ A - B ＊ C) / (n[c] ＊ D - B ＊ B);                            ┐  ③
b = (C - a ＊ B) / n[c];                                                   ┘

return (a);   //lsq2 は b を返す。------------------------------------------- ④

}
```

① lsq1 関数に引数として測定点の２次元配列 nx[][50]と ny[][50]、測定点の合計の要素 n[],測定点の配列の要素 a を渡す。

② 式（３．９)、（３．１０）の処理。

③ 式（３．１１）の処理。②を a と b について連立方程式を解く。

④ a（lsq2 の場合は b）を返す

５．２　グラフ作成ソフトの使用法

新規にグラフを作成する

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムの新規作成を選択する。（図 8-1 を参照）

以前のファイルを開く

1. メニューバーのファイルを選択する。（図 7 を参照））
2. メニューアイテムの開くを選択する。（図 8-1 を参照）

ファイルの保存

「名前を付けて保存」の場合

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムの名前を付けて保存を選択する。（図 8-1 を参照）

「上書保存」の場合

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムの上書保存を選択する。（図 8-1 を参照）

グラフの保存

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムの画像の保存を選択する。（図 8-1 を参照）

プリンタの設定

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムのプリンタの設定を選択する。（図 8-1 を参照）

印刷

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムの印刷を選択する。（図 8-1 を参照）

ソフトを終了する

1. メニューバーのファイルを選択する。（図 7 を参照）
2. メニューアイテムの終了もしくはタイトルバー右上の×印を選択する。（図 8-1、7 を参照）

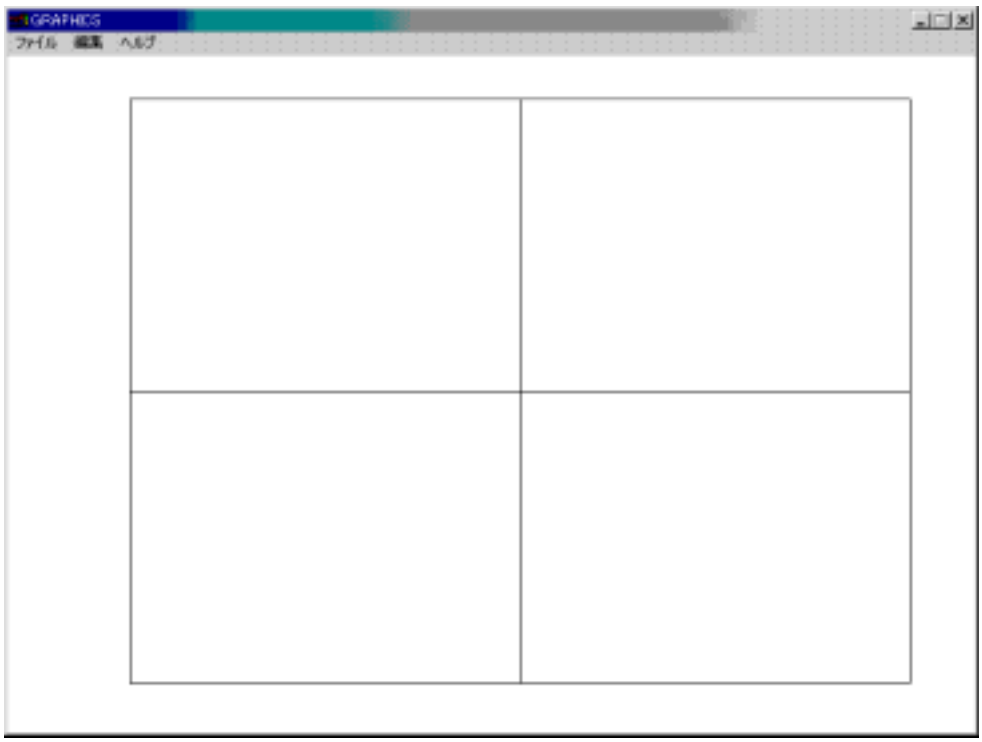

図 7

図 8 - 1

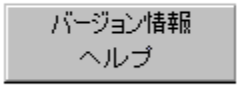

図 8 - 2

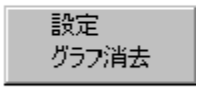

図 8 - 3

グラフの作成

1. メニューバーの編集を選択する。(図 7 を参照)
2. メニューアイテムの設定を選択する。(図 8-3 を参照)
3. プロパティーシートに数値を入力し OK ボタンをクリックする。(図 10 を参照)
4. メニューバーの編集を選択する。(図 7 を参照)
5. プロパティーシートでグラフの設定をし OK ボタンをクリックする。(図 9 を参照)

グラフの消去

1. メニューバーの編集を選択する。(図 7 を参照)
2. メニューアイテムのグラフの消去を選択する。(図 8-3 を参照)

バージョン情報を見る

1. メニューバーのヘルプを選択する。(図 7 を参照)
2. メニューアイテムのバージョン情報を選択する。(図 8-2 を参照)

ヘルプを見る

1. メニューバーのヘルプを選択する。(図 7 を参照)
2. メニューアイテムのヘルプを選択する。(図 8-2 を参照)

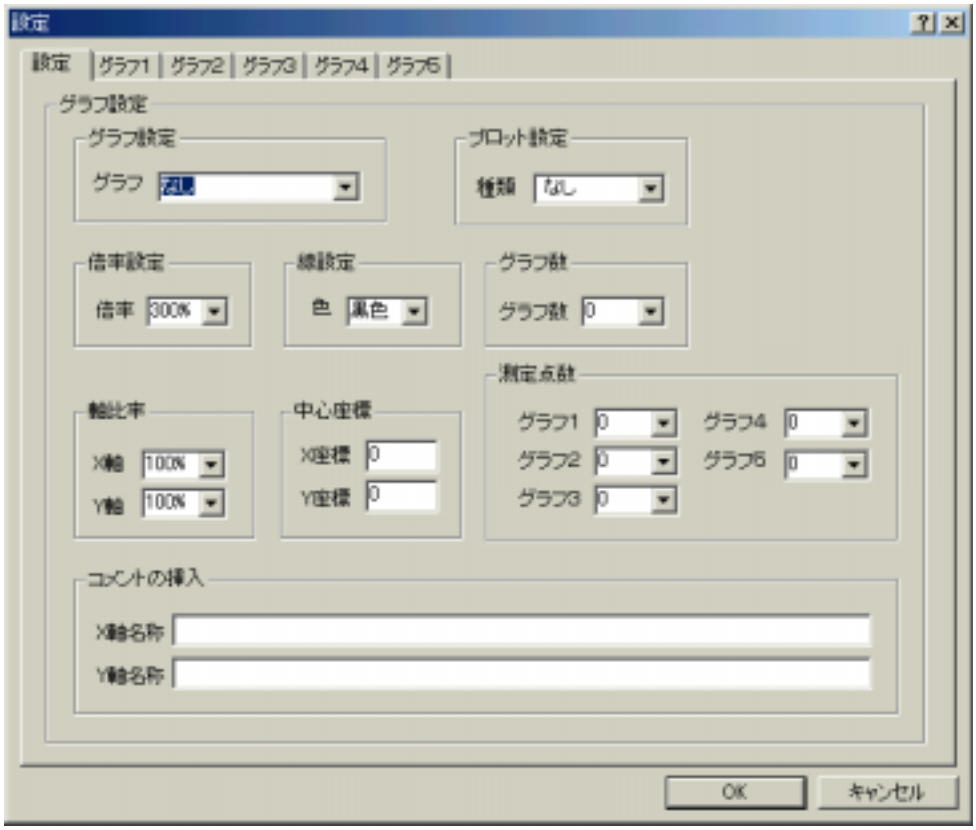
設定
設定
グラフ1
グラフ2
グラフ3
グラフ4
グラフ5
グラフ設定
グラフ設定
グラフ
プロット設定
種類
なし
倍率設定
倍率
300%
線設定
色
黒色
グラフ数
グラフ数
0
軸比率
X軸
100%
Y軸
100%
中心座標
X座標
0
Y座標
0
測定点数
グラフ1
0
グラフ2
0
グラフ3
0
グラフ4
0
グラフ5
0
コメントの挿入
X軸名称
Y軸名称
OK
キャンセル

図 9

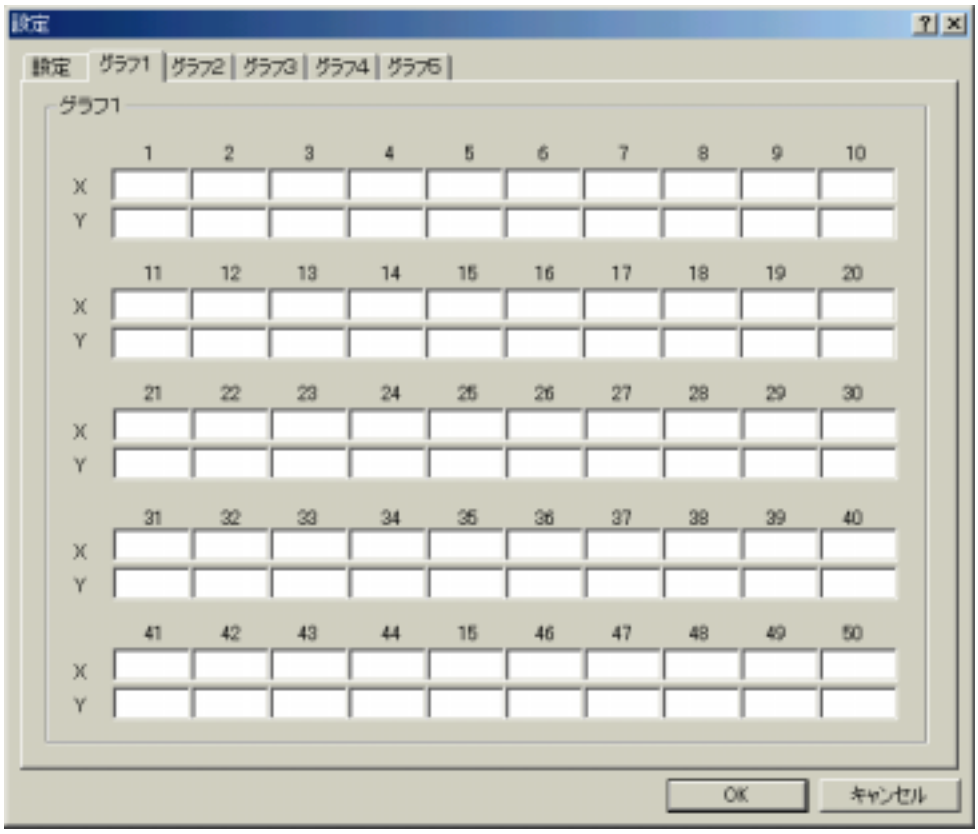
設定
設定
グラフ1
グラフ2
グラフ3
グラフ4
グラフ5
グラフ1
1 2 3 4 5 6 7 8 9 10
X
Y
11 12 13 14 15 16 17 18 19 20
X
Y
21 22 23 24 25 26 27 28 29 30
X
Y
31 32 33 34 35 36 37 38 39 40
X
Y
41 42 43 44 15 46 47 48 49 50
X
Y
OK
キャンセル

図 10

第６章　結論

本研究では科学技術の分野での実験データや理論計算などに使用するグラフ作成ソフトを開発した。

システム開発で最も一般的なC言語を使用し、現在、企業や研究機関、一般家庭を含めて広く使用されているパソコンのOSであるWindows上で動作するソフトウェアを開発するためにWindowsプログラミングの学習をした。そして、科学技術の分野でグラフを描く場合に必要な補間法などの数値計算法を学習することによりグラフソフトのVer.1の完成に至った。

しかし、市販されているソフトやフリーソフトなどと本グラフソフトを比較すると操作性や機能の面で及ばない。今後の課題として

- 「対数グラフ化」
- 「コメントの挿入」
- 「エラーバーの挿入」
- 「その他の補間法の追加」
- 「txtファイルからの数値の読み込み」

などが挙げられる。これらの機能を随時付加することにより本グラフ作成ソフトをより完成なものに近づけてゆきたい。

参考文献

１．　定本 明解 C 言語入門
柴田　望洋　著
ソフトバンクパブリッシング

２．Win32 API オフィシャルリファレンス　改訂版
マイクロソフト株式会社　編集　アスキー書籍編集部　編
株式会社アスキー　１９９９年４月１日　第１版第７刷

３．C／C++プログラマのための Windows95 プログラミング
ハーバート・シルト　著　柏原　正三　訳・監修
株式会社翔泳社　１９９８年５月３０日　初版第８刷

４．Fortran 95, C & JAVA による新数値計算法
小国　力　著
株式会社サイエンス社　１９９７年１２月２５日　初版

５．改訂 C 言語によるはじめてのアルゴリズム入門
川西　朝雄　著
株式会社技術評論社　２００１年７月　初版

謝辞

本稿を書くにあたって、指導教員である原央先生と武田光由先生から数々の指導・助言を頂いたことを深く感謝し、また論文を書くにあたり参考にさせて頂いた著者の方々に対しても感謝の意を表したい。

付録　プログラムソース

※ はページ数の都合から同じ処理は省略した。

graph.h

