Interpolation/Extrapolation and Its Application to Solar Cells

Computer Course Work

Phys601

Department of Physics & Astrophysics

University of Delhi

Submitted by:

Vinamrita Singh

Introduction

Over the last few decades, computers have become part of everyday life. Computational physics provides a means to solve complex numerical problems. An advantage of computational physics is that one can start with a simple problem which is easily solvable analytically. The analytical solution illustrates the underlying physics and allows one the possibility to compare the computer program with the analytical solution. Once a program has been written which can handle the case with the typical physicist's approximation, then you add more and more complex real-world factors.

An important part in a scientist's life is the interpretation of measured data or theoretical calculations. Usually when you do a measurement you will have a discrete set of points representing your experiment. For simplicity, we assume your experiment to be represented by pairs of values: an independent variable "*x*," which you vary and a quantity "*y*," which is the measured value at the point *x*. As an illustration, consider a radioactive source and a detector, which counts the number of decays. In order to determine the half-life of this source, you would count the number of decays N_0, N_1, N_2, …N_k at times t_0, t_1, t_2… t_k. In this case *t* would be your independent variable, which you hopefully would choose in such a way that it is suitable for your problem. However, what you measure is a discrete set of pairs of numbers (t_k, N_k) in the range of [t_0, t_k]. In order to extract information from such an experiment, we would like to be able to find an analytical function which would give us N for any arbitrary chosen point t. But, sometimes trying to find an analytical function is impossible, or even though the function might be known, it is too time consuming to calculate or we might be only interested in a small local region of the independent variable. To illustrate this point, assume your radioactive source is 241Am, an α-emmiter. Its half-life is $\tau_{1/2}$ = 430 years. Clearly you cannot determine the half-life by measuring it. Because it is very slowly decaying you probably will measure the activity over a longer time period, say every Monday for a couple of months. After five months you would stop and look at the data. One question you might want to answer is: what was the activity on Wednesday of the third week? Because this day is inside your range of [t_0, t_k] you would use interpolation techniques to determine this value. If, on the other hand, you want to know the activity eight months from the end of your measurement, you would extrapolate to this point from the previous series of measurements. The idea of interpolation is to select a function g(x) such that $g(x_i)= f_i$

for each data point i and that this function is a good approximation for any other x lying between the original data points.

Definitions:

1. **Interpolation**: Estimating the attribute values of locations that are *within* the range of available data using known data values.
2. **Extrapolation**: Estimating the attribute values of locations *outside* the range of available data using known data values.

Interpolation:

There are many methods for interpolation. Interpolation is done by generating a function which best fits the known points. Interpolation is carried out using approximating functions such as:

1. Polynomials
2. Trigonometric functions
3. Exponential functions
4. Fourier methods

Following interpolating methods are most polular:

1. Lagrange Interpolation (unevenly spaced data)
2. Newton's Divided Difference (evenly spaced data)
3. Central difference method

LANGRANGE'S INTERPOLATION:

Suppose that our data pairs are $(x_0,f(x_0))$, $(x_1,f(x_1))$,…, $(x_n,f(x_n))$. The Langrange's polynomial is given by:

$$P(x)=\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}f(x_0)+\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}f(x_1)$$
$$+\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}f(x_2)+\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}f(x_3)$$

In short, it can be written as:

$$P(x)=\sum_{j=1}^{n}P_j(x)$$

Where

$$P_j(x)=y_j\prod_{\substack{k=1\\k\neq j}}^{n}\frac{x-x_k}{x_j-x_k}$$

Hence, by putting the value of x_i and y_i we can calculate the value of x at any unknown point.

