**交叉项目制课程:力学基础程序设计**

王旌 2019141410308 力学软件实验班

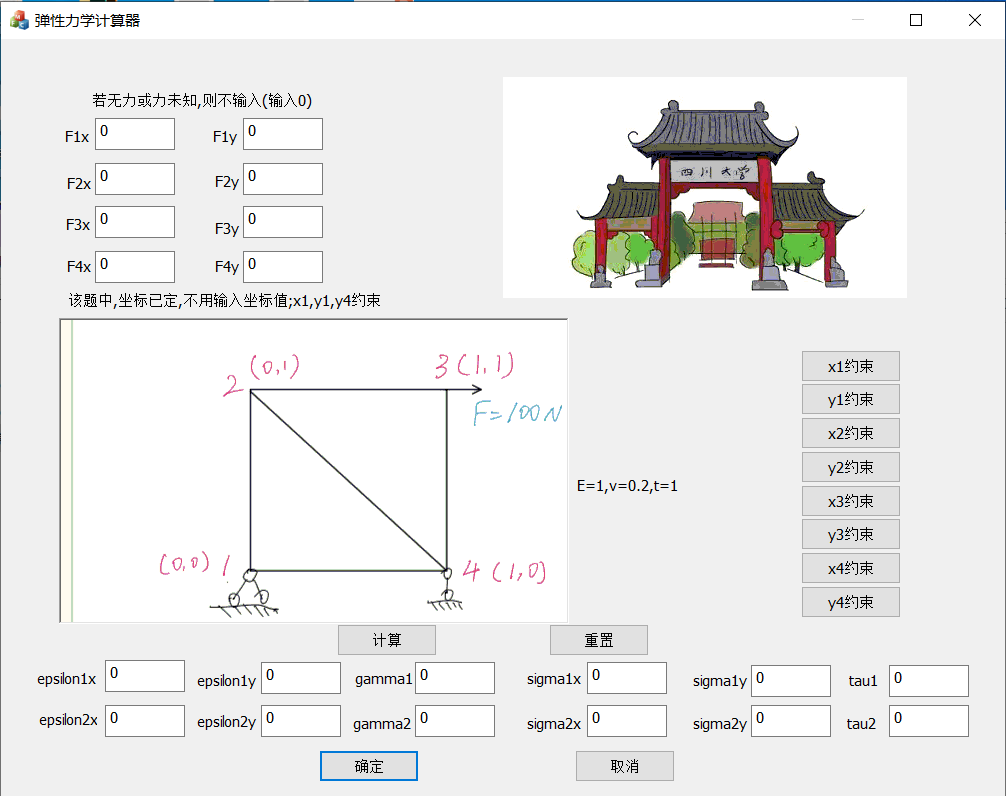
1. **项目介绍及设计思路**

本课程要求设计出一个具有人机交互功能的可执行文件。该文件可以让用户输入初始值并得到该弹性力学问题的解。

我是采用MFC来实现此文件.我最开始就搭建好了用户界面,包括输入、输出框、文本框、实现计算功能的按钮，最后再插入题目和图片。因为这样就能提醒我最早定义好正确数量的变量，便于以后的编程和降低犯错的可能。之后，我添加了可以能让用户自由更改约束条件的功能的按钮，使程序更加具有拓展性。

对于主程序部分，我分别对K,B,S矩阵写了三个含七个参数(及三坐标点和具体的二维数组)的函数,里面运用for循环赋值,这样在不同的小三角形中我只需调用这些函数并注意参数的对应就行了,大大减少了代码量.另外,对于双刚矩阵还需自己加以计算,并注意与单刚矩阵的对应,为减少代码量尽可能多的使用for循环.然后,我使用八个按钮来允许用户自己选择约束形式,并将对应约束的对角元素置一.最后我使用Jordan消去法进行矩阵方程求解,再用所得位移求得应力应变.

