NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

Submission of project report for the evaluation of the final year project

Titled

Resolving Power of Optical Instruments using Finite Element Method

Under the guidance of:
Prof. S. K. Pramanik
Dept. of Electronics and Communication Engineering
NIT, Rourkela.

Submitted by:
Rajdeep Karmakar
10307005
B.Tech IV,
Dept. of Electronics and Instrumentation Engg.
NIT, Rourkela

Submitted by:
Nabin Bismay Prakash Minz
10307027
B.Tech IV,
Dept. of Electronics and Instrumentation Engg.
NIT, Rourkela

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

Submission of project report for the evaluation of the final year project

Titled

Resolving Power of Optical Instruments using Finite Element Method

Under the guidance of:
Prof. S. K. Pramanik
Dept. of Electronics and Communication Engineering
NIT, Rourkela.

Submitted by:
Rajdeep Karmakar
10307005
B.Tech IV,
Dept. of Electronics and Instrumentation Engg.
NIT, Rourkela

Submitted by:
Nabin Bismay Prakash Minz
10307027
B.Tech IV,
Dept. of Electronics and Instrumentaion Engg.
NIT, Rourkela

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, “RESOLVING POWER OF OPTICAL INSTRUMENTS USING FINITE ELEMENT METHOD” submitted by Sri RAJDEEP KARMAKAR and Sri NABIN BISMAY PRAKASH MINZ in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Electronics and Instrumentation Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. S.K. Pramanik
Dept. of Electronics and
Communication Engineering
NIT, Rourkela.

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and indebtedness to Dr. S. K. Pramanik, Department of Electronics and Communication Engineering, N.I.T Rourkela for introducing the present topic and for his inspiring guidance, constructive criticism and valuable suggestion throughout this project work.

I would like to express my gratitude to Dr. G. Panda (Head of the Department) for his constant support and encouragement. I am also thankful to all staff members of Department of Electronics and Communication Engineering NIT Rourkela.

Date:

Rajdeep Karmakar
Roll: 10307005

Nabin Bismay Prakash Minz
Roll: 10307027

Abstract

The aim of the project is to study and simulate the resolving power of various optical instruments namely,

a) Prism

b) Telescope

c) Microscope

d) Plane Diffraction Grating

The provided simulation takes in various tasks in various parameters to calculate the resolving power of the optical instruments and also conveniently simulates their behavior.

To calculate the resolving power of the optical instruments such as prism,telescope or microscope a special technique known as "FINITE ELEMENT METHOD" has been used. Basically in this method the convex lens is divided into several segments and resolving power of each segment is computed separately, details of which is provided in the concerned topic.

Finally a software model has been developed for prism, single convex lens and combination of lenses to study and show the required behavior. This software model is developed using Visual C++.

CONTENTS

Chapter 1

GENERAL INTRODUCTION

1.1 INTRODUCTION:

1.1.1 RESOLVING POWER OF OPTICAL INSTRUMENTS

The ability of an optical instrument, expressed in numerical measure, to resolve the image of two nearby points is termed as its resolving power. In the case of a prism or a grating the term resolving power is referred to the ability of the prism or grating to resolve two nearby spectral lines so that the two lines can be viewed or photographed as separate lines.

1.1.2 BASIC CRITERION FOR RESOLUTION

To express the resolving power of an optical instrument as a numerical value, Lord Rayleigh proposed an arbitrary criterion. According to him two nearby images are said to be resolved if the position of the central maximum of one coincides with the first secondary minimum of the other and vice-versa. The same criterion can be conveniently applied to calculate the resolving power of a telescope, microscope, grating, prism, etc.

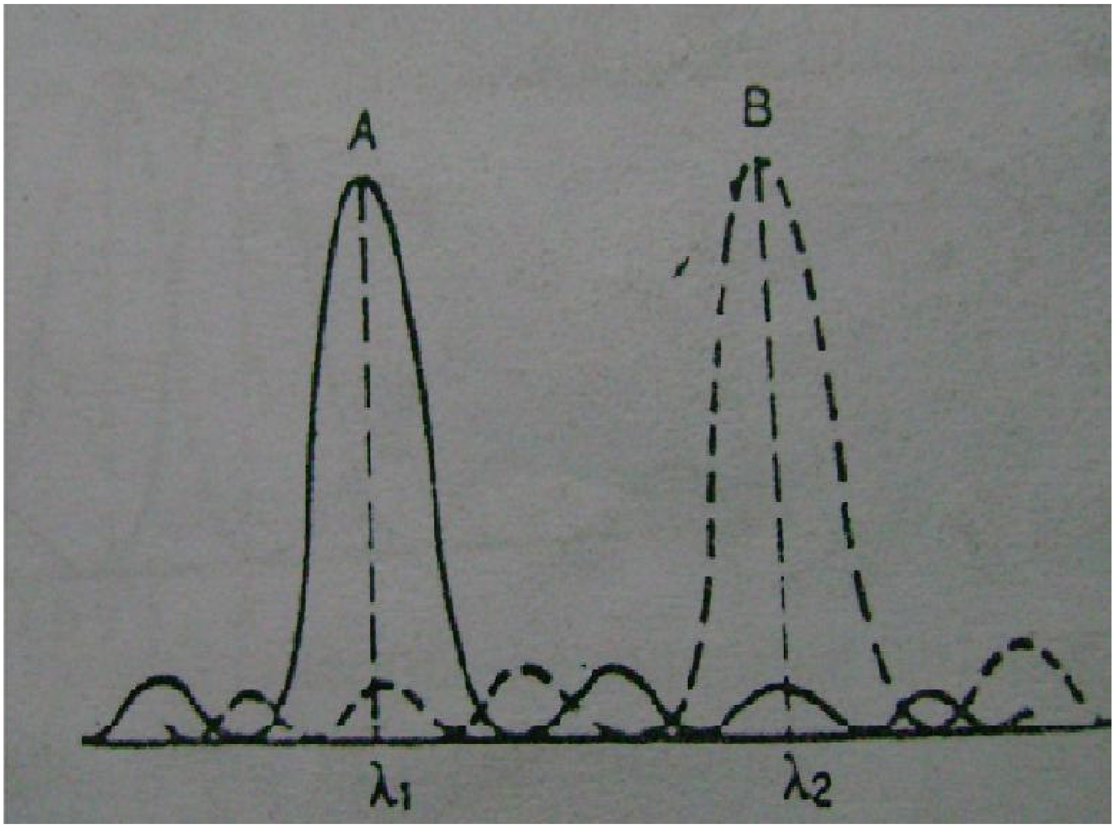


Fig-A

In Fig-A, A and B are the central maxima of the diffraction patterns of two spectral lines of wavelength λ_1 and λ_2. The difference in the angle of diffraction is large and two images can be seen as separate ones. The angle of diffraction is greater than the angle of diffraction corresponding to the first minimum at the sight of A. Hence the two spectral lines will appear well resolved.

1.2 RESOLVING POWER OF PRISM

Resolving power signifies the ability of the instrument to form separate images of two neighboring wavelengths λ and λ+dλ in the wavelength region λ.

In the given Figure, S is a source of light, L1 is a collimating lens and L2 is the telescopic objective. As the two wavelengths λ and λ+dλ are very close if the prism is in minimum deviation position it would hold good for both the wavelengths.

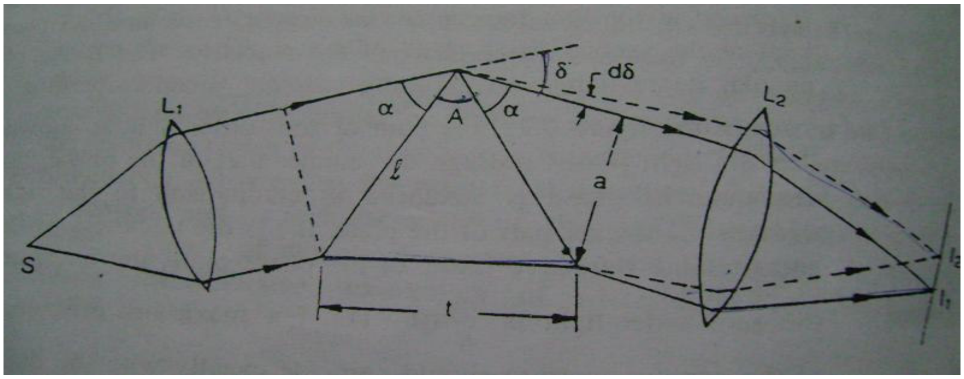


Fig-B

The final image I1 corresponds to the principal maximum for wavelength λ and image I2 corresponds to the principal maxima for wavelength λ+dλ. I1 and I2 are formed at the focal plane of the telescopic objective I2.

In the case of diffraction at a rectangular aperture the position of I2 will correspond to the first maximum of image I1 for wavelength λ1 provided

$a(d\delta) = \lambda$

$d\delta = \lambda/a$ ------------------------------------ (1)

where

δ=angle of minimum deviation for wavelength λ

From Figure

$\alpha + A + \alpha + \delta = \pi$

$\Rightarrow \alpha = [(\pi/2) - (A + \delta)/2]$

Therefore, $\sin\alpha = \sin[(\pi/2)-(A + \delta)/2]$

or $\sin\alpha = \cos[(A + \delta)/2]$

But $\sin\alpha = a/l$

Therefore $\cos[(A + \delta)/2] = a/l$ ------------------------------------ (2)

Also $\sin[A/2]=t/2l$ ------------------------------------ (3)

In case of prism

$\mu = \sin[(A + \delta)/2]/\sin[A/2]$

Therefore $\sin[(A + \delta)/2] = \mu \sin[A/2]$ ----------------------------------- (4)

Here μ and δ are dependent on wavelength of light λ.

Differentiating equation (4) with respect to λ

$(½)(\cos[(A + \delta)/2](d\delta/d\lambda)=[d\mu/d\lambda](\sin[A/2])$

Using equations (2) and (3)

$(a)[d\delta/d\lambda]=(t)[d\mu/d\lambda]$

Therefore

$\lambda/d\lambda=(t)[d\mu/d\lambda]$

The expression λ/dλ measures the resolving power of the prism.