NEWTON'S FORMULAE:

Another popular method for interpolation is Newton's formula. There are three types of formulae:

1. Forward difference interpolation formula
2. Backward difference interpolation formula
3. Divided difference interpolation formula

Forward difference interpolation formula:

$$P(x) = y_0 + p\Delta y_0 + \frac{p(p-1)}{2}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{3}\Delta^3 y_0 + \frac{p(p-1)(p-2)(p-3)}{4!}\Delta^4 y_0 + \dots$$

Where $p = \frac{x - x_0}{h}$

Backward difference interpolation formula:

$$P(x) = y_n + p\nabla y_n + \frac{p(p+1)}{2!}\nabla^2 y_n + \dots + \frac{p(p+1)(p+2)}{3!}\nabla^3 y_n + \frac{p(p+1)(p+2)(p+3)}{4!}\nabla^4 y_n + \frac{p(p+1)(p+2)(p+3)(p+4)}{5!}\nabla^5 y_n$$

Divided difference interpolation formula:

A divided difference is defined as the difference in the function values at two points, divided by the difference in the values of the corresponding independent variable.

Thus, the first divided difference at point is defined as

$$f[x_0, x_1] = \frac{f_0 - f_1}{x_0 - x_1}$$

$P(x) = y_0 + f[x_0, x_1](x - x_0) + f[x_0, x_2](x - x_0)(x - x_1) + f[x_0, x_3](x - x_0)(x - x_1)(x - x_2) + \dots$

EXTRAPOLTION

There are many methods of extrapolation. The following methods will be briefly outlined:

1. Linear Extrapolation

2. Polynomial Extrapolation
3. Conic Extrapolation

Linear Extrapolation:

- Linear Extrapolation means creating a tangent line at the end of the known data and extending it beyond that limit.

- Linear extrapolation will provide good results only when used to extend the graph of an approximately linear function or not too far beyond the known data.

- If the two data points nearest to the point x_* to be extrapolated are (x_k, y_k) and (x_{k-1}, y_{k-1}), linear extrapolation gives the function

$$y(x_*) = y_{k-1} + \frac{x_* - x_{k-1}}{x_k - x_{k-1}}(y_k - y_{k-1}).$$

Polynomial Extrapolation:

A polynomial curve can be created through the entire known data or just near the end. The resulting curve can then be extended beyond the end of the known data. Polynomial extrapolation is typically done by means of Lagrange interpolation or using Newton's method of finite differences to create a Newton series that fits the data. The resulting polynomial may be used to extrapolate the data.

Conic Extrapolation:

A conic section can be created using five points near the end of the known data. If the conic section created is an ellipse or circle, it will loop back and rejoin itself. A parabolic or hyperbolic curve will not rejoin itself, but may curve back relative to the X-axis. This type of extrapolation could be done with a conic sections template (on paper) or with a computer.

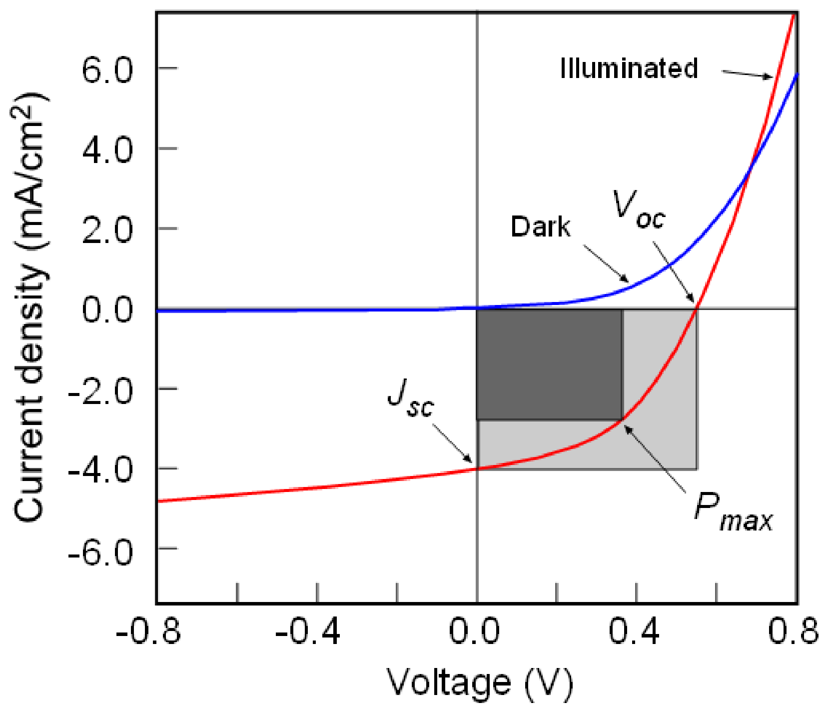
Introduction to Solar Cells:

Solar cell technology is an important technology, which converts directly the incident light into electricity. Solar cells could be a godsend for the 2 billion-plus people who don't have access to electricity. Improved solar cells became a reliable source of electricity for satellites, but their price horrified electric utilities, so they were only used when cheaper alternatives were not available. Because of very high cost of Si solar cells the scientists and technologists are searching for some different kind of solar cells which could produce electricity at very low cost. Organic solar cells have been recognized as one of the important sources to produce cheap and clean energy.

Organic solar cell research has developed during the past 30 years, but especially in the last decade it has attracted scientific and economic interest triggered by a rapid increase in power conversion efficiencies. This was achieved by the introduction of new materials, improved materials engineering, and more sophisticated device structures. Today, solar power conversion efficiencies in excess of 8% have been accomplished with several device concepts. Though efficiencies of these thin-film organic devices have not yet reached those of their inorganic counterparts (≈ 10–20%); the perspective of cheap production (employing, e.g., roll-to-roll processes) drives the development of organic photovoltaic devices further in a dynamic way.

Performance evaluation of an organic solar cell

The performance of an organic solar cell is determined by measuring its current-voltage (*J-V*) characteristics in dark and under sun light illumination. The *J-V* characteristic of a solar cell under illumination gives several photovoltaic parameters such as short circuit current (J_{sc}), open circuit voltage (V_{oc}), fill factor (*FF*) and power conversion efficiency (η) at an instance. The typical *J-V* characteristics of an organic solar cell in dark and under light illumination are shown in Fig.5. The J_{sc}, V_{oc}, *FF* and η of a solar cell are discussed below in brief.



Schematic representation of dark and illuminated J-V characteristics of an organic solar cell.

Short circuit current (J_{sc})

J_{sc} is the current driven from the illuminated solar cell under short circuit condition i.e. when both the electrodes are connected through a highly conducting wire. In this case the voltage across the cell will be zero. In the measured *J-V* characteristics J_{sc} is the current in the cell at zero applied voltage. J_{sc} is also known as photocurrent in short circuit condition. The photocurrent depends directly on material properties and the illumination intensity. The illuminated current in the *J-V* characteristics is the combination of both the dark and photocurrent. It can be seen clearly from Fig. 5 that in the first quadrant the illuminated current intersects the dark current. The intersection of illuminated current with dark current corresponds to the zero photocurrent at the intersection point. This point of intersection of dark and illuminated characteristics corresponds to the built-in-voltage (V_{bi}) in the sample. It is interesting to see that unlike Si solar cells, beyond V_{bi} the illuminated current in organic solar cells becomes more that the dark current. Below the intersection point $V_{bi} > V > 0$, the photocurrent flows in the opposite direction to the dark current whereas for $V > V_{bi}$ the photocurrent get reversed in the direction due to change in the direction of the effect field $[(V_{bi}-V)/d]$ in the sample and it is now being added to the dark current.

