Shaik Ahmad Ashraf Ali

214103006

ME 543 (CFD)

Home Work Assignment 2

C-Code for Lid driven Square Cavity

```
#include <stdio.h>                /*Re = 400; Beta = 1 and velocity of lid =1*/
#include <math.h>                 /*above terms are considered and simplified formula was written
                                  in the code*/
#include <stdlib.h>
int main()
{
	int i, j,c,m=129,n=129;		/*c=count;m and n are number of grid points*/

	double x[m], y[n], es=0, er=1, dx, dy, l=128, xi1[m][n], xi2[m][n], w1[m][n],
		w2[m][n], tol, sumn=0, sumd=0, u[65][j], v[i][65];
	dx= 1/l;		/*x[m]andy[n] to get grid points*/
	dy=dx;		/*es - error sum; er - error; dx,dy-length of one grid*/
	tol= 1/pow(10,5);		/*xi - stream finction; w - vorticity function*/
	printf ("%lf \t %lf \t %lf \t %lf\t",es,er,dx,dy);	/*tol - to limit error; u,v- centre line
								velocities*/
	for(i=1;i<=m;i++)		/*intial guess*/
	{
		for(j=1;j<=n;j++)
		{
			xi1[i][j] = 0;			/*stream function is zero every where*/
			if(i==1 || i==m)		/*voritcity conditions */
			{
				w1[i][j] = 0;
			}
			else
			{
				if(j==n)
				{
					w1[i][j] = -2/dy;
				}
				else
				{
					w1[i][j] = 0;
				}
			}
		}
	}
	for(i=1;i<=m;i++)		/*storing values on xi1 and w1 into xi2 and w2*/
	{
		for(j=1;j<=n;j++)
		{
```

```
                xi2[i][j] = xi1[i][j];
                w2[i][j] = w1[i][j];
        }
}
while (er > tol)        /*for updating values of xi and w using iteration*/
{
        for(c=1;c<=10;c++)              /*iterations of  stream function*/
        {
                for (i = 2; i < m; i++)
                {
                        for (j = 2; j < n; j++)         /*using simplified xi formula*/
                        {
                                xi2[i][j] = 0.25 * (xi2[i+1][j] + xi2[i-1][j] + xi2[i][j+1] +
                                xi2[i][j-1] + (dx * dx * w1[i][j]));
                        }
                }
        }
        for (j = 2; j < n; j++) /*updating boundary condition of vorticity function at
                                top lid and bottom surface*/
        {
                w1[1][j] = (-2 * xi2[2][j]) / (dx * dx);
                w2[1][j] = w1[1][j];
                w1[129][j] = (-2 * xi2[128][j]) / (dx * dx);
                w2[129][j] = w1[129][j];
        }
        for (i = 2; i < m; i++) /*updating boundary condition of vorticity function at
                                left and right walls*/
        {
                w1[i][1] = (-2 * xi2[i][2]) / (dy * dy);
                w2[i][1] = w1[i][1];
                w1[i][129] = ((-2 * xi2[i][128]) - (2 * dy * 1)) /(dy * dy);
                w2[i][129] = w1[i][129];
        }
        for(c=1;c<=2;c++) /*updating values of vorticity function*/
        {
                for (i = 2; i < m; i++)
                {
                        for (j = 2; j < n; j++) /*using simplified w formula*/
                        {
                        w2[i][j] = 0.25 * (w1[i+1][j] + w2[i-1][j] + w1[i][j+1] + w2[i][j-
                        1] - (100 * (w1[i+1][j] - w2[i-1][j]) * (xi1[i][j+1] - xi1[i][j-1]))
                        + (100 * (w1[i][j+1] -w2[i][j-1]) * (xi1[i+1][j] - xi1[i-1][j])));
                        }
                }
        }
        for (i = 1; i <= m; i++) /*evaluation of error*/
        {
                for (j = 1; j <= n; j++)
                {
                        sumn = sumn + fabs((w2[i][j] - w1[i][j]));
```

```
                    sumd = sumd + fabs(w2[i][j]);
                }
            }
            es = es + (sumn / sumd);
            for (i = 1; i <= m; i++) /*updating xi and w values*/
            {
                for (j = 1; j <= n; j++)
                {
                    xi1[i][j] = xi2[i][j];
                    w1[i][j] = w2[i][j];
                }
            }
            er = es;
            es = 0;
            sumn = 0;
            sumd = 0;
        }
        FILE *xy,*uvel,*vvel,*xi,*sf,*U,*V; /*opening text files to plot velocities and stream
                                                         functions */
        xy=fopen("coordinate.txt","w");
        xi=fopen("stream_func.txt","w");
        sf=fopen("stream_function.txt","w");
        uvel=fopen("u_vel_central.txt","w");
        U=fopen("U_velocity.txt","w");
        vvel=fopen("v_vel_central.txt","w");
        V=fopen("V_velocity.txt","w");
        fprintf(xy,"x\ty\n");                               /*headers in text file*/
        fprintf(uvel,"x\ty\tu\n");
        fprintf(vvel,"x\ty\tv\n");
        fprintf(xi,"x\ty\txi");
        x[1]=0;                  /*coordinates of grid*/
        x[129]=1;
        y[1]=0;
        y[129]=1;
        for(i=2;i<m;i++)
        {
            for(j=2;j<n;j++)
            {
                x[i]=x[i-1]+dx;
                y[j]=y[j-1]+dy;
            }
        }
        for(i=1;i<=m;i++)
        {
            for(j=1;j<=n;j++)
            {
                fprintf(xy,"%lf\t%lf\n",x[i],y[j]);
            }
        }
        for(i = 1; i <= m; i++)                             /*stream function*/
```