1.3 RESOLVING POWER OF MICROSCOPE

In the case of microscope the object is very near the objective of the microscope and the objects subtend very large angle at the objective. The limit of resolution of a microscope is determined by the least permissible linear distance between the two objects so that the two images are just resolved.

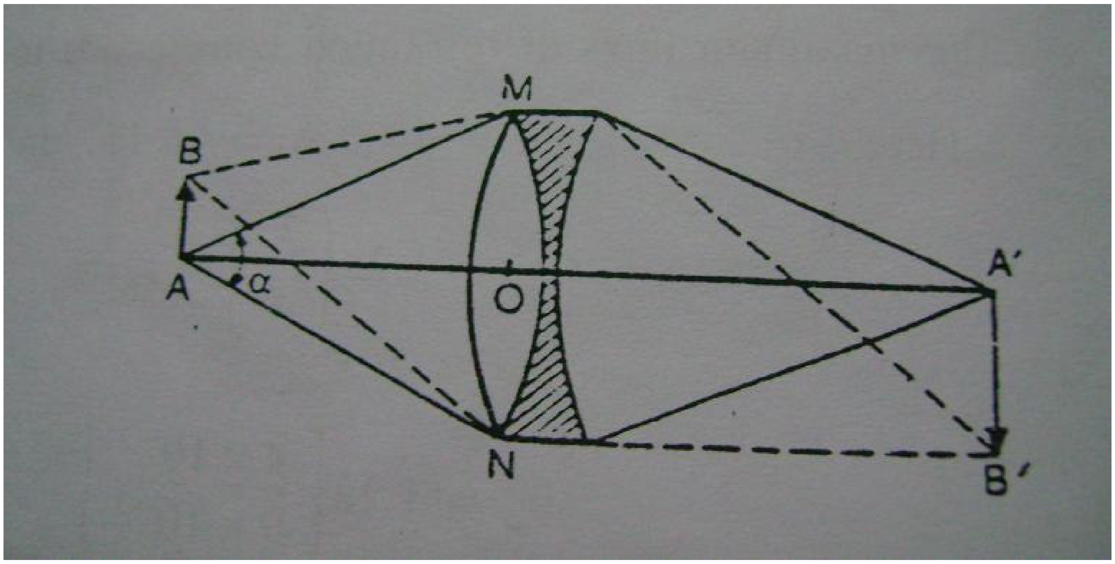


Fig-C

In above figure MN is the aperture of the objective of a microscope and A and B are two object points at a distance'd' apart. A' is the position of the central maximum of A and B' is the central maximum of B. A' and B' are surrounded by alternate dark and bright diffraction rings.

The path difference between the extreme rays from the point B and reaching A' is given by

(BN+ NA') - (BM+MA')

But NA'=MA'

Therefore,

Path Difference=BN-BM

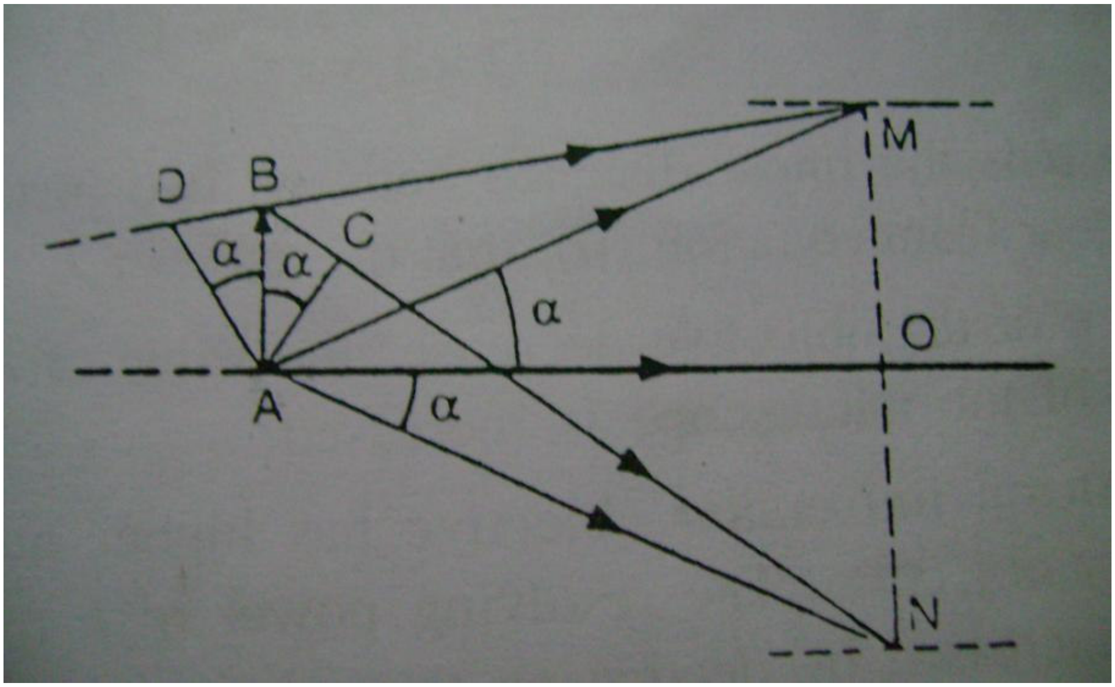


Fig-D

In above figure AD is perpendicular to DM and AC is perpendicular to BN

Therefore,

BN-BM=(BC+CN)-(DM-DB)

But

CN=AN=AM=DM

Therefore,

Path Difference=BC+DB

From triangles ACB and ADB

BC=AB sin α=d sin α

and DB=AB sin α=d sin α

Path Difference=2d sin α

If this path difference 2d sin α=1.22λ then A' corresponds to the first minimum of image B' and two images appear just resolved

Therefore,

$$2d \sin \alpha = 1.22\alpha$$

or $$d = (1.22\lambda)/(2\sin \lambda)$$

Thus above equations give the resolving power of microscope.

1.4 RESOLVING POWER OF TELESCOPE

Let'd' be the diameter of the objective of the telescope considering the incident ray of light from two neighboring points of a distant object. The image of each point object is a Fraunhofer diffraction pattern.

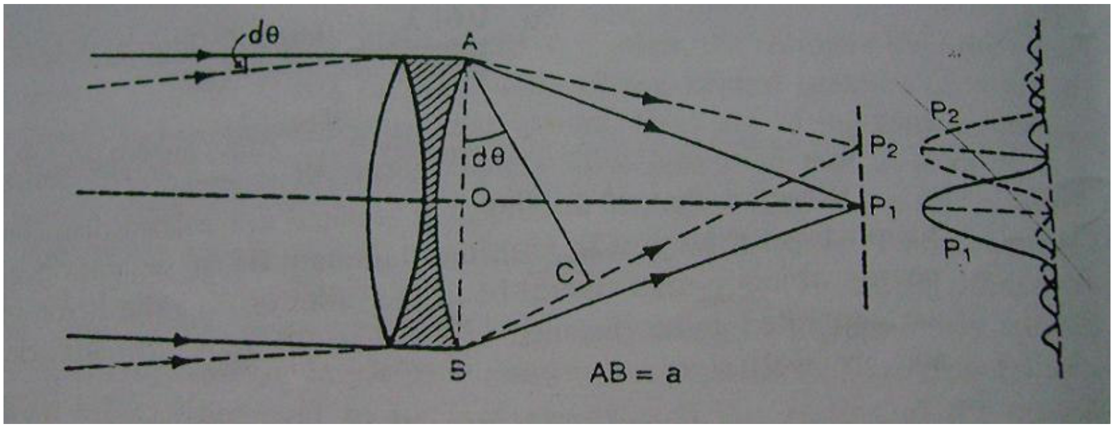


Fig-E

Let P1 and P2 be the position of the central maximum of two images. These two images are resolved if the position of central maximum of second image coincides with the first maximum of the first image and vice-versa. The path difference between the second wave traveling in the directions AP1 and BP1 is low and hence they reinforce with one another at

P1. The secondary waves traveling in the directions AP2 and BP2 will meet at P2 on the screen. Let the angle P2AP1 be dθ. The path difference between the secondary waves traveling in the directions BP2 and AP2 is equal to BC.

Therefore,

$$BC=AB \sin d\theta=AB\, d\theta=a\, d\theta$$

If this path difference [a dθ = λ], the position of P2 corresponds to the first minimum of the first image. But P2 is also the position of the central maximum of the second image. Thus Rayleigh's condition of resolution is satisfied if

$$a\, d\theta=\lambda$$

or $$d\theta=\lambda/a$$

The whole aperture AB can be considered to be made up of two halves AO and OB. The path difference between the secondary waves from the corresponding points in the two halves will be λ/2. The equation dθ=λ/a holds good for rectangular aperture. For circular aperture this equation can be written as

$$d\theta =1.22\lambda/a$$

The reciprocal of d(Theta) measures the resolving power of the telescope.

Therefore,

$$1/d\theta=a/[1.22\lambda]$$

1.5 <u>RESOLVING POWER OF PLANE DIFFRACTION GRATING</u>

The resolving power of a grating is defined as the ratio of the wavelength of any spectral line to the difference in wavelength between this line and a neighboring line such that two lines appear to be just resolved.