Open circuit voltage (V_{oc})

V_{oc} is the voltage developed across the electrodes of an illuminated solar cell when no current is driven from the cell i.e. the cell is in open circuit condition. In *J-V* characteristics V_{oc} is the voltage at which the net current in the cell is zero. The origin of V_{oc} in organic solar cells is still not very clear. V_{oc} has been correlated to the difference of HOMO of donor and LUMO of acceptor, whereas a direct dependence of V_{oc} on work function difference of the electrodes (ΔW) has also been observed. V_{oc} has been observed to vary with donor/acceptor, temperature, electrode materials, illumination intensity and thickness of the active layer.

Fill factor (*FF*)

FF of a solar cell is the measure of the power that can be extracted from the cell and is defined as,

$$FF = \frac{J_{\max} V_{\max}}{J_{sc} V_{oc}}$$

where J_{max} and V_{max} are the current and voltage corresponding to the maximum power point on the *J-V* characteristics. *FF* is the measure of the shape of the *J-V* curve and can also be defined as the ratio of areas of the two shaded rectangles, area of smaller rectangle over the larger one, in Fig.5. Ideally *FF* should be 1.0 but because of losses due to transport and recombination of charge carriers it always remains less than 1.0. The devices having high *FF* consists of convex shaped *I-V* characteristics ($d^2I/dV^2 > 0$) whereas devices with low *FF* consists of linear or concave shaped *I-V* characteristics ($d^2I/dV^2 < 0$).

Power conversion efficiency (η)

The power conversion efficiency is simply the ultimate measure of the efficiency of the device to convert the light photons into electricity. The power conversion efficiency is directly related to J_{sc}, V_{oc} and *FF* and is calculated from

$$\eta = \frac{I_{sc} V_{oc} FF}{P_{light}}$$

Where P_{light} is the incident optical power. The efficiency of an organic solar cell is determined by light absorption in the active layer, exciton diffusion, exciton dissociation, charge transport and charge collection. Active layer refers to the layer where the majority of the incident light is absorbed and charge carriers are generated. The photocurrent generated at different wavelengths is known as spectral response ($S(\lambda)$) and is defined as

$$S(\lambda) = \frac{J_{sc}(\lambda)}{\phi(\lambda)}$$

where $J_{sc}(\lambda)$ is the short circuit current at wavelength λ, $\phi(\lambda)$ is the photon flux at wavelength λ.

UV-visible absorption spectrophotometer

Spectrophotometer is an instrument which measures and compares the incident, reflected and transmitted light of a sample. There are three types of spectroscopic measurements; Transmittance, Absorbance and Reflectance. Transmittance and Absorbance measurements are made on transparent liquids and clear solids such as thin films and filters. Reflectance measurements are performed on the completely opaque or relatively thick samples. The ratio of the two light intensities, transmitted light (I) over the incident light (I_0) is known as the transmittance of the sample. And the absorbance is calculated by

$$A = -\log\left(\frac{I}{I_0}\right)$$

where I and I_0 are respectively the transmitted and incident light intensities. I_0 is also referred to as the background or reference beam, which is measured with only the solvent or the substrate same as used for I. Different materials absorb different wavelengths of light. Therefore, the wavelength of maximum absorption by a material is one of the characteristic properties of that material.

