Indian Institute of Technology,Guwahati

Depratment Of Mechanical Engineering

Computational Fluid Dynamics (ME543)

Home Assignment 1

Submitted To:
Prof. Anoop K.Das

Submitted By:
Mritunjay
Roll No. 214103004

Table of Content

1. C codes

Problem 1. Gauss Seidal Mthod:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void main()
{
   double u[40][20],error=0,store[40][20];
   int i,j;

   for(i=0;i<21;i++)
   {
   u[0][i]=100;
   store[0][i]=100;
   }

   for(j=1;j<41;j++)
   {
      for(i=0;i<21;i++)
      {
         u[j][i]=0;
         store[j][i]=0;

      }
   }
   do
```

```
{
   error=0;
   for(j=1;j<40;j++)
   {

      for(i=1;i<20;i++)
      {
         u[j][i]=(u[j+1][i]+u[j-1][i]+u[j][i+1]+u[j][i-
1])*0.25;
         error=error+fabs(u[j][i]-store[j][i]);
         store[j][i]=u[j][i];
      }
   }
}while(error>.01);
   printf("\nthe error is =%f",error);
   printf("\n the solutions are\n");
   for(j=0;j<41;j++)
   {
      for(i=0;i<21;i++)
      {
         printf("%d\t%d\t%f\n",j+1,i+1,u[j][i]);
      }
      printf("\n");
   }
}
```

Problem 2. Time marching method:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
void main()
{
   double u[40][20],error=0,store[40][20];
   int i,j;

   for(i=0;i<21;i++)
   {
   u[0][i]=100;
   store[0][i]=100;
   }

   for(j=1;j<41;j++)
   {
      for(i=0;i<21;i++)
      {
         u[j][i]=0;
         store[j][i]=0;

      }
   }
```

```
   do
{
   error=0;
   for(j=1;j<40;j++)
   {

      for(i=1;i<20;i++)
      {
         u[j][i]=(u[j+1][i]+u[j-1][i]+u[j][i+1]+u[j][i-
1])*0.25;
        error=error+fabs(u[j][i]-store[j][i]);
        store[j][i]=u[j][i];
      }
   }
}while(error>.01);
   printf("\nthe error is =%f",error);
   printf("\n the solutions are\n");
   for(j=0;j<41;j++)
   {
      for(i=0;i<21;i++)
      {
         printf("%d\t%d\t%f\n",j+1,i+1,u[j][i]);
      }
      printf("\n");
   }
}
```

Problem 3. PSOR Method:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#define pi 3.1428
void main()
{
   double u[40][20],error=0,store[40][20];
   int i,j;
   float a,w;
a=pow((cos(pi/20)+cos(pi/40))/2,2);
w=(2-2*sqrt(1-a))/a;
printf("a=%f\n",a);
printf("w=%f\n",w);

   for(i=0;i<21;i++)
   {
   u[0][i]=100;
   store[0][i]=100;
   }

   for(j=1;j<41;j++)
   {
      for(i=0;i<21;i++)
      {
         u[j][i]=0;
```

```
            store[j][i]=0;

        }
    }
    do
{
    error=0;
    for(j=1;j<40;j++)
    {

       for(i=1;i<20;i++)
       {
          u[j][i]=(1-w)*u[j][i]+w*((u[j+1][i]+u[j-1][i]
+u[j][i+1]+u[j][i-1])/4);
          error=error+fabs(u[j][i]-store[j][i]);
          store[j][i]=u[j][i];
       }
    }
}while(error>.01);
    printf("\nthe error is =%f",error);
    printf("\n the solutions are\n");
    for(j=0;j<41;j++)
    {
        for(i=0;i<21;i++)
        {
           printf("%d\t%d\t%f\n",j+1,i+1,u[j][i]);
        }
        printf("\n");
```

```
    }
}
```

Problem 4. PSOR with Different Over Relaxation Factor:

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>
#define pi 3.1428
void main()
{
FILE *out;
out=fopen("gauss1.dat","w");
fprintf(out,"#It\tRelaxation factor\n");

    double u[40][20],error=0,store[40][20];
    int i,j,count=1;
    float a,w;
for(w=0.8;w<=2.0;w=w+0.1)
{
count=1;
for(i=0;i<21;i++)
    {
    u[0][i]=100;
    store[0][i]=100;
    }

    for(j=1;j<41;j++)
    {
```

```
        for(i=0;i<21;i++)
        {
           u[j][i]=0;
           store[j][i]=0;

        }
    }
    do
{
    error=0;
    for(j=1;j<40;j++)
    {

       for(i=1;i<20;i++)
       {
          u[j][i]=(1-w)*u[j][i]+w*((u[j+1][i]+u[j-1][i]
+u[j][i+1]+u[j][i-1])/4);
          error=error+fabs(u[j][i]-store[j][i]);
          store[j][i]=u[j][i];
           //fprintf(out,"%d\t%f\n",count,w);
       }
    }
    count++;


}while(error>.01);
    printf("%d\t%f\n",count,w);
    fprintf(out,"%d\t%f\n",count,w);
```

```
    }
    printf("\n");
     /*printf("\nthe error is =%f",error);
     printf("\n the solutions are\n");
     for(j=0;j<41;j++)
     {
        for(i=0;i<21;i++)
        {
           //printf("%d\t%d\t%f\n",j+1,i+1,u[j][i]);
        }
        //printf("\n");
     }*/
}
```