```
LRESULT CALLBACK WinProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg1Proc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);
LRESULT CALLBACK MyDlg2Proc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);
LRESULT CALLBACK MyDlg3Proc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);
LRESULT CALLBACK MyDlg4Proc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);
LRESULT CALLBACK MyDlg5Proc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);
LRESULT CALLBACK MyDlg0Proc(HWND hWnd, UINT msg, WPARAM wp, LPARAM lp);

#define N 999

#ifndef INSTANCE
extern
#endif
int nx, ny;

#ifndef INSTANCE
extern
#endif
long arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9, arg10, arg11, arg12, arg13, arg14;

#ifndef INSTANCE
extern
#endif
int Xmax, Xmin, Ymax, Ymin, width_x, width_y, volume;

#ifndef INSTANCE
extern
#endif
char X_title[256], Y_title[256], XPOS[256], YPOS[256];
```

```
#ifndef INSTANCE
extern
#endif
long double px[5][50], py[5][50], Xpos, Ypos;

HANDLE hInst;

HPEN hBlackPen, hRedPen, hBluePen, hGreenPen, hYellowPen, hGreyPen, hWhitePen;
HWND hParent;
HFONT hFont1, hFont2

#ifndef INSTANCE
extern
#endif
char xa1[10], …,  xa50[10],
    ya1[10], …,  ya50[10],
    xb1[10], …,  xb50[10],
    yb1[10], …,  yb50[10],
    xc1[10], …,  xc50[10],
    yc1[10], …,  yc50[10],
    xd1[10], …,  xd50[10],
    yd1[10], …,  yd50[10],
    xe1[10], …,  xe50[10],
    ye1[10], …,  ye50[10];
```

※

Setup.h

```
/*セクション*/
char strSection1[] = "設定", strSection2[] = "グラフ１", strSection3[] = "グラフ２",
      strSection4[] = "グラフ３", strSection5[] = "グラフ４", strSection6[] = "グラフ５";
/*設定*/
char strKey1[] = "GraphType", strKey2[] = "Magnification", strKey3[] = "PlotType",
      strKey4[] = "Color", strKey5[] = "CenterX", strKey6[] = "CenterY", strKey7[] = "XTitle",
      strKey8[] = "YTitle", strKey9[] = "Xmemori", strKey10[] = "Ymemori", strKey11[] = "graphvol",
```

```
      strKey12[] = "graph1", strKey13[] = "graph2", strKey14[] = "graph3", strKey15[] = "graph4",
      strKey16[] = "graph5", strKey17[] = "Xmag", strKey18[] = "Ymag";
/*グラフ１～５*/
char strxa1[] = "Xa1", … strxa50[] = "Xa50";
char strya1[] = "Ya1", … strya50[] = "Ya50";
char strxb1[] = "Xb1", … strxb50[] = "Xb50";
char stryb1[] = "Yb1", … stryb50[] = "Yb50";
char strxc1[] = "Xc1", … strxc50[] = "Xc50";
char stryc1[] = "Yc1", … stryc50[] = "Yc50";
char strxd1[] = "Xd1", … strxd50[] = "Xd50";
char stryd1[] = "Yd1", … stryd50[] = "Yd50";
char strxe1[] = "Xe1", … strxe50[] = "Xe50";
char strye1[] = "Ye1", … strye50[] = "Ye50";
/*設定*/
char strData1[256], … strData18[256];
char strxaData1[256], … strxaData50[256];
char stryaData1[256], … stryaData50[256];
char strxbData1[256], … strxbData50[256];
char strybData1[256], … strybData50[256];
char strxcData1[256], … strxcData50[256];
char strycData1[256], … strycData50[256];
char strxdData1[256], … strxdData50[256];
char strydData1[256], … strxeData50[256];
char stryeData1[256], … stryeData50[256];
```

※

※

resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by riso.rc
//
#define IDC_COMBO1                      1000
        ⋮                               ⋮
#define IDC_COMBO14                     1013
```

※

```
#define IDC_EDIT1                1021
        ⋮                        ⋮
#define IDC_EDIT11               1031
#define IDC_xa1                  1100
        ⋮                        ⋮
#define IDC_xa50                 1149
#define IDC_ya1                  1200
        ⋮                        ⋮
#define IDC_ya50                 1249
#define IDC_xb1                  1300
        ⋮                        ⋮
#define IDC_xb50                 1349
#define IDC_yb1                  1400
        ⋮                        ⋮
#define IDC_yb50                 1449
#define IDC_xc1                  1500
        ⋮                        ⋮
#define IDC_xc50                 1549
#define IDC_yc1                  1600
        ⋮                        ⋮
#define IDC_yc50                 1649
#define IDC_xd1                  1700
        ⋮                        ⋮
#define IDC_xd50                 1749
#define IDC_yd1                  1800
        ⋮                        ⋮
#define IDC_yd50                 1849
#define IDC_xe1                  1900
        ⋮                        ⋮
#define IDC_xe1                  1949
#define IDC_ye1                  2000
        ⋮                        ⋮
#define IDC_ye50                 2049
#define GRAPH                   40001
#define DEL                     40002
#define VERSION                 40003
```

※

```
#define SAVEBMP                         40004
#define NEWFILE                         40006
#define FILEOPEN                        40007
#define SAVENAMEFILE                    40009
#define SAVEFILE                        40010
#define SETPRINTER                      40011
#define PRINT                           40012
#define EXIT                            40013
#define SETGRAPH1                       40016
#define SETGRAPH2                       40017
#define SETGRAPH3                       40018
#define SETGRAPH4                       40019
#define SETGRAPH5                       40020

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        125
#define _APS_NEXT_COMMAND_VALUE         40021
#define _APS_NEXT_CONTROL_VALUE         1269
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

ceatememdc.c

```
#include <windows.h>
#include "resource.h"


void CreateMemdc(HWND hWnd, HDC memdc, HBITMAP *hBit,int fnObject)
{

    HWND hDesk;
    RECT ScrnSize;
    HDC hdc;
```

```
    HBRUSH hBrush,hDefBrush;
    hDesk = GetDesktopWindow();
    GetClientRect(hWnd, &ScrnSize);

    hdc = GetDC(hWnd);

    *memdc = CreateCompatibleDC(hdc);
    *hBit = CreateCompatibleBitmap(hdc,ScrnSize.right,ScrnSize.bottom);

    SelectObject(*memdc, *hBit);
    hBrush = (HBRUSH)GetStockObject(fnObject);
    hDefBrush = SelectObject(*memdc,hBrush);
    PatBlt(*memdc,0,0,ScrnSize.right,ScrnSize.bottom,PATCOPY);
    SelectObject(*memdc,hDefBrush);
    DeleteObject(hBrush);
    ReleaseDC(hWnd,hdc);

    return;

}
```

dlgproc.c

```
#include <windows.h>
#include "graph.h"
#include "resource.h"

LRESULT CALLBACK MyDlgOProc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;
    BOOL Error;
    int i;
    LPCTSTR fuc[] = {"なし", "３次スプライン補間", "ラグランジュ補間",
                     "線形最小２乗法", "折れ線", "棒グラフ"};
    LPCTSTR percent[] = {"300%", "250%", "200%", "150%", "100%", "75%", "50%"};
```