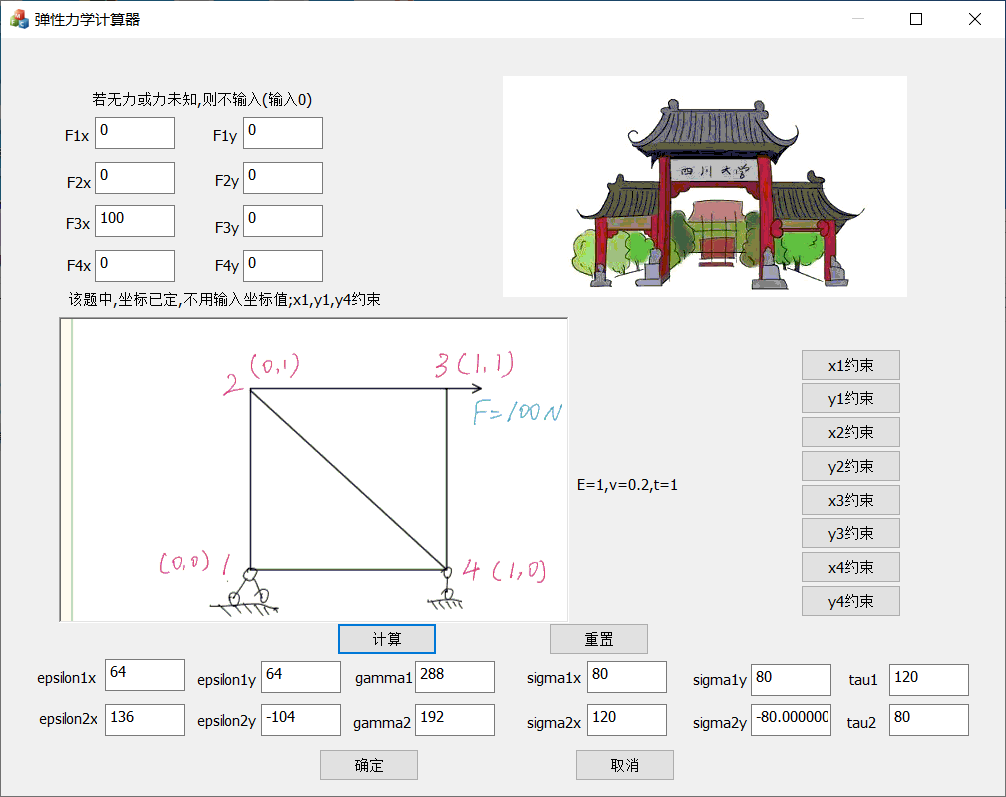
1. **程序介绍**



如图所示，用户总共有16个输入，分别是4个点的x和y方向上的受力,4个点的x和y方向上的约束条件.若用户不输入力的值则默认置0,用户可按下对应约束条件的按钮来实现约束功能,若用户按错或输错可点击重置按钮重新输入.(在该题中F3x=100,x1、y1、y4约束)

输入完成后点击计算按钮即可得到问题的解，即得到两小三角形的x、y方向应力应变及切应力和切应变（左下三角形为第一单元，右上三角形为第二单元）

得到该题的解如下所示：



1. **问题及总结**

在第一次算出结果的时候,我的值是错的,是因为在一个矩阵函数中的一行代码中赋值错误(复制粘贴导致三个函数的那句话都错了).而那句话中的变量颜色与其他语句的变量颜色明显不同,而自己没注意到这是VS中的提醒.以后需更加仔细,在未来编写大型程序的时候,调试会相当困难,应尽量减少低级错误.

另外,在最开始实现约束的时候,结果也是不正确的,问题在于按钮点击后没有改变在计算按钮函数中应改变的变量的值,所以我只能将该类变量定义为全局变量.

总之,通过此次训练,为自己以后开发软件奠定了良好的基础,自己也从这次任务中收获了很多,比如编代码的条理性和简洁性.

