

Incompressible Couette Flow ^{*}

Maciej Matyka[†]
email: maq@panoramix.ift.uni.wroc.pl

Exchange Student
at
University of Linköping

^{*}Thanks for Grzegorz Juraszek (for English language checking).

[†]Student of Computational Physics subsection of Theoretical Physics at University of Wrocław in Poland. Department of Physics and Astronomy.

Contents

1	Introduction	4
2	Fundamental Equations	4
3	Mathematical Formulation of the Couette Problem	4
3.1	Analytical solution	5
3.2	Boundary Conditions for The Analytical Solution	5
4	Numerical Solution	5
4.1	Non-dimensional Form	5
4.2	Finite - Difference Representation	5
4.3	Crank - Nicolson Implicit scheme	6
5	Solving The System of Linear Equations	6
5.1	Gauss Method	6
5.2	Thomas Method	6
6	Results	7
6.1	Different Time Steps	7
6.2	Results for Different Reynolds Numbers	7
6.3	Results for Different Grid Density	7
6.4	Conclusion	8
7	APPENDIX A	9
8	APPENDIX B	12

Abstract

This project work report provides a full solution of simplified Navier Stokes equations for The Incompressible Couette Problem. The well known analytical solution to the problem of incompressible couette is compared with a numerical solution. In that paper, I will provide a full solution with simple C code instead of MatLab or Fortran codes, which are known. For discrete problem formulation, implicit Crank-Nicolson method was used. Finally, the system of equation (tridiagonal) is solved with both Thomas and simple Gauss Method. Results of both methods are compared.

1 Introduction

Main problem is shown in figure (1). There is viscous flow between two parallel plates. Upper plate is moving in x direction with constant velocity ($U = U_e$). Lower one is not moving ($U = 0$). We are looking for a solution to describe velocity vector field in the model (between two plates).

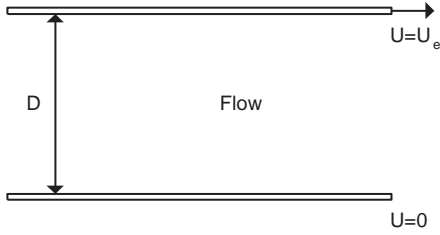


Figure 1: Schematic representation of Couette Flow Problem

2 Fundamental Equations

Most of incompressible fluid mechanics (dynamics) problems are described by simple Navier-Stokes equation for incompressible fluid velocity, which can be written with a form:

$$\frac{\partial \vec{u}}{\partial t} = (-\vec{u}\vec{\nabla})\vec{u} - \vec{\nabla}\varphi + \nu\vec{\nabla}^2\vec{u} + \vec{g}, \quad (1)$$

where φ is defined as the relation of pressure to density:

$$\varphi = \frac{p}{\rho} \quad (2)$$

and ν is a kinematics viscosity of the fluid.

We will also use a continuity equation, which can be written as follows ¹:

$$D = \vec{\nabla} \cdot \vec{v} = 0 \quad (3)$$

Of course, in a case of couette incompressible flow we will use several simplifications of (1).

3 Mathematical Formulation of the Couette Problem

Incompressible couette problem is not needed to solve full Navier-Stokes equations. There is no external force, so first simplification of (1) will be:

¹I assume constant density of the fluid.

$$\frac{\partial \vec{u}}{\partial t} = (-\vec{u}\vec{\nabla})\vec{u} - \vec{\nabla}\varphi + \nu\vec{\nabla}^2\vec{u}, \quad (4)$$

In [1] there can be found easy proof that in couette problem there are no pressure gradients, which means that:

$$\vec{\nabla}\varphi = 0, \quad (5)$$

We will ignore a convection effects so, equation (4) can be written with a form:

$$\frac{\partial \vec{u}}{\partial t} = \nu\vec{\nabla}^2\vec{u} \quad (6)$$

Now we have simple differential equation for velocity vector field. That equation is a vector type and can be simplified even more. Let us write continuity equation (3) in differential form. Let $\vec{u} = (u, v)$, then continuity equation can be expanded as follows²:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (7)$$

We know that there is no u velocity component gradient along x axis (symmetry of the problem), so:

$$\frac{\partial u}{\partial x} = 0 \quad (8)$$

Evaluation of Taylor series³ at points $y = 0$ and $y = D$ gives us a proof that only one possible and physically correct value for y component of velocity \vec{u} is:

$$v = 0 \quad (9)$$

Because of (9) equation (6) can be written as follows:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} \quad (10)$$

Our problem is now simplified to mathematical problem of solving equations like (10). That is now a governing problem of incompressible couette flow analysis.

²Only with assumption of non compressible fluid.

³Those simple expressions can be found in [1], chapter 9.2.

3.1 Analytical solution

Analytical solution for velocity profile of steady flow, without time-changes (steady state) can be found very easily in an equation:

$$v \frac{\partial^2 u}{\partial y^2} = 0 \quad (11)$$

And without any changes of viscosity v it can be written in form:

$$\frac{\partial^2 u}{\partial y^2} = 0 \quad (12)$$

After simple two times integration of equation (12) we have analytical solution function of (12):

$$u = c_1 \cdot y + c_2 \quad (13)$$

where c_1 and c_2 are integration constans.

3.2 Boundary Conditions for The Analytical Solution

Simple boundary conditions are provided in that problem. We know that:

$$u = \begin{cases} 0 & y=0 \\ u_e & y=D \end{cases} \quad (14)$$

Simple applying it to our solution (13) gives a more specified one, where $c_1 = \frac{u_e}{D}$ and $c_2 = 0$:

$$u = \frac{u_e}{D} \cdot y \quad (15)$$

It means, that a relationship between u and y is linear. A Better idea to write that with mathematical expression is:

$$\frac{u}{y} = \frac{u_e}{D} \quad (16)$$

Where $\frac{u_e}{D}$ is a constans for the problem (initial u velocity vs size of the model).

4 Numerical Solution

4.1 Non-dimensional Form

Let us define some new non-dimensional variables⁴:

$$u = \begin{cases} u' & = \frac{u}{u_e} \\ y' & = \frac{y}{D} \\ t' & = \frac{t}{D/u_e} \end{cases} \quad (17)$$

⁴Exacly the same, like in [1].

Now let us place these variables into equation (10) and we have now a non-dimensional equation written as follows:

$$\rho \frac{\partial u/u_e}{(t \cdot u_e)/D} \left(\frac{u_e^2}{D} \right) = v \frac{\partial^2 (u/u_e)}{\partial (y/D)^2} \left(\frac{u_e}{D^2} \right) \quad (18)$$

Now we replace all the variables to nondimensional, like defined in (17):

$$\rho \frac{\partial u'}{\partial t'} \left(\frac{u_e^2}{D} \right) = v \frac{\partial^2 u'}{\partial y'^2} \left(\frac{u_e}{D^2} \right) \quad (19)$$

Now we will remove all ' chars from that equation (only for simplification of notation), and the equation becomes⁵ to:

$$\frac{\partial u}{\partial t} = \left(\frac{v}{D \rho u_e} \right) \frac{\partial^2 u}{\partial y^2} \quad (20)$$

In that equation Reynold's number Re appears, and is defined as:

$$Re = \frac{D \rho u_e}{v} \quad (21)$$

Where Re is Reynold's number that depends on D height of couette model. Finally, the last form of the equation for the couette problem can be written as follows:

$$\frac{\partial u}{\partial t} = \left(\frac{1}{Re} \right) \frac{\partial^2 u}{\partial y^2} \quad (22)$$

We will try to formulate numerical solution of the equation (22).

4.2 Finite - Difference Representation

In our solution of equation (22) we will use Crank-Nicolson technique, so discrete representation of that equation can be written⁶ as:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{2(\Delta y)^2 Re} (u_{j+1}^{n+1} + u_{j+1}^n - 2u_j^{n+1} + u_{j-1}^{n+1} + u_{j-1}^n) \quad (23)$$

Simple grouping of all terms which are placed in time step (n+1) on the left side and rest of them - on right side, gives us an equation which can be written as:

⁵Constans simplification also implemented

⁶That representation is based on central discrete differential operators.

of free (zero) values. In the Gauss method these values join the calculation, what is useless.

Thomas' simplification for tridiagonal matrices is to get only values from non-zero tridiagonal part of matrix. Simply applying a Thomas' equations for our governing matrix equation (28) gives us:

$$d'_i = B - \frac{A^2}{d'_{i-1}} \quad (32)$$

$$u'_i = u_i - \frac{u'_i A}{d'_{i-1}} \quad (33)$$

We know that exact value of u_M is defined as follows:

$$u_M = \frac{u'_M}{B} \quad (34)$$

Now solution of the system of equations will be rather easy. We will use recursion like that:

$$u_{i-1} = \frac{u'_{i-1} - A \cdot u_i}{B} \quad (35)$$

That easy recursion provides us a solution for the linear system of equations.

6 Results

Main results are provided as plots of the function:

$$\frac{y}{D} = f\left(\frac{u}{u_e}\right) \quad (36)$$

In figure (2) there is drawn an analytical solution to the problem of couette flow. That is linear function, and we expect that after a several time steps of numerical procedure we will have the same configuration of velocity field.

6.1 Different Time Steps

In figure (3) there are results of velocity u calculation for several different time steps. Analytical solution is also drawn there.

As we can see in the figure (3) - the solution is going to be same as analytical one. Beginning state (known from boundary conditions) is changing and relaxing.

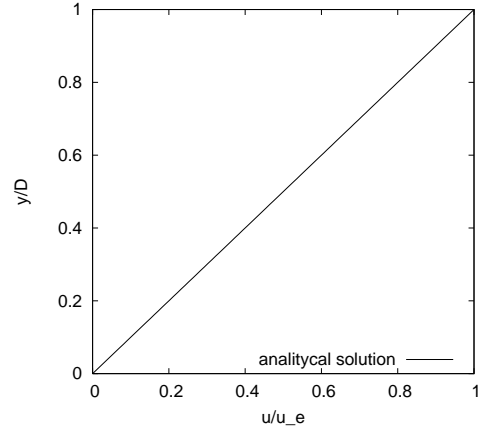


Figure 2: Analytical exact solution.

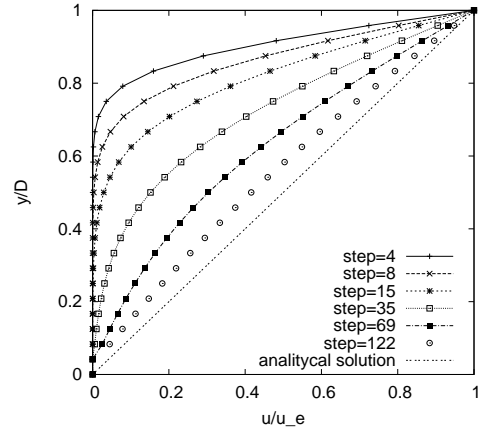


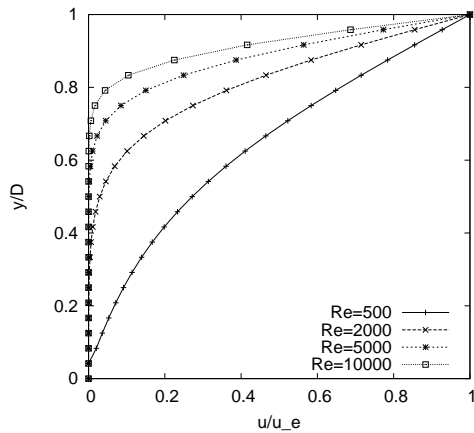
Figure 3: Results for different time steps of numerical calculations.

6.2 Results for Different Reynolds Numbers

In the figure (4) there is plot of numerical calculations for different Reynold's numbers. For example Reynold's number depends on i.e. viscosity of the fluid, size of couette model. As it is shown on the plot there is strong relationship between the speed of the velocity field changes and Reynold's number. In a couple of words: when Reynolds number increases - frequency of changes also increases.

6.3 Results for Different Grid Density

In figure (5) there is an example of calculations of velocity field for different grid density (N number).



scheme. Well known linear relationships were observed.

Figure 4: Calculation for same time ($t = 2500$) and different Reynold's numbers.

We see that there is also strong correlation between grid density, and speed of changes on the grid. Also, very interesting case $N = 10$ shows, that for low density of the grid changes are very fast, and not accurate.

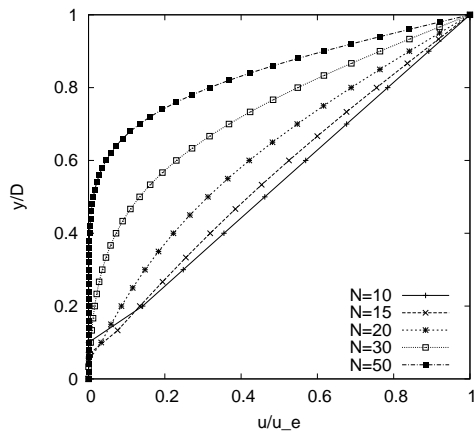


Figure 5: Calculation for same time ($t = 2500$), same Reynold's numbers ($Re=5000$) and different grid density (N number).

6.4 Conclusion

Solving of Incompressible Couette Problem can be good way to check numerical method, because of existing analytical solution. In that report there were presented two methods of solving system of equations: Gauss and Thomas' method. System of equations was taken from Crank-Nicolson Implicit

7 APPENDIX A

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define N (40)
#define NN (N+1)

void Zamien(double *a, double *b) {
    double c;
    c=*a;  *a=*b;  *b=c;
}

void WypiszMacierz(double A[NN][NN], int n) {
    int i,j;

    for(j=1;j<n;j++)
    {
        for(i=1;i<n;i++)          // show matrix
        {
            printf("%2.4f  ",A[i][j]);
        }
        printf("\n");
    }
}

void Gauss(double A[NN][NN], double *b, double *x, int n) {
    int i,j,k;
    double m;

    // Gauss Elimination

    for(i=0;i<n;i++)
    {
        // Step #1: Change governing element

        m=fabs(A[i][i]);
        k=i;

        for(j=i+1;j<n;j++)
        if(fabs(A[i][j])>m)
        {
            m=fabs(A[i][j]);
            k=j;
        }
        if(k!=i)
        for(j=0;j<n;j++)
        {
            Zamien(&A[j][i],&A[j][k]);
            Zamien(&b[i+1],&b[k+1]);
        }
    }
}
```

```

    }

    // Step #2: make it triangle
    for(j=i+1;j<n;j++)
    {
        m = A[i][j]/A[i][i];

        for(k=i;k<n;k++)
            A[k][j] = A[k][j] - m*A[k][i];

        b[j+1] = b[j+1] - m*b[i+1];
    }
}

// Step#3: Solve now
for(i=n-1;i>=1;i--)
{
    for(j=i+1;j<n;j++)
        b[i+1] = b[i+1]-A[j][i]*x[j+1];
    x[i+1] = b[i+1]/A[i][i];
}
}

int main(void)
{
    double U[N*2+2]={0},A[N*2+2]={0},B[N*2+2]={0},C[N*2+2]={0},D[N*2+2]={0},Y[N*2+2]={0};
    // initialization
    double OneOverN = 1.0/(double)N;
    double Re=5000; // Reynolds number
    double EE=1.0; // dt parameter
    double t=0;
    double dt=EE*Re*OneOverN*2; // delta time
    double AA=-0.5*EE;
    double BB=1.0+EE;
    int KKEND=1122;
    int KKM0D=1;
    int KK; // for a loop
    int i,j,k; // for loops too
    int M; // temporary needed variable
    double GMatrix[NN][NN]={0}; // for Gauss Elimination
    double test;
    Y[1]=0; // init

    // apply boundary conditions for Couette Problem

    U[1]=0.0;
    U[NN]=1.0;

    // initial conditions (zero as values of vertical velocity inside of the couette model)

    for(j=2;j<=N;j++)
        U[j]=0.0;

```

```

A[1]=B[1]=C[1]=D[1]=1.0;

for(KK=1;KK<=KKEND;KK++)
{
    for(j=2;j<=N;j++)
    {
        Y[j]=Y[j-1]+OneOverN;
        A[j]=AA;
        if(j==N)
            A[j]=0.0;
        D[j]=BB;
        B[j]=AA;
        if(j==2)
            B[j]=0.0;
        C[j]=(1.0-EE)*U[j]+0.5*EE*(U[j+1]+U[j-1]);
        if(j==N)
            C[j]=C[j]-AA*U[NN];
    }

// Gauss
// C[]      -   free
// A[]B[]D[] -   for matrix calculation
// U[]      -   X

// calculate matrix for Gauss Elimination

GMatrix[0][0]=D[1];
GMatrix[1][0]=A[1];

for(i=1;i<N-1;i++)
{
    GMatrix[i-1][i]=B[i+1];    // GMatrix[1][2]=B[2]
    GMatrix[i][i]=D[i+1];     // GMatrix[2][2]=D[2]
    GMatrix[i+1][i]=A[i+1];   // GMatrix[3][2]=A[2]
}

GMatrix[N-2][N-1]=B[N];
GMatrix[N-1][N-1]=D[N];

Gauss(GMatrix,C,U,N);        // Gauss solving function

Y[1]=0.0;
Y[NN]=Y[N]+OneOverN;

t=t+dt;                      // time increment
test=KK % KKMOD;

if(test < 0.01)              // print the results
{
    printf("KK,TIME\n");      // info 1
    printf("%d,%f\n",KK,t);
}

```

```

        printf(",J,Y[J],U[j]\n"); // info 2
    for(j=1;j<=NN;j++)
        printf("%d , %f, %f\n",j,U[j],Y[j]);
        printf("\n \n \n \n"); // for nice view of several datas
    }
}

return (1);
}

```

8 APPENDIX B

```

#include <stdlib.h>
#include <stdio.h>

#define N (50)
#define NN (N+1)

int main(void)
{
    double U[N*2+2],A[N*2+2],B[N*2+2],C[N*2+2],D[N*2+2],Y[N*2+2];
// initialization

    double OneOverN = 1.0/(double)N;
    double Re=7000; // Reynolds number
    double EE=1.0; // dt parameter
    double t=0;
    double dt=EE*Re*OneOverN*2; // delta time
    double AA=-0.5*EE;
    double BB=1.0+EE;
    int KKEND=122;
    int KKM0D=1;
    int KK; // for a loop
    int j,k; // for loops too
    int M; // temporary needed variable

    double test;
    Y[1]=0; // init

// apply boundary conditions for Couette Problem
    U[1]=0.0;
    U[NN]=1.0;

// initial conditions (zero as values of vertical velocity inside of the couette model)
    for(j=2;j<=N;j++)
        U[j]=0.0;
    A[1]=B[1]=C[1]=D[1]=1.0;

    printf("dt=%f, Re=%f, N=%d \n",dt,Re, N);
}

```

```

for(KK=1;KK<=KKEND;KK++)
{
    for(j=2;j<=N;j++)
    {
        Y[j]=Y[j-1]+OneOverN;
        A[j]=AA;
        if(j==N)
            A[j]=0.0;

        D[j]=BB;
        B[j]=AA;

        if(j==2)
            B[j]=0.0;
        C[j]=(1.0-EE)*U[j]+0.5*EE*(U[j+1]+U[j-1]);

        if(j==N)
            C[j]=C[j]-AA*U[NN];
    }

    // upper bidiagonal form
    for(j=3;j<=N;j++)
    {
        D[j]=D[j]-B[j]*A[j-1]/D[j-1];
        C[j]=C[j]-C[j-1]*B[j]/D[j-1];
    }

    // calculation of U[j]
    for(k=2;k<N;k++)
    {
        M=N-(k-2);
        U[M]=(C[M]-A[M]*U[M+1])/D[M]; // Appendix A
    }

    Y[1]=0.0;
    Y[NN]=Y[N]+OneOverN;
    t=t+dt; // time increment
    test=KK % KKMOD;

    if(test < 0.01) // print the results
    {
        printf("KK,TIME\n"); // info 1
        printf("%d,%f\n",KK,t);

        printf(",J,Y[J],U[j]\n"); // info 2
        for(j=1;j<=NN;j++)
            printf("%d , %f, %f\n",j,U[j],Y[j]);
        printf("\n \n \n \n"); // for nice view of several datas
    }
}
return (1);
}

```

References

- [1] John D. Anderson, Jr. '*Computational Fluid Dynamics: The Basics with Applications*', McGraw-Hill Inc, 1995.
- [2] David Potter '*Metody obliczeniowe fizyki*', PWN 1982.
- [3] Ryszard Grybos, '*Podstawy mechaniki płynów*' (Tom 1 i 2), PWN 1998.