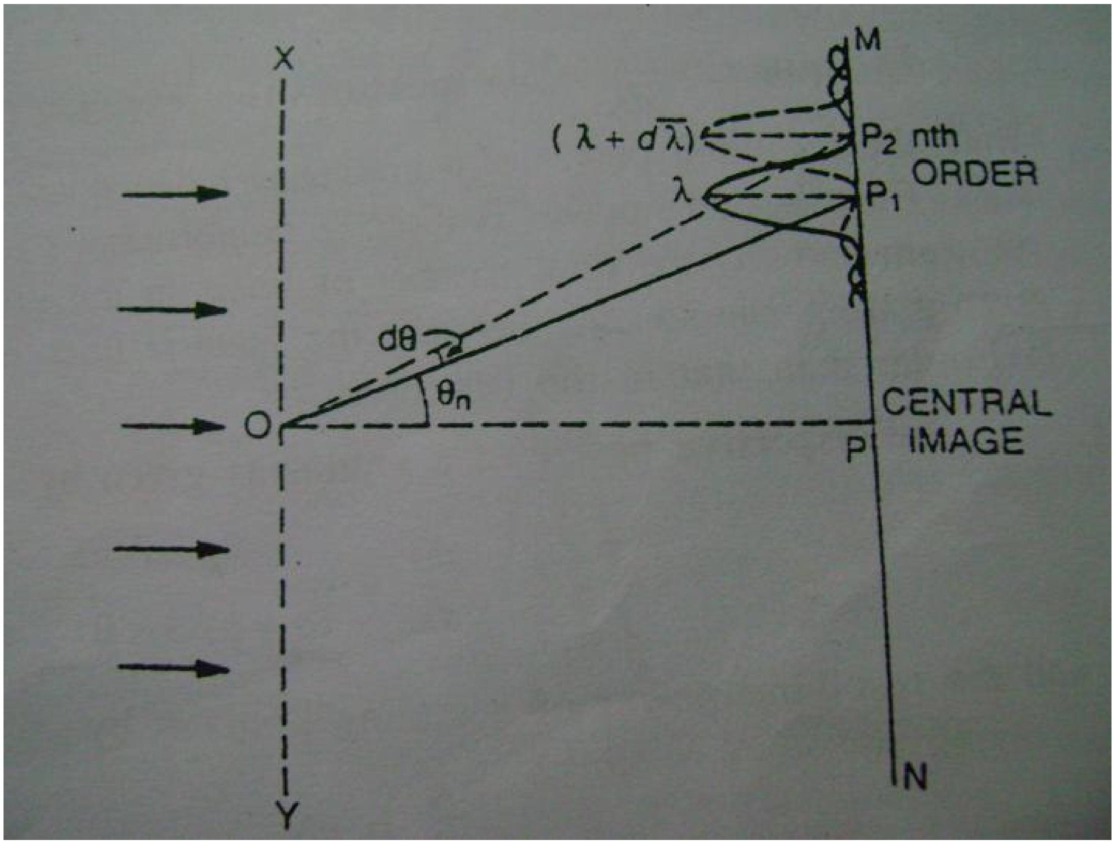


Fig-F

In above figure, XY is a grating surface and MN is the field of view of the telescope.P1 is the nth primary maximum of a spectral line of wavelength λ at an angle of diffraction θ_n . P_2 is the nth primary maximum of a second spectral line of wavelength λ+dλ at a diffracting angle of θ_n+dθ.P_1 and P_2 are the spectral lines in the nth order.

The direction of nth primary maximum for a wavelength λ is given by

$(a + b) \sin \theta_n = n \lambda$ ------------------------------------ (i)

The dirction of nth primary maximum for a wavelength (λ+dλ) is given by

$(a + b) \sin(\theta_n+d\theta) = n (\lambda+d\lambda)$ ------------------------------------ (ii)

The two lines will appear just resolved if the angle of diffraction (θ_n + dθ) also corresponds to the direction of the first secondary minimum after the nth primary maximum at P_1. This is possible if the extra path difference introduced is λ/N where N is the total number of lines of the grating surface.

Therefore

$(a + b) \sin (\theta_n+d\theta) = \quad n\lambda + \lambda/N$ ------------------------------------ (iii)

Equating the right hand sides of (ii) and (iii)

$n (\lambda+d\lambda) = n\lambda + \lambda/N$

$\lambda/d\lambda=nN$

The quantity $\lambda/d\lambda = nN$ measures the resolving power of a grating.

1.6 FINITE ELEMENT METHOD

Basically the finite element method is used to compute the resolving power of a single convex lens or the combination of lenses. It simplifies the procedure for determining the resolving power of optical instrument.

Basically in this unique method the lens is divided into number of smaller segments as shown below. Each segment is treated as a simple prism.

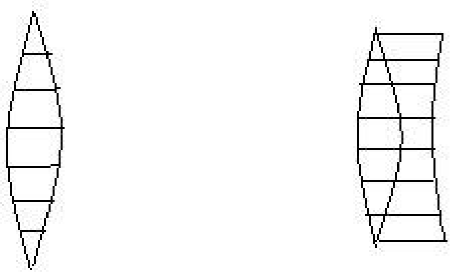


Fig-G

Fig shows the division for a simple convex lens while Fig (b) shows the smaller segments for a combination of lens (convex lens followed by concave lens). To compute the net resolving power of the convex lens shown in (a), the resolving power of each divided segment is computed separately. Finally the minimum resolving power of the segment computed gives the net resolving power of lens as shown in fig (a).

Thus this can be stated as

net $R=\min[R_1\, R_2\, R_3 \ldots\ldots R_n]$

where $R_1,\ R_2, \ldots\ldots\ldots\ R_n$ are the resolving power of the concerned n segments.

For a combination of lens such as shown in fig (b) each segment can be considered as one prism followed by another inverted prism. Thus to calculate the resolving power for such a combination using finite element method the resolving power of first prism R_1 is added to the resolving power of inverted prism R_2 which is then added to the first to obtain the resolving power of a segment. This is repeated for each segment. Finally the minimum resolving power of the segment gives the net resolving power for combination of the lenses.

Chapter 2

ALGORITHM

2 ALGORITHMS FOR DRAWING CONVEX LENS AND CALCULATION OF RESOLVING POWER:

For drawing the convex lens following steps are followed:

a) An interface accepts height and width of lens as input from the user.

b) Using Visual C++ routine, developed, it draws the convex lens as per the inputs.

c) The upper half of the convex lens is then divided into 5 equal divisions.

d) Using another subroutine each division is then drawn separately which is a prism. The diagram is magnified for better and easier observation.

e) Using Cauchy's equation the resolving power of each separate lens is calculated.

Various subroutines to carry out each step are explained separately.

Chapter 3

SUBROUTINES DEVELOPED

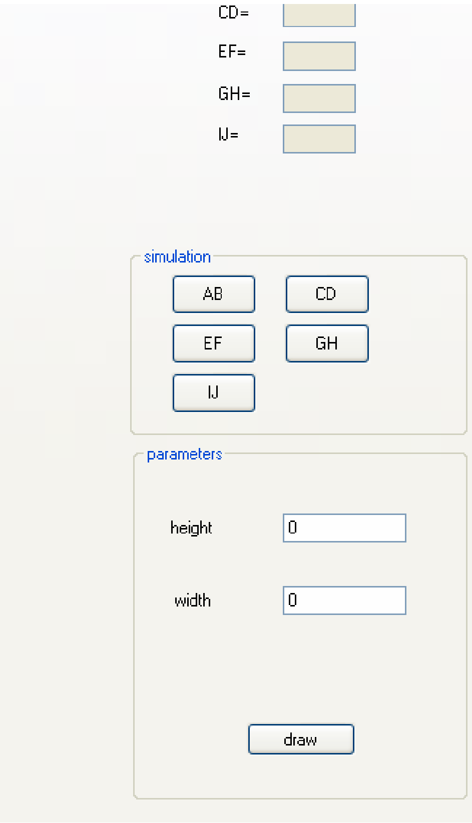
3.1 SUBROUTINE TO DRAW THE ACCEPT USER INPUTS

a) “maskedTextBox2” accepts width of the lens from user.

b) “maskedTextBox1” accepts height of the lens from user.

```
private: System::Void button1_Click(System::Object^  sender, System::EventArgs^  e) {
                                try{
                        width =Convert::ToDouble(maskedTextBox2->Text);
                       height =Convert::ToDouble(maskedTextBox1->Text);
                        if (width>=0 && height>=0){error=0;}
                        {
                          this->tabPage1->Hide();
                          this->tabPage1->Show();
                         this->tabPage2->Show();
                        this->tabPage2->Show();
                        }
                          }
                        catch( Exception^ )
                      {error=1;MessageBox::Show("invalid  value");}
                        }
```

Following is the above subroutine output.



height 0

width 0

3.2 SUBROUTINE TO DRAW THE CONVEX LENS

```
private: System::Void tabPage1_Paint(System::Object^ sender,
System::Windows::Forms::PaintEventArgs^  e) {
            if(error==0){
                Pen^ p = gcnew Pen( Color::Blue,1.0f );
               Graphics^ g = e->Graphics;
               g->DrawEllipse(p,300,30,width*5,500);
                g->DrawLine(p,300,30+250,300+width*5,30+250);
               a=(width*5/2)*(width*5/2);
                b=(500/2)*(500/2);
                 int c=0;
                 for(int i=1;i<=4;i++){
                 c=c+1;
                 int y=30+500*c/(2*5);
                 int x=Math::Sqrt(a*(1-((y-30-250)*(y-
280)/b)))+300+width*5/2;
                 Point p1=Point(x,y);
                 Point p2=Point(2*(300+width*5/2)-x,y);
                 /*int vb=2*(300+width*5/2)-x;
                 MessageBox::Show(" "+vb);*/
                 g->DrawLine(p,p1,p2);
                 if(c==1){this->label12->Location =
System::Drawing::Point(x+8, y);
                 this->label12->Show();
                 this->label11->Location =
System::Drawing::Point(2*(300+width*5/2)-x-15,y);
                 this->label11->Show();}//1stif
     if(c==2){this->label10->Location = System::Drawing::Point(x+10, y);
                 this->label10->Show();
```