源代码

“matrix.h”

#define v 0.2

#define E 1

void Kmatrix(double xi, double yi, double xj, double yj, double xk, double yk,double K[][6])

{

double bi = yj - yk, bj = yk - yi, bk = yi - yj, ci = -xj + xk, cj = -xk + xi, ck = -xi + xj;

double b[3] = { bi,bj,bk }, c[3] = { ci,cj,ck }, k[6][6], A = ((yk - yi) \* (xj - xi) - (yi - yj) \* (xi - xk)) / 2;

int row, col;

for (row = 0; row < 5; row += 2)

{

for (col = 0; col < 5; col += 2)

k[row][col] = b[row / 2] \* b[col / 2] + (1 - 0.2) / 2 \* c[row / 2] \* c[col / 2];

for (col = 1; col < 6; col += 2)

k[row][col] = v \* b[row / 2] \* c[(col - 1) / 2] + (1 - v) / 2 \* c[row / 2] \* b[(col - 1) / 2];

}

for (row = 1; row < 6; row += 2)

{

for (col = 0; col < 5; col += 2)

k[row][col] = v \* c[(row - 1) / 2] \* b[col / 2] + (1 - v) / 2 \* b[(row - 1) / 2] \* c[col / 2];

for (col = 1; col < 6; col += 2)

k[row][col] = c[(row - 1) / 2] \* c[(col - 1) / 2] + (1 - v) / 2 \* b[(row - 1) / 2] \* b[(col - 1) / 2];

}

for (row = 0; row < 6; row++)

for (col = 0; col < 6; col++)

K[row][col] = k[row][col] \* E / (4 \* (1 - v \* v) \* A);

}

void Bmatrix(double xi, double yi, double xj, double yj, double xk, double yk, double B[][6])

{

double bi = yj - yk, bj = yk - yi, bk = yi - yj, ci = -xj + xk, cj = -xk + xi, ck = -xi + xj;

double b[3][6] = { {bi,0,bj,0,bk,0},{0,ci,0,cj,0,ck},{ci,bi,cj,bj,ck,bk} },A = ((yk - yi) \* (xj - xi) - (yi - yj) \* (xi - xk)) / 2;

int row, col;

for (row = 0; row < 3; row++)

for (col = 0; col < 6; col++)

B[row][col] = b[row][col] / (2 \* A);

}

void Smatrix(double xi, double yi, double xj, double yj, double xk, double yk, double S[][6])

{

double bi = yj - yk, bj = yk - yi, bk = yi - yj, ci = -xj + xk, cj = -xk + xi, ck = -xi + xj,A = ((yk - yi) \* (xj - xi) - (yi - yj) \* (xi - xk)) / 2;

double s[3][6] = { {bi,v \* ci,bj,v \* cj,bk,v \* ck},{v \* bi,ci,v \* bj,cj,v \* bk,ck},{(1-v)\*ci/2,(1 - v) \* bi / 2,(1 - v) \* cj / 2,(1 - v) \* bj / 2,(1 - v) \* ck / 2,(1 - v) \* bk / 2} };

int row, col;

for (row = 0; row < 3; row++)

for (col = 0; col < 6; col++)

S[row][col] = s[row][col]\*E / ((2 \* A)\*(1-v\*v));

}

// MFCApplication1Dlg.cpp: 实现文件

//

#include "pch.h"

#include "framework.h"

#include "MFCApplication1.h"

#include "MFCApplication1Dlg.h"

#include "afxdialogex.h"

#include "matrix.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#endif

int d[8] = {1,1,1,1,1,1,1,1};

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx

{

public:

CAboutDlg();

// 对话框数据

#ifdef AFX\_DESIGN\_TIME

enum { IDD = IDD\_ABOUTBOX };

#endif

protected:

virtual void DoDataExchange(CDataExchange\* pDX); // DDX/DDV 支持

// 实现

protected:

DECLARE\_MESSAGE\_MAP()

};

CAboutDlg::CAboutDlg() : CDialogEx(IDD\_ABOUTBOX)

{

}

void CAboutDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

}

BEGIN\_MESSAGE\_MAP(CAboutDlg, CDialogEx)

END\_MESSAGE\_MAP()

// CMFCApplication1Dlg 对话框

CMFCApplication1Dlg::CMFCApplication1Dlg(CWnd\* pParent /\*=nullptr\*/)

: CDialogEx(IDD\_MFCAPPLICATION1\_DIALOG, pParent)

, F1x(0)

, F1y(0)

, F2x(0)

, F2y(0)

, F3x(0)

, F3y(0)

, F4x(0)

, F4y(0)

, epsilon1x(0)

, epsilon1y(0)

, gamma1(0)

, epsilon2x(0)

, epsilon2y(0)

, gamma2(0)

, sigma1x(0)

, sigma1y(0)

, tau1(0)

, sigma2x(0)

, sigma2y(0)

, tau2(0)

{

m\_hIcon = AfxGetApp()->LoadIcon(IDR\_MAINFRAME);

}

void CMFCApplication1Dlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Text(pDX, IDC\_EDIT1, F1x);

DDX\_Text(pDX, IDC\_EDIT2, F1y);

DDX\_Text(pDX, IDC\_EDIT3, F2x);

DDX\_Text(pDX, IDC\_EDIT4, F2y);

DDX\_Text(pDX, IDC\_EDIT5, F3x);

DDX\_Text(pDX, IDC\_EDIT6, F3y);

DDX\_Text(pDX, IDC\_EDIT7, F4x);

DDX\_Text(pDX, IDC\_EDIT8, F4y);

DDX\_Text(pDX, IDC\_EDIT17, epsilon1x);

DDX\_Text(pDX, IDC\_EDIT19, epsilon1y);

DDX\_Text(pDX, IDC\_EDIT20, gamma1);

DDX\_Text(pDX, IDC\_EDIT21, epsilon2x);

DDX\_Text(pDX, IDC\_EDIT23, epsilon2y);

DDX\_Text(pDX, IDC\_EDIT24, gamma2);

DDX\_Text(pDX, IDC\_EDIT25, sigma1x);

DDX\_Text(pDX, IDC\_EDIT26, sigma1y);

DDX\_Text(pDX, IDC\_EDIT27, tau1);

DDX\_Text(pDX, IDC\_EDIT28, sigma2x);

DDX\_Text(pDX, IDC\_EDIT29, sigma2y);

DDX\_Text(pDX, IDC\_EDIT30, tau2);

}

BEGIN\_MESSAGE\_MAP(CMFCApplication1Dlg, CDialogEx)

ON\_WM\_SYSCOMMAND()

ON\_WM\_PAINT()

ON\_WM\_QUERYDRAGICON()

ON\_BN\_CLICKED(IDC\_BUTTON1, &CMFCApplication1Dlg::OnBnClickedButton1)

ON\_BN\_CLICKED(IDC\_BUTTON2, &CMFCApplication1Dlg::OnBnClickedButton2)

ON\_BN\_CLICKED(IDC\_BUTTON9, &CMFCApplication1Dlg::OnBnClickedButton9)

ON\_BN\_CLICKED(IDC\_BUTTON3, &CMFCApplication1Dlg::OnBnClickedButton3)

ON\_BN\_CLICKED(IDC\_BUTTON4, &CMFCApplication1Dlg::OnBnClickedButton4)

ON\_BN\_CLICKED(IDC\_BUTTON5, &CMFCApplication1Dlg::OnBnClickedButton5)

ON\_BN\_CLICKED(IDC\_BUTTON6, &CMFCApplication1Dlg::OnBnClickedButton6)

ON\_BN\_CLICKED(IDC\_BUTTON7, &CMFCApplication1Dlg::OnBnClickedButton7)