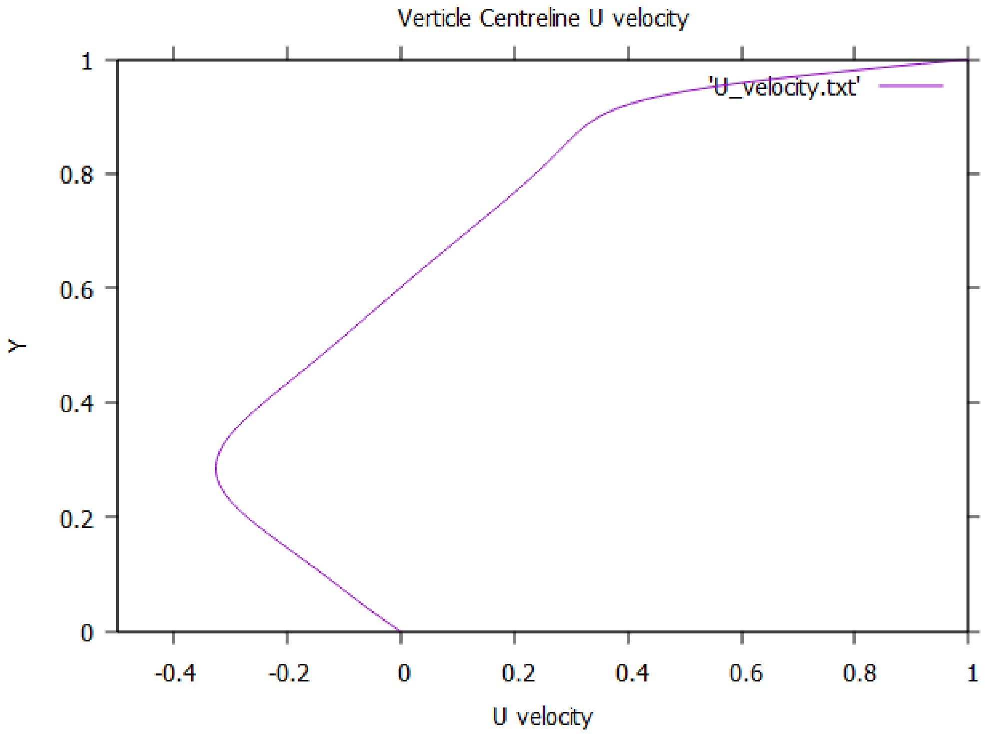
```
	{
		for(j = 1; j < n; j++)
		{
			fprintf(xi,"%lf \t %lf \t %lf \n",x[i],y[j],xi1[i][j]);
			fprintf(sf,"%lf \t %lf \t %lf \n",x[i],y[j],xi1[i][j]);
		}
	}
	for(j=1;j<=n;j++)				/*verticle centre line velocity u*/
	{
		if(j==1)
		{
			u[65][j]=0;
			fprintf(uvel,"%f\t%f\t%f\n",x[65],y[j],u[65][j]);
			fprintf(U,"%f\t%f\n",u[65][j],y[j]);
		}
		else if(j==n)
		{
			u[65][j]=1;
			fprintf(uvel,"%f\t%f\t%f\n",x[65],y[j],u[65][j]);
			fprintf(U,"%f\t%f\n",u[65][j],y[j]);
		}
		else
		{
			u[65][j] = (xi1[65][j+1]-xi1[65][j-1])/(2*dy);
			fprintf(uvel,"%f\t%f\t%f\n",x[65],y[j],u[65][j]);
			fprintf(U,"%f\t%f\n",u[65][j],y[j]);
		}
	}
	for(i=1;i<=m;i++)	/*horizontal centre line velocity v*/
	{
		if(i==1 || i==m)
		{
			v[i][65]=0;
			fprintf(vvel,"%f\t%f\t%f\n",x[i],y[65],v[i][65]);
			fprintf(V,"%f\t%f\n",x[i],v[i][65]);
		}
		else
		{
			v[i][65] = (xi1[i+1][65]-xi1[i-1][65])/(2*dx);
			fprintf(vvel,"%f\t%f\t%f\n",x[i],y[65],v[i][65]);
			fprintf(V,"%f\t%f\n",x[i],v[i][65]);
		}
	}
	return 0;
}
```