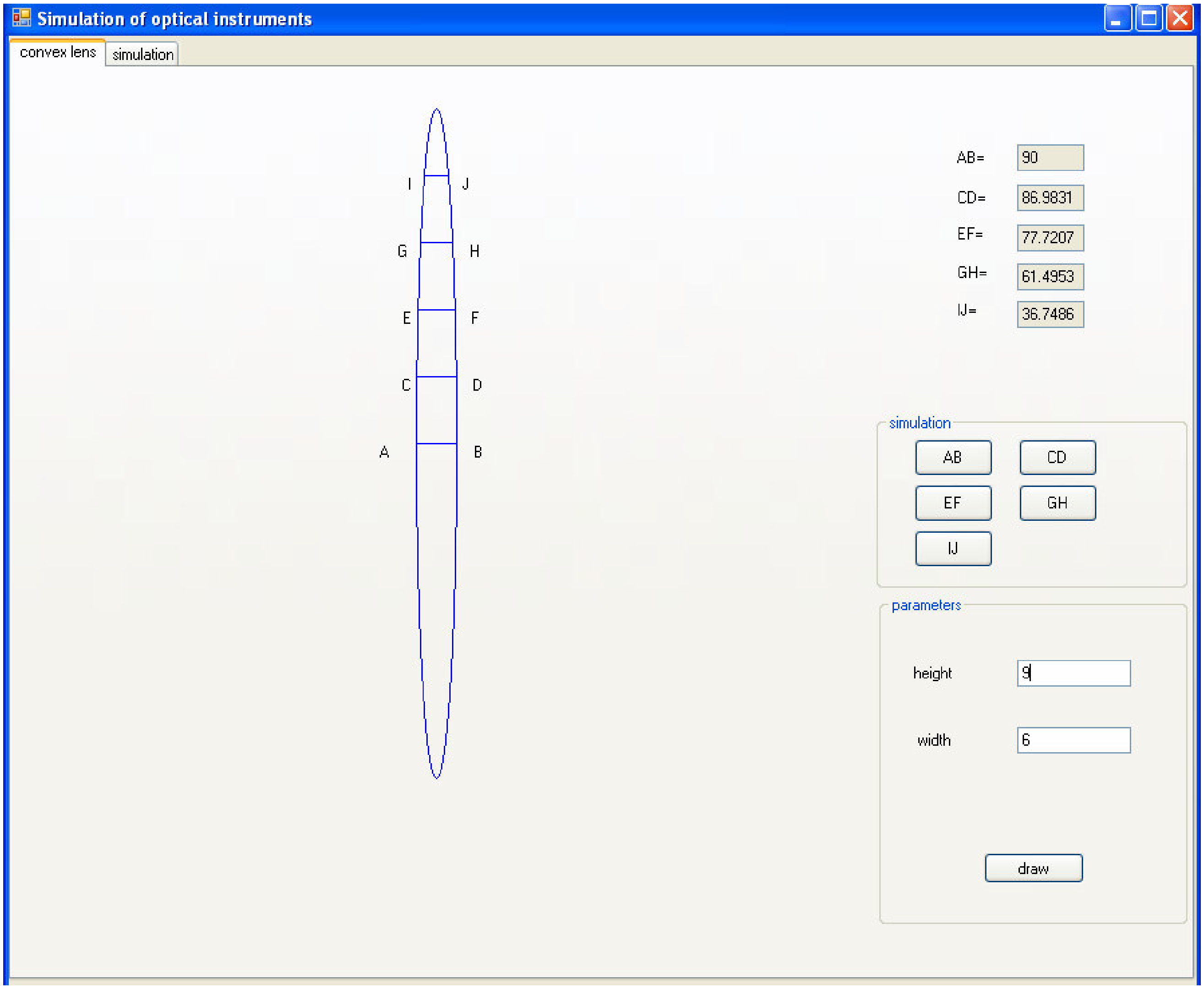
```
 this->label9->Location = System::Drawing::Point(2*(300+width*5/2)-x-20,y);
                              this->label9->Show();}//1stif
        if(c==3){this->label8->Location = System::Drawing::Point(x+10, y);
                              this->label8->Show();
                              this->label7->Location                                    =
System::Drawing::Point(2*(300+width*5/2)-x-15,y);
                              this->label7->Show();}//1stif
        if(c==4){this->label6->Location = System::Drawing::Point(x+10, y);
                              this->label6->Show();
                              this->label5->Location                                    =
System::Drawing::Point(2*(300+width*5/2)-x-15,y);
                              this->label5->Show();}//1stif
                              }//endfor
                              this->label3->Location    =    System::Drawing::Point(270,
30+250);
                              this->label3->Show();
          this->label4->Location = System::Drawing::Point(310+width*5,30+250);
                              this->label4->Show();
                              ////////////////////////////////////////////////////////////////////////////
                              float he=height*30;
                              float wi=(height/3)*30;
                              float ex=he-wi/2;
                              float calc=he/2*he/2-(ex-he/2)*(ex-he/2);
                              float y=(Math::Sqrt(calc))+he/2;
                              acth=(he/2-(he-y));
                              this->textBox1->Text=Convert::ToString(wi);
                              ab=wi;
                              y=he/2-(acth/5);
 ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                              this->textBox2->Text=Convert::ToString(2*(ex-he+wi/2));
                              cd=2*(ex-he+wi/2);
          /*g->DrawLine(p,ex,y,he-2*(ex-he+wi/2),y);*/
                              y=he/2-2*(acth/5);
                              ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                              this->textBox3->Text=Convert::ToString(2*(ex-he+wi/2));
                              ef=2*(ex-he+wi/2);
      /*   g->DrawLine(p,ex,y,he-wi/2,y);*/
                              y=he/2-3*(acth/5);
                              ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                              this->textBox4->Text=Convert::ToString(2*(ex-he+wi/2));
                              gh=2*(ex-he+wi/2);
          /* g->DrawLine(p,ex,y,he-wi/2,y);*/
                              y=he/2-4*(acth/5);
                              ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                              this->textBox5->Text=Convert::ToString(2*(ex-he+wi/2));
                              ij=2*(ex-he+wi/2);
        /*  g->DrawLine(p,ex,y,he-wi/2,y);*/

                       }//endif
               }
```

Following is output of the above subroutine:

a) It shows various divisions of the lens.

b) It shows the width of each division.



AB= 90

CD= 86.9831

EF= 77.7207

GH= 61.4953

IJ= 36.7486

height 9

width 6

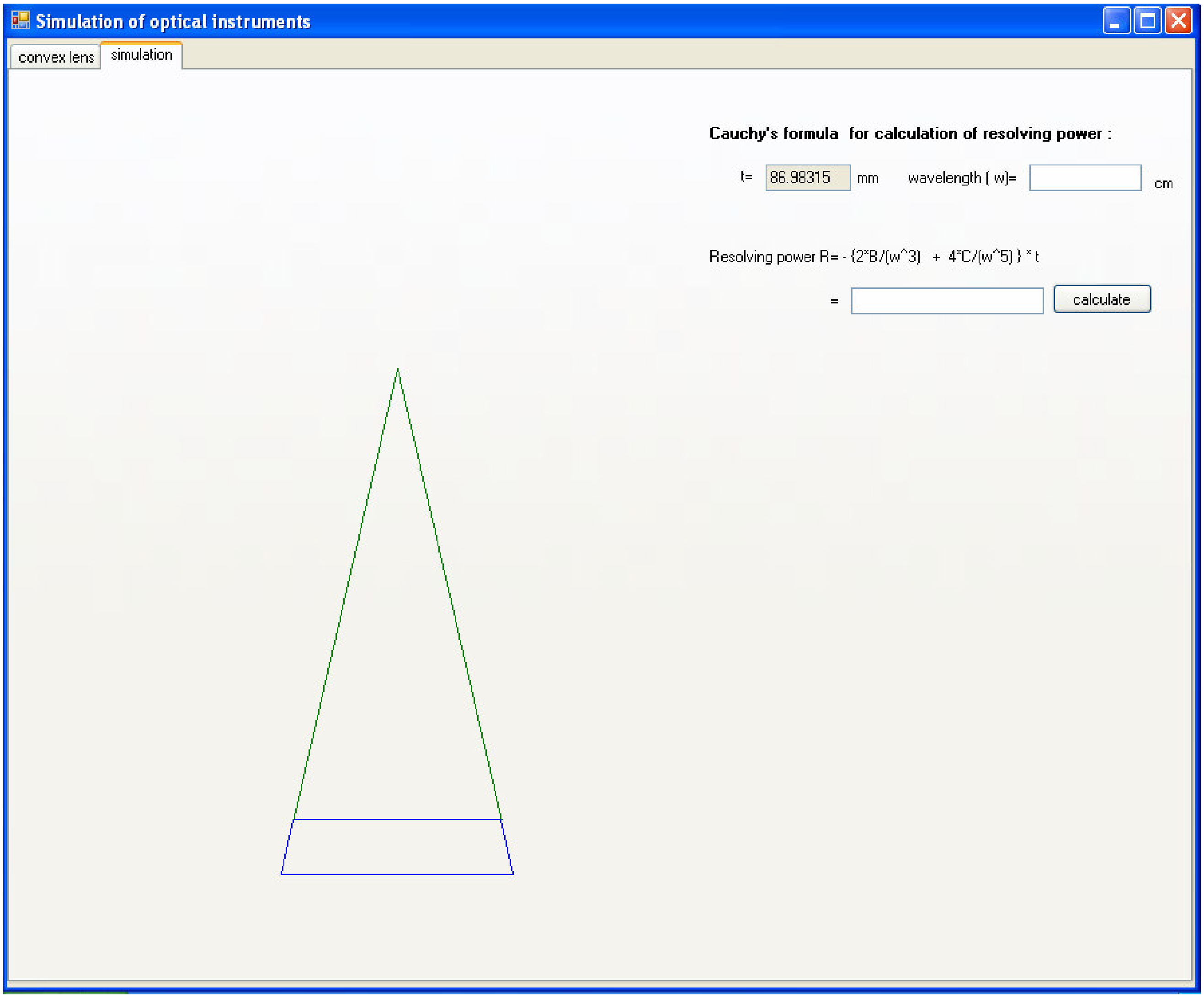
3.3 SUBROUTINE TO DRAW THE PRISMS

```
private: System::Void tabPage2_Paint(System::Object^ sender,
System::Windows::Forms::PaintEventArgs^ e) {
    Pen^ p = gcnew Pen( Color::Blue,1.0f );
            Pen^ p2 = gcnew Pen( Color::Green,1.0f );
    Graphics^ g = e->Graphics;
            if(button==1){
            this->textBox6->Text=Convert::ToString(ij);
            Point point1 = Point(200,400);
            Point point2 = Point(200+6*ij,400);
            Point point3 = Point(200+3*ij,400-6*acth/5);
            array<Point>^ curvePoints = {point1,point2,point3};
            g->DrawPolygon(p,curvePoints);}
            ;
            if(button==2){
            this->textBox6->Text=Convert::ToString(gh);
            Point point1 = Point(200,600);
            Point point2 = Point(200+6*gh,600);
            Point point3 = Point(200+3*gh+3*ij,600-6*acth/5);
            Point point4 = Point(200+3*gh-3*ij,600-6*acth/5);
            array<Point>^ curvePoints =
{point1,point2,point3,point4};
            g->DrawPolygon(p,curvePoints);
            float m=(600-(600-6*acth/5))/(200-(200+3*gh-3*ij));
            float x=200+3*gh;
            float y=m*(x-200)+600;
            g->DrawLine(p2,x,y,200+3*gh+3*ij,600-6*acth/5);
            g->DrawLine(p2,x,y,200+3*gh-3*ij,600-6*acth/5);
            }
```

```
        if (button==3){
                this->textBox6->Text=Convert::ToString(ef);
                Point point1 = Point(200,600);
                Point point2 = Point(200+6*ef,600);
                Point point3 = Point(200+3*ef+3*gh,600-6*acth/5);
                Point point4 = Point(200+3*ef-3*gh,600-6*acth/5);
                array<Point>^          curvePoints          =
{point1,point2,point3,point4};
                g->DrawPolygon(p,curvePoints);
                float m=(600-(600-6*acth/5))/(200-(200+3*ef-3*gh));
                float x=200+3*ef;
                float y=m*(x-200)+600;
                g->DrawLine(p2,x,y,200+3*ef+3*gh,600-6*acth/5);
                g->DrawLine(p2,x,y,200+3*ef-3*gh,600-6*acth/5);
                }
                if (button==4){
                this->textBox6->Text=Convert::ToString(cd);
                Point point1 = Point(200,600);
                Point point2 = Point(200+2*cd,600);
                Point point3 = Point(200+1*cd+1*ef,600-2*acth/5);
                Point point4 = Point(200+1*cd-1*ef,600-2*acth/5);
                array<Point>^          curvePoints          =
{point1,point2,point3,point4};
                g->DrawPolygon(p,curvePoints);
                float m=(600-(600-2*acth/5))/(200-(200+1*cd-1*ef));
                float x=200+1*cd;
                float y=m*(x-200)+600;
                g->DrawLine(p2,x,y,200+1*cd+1*ef,600-2*acth/5);
        g->DrawLine(p2,x,y,200+1*cd-1*ef,600-2*acth/5);
                }
                if (button==5){
                this->textBox6->Text=Convert::ToString(ab);
                Point point1 = Point(200,600);
                Point point2 = Point(200+2*ab,600);
                Point point3 = Point(200+1*ab+1*cd,600-2*acth/5);
                Point point4 = Point(200+1*ab-1*cd,600-2*acth/5);
                array<Point>^         curvePoints         =
{point1,point2,point3,point4};
                g->DrawPolygon(p,curvePoints);
                float m=(600-(600-2*acth/5))/(200-(200+1*ab-1*cd));
                float x=200+1*ab;
                float y=m*(x-200)+600;
                g->DrawLine(p2,x,y,200+1*ab+1*cd,600-2*acth/5);
                g->DrawLine(p2,x,y,200+1*ab-1*cd,600-2*acth/5);
                }
```