```
LPCTSTR percent2[] = {"100%", "90%", "80%", "70%", "60%", "50%", "40%",
                          "30%", "20%", "10%"};
LPCTSTR mesure[] = {"なし", "1", "5", "10", "50", "100", "500", "1000"};
LPCTSTR color[] = {"黒色", "赤色", "青色", "緑色", "黄色"};
LPCTSTR line[] = {"なし", "実線", "点線", "破線", "一点破線", "二点破線"};
LPCTSTR plot[] = {" なし","●円形", "▲三角形", "▼逆三角形", "◆菱形", "■四角形"};
LPCTSTR gvolume[] = {"0", "1", "2", "3", "4", "5"};
LPCTSTR pvolume[] = {"0", … ,"50"};

switch(msg) {
case WM_INITDIALOG:
      for (i = 0; i < 5; i++) {
          SendDlgItemMessage(hDlg,IDC_COMBO4, CB_INSERTSTRING, (WPARAM)i, (LPARAM)color[i]);
      }

      for (i = 0; i < 6; i++) {
          SendDlgItemMessage(hDlg, IDC_COMBO1, CB_INSERTSTRING, (WPARAM)i, (LPARAM)fuc[i]);
           SendDlgItemMessage(hDlg,IDC_COMBO3,CB_INSERTSTRING,(WPARAM)i,(LPARAM)plot[i]);
            SendDlgItemMessage(hDlg,IDC_COMBO7,CB_INSERTSTRING,(WPARAM)i,
                              (LPARAM)gvolume[i]);
      }

      for (i = 0; i < 7; i++) {
          SendDlgItemMessage(hDlg,IDC_COMBO2,CB_INSERTSTRING,(WPARAM)i,
                              (LPARAM)percent[i]);
      }

      for (i = 0; i < 8; i++) {
          SendDlgItemMessage(hDlg,IDC_COMBO5,CB_INSERTSTRING,(WPARAM)i,
                             (LPARAM)mesure[i]);
          SendDlgItemMessage(hDlg,IDC_COMBO6,CB_INSERTSTRING,(WPARAM)i,
                             (LPARAM)mesure[i]);
      }
```

```
for (i = 0; i < 10; i++) {
    SendDlgItemMessage(hDlg,IDC_COMBO13,CB_INSERTSTRING, (WPARAM)i,
                    (LPARAM)percent2[i]);
    SendDlgItemMessage(hDlg,IDC_COMBO14,CB_INSERTSTRING,(WPARAM)i,
                     (LPARAM)percent2[i]);
}

for (i = 0; i < 51; i++) {
    SendDlgItemMessage(hDlg,IDC_COMBO8,CB_INSERTSTRING,(WPARAM)i,
                     (LPARAM)pvolume[i]);
    SendDlgItemMessage(hDlg,IDC_COMBO9,CB_INSERTSTRING,(WPARAM)i,
                     (LPARAM)pvolume[i]);
    SendDlgItemMessage(hDlg,IDC_COMBO10,CB_INSERTSTRING,(WPARAM)i,
                      (LPARAM)pvolume[i]);
    SendDlgItemMessage(hDlg,IDC_COMBO11,CB_INSERTSTRING,(WPARAM)i,
                       (LPARAM)pvolume[i]);
    SendDlgItemMessage(hDlg,IDC_COMBO12,CB_INSERTSTRING,(WPARAM)i,
                     (LPARAM)pvolume[i]);
}

  /*中心座標*/
SetDlgItemText(hDlg,IDC_EDIT1,(LPTSTR)XPOS);
SetDlgItemText(hDlg,IDC_EDIT2,(LPTSTR)YPOS);

/*軸名称*/
SetDlgItemText(hDlg,IDC_EDIT3,(LPCTSTR)X_title);
SetDlgItemText(hDlg,IDC_EDIT4,(LPCTSTR)Y_title);

/*コンボボックス*/
SendDlgItemMessage(hDlg,IDC_COMBO1,CB_SETCURSEL,(WPARAM)arg1,0L);   //グラフ
SendDlgItemMessage(hDlg,IDC_COMBO2,CB_SETCURSEL,(WPARAM)arg2,0L);   //倍率
SendDlgItemMessage(hDlg,IDC_COMBO3,CB_SETCURSEL,(WPARAM)arg3,0L);   //点
SendDlgItemMessage(hDlg,IDC_COMBO4,CB_SETCURSEL,(WPARAM)arg4,0L);   //色
SendDlgItemMessage(hDlg,IDC_COMBO5,CB_SETCURSEL,(WPARAM)arg5,0L);   //X メモリ
SendDlgItemMessage(hDlg,IDC_COMBO6,CB_SETCURSEL,(WPARAM)arg6,0L);   //Y メモリ
```

```
  SendDlgItemMessage(hDlg,IDC_COMBO7,CB_SETCURSEL,(WPARAM)arg7,0L);   //グラフ数
  SendDlgItemMessage(hDlg,IDC_COMBO8,CB_SETCURSEL,(WPARAM)arg8,0L);   //グラフ１
  SendDlgItemMessage(hDlg,IDC_COMBO9,CB_SETCURSEL,(WPARAM)arg9,0L);   //グラフ２
  SendDlgItemMessage(hDlg,IDC_COMBO10,CB_SETCURSEL,(WPARAM)arg10,0L);   //グラフ３
  SendDlgItemMessage(hDlg,IDC_COMBO11,CB_SETCURSEL,(WPARAM)arg11,0L);   //グラフ４
  SendDlgItemMessage(hDlg,IDC_COMBO12,CB_SETCURSEL,(WPARAM)arg12,0L);   //グラフ５
  SendDlgItemMessage(hDlg,IDC_COMBO13,CB_SETCURSEL,(WPARAM)arg13,0L);
  SendDlgItemMessage(hDlg,IDC_COMBO14,CB_SETCURSEL,(WPARAM)arg14,0L);


  return TRUE;
case WM_NOTIFY:
    nmhdr = (NMHDR *)lp;
    switch(nmhdr->code) {
        case PSN_APPLY:
            arg1 = SendDlgItemMessage(hDlg, IDC_COMBO1, CB_GETCURSEL, 0L, 0L);
            arg2 = SendDlgItemMessage(hDlg, IDC_COMBO2, CB_GETCURSEL, 0L, 0L);
            arg3 = SendDlgItemMessage(hDlg, IDC_COMBO3, CB_GETCURSEL, 0L, 0L);
            arg4 = SendDlgItemMessage(hDlg, IDC_COMBO4, CB_GETCURSEL, 0L, 0L);
            arg5 = SendDlgItemMessage(hDlg, IDC_COMBO5, CB_GETCURSEL, 0L, 0L);
            arg6 = SendDlgItemMessage(hDlg, IDC_COMBO6, CB_GETCURSEL, 0L, 0L);
            arg7 = SendDlgItemMessage(hDlg, IDC_COMBO7, CB_GETCURSEL, 0L, 0L);
            arg8 = SendDlgItemMessage(hDlg, IDC_COMBO8, CB_GETCURSEL, 0L, 0L);
            arg9 = SendDlgItemMessage(hDlg, IDC_COMBO9, CB_GETCURSEL, 0L, 0L);
            arg10 = SendDlgItemMessage(hDlg, IDC_COMBO10, CB_GETCURSEL, 0L, 0L);
            arg11 = SendDlgItemMessage(hDlg, IDC_COMBO11, CB_GETCURSEL, 0L, 0L);
            arg12 = SendDlgItemMessage(hDlg, IDC_COMBO12, CB_GETCURSEL, 0L, 0L);
            arg13 = SendDlgItemMessage(hDlg, IDC_COMBO13, CB_GETCURSEL, 0L, 0L);
            arg14 = SendDlgItemMessage(hDlg, IDC_COMBO14, CB_GETCURSEL, 0L, 0L);
            width_x = 421 - strlen(X_title) * 4;//746 421
            width_y = 275 + strlen(Y_title) * 4;//516 275
            /*中心座標*/
            GetDlgItemText(hDlg, IDC_EDIT1, (LPTSTR)XPOS,255);
            GetDlgItemText(hDlg, IDC_EDIT2, (LPTSTR)YPOS,255);
            Xpos= atof(XPOS);
            Ypos = atof(YPOS);
```

```
                    /*軸名称*/
                    GetDlgItemText(hDlg,IDC_EDIT3,(LPTSTR)X_title,255);
                    GetDlgItemText(hDlg,IDC_EDIT4,(LPTSTR)Y_title,255);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;

                default:
                    return FALSE;
            }
            break;

    }
    return FALSE;
}

LRESULT CALLBACK MyDlg1Proc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;

    switch(msg) {
        case WM_INITDIALOG:
            SetDlgItemText(hDlg,IDC_xa1,(LPCTSTR)xa1);
                                ⋮
            SetDlgItemText(hDlg,IDC_xa50,(LPCTSTR)xa50);
            SetDlgItemText(hDlg,IDC_ya1,(LPCTSTR)ya1);
                                ⋮
            SetDlgItemText(hDlg,IDC_ya50,(LPCTSTR)ya50);
            return TRUE;
        case WM_NOTIFY:
            nmhdr = (NMHDR *)lp;
            switch(nmhdr->code) {
                case PSN_APPLY:
                   GetDlgItemText(hDlg, IDC_xa1, (LPTSTR)xa1, 10);
                                ⋮
                    GetDlgItemText(hDlg, IDC_xa1, (LPTSTR)xa1, 10);
```

※

※

```
                    GetDlgItemText(hDlg, IDC_xa1, (LPTSTR)xa1, 10);
                                        ⋮
                    GetDlgItemText(hDlg, IDC_xa50, (LPTSTR)xa50, 10);
                    GetDlgItemText(hDlg, IDC_ya1, (LPTSTR)ya1, 10);
                                        ⋮
                    GetDlgItemText(hDlg, IDC_ya50, (LPTSTR)ya50, 10);        ※

                    px[0][0]  = atof(xa1);    py[0][0]  = atof(ya1);
                              ⋮                         ⋮
                    px[0][49]  = atof(xa50);  py[0][49]  = atof(ya50);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;
                default:
                     return FALSE;
            }
            break;
    }

    return FALSE;
}

LRESULT CALLBACK MyDlg1Proc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;

    switch(msg) {
        case WM_INITDIALOG:
            SetDlgItemText(hDlg,IDC_xa1,(LPCTSTR)xa1);
                                        ⋮
            SetDlgItemText(hDlg,IDC_xa50,(LPCTSTR)xa50);
            SetDlgItemText(hDlg,IDC_ya1,(LPCTSTR)ya1);                       ※
                                        ⋮
            SetDlgItemText(hDlg,IDC_ya50,(LPCTSTR)ya50);
```

```
            return TRUE;
        case WM_NOTIFY:
            nmhdr = (NMHDR *)lp;
            switch(nmhdr->code) {
                case PSN_APPLY:
                    GetDlgItemText(hDlg, IDC_xa1, (LPTSTR)xa1, 10);
                                        ⋮
                    GetDlgItemText(hDlg, IDC_xa50, (LPTSTR)xa50, 10);
                    GetDlgItemText(hDlg, IDC_ya1, (LPTSTR)ya1, 10);
                                        ⋮
                    GetDlgItemText(hDlg, IDC_ya50, (LPTSTR)ya50, 10);      ※

                    px[0][0]  = atof(xa1);    py[0][0]  = atof(ya1);
                              ⋮                         ⋮
                    px[0][49]  = atof(xa50);  py[0][49]  = atof(ya50);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;
                default:
                     return FALSE;
            }
            break;
    }

    return FALSE;
}

LRESULT CALLBACK MyDlg2Proc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;

    switch(msg) {
        case WM_INITDIALOG:
            SetDlgItemText(hDlg,IDC_xb1,(LPCTSTR)xb1);
                                    ⋮                          ※
            SetDlgItemText(hDlg,IDC_xb50,(LPCTSTR)xb50);
```

