Shaik Ahmad Ashraf Ali

214103006

ME 543 (CFD)

Home Work Assignment 1

1(a) C – Code for Gauss-Seidel Method :

```
#include<stdio.h>
#include<conio.h>
int main()
{
	int i,j,m=21,n=41,p,q,r,s,count;
	float T1[m][n],T2[m][n],x,y;
	/*file opening*/
	FILE *gs;
	gs = fopen("gauss_seidel.txt","w");
	/*initial condition*/
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			if(j==0)
			{
				T1[i][j] = 100;
			}
			else
			{
				T1[i][j] = 0;
			}
		}
	}
	/*storing in to other variable*/
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			T2[i][j] = T1[i][j];
			printf("%f\t",T1[i][j]);
		}
		printf("\n");
	}
	printf("\n\n");
	/*to count number of iterations*/
	for(count=1;count<10000000;count++)
	{
		y=0;
	/*defining terms in the problem*/
		for(i=1;i<m-1;i++)
```

```
		{
			p = i+1;
			q = i-1;
			for(j=1;j<n-1;j++)
			{
				r = j+1;
				s = j-1;
	/*using the formula of gauss seidel*/
				T1[i][j] = 0.25 * (T1[p][j] + T1[q][j] + T1[i][r] + T1[i][s]);
	/*calculating error*/
				x = T1[i][j] - T2[i][j];
				if(x<0)
				{
					x = 0-x;
				}
				y = y+x;
				T2[i][j] = T1[i][j];
			}
		}
		if(y>0.01)
		{
			continue;
		}
		else
		{
			break;
		}
	}
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("%f\t",T1[i][j]);
		}
		printf("\n");
	}
	printf("\n\n number of iterations is %d \n\n\n",count);
	/*writing in the output text file*/
	fprintf(gs,"i\t\tj\t\tT (gauss Seidel)\n");
	for(j=0;j<n;j++)
	{
		i = 10;
		fprintf(gs,"%d\t\t%d\t\t%f\n",i+1,j+1,T1[i][j]);
	}
	fprintf(gs,"number of iterations is %d",count);
	getch ();
	return 0;
}
```

1(b) C – Code for Time-Marching Method:

```
#include<stdio.h>
#include<conio.h>
int main()
{
	int i,j,m=21,n=41,p,q,r,s,count;
	float T1[m][n],T2[m][n],x,y;
	/*file opening*/
	FILE *tm;
	tm = fopen("time_marching.txt","w");
	/*initial condition*/
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			if(j==0)
			{
				T1[i][j] = 100;
			}
			else
			{
				T1[i][j] = 0;
			}
		}
	}
	/*storing into other variable*/
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			T2[i][j] = T1[i][j];
			printf("%f\t",T1[i][j]);
		}
		printf("\n");
	}
	printf("\n\n");
	/*to count number of iterations*/
	for(count=1;count<10000000;count++)
	{
		y=0;
	/*defining problem terms*/
		for(i=1;i<m-1;i++)
		{
			p = i+1;
			q = i-1;
			for(j=1;j<n-1;j++)
			{
				r = j+1;
				s = j-1;
```

```
        /*using the formula of time marching*/
                                T1[i][j] = 0.25 * (T1[p][j] + T1[q][j] + T1[i][r] + T1[i][s]);
        /*error calculating*/
                                x = T1[i][j] - T2[i][j];
                                if(x<0)
                                {
                                        x = 0-x;
                                }
                                y = y+x;
                                T2[i][j] = T1[i][j];
                        }
                }
                if(y>0.01)
                {
                        continue;
                }
                else
                {
                        break;
                }
        }
        for(i=0;i<m;i++)
        {
                for(j=0;j<n;j++)
                {
                        printf("%f\t",T1[i][j]);
                }
                printf("\n");
        }
        printf("\n\n number of iterations is %d \n\n\n",count);
        /*writing in to the text file*/
        fprintf(tm,"i\t\tj\t\tT (Time Marching)\n");
        for(j=0;j<n;j++)
        {
                i = 10;
                fprintf(tm,"%d\t\t%d\t\t%f\n",i+1,j+1,T1[i][j]);
        }
        fprintf(tm,"number of iterations is %d",count);
        getch ();
        return 0;
}
```