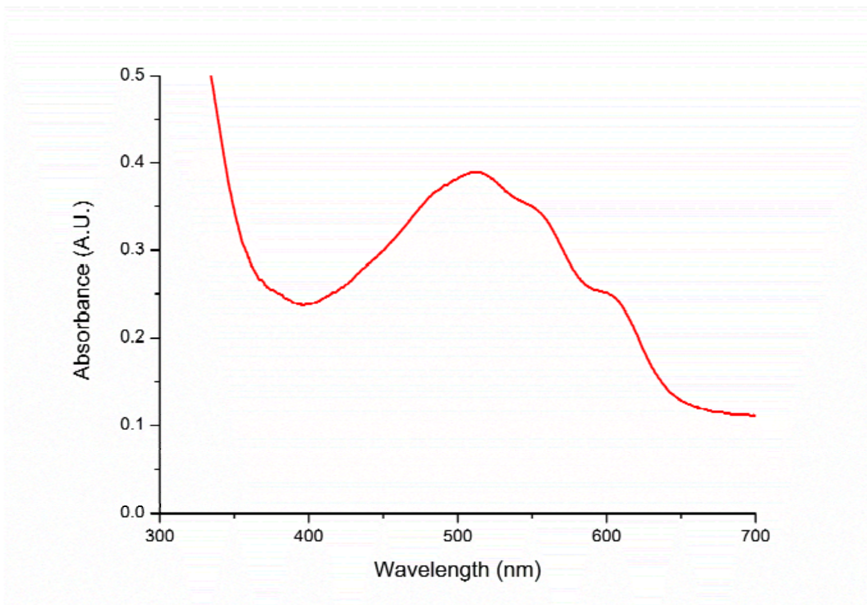
Fabrication Process

1. ITO coated glass substrates are etched in the desired pattern.
2. Substrates are cleaned with soap solution in ultrasonic bath for 15 minutes followed by rinsing in distilled water.
3. Substrates are then subsequently boiled in acetone, trichloroethylene and iso-propanol for 15 minutes each. After cleaning the substrates are dried in vacuum oven for 30 minutes.
4. To obtain hydrophilic surface and to increase the wet-ability the substrates are exposed to oxygen plasma for 5 minutes.

5. PEDOT:PSS layer is spin coated at 2000 rpm for 2 minutes resulting in a film thickness of approximately 50 nm. It is then annealed at a temperature of 100 °C in vacuum oven.
6. The P3HT:PCBM solution (1:1 ratio by weight) with chloro-benzene as solvent is spin coated at 1000 rpm to form 150 nm thick film.
7. The sample formed is annealed at 120°C for 30 minutes. Finally Al is thermally evaporated for the electrode.

Analysis of UV-Visible absorption spectrum:

The figure below shows the UV-Visible absorption spectrum of P3HT:PCBM blend.



The energy band gap can be calculated

using absorption spectrum.

Absorption coefficient, α, can be extracted from absorbance:

$$\alpha = A/t;$$

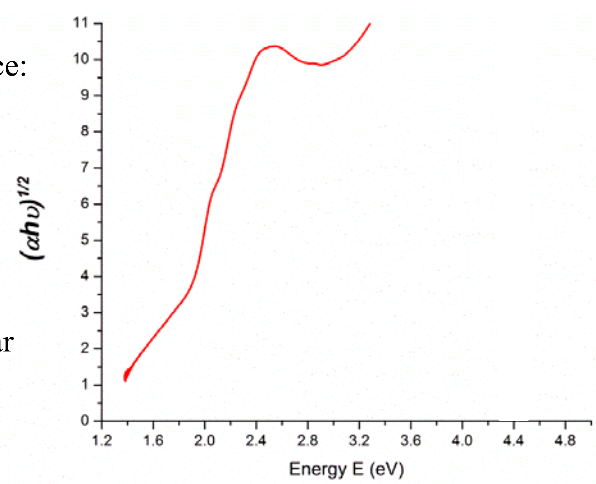
where t = thickness

A graph between $(\alpha h\nu)^{1/2}$ and Energy E can be

used to find the energy band gap by extrapolating the linear

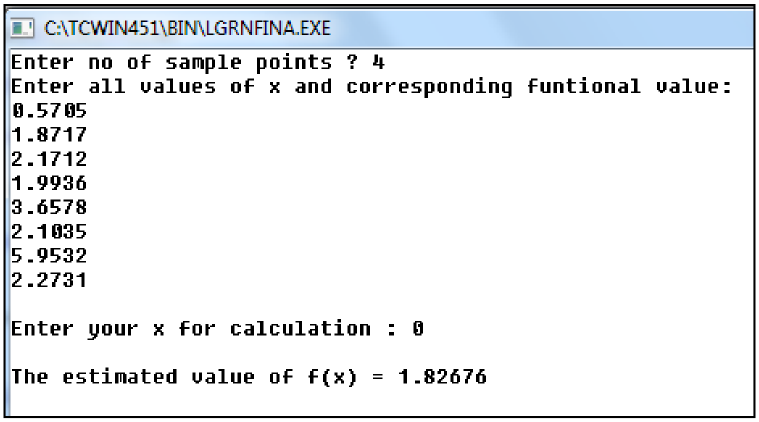
region.

