

Parameters required for designing of contour bund by using visual studio

M. M. Hyalij¹, P. R. Kolhe², V. D. Jadhav³, A. R. Patil⁴, H. N. Bhang⁵, S. M. Bhagwat⁶, R. S. Mhatre⁷

¹M.Tech, College of Agriculture Engineering and Technology, Dr. BSKKV, Dapoli Maharashtra, India

²Associate Professor (CAS), College of Agriculture Engineering and Technology, Dr. BSKKV, Dapoli Maharashtra, India

³Technical Assistant, AKMU, Dr. BSKKV, Dapoli Maharashtra, India.

⁴Senior Research Assistant, AKMU, Dr. BSKKV, Dapoli Maharashtra, India.

⁵Assistant professor, College of Agriculture Engineering and Technology, Dr. BSKKV, Dapoli Maharashtra, India.

⁶M.Tech, College of Agriculture Engineering and Technology, Dr. BSKKV, Dapoli Maharashtra, India.

⁷M.Tech, College of Agriculture Engineering and Technology, Dr. BSKKV, Dapoli Maharashtra, India.

Submitted: 01-11-2022

Accepted: 12-11-2022

ABSTRACT-

Contour Bunding are disbursed in several components of India- notably in geographical region, Gujarat, Tamil Nadu, Karnataka and province. It consists of building material embankments across the slope of the land, following the contour as closely as attainable. A series of such bunds divide the realm into strips and act as barriers to the flow of water, so reducing the number and rate of the runoff. It is important soil and water conservation structure and most adoptable in various region. The designing calculations of contour bunds are time consuming so by keeping this in mind in this work designing of the contour bund is done by using visual studio. In this study work by using the programming the design procedure is making for different individual designing steps. After running the program the designing parameters are calculated by simply putting the values of different components.

Key Words: Visual Studio, Technology, Agriculture, Computerized, Soil and Water Conservation Engineering.

I. INTRODUCTION

Design Criteria for Bunds:

The following factors are to be thought of whereas developing criterion for contour bunds

1.1 Allowable submergence of land:

The amount of land submerged thanks to unfinished and length of unfinished can have an effect on crops. So quantity of waste weir and also the amount of land to be submerged ought to be

determined by the cropping observe to be followed and also the infiltration rate for the soil.

1.2 Moisture Conservation:

For paddy lands it's fascinating to store all the rain water for the employment of the plants. So the bunds ought to be of such dimensions on allow no runoff. For different crops, the capability of the bund ought to be determined by the common consumptive use of the crop planned and also the most length of dry amount in season. The heights of waste weirs ought to be specified the bunds store simply decent water to fulfill demand of crop.

1.3 Critical Length:

Another approach in fixing the spacing of bunds by determinative the vital length of land between adjacent bunds. Increase in geographical area will increase each speed and quantity of runoff gathering in narrow channel. However the vital length approach, the try is to area bunds in such the way that the speed remains at intervals non-erosive limit.

1.4 Seepage consideration:

While coming up with the bund cross-section, the oozing through bunds thanks to accumulation of water behind it ought to be taken into consideration. The oozing rate is affected primarily by the pinnacle of water impounded, the facet slopes of the sure and also the porousness of the soil.

1.5 Location of contour bund:

Moderate slopes (5–10%) with light-weight or medium soil texture and fewer than 700 mm of rain per year.

II. EXISTING SYSTEM

The Existing system is totally manual. The coming up with of contour bund contains completely different formulas calculate the assorted values of their parameters. These formulas are terribly complicated to calculate with the employment of calculators. It's strictly paper primarily based to store the records of the quondam calculated values of these formulas. The calculations are through with calculator by coming into every worth of formula manually within the calculator.

