**Abstract**

Our topic is “Mathematical and Computational study of all aspects of RAMANUJAN’S METHOD”.

For finding the smallest root of a given equation, we use Ramanujan’s method. First of all, Ramanujan’s method is an iterative method. According to Ramanujan’s method, we calculate the smallest root of a give equation and verify the with available methods (Bair stow Method). We are implementing method by using the object oriented programming language of the complier turbo c7

**INTRODUCTION**

Srinivasa Ramanujan was born on 22 December 1887 in Erode, Madras Presidency, to K. Srinivasa Iyengar and his wife Komalatammal. His family was a humble one and his father worked as a clerk in a sari shop. His mother gave birth to several children after Ramanujan, but none of them survived infancy.

He soon discovered a book on advanced trigonometry written by S. L. Loney which he mastered by the time he was 13. He proved to be brilliant student and won several merit certificates and academic awards.

In 1903, he got his hands on a book called ‘A Synopsis of Elementary Results in Pure and Applied Mathematics’ by G.S. Carr which was a collection of 5000 theorems. He was thoroughly fascinated by the book and spent months studying it in detail. This book is credited to have awakened the mathematical genius in him.

Srinivasa Ramanujan was an Indian mathematician who made significant contributions to **mathematical analysis, number theory, continued fractions and Ramanujan method.** We are only concerned about Ramanujan method in this project .

First of all**, Ramanujan’s method** is an iterative method. For finding the smallest root of a given equation, we use Ramanujan’s method According to Ramanujan’s method, we calculate the smallest root of a give equation and verify the with available methods (Bair stow Method). We are implementing method by using the oops of the complier Turbo c7

It is not surprising to know that Ramanujan’s method is used to find approximated value only. Discrete Mathematics always heads us towards an approximated value. According to Ramanujan’s method, the smallest root, , is given by :

In coding part, we have used attributes like Class, Input/Output stream header file, function declaration and calling etc.

( 22 December 1887 - 26 April 1920 )

**STEPS TO SOLVE GIVEN EQUATION USING RAMANUJAN’S METHOD:-**

Srinivasa Ramanujan (1887-1920) described an iterative method (see Berndt[1985,p.41])which can be used to determine the smallest root of the equation.

(i)

Where f(x) is of the form

(ii)

For smaller values of x,we can write

(iii)

Expanding the left-hand side by binomial theorem,we obtain

(iv)

Comparing the coefficient of like powers of x on both sides of (iv),we get

……………………………………

Without any proof, Ramanujan states that the successive convergents ,viz..,

Approach a root of the equation f(x)=0 where f(x) is given by (ii). The following examples illustrate the application of this method.

**MATERIAL:-**

**2.10 HEADER FILES**

**Table 2.1 Header files**

|  |  |
| --- | --- |
| #include<stdio.h> | <stdio.h> contains the definition of basic\_stdio class template, which implements formatted input and output. |
| #include<conio.h> | The conio.h header is used to console input/output. |
| #include<stdlib.h> | stdlib.h is the header of the general purposestandard library of C programming language which includes functions involving memory allocation, process control, conversions and others. |

**2.11 FUNCTIONS USED**

**Table 2.2 Functions**

|  |  |
| --- | --- |
| void getdata(), | To enter order of the equation displayed on the home screen |
| void getvalue() | To enter all the coffeicients of the given equation. |
| Void method() | To display the smallest root of the given equation by using the ramanujan methods. |
| void precision() | To taking the four values after decimal |

**METHOD:-**

**Source code are worked on Turbo C++**

**2.12 SOURCE CODE**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

class maths

{

Int i,y,d1,a,b,c,d,b1,a4,a5,a6;

float a1,a2,a3,b2,b3,b4,b5,b6,b7,b8,b9;

public:

void get\_data()

{

cout<<"Enter the order of equation(press 1): ";

cin>>y;

}

Void get\_value(float a,floatb,floatc,float d)

{

b1=1;

float a4=(a/d);

float a5=(b/d);

float a6=(c/d);

a1=-a4;

a2=-a5;

a3=-a6;

cout.precision(4);

cout<<"a1 is "<<a1<<endl;

cout<<endl<<"a2 is "<<a2<<endl;

cout<<endl<<"a3 is "<<a3<<endl;

}

void method()