Here, $E\ (eV) = 1240/\lambda$



The energy band gap of P3HT:PCBM lies between 1.85-2 eV.

The results obtained using Lagrange's extrapolation formula are:



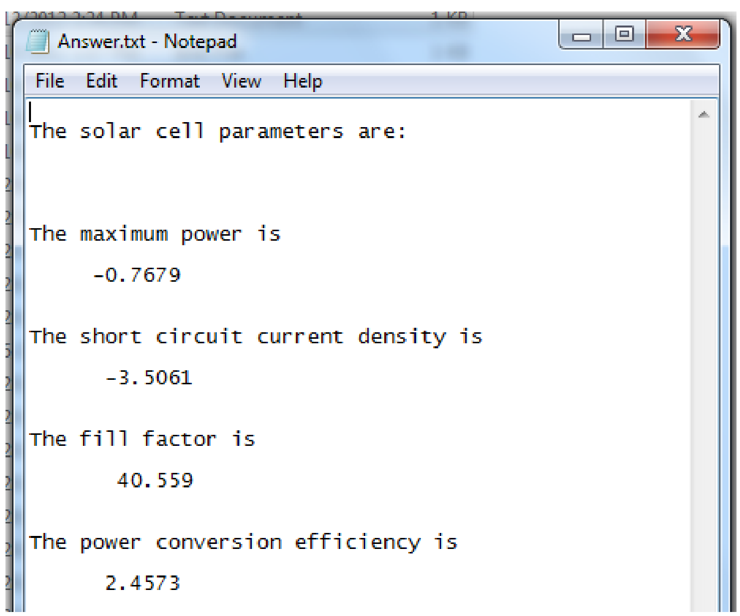
```
C:\TCWIN451\BIN\LGRNFINA.EXE
Enter no of sample points ? 4
Enter all values of x and corresponding funtional value:
0.5705
1.8717
2.1712
1.9936
3.6578
2.1035
5.9532
2.2731

Enter your x for calculation : 0

The estimated value of f(x) = 1.82676
```

Analysis of IV Characteristics:

The IV characteristics of samples were taken using Keithley Source meter under 1 sun. The data was used to calculate the short circuit current, maximum power, fill factor, and power conversion efficiency using the above described formulae. C++ programming was made to calculate the parameters. The results obtained matched well with manually calculated values and are tabulated below.



```
Answer.txt - Notepad
File Edit Format View Help

The solar cell parameters are:


The maximum power is
     -0.7679

The short circuit current density is
      -3.5061

The fill factor is
       40.559

The power conversion efficiency is
      2.4573
```

Program Codes

Lagrange Interpolation

```
#include<iostream.h>
#include<conio.h>
int main()
{
   int n,i,j;
   float mult,sum=0,x[10],f[10],a;
   clrscr();
   cout<<"Enter no of sample points ? ";
   cin>>n;
   cout<<"Enter all values of x and corresponding funtional value: "<<endl;
   for(i=0;i<n;i++)
      cin>>x[i]>>f[i];

   cout<<"\nEnter your x for calculation : ";
   cin>>a;

   for(i=0;i<=n-1;i++)
   {
   mult=1;
      for(j=0;j<=n-1;j++)
      {
         if(j!=i)
         mult*=(a-x[j])/(x[i]-x[j]);
      }
      sum+=mult*f[i];
   }
   cout<<"\nThe estimated value of f(x) = "<<sum;
   getch();
   return 0;
}
```