```
            SetDlgItemText(hDlg,IDC_yb1,(LPCTSTR)yb1);
                                ⋮
            SetDlgItemText(hDlg,IDC_yb50,(LPCTSTR)yb50);
        return TRUE;
        case WM_NOTIFY:
            nmhdr = (NMHDR *)lp;
            switch(nmhdr->code) {
                case PSN_APPLY:
                    GetDlgItemText(hDlg, IDC_xb1, (LPTSTR)xb1, 10);
                                ⋮
                    GetDlgItemText(hDlg, IDC_xb50, (LPTSTR)xb50, 10);
                    GetDlgItemText(hDlg, IDC_yb1, (LPTSTR)yb1, 10);
                                ⋮                                    ※
                    GetDlgItemText(hDlg, IDC_yb50, (LPTSTR)yb50, 10);

                    px[1][0]  = atof(xb1);    py[1][0]  = atof(yb1);
                          ⋮                         ⋮
                    px[1][49]  = atof(xb50);  py[1][49]  = atof(yb50);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;
                default:
                     return FALSE;
            }
            break;
    }

    return FALSE;
}

LRESULT CALLBACK MyDlg3Proc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;

    switch(msg) {
        case WM_INITDIALOG:
```

```
            SetDlgItemText(hDlg,IDC_xc1,(LPCTSTR)xc1);

            SetDlgItemText(hDlg,IDC_xc50,(LPCTSTR)xc50);
            SetDlgItemText(hDlg,IDC_yc1,(LPCTSTR)yc1);

            SetDlgItemText(hDlg,IDC_yc50,(LPCTSTR)yc50);
        return TRUE;
        case WM_NOTIFY:
            nmhdr = (NMHDR *)lp;
            switch(nmhdr->code) {
                case PSN_APPLY:
                     GetDlgItemText(hDlg, IDC_xc1, (LPTSTR)xc1, 10);

                    GetDlgItemText(hDlg, IDC_xc50, (LPTSTR)xc50, 10);
                    GetDlgItemText(hDlg, IDC_yc1, (LPTSTR)yc1, 10);

                    GetDlgItemText(hDlg, IDC_yc50, (LPTSTR)yc50, 10);

                    px[2][0]  = atof(xc1);    py[2][0]  = atof(yc1);

                    px[2][49]  = atof(xc50);  py[2][49]  = atof(yc50);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;
                default:
                     return FALSE;
            }
            break;
    }

    return FALSE;
}

LRESULT CALLBACK MyDlg4Proc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;
```

```
    switch(msg) {
        case WM_INITDIALOG:
            SetDlgItemText(hDlg,IDC_xd1,(LPCTSTR)xd1);

            SetDlgItemText(hDlg,IDC_xd50,(LPCTSTR)xd50);
            SetDlgItemText(hDlg,IDC_yd1,(LPCTSTR)yd1);

            SetDlgItemText(hDlg,IDC_yd50,(LPCTSTR)yd50);
        return TRUE;
        case WM_NOTIFY:
            nmhdr = (NMHDR *)lp;
            switch(nmhdr->code) {
                case PSN_APPLY:
                     GetDlgItemText(hDlg, IDC_xd1, (LPTSTR)xd1, 10);

                    GetDlgItemText(hDlg, IDC_xd50, (LPTSTR)xd50, 10);
                    GetDlgItemText(hDlg, IDC_yd1, (LPTSTR)yd1, 10);

                    GetDlgItemText(hDlg, IDC_yd50, (LPTSTR)yd50, 10);

                    px[3][0]  = atof(xd1);    py[3][0]  = atof(yd1);

                    px[3][49]  = atof(xd50);  py[3][49]  = atof(yd50);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;
                default:
                     return FALSE;
            }
            break;
    }

    return FALSE;
}
```

```
 LRESULT CALLBACK MyDlg5Proc(HWND hDlg, UINT msg, WPARAM wp, LPARAM lp)
{
    NMHDR *nmhdr;

    switch(msg) {
        case WM_INITDIALOG:
            SetDlgItemText(hDlg,IDC_xe1,(LPCTSTR)xe1);

            SetDlgItemText(hDlg,IDC_xe50,(LPCTSTR)xe50);
            SetDlgItemText(hDlg,IDC_ye1,(LPCTSTR)ye1);

            SetDlgItemText(hDlg,IDC_ye50,(LPCTSTR)ye50);
        return TRUE;
        case WM_NOTIFY:
            nmhdr = (NMHDR *)lp;
            switch(nmhdr->code) {
                case PSN_APPLY:
                     GetDlgItemText(hDlg, IDC_xe1, (LPTSTR)xe1, 10);

                    GetDlgItemText(hDlg, IDC_xe50, (LPTSTR)xe50, 10);
                    GetDlgItemText(hDlg, IDC_ye1, (LPTSTR)ye1, 10);

                    GetDlgItemText(hDlg, IDC_ye50, (LPTSTR)ye50, 10);

                    px[4][0]  = atof(xe1);    py[4][0]  = atof(ye1);

                    px[4][49]  = atof(xe50);  py[4][49]  = atof(ye50);

                     InvalidateRect(hParent, NULL, TRUE);
                     return TRUE;
                default:
                     return FALSE;
            }
            break;
    }
```

```
      }

    return FALSE;
}

void CreateMyProp(HWND hWnd)
{
    HINSTANCE hInst;
    PROPSHEETPAGE psp;
    PROPSHEETHEADER psh;
    HPROPSHEETPAGE hpsp[6];

    hInst = (HINSTANCE)GetWindowLong(hWnd, GWL_HINSTANCE);

    psp.dwSize = sizeof(PROPSHEETPAGE);
    psp.dwFlags = PSP_DEFAULT;
    psp.hInstance = hInst;

    psp.pszTemplate = "PAGE0";
    psp.pfnDlgProc = (DLGPROC)MyDlg0Proc;
    hpsp[0] = CreatePropertySheetPage(&psp);

    psp.pszTemplate = "PAGE1";
    psp.pfnDlgProc = (DLGPROC)MyDlg1Proc;
    hpsp[1] = CreatePropertySheetPage(&psp);

    psp.pszTemplate = "PAGE2";
    psp.pfnDlgProc = (DLGPROC)MyDlg2Proc;
    hpsp[2] = CreatePropertySheetPage(&psp);

    psp.pszTemplate = "PAGE3";
    psp.pfnDlgProc = (DLGPROC)MyDlg3Proc;
    hpsp[3] = CreatePropertySheetPage(&psp);

    psp.pszTemplate = "PAGE4";
    psp.pfnDlgProc = (DLGPROC)MyDlg4Proc;.
```

```
    hpsp[4] = CreatePropertySheetPage(&psp);

    psp.pszTemplate = "PAGE5";
    psp.pfnDlgProc = (DLGPROC)MyDlg5Proc;
    hpsp[5] = CreatePropertySheetPage(&psp);

    memset(&psh, 0, sizeof(PROPSHEETHEADER));
    psh.dwSize = sizeof(PROPSHEETHEADER);
    psh.dwFlags = PSH_NOAPPLYNOW;
    psh.hInstance = hInst;
    psh.hwndParent = hWnd;
    psh.nPages = 6;
    psh.phpage = hpsp;
    psh.pszCaption = "設定";
    (HWND)PropertySheet(&psh);

}
```

interpolate.c

```
#include <windows.h>
#include <math.h>
#include "graph.h"
#include "resource.h"

double ymin(double rm, double r);
double spline(double *, double *, int *, double, int);
double lagrange(double *, double *, int *, double, int);
double lsq1(double *, double *, int *, int);
double lsq2(double *, double *, int *, int);

void plot(HDC, double *, double *, int *, long, double, double, double, double, double, double,
int, int);

HPEN hPen;
HBRUSH hBrush;
```

```
void interpolate(HDC memdc, double px[][50], double py[][50], double Xpos, double Ypos,
long ans1, long ans2, long ans3,long ans4, long ans5, long ans6, long ans7, long ans8,
long ans9, long ans10, long ans11, long ans12, long ans13, long ans14,int z)
{
          RECT re1 = {0,0,800,600},
               re2 = {0,0,794,35},
               re3 = {0,35,100,556},
               re4 = {741,35,794,556},
               re5 = {100,516,794,556},
               re6 = {101, 35, 740, 516};

          int i, j, k, Xaxis, Yaxis, xpos, ypos, judge;
          long h;
          int NUM[5];
          long double t, Ry, Rx, MAGx, MAGy, xaxis, yaxis, xax, yax, numx, numy;
          long double x[5][1000], y[5][1000];
          double M[5], a[5], b[5];
         double mag[] = {3, 2.5, 2, 1.5, 1, 0.75, 0.5};
          double mag2[] = {0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.01};
          int figure = ans7;

          Rectangle(memdc, 100*z, 35*z, 741*z, 516*z);
          NUM[0] = ans8;
          NUM[1] = ans9;
          NUM[2] = ans10;
          NUM[3] = ans11;
          NUM[4] = ans12;

          for (i = 0; i < figure; i++) {
              M[i] = fabs(px[i][NUM[i]-1] - px[i][0]) / (N+1);
          }

          if (ans1 == 1) {
               for (i = 0; i < figure; i++) {
                    for (t = px[i][0], k = 0; t <= px[i][NUM[i]-1], k <= N; t += M[i], k++) {
                         x[i][k] = t;
```

```
                y[i][k] = spline(px, py, NUM, t, i);
            }
        }
        judge = N;
    }

    else if (ans1 == 2) {
        for (i = 0; i < figure; i++) {
            for (t = px[i][0], k = 0; t <= px[i][NUM[i]-1], k <= N; t += M[i], k++) {
                x[i][k] = t;
                y[i][k] = lagrange(px, py, NUM, t, i);
            }
        }
        judge = N;
    }

    else if (ans1 == 3) {
        for (i = 0; i < figure; i++) {
            a[i] = lsq1(px, py, NUM, i);
            b[i] = lsq2(px, py, NUM, i);
        }
        for (i = 0; i < figure; i++) {
            x[i][0] = px[i][0];
            y[i][0] = a[i] * x[i][0] + b[i];
        }
        for (i = 0; i < figure; i++) {
            x[i][1] = px[i][NUM[i]-1];
            y[i][1] = a[i] * x[i][1] + b[i];
        }
        judge = 1;
    }

    if (ans1 == 4) {
        Ry = 0;
        for (i = 0; i < figure; i++) {
            for (j = 0; j < NUM[i]; j++) {
```

