# 计算方法实验

**题目：数据插值与拟合**

**班级：021161**

**学号：02116026**

**姓名：沈云彬**

# 1 实验目的

要求：

1. 给定一组数据，使用拉格朗日和牛顿插值法计算

2. 对计算所得数据进行多项式拟合

3. 要求提供用户交互界面供显示差值结果和拟合结果

功能：

功能一：拉格朗日插值及输出插值曲线

功能二：牛顿插值及输出插值曲线

功能三：曲线拟合及输出拟合曲线

功能四：多曲线比较及横纵坐标显示

功能五：曲线图像放大缩小

功能六：曲线拖动

功能七：完备的错误提示功能

# 2 实验步骤

我用800多行VB.NET代码，直观的实现。

# 3 代码（软件界面、功能在22页）

以下是VB.NET的代码（804行）：

Imports System.Text.RegularExpressions

Public Class Form1

Dim myXmin, myXmax, myYmin, myYmax, lenable, nenable, niheenable, enlarge, wrong, numberamount, x0, y0, myxl, myyl, Xmm, Ymm, Xxx, Yxx, tienable As Double

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles Me.Load

enlarge = 1

End Sub

Dim expression As New Regex("(.\*?),")

Declare Function SendMessage Lib "user32" Alias "SendMessageA" ( \_

ByVal hwnd As IntPtr, \_

ByVal wMsg As Integer, \_

ByVal wParam As Integer, \_

ByVal lParam As Integer) \_

As Boolean

Declare Function ReleaseCapture Lib "user32" Alias "ReleaseCapture" () As Boolean

Const WM\_SYSCOMMAND = &H112

Const SC\_MOVE = &HF010&

Const HTCAPTION = 2

Private Sub RectangleShape1\_MouseDown(ByVal sender As Object, \_

ByVal e As System.Windows.Forms.MouseEventArgs) \_

Handles RectangleShape1.MouseDown

ReleaseCapture()

SendMessage(Me.Handle, WM\_SYSCOMMAND, SC\_MOVE + HTCAPTION, 0)

End Sub

Private Sub label14\_MouseDown(ByVal sender As Object, \_

ByVal e As System.Windows.Forms.MouseEventArgs) \_

Handles Label14.MouseDown

ReleaseCapture()

SendMessage(Me.Handle, WM\_SYSCOMMAND, SC\_MOVE + HTCAPTION, 0)

End Sub

Private Function lagrange(ByVal x() As Double, ByVal y() As Double, ByVal xx As Double, ByVal n As Integer)

Dim i, j As Integer

Dim a(n), yy As Double

For i = 1 To n

a(i) = y(i)

For j = 1 To n

If j <> i Then

a(i) \*= (xx - x(j)) / (x(i) - x(j))

End If

Next

yy += a(i)

Next

lagrange = yy

End Function

Private Function newtint(ByVal x() As Double, ByVal y() As Double, ByVal xhat As Double, ByVal n As Integer)

Dim c(n), yhat As Double

For i = 1 To n

c(i) = y(i)

Next

For j = 2 To n

For i = n To j Step -1

c(i) = (c(i) - c(i - 1)) / (x(i) - x(i - j + 1))

Next

Next

yhat = c(n)

For i = n - 1 To 1 Step -1

yhat = yhat \* (xhat - x(i)) + c(i)

Next

newtint = yhat

End Function

Function MRinv(ByVal n As Integer, ByRef mtxA(,) As Double) As Boolean

' 局部变量

Dim nIs(n), nJs(n) As Integer

Dim i As Integer, j As Integer, k As Integer

Dim d As Double, p As Double

' 全选主元，消元

For k = 1 To n

d = 0.0#

For i = k To n

For j = k To n

p = Math.Abs(mtxA(i, j))

If (p > d) Then

d = p

nIs(k) = i

nJs(k) = j

End If

Next j

Next i

' 求解失败

If (d + 1.0# = 1.0#) Then

MRinv = False

Exit Function

End If

If (nIs(k) <> k) Then

For j = 1 To n

p = mtxA(k, j)

mtxA(k, j) = mtxA(nIs(k), j)

mtxA(nIs(k), j) = p

Next j

End If

If (nJs(k) <> k) Then

For i = 1 To n

p = mtxA(i, k)

mtxA(i, k) = mtxA(i, nJs(k))

mtxA(i, nJs(k)) = p

Next i

End If

mtxA(k, k) = 1.0# / mtxA(k, k)

For j = 1 To n

If (j <> k) Then mtxA(k, j) = mtxA(k, j) \* mtxA(k, k)

Next j

For i = 1 To n

If (i <> k) Then

For j = 1 To n

If (j <> k) Then mtxA(i, j) = mtxA(i, j) - mtxA(i, k) \* mtxA(k, j)

Next j

End If

Next i

For i = 1 To n

If (i <> k) Then mtxA(i, k) = -mtxA(i, k) \* mtxA(k, k)

Next i

Next k

' 调整恢复行列次序

For k = n To 1 Step -1

If (nJs(k) <> k) Then

For j = 1 To n

p = mtxA(k, j)

mtxA(k, j) = mtxA(nJs(k), j)

mtxA(nJs(k), j) = p

Next j

End If

If (nIs(k) <> k) Then

For i = 1 To n

p = mtxA(i, k)

mtxA(i, k) = mtxA(i, nIs(k))

mtxA(i, nIs(k)) = p

Next i

End If

Next k

' 求解成功

MRinv = True

End Function

Private Function ZXE(ByVal x() As Double, ByVal y() As Double, ByVal m As Integer, ByVal num As Integer)

Dim xls(num), xls2(2 \* m + 1), Ab(m + 1, m + 1), Ac(m + 1, m + 1) As Double

Dim S(1, 2 \* m + 1), T(m + 1, 1) As Double

Dim a(m + 1) As Single

For k = 1 To 2 \* m + 1

For i = 1 To num REM 曲线拟合有问题 要改！----已改正

xls(i) = x(i) ^ (k - 1)