2.1 Limitations in Present System

There are sure limitations in system, that are as follow: -

- 1) There are numerous steps or formulae concerned in to coming up with of contour bund, therefore it's tough to handle manually.
- 2) Manually calculations are tough to estimating numerous parameters within the coming up with of contour and additionally longer needed.
- 3) For recording purpose, it ought to be note in some paper, so probabilities of losing the information are will increase.
- 4) It is tough to look the recent record of knowledge.
- 5) Manually calculations embody a lot of errors with less accuracy.
- 6) To need the person power for calculation on the sphere and create it secure.

III. PROPOSED SYSTEM:

The projected system is deals with the practicality of calculations of coming up with parameters.

To remove the complexities, errors and difficulties the projected system is developed. These difficulties will come about at the time of managing the roles at establishment. these days the medical aid is directly go together with cybernation, that the students, researchers are returning below the one roof to find out regarding medical aid and cybernation. Calculation of coming up with parameters formulas manually on the paper are terribly complicated technique to search out the correct worth of the particular part. Since these works will manually so it takes longer and there are probabilities of errors like wrong details, wrong

worth at wrong place etc. So to scale back these faults "To calculation of coming up with parameters of contour bund victimization Visual Studio" has been developed. The projected system deals with all the activities that are done by the granger for calculation of coming up with parameters. This paper is all regarding however will info communication technology scale back the manual work load and provides the correct results of the correct formulas.

3.1 Advantages of Proposed System:

1. Calculations are done very fast, efficient and it saves the time.
2. No need to data note down on the paper because it stores digitally in computer.
3. Modified the data as per requirement.
4. System should be window based, user-friendly and efficient.
5. Accuracy can be maintained.

IV. MATERIALS

This chapter deals with the methodology used to calculate designing parameters required for contour bund designing by using Visual Studio. It includes different types of formulae and theoretical considerations those are used for calculation. It also encapsulates the configuration of the system and information to design the system.

V. SYSTEM REQUIREMENTS

- Windows 7 ultimate
- Installed memory (RAM): 4.00 GB (3.39 GB usable)
- System type: 32 – bit Operating System

5.1 About the platform

- Visual Studio 2022
- Copyright (c) 1990, 1992 by Borland International, Inc.

VI. FORMULAS FOR CALCULATION OF DESIGNING PARAMETERS

1. Vertical Interval (VI) = $0.15s + 0.6$ (m)

Where,

VI = Vertical interval, m
S = Original land slope, %

2. Horizontal Interval = $\frac{VI \times 100}{s}$

Where

HI = Horizontal interval, m
S = Original land slope, %

3. Runoff volume to be stored (R_v) = $P_e \times A$ (m^3)
 Where
 P_e = Excess rainfall depth or surface runoff, m
 A = Area of watershed behind two bunds, m^2 .

```
{
  Form3 frm3 = new Form3();
  frm3.Show();
}
```

4. Storage volume (S_v) = $A \times L$ (m^2)
 Where
 A = Area of watershed behind two bunds, m^2
 L = Length of bund behind which the runoff is stored, m

```
private void button3_Click(object sender,
EventArgs e)
{
  Form4 frm4 = new Form4();
  frm4.Show();
}
```

5. Bund cross section
 a) Total height of bund (H) = ($d+20\%$ of d) (m)
 b) Base width (B) = ($nd+4d$) (m)
 c) Top width (T) = ($B-2nH$) (m)

```
private void button4_Click(object sender,
EventArgs e)
{
  Form5 frm5 = new Form5();
  frm5.Show();
}
```

6. Earth work due to bunding
 a) Length of contour bund per ha (L) = $\frac{10^4}{HI}$ for main bund
 b) Earthwork per ha = $\left[\frac{1}{2} \times (B + T) \times H \right] \times L$

```
}
private void button5_Click(object sender,
EventArgs e)
```