Problem 5. Analytical Method:

```
#include<stdio.h>
#include<math.h>
int main()
{

float pi=3.1416,X,Y,sum;
float T[41][21];
int n,i,j;
for(i=0;i<41;i++)
{
for(j=0;j<21;j++)
{
if(i==0)
T[i][j]=100;
else
T[i][j]=0;
}
}

for(i=1;i<40;i++)
{
for(j=1;j<20;j++)
{
X=i*0.05;
Y=j*0.05;
sum=0;
```

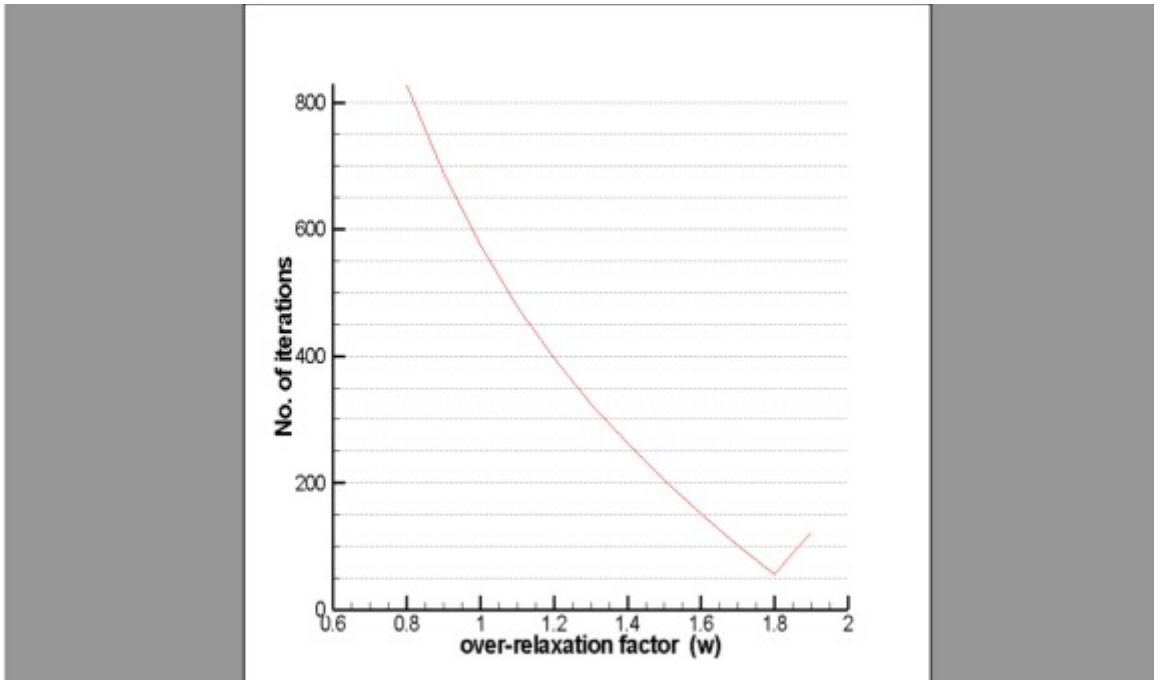
```
for(n=1;n<=110;n++)
{
sum=sum+((1-(pow(-1,n)))/(n*pi))*sinh((n*pi*(2-
X))/1)*sin(n*pi*Y/1)/sinh(n*pi*2/1);
}
T[i][j]=100*2*sum;
}
}
for(i=2;i<41;i++)
{
for(j=2;j<21;j++)
{
printf("T[%d][%d]=%f\t",i,j,T[i][j]);
printf("\n");
}

}
}
```

2.Table of Result

i	J	Gauss Seidal	Time Marching	PSOR	Analytical Method
11	1	0	0	0	0
11	2	4.34	4.34	4.35	4.32
11	3	8.53	8.53	8.54	8.51
11	4	12.44	12.44	12.44	12.41
11	5	15.94	15.94	15.95	15.92
11	6	19.01	19.01	19.01	18.96
11	7	21.55	21.55	21.54	21.52
11	8	23.55	23.55	23.55	23.51
11	9	24.96	24.96	24.97	24.94
11	10	25.82	25.82	25.82	25.79
11	11	26.11	26.11	26.13	26.08
11	12	25.82	25.82	25.83	25.79
11	13	24.96	24.96	24.96	24.94
11	14	23.53	23.53	23.54	23.50
11	15	21.55	21.55	21.56	21.55
11	16	19.03	19.03	19.04	19.01
11	17	15.96	15.96	15.94	15.92
11	18	12.44	12.44	12.45	12.42
11	19	8.53	8.53	8.53	8.51
11	20	4.33	4.33	4.32	4.31

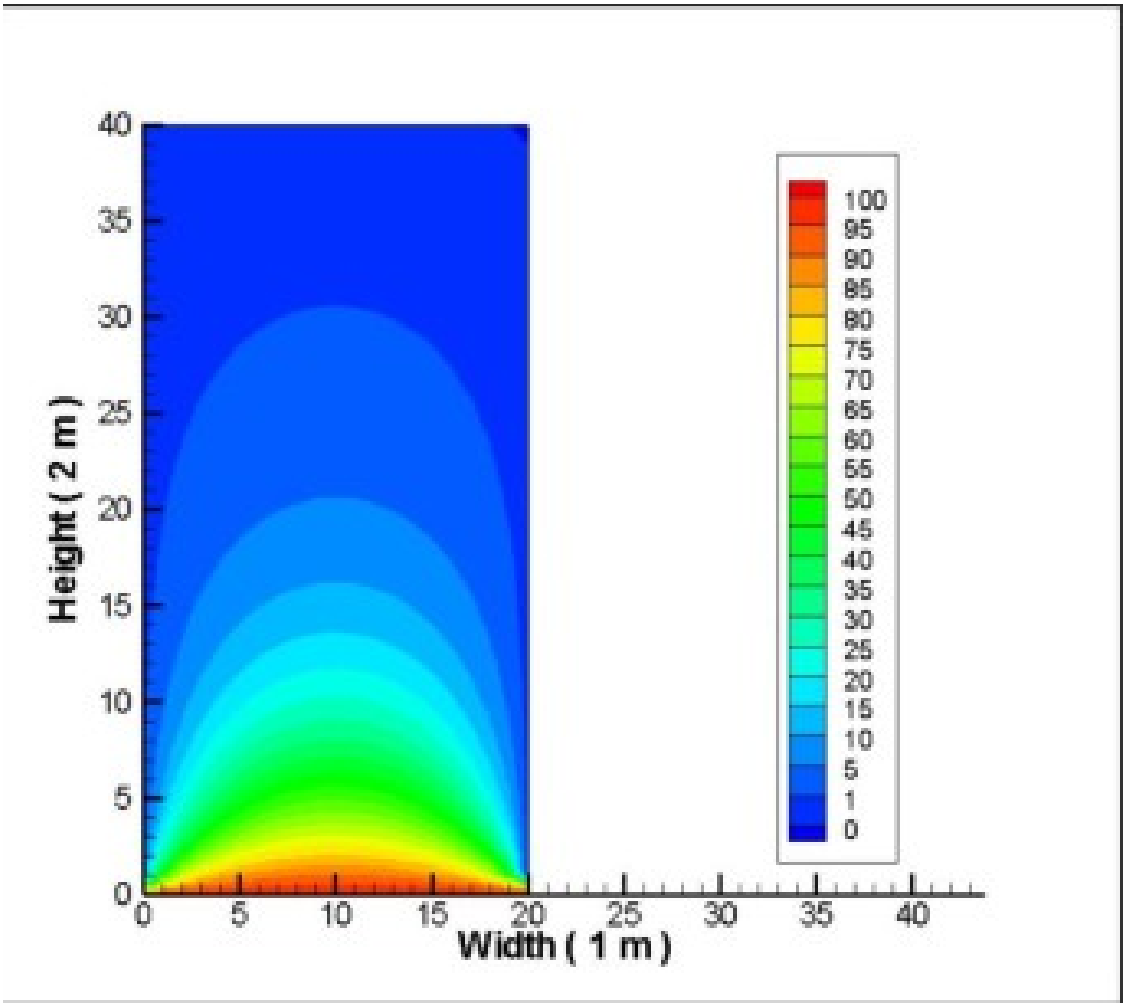
3. Graph between No. Of Iterations V/s Different Over Relaxation factor:




```
829     0.800000
690     0.900000
575     1.000000
479     1.100000
397     1.200000
326     1.300000
262     1.400000
205     1.500000
153     1.600000
103     1.700000
56      1.800000
121     1.900000
```

Conclusion: From the above graph we can conclude that as the Over-relaxation factor Increases upto a certain limit No. Iterations decreases as it reaches its optimum value it again stars increasing.

4.Temperature Contour



Conclusion: The Temperature distribution can be seen on above tecplot.