Following is the output of above subroutine:

a) “t=86.98315 mm” shows the base width of the prism.
b) “wavelength(w)= “ has to be fed by the user.
c) “Resolving power (R) =” gives the resolving power of the prism.
d) The green lines is drawn by the subroutine , its height being calculated on the basis of widths of upper width an lower width of the blue trapezium.



Cauchy's formula for calculation of resolving power :

t= 86.98315 mm   wavelength (w)= cm

Resolving power R= - {2*B/(w^3) + 4*C/(w^5) } * t

=

Chapter 4

ENTIRE CODE OF THE SIMULATION IN VISUAL C++

Below is shown the entire code that was used to develop the simulation:

```
#pragma once



namespace sim {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace System::IO;

	/// <summary>
	/// Summary for Form1
	///
	/// WARNING: If you change the name of this class, you will need to change the
	///          'Resource File Name' property for the managed resource compiler tool
	///          associated with all .resx files this class depends on.  Otherwise,
	///          the designers will not be able to interact properly with localized
	///          resources associated with this form.
	/// </summary>
	public ref class Form1 : public System::Windows::Forms::Form
	{
	public:
		Form1(void)
		{
```

```
InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
    }

protected:
    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    ~Form1()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::TabControl^  tabControl1;
protected:
private: System::Windows::Forms::TabPage^  tabPage1;

private: System::Windows::Forms::GroupBox^  groupBox1;
private: System::Windows::Forms::Button^  button1;
private: System::Windows::Forms::Label^  label2;
private: System::Windows::Forms::Label^  label1;
private: System::Windows::Forms::MaskedTextBox^  maskedTextBox2;
private: System::Windows::Forms::MaskedTextBox^  maskedTextBox1;
private: System::Windows::Forms::Label^  label12;
private: System::Windows::Forms::Label^  label11;
```

```
private: System::Windows::Forms::Label^  label10;
private: System::Windows::Forms::Label^  label9;
private: System::Windows::Forms::Label^  label8;
private: System::Windows::Forms::Label^  label7;
private: System::Windows::Forms::Label^  label6;
private: System::Windows::Forms::Label^  label5;
private: System::Windows::Forms::Label^  label4;
private: System::Windows::Forms::Label^  label3;
private: System::Windows::Forms::TabPage^  tabPage2;


private: System::Windows::Forms::Label^  label17;
private: System::Windows::Forms::Label^  label16;
private: System::Windows::Forms::Label^  label15;
private: System::Windows::Forms::Label^  label14;
private: System::Windows::Forms::Label^  label13;
private: System::Windows::Forms::TextBox^  textBox5;
private: System::Windows::Forms::TextBox^  textBox4;
private: System::Windows::Forms::TextBox^  textBox3;
private: System::Windows::Forms::TextBox^  textBox2;
private: System::Windows::Forms::TextBox^  textBox1;
private: System::Windows::Forms::GroupBox^  groupBox2;
private: System::Windows::Forms::Button^  button3;
private: System::Windows::Forms::Button^  button2;
private: System::Windows::Forms::Button^  button6;
private: System::Windows::Forms::Button^  button5;
private: System::Windows::Forms::Button^  button4;
private: System::Windows::Forms::Label^  label18;
private: System::Windows::Forms::TextBox^  textBox6;
```

```
	private: System::Windows::Forms::Label^  label19;
	private: System::Windows::Forms::Label^  label20;
	private: System::Windows::Forms::TextBox^  textBox7;

	private: System::Windows::Forms::Label^  label21;
	private: System::Windows::Forms::Label^  label22;
	private: System::Windows::Forms::TextBox^  textBox8;
	private: System::Windows::Forms::Label^  label23;
	private: System::Windows::Forms::Label^  label24;
	private: System::Windows::Forms::Button^  button7;




	private:
		/// <summary>
		/// Required designer variable.
		/// </summary>
		System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
		/// <summary>
		/// Required method for Designer support - do not modify
		/// the contents of this method with the code editor.
		/// </summary>
		void InitializeComponent(void)
		{
        this->tabControl1 = (gcnew System::Windows::Forms::TabControl());
                this->tabPage1 = (gcnew System::Windows::Forms::TabPage());
                this->groupBox2                =                (gcnew
System::Windows::Forms::GroupBox());
                this->button6 = (gcnew System::Windows::Forms::Button());
                this->button5 = (gcnew System::Windows::Forms::Button());
                this->button4 = (gcnew System::Windows::Forms::Button());
                this->button3 = (gcnew System::Windows::Forms::Button());
                this->button2 = (gcnew System::Windows::Forms::Button());
                this->textBox5 = (gcnew System::Windows::Forms::TextBox());
                this->textBox4 = (gcnew System::Windows::Forms::TextBox());
                this->textBox3 = (gcnew System::Windows::Forms::TextBox());
                this->textBox2 = (gcnew System::Windows::Forms::TextBox());
                this->textBox1 = (gcnew System::Windows::Forms::TextBox());
                this->label17 = (gcnew System::Windows::Forms::Label());
                this->label16 = (gcnew System::Windows::Forms::Label());
                this->label15 = (gcnew System::Windows::Forms::Label());
                this->label14 = (gcnew System::Windows::Forms::Label());
                this->label13 = (gcnew System::Windows::Forms::Label());
                this->label12 = (gcnew System::Windows::Forms::Label());
                this->label11 = (gcnew System::Windows::Forms::Label());
                this->label10 = (gcnew System::Windows::Forms::Label());
                this->label9 = (gcnew System::Windows::Forms::Label());
                this->label8 = (gcnew System::Windows::Forms::Label());
                this->label7 = (gcnew System::Windows::Forms::Label());
                this->label6 = (gcnew System::Windows::Forms::Label());
                this->label5 = (gcnew System::Windows::Forms::Label());
                this->label4 = (gcnew System::Windows::Forms::Label());
                this->label3 = (gcnew System::Windows::Forms::Label());
        this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());
                this->button1 = (gcnew System::Windows::Forms::Button());
                this->label2 = (gcnew System::Windows::Forms::Label());
                this->label1 = (gcnew System::Windows::Forms::Label());
                this->maskedTextBox2 = (gcnew
System::Windows::Forms::MaskedTextBox());
                this->maskedTextBox1 = (gcnew
System::Windows::Forms::MaskedTextBox());
                this->tabPage2 = (gcnew System::Windows::Forms::TabPage());
                this->button7 = (gcnew System::Windows::Forms::Button());
                this->label24 = (gcnew System::Windows::Forms::Label());
                this->label23 = (gcnew System::Windows::Forms::Label());
                this->textBox8 = (gcnew System::Windows::Forms::TextBox());
                this->label22 = (gcnew System::Windows::Forms::Label());
                this->label21 = (gcnew System::Windows::Forms::Label());
                this->textBox7 = (gcnew System::Windows::Forms::TextBox());
                this->label20 = (gcnew System::Windows::Forms::Label());
                this->textBox6 = (gcnew System::Windows::Forms::TextBox());
                this->label19 = (gcnew System::Windows::Forms::Label());
                this->label18 = (gcnew System::Windows::Forms::Label());
                this->tabControl1->SuspendLayout();
                this->tabPage1->SuspendLayout();
                this->groupBox2->SuspendLayout();
                this->groupBox1->SuspendLayout();
                this->tabPage2->SuspendLayout();
                this->SuspendLayout();
                //
                // tabControl1
                //
```

```
this->tabControl1->Controls->Add(this->tabPage1);
this->tabControl1->Controls->Add(this->tabPage2);
this->tabControl1->Location = System::Drawing::Point(0, 2);
this->tabControl1->Name = L"tabControl1";
this->tabControl1->SelectedIndex = 0;
this->tabControl1->Size = System::Drawing::Size(887, 704);
this->tabControl1->TabIndex = 0;
//
// tabPage1
//
this->tabPage1->Controls->Add(this->groupBox2);
this->tabPage1->Controls->Add(this->textBox5);
this->tabPage1->Controls->Add(this->textBox4);
this->tabPage1->Controls->Add(this->textBox3);
this->tabPage1->Controls->Add(this->textBox2);
this->tabPage1->Controls->Add(this->textBox1);
this->tabPage1->Controls->Add(this->label17);
this->tabPage1->Controls->Add(this->label16);
this->tabPage1->Controls->Add(this->label15);
this->tabPage1->Controls->Add(this->label14);
this->tabPage1->Controls->Add(this->label13);
this->tabPage1->Controls->Add(this->label12);
this->tabPage1->Controls->Add(this->label11);
this->tabPage1->Controls->Add(this->label10);
this->tabPage1->Controls->Add(this->label9);
this->tabPage1->Controls->Add(this->label8);
this->tabPage1->Controls->Add(this->label7);
this->tabPage1->Controls->Add(this->label6);
this->tabPage1->Controls->Add(this->label5);
```

```
            this->tabPage1->Controls->Add(this->label4);
            this->tabPage1->Controls->Add(this->label3);
            this->tabPage1->Controls->Add(this->groupBox1);
            this->tabPage1->Location = System::Drawing::Point(4, 22);
            this->tabPage1->Name = L"tabPage1";
            this->tabPage1->Padding = System::Windows::Forms::Padding(3);
            this->tabPage1->Size = System::Drawing::Size(879, 678);
            this->tabPage1->TabIndex = 0;
            this->tabPage1->Text = L"convex lens";