VII. SOURCE CODE FOR CALCULATION OF DESIGNING PARAMETERS

Home Page

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contour_Bund
{
  public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }

    private void button1_Click(object sender,
EventArgs e)
    {
      Form2 frm2 = new Form2();
      frm2.Show();
    }

    private void button2_Click(object sender,
EventArgs e)
```

```
{
  Form6 frm6 = new Form6();
  frm6.Show();
}

private void button6_Click(object sender,
EventArgs e)
{
  Form7 frm7 = new Form7();
  frm7.Show();
}

private void Form1_Load(object sender,
EventArgs e)
{
}

private void textBox1_TextChanged(object sender,
EventArgs e)
{
}
}
```

Vertical Interval (VI)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```

```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace Contour_Bund
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }
    }
}
```

```
private void Calculate_Click(object sender,
EventArgs e)
{
    double s, VI;
    s = Convert.ToDouble(textBoxS.Text);
    VI = 0.15 * s + 0.6;
    textBoxVI.Text = VI.ToString();
}
```

```
private void label2_Click(object sender,
EventArgs e)
{
}
}
```

Horizontal Interval (HI)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace Contour_Bund
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }
    }
}
```

```
private void button1_Click(object sender,
EventArgs e)
{
}
```

```
double VI, s, HI;
VI = Convert.ToDouble(textBoxVI.Text);
s = Convert.ToDouble(textBoxS.Text);
HI = VI * 100 / s;
textBoxHI.Text = HI.ToString();
}

private void flowLayoutPanel1_Paint(object
sender, PaintEventArgs e)
{
}
}
```

Runoff volume to be stored (Rv)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace Contour_Bund
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }
    }
}
```

```
private void Calculate_Click(object sender,
EventArgs e)
{
    Double Pe, A, Rv;
    Pe = Convert.ToDouble(textBoxPe.Text);
    A = Convert.ToDouble(textBoxA.Text);
    Rv = Pe * A;
    textBoxRv.Text = Rv.ToString();
}
```

```
private void label4_Click(object sender,
EventArgs e)
{
}
}
```

Storage volume (Sv)

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contour_Bund
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();

            private void textBox1_TextChanged(object sender, EventArgs e)
            {

            }

            private void button1_Click(object sender, EventArgs e)
            {
                double A, L, Sv;
                A = Convert.ToDouble(textBoxA.Text);
                L = Convert.ToDouble(textBoxL.Text);
                Sv = A * L;
                textBoxSv.Text = Sv.ToString();
            }

            private void Form5_Load(object sender, EventArgs e)
            {

            }
        }
    }
}
```

Bund Cross Section

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contour_Bund
{
```

```
    public partial class Form6 : Form
    {
        public Form6()
        {
            InitializeComponent();

            private void textBox2_TextChanged(object sender, EventArgs e)
            {

            }

            private void label8_Click(object sender, EventArgs e)
            {

            }

            private void textBox8_TextChanged(object sender, EventArgs e)
            {

            }

            private void label7_Click(object sender, EventArgs e)
            {

            }

            private void textBox7_TextChanged(object sender, EventArgs e)
            {

            }

            private void label4_Click(object sender, EventArgs e)
            {

            }

            private void Calculate_Click(object sender, EventArgs e)
            {
                double d, f, H;
                d = Convert.ToDouble(textBoxd.Text);
                f = Convert.ToDouble(textBoxf.Text);
                H = d + (20 * d)/100;
                textBoxH.Text = H.ToString();
            }

            private void button2_Click(object sender, EventArgs e)
            {

            }
        }
    }
}
```

```
{
    double n, d, B;
    n = Convert.ToDouble(textBoxn.Text);
    d = Convert.ToDouble(textBoxd.Text);
    B = n * d + 4 * d;
    textBoxB.Text = B.ToString();
}

private void button3_Click(object sender,
EventArgs e)
{
    double B, n, H, T;
    B = Convert.ToDouble(textBoxB.Text);
    n = Convert.ToDouble(textBoxn.Text);
    H = Convert.ToDouble(textBoxH.Text);
    T = B - 2 * n * H;
    textBoxT.Text = T.ToString();
}