ON\_BN\_CLICKED(IDC\_BUTTON8, &CMFCApplication1Dlg::OnBnClickedButton8)

ON\_BN\_CLICKED(IDC\_BUTTON10, &CMFCApplication1Dlg::OnBnClickedButton10)

ON\_EN\_CHANGE(IDC\_EDIT26, &CMFCApplication1Dlg::OnEnChangeEdit26)

ON\_EN\_CHANGE(IDC\_EDIT6, &CMFCApplication1Dlg::OnEnChangeEdit6)

ON\_EN\_CHANGE(IDC\_EDIT5, &CMFCApplication1Dlg::OnEnChangeEdit5)

END\_MESSAGE\_MAP()

// CMFCApplication1Dlg 消息处理程序

BOOL CMFCApplication1Dlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

// 将“关于...”菜单项添加到系统菜单中。

// IDM\_ABOUTBOX 必须在系统命令范围内。

ASSERT((IDM\_ABOUTBOX & 0xFFF0) == IDM\_ABOUTBOX);

ASSERT(IDM\_ABOUTBOX < 0xF000);

CMenu\* pSysMenu = GetSystemMenu(FALSE);

if (pSysMenu != nullptr)

{

BOOL bNameValid;

CString strAboutMenu;

bNameValid = strAboutMenu.LoadString(IDS\_ABOUTBOX);

ASSERT(bNameValid);

if (!strAboutMenu.IsEmpty())

{

pSysMenu->AppendMenu(MF\_SEPARATOR);

pSysMenu->AppendMenu(MF\_STRING, IDM\_ABOUTBOX, strAboutMenu);

}

}

// 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动

SetIcon(m\_hIcon, TRUE); // 设置大图标

SetIcon(m\_hIcon, FALSE); // 设置小图标

// TODO: 在此添加额外的初始化代码

return TRUE; // 除非将焦点设置到控件，否则返回 TRUE

}

void CMFCApplication1Dlg::OnSysCommand(UINT nID, LPARAM lParam)

{

if ((nID & 0xFFF0) == IDM\_ABOUTBOX)

{

CAboutDlg dlgAbout;

dlgAbout.DoModal();

}

else

{

CDialogEx::OnSysCommand(nID, lParam);

}

}

// 如果向对话框添加最小化按钮，则需要下面的代码

// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，

// 这将由框架自动完成。

void CMFCApplication1Dlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // 用于绘制的设备上下文

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作区矩形中居中

int cxIcon = GetSystemMetrics(SM\_CXICON);

int cyIcon = GetSystemMetrics(SM\_CYICON);

CRect rect;

GetClientRect(&rect);

int x = (rect.Width() - cxIcon + 1) / 2;

int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialogEx::OnPaint();

}

}

//当用户拖动最小化窗口时系统调用此函数取得光标

//显示。

HCURSOR CMFCApplication1Dlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CMFCApplication1Dlg::OnBnClickedButton1()

{

UpdateData(TRUE);

double K1[6][6] = { 0 }, K2[6][6] = {0};

double a[8][8] = { 0 }, b[8] = { F1x,F1y,F2x,F2y,F3x,F3y,F4x,F4y };

int i,j,k,n=8;

double sum=0;

double B1[3][6] = { 0 }, B2[3][6] = { 0 }, S1[3][6] = { 0 }, S2[3][6] = { 0 };

double delta1[6] = { 0 }, delta2[6] = {0};

double x1 = 0, y1 = 0, x2 = 0,y2 = 1,x3 = 1,y3 = 1,x4 = 1,y4 = 0;

Kmatrix(x1, y1, x4, y4, x2, y2, K1);

Kmatrix(x4, y4, x3, y3, x2, y2, K2);

Bmatrix(x1, y1, x4, y4, x2, y2, B1);

Bmatrix(x4, y4, x3, y3, x2, y2, B2);