1(c) C – Code for Point Successive over Relaxation at optimum value of relaxation factor :

```
#include<stdio.h>
#include<conio.h>
int main()
{
	int i,j,m=21,n=41,p,q,r,s,count;
	float T1[m][n],T2[m][n],x,y;
	/*file opening*/
	FILE *pcw;
	pcw = fopen("psor_cw.txt","w");
	/*initial condition*/
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			if(j==0)
			{
				T1[i][j] = 100;
			}
			else
			{
				T1[i][j] = 0;
			}
		}
	}
	/*storing into other variable*/
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			T2[i][j] = T1[i][j];
			printf("%f\t",T1[i][j]);
		}
		printf("\n");
	}
	printf("\n\n");
	/*to count number of iterations*/
	for(count=1;count<10000000;count++)
	{
		y=0;
	/*definig terms in formula*/
		for(i=1;i<m-1;i++)
		{
			p = i+1;
			q = i-1;
			for(j=1;j<n-1;j++)
			{
				r = j+1;
```

```
                        s = j-1;
        /*applying formula*/
                        T1[i][j] = 0.25 * (T1[p][j] + T1[q][j] + T1[i][r] + T1[i][s]);
                        T1[i][j] = ((-0.7796207)* T2[i][j]) + ((1.7796207) * T1[i][j]);
        /*calculating error*/
                        x = T1[i][j] - T2[i][j];
                        if(x<0)
                        {
                                x = 0-x;
                        }
                        y = y+x;
                        T2[i][j] = T1[i][j];
                }
        }
        if(y>0.01)
        {
                continue;
        }
        else
        {
                break;
        }
}
for(i=0;i<m;i++)
{
        for(j=0;j<n;j++)
        {
                printf("%f\t",T1[i][j]);
        }
        printf("\n");
}
printf("\n\n number of iterations is %d \n\n\n",count);
/*writing in the output text file*/
fprintf(pcw,"i\t\tj\t\tT (PSOR)\n");
for(j=0;j<n;j++)
{
        i = 10;
        fprintf(pcw,"%d\t\t%d\t\t%f\n",i+1,j+1,T1[i][j]);
}
fprintf(pcw,"number of iterations is %d",count);
getch ();
return 0;
}
```

1(d) C – Code for Point Successive over Relaxation at varying value of relaxation factor :

```
#include<stdio.h>
#include<conio.h>
int main()
{
	int i,j,m=21,n=41,p,q,r,s,count;
	float T1[m][n],T2[m][n],x,y,W;
	/*opening text file*/
	FILE *pvw;
	pvw = fopen("psor_vw.txt","w");
	/*this for loop is for varying over relaxation factor*/
	for(W=0.8;W<=2;W=W+0.1)
	{
	/*initial conditions*/
		for(i=0;i<m;i++)
		{
			for(j=0;j<n;j++)
			{
				if(j==0)
				{
					T1[i][j] = 100;
				}
				else
				{
					T1[i][j] = 0;
				}
			}
		}
	/*storing into other variable*/
		for(i=0;i<m;i++)
		{
			for(j=0;j<n;j++)
			{
				T2[i][j] = T1[i][j];
				printf("%f\t",T1[i][j]);
			}
			printf("\n");
		}
		printf("\n\n");
	/*to calculate number of iterations for each relaxation factor*/
		for(count=1;count<10000000;count++)
		{
			y=0;
	/*defining terms in formula*/
			for(i=1;i<m-1;i++)
			{
				p = i+1;
				q = i-1;
```

```
                        for(j=1;j<n-1;j++)
                        {
                                r = j+1;
                                s = j-1;
        /*applying the formula*/
                                T1[i][j] = 0.25 * (T1[p][j] + T1[q][j] + T1[i][r] +
                                T1[i][s]);
                                T1[i][j] = ((1-W)*T2[i][j]) + (W * (T1[i][j]));
        /*calculating error*/
                                x = T1[i][j] - T2[i][j];
                                if(x<0)
                                {
                                        x = 0-x;
                                }
                                y = y+x;
                                T2[i][j] = T1[i][j];
                        }
                }
                if(y>0.01)
                {
                        continue;
                }
                else
                {
                        break;
                }
        }
        for(i=0;i<m;i++)
        {
                for(j=0;j<n;j++)
                {
                        printf("%f\t",T1[i][j]);
                }
                printf("\n");
        }
        printf("\n\n number of iterations is %d \n\n\n",count);
    /*writing in the output text flie*/
        fprintf(pvw,"W\t\t\ti\t\tj\t\tT\n");
        for(j=0;j<n;j++)
        {
                i = 10;
                fprintf(pvw,"%f\t\t%d\t\t%d\t\t%f\n",W,i+1,j+1,T1[i][j]);
        }
        fprintf(pvw,"number of iterations is %d\n\n",count);
    }
    getch ();
    return 0;
}
```