private void label12_Click(object sender,
EventArgs e)
{
}

private void label1_Click(object sender,
EventArgs e)
{
}

private void label2_Click(object sender,
EventArgs e)
{
}

private void label3_Click(object sender,
EventArgs e)
{
}

private void label6_Click(object sender,
EventArgs e)
{
}

private void Form6_Load(object sender,
EventArgs e)
{
    private void textBoxd_TextChanged(object
sender, EventArgs e)
    {
    }

    private void textBoxf_TextChanged(object
sender, EventArgs e)
    {
    }

    private void textBoxH_TextChanged(object
sender, EventArgs e)
    {
    }

    private void textBoxn_TextChanged(object
sender, EventArgs e)
    {
    }

    private void textBox9_TextChanged(object
sender, EventArgs e)
    {
    }

    private void textBoxT_TextChanged(object
sender, EventArgs e)
    {
    }

    private void label5_Click(object sender,
EventArgs e)
    {
    }
}

Earthwork due to bunding
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace Contour_Bund
{
    public partial class Form7 : Form
    {
        public Form7()
        {
            InitializeComponent();

            private void button1_Click(object sender,
            EventArgs e)
            {
                double HI, L;
                HI = Convert.ToDouble(textBoxHI.Text);
                double z = Math.Pow(10, 4);
                L = z / HI;
                textBoxL.Text = L.ToString();
            }

            private void button2_Click(object sender,
            EventArgs e)
            {
                double B, T, H, L, E;
                B = Convert.ToDouble(textBoxB.Text);
                T = Convert.ToDouble(textBoxT.Text);
                H = Convert.ToDouble(textBoxH.Text);
                L = Convert.ToDouble(textBoxL.Text);
                E = 0.5 * (B + T) * H * L;
                textBoxE.Text = E.ToString();
            }

            private void Form7_Load(object sender,
            EventArgs e)
            {
            }

            private void label1_Click(object sender,
            EventArgs e)
            {
            }

            private void label2_Click(object sender,
            EventArgs e)
            {
            }

            private void label5_Click(object sender,
            EventArgs e)
            {
            }

            private void label4_Click(object sender,
            EventArgs e)
            {
            }

            private void label6_Click(object sender,
            EventArgs e)
            {
            }

            private void label3_Click(object sender,
            EventArgs e)
            {
            }
        }
    }
}
```

VIII. RESULT AND DISCUSSION

This chapter deals with the results obtained by calculation of design parameters required for design contour bund formulae using Visual Studio. The programme is prepared in the C# language, Windows Form Application, Visual Studio 2022.

8.1 Steps to be followed

- 1) Enter the option from 1 to 9 serially.
- 2) Give the input values.
- 3) After pressing enter key, output will be displayed.
- 4) Again press the enter key to switch over to the main page.
- 5) Follow the same procedure to design the different parameters.



Fig 8.1 Main Page

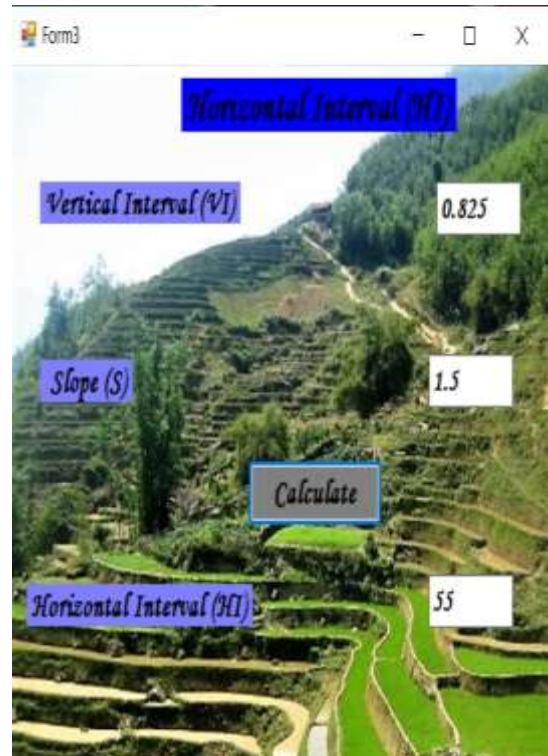


Fig 8.3 Horizontal Interval (HI)

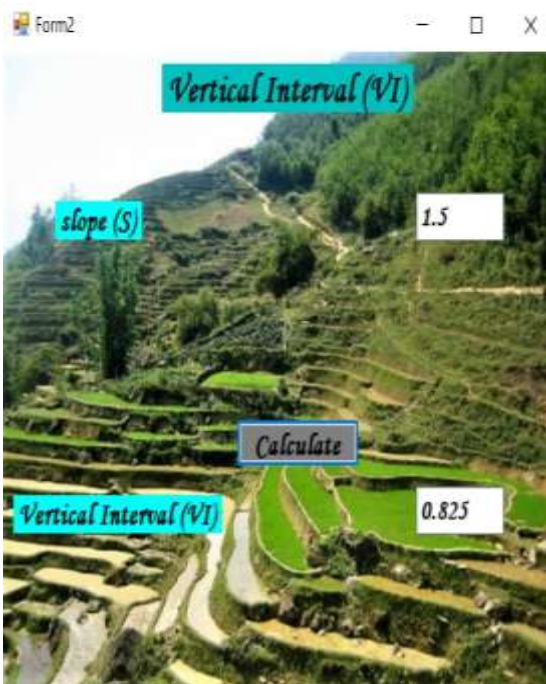


Fig 8.2 Vertical Interval (VI)

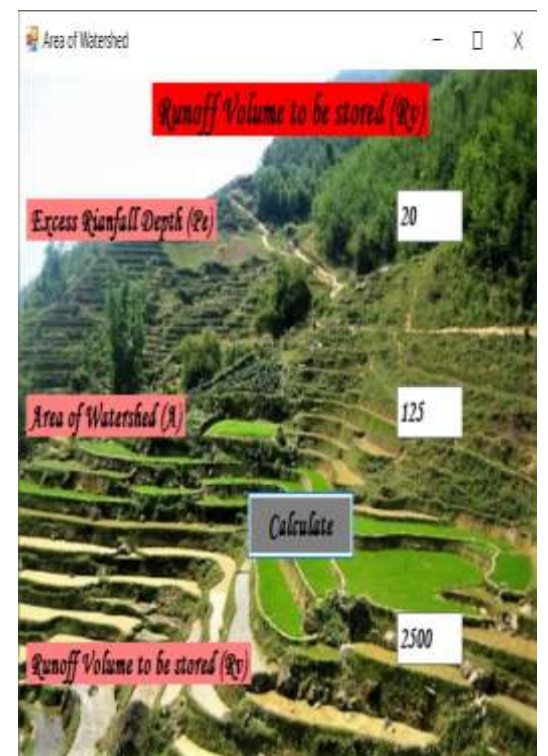


Fig 8.4 Runoff Volume to be stored (Rv)



Fig 8.5 Storage volume (Sv)



Fig 8.7 Earthwork due to bunding



Fig 8.6 Bund cross section

VI. CONCLUSIONS

The designing parameters for contour bund were accurately determined. Thus, this is time consuming. So by using the software windows form application, Visual Studio 2022; project work was done accurately and in very less time. Thus software found to be beneficial for calculate the designing parameters required for contour bund design.

The calculations were done accurately using windows form application.

It is user friendly.

It is time efficient.

REFERENCES

- [1]. Balagurusamy E. 1992. Programming in ANSI C. Tata.MC Grew-Hill Publishing Company Limited, New Delhi.
- [2]. Michael A. M. and Ojha T. P., 2020. Principles of agricultural engineering. Jain brothers, 2.
- [3]. Suresh R., 2020. Soil and water conservation engineering. Standard publishers and distributors.
- [4]. Tan A. 1999. C programming for engineers and computer science. McGraw-Hill, New York.