            this->tabPage1->UseVisualStyleBackColor = true;
            this->tabPage1->Paint                    +=                    gcnew
System::Windows::Forms::PaintEventHandler(this, &Form1::tabPage1_Paint);
            //
            // groupBox2
            //
            this->groupBox2->Controls->Add(this->button6);
            this->groupBox2->Controls->Add(this->button5);
            this->groupBox2->Controls->Add(this->button4);
            this->groupBox2->Controls->Add(this->button3);
            this->groupBox2->Controls->Add(this->button2);
            this->groupBox2->Location = System::Drawing::Point(644, 258);
            this->groupBox2->Name = L"groupBox2";
            this->groupBox2->Size = System::Drawing::Size(232, 130);
            this->groupBox2->TabIndex = 31;
            this->groupBox2->TabStop = false;
            this->groupBox2->Text = L"simulation";
            //
            // button6
            //
            this->button6->Location = System::Drawing::Point(28, 87);
            this->button6->Name = L"button6";
            this->button6->Size = System::Drawing::Size(59, 28);
            this->button6->TabIndex = 32;
            this->button6->Text = L"IJ";
            this->button6->UseVisualStyleBackColor = true;
            this->button6->Click += gcnew System::EventHandler(this,
&Form1::button6_Click);
            //
            // button5
            //
            this->button5->Location = System::Drawing::Point(106, 53);
            this->button5->Name = L"button5";
            this->button5->Size = System::Drawing::Size(59, 28);
            this->button5->TabIndex = 32;
            this->button5->Text = L"GH";
            this->button5->UseVisualStyleBackColor = true;
            this->button5->Click += gcnew System::EventHandler(this,
&Form1::button5_Click);
            //
            // button4
            //
            this->button4->Location = System::Drawing::Point(28, 53);
            this->button4->Name = L"button4";
            this->button4->Size = System::Drawing::Size(59, 28);
            this->button4->TabIndex = 2;
            this->button4->Text = L"EF";
            this->button4->UseVisualStyleBackColor = true;
            this->button4->Click += gcnew System::EventHandler(this,
        &Form1::button4_Click);
            //
            // button3
            //
            this->button3->Location = System::Drawing::Point(106, 19);
            this->button3->Name = L"button3";
            this->button3->Size = System::Drawing::Size(59, 28);
            this->button3->TabIndex = 1;
            this->button3->Text = L"CD";
            this->button3->UseVisualStyleBackColor = true;
            this->button3->Click += gcnew System::EventHandler(this,
&Form1::button3_Click);
            //
            // button2
            //
            this->button2->Location = System::Drawing::Point(28, 19);
            this->button2->Name = L"button2";
            this->button2->Size = System::Drawing::Size(59, 28);
            this->button2->TabIndex = 0;
            this->button2->Text = L"AB";
            this->button2->UseVisualStyleBackColor = true;
            this->button2->Click += gcnew System::EventHandler(this,
&Form1::button2_Click);
            //
            // textBox5
            //
            this->textBox5->Location = System::Drawing::Point(749, 174);
            this->textBox5->Name = L"textBox5";
```

```
this->textBox5->ReadOnly = true;
this->textBox5->Size = System::Drawing::Size(50, 20);
this->textBox5->TabIndex = 30;
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(749, 146);
this->textBox4->Name = L"textBox4";
this->textBox4->ReadOnly = true;
this->textBox4->Size = System::Drawing::Size(50, 20);
this->textBox4->TabIndex = 29;
//
// textBox3
//
this->textBox3->Location = System::Drawing::Point(749, 117);
this->textBox3->Name = L"textBox3";
this->textBox3->ReadOnly = true;
this->textBox3->Size = System::Drawing::Size(50, 20);
this->textBox3->TabIndex = 28;
//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(749, 87);
this->textBox2->Name = L"textBox2";
this->textBox2->ReadOnly = true;
this->textBox2->Size = System::Drawing::Size(50, 20);
this->textBox2->TabIndex = 27;
//
// textBox1
```

```
//
this->textBox1->Cursor = System::Windows::Forms::Cursors::No;
this->textBox1->Location = System::Drawing::Point(749, 57);
this->textBox1->Name = L"textBox1";
this->textBox1->ReadOnly = true;
this->textBox1->Size = System::Drawing::Size(50, 20);
this->textBox1->TabIndex = 26;
//
// label17
//
this->label17->AutoSize = true;
this->label17->Location = System::Drawing::Point(701, 174);
this->label17->Name = L"label17";
this->label17->Size = System::Drawing::Size(21, 13);
this->label17->TabIndex = 25;
this->label17->Text = L"IJ=";
//
// label16
//
this->label16->AutoSize = true;
this->label16->Location = System::Drawing::Point(701, 146);
this->label16->Name = L"label16";
this->label16->Size = System::Drawing::Size(29, 13);
this->label16->TabIndex = 24;
this->label16->Text = L"GH=";
//
// label15
//
this->label15->AutoSize = true;
```

```
this->label15->Location = System::Drawing::Point(701, 117);
this->label15->Name = L"label15";
this->label15->Size = System::Drawing::Size(26, 13);
this->label15->TabIndex = 23;
this->label15->Text = L"EF=";
//
// label14
//
this->label14->AutoSize = true;
this->label14->Location = System::Drawing::Point(701, 90);
this->label14->Name = L"label14";
this->label14->Size = System::Drawing::Size(28, 13);
this->label14->TabIndex = 22;
this->label14->Text = L"CD=";
//
// label13
//
this->label13->AutoSize = true;
this->label13->Location = System::Drawing::Point(701, 60);
this->label13->Name = L"label13";
this->label13->Size = System::Drawing::Size(27, 13);
this->label13->TabIndex = 21;
this->label13->Text = L"AB=";
//
// label12
//
this->label12->AutoSize = true;
this->label12->Location = System::Drawing::Point(180, 626);
this->label12->Name = L"label12";
```

```
this->label12->Size = System::Drawing::Size(12, 13);
this->label12->TabIndex = 20;
this->label12->Text = L"J";
//
// label11
//
this->label11->AutoSize = true;
this->label11->Location = System::Drawing::Point(164, 626);
this->label11->Name = L"label11";
this->label11->Size = System::Drawing::Size(10, 13);
this->label11->TabIndex = 19;
this->label11->Text = L"I";
//
// label10
//
this->label10->AutoSize = true;
this->label10->Location = System::Drawing::Point(143, 626);
this->label10->Name = L"label10";
this->label10->Size = System::Drawing::Size(15, 13);
this->label10->TabIndex = 18;
this->label10->Text = L"H";
//
// label9
//
this->label9->AutoSize = true;
this->label9->Location = System::Drawing::Point(122, 626);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(15, 13);
this->label9->TabIndex = 17;
```

```
this->label9->Text = L"G";
//
// label8
//
this->label8->AutoSize = true;
this->label8->Location = System::Drawing::Point(103, 626);
this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(13, 13);
this->label8->TabIndex = 16;
this->label8->Text = L"F";
//
// label7
//
this->label7->AutoSize = true;
this->label7->Location = System::Drawing::Point(83, 626);
this->label7->Name = L"label7";
this->label7->Size = System::Drawing::Size(14, 13);
this->label7->TabIndex = 15;
this->label7->Text = L"E";
//
// label6
//
this->label6->AutoSize = true;
this->label6->Location = System::Drawing::Point(62, 626);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(15, 13);
this->label6->TabIndex = 14;
this->label6->Text = L"D";
//
```

```
// label5
//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(42, 626);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(14, 13);
this->label5->TabIndex = 13;
this->label5->Text = L"C";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Location = System::Drawing::Point(22, 626);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(14, 13);
this->label4->TabIndex = 12;
this->label4->Text = L"B";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Location = System::Drawing::Point(3, 626);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(14, 13);
this->label3->TabIndex = 11;
this->label3->Text = L"A";
//
// groupBox1
//
```

```
                        this->groupBox1->Controls->Add(this->button1);
                        this->groupBox1->Controls->Add(this->label2);
                        this->groupBox1->Controls->Add(this->label1);
                        this->groupBox1->Controls->Add(this->maskedTextBox2);
                        this->groupBox1->Controls->Add(this->maskedTextBox1);
                        this->groupBox1->Location = System::Drawing::Point(646, 394);
                        this->groupBox1->Name = L"groupBox1";
                        this->groupBox1->Size = System::Drawing::Size(230, 245);
                        this->groupBox1->TabIndex = 1;
                        this->groupBox1->TabStop = false;
                        this->groupBox1->Text = L"parameters";
                        //
                        // button1
                        //
                        this->button1->Location = System::Drawing::Point(78, 192);
                        this->button1->Name = L"button1";
                        this->button1->Size = System::Drawing::Size(75, 23);
                        this->button1->TabIndex = 4;
                        this->button1->Text = L"draw";
                        this->button1->UseVisualStyleBackColor = true;
                        this->button1->Click += gcnew System::EventHandler(this,
&Form1::button1_Click);
                        //
                        // label2
                        //
                        this->label2->AutoSize = true;
                        this->label2->Location = System::Drawing::Point(25, 101);