```
            if (Ry < fabs(py[i][j]))
                Ry = py[i][j];
        }
    }
    Rx = 0;
        for (i = 0; i < figure; i++) {
            for (j = 0; j < 2; j++) {
               if (Rx < fabs(px[i][j*NUM[i]-j]))
                   Rx = px[i][j*NUM[i]-j];
            }
        }
}
else if (ans1 == 3) {
    Ry = 0;
     for (i = 0; i < figure; i++) {
         for (j = 0; j < NUM[i]; j++) {
             if (Ry < fabs(py[i][j]))
                 Ry = py[i][j];
         }
     }
     Rx = 0;
     for (i = 0; i < figure; i++) {
         for (j = 0; j < 2; j++) {
             if (Rx < fabs(px[i][j*NUM[i]-j]))
                 Rx = px[i][j*NUM[i]-j];
         }
      }
  }
  else {
      Ry = 0;
       for (i = 0; i < figure; i++) {
           for (j = 0; j <= N; j++) {
               if (Ry < fabs(y[i][j]))
                Ry = y[i][j];
           }
       }
```

```
        Rx = 0;
            for (i = 0; i < figure; i++) {
                for (j = 0; j < 2; j++) {
                    if (Rx < fabs(px[i][j*NUM[i]-j]))
                        Rx = px[i][j*NUM[i]-j];
                }
            }
    }
    MAGx = mag2[ans13] * mag[ans2];
    MAGy = mag2[ans14] * mag[ans2];

    xaxis = 320 / fabs(Rx);
    yaxis = 240 / fabs(Ry);

    xpos = (int)(-Xpos*xaxis*MAGx + 420);
    ypos = (int)(Ypos*yaxis*MAGy + 275);

    Xaxis = (int)(320 / (xaxis * MAGx));
    Yaxis = (int)(240 / (yaxis * MAGy));

    for (i = 0; i < figure; i++) {
        SelectObject(memdc, hBlackPen);
        plot(memdc, px, py, NUM, ans3, xaxis, yaxis, MAGx, MAGy, Xpos, Ypos, i, z);
        if ((ans4+i == 0) || (ans4+i == 5))
            hPen = hBlackPen;
        else if ((ans4+i == 1) || (ans4+i == 6))
            hPen = hRedPen;
        else if ((ans4+i == 2) || (ans4+i == 7))
            hPen = hBluePen;
        else if ((ans4+i == 3) || (ans4+i == 8))
            hPen = hGreenPen;
        else
            hPen = hYellowPen;

        SelectObject(memdc, hPen);
```

```
    if ((ans1 == 1) || (ans1 == 2)) {
        for (j = 0; j < judge; j++) {
            MoveToEx(memdc, (int)(x[i][j]*MAGx*xaxis + xpos)*z,
                            (int)(-y[i][j]*MAGy*yaxis+ypos)*z, NULL);
             LineTo(memdc, (int)(x[i][j+1]*MAGx*xaxis + xpos)*z,
                            (int)(-y[i][j+1]*MAGy*yaxis+ypos)*z);
        }
     }
     else if (ans1 == 3) {
         MoveToEx(memdc, (int)(x[i][0]*MAGx*xaxis + xpos)*z,
                        (int)(-y[i][0]*MAGy*yaxis+ypos)*z, NULL);
          LineTo(memdc, (int)(x[i][1]*MAGx*xaxis + xpos)*z,
                        (int)(-y[i][1]*MAGy*yaxis+ypos)*z);
      }




      else if (ans1 == 4) {
           for (j = 0; j < NUM[i]-1; j++) {
               MoveToEx(memdc, (int)(px[i][j]*MAGx*xaxis + xpos)*z,
                               (int)(-py[i][j]*MAGy*yaxis+ypos)*z, NULL);
                LineTo(memdc, (int)(px[i][j+1]*MAGx*xaxis + xpos)*z,
                               (int)(-py[i][j+1]*MAGy*yaxis+ypos)*z);
            }
       }
}

hPen = hBlackPen;
SelectObject(memdc, hPen);

MoveToEx(memdc, 100*z, ypos*z, NULL);     //X 軸(原点)
 LineTo(memdc, 741*z, ypos*z);
MoveToEx(memdc, xpos*z, 35*z, NULL);      //Y 軸(原点)
 LineTo(memdc, xpos*z, 516*z);
```

```
}

        for (h = ypos; h >= 35; h = h - (int)(yax+0.5)) {
             MoveToEx(memdc, (int)(xpos-1)*z, (int)(h)*z, NULL);
             LineTo(memdc, (int)(xpos+2)*z, (int)(h)*z);
        }
        for (h = ypos; h <= 516; h = h + (int)(yax*sumy+0.5)) {
            MoveToEx(memdc, (int)(xpos-1)*z, (int)(h)*z, NULL);
             LineTo(memdc, (int)(xpos+2)*z, (int)(h)*z);
         }

          for (h = xpos; h >= 100; h = h - (int)(xax+0.5)) {
               MoveToEx(memdc, (int)(h)*z, (int)(ypos-1)*z, NULL);
               LineTo(memdc, (int)(h)*z, (int)(ypos+2)*z);
           }
          for (h = xpos; h <= 741; h = h + (int)(xax+0.5)) {
               MoveToEx(memdc, (int)(h)*z, (int)(ypos-1)*z, NULL);
               LineTo(memdc, (int)(h)*z, (int)(ypos+2)*z);
          }

double ymin(double rm, double r)
{
    if (r * rm > 240)
        return (240 / (r * rm));
    else
        return (r*rm/240);
}
```

plot.c

```
#include <windows.h>

void plot(HDC memdc, double prox[][50], double proy[][50], int num[],
long pro, double xax, double yax, double Mx, double My, double xpo, double ypo, int p, int z)
{
```

```
int i;


if (pro != 0) {
    if ((pro + p == 1) || (pro + p == 6)) {
        for (i = 0; i < num[p]; i++)
            Ellipse(memdc,(int)(prox[p][i]*Mx*xax+417-xpo*Mx*xax)*z,
                        (int)(-proy[p][i]*My*yax+272+ypo*My*yax)*z,
                        (int)(prox[p][i]*Mx*xax+423-xpo*Mx*xax)*z,
                        (int)(-proy[p][i]*My*yax+278+ypo*My*yax)*z);
    }
    else if ((pro + p == 2) || (pro + p == 7)) {
        for (i = 0; i < num[p]; i++) {
            MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z, NULL);
            LineTo(memdc, (int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                          (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z);
            MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                          (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z, NULL);
            LineTo(memdc, (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                           (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z);
            MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                           (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z, NULL);
            LineTo(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                           (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z);
        }
    }

    else if ((pro + p == 3) || (pro + p == 8)) {
        for (i = 0; i < num[p]; i++) {
            MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z, NULL);
            LineTo(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                          (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z);
            MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                          (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z, NULL);
```

```
        LineTo(memdc, (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                        (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z);
        MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z, NULL);
        LineTo(memdc, (int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z);
    }
}

else if ((pro + p == 4) || (pro + p == 9)) {
    for (i = 0; i < num[p]; i++) {
        MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                      (int)(-proy[p][i]*yax*My+275+ypo*yax*My)*z, NULL);
        LineTo(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                        (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z);
        MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                        (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z, NULL);
        LineTo(memdc, (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+275+ypo*yax*My)*z);
        MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+275+ypo*yax*My)*z, NULL);
        LineTo(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z);
        MoveToEx(memdc, (int)(prox[p][i]*xax*Mx+420-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z, NULL);
        LineTo(memdc, (int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+275+ypo*yax*My)*z);

    }
}

else if ((pro + p == 5) || (pro + p == 10)) {
    for (i = 0; i < num[p]; i++)
        Rectangle(memdc,(int)(prox[p][i]*xax*Mx+417-xpo*xax*Mx)*z,
                         (int)(-proy[p][i]*yax*My+272+ypo*yax*My)*z,
```

```
                                (int)(prox[p][i]*xax*Mx+423-xpo*xax*Mx)*z,
                                (int)(-proy[p][i]*yax*My+278+ypo*yax*My)*z);
        }
    }
}
```

print.c

```
#include <windows.h>
#include <math.h>
#include "graph.h"
#include "resource.h"

HDC GetPrintInfo(void);
void plot(HDC, double *, double *, int *, long, double, double, double, double, double, double,
int, int);
void interpolate(HDC, long double *, long double *, double, double, long, long,
           long,long, long, long, long, long,long, long, long, long, long, long, int);
PRINTER_INFO_5 prninfo[3];

HPEN hPen;
HBRUSH hBrush;

int MyPrint(HDC memdc)
{

    DOCINFO docinfo;

    hFont1 = CreateFont(72,0,0,0,FW_NORMAL,0,0,0,SHIFTJIS_CHARSET,
                    OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
                    DEFAULT_PITCH | FF_DONTCARE,"MS ゴシック");

    hFont2 = CreateFont(72,0,900,0,FW_NORMAL,0,0,0,SHIFTJIS_CHARSET,
                    OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
                    DEFAULT_PITCH | FF_DONTCARE,"MS ゴシック");
```

```
    memset(&docinfo, 0, sizeof(DOCINFO));
    docinfo.cbSize = sizeof(DOCINFO);
    docinfo.lpszDocName = "testprint";

    memdc = GetPrintInfo();

    StartDoc(memdc, &docinfo);
    StartPage(memdc);
    interpolate(memdc, px, py, Xpos, Ypos, arg1, arg2, arg3, arg4, arg5,
              arg6, arg7, arg8, arg9, arg10, arg11, arg12, arg13, arg14, 4);
    hPen = hWhitePen;
    SelectObject(memdc, hPen);
    Rectangle(memdc, 0, 0, 3175+1000, 139);
    Rectangle(memdc, 0, 140, 399, 2223+1000);
    Rectangle(memdc, 400, 2064, 3175, 2223+1000);
    Rectangle(memdc, 2964, 140, 3175+1000, 2063);

    SelectObject(memdc, hFont1);
    TextOut(memdc, width_x*4, 530*4, X_title, strlen(X_title));
    SelectObject(memdc, hFont2);
    TextOut(memdc, 70*4, width_y*4, Y_title, strlen(Y_title));

    EndPage(memdc);
    EndDoc(memdc);
    SetFocus(hParent);
    DeleteDC(memdc);
    return 0;
}

HDC GetPrintInfo()
{
    DWORD dwNeeded, dwReturned;
    if (EnumPrinters(PRINTER_ENUM_DEFAULT, NULL, 5, (LPBYTE)&prninfo,
        sizeof(prninfo), &dwNeeded, &dwReturned)) {
        return CreateDC(NULL, prninfo[0].pPrinterName, NULL, NULL);
    }
```