Newton’s Forward Interpolation

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>

float x[15],y[15],z[15][15];
int n;
void horizontal_line()
{
	char ch=196;
	for(int i=0;i<75;i++)
		cout<<ch;
	cout<<endl;
}
void enter()
{
	int i;
	cout<<"\nenter the no.(n)";
	cin>>n;
	cout<<"\nenter the eqally spaced values of x::>\n";
	for(i=0;i<n;i++)
		cin>>x[i];
	cout<<"\nenter the corresponding values of y::>\n";
	for(i=0;i<n;i++)
		cin>>y[i];
}
void forward_difference_table()
{
	int i,j;
	for(i=0;i<n-1;i++)
		z[i][0]=y[i+1]-y[i];
	for(j=1;j<n-1;j++)
		for(i=0;i<n-1-j;i++)
			z[i][j]=z[i+1][j-1]-z[i][j-1];
	cout<<"\n\n\tFORWARD DIFFRENCE TABLE IS:::>";
	cout<<"\n\n";
	for(i=0;i<n-1;i++)
	{
		cout<<setw(8)<<setprecision(5)<<x[i]<<setw(8)<<setprecision(4)<<y[i];
		for(j=0;j<n-1-i;j++)
			cout<<setw(10)<<setprecision(2)<<z[i][j];
		cout<<endl;
//		horizontal_line();
	}
	cout<<setw(8)<<setprecision(5)<<x[n-1]<<setw(8)<<y[n-1]<<endl;
//	horizontal_line();
}
float fact(int j)
{
	float f=1;
	if(j==0)
		return 1;
	for(int i=j;i>=1;i--)
		f*=j;
```

```
        return f;
}
float calculate(float u)
{
        float h,p,res,pro;
        int flag=-1,i,k,j,pos;
        for(i=0;i<n-1;i++)
        {
                if((x[i]<=u)&&(x[i+1]>=u))
                {
                        pos=i;
                        flag=0;
                }
        }
        if(flag==-1)
                pos=0;
        h=x[1]-x[0];
        p=(u-x[pos])/h;
        res=y[pos];
        for(j=0;j<n-1-pos;j++)
        {pro=1;
        for(k=0;k<=j;k++)
                pro*=(p-k);

        res+=(pro*z[pos][j])/fact(j+1);
        }
        return res;
        }

void main()
{ float u;

        enter();
        forward_difference_table();
RITS:
        cout<<"\nenter the value of x for which y(x) needs to be calculated\n";
        cin>>u;
        cout<<"\n\n\twhen x="<<u<<"\ty["<<u<<"]=="<<calculate(u);
        getch();
        char ch;
        cout<<"\n\n\n Do you want to do more interpolation(y/n)";
        cin>>ch;
        if(ch=='y'||ch=='Y')
                goto RITS;