                        this->label2->Name = L"label2";
                        this->label2->Size = System::Drawing::Size(32, 13);
```

```
            this->label2->TabIndex = 3;
            this->label2->Text = L"width";
            //
            // label1
            //
            this->label1->AutoSize = true;
            this->label1->Location = System::Drawing::Point(22, 51);
            this->label1->Name = L"label1";
            this->label1->Size = System::Drawing::Size(36, 13);
            this->label1->TabIndex = 2;
            this->label1->Text = L"height";
            //
            // maskedTextBox2
            //
            this->maskedTextBox2->Location = System::Drawing::Point(103,
98);
            this->maskedTextBox2->Mask = L"0";
            this->maskedTextBox2->Name = L"maskedTextBox2";
            this->maskedTextBox2->PromptChar = '0';
            this->maskedTextBox2->Size = System::Drawing::Size(85, 20);
            this->maskedTextBox2->TabIndex = 1;
            //
            // maskedTextBox1
            //
            this->maskedTextBox1->Location = System::Drawing::Point(103,
48);
            this->maskedTextBox1->Mask = L"0";
            this->maskedTextBox1->Name = L"maskedTextBox1";
            this->maskedTextBox1->PromptChar = '0';
```

```
this->maskedTextBox1->Size = System::Drawing::Size(85, 20);
this->maskedTextBox1->TabIndex = 0;
//
// tabPage2
//
this->tabPage2->Controls->Add(this->button7);
this->tabPage2->Controls->Add(this->label24);
this->tabPage2->Controls->Add(this->label23);
this->tabPage2->Controls->Add(this->textBox8);
this->tabPage2->Controls->Add(this->label22);
this->tabPage2->Controls->Add(this->label21);
this->tabPage2->Controls->Add(this->textBox7);
this->tabPage2->Controls->Add(this->label20);
this->tabPage2->Controls->Add(this->textBox6);
this->tabPage2->Controls->Add(this->label19);
this->tabPage2->Controls->Add(this->label18);
this->tabPage2->Location = System::Drawing::Point(4, 22);
this->tabPage2->Name = L"tabPage2";
this->tabPage2->Padding = System::Windows::Forms::Padding(3);
this->tabPage2->Size = System::Drawing::Size(879, 678);
this->tabPage2->TabIndex = 1;
this->tabPage2->Text = L"simulation";
this->tabPage2->UseVisualStyleBackColor = true;
this->tabPage2->Paint += gcnew System::Windows::Forms::PaintEventHandler(this, &Form1::tabPage2_Paint);
//
// button7
//
this->button7->Location = System::Drawing::Point(776, 159);
```

```
            this->button7->Name = L"button7";
            this->button7->Size = System::Drawing::Size(75, 23);
            this->button7->TabIndex = 10;
            this->button7->Text = L"calculate";
            this->button7->UseVisualStyleBackColor = true;
            this->button7->Click += gcnew System::EventHandler(this,
&Form1::button7_Click);
            //
            // label24
            //
            this->label24->AutoSize = true;
            this->label24->Location = System::Drawing::Point(849, 77);
            this->label24->Name = L"label24";
            this->label24->Size = System::Drawing::Size(21, 13);
            this->label24->TabIndex = 9;
            this->label24->Text = L"cm";
            //
            // label23
            //
            this->label23->AutoSize = true;
            this->label23->Location = System::Drawing::Point(627, 73);
            this->label23->Name = L"label23";
            this->label23->Size = System::Drawing::Size(23, 13);
            this->label23->TabIndex = 8;
            this->label23->Text = L"mm";
            //
            // textBox8
            //
            this->textBox8->Location = System::Drawing::Point(626, 162);
```

```
this->textBox8->Name = L"textBox8";
this->textBox8->Size = System::Drawing::Size(144, 20);
this->textBox8->TabIndex = 7;
//
// label22
//
this->label22->AutoSize = true;
this->label22->Location = System::Drawing::Point(607, 165);
this->label22->Name = L"label22";
this->label22->Size = System::Drawing::Size(13, 13);
this->label22->TabIndex = 6;
this->label22->Text = L"=";
//
// label21
//
this->label21->AutoSize = true;
this->label21->Location = System::Drawing::Point(517, 132);
this->label21->Name = L"label21";
this->label21->Size = System::Drawing::Size(253, 13);
this->label21->TabIndex = 5;
this->label21->Text = L"Resolving power R= - {2*B/(w^3) + 4*C/(w^5) } * t";
//
// textBox7
//
this->textBox7->Location = System::Drawing::Point(759, 70);
this->textBox7->Name = L"textBox7";
this->textBox7->Size = System::Drawing::Size(84, 20);
this->textBox7->TabIndex = 4;
```

```
//
// label20
//
this->label20->AutoSize = true;
this->label20->Location = System::Drawing::Point(665, 73);
this->label20->Name = L"label20";
this->label20->Size = System::Drawing::Size(88, 13);
this->label20->TabIndex = 3;
this->label20->Text = L"wavelength ( w)=";
//
// textBox6
//
this->textBox6->Location = System::Drawing::Point(562, 70);
this->textBox6->Name = L"textBox6";
this->textBox6->ReadOnly = true;
this->textBox6->Size = System::Drawing::Size(64, 20);
this->textBox6->TabIndex = 2;
//
// label19
//
this->label19->AutoSize = true;
this->label19->Location = System::Drawing::Point(540, 72);
this->label19->Name = L"label19";
this->label19->Size = System::Drawing::Size(16, 13);
this->label19->TabIndex = 1;
this->label19->Text = L"t=";
//
// label18
```

```
                        //
                        this->label18->AutoSize = true;
                        this->label18->Font = (gcnew System::Drawing::Font(L"Microsoft
Sans Serif", 8.25F, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::Point,
                                static_cast<System::Byte>(0)));
                        this->label18->Location = System::Drawing::Point(517, 40);
                        this->label18->Name = L"label18";
                        this->label18->Size = System::Drawing::Size(308, 13);
                        this->label18->TabIndex = 0;
                        this->label18->Text = L"Cauchy\'s formula  for calculation of
resolving power :";
                        //
                        // Form1
                        //
                        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
                        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
                        this->ClientSize = System::Drawing::Size(887, 709);
                        this->Controls->Add(this->tabControl1);
                        this->Name = L"Form1";
                        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterParent;
                        this->Text = L"Simulation of optical instruments";
                        this->Load += gcnew System::EventHandler(this,
&Form1::Form1_Load);
                        this->tabControl1->ResumeLayout(false);
                        this->tabPage1->ResumeLayout(false);
                        this->tabPage1->PerformLayout();
                        this->groupBox2->ResumeLayout(false);
                        this->groupBox1->ResumeLayout(false);
                        this->groupBox1->PerformLayout();
                        this->tabPage2->ResumeLayout(false);
                        this->tabPage2->PerformLayout();
                        this->ResumeLayout(false);

                }
#pragma endregion
                static double width;
                static double height;
                static float a;
                static float b;
           static int error=1;
                static float ab,cd,ef,gh,ij,acth,button,t,w,calc=0;
                        private:   System::Void   button1_Click(System::Object^     sender,
System::EventArgs^  e) {
                                  try{
                         width=Convert::ToDouble(maskedTextBox2->Text);
                    height=Convert::ToDouble(maskedTextBox1->Text);
                         if(width>=0 && height>=0){error=0;}
                         {
                                 this->tabPage1->Hide();
                            this->tabPage1->Show();
                                 this->tabPage2->Show();
            this->tabPage2->Show();

                         }
                           }//try
                         catch( Exception^ e ){error=1; MessageBox::Show("invalid
                value");}
                         }
private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^  e) {
                         this->label3->Hide ();
                 this->label4->Hide ();
                         this->label5->Hide ();
                         this->label6->Hide ();
                         this->label7->Hide ();
                         this->label8->Hide ();
                         this->label9->Hide ();
                         this->label10->Hide ();
                         this->label11->Hide ();
                         this->label12->Hide ();
                }
private: System::Void tabPage1_Paint(System::Object^ sender,
System::Windows::Forms::PaintEventArgs^  e) {
                         if(error==0){
                                 Pen^ p = gcnew Pen( Color::Blue,1.0f );
                       Graphics^ g = e->Graphics;
                       g->DrawEllipse(p,300,30,width*5,500);
                                  g->DrawLine(p,300,30+250,300+width*5,30+250);
                       a=(width*5/2)*(width*5/2);
                       b=(500/2)*(500/2);
                                  int c=0;
                                  for(int i=1;i<=4;i++){
                                  c=c+1;
                                  int y=30+500*c/(2*5);
```

```
                        int x=Math::Sqrt(a*(1-((y-30-250)*(y-280)/b)))+300+width*5/2;
                        Point p1=Point(x,y);
                        Point p2=Point(2*(300+width*5/2)-x,y);