```
    return 0;
}

int PrinterSet()
{
    HANDLE hPrint;
    GetPrintInfo();
    OpenPrinter(prninfo[0].pPrinterName, &hPrint, NULL);
    PrinterProperties(hParent, hPrint);
    ClosePrinter(hPrint);
    return 0;
}
```

savebmp.c

```
#include <windows.h>
#include "resource.h"

BOOL SaveBitmapFile(HDC memdc,HBITMAP bmp,LPCTSTR lpFileName)
{

    HANDLE hFile;
    LPBYTE bits;
    HDC hdc;
    DWORD filesize,bitsize;
    BITMAPFILEHEADER fh;
    BITMAPINFO bi;
    BITMAP bm;
    HBITMAP tmpbmp,oldbmp;
    DWORD NumOfByteWritten;
    BOOL RetVal = FALSE;
    int rest;

    if(GetObject(bmp,sizeof(BITMAP),&bm) == 0)
        goto ERRHANDLER_L1;
```

```
if(bm.bmBitsPixel < 16)
    goto ERRHANDLER_L1;

bitsize = bm.bmWidthBytes;

if(rest = bitsize % 4)
bitsize += 4 - rest;

bitsize *= bm.bmHeight;
filesize = bitsize + sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER);

hFile = CreateFile(lpFileName,GENERIC_WRITE,0,NULL,CREATE_ALWAYS
                  ,FILE_ATTRIBUTE_ARCHIVE,NULL);

if(hFile == INVALID_HANDLE_VALUE)
    goto ERRHANDLER_L1;

ZeroMemory(&fh,sizeof(BITMAPFILEHEADER));
fh.bfType = 0x4d42; //BM
fh.bfSize = filesize;
fh.bfOffBits = sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER);

if(WriteFile(hFile,(LPCVOID)&fh,(DWORD)sizeof(BITMAPFILEHEADER),
            &NumOfByteWritten,NULL) == FALSE)
    goto ERRHANDLER_L2;

ZeroMemory(&bi,sizeof(BITMAPINFOHEADER));
bi.bmiHeader.biSize = sizeof(BITMAPINFOHEADER);
bi.bmiHeader.biWidth = bm.bmWidth;
bi.bmiHeader.biHeight = bm.bmHeight;
bi.bmiHeader.biPlanes = 1;
bi.bmiHeader.biBitCount = bm.bmBitsPixel;
bi.bmiHeader.biCompression = BI_RGB;
bi.bmiHeader.biSizeImage = bitsize;

if(WriteFile(hFile,(LPCVOID)&bi,(DWORD)sizeof(BITMAPINFOHEADER),
```

```
                &NumOfByteWritten,NULL) == FALSE)
        goto ERRHANDLER_L2;

    if((bits = (LPBYTE)malloc(bitsize)) == NULL)
        goto ERRHANDLER_L2;

    hdc = GetDC(NULL);
    if(memdc != NULL){
        tmpbmp = CreateCompatibleBitmap(hdc,10,10);
        oldbmp = SelectObject(memdc,tmpbmp);
    }

    if(!GetDIBits(hdc,bmp,0,bm.bmHeight,(LPVOID)bits,&bi,DIB_RGB_COLORS))
        goto ERRHANDLER_L3;

    if(WriteFile(hFile,(LPCVOID)bits,(DWORD)bitsize,&NumOfByteWritten,NULL) == FALSE)
        goto ERRHANDLER_L3;

    RetVal = TRUE;

    ERRHANDLER_L3:

    free(bits);
    ReleaseDC(NULL,hdc);

    if(memdc != NULL)
        DeleteObject(SelectObject(memdc,oldbmp));

    ERRHANDLER_L2:

    CloseHandle(hFile);

    ERRHANDLER_L1:

    return RetVal;
}
```

main.c

```
#include <windows.h>
#include <stdio.h>
#include <math.h>

#include <commctrl.h>

#define INSTANCE

#include "resource.h"
#include "graph.h"
#include "setup.h"

BOOL SaveBitmapFile(HDC,HBITMAP,LPCTSTR);
LRESULT CALLBACK WinProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg1Proc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg2Proc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg3Proc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg4Proc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg5Proc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK MyDlg0Proc(HWND, UINT, WPARAM, LPARAM);

void CreateMemdc(HWND, HDC *, HBITMAP *, int);
void OpenFiles(HWND);
void CreateMyProp(HWND);
void interpolate(HDC, long double *, long double *, double, double, long, long,
                     long,long, long, long, long, long,long, long, long, long, long, long, int);
int WriteMyFile(HWND);
int MyPrint(HDC);
int PrinterSet();
int OverWriteMyFile(HWND);
int MakeNewFile(HWND);
double ymin(double, double);

char szWinName[] = "grph01";
char FileName[MAX_PATH], FileTitle[64];
```

```
char File[64] = "無題";
int filename_type;
int show = 0;

int WINAPI WinMain(HINSTANCE hThisInst, HINSTANCE hPrevInst,
LPSTR lpszArgs, int nWinMode)
{

    HWND hwnd;
    MSG msg;
    WNDCLASSEX wcl;
    hInst = hThisInst;

    wcl.cbSize = sizeof(WNDCLASSEX);

    wcl.hInstance = hInst;
    wcl.lpszClassName = szWinName;
    wcl.lpfnWndProc = WinProc;
    wcl.style = CS_HREDRAW | CS_VREDRAW | CS_OWNDC;

    wcl.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wcl.hIconSm = LoadIcon(NULL, IDI_WINLOGO);
    wcl.hCursor = LoadCursor(NULL, IDC_ARROW);

    wcl.lpszMenuName = "MENU";
    wcl.cbClsExtra = 0;
    wcl.cbWndExtra = 0;

    wcl.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);

    if(!RegisterClassEx(&wcl))
        return (0);

    hwnd = CreateWindow(
    szWinName,
    "グラフ作成ソフト",
```

```
    WS_MINIMIZEBOX | WS_SYSMENU | WS_OVERLAPPED,
    CW_USEDEFAULT,
    CW_USEDEFAULT,
    800,600,
    HWND_DESKTOP,
    NULL,
    hThisInst,
    NULL
          );

    ShowWindow(hwnd, nWinMode);
    UpdateWindow(hwnd);

    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

LRESULT CALLBACK WinProc(HWND hwnd, UINT msg, WPARAM wP, LPARAM lP)
{

    HDC hdc;
    static HPEN hPen;
    static HBRUSH hBrush;
    static HDC memdc;
    static HBITMAP hBit;
    static HFONT hFont1, hFont2;
    char strTitleText[256];
    static OPENFILENAME ofn;
    static char GotFileName[256];

    PAINTSTRUCT ps;
          RECT re1 = {0,0,800,600},
               re2 = {0,0,794,35},
```

```
            re3 = {0,35,100,556},
            re4 = {741,35,794,556},
            re5 = {100,516,794,556},
            re6 = {101, 35, 740, 516};

switch (msg) {
    case WM_CREATE:
        InitCommonControls();
        ofn.lStructSize = sizeof(OPENFILENAME);
        ofn.hwndOwner = hwnd;
        ofn.lpstrFilter = "ビットマップファイル (*.bmp)\0*.bmp\0 全てのファイル\0*.*\0";
        ofn.nFilterIndex = 1;
        ofn.lpstrFile = GotFileName;
        ofn.nMaxFile = 256;
        ofn.Flags = OFN_OVERWRITEPROMPT;
        ofn.lpstrDefExt = "bmp";

        hBrush = CreateSolidBrush(RGB(255,255,255));

        CreateMemdc(hwnd, &memdc, &hBit, WHITE_BRUSH);

        hBlackPen = CreatePen(PS_SOLID,0,RGB(0,0,0));
        hRedPen = CreatePen(PS_SOLID,0,RGB(255,0,0));
        hBluePen = CreatePen(PS_SOLID,0,RGB(0,0,255));
        hGreenPen = CreatePen(PS_SOLID,0,RGB(0,255,0));
        hYellowPen = CreatePen(PS_SOLID,0,RGB(255,255,0));
        hGreyPen = CreatePen(PS_SOLID,0,RGB(220,220,220));
        hWhitePen = CreatePen(PS_SOLID,0,RGB(255,255,255));
        hPen = hBlackPen;
        SelectObject(memdc, hPen);

        hFont1 = CreateFont(18,0,0,0,FW_NORMAL,0,0,0,SHIFTJIS_CHARSET,
                 OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
                 DEFAULT_PITCH | FF_DONTCARE,"MS ゴシック");

        hFont2 = CreateFont(18,0,900,0,FW_NORMAL,0,0,0,SHIFTJIS_CHARSET,
```

```
            OUT_DEFAULT_PRECIS,CLIP_DEFAULT_PRECIS,DEFAULT_QUALITY,
            DEFAULT_PITCH | FF_DONTCARE,"MS ゴシック");
    break;

    case WM_COMMAND:
        switch (LOWORD(wP)){
        case SAVENAMEFILE:
            WriteMyFile(hwnd);
            break;
        case SAVEFILE:
            OverWriteMyFile(hwnd);
            break;
        case NEWFILE:
            MakeNewFile(hwnd);
            break;
        case FILEOPEN:
            OpenFiles(hwnd);
            GetPrivateProfileString(strSection1,strKey1,"false",
                    strData1,sizeof(strData1),FileName);
                                    ⋮
            GetPrivateProfileString(strSection1,strKey18,"false",
                     strData18,sizeof(strData18),FileName);
            GetPrivateProfileString(strSection2,strxa1,"false",
                      strxaData1,sizeof(strxaData1),FileName);
                                    ⋮
            GetPrivateProfileString(strSection2,strxa50,"false",
                      strxaData50,sizeof(strxaData50),FileName);
            GetPrivateProfileString(strSection2,strya1,"false",
                      stryaData1,sizeof(stryaData1),FileName);
                                    ⋮
            GetPrivateProfileString(strSection2,strya50,"false",
                      stryaData50,sizeof(stryaData50),FileName);
            GetPrivateProfileString(strSection3,strxb1,"false",
                      strxbData1,sizeof(strxbData1),FileName);
                                    ⋮
```

※