Next

S(1, k) = xls.Sum

Next

For k = 1 To m + 1

For i = 1 To num

xls(i) = (x(i) ^ (k - 1)) \* y(i)

Next

T(k, 1) = xls.Sum

Next

For i = 1 To m + 1

For j = 1 To m + 1

Ab(i, j) = S(1, i + j - 1)

Next

Next

Ac = Ab

If MRinv(m + 1, Ac) = True Then

For i = 0 To m + 1

For j = 0 To m + 1

a(i) = Ac(i, j) \* T(j, 1) + a(i)

Next

Next

End If

ZXE = a

End Function

Private Sub math\_l()

ERRO.Text = "注:请以a,b,c,...的格式输入"

Dim cs, vinput, output As Double

Dim num As Integer

Dim grawidth, graheight, Yd1, Yd2, Yd, Xd, cc As Int64

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

Dim pengre As New Pen(Color.Green, 1)

Dim penblu As New Pen(Color.Blue, 1)

Dim penblk As New Pen(Color.Black, 2)

Dim penred As New Pen(Color.Red, 1)

grawidth = GRA.Width

graheight = GRA.Height

Dim Yiiii(grawidth), Yi(grawidth), Yz(grawidth) As Double

Dim mc As MatchCollection = expression.Matches(Xine.Text)

num = mc.Count

Dim Yinput2(num), Xinput(num), Yinput(num) As Double

For i = 1 To num

Xinput(i) = mc(i - 1).ToString

Next

Dim md As MatchCollection = expression.Matches(Yine.Text)

If mc.Count > 2 Then

If num = md.Count Then

For i = 1 To num

Yinput(i) = md(i - 1).ToString

Next

cc = TextBox5.Text + 1

Dim mcll As MatchCollection = expression.Matches(VIN.Text)

vinput = VIN.Text

output = mcll.Count

If output > 0 Then

TextBox1.Text = ""

Dim cath(output) As Double

For i = 1 To output

cath(i) = mcll(i - 1).ToString

TextBox1.Text += lagrange(Xinput, Yinput, cath(i), cc).ToString + ","

Next

End If

TextBox2.Text = num.ToString

For i = 1 To num

Yinput2(i) = Yinput(i)

Next

For ik = 1 To grawidth

For i = 1 To num

Yinput(i) = Yinput2(i)

Next

cs = (ik / grawidth) \* (myXmax - myXmin) + myXmin REM ok1

Yz(ik) = lagrange(Xinput, Yinput, cs, cc)

Next

For i = 1 To grawidth

Yiiii(i) = ((Yz(i) - myYmin) / (myYmax - myYmin)) \* graheight

Next

For xi7 = 1 To grawidth - 1

Yd1 = graheight - Yiiii(xi7)

Yd2 = graheight - Yiiii(xi7 + 1)

gr1.DrawLine(penblu, xi7, Yd1, xi7 + 1, Yd2)

Next

Yinput = Yinput2

For i = 1 To num

Yd = graheight - ((Yinput(i) - myYmin) / (myYmax - myYmin)) \* graheight

Xd = ((Xinput(i) - myXmin) / (myXmax - myXmin)) \* grawidth

gr1.DrawLine(penblk, Xd - 5, Yd - 5, Xd + 5, Yd + 5)

gr1.DrawLine(penblk, Xd + 5, Yd - 5, Xd - 5, Yd + 5)

Next

LabelYmax.Text = "Ymax:" + Format(Val(myYmax.ToString), "0.0")

LabelYmin.Text = "Ymin:" + Format(Val(myYmin.ToString), "0.0")

LabelXmax.Text = "Xmax:" + Format(Val(myXmax.ToString), "0.0")

LabelXmin.Text = "Xmin:" + Format(Val(myXmin.ToString), "0.0")

Else

ERRO.Text = "错误！x,y输入个数不同！"

End If

Else

ERRO.Text = "错误！x,y输入个数太少！"

End If

End Sub

Private Sub math\_n()

ERRO.Text = "注:请以a,b,c,...的格式输入"

Dim vinput, output, cs As Double

Dim num As Integer

Dim grawidth, graheight, Yd1, Yd2, Yd, Xd As Int64

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

Dim pengre As New Pen(Color.Green, 1)

Dim penblu As New Pen(Color.Blue, 1)

Dim penblk As New Pen(Color.Black, 2)

Dim penred As New Pen(Color.Red, 1)

grawidth = GRA.Width

graheight = GRA.Height

Dim Yiiii(grawidth), Yi(grawidth), Yz(grawidth) As Double

Dim mc As MatchCollection = expression.Matches(Xine.Text)

num = mc.Count

Dim Xinput(num), Yinput(num), Yinput2(num) As Double

For i = 1 To num

Xinput(i) = mc(i - 1).ToString

Next

Dim md As MatchCollection = expression.Matches(Yine.Text)

If mc.Count > 2 Then

If num = md.Count Then

For i = 1 To num

Yinput(i) = md(i - 1).ToString

Next

vinput = VIN.Text

For i = 1 To num

Yinput2(i) = Yinput(i)

Next

Dim mcll As MatchCollection = expression.Matches(VIN.Text)

vinput = VIN.Text

output = mcll.Count

If output > 0 Then

TextBox1.Text = ""

Dim cath(output) As Double

For i = 1 To output

cath(i) = mcll(i - 1).ToString

TextBox1.Text += newtint(Xinput, Yinput, cath(i), num).ToString + ","

Next

End If

TextBox2.Text = num.ToString

For ik = 1 To grawidth

For i = 1 To num

Yinput(i) = Yinput2(i)

Next

cs = (ik / grawidth) \* (myXmax - myXmin) + myXmin REM ok2

Yz(ik) = newtint(Xinput, Yinput, cs, num)

Next

For i = 1 To grawidth

Yiiii(i) = ((Yz(i) - myYmin) / (myYmax - myYmin)) \* graheight

Next

For xi7 = 1 To grawidth - 1

Yd1 = graheight - Yiiii(xi7)

Yd2 = graheight - Yiiii(xi7 + 1)

gr1.DrawLine(penred, xi7, Yd1, xi7 + 1, Yd2)

Next

Yinput = Yinput2

For i = 1 To num

Yd = graheight - ((Yinput(i) - myYmin) / (myYmax - myYmin)) \* graheight

Xd = ((Xinput(i) - myXmin) / (myXmax - myXmin)) \* grawidth

gr1.DrawLine(penblk, Xd - 5, Yd - 5, Xd + 5, Yd + 5)

gr1.DrawLine(penblk, Xd + 5, Yd - 5, Xd - 5, Yd + 5)

Next

LabelYmax.Text = "Ymax:" + Format(Val(myYmax.ToString), "0.0")

LabelYmin.Text = "Ymin:" + Format(Val(myYmin.ToString), "0.0")

LabelXmax.Text = "Xmax:" + Format(Val(myXmax.ToString), "0.0")

LabelXmin.Text = "Xmin:" + Format(Val(myXmin.ToString), "0.0")

Else

ERRO.Text = "错误！x,y输入个数不同！"

End If

Else

ERRO.Text = "错误！x,y输入个数太少！"

End If

End Sub

Private Sub math\_nihe()

ERRO.Text = "注:请以a,b,c,...的格式输入"

Dim cishu As Double

Dim num, grawidth, graheight, Yd1, Yd2, Yd, Xd As Int64

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

Dim pengre As New Pen(Color.Green, 1)

Dim penblu As New Pen(Color.Blue, 1)

Dim penblk As New Pen(Color.Black, 2)

Dim penred As New Pen(Color.Red, 1)

grawidth = GRA.Width

graheight = GRA.Height

Dim Yii(grawidth), Yi(grawidth) As Single

Dim mc As MatchCollection = expression.Matches(Xine.Text)

num = mc.Count

Dim Xinput(num), Yinput(num) As Double

For i = 1 To num

Xinput(i) = mc(i - 1).ToString

Next

Dim md As MatchCollection = expression.Matches(Yine.Text)

If mc.Count > 2 Then

If num = md.Count Then

For i = 1 To num

Yinput(i) = md(i - 1).ToString

Next

TextBox2.Text = num.ToString

cishu = TextBox3.Text

Dim out(cishu) As Single

out = ZXE(Xinput, Yinput, cishu, num)

TextY.Text = ""

For i = cishu + 1 To 1 Step -1

TextY.Text += Format(Val(out(i).ToString), "0.00")

TextY.Text += " x^"

TextY.Text += (i - 1).ToString

If i <> 1 Then

TextY.Text += " + "

End If

Next

For i = 1 To grawidth

For j = 1 To cishu + 1

Yi(i) += out(j) \* ((i / grawidth) \* (myXmax - myXmin) + myXmin) ^ (j - 1) REM ok3

Next

Next

For i = 1 To grawidth

Yii(i) = ((Yi(i) - myYmin) / (myYmax - myYmin)) \* graheight

Next

For xi7 = 1 To grawidth - 1

Yd1 = graheight - Yii(xi7)

Yd2 = graheight - Yii(xi7 + 1)

gr1.DrawLine(pengre, xi7, Yd1, xi7 + 1, Yd2)

Next

For i = 1 To num

Yd = graheight - ((Yinput(i) - myYmin) / (myYmax - myYmin)) \* graheight

Xd = ((Xinput(i) - myXmin) / (myXmax - myXmin)) \* grawidth

gr1.DrawLine(penblk, Xd - 5, Yd - 5, Xd + 5, Yd + 5)

gr1.DrawLine(penblk, Xd + 5, Yd - 5, Xd - 5, Yd + 5)

Next

LabelYmax.Text = "Ymax:" + Format(Val(myYmax.ToString), "0.0")

LabelYmin.Text = "Ymin:" + Format(Val(myYmin.ToString), "0.0")

LabelXmax.Text = "Xmax:" + Format(Val(myXmax.ToString), "0.0")

LabelXmin.Text = "Xmin:" + Format(Val(myXmin.ToString), "0.0")

Else

ERRO.Text = "错误！x,y输入个数不同！"

End If

Else

ERRO.Text = "错误！x,y输入个数太少！"

End If

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles L\_chazhi.Click

If CheckBox1.Checked = True Then

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

nenable = 0

niheenable = 0

lenable = 0

End If

lenable = 1

If wrong = 0 Then

If numberamount > TextBox5.Text Then

Call math\_l()

Else

ERRO.Text = "错误！x,y输入个数少与插值次数的要求！"

End If

End If

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles N\_chazhi.Click

If CheckBox1.Checked = True Then

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

nenable = 0

niheenable = 0

lenable = 0

End If

nenable = 1

If wrong = 0 Then

Call math\_n()

End If

End Sub

Private Sub Button1\_Click\_1(sender As Object, e As EventArgs) Handles Line\_nihe.Click

If CheckBox1.Checked = True Then

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

nenable = 0

niheenable = 0

lenable = 0

End If

niheenable = 1

If wrong = 0 Then

If numberamount > TextBox3.Text Then

Call math\_nihe()

Else

ERRO.Text = "错误！x,y输入个数少与拟合次数的要求！"

End If

End If

End Sub

Private Sub RectangleShape2\_MouseClick(sender As Object, e As MouseEventArgs) Handles RectangleShape2.MouseClick

Me.Close()

End Sub

Private Sub RectangleShape2\_MouseLeave(sender As Object, e As EventArgs) Handles RectangleShape2.MouseLeave

RectangleShape2.FillColor = Color.Red

RectangleShape2.BorderColor = Color.Red

Label13.BackColor = Color.Red

Label13.ForeColor = Color.White

End Sub

Private Sub RectangleShape2\_MouseMove(sender As Object, e As MouseEventArgs) Handles RectangleShape2.MouseMove

RectangleShape2.FillColor = Color.White

Label13.BackColor = Color.White

Label13.ForeColor = Color.Red

End Sub

Private Sub Label13\_MouseClick(sender As Object, e As MouseEventArgs) Handles Label13.MouseClick

Me.Close()

End Sub

Private Sub Label13\_MouseLeave(sender As Object, e As EventArgs) Handles Label13.MouseLeave

RectangleShape2.FillColor = Color.Red

RectangleShape2.BorderColor = Color.Red

Label13.BackColor = Color.Red

Label13.ForeColor = Color.White

End Sub

Private Sub Label13\_MouseMove(sender As Object, e As MouseEventArgs) Handles Label13.MouseMove

RectangleShape2.FillColor = Color.White

Label13.BackColor = Color.White

Label13.ForeColor = Color.Red

End Sub

Private Sub LabelYmin\_Click(sender As Object, e As EventArgs) Handles LabelYmin.Click

Dim mc As MatchCollection = expression.Matches(Xine.Text)

LabelYmin.Text = mc(0).ToString

End Sub

Private Sub GRA\_MouseMove(sender As Object, e As MouseEventArgs) Handles GRA.MouseMove

Me.GRA.Focus()

If tienable = 1 Then

Dim xl, yl As Double

xl = MousePosition.X - x0

yl = MousePosition.Y - y0

myXmax = (-xl / GRA.Width) \* myxl + Xxx

myXmin = (-xl / GRA.Width) \* myxl + Xmm

myYmax = (yl / GRA.Height) \* myyl + Yxx

myYmin = (yl / GRA.Height) \* myyl + Ymm

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

If lenable = 1 Then

Call math\_l()

End If

If nenable = 1 Then

Call math\_n()

End If

If niheenable = 1 Then

Call math\_nihe()

End If

End If

End Sub

Private Sub GRA\_MouseWheel(sender As Object, e As MouseEventArgs) Handles GRA.MouseWheel

If e.Delta > 0 Then

enlarge = 0.1 + enlarge

ElseIf enlarge > 0.2 Then

enlarge = enlarge - 0.1

End If

Dim num(1) As Double

Dim mc1 As MatchCollection = expression.Matches(Xine.Text)

num(0) = mc1.Count

Dim mc2 As MatchCollection = expression.Matches(Yine.Text)

num(1) = mc2.Count

TextBox2.Text = num.Min

If num.Min < 3 Then

ERRO.Text = "警告！x,y输入个数太少！"

ElseIf num(1) <> num(0) Then

ERRO.Text = "警告！x,y输入个数不同！"

Else

ERRO.Text = "正确！注：格式为a,b,c,..."

End If

If num.Min > 0 Then

Dim x(num.Min - 1), y(num.Min - 1) As Double

For i = 0 To num.Min - 1

x(i) = mc1(i).ToString

y(i) = mc2(i).ToString

Next

myXmax = (x.Max + x.Min) / 2 + ((x.Max - x.Min) / enlarge) / 2

myXmin = (x.Max + x.Min) / 2 - ((x.Max - x.Min) / enlarge) / 2

myYmax = (y.Max + y.Min) / 2 + ((y.Max - y.Min) / enlarge) / 2

myYmin = (y.Max + y.Min) / 2 - ((y.Max - y.Min) / enlarge) / 2

LabelYmax.Text = "Ymax:" + Format(Val(myYmax.ToString), "0.0")

LabelYmin.Text = "Ymin:" + Format(Val(myYmin.ToString), "0.0")

LabelXmax.Text = "Xmax:" + Format(Val(myXmax.ToString), "0.0")

LabelXmin.Text = "Xmin:" + Format(Val(myXmin.ToString), "0.0")

End If

Label17.Text = " 现在图像的放大倍数为：" + Format(Val(enlarge.ToString), "0.0")

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

If lenable = 1 Then

Call math\_l()

End If

If nenable = 1 Then

Call math\_n()

End If

If niheenable = 1 Then

Call math\_nihe()

End If

End Sub

Private Sub GRA\_Move(sender As Object, e As EventArgs) Handles GRA.Move

Me.GRA.Focus()

End Sub

Private Sub GRA\_MouseDown(sender As Object, e As MouseEventArgs) Handles GRA.MouseDown REM nonononononon

x0 = MousePosition.X

y0 = MousePosition.Y

myxl = myXmax - myXmin

myyl = myYmax - myYmin

Xxx = myXmax

Xmm = myXmin

Yxx = myYmax

Ymm = myYmin

tienable = 1

End Sub

Private Sub GRA\_MouseUp(sender As Object, e As MouseEventArgs) Handles GRA.MouseUp

tienable = 0

End Sub

Private Sub Yine\_TextChanged(sender As Object, e As EventArgs) Handles Yine.TextChanged

Dim num(1) As Double

Dim mc1 As MatchCollection = expression.Matches(Xine.Text)

num(0) = mc1.Count

Dim mc2 As MatchCollection = expression.Matches(Yine.Text)

num(1) = mc2.Count

TextBox2.Text = num.Min

numberamount = num.Min

If num.Min < 3 Then

ERRO.Text = "警告！x,y输入个数太少！"

ElseIf num(1) <> num(0) Then

ERRO.Text = "警告！x,y输入个数不同！"

Else

ERRO.Text = "正确！注：格式为a,b,c,..."

End If

wrong = 0

If num.Min > 0 Then

Dim x(num.Min - 1), y(num.Min - 1) As Double

For i = 0 To num.Min - 1

x(i) = mc1(i).ToString

y(i) = mc2(i).ToString

Next

For i = 0 To x.Count - 2

For j = i + 1 To x.Count - 1

If x(i) = x(j) Then

ERRO.Text = "警告！x输入的数中有重复！"

wrong = 1

End If

Next

Next

myXmax = (x.Max + x.Min) / 2 + ((x.Max - x.Min) / enlarge) / 2

myXmin = (x.Max + x.Min) / 2 - ((x.Max - x.Min) / enlarge) / 2

myYmax = (y.Max + y.Min) / 2 + ((y.Max - y.Min) / enlarge) / 2

myYmin = (y.Max + y.Min) / 2 - ((y.Max - y.Min) / enlarge) / 2

LabelYmax.Text = "Ymax:" + Format(Val(myYmax.ToString), "0.0")

LabelYmin.Text = "Ymin:" + Format(Val(myYmin.ToString), "0.0")

LabelXmax.Text = "Xmax:" + Format(Val(myXmax.ToString), "0.0")

LabelXmin.Text = "Xmin:" + Format(Val(myXmin.ToString), "0.0")

End If

End Sub

Private Sub Xine\_TextChanged(sender As Object, e As EventArgs) Handles Xine.TextChanged

Dim num(1) As Double

Dim mc1 As MatchCollection = expression.Matches(Xine.Text)

num(0) = mc1.Count

Dim mc2 As MatchCollection = expression.Matches(Yine.Text)

num(1) = mc2.Count

TextBox2.Text = num.Min

numberamount = num.Min

If num.Min < 3 Then

ERRO.Text = "警告！x,y输入个数太少！"

ElseIf num(1) <> num(0) Then

ERRO.Text = "警告！x,y输入个数不同！"

Else

ERRO.Text = "正确！注：格式为a,b,c,..."

End If

wrong = 0

If num.Min > 0 Then

Dim x(num.Min - 1), y(num.Min - 1) As Double

For i = 0 To num.Min - 1

x(i) = mc1(i).ToString

y(i) = mc2(i).ToString

Next

For i = 0 To x.Count - 2

For j = i + 1 To x.Count - 1

If x(i) = x(j) Then

ERRO.Text = "警告！x输入的数中有重复！"

wrong = 1

End If

Next

Next

myXmax = (x.Max + x.Min) / 2 + ((x.Max - x.Min) / enlarge) / 2

myXmin = (x.Max + x.Min) / 2 - ((x.Max - x.Min) / enlarge) / 2

myYmax = (y.Max + y.Min) / 2 + ((y.Max - y.Min) / enlarge) / 2

myYmin = (y.Max + y.Min) / 2 - ((y.Max - y.Min) / enlarge) / 2

LabelYmax.Text = "Ymax:" + Format(Val(myYmax.ToString), "0.0")

LabelYmin.Text = "Ymin:" + Format(Val(myYmin.ToString), "0.0")

LabelXmax.Text = "Xmax:" + Format(Val(myXmax.ToString), "0.0")

LabelXmin.Text = "Xmin:" + Format(Val(myXmin.ToString), "0.0")

End If

End Sub

Private Sub Button1\_Click\_2(sender As Object, e As EventArgs) Handles Button1.Click

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

nenable = 0

niheenable = 0

lenable = 0

End Sub

Private Sub Labelup\_Click(sender As Object, e As EventArgs) Handles Labelup.Click REM now 12.19

Dim myyl, myxl As Double

myxl = myXmax - myXmin

myyl = myYmax - myYmin

myXmax = (0) \* myxl + myXmax

myXmin = (0) \* myxl + myXmin

myYmax = (-0.1) \* myyl + myYmax

myYmin = (-0.1) \* myyl + myYmin

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

If lenable = 1 Then

Call math\_l()

End If

If nenable = 1 Then

Call math\_n()

End If

If niheenable = 1 Then

Call math\_nihe()

End If

End Sub

Private Sub Labeldown\_Click(sender As Object, e As EventArgs) Handles Labeldown.Click

Dim myyl, myxl As Double

myxl = myXmax - myXmin

myyl = myYmax - myYmin

myXmax = (0) \* myxl + myXmax

myXmin = (0) \* myxl + myXmin

myYmax = (0.1) \* myyl + myYmax

myYmin = (0.1) \* myyl + myYmin

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

If lenable = 1 Then

Call math\_l()

End If

If nenable = 1 Then

Call math\_n()

End If

If niheenable = 1 Then

Call math\_nihe()

End If

End Sub

Private Sub Labelleft\_Click(sender As Object, e As EventArgs) Handles Labelleft.Click

Dim myyl, myxl As Double

myxl = myXmax - myXmin

myyl = myYmax - myYmin

myXmax = (0.1) \* myxl + myXmax

myXmin = (0.1) \* myxl + myXmin

myYmax = (0) \* myyl + myYmax

myYmin = (0) \* myyl + myYmin

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

If lenable = 1 Then

Call math\_l()

End If

If nenable = 1 Then

Call math\_n()

End If

If niheenable = 1 Then

Call math\_nihe()

End If

End Sub

Private Sub Labelright\_Click(sender As Object, e As EventArgs) Handles Labelright.Click

Dim myyl, myxl As Double

myxl = myXmax - myXmin

myyl = myYmax - myYmin

myXmax = (-0.1) \* myxl + myXmax

myXmin = (-0.1) \* myxl + myXmin

myYmax = (0) \* myyl + myYmax

myYmin = (0) \* myyl + myYmin

Dim gr1 As Graphics

gr1 = GRA.CreateGraphics

gr1.Clear(Color.White)

If lenable = 1 Then

Call math\_l()

End If

If nenable = 1 Then

Call math\_n()

End If

If niheenable = 1 Then

Call math\_nihe()