Verticle Centre line U velocity:

comparison between obtained values and from Ghia et al.

grid num	u	ghia	grid num	u	ghia	grid num	u	ghia
1	0	0	44	-0.305728		87	0.084771	
2	-0.011955		45	-0.299592		88	0.09433	
3	-0.023501		46	-0.292813		89	0.103907	
4	-0.034704		47	-0.285462		90	0.113496	
5	-0.045623		48	-0.27761		91	0.123088	
6	-0.056312		49	-0.269328		92	0.132673	
7	-0.066825		50	-0.260683		93	0.142241	
8	-0.077209	-0.08186	51	-0.251741		94	0.151781	
9	-0.087511	-0.09266	52	-0.24256		95	0.16128	0.16256
10	-0.097772	-0.10338	53	-0.233196		96	0.170726	
11	-0.108031		54	-0.223698		97	0.180104	
12	-0.118319		55	-0.214109		98	0.189401	
13	-0.128664		56	-0.204466		99	0.1986	
14	-0.139085	-0.14612	57	-0.1948		100	0.207688	
15	-0.149593		58	-0.185138		101	0.216648	
16	-0.160193		59	-0.175498	-0.17119	102	0.225467	
17	-0.170877		60	-0.165897		103	0.234132	
18	-0.18163		61	-0.156346		104	0.242632	
19	-0.192427		62	-0.14685		105	0.25096	
20	-0.203231		63	-0.137413		106	0.259116	
21	-0.213999		64	-0.128035		107	0.267109	
22	-0.224676		65	-0.118715	-0.11477	108	0.274962	
23	-0.235201	-0.24299	66	-0.109447		109	0.282719	
24	-0.245507		67	-0.100227		110	0.290453	0.29093
25	-0.255518		68	-0.091046		111	0.298276	
26	-0.265157		69	-0.081899		112	0.306354	
27	-0.274344		70	-0.072775		113	0.314925	
28	-0.282998		71	-0.063668		114	0.324316	
29	-0.29104		72	-0.054569		115	0.33496	
30	-0.298392		73	-0.04547		116	0.347419	
31	-0.304984		74	-0.036363		117	0.362393	
32	-0.310749		75	-0.027242		118	0.380724	
33	-0.31563		76	-0.0181		119	0.403379	
34	-0.319581		77	-0.008932		120	0.431412	
35	-0.322565		78	0.000267		121	0.465896	
36	-0.324556		79	0.009501		122	0.507811	
37	-0.325542	-0.32726	80	0.018773	0.02135	123	0.557905	0.55892
38	-0.325522		81	0.028085		124	0.61652	0.61756
39	-0.324509		82	0.037437		125	0.683395	0.68439
40	-0.322526		83	0.04683		126	0.7575	0.75837
41	-0.319608		84	0.056262		127	0.836926	
42	-0.315799		85	0.065732		128	0.918898	
43	-0.311152		86	0.075236		129	1	1