```
GetPrivateProfileString(strSection3,strxb50,"false",
        Strxb50,sizeof(strxbData50),FileName);
GetPrivateProfileString(strSection4,strxc1,"false",
          strxcData1,sizeof(strxcData1),FileName);
                    ⋮
GetPrivateProfileString(strSection4,strxc50,"false",
          strxcData50,sizeof(strxcData50),FileName);
                    ⋮
GetPrivateProfileString(strSection4,stryc1,"false",
          strycData1,sizeof(strycData1),FileName);
                    ⋮
GetPrivateProfileString(strSection4,stryc50,"false",
          strycData50,sizeof(strycData50),FileName);
GetPrivateProfileString(strSection5,strxd1,"false",
          strxdData1,sizeof(strxdData1),FileName);
                    ⋮
GetPrivateProfileString(strSection5,strxd50,"false",
          strxdData50,sizeof(strxdData50),FileName);
GetPrivateProfileString(strSection5,stryd1,"false",
          strydData1,sizeof(strydData1),FileName);
                    ⋮
GetPrivateProfileString(strSection5,stryd50,"false",
          strydData50,sizeof(strydData50),FileName);
GetPrivateProfileString(strSection6,strxe1,"false",
          strxeData1,sizeof(strxeData1),FileName);
                    ⋮
GetPrivateProfileString(strSection6,strxe50,"false",
          strxeData50,sizeof(strxeData50),FileName);
                    ⋮
GetPrivateProfileString(strSection6,stryd1,"false",
          stryeData1,sizeof(stryeData1),FileName);
                    ⋮
GetPrivateProfileString(strSection6,strye50,"false",
          stryeData50,sizeof(stryeData50),FileName);
```

※

```
arg1 = atol(strData1);
arg2 = atol(strData2);
arg3 = atol(strData3);
arg4 = atol(strData4);
arg5 = atol(strData9); //メモリ幅
arg6 = atol(strData10); //メモリ幅
arg7 = atol(strData11); //グラフ数
arg8 = atol(strData12); //グラフ１
arg9 = atol(strData13); //２
arg10 = atol(strData14); //３
arg11 = atol(strData15); //４
arg12 =atol(strData16); //５
arg13 = atol(strData17); //４
arg14 =atol(strData18);
wsprintf(XPOS, "%s", strData5);
wsprintf(YPOS, "%s", strData6);
wsprintf(X_title, "%s", strData7);
wsprintf(Y_title, "%s", strData8);

wsprintf(xa1,"%s",strxaData1);    wsprintf(ya1,"%s",stryaData1);
                ⋮                                 ⋮
wsprintf(xa50,"%s",strxaData1);   wsprintf(ya50,"%s",stryaData1);
wsprintf(xb1,"%s",strxbData1);    wsprintf(yb1,"%s",strybData1);
                ⋮                                 ⋮
wsprintf(xb50,"%s",strxbData1);   wsprintf(yb50,"%s",strybData1);
wsprintf(xc1,"%s",strxcData1);    wsprintf(yc1,"%s",strycData1);
                ⋮                                 ⋮
wsprintf(xc50,"%s",strxcData1);   wsprintf(yc50,"%s",strycData1);
wsprintf(xd1,"%s",strxdData1);    wsprintf(yd1,"%s",strydData1);
                ⋮                                 ⋮
wsprintf(xd50,"%s",strxdData1);   wsprintf(yd50,"%s",strydData1);
wsprintf(xe1,"%s",strxeData1);    wsprintf(ye1,"%s",stryeData1);
                ⋮                                 ⋮
wsprintf(xe50,"%s",strxeData1);   wsprintf(ye50,"%s",stryeData1);

break;
```

※

```
case GRAPH:
    CreateMyProp(hwnd);
    FillRect(memdc, &re1, hBrush);
    SelectObject(memdc, hPen);
    Rectangle(memdc, 100, 15, 741, 456);
    MoveToEx(memdc, 420, 15, NULL);
    LineTo(memdc, 420, 456);
    MoveToEx(memdc, 100, 235, NULL);
    LineTo(memdc, 741, 235);

    interpolate(memdc, px, py, Xpos, Ypos, arg1, arg2, arg3, arg4, arg5,
              arg6, arg7, arg8, arg9, arg10, arg11, arg12, arg13, arg14, 1);

    FillRect(memdc, &re2, hBrush);
    FillRect(memdc, &re3, hBrush);
    FillRect(memdc, &re4, hBrush);
    FillRect(memdc, &re5, hBrush);
    InvalidateRect(hwnd, NULL, 1);
break;

case DEL:
    SelectObject(memdc, hBrush);
    FillRect(memdc, &re1, hBrush);
    SelectObject(memdc, hBlackPen);
    Rectangle(memdc, 100, 15, 741, 456);
    SelectObject(memdc, hBlackPen);
    MoveToEx(memdc, 420, 15, NULL);    //X軸
    LineTo(memdc, 420, 456);
    MoveToEx(memdc, 100, 235, NULL);   //Y軸
    LineTo(memdc, 741, 235);
    InvalidateRect(hwnd, NULL, 1);
break;

case SAVEBMP:
    if(GetSaveFileName(&ofn)){
        if(SaveBitmapFile(memdc,hBit,GotFileName) == TRUE)
```

```
                wsprintf(strTitleText,"Russian Paint - %s",GotFileName);
            else
                wsprintf(strTitleText,"Russian Paint
                                - ファイル書き出しに失敗しました");
            SetWindowText(hwnd,strTitleText);
        }
        break;

        case PRINT:
            MyPrint(memdc);
            break;

        case SETPRINTER:
            PrinterSet();
            break;

        case VERSION:
            MessageBox(hwnd, "Version 1.00α", "バージョン情報", MB_OK);
            break;

        case EXIT:
            PostQuitMessage(0);
            break;
        }
        break;

    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        BitBlt(hdc,ps.rcPaint.left,ps.rcPaint.top,
                    ps.rcPaint.right-ps.rcPaint.left,
                    ps.rcPaint.bottom-ps.rcPaint.top,
                    memdc,ps.rcPaint.left,ps.rcPaint.top,SRCCOPY);
        SelectObject(hdc, hFont1);
        TextOut(hdc, width_x, 530, X_title, strlen(X_title));
        SelectObject(hdc, hFont2);
        TextOut(hdc, 70, width_y, Y_title, strlen(Y_title));
```

```
                EndPaint(hwnd, &ps);
                break;

            case WM_DESTROY:
                DeleteDC(memdc);
                DeleteObject(hBit);
                DeleteObject(hPen);
                DeleteObject((HGDIOBJ)hFont1);
                DeleteObject((HGDIOBJ)hFont2);

                PostQuitMessage(0);
                break;

            default:
                return (DefWindowProc(hwnd, msg, wP, lP));
    }
    return 0L;
}

void OpenFiles(HWND hWnd)
{
    OPENFILENAME ofn;
    memset(&ofn, 0, sizeof(OPENFILENAME));
    ofn.lStructSize = sizeof(OPENFILENAME);
    ofn.hwndOwner = hWnd;
    ofn.lpstrFilter =
        "設定ファイル(*.ini)\0*.ini\0All Files(*.*)\0*.*\0\0";
    ofn.lpstrFile = FileName;
    ofn.nMaxFile = MAX_PATH;
    ofn.Flags = OFN_FILEMUSTEXIST;
    ofn.lpstrDefExt = "ini";
    ofn.nMaxFileTitle = 64;
    ofn.lpstrFileTitle = FileTitle;
    ofn.lpstrTitle = "設定ファイルを開く";

    if (GetOpenFileName(&ofn)) {
```

```
        show = 1;
        InvalidateRect(hWnd, NULL, TRUE);
    }

}

int WriteMyFile(HWND hwnd)
{
   OPENFILENAME ofn;

   memset(&ofn, 0, sizeof(OPENFILENAME));
   ofn.lStructSize = sizeof(OPENFILENAME);
   ofn.hwndOwner = hwnd;
   ofn.lpstrFilter = "設定ファイル(*.ini)\0*.ini\0All files(*.*)\0*.*\0\0";
   ofn.lpstrFile = FileName;
   ofn.lpstrFileTitle = File;
   ofn.nFilterIndex = 1;
   ofn.nMaxFile = sizeof(FileName);
   ofn.nMaxFileTitle = sizeof(File);
   ofn.Flags = OFN_OVERWRITEPROMPT | OFN_HIDEREADONLY;
   ofn.lpstrDefExt = "ini";
   ofn.lpstrTitle = "名前を付けて保存する";

   if(GetSaveFileName(&ofn) == 0)
      return -1;
   wsprintf(strData1, "%d", arg1);
   wsprintf(strData2, "%d", arg2);
   wsprintf(strData3, "%d", arg3);
   wsprintf(strData4, "%d", arg4);
   wsprintf(strData5, "%s", XPOS);
   wsprintf(strData6, "%s", YPOS);
   wsprintf(strData7, "%s", X_title);
   wsprintf(strData8, "%s", Y_title);
   wsprintf(strData9, "%d", arg5);
   wsprintf(strData10, "%d", arg6);
   wsprintf(strData13, "%d", arg9);
```

```
wsprintf(strData14, "%d", arg10);
wsprintf(strData15, "%d", arg11);
wsprintf(strData16, "%d", arg12);
wsprintf(strData17, "%d", arg13);
wsprintf(strData18, "%d", arg14);

wsprintf(xa1,"%s",strxaData1);    wsprintf(ya1,"%s",stryaData1);
                ⋮                                 ⋮
wsprintf(xa50,"%s",strxaData1);   wsprintf(ya50,"%s",stryaData1);
wsprintf(xb1,"%s",strxbData1);    wsprintf(yb1,"%s",strybData1);
                ⋮                                 ⋮
wsprintf(xb50,"%s",strxbData1);   wsprintf(yb50,"%s",strybData1);
wsprintf(xc1,"%s",strxcData1);    wsprintf(yc1,"%s",strycData1);
                ⋮                                 ⋮
wsprintf(xc50,"%s",strxcData1);   wsprintf(yc50,"%s",strycData1);
wsprintf(xd1,"%s",strxdData1);    wsprintf(yd1,"%s",strydData1);
                ⋮                                 ⋮
wsprintf(xd50,"%s",strxdData1);   wsprintf(yd50,"%s",strydData1);
wsprintf(xe1,"%s",strxeData1);    wsprintf(ye1,"%s",stryeData1);
                ⋮                                 ⋮
wsprintf(xe50,"%s",strxeData1);   wsprintf(ye50,"%s",stryeData1);

WritePrivateProfileString(strSection1, strKey1, strData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection1, strKey18, strData18, FileName);
WritePrivateProfileString(strSection2, strxa1, strxaData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection2, strxa50, strxaData50, FileName);
WritePrivateProfileString(strSection2, strya1, stryaData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection2, strya50, stryaData50, FileName);
WritePrivateProfileString(strSection3, strxb1, strxbData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection3, strxb50, strxbData50, FileName);
WritePrivateProfileString(strSection3, stryb1, strybData1, FileName);
```

※