End If

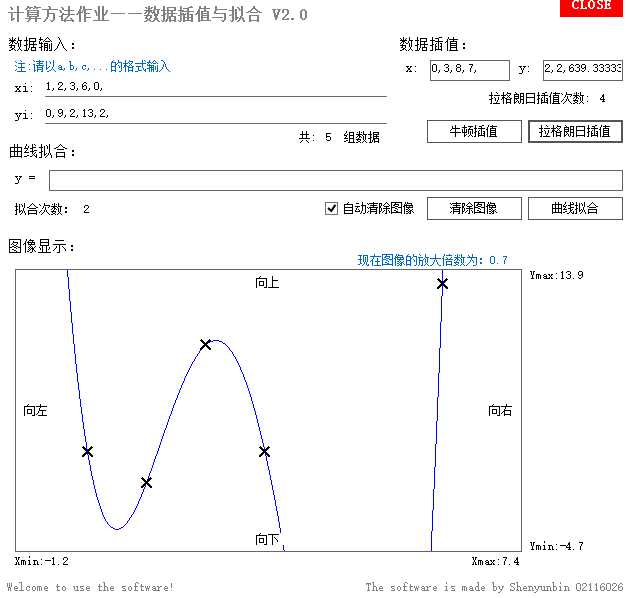
End Sub

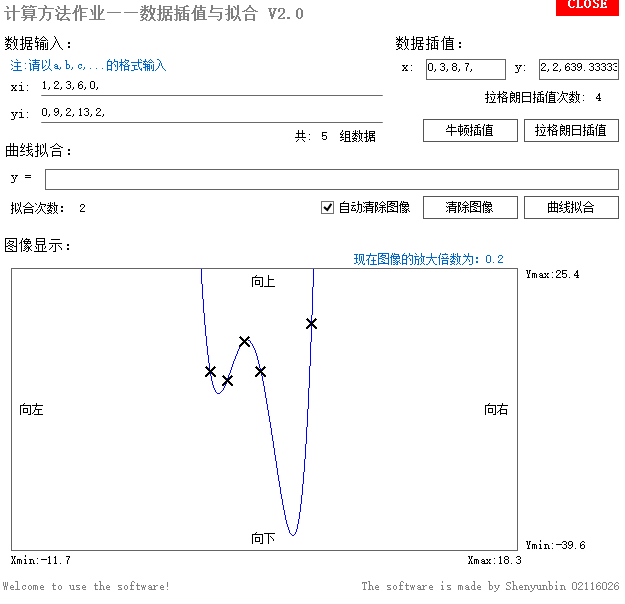
End Class

# 4运行结果（软件功能）

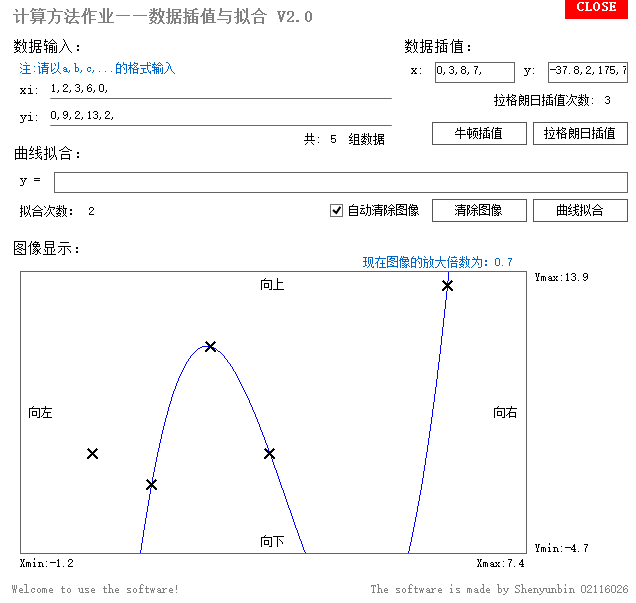
**功能一：拉格朗日插值及输出插值曲线**

**4次插值**

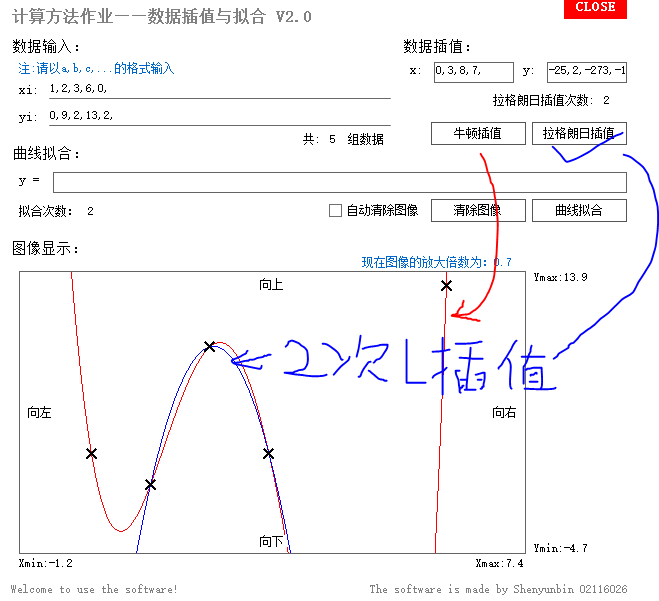




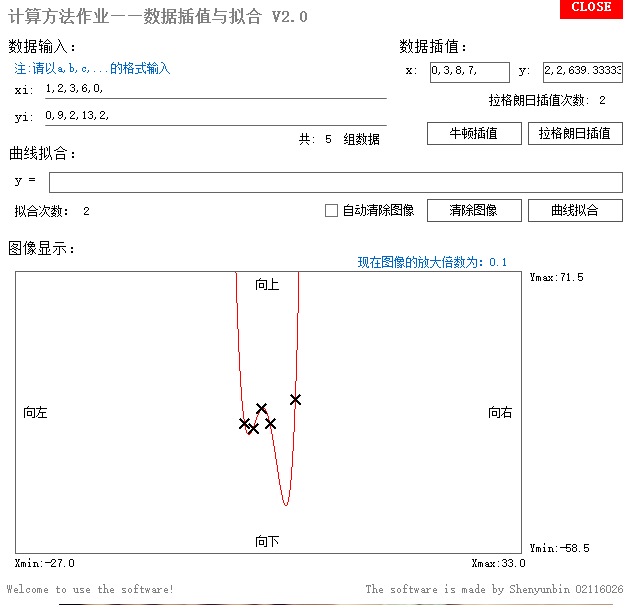
**3次插值**

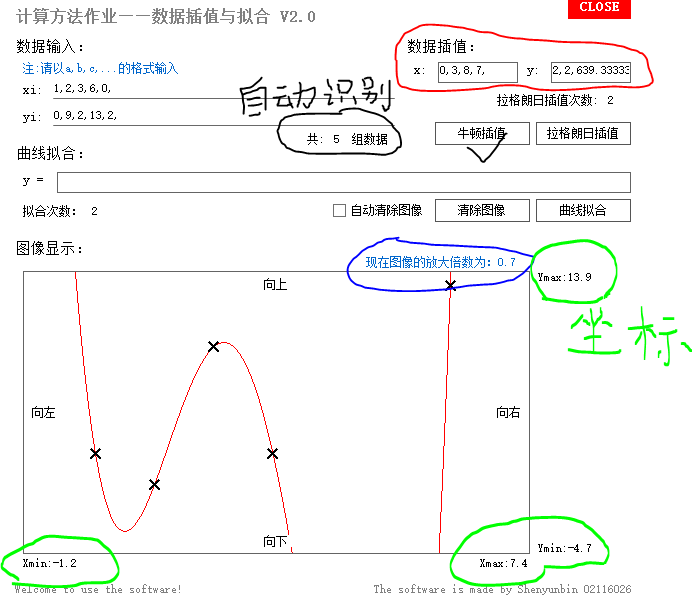


**2次插值与牛顿插值对比**



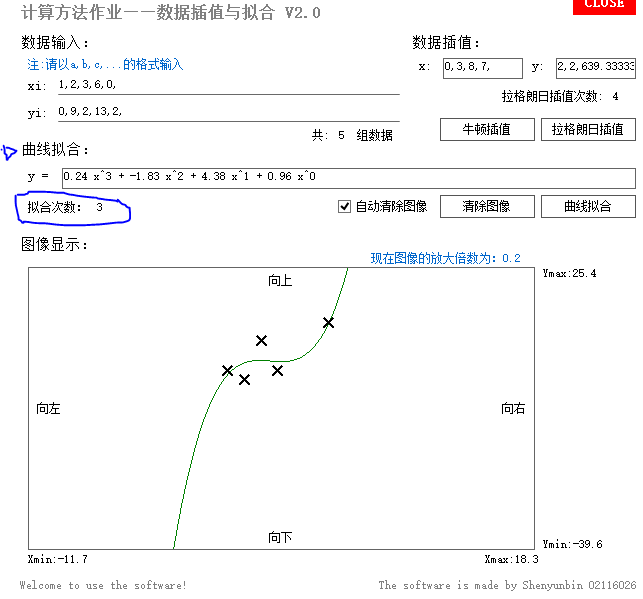
**功能二：牛顿插值及输出插值曲线**



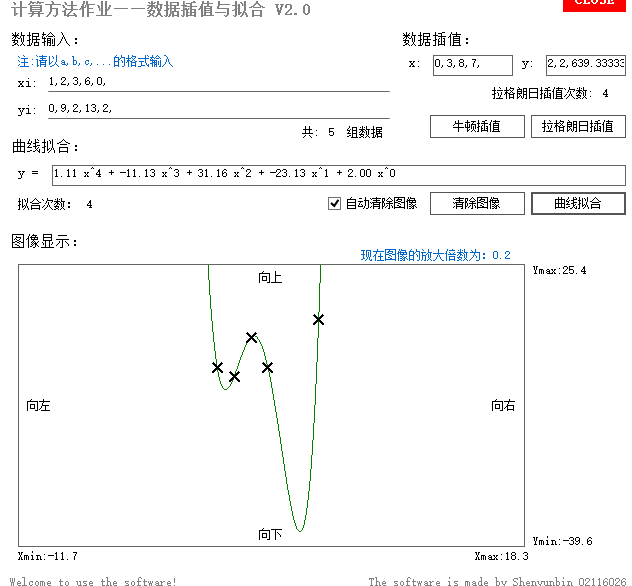