Smatrix(x1, y1, x4, y4, x2, y2, S1);

Smatrix(x4, y4, x3, y3, x2, y2, S2);

//a[8][8]

for (i = 0; i < 2; i++)

for (j = 0; j < 2; j++)

a[i][j] = K1[i][j];

for (i = 0; i < 2; i++)

for (j = 2; j < 4; j++)

a[i][j] = K1[i][j+2];

for (i = 0; i < 2; i++)

for (j = 6; j < 8; j++)

a[i][j] = K1[i][j-4];

for (i = 2; i < 4; i++)

for (j = 0; j < 2; j++)

a[i][j] = K1[i+2][j];

for (i = 2; i < 4; i++)

for (j = 2; j < 4; j++)

a[i][j] = K1[i + 2][j+2]+K2[i+2][j+2];

for (i = 2; i < 4; i++)

for (j = 4; j < 6; j++)

a[i][j] = K2[i + 2][j-2];

for (i = 2; i < 4; i++)

for (j = 6; j < 8; j++)

a[i][j] = K1[i+2][j-4]+K2[i + 2][j - 6];

for (i = 4; i < 6; i++)

for (j = 2; j < 4; j++)

a[i][j] = K2[i - 2][j + 2];

for (i = 4; i < 6; i++)

for (j = 4; j < 6; j++)

a[i][j] = K2[i - 2][j - 2];

for (i = 4; i < 6; i++)

for (j = 6; j < 8; j++)

a[i][j] = K2[i - 2][j - 6];

for (i = 6; i < 8; i++)

for (j = 0; j < 2; j++)

a[i][j] = K1[i - 4][j];

for (i = 6; i < 8; i++)

for (j = 2; j < 4; j++)

a[i][j] = K1[i - 4][j+2]+K2[i-6][j+2];

for (i = 6; i < 8; i++)

for (j = 4; j < 6; j++)

a[i][j] = K2[i - 6][j-2];

for (i = 6; i < 8; i++)

for (j = 6; j < 8; j++)

a[i][j] = K1[i - 4][j -4] + K2[i - 6][j -6];

//setting 0

for(i=0;i<8;i++)

if (d[i] == 0)

{

for (j = 0; j < 8; j++)

a[i][j] = 0;

for (j = 0; j < 8; j++)

a[j][i] = 0;

a[i][i] = 1;

}

//Jordan

for (k = 0; k < n; k++)

{

for (j = k + 1; j < n; j++)

a[k][j] = a[k][j] / a[k][k];

b[k] = b[k] / a[k][k];

for (i = 0; i < n; i++)

{

if (i == k) continue;

for (j = k + 1; j < n; j++)

a[i][j] = a[i][j] - a[i][k] \* a[k][j];

b[i] = b[i] - a[i][k] \* b[k];

}

}

//epsilon

delta1[0] = b[0]; delta1[1] = b[1]; delta1[2] = b[6]; delta1[3] = b[7]; delta1[4] = b[2]; delta1[5] = b[3];

delta2[0] = b[6]; delta2[1] = b[7]; delta2[2] = b[4]; delta2[3] = b[5]; delta2[4] = b[2]; delta2[5] = b[3];

sum = 0;

for (i=0,j = 0; j < 6; j++)

sum = sum + B1[i][j] \* delta1[j];

epsilon1x = sum;

sum = 0;

for (i = 1, j = 0; j < 6; j++)

sum = sum + B1[i][j] \* delta1[j];

epsilon1y = sum;

sum = 0;

for (i = 2, j = 0; j < 6; j++)

sum = sum + B1[i][j] \* delta1[j];

gamma1 = sum;

sum = 0;

for (i = 0, j = 0; j < 6; j++)

sum = sum + B2[i][j] \* delta2[j];

epsilon2x = sum;

sum = 0;

for (i = 1, j = 0; j < 6; j++)

sum = sum + B2[i][j] \* delta2[j];

epsilon2y = sum;

sum = 0;

for (i = 2, j = 0; j < 6; j++)

sum = sum + B2[i][j] \* delta2[j];

gamma2 = sum;