{

a4=a5=a6=0;

b1=1;

b2=b1\*a1;

cout<<"Value of b2 is :"<<b2<<endl;

cout<<" b1/b2 = "<<(b1/b2)<<endl;

b3=((a1\*b2)+(a2\*b1));

cout<<" Value of b3 is :"<<b3<<endl;

cout<<" b2/b3 = "<<(b2/b3)<<endl;

b4=((a1\*b3)+(a2\*b2)+(a3\*b1));

cout<<"Value of b4 is : "<<b4<<endl;

cout<<" b3/b4 = "<<(b3/b4)<<endl;

b5=((a1\*b4)+(a2\*b3)+(a3\*b2));

cout<<"Value of b5 is: "<<b5<<endl;

cout<<"b4/b5 ="<<(b4/b5)<<endl;

b6=((a1\*b5)+(a2\*b4)+(a3\*b3));

cout<<"Value of b6 is: "<<b6<<endl;

cout<<"b5/b6 = "<<(b5/b6)<<endl;

b7=((a1\*b6)+(a2\*b5)+(a3\*b4));

cout<<"Value of b7 is: "<<b7<<endl;

cout<<"b6/b7 = "<<(b6/b7)<<endl;

b8=((a1\*b7)+(a2\*b6)+(a3\*b5));

cout<<"Value of b8 is : "<<b8<<endl;

cout<<"b7/b8 = "<<(b7/b8)<<endl;

b9=((a1\*b8)+(a2\*b7)+(a3\*b6));

cout<<"Value of b9 is : "<<b9<<endl;

cout<<" b8/b9 = "<<(b8/b9)<<endl;

cout<<endl<<"Smallest root of given equation : ";

cout<<endl<<(b8/b9);

}

};

void main()

{

clrscr();

maths m;

inta,b,c,d;

cout<<"Enter the coefficent of x : "<<endl;

cin>>a;

cout<<"Enter the coefficent of x^2 : "<<endl;

cin>>b;

cout<<"Enter the coefficent of x^3 : "<<endl;

cin>>c;

cout<<"Enter the constant value : "<<endl;

cin>>d;

m.get\_data();

m.get\_value(a,b,c,d);

m.method();

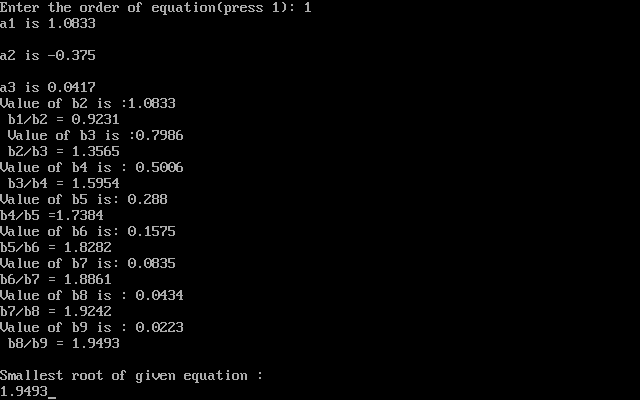
getch();

}

**RESULT:**

**Snapshots:**





8 **APPLICATION:-**

The application of the Ramanujan’s work is seen in statistical physics. For example : Imagine studying the statistics of a gas made of electrons confined to 2D.You could do something complicated like model the exact positions and momenta of many of electrons along with the force between them. Or you can simplify by imagining that the electrons can only occupy positions on a discrete triangular lattice, and instead of a repulsive force you can make the simple approximation that two electrons aren’t allowed to be next to each other. This result is the Hard Hexagon Model. And some work of the **RAMANUJAN’S** appears when you try to model it. Even if it’s not physically realistic, these models share characteristics with more realistic physical models and give useful insight.

In fact a whole bunch of different identities related to Ramanujan’s work can appear when you study these kinds of simple physical models, especially 2-Dimensional models.

7

**CONCLUSION:-**

After doing so much research work, We concluded that Ramanujan methods an easier method to apply and also heads us a step more towards accuracy. During research, We came to know about some interesting facts about **S. Ramanujan**, which inspired us a lot. We come to know about the real world application of this method. We gain so much knowledge while designing the algorithm for the Ramanujan’s Method. Last but not least we can say that **“If you want to find smallest root, Ramanujan’s Method is the best.”**

**REFERENCES:-**

**Books Referred:**

1.Babu Ram ,(Engineering Mathematics),Pearson,2010

2.Ramana B.V.(Higher Engineering Mathematics,TMH,2006)

3.S.S. Sastry( Introductory Methods Of Numerical Analysis,5TH EDITION ,Pearson)

4.Erwin Kreyszig (Advanced Engineering Mathematics,9TH EDITION,Reprint 2011)

**Website Referred:-**

1. <https>[://faculty.math.illinois.edu/~berndt/articles/bgk.pdf](https://faculty.math.illinois.edu/~berndt/articles/bgk.pdf)
2. <https>[://mat.iitm.ac.in/home/sryedida/public\_html/caimna/transcendental/polynomial%20methods/brs%20method.html](https://mat.iitm.ac.in/home/sryedida/public_html/caimna/transcendental/polynomial%20methods/brs%20method.html)

11