**功能三：曲线拟合及输出拟合曲线**

**3次拟合**

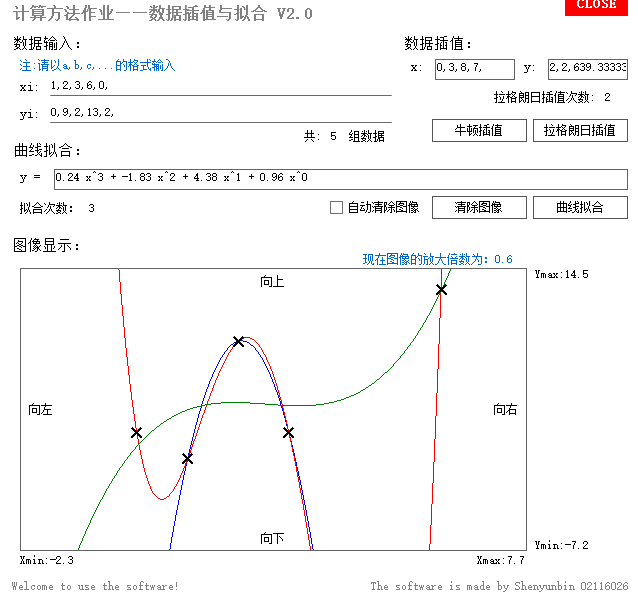


**4次拟合**



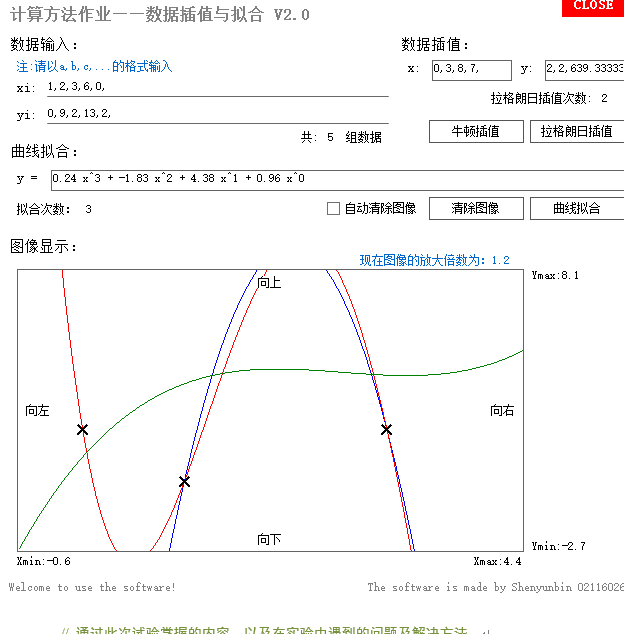
**功能四：多曲线比较及横纵坐标显示**

**牛顿插值、2次拉格朗日、3次拟合综合图像**



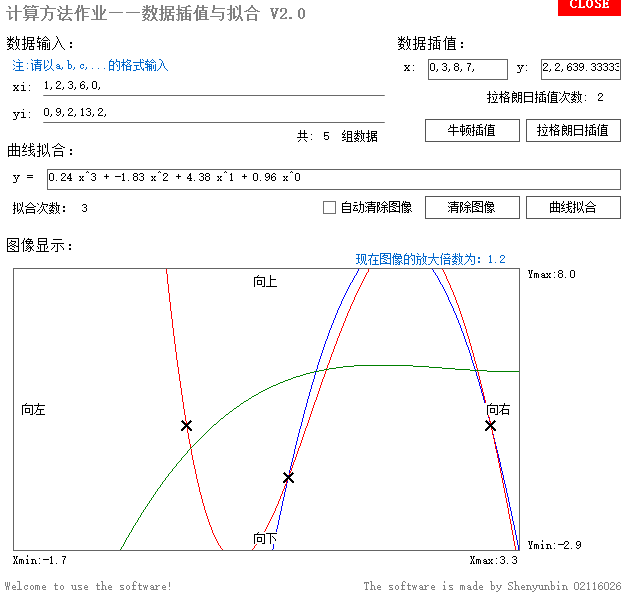
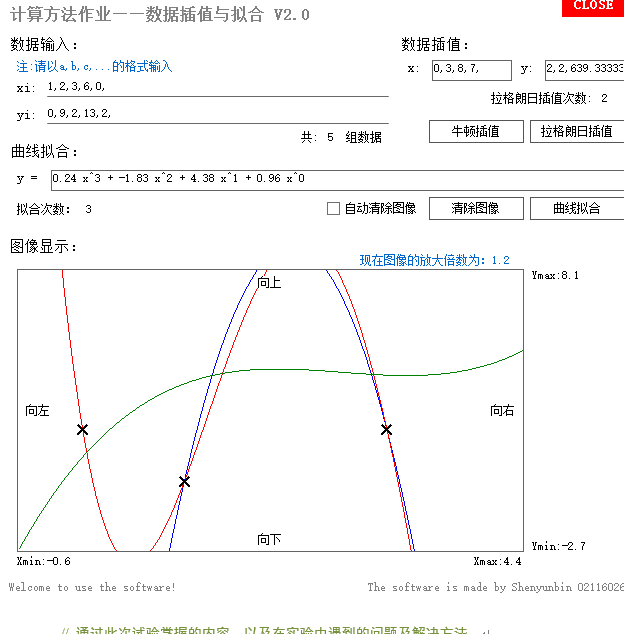
**功能五：曲线图像放大缩小**

**将前面图像放大后**



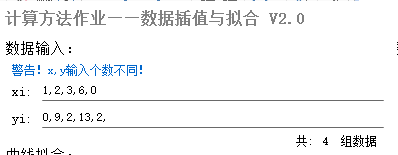
**功能六：曲线拖动**

**两张拖动前后照片对比**

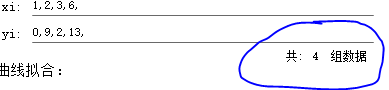


**功能七：完备的提示功能**

1. **输入个数不同提示**