        getch();
}
```

Newton's Backward Interpolation

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>

float x[15],y[15],z[15][15];
int n;
void horizontal_line()
{
		char ch=196;
		for(int i=0;i<75;i++)
				cout<<ch;
		cout<<endl;
}
void enter()
{
		int i;
		cout<<"\nenter the no.(n)";
		cin>>n;
		cout<<"\nenter the eqally spaced values of x::>\n";
		for(i=0;i<n;i++)
				cin>>x[i];
		cout<<"\nenter the corresponding values of y::>\n";
		for(i=0;i<n;i++)
				cin>>y[i];
}
void backward_difference_table()
{
		int i,j;
		for(i=1;i<n;i++)
				z[i][0]=y[i]-y[i-1];
		for(j=1;j<n-1;j++)
				for(i=j+1;i<n;i++)
						z[i][j]=z[i][j-1]-z[i-1][j-1];
		cout<<"\n\n\tBACKWARD DIFFRENCE TABLE IS:::>";
		cout<<"\n\n";
		cout<<setw(8)<<setprecision(4)<<x[0]<<setw(8)<<y[0];
		cout<<endl;
//		horizontal_line();
		for(i=1;i<n;i++)
		{
				cout<<setw(8)<<setprecision(5)<<x[i]<<setw(8)<<setprecision(4)<<y[i];
				for(j=0;j<=i-1;j++)
						cout<<setw(10)<<setprecision(2)<<z[i][j];
				cout<<endl;
//				horizontal_line();
		}

}
float fact(int j)
{
		float f=1;
		if(j==0)
				return 1;
```

```
	for(int i=j;i>=1;i--)
		f*=j;
	return f;
}
float calculate(float u)
{
	float h,p,res,pro;
	int flag=-1,i,k,j,pos;
	for(i=0;i<n-1;i++)
	{
		if((x[i]<=u)&&(x[i+1]>=u))
		{
			pos=i+1;
			flag=0;
		}
	}
	if(flag==-1)
		pos=n-1;
	h=x[1]-x[0];
	p=(u-x[pos])/h;
	res=y[pos];
	for(j=0;j<pos;j++)
	{pro=1;
	for(k=0;k<=j;k++)
		pro*=(p-k);

	res+=(pro*z[pos][j])/fact(j+1);
	}
	return res;
	}

void main()
{ float u;

	enter();
	backward_difference_table();
abc:
	cout<<"\nenter the value of x for which y(x) needs to be calculated: ";
	cin>>u;
	cout<<"\n\n\twhen x="<<setprecision(4)<<u<<"\ty["<<setprecision(4)<<u<<"]=="<<calculate(u);
	getch();
	char ch;
	cout<<"\n\n\n Do you want to do more interpolation(y/n)";
	cin>>ch;
	if(ch=='y'||ch=='Y')
		goto abc;

	getch();
}
```

For solar cell's parameter calculations

```
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <conio.h>
#include <math.h>

int main()
{
int i;
float x[100], y[100], z[100],  min, num, jsc, FF, PCE, A;
float c1[100];
ifstream ins; // input stream
ifstream ins1; // input stream
ofstream outs; // output stream
ins.open("volt.txt");
ins1.open("crnt.txt");
outs.open("Answer.txt");
//outs.open("power.txt");
cout<<"\n Enter the pixel of cell\n";
cin>>A;
for (i=0; i<100; i++)
{
ins >> x[i];
ins1 >> y[i];
//power = x[i]*y[i];
z[i] = x[i]*y[i];
c1[i] = y[i];
//outs << "" << setprecision(10) << power << endl;
//outs1 << "" << setprecision(6) << absorb << endl;
}
ins.close();
//outs.close();
ins1.close();
min=0;
for (i=50; i<80; i++)
{
if (z[i]<min)
min = z[i];
else
num = min;
}
cout<<"\nThe maximum power is "<< setprecision(10)<<num;
jsc = c1[50];
cout<<"\n\nThe short circuit current density is\n ";
cout<<setw(8)<< setprecision(10)<<jsc;
FF = num/(jsc*0.54)*100;
cout<<"\n\nThe fill factor is\n ";
cout<<setw(8)<< setprecision(10)<<FF;
PCE = -jsc*A*0.54*FF;
cout<<"\n\nThe power conversion efficiency is\n ";
cout<< setw(8)<< setprecision(10)<<PCE;
outs<<"\nThe solar cell parameters are:\n\n";
outs<<"\n\n\nThe maximum power is \n\n" <<setw(12)<<setprecision(5)<<num;
```

```
outs<<"\n\n\nThe short circuit current density is\n\n ";
outs<<setw(12)<< setprecision(5)<<jsc;
outs<<"\n\n\nThe fill factor is\n\n ";
outs<<setw(12)<< setprecision(5)<<FF;
outs<<"\n\n\nThe power conversion efficiency is\n\n";
outs<<setw(12)<< setprecision(5)<<PCE << endl;
outs.close();
return 0;
}
```