1(e) C – Code for Analytical Method :

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
	int i,j,n;
	float x,y,pi=3.1428571,sum = 0,T[21][41];
	/*opening text file*/
	FILE *ana;
	ana = fopen("analytic_sol.txt","w");
	/*initial conditions*/
	for(i=0;i<21;i++)
	{
		for(j=0;j<41;j++)
		{
			if(j==0)
			{
				T[i][j] = 100;
			}
			else
			{
				T[i][j] = 0;
			}
		}
	}
	/*defining problem terms*/
	for(i=1;i<20;i++)
	{
		for(j=1;j<40;j++)
		{
```

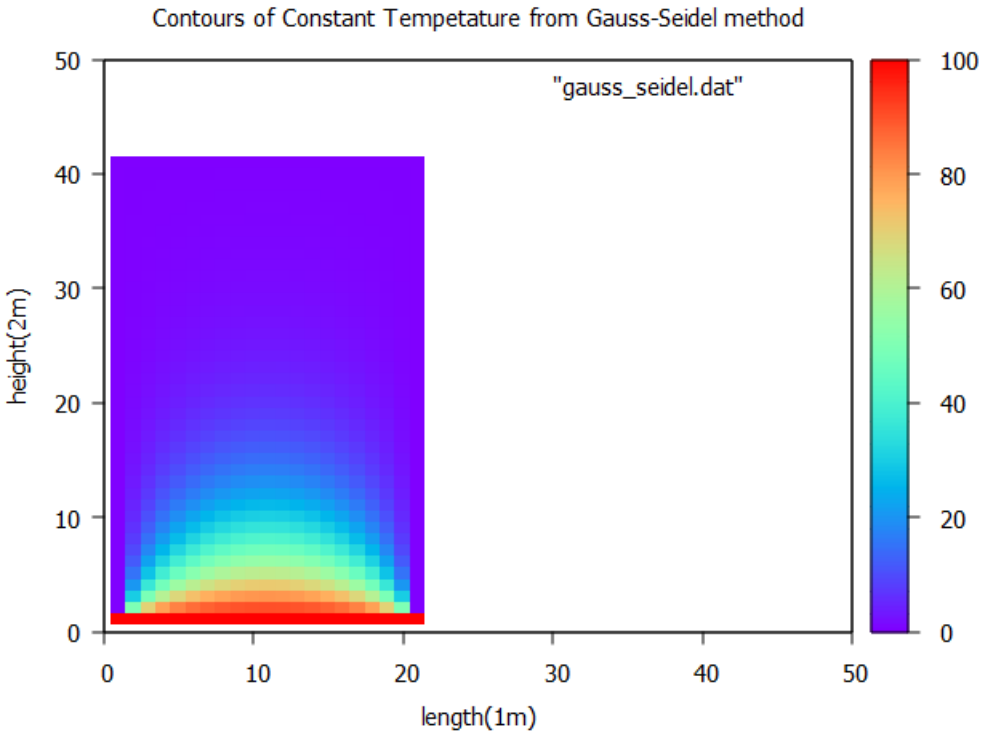
```
                x = 0.05 * i;
                y = 0.05 * j;
                sum = 0;
                for(n=1;n<=110;n++)
                {
/*applying formula*/
                        sum = sum + (((1-pow(-1,n))/(n*pi)) * ((sinh(n*pi*(2-
                        y)))/(sinh(2*pi*n))) * (sin(n*pi*x)));
                }
                T[i][j] = 100 * 2 * sum;
        }
    }
    printf("\n\n");
    /*writing in the output text file*/
    fprintf(ana,"i\t\tj\t\tT (analytical)\n");
    for(j=0;j<41;j++)
    {
        i=10;
        fprintf(ana,"%d\t\t%d\t\t%f\n",i+1,j+1,T[i][j]);
    }
    getch();
    return 0;
}
```

Table of Results

I	j	T (Gauss - Seidel)	T (Time Marching	T (PSOR)	T (Analytical)
11	1	100	100	100	100
11	2	90.000671	90.000671	90.000839	90.000526
11	3	80.25032	80.25032	80.250648	80.280838
11	4	70.966095	70.966095	70.966599	71.012566
11	5	62.311859	62.311859	62.312523	62.359589
11	6	54.390808	54.390808	54.391624	54.428925
11	7	47.249847	47.249847	47.25079	47.272415
11	8	40.890518	40.890518	40.891563	40.895931
11	9	35.281956	35.281956	35.283112	35.271561
11	10	30.37281	30.37281	30.374084	30.34944
11	11	26.100832	26.100832	26.102224	26.067818
11	12	22.399801	22.399801	22.401297	22.360302
11	13	19.204161	19.204161	19.205744	19.160894
11	14	16.451849	16.451849	16.453506	16.40699
11	15	14.085817	14.085817	14.087553	14.040986
11	16	12.054682	12.054682	12.056486	12.011051
11	17	10.31282	10.31282	10.314673	10.271183
11	18	8.820137	8.820137	8.822021	8.78099
11	19	7.541649	7.541649	7.543535	7.50528
11	20	6.446999	6.446999	6.44891	6.413527
11	21	5.509941	5.509941	5.511847	5.479383
11	22	4.707851	4.707851	4.709742	4.680138
11	23	4.021261	4.021261	4.023126	3.996272
11	24	3.433441	3.433441	3.435267	3.411027
11	25	2.93003	2.93003	2.931804	2.910019
11	26	2.498699	2.498699	2.500411	2.480915
11	27	2.128871	2.128871	2.13051	2.113135
11	28	1.811466	1.811466	1.813024	1.797603
11	29	1.538681	1.538681	1.540151	1.526528
11	30	1.30381	1.30381	1.305184	1.29321
11	31	1.101072	1.101072	1.10234	1.091883
11	32	0.925476	0.925476	0.926632	0.917567
11	33	0.772697	0.772697	0.773736	0.765953
11	34	0.638972	0.638972	0.63989	0.63329
11	35	0.521006	0.521006	0.521797	0.516296
11	36	0.415889	0.415889	0.416551	0.412076
11	37	0.321031	0.321031	0.321562	0.318053
11	38	0.234092	0.234092	0.23449	0.231899
11	39	0.152926	0.152926	0.153191	0.151483
11	40	0.075532	0.075532	0.075664	0.074816
11	41	0	0	0	0
Number of Iterations		574	574	52	-

1(f) contours of constant temperature for the rectangular plate from results obtained by Gauss-Seidel method :

Here Gnu plot is Used.



	Temperature at 100°C.
	Temperature at 0°C.