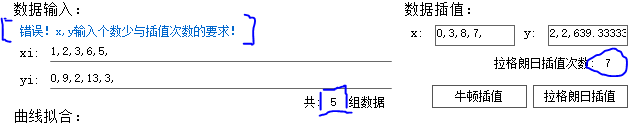
1. **输入数组数提示**

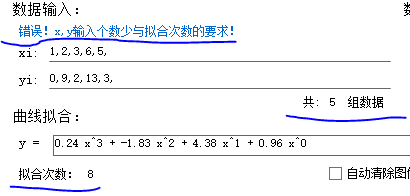


1. **X输入重复提示**

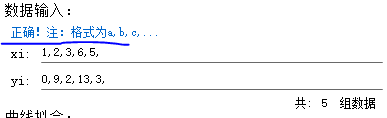


1. **输入数组数达不到插值拟合次数提示**

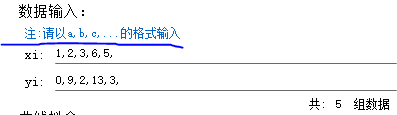




1. **正确输入提示**



1. **输入格式提示**



# 5总结

// 通过此次试验掌握的内容，以及在实验中遇到的问题及解决方法。

# 6参考资料

// 学习相关理论、编写程序及为了完成实验查阅的书籍和文献

// 英文参考文献格式

// 期刊

// [序号] 主要责任者. 文献题名[J]. 刊名, 年, 卷(期): 起止页码.

// 专著、论文集、学位论文、报告

// [序号] 主要责任者. 文献题名[文献类型标识]. 出版地: 出版者, 出版年. 起止页码.