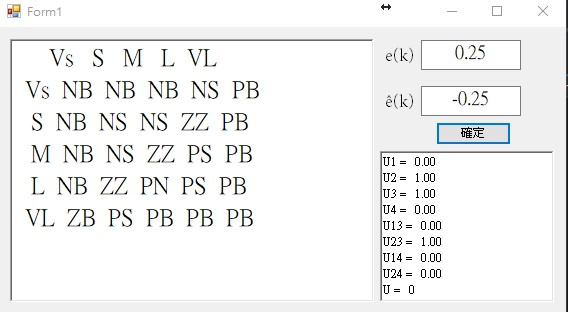
房志剛-1103105345-W04-20170413

結果



程式碼:

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public String[,] table ={

{ "NB", "NB", "NB", "NS", "PB" } ,

{ "NB", "NS", "NS", "ZZ", "PB" } ,

{ "NB", "NS", "ZZ", "PS", "PB" } ,

{ "NB", "ZZ", "PN", "PS", "PB" } ,

{ "ZB", "PS", "PB", "PB", "PB" } };

public Double[] Single\_pole = { -10, -5, 0, 5, 10 };

private void Form1\_Load(object sender, EventArgs e)

{

int N = 5;

richTextBox\_table\_out.AppendText(" ");

for (int i = 0; i < N; i++)

{

richTextBox\_table\_out.AppendText(String.Format("{0,4}", RankCovert(i)));

}

richTextBox\_table\_out.AppendText("\n");

for (int i = 0; i < N; i++)

{

richTextBox\_table\_out.AppendText(String.Format("{0,4}", RankCovert(i)));

for (int j = 0; j < N; j++)

{

richTextBox\_table\_out.AppendText(String.Format("{0,4}", table[i, j]));

}

richTextBox\_table\_out.AppendText("\n");

}

}

private int Rank\_D(Double x)

{

Double[] num = { 0, 0.25, 0.5, 0.75, 1 };

Double[] distanse = { 0, 0.25, 0.5, 0.75, 1 };

for (int i = 0; i < 5; i++)

{

distanse[i] = Math.Abs(x - num[i]);

}

Double min = distanse[0];

int index = 0;

for (int i = 1; i < 5; i++)

{

if (distanse[i] < min)

{

min = distanse[i];

index = i;

}

}

return (index + 1);

}

private String Rank\_NP(Double x)

{

Double[] num = { -10, -5, 0, 5, 10 };

Double[] distanse = new double[5];

for (int i = 0; i < 5; i++)

{

distanse[i] = Math.Abs(x - num[i]);

}

Double min = distanse[0];

int index = 0;

for (int i = 1; i < 5; i++)

{

if (distanse[i] < min)

{

min = distanse[i];

index = i;

}

}

switch (index)

{

case 0:

return "NB";

case 1:

return "NS";

case 2:

return "ZZ";

case 3:

return "PS";

case 4:

return "PB";

default:

return "PB";

}

}

private Double Rank\_NP(String str)

{

switch (str)

{

case "NB":

return Single\_pole[0];

case "NS":

return Single\_pole[1];

case "ZZ":

return Single\_pole[2];

case "PS":

return Single\_pole[3];

case "PB":

return Single\_pole[4];

default:

return Single\_pole[0];

}

}

private String RankCovert(int i)

{

switch (i)

{

case 0: return "Vs";

case 1: return "S";

case 2: return "M";

case 3: return "L";

case 4: return "VL";

default: return "";

}

}

private void button\_confirm\_Click(object sender, EventArgs e)

{

if (CheckRange())

{

CalculateMF();

}

else

{

MessageBox.Show("數值範圍錯誤", "錯誤");

}

}

private void CalculateMF()

{

Double[] Y2 = { 1, 0, 1, 0, 1 };

Double[] Y1 = { 0, 1, 0, 1, 0 };

Double[] X = { -1, -0.25, 0, 0.25, 1 };

Double e = Convert.ToDouble(textBox\_X1.Text);

Double e\_dot = Convert.ToDouble(textBox\_X2.Text);

int zone\_e = -1, zone\_e\_dot = -1;

Double u1, u2, u3, u4;

for (int i = 0; i < X.Length - 1; i++)

{

if (e >= X[i] && e < X[i + 1]) zone\_e = i;

if (e\_dot >= X[i] && e\_dot < X[i + 1]) zone\_e\_dot = i;

}

if (Y1[zone\_e] > Y1[zone\_e + 1])

{

u1 = (e - X[zone\_e]) / 0.25;

u2 = 1 - u1;

}

else

{

u1 = 1 - (e - X[zone\_e]) / 0.25;

u2 = 1 - u1;

}

if (Y2[zone\_e\_dot] > Y2[zone\_e\_dot + 1])

{

u3 = (e\_dot - X[zone\_e\_dot]) / 0.25;

u4 = 1 - u3;

}

else

{

u3 = 1 - (e\_dot - X[zone\_e\_dot]) / 0.25;

u4 = 1 - u3;

}

Double u13 = (u1 \* u3), u23 = (u2 \* u3), u14 = (u1 \* u4), u24 = (u2 \* u4);

richTextBox\_MF\_out.Text += "U1 = " + u1.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U2 = " + u2.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U3 = " + u3.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U4 = " + u4.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U13 = " + u13.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U23 = " + u23.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U14 = " + u14.ToString("0.00") + "\n";

richTextBox\_MF\_out.Text += "U24 = " + u24.ToString("0.00") + "\n";

Double U = (u13 \* Rank\_NP(table[zone\_e\_dot, zone\_e]) + u23 \* Rank\_NP(table[zone\_e\_dot, zone\_e]) + u14 \* Rank\_NP(table[zone\_e\_dot + 1, zone\_e]) + u24 \* Rank\_NP(table[zone\_e\_dot + 1, zone\_e + 1])) / (u13 + u23 + u14 + u24);

richTextBox\_MF\_out.Text += "U = " + U;

}

private bool CheckRange()

{

try

{

if (Convert.ToDouble(textBox\_X1.Text) < -1)

return false;

if (Convert.ToDouble(textBox\_X1.Text) > 1)

return false;

if (Convert.ToDouble(textBox\_X2.Text) < -1)

return false;

if (Convert.ToDouble(textBox\_X2.Text) > 1)

return false;

}

catch (Exception e)

{

return false;

}

return true;

}

}