```
    WritePrivateProfileString(strSection3, stryb50, strybData50, FileName);
    WritePrivateProfileString(strSection4, strxc1, strxcData1, FileName);
                                        ⋮
    WritePrivateProfileString(strSection4, strxc50, strxcData50, FileName);
    WritePrivateProfileString(strSection4, stryc1, strycData1, FileName);
                                        ⋮
    WritePrivateProfileString(strSection4, stryc50, strycData50, FileName);
    WritePrivateProfileString(strSection5, strxd1, strxdData1, FileName);
                                        ⋮
    WritePrivateProfileString(strSection5, strxd50, strxdData50, FileName);
    WritePrivateProfileString(strSection5, stryd1, strydData1, FileName);
                                        ⋮
    WritePrivateProfileString(strSection5, stryd50, strydData50, FileName);
    WritePrivateProfileString(strSection6, strxe1, strxeData1, FileName);
                                        ⋮
    WritePrivateProfileString(strSection6, strxe50, strxeData50, FileName);
    WritePrivateProfileString(strSection6, strye1, stryeData1, FileName);
                                        ⋮
    WritePrivateProfileString(strSection6, strye50, stryeData50, FileName);
    return 0;
}

int OverWriteMyFile(HWND hWnd)
{
    wsprintf(strData1, "%d", arg1);
    wsprintf(strData2, "%d", arg2);
    wsprintf(strData3, "%d", arg3);
    wsprintf(strData4, "%d", arg4);
    wsprintf(strData5, "%s", XPOS);
    wsprintf(strData6, "%s", YPOS);
    wsprintf(strData7, "%s", X_title);
    wsprintf(strData8, "%s", Y_title);
    wsprintf(strData9, "%d", arg5);
    wsprintf(strData10, "%d", arg6);
    wsprintf(strData13, "%d", arg9);
    wsprintf(strData14, "%d", arg10);
```

※

```
wsprintf(strData15, "%d", arg11);
wsprintf(strData16, "%d", arg12);
wsprintf(strData17, "%d", arg13);
wsprintf(strData18, "%d", arg14);

wsprintf(xa1,"%s",strxaData1);    wsprintf(ya1,"%s",stryaData1);
                ⋮                                 ⋮
wsprintf(xa50,"%s",strxaData1);   wsprintf(ya50,"%s",stryaData1);
wsprintf(xb1,"%s",strxbData1);    wsprintf(yb1,"%s",strybData1);
                ⋮                                 ⋮
wsprintf(xb50,"%s",strxbData1);   wsprintf(yb50,"%s",strybData1);
wsprintf(xc1,"%s",strxcData1);    wsprintf(yc1,"%s",strycData1);
                ⋮                                 ⋮
wsprintf(xc50,"%s",strxcData1);   wsprintf(yc50,"%s",strycData1);
wsprintf(xd1,"%s",strxdData1);    wsprintf(yd1,"%s",strydData1);
                ⋮                                 ⋮
wsprintf(xd50,"%s",strxdData1);   wsprintf(yd50,"%s",strydData1);
wsprintf(xe1,"%s",strxeData1);    wsprintf(ye1,"%s",stryeData1);
                ⋮                                 ⋮
wsprintf(xe50,"%s",strxeData1);   wsprintf(ye50,"%s",stryeData1);

WritePrivateProfileString(strSection1, strKey1, strData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection1, strKey18, strData18, FileName);
WritePrivateProfileString(strSection2, strxa1, strxaData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection2, strxa50, strxaData50, FileName);
WritePrivateProfileString(strSection2, strya1, stryaData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection2, strya50, stryaData50, FileName);
WritePrivateProfileString(strSection3, strxb1, strxbData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection3, strxb50, strxbData50, FileName);
WritePrivateProfileString(strSection3, stryb1, strybData1, FileName);
                                  ⋮
WritePrivateProfileString(strSection3, stryb50, strybData50, FileName);
```

※

```
   WritePrivateProfileString(strSection4, strxc1, strxcData1, FileName);
                                    ⋮
   WritePrivateProfileString(strSection4, strxc50, strxcData50, FileName);
   WritePrivateProfileString(strSection4, stryc1, strycData1, FileName);
                                    ⋮
   WritePrivateProfileString(strSection4, stryc50, strycData50, FileName);
   WritePrivateProfileString(strSection5, strxd1, strxdData1, FileName);
                                    ⋮
   WritePrivateProfileString(strSection5, strxd50, strxdData50, FileName);
   WritePrivateProfileString(strSection5, stryd1, strydData1, FileName);
                                    ⋮
   WritePrivateProfileString(strSection5, stryd50, strydData50, FileName);
   WritePrivateProfileString(strSection6, strxe1, strxeData1, FileName);
                                    ⋮
   WritePrivateProfileString(strSection6, strxe50, strxeData50, FileName);
   WritePrivateProfileString(strSection6, strye1, stryeData1, FileName);
                                    ⋮
   WritePrivateProfileString(strSection6, strye50, stryeData50, FileName);
   return 0;
}
int MakeNewFile(HWND hWnd)
{
   int id;
   id = MessageBox(hWnd,"以前のデータは消去されます。保存しますか？",
                "注意", MB_YESNO | MB_ICONQUESTION);
   if (id == IDYES)
      WriteMyFile(hWnd);

   arg1 = 0;
   arg2 = 0;
   arg3 = 0;
   arg4 = 0;
   arg5 = 0; //メモリ幅
   arg6 = 0; //メモリ幅
   arg7 = 0; //グラフ数
   arg8 = 0; //グラフ１
```

※

```
arg9 = 0; //2
arg10 =0;  //3
arg11 =0;  //4
arg12 =0; //5
arg13 =0;  //4
arg14 =0; //
wsprintf(XPOS, "%s", "");
wsprintf(YPOS, "%s", "");
wsprintf(X_title, "%s", "");
wsprintf(Y_title, "%s", "");
wsprintf(xa1,"%s","");    wsprintf(ya1,"%s","");
          ┆                       ┆
wsprintf(xa50,"%s","");   wsprintf(ya50,"%s","");
wsprintf(xb1,"%s","");    wsprintf(yb1,"%s","");
          ┆                       ┆
wsprintf(xb50,"%s","");   wsprintf(yb50,"%s","");
wsprintf(xc1,"%s", "");   wsprintf(yc1,"%s","");
          ┆                       ┆                    ※
wsprintf(xc50,"%s","");   wsprintf(yc50,"%s","");
wsprintf(xd1,"%s", "");   wsprintf(yd1,"%s","");
          ┆                       ┆
wsprintf(xd50,"%s","");   wsprintf(yd50,"%s","");
wsprintf(xe1,"%s","");    wsprintf(ye1,"%s","");
          ┆                       ┆
wsprintf(xe50,"%s","");   wsprintf(ye50,"%s","");
SetDlgItemText(hWnd,IDC_EDIT1,XPOS);
SetDlgItemText(hWnd,IDC_EDIT2,YPOS);
SetDlgItemText(hWnd,IDC_EDIT3,(LPCTSTR)X_title);
SetDlgItemText(hWnd,IDC_EDIT4,(LPCTSTR)Y_title);
SendDlgItemMessage(hWnd,IDC_COMBO1,CB_SETCURSEL,(WPARAM)arg1,0L);   //グラフ
SendDlgItemMessage(hWnd,IDC_COMBO2,CB_SETCURSEL,(WPARAM)arg2,0L);   //倍率
SendDlgItemMessage(hWnd,IDC_COMBO3,CB_SETCURSEL,(WPARAM)arg3,0L);   //点
SendDlgItemMessage(hWnd,IDC_COMBO4,CB_SETCURSEL,(WPARAM)arg4,0L);   //色
SendDlgItemMessage(hWnd,IDC_COMBO5,CB_SETCURSEL,(WPARAM)arg5,0L);   //X メモリ
SendDlgItemMessage(hWnd,IDC_COMBO6,CB_SETCURSEL,(WPARAM)arg6,0L);   //Y メモリ
SendDlgItemMessage(hWnd,IDC_COMBO7,CB_SETCURSEL,(WPARAM)arg7,0L);   //グラフ数
```

```
    SendDlgItemMessage(hWnd,IDC_COMBO8,CB_SETCURSEL,(WPARAM)arg8,0L);    //グラフ１
    SendDlgItemMessage(hWnd,IDC_COMBO9,CB_SETCURSEL,(WPARAM)arg9,0L);    //グラフ２
    SendDlgItemMessage(hWnd,IDC_COMBO10,CB_SETCURSEL,(WPARAM)arg10,0L);   //グラフ３
    SendDlgItemMessage(hWnd,IDC_COMBO11,CB_SETCURSEL,(WPARAM)arg11,0L);   //グラフ４
    SendDlgItemMessage(hWnd,IDC_COMBO12,CB_SETCURSEL,(WPARAM)arg12,0L);   //グラフ５
    SendDlgItemMessage(hWnd,IDC_COMBO13,CB_SETCURSEL,(WPARAM)arg13,0L);
    SendDlgItemMessage(hWnd,IDC_COMBO14,CB_SETCURSEL,(WPARAM)arg14,0L);

    SetDlgItemText(hWnd, IDC_xa1, xa1);     SetDlgItemText(hWnd, IDC_ya1,ya1);
                        ⋮                                   ⋮
    SetDlgItemText(hWnd, IDC_xa50, xa50);   SetDlgItemText(hWnd, IDC_ya50,ya50);
    SetDlgItemText(hWnd, IDC_xb1, xb1);     SetDlgItemText(hWnd, IDC_yb1,yb1);
                        ⋮                                   ⋮
    SetDlgItemText(hWnd, IDC_xb50, xb50);   SetDlgItemText(hWnd, IDC_yb50,yb50);
    SetDlgItemText(hWnd, IDC_xc1, xc1);     SetDlgItemText(hWnd, IDC_yc1,yc1);
                        ⋮                                   ⋮
    SetDlgItemText(hWnd, IDC_xc50, xc50);   SetDlgItemText(hWnd, IDC_yc50,yc50);     ※
    SetDlgItemText(hWnd, IDC_xd1, xd1);     SetDlgItemText(hWnd, IDC_yd1,yd1);
                        ⋮                                   ⋮
    SetDlgItemText(hWnd, IDC_xd50, xd50);   SetDlgItemText(hWnd, IDC_yd50,yd50);
    SetDlgItemText(hWnd, IDC_xe1, xe1);     SetDlgItemText(hWnd, IDC_ye1,ye1);
                        ⋮                                   ⋮
    SetDlgItemText(hWnd, IDC_xe50, xe50);   SetDlgItemText(hWnd, IDC_ye50,ye50);

    return 0;
}
```