                        /*int vb=2*(300+width*5/2)-x;
                        MessageBox::Show(" "+vb);*/
                        g->DrawLine(p,p1,p2);
                        if(c==1){this->label12->Location = System::Drawing::Point(x+8, y);
                        this->label12->Show();
                        this->label11->Location = System::Drawing::Point(2*(300+width*5/2)-x-15,y);
                        this->label11->Show();}//1stif
            if(c==2){this->label10->Location = System::Drawing::Point(x+10, y);
                        this->label10->Show();
                        this->label9->Location = System::Drawing::Point(2*(300+width*5/2)-x-20,y);
                        this->label9->Show();}//1stif
            if(c==3){this->label8->Location = System::Drawing::Point(x+10, y);
                        this->label8->Show();
                        this->label7->Location = System::Drawing::Point(2*(300+width*5/2)-x-15,y);
                        this->label7->Show();}//1stif
            if(c==4){this->label6->Location = System::Drawing::Point(x+10, y);
                        this->label6->Show();
                        this->label5->Location = System::Drawing::Point(2*(300+width*5/2)-x-15,y);
                        this->label5->Show();}//1stif
                        }//endfor
```

```
                            this->label3->Location    =    System::Drawing::Point(270,
                    30+250);
                            this->label3->Show();
          this->label4->Location = System::Drawing::Point(310+width*5,30+250);
                            this->label4->Show();
                            ////////////////////////////////////////////////////////////////////////////
                            float he=height*30;
                            float wi=(height/3)*30;
                            float ex=he-wi/2;
                            float calc=he/2*he/2-(ex-he/2)*(ex-he/2);
                            float y=(Math::Sqrt(calc))+he/2;
                            acth=(he/2-(he-y));
                            this->textBox1->Text=Convert::ToString(wi);
                            ab=wi;
                            y=he/2-(acth/5);
                            ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                            this->textBox2->Text=Convert::ToString(2*(ex-
he+wi/2));
                            cd=2*(ex-he+wi/2);
          /*g->DrawLine(p,ex,y,he-2*(ex-he+wi/2),y);*/
                            y=he/2-2*(acth/5);
                            ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                            this->textBox3->Text=Convert::ToString(2*(ex-
he+wi/2));
                            ef=2*(ex-he+wi/2);
        /*   g->DrawLine(p,ex,y,he-wi/2,y);*/
                            y=he/2-3*(acth/5);
                            ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
```

```
                        this->textBox4->Text=Convert::ToString(2*(ex-he+wi/2));
                         gh=2*(ex-he+wi/2);
        /* g->DrawLine(p,ex,y,he-wi/2,y);*/
                         y=he/2-4*(acth/5);
                         ex=Math::Sqrt(he/2*he/2-(y-he/2)*(y-he/2))+he/2;
                         this->textBox5->Text=Convert::ToString(2*(ex-
he+wi/2));
                         ij=2*(ex-he+wi/2);
        /*  g->DrawLine(p,ex,y,he-wi/2,y);*/

                  }//endif
            }
private:      System::Void      tabPage2_Paint(System::Object^           sender,
System::Windows::Forms::PaintEventArgs^  e) {
        Pen^ p = gcnew Pen( Color::Blue,1.0f );
                        Pen^ p2 = gcnew Pen( Color::Green,1.0f );
        Graphics^ g = e->Graphics;
                        if(button==1){
                        this->textBox6->Text=Convert::ToString(ij);
                        Point point1 = Point(200,400);
                        Point point2 = Point(200+6*ij,400);
                        Point point3 = Point(200+3*ij,400-6*acth/5);
                        array<Point>^ curvePoints = {point1,point2,point3};
                        g->DrawPolygon(p,curvePoints);}
                        ;
                        if(button==2){
                        this->textBox6->Text=Convert::ToString(gh);
                        Point point1 = Point(200,600);
                        Point point2 = Point(200+6*gh,600);
```

```
                    Point point3 = Point(200+3*gh+3*ij,600-6*acth/5);
                    Point point4 = Point(200+3*gh-3*ij,600-6*acth/5);
                    array<Point>^            curvePoints            =
{point1,point2,point3,point4};
                    g->DrawPolygon(p,curvePoints);
                    float m=(600-(600-6*acth/5))/(200-(200+3*gh-3*ij));
                    float x=200+3*gh;
                    float y=m*(x-200)+600;
                    g->DrawLine(p2,x,y,200+3*gh+3*ij,600-6*acth/5);
                    g->DrawLine(p2,x,y,200+3*gh-3*ij,600-6*acth/5);
                    }
                    if (button==3){
                    this->textBox6->Text=Convert::ToString(ef);
                    Point point1 = Point(200,600);
                    Point point2 = Point(200+6*ef,600);
                    Point point3 = Point(200+3*ef+3*gh,600-6*acth/5);
                    Point point4 = Point(200+3*ef-3*gh,600-6*acth/5);
                    array<Point>^            curvePoints            =
{point1,point2,point3,point4};
                    g->DrawPolygon(p,curvePoints);
                    float m=(600-(600-6*acth/5))/(200-(200+3*ef-3*gh));
                    float x=200+3*ef;
                    float y=m*(x-200)+600;
                    g->DrawLine(p2,x,y,200+3*ef+3*gh,600-6*acth/5);
                    g->DrawLine(p2,x,y,200+3*ef-3*gh,600-6*acth/5);
                    }
                    if (button==4){
                    this->textBox6->Text=Convert::ToString(cd);
                    Point point1 = Point(200,600);
```

```
                Point point2 = Point(200+2*cd,600);
                Point point3 = Point(200+1*cd+1*ef,600-2*acth/5);
                Point point4 = Point(200+1*cd-1*ef,600-2*acth/5);
                array<Point>^           curvePoints           =
{point1,point2,point3,point4};
                g->DrawPolygon(p,curvePoints);
                float m=(600-(600-2*acth/5))/(200-(200+1*cd-1*ef));
                float x=200+1*cd;
                float y=m*(x-200)+600;
                g->DrawLine(p2,x,y,200+1*cd+1*ef,600-2*acth/5);
                g->DrawLine(p2,x,y,200+1*cd-1*ef,600-2*acth/5);
                }
                if (button==5){
                this->textBox6->Text=Convert::ToString(ab);
                Point point1 = Point(200,600);
                Point point2 = Point(200+2*ab,600);
                Point point3 = Point(200+1*ab+1*cd,600-2*acth/5);
                Point point4 = Point(200+1*ab-1*cd,600-2*acth/5);
                array<Point>^           curvePoints           =
{point1,point2,point3,point4};
                g->DrawPolygon(p,curvePoints);
                float m=(600-(600-2*acth/5))/(200-(200+1*ab-1*cd));
                float x=200+1*ab;
                float y=m*(x-200)+600;
                g->DrawLine(p2,x,y,200+1*ab+1*cd,600-2*acth/5);
                g->DrawLine(p2,x,y,200+1*ab-1*cd,600-2*acth/5);
                }

        }
```

```
private: System::Void button2_Click(System::Object^  sender, System::EventArgs^  e) {
                         button=5;
                 }
private: System::Void button3_Click(System::Object^  sender, System::EventArgs^  e) {
                         button=4;
                 }
private: System::Void button4_Click(System::Object^  sender, System::EventArgs^  e) {
                         button=3;
                 }
private: System::Void button5_Click(System::Object^  sender, System::EventArgs^  e) {
                         button=2;
                 }
private: System::Void button6_Click(System::Object^  sender, System::EventArgs^  e) {
                         button=1;
                 }
private: System::Void button7_Click(System::Object^  sender, System::EventArgs^  e) {
                         try{
                                 calc=0;
                            w=Convert::ToDouble(this->textBox7->Text);
                           }
                         catch(Exception^          e          ){calc=1;this->textBox8->Text="";
MessageBox::Show("invalid  value of wavelength");}
                         if(calc==0){
                                 double base=Convert::ToDouble(this->textBox6->Text);
                                 this->textBox8-
>Text=Convert::ToString((2*4.608)/(w*w*w*.00000000001)                         +
(4*6.88)/(w*w*w*w*w*(Math::Pow(10,-22)))*base);
                         }
                 }
```

Chapter 5

CONCLUSION

CONCLUSION:

Different subroutines have been made accordingly for different number of finite element (prisms) in the lens. The system has been analyzed for different values of inputs. OOPs is used in the project to make the program look simpler. Also if the program is required to be modified in future it will be easy since object oriented approach is used.

The program has a graphical interface which makes it easy for the user to make the simulation.

Lens is used mainly in cameras where it will focus the object. Software for lens will make it easy to analyze which lens to use where and when. The advantage of a software design of Optical Systems is that the specifications of optical instruments to be used (say lens) can be found out without actually determining them at the work place but can be found out in the computer.

REFERENCES:

- Optics-Ajay Ghatak

  Tata Mc Graw Hill
- Programming in Turbo C++ by Robert Lafore