//sigma

sum = 0;

for (i = 0, j = 0; j < 6; j++)

sum = sum + S1[i][j] \* delta1[j];

sigma1x = sum;

sum = 0;

for (i = 1, j = 0; j < 6; j++)

sum = sum + S1[i][j] \* delta1[j];

sigma1y = sum;

sum = 0;

for (i = 2, j = 0; j < 6; j++)

sum = sum + S1[i][j] \* delta1[j];

tau1 = sum;

sum = 0;

for (i = 0, j = 0; j < 6; j++)

sum = sum + S2[i][j] \* delta2[j];

sigma2x = sum;

sum = 0;

for (i = 1, j = 0; j < 6; j++)

sum = sum + S2[i][j] \* delta2[j];

sigma2y = sum;

sum = 0;

for (i = 2, j = 0; j < 6; j++)

sum = sum + S2[i][j] \* delta2[j];

tau2 = sum;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnDropdownButton1(NMHDR\* pNMHDR, LRESULT\* pResult)

{

LPNMBCDROPDOWN pDropDown = reinterpret\_cast<LPNMBCDROPDOWN>(pNMHDR);

// TODO: 在此添加控件通知处理程序代码

\*pResult = 0;

}

void CMFCApplication1Dlg::OnBnClickedButton2()

{

UpdateData(TRUE);

d[0] = 0;

UpdateData(FALSE);

}

void CMFCApplication1Dlg::OnBnClickedButton3()

{

UpdateData(TRUE);

d[1] = 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton4()

{

UpdateData(TRUE);

d[2] = 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton5()

{

UpdateData(TRUE);

d[3] = 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton6()

{

UpdateData(TRUE);

d[4]= 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton7()

{

UpdateData(TRUE);

d[5] = 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton8()

{

UpdateData(TRUE);

d[6] = 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton9()

{

UpdateData(TRUE);

d[7] = 0;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnBnClickedButton10()

{

UpdateData(TRUE);

int a;

F1x = 0;

F1y = 0;

F2x = 0;

F2y = 0;

F3x = 0;

F3y = 0;

F4x = 0;

F4y = 0;

epsilon1x = 0;

epsilon1y = 0;

gamma1 = 0;

epsilon2x = 0;

epsilon2y = 0;

gamma2 = 0;

sigma1x = 0;

sigma1y = 0;

tau1 = 0;

sigma2x = 0;

sigma2y = 0;

tau2 = 0;

for (a = 0; a < 8; a++)

d[a] = 1;

UpdateData(FALSE);

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnEnChangeEdit26()

{

// TODO: 如果该控件是 RICHEDIT 控件，它将不

// 发送此通知，除非重写 CDialogEx::OnInitDialog()

// 函数并调用 CRichEditCtrl().SetEventMask()，

// 同时将 ENM\_CHANGE 标志“或”运算到掩码中。

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnEnChangeEdit6()

{

// TODO: 如果该控件是 RICHEDIT 控件，它将不

// 发送此通知，除非重写 CDialogEx::OnInitDialog()

// 函数并调用 CRichEditCtrl().SetEventMask()，

// 同时将 ENM\_CHANGE 标志“或”运算到掩码中。

// TODO: 在此添加控件通知处理程序代码

}

void CMFCApplication1Dlg::OnEnChangeEdit5()

{

// TODO: 如果该控件是 RICHEDIT 控件，它将不

// 发送此通知，除非重写 CDialogEx::OnInitDialog()

// 函数并调用 CRichEditCtrl().SetEventMask()，

// 同时将 ENM\_CHANGE 标志“或”运算到掩码中。

// TODO: 在此添加控件通知处理程序代码

}