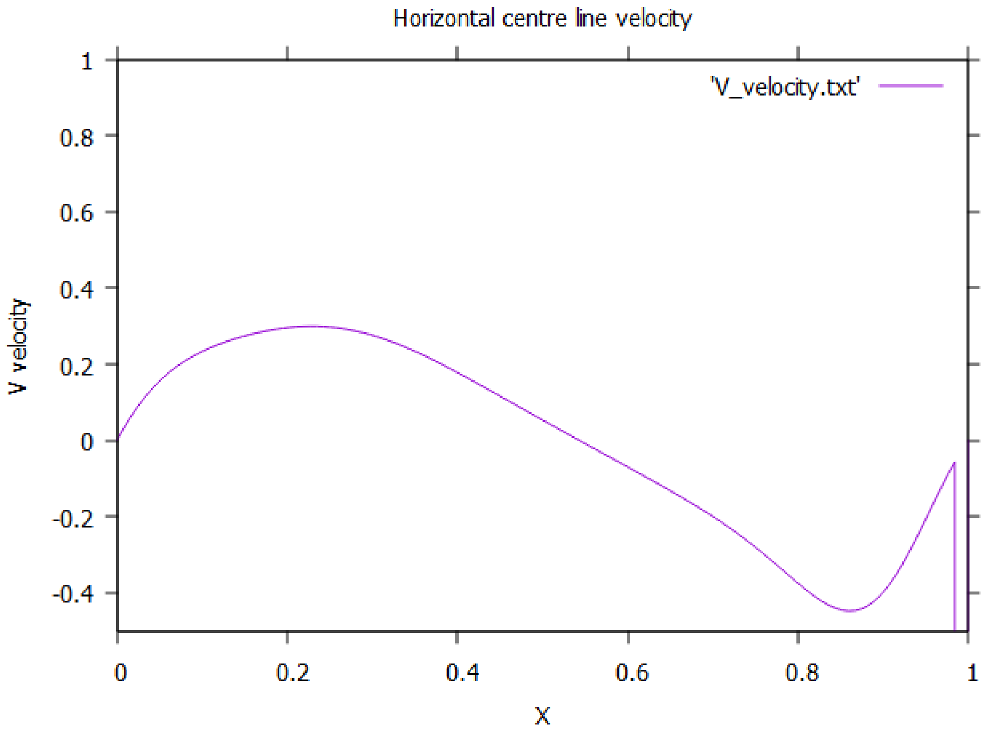
Plot of verticle Centre line U velocity:



Verticle Centre line U velocity:

comparison between obtained values and from Ghia et al.

grid num	v	ghia	grid num	v	ghia	grid num	v	ghia
1	0	0	44	0.247257		87	-0.160433	
2	0.031177		45	0.239869		88	-0.171016	
3	0.059613		46	0.232121		89	-0.181834	
4	0.085425		47	0.224044		90	-0.192921	
5	0.108749		48	0.215671		91	-0.204309	
6	0.129734		49	0.207033		92	-0.216031	
7	0.148545		50	0.198159		93	-0.228116	
8	0.16536		51	0.189082		94	-0.240592	
9	0.180366	0.1836	52	0.179829		95	-0.253477	
10	0.193751	0.19713	53	0.170427		96	-0.266783	
11	0.205701	0.2092	54	0.160902		97	-0.280509	
12	0.216397		55	0.151278		98	-0.294634	
13	0.226004	0.22965	56	0.141576		99	-0.30912	
14	0.234672		57	0.131816		100	-0.323898	
15	0.242533		58	0.122016		101	-0.338871	
16	0.249699		59	0.112189		102	-0.353903	
17	0.256258		60	0.10235		103	-0.368817	
18	0.262283		61	0.092511		104	-0.38339	-0.38598
19	0.267823		62	0.082679		105	-0.397352	
20	0.272915		63	0.072864		106	-0.410384	
21	0.277577	0.28124	64	0.06307		107	-0.422125	
22	0.281819		65	0.053302	0.05186	108	-0.432173	
23	0.285637		66	0.043563		109	-0.4401	
24	0.289022		67	0.033854		110	-0.445467	
25	0.291961		68	0.024177		111	-0.44784	-0.44993
26	0.294436		69	0.01453		112	-0.446819	
27	0.296428		70	0.004912		113	-0.44206	
28	0.297917		71	-0.00468		114	-0.433299	
29	0.298885		72	-0.014249		115	-0.420381	
30	0.299314	0.20203	73	-0.023799		116	-0.40328	
31	0.299192	0.30174	74	-0.033335		117	-0.382108	-0.23827
32	0.298507		75	-0.042863		118	-0.357125	
33	0.297251		76	-0.05239		119	-0.328732	
34	0.29542		77	-0.061924		120	-0.297458	
35	0.293013		78	-0.071474		121	-0.263936	
36	0.290035		79	-0.081051		122	-0.228874	-0.22847
37	0.286493		80	-0.090666		123	-0.193021	-0.19254
38	0.282398		81	-0.100333		124	-0.157131	-0.15663
39	0.277764		82	-0.110067		125	-0.121929	
40	0.272611		83	-0.119886		126	-0.088084	-0.12146
41	0.266958		84	-0.129809		127	-0.056179	
42	0.26083		85	-0.139858		128	-31.526696	
43	0.254254		86	-0.150057		129	0	0



Plot of Stream Lines: