Universitat Politècnica de Catalunya
BarcelonaTech
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Aerospace Technology Engineering

EXERCISE 5

Smith-Hutton case

APMC++ to Thermal Engineering Problems

Abril Bertran Garrigos

May 2025

Index

1 Case of study **1**

2 Algorithm **2**

2.1 Physical Properties 2

2.2 Numerical Properties 2

2.3 Velocity Field 3

2.4 Calculation of ϕ 3

2.5 Complete code 5

3 Results **9**

3.1 Comparison of ρ/Γ ratios 9

3.2 Plot of the ϕ value at the outlet 9

3.3 Color map plot 10

1 Case of study

The main objective of this exercise is to analyze the Smith-Hutton case where there is a solenoidal flow in the domain shown below:

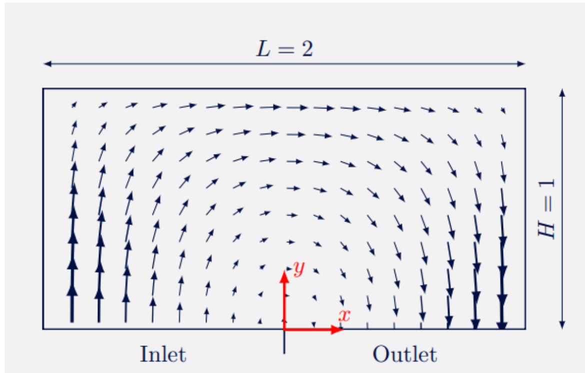

Figure 1: Smith-Hutton case

And the velocity field is defined as:

$$u(x, y) = 2y(1 - x^2)$$

$$v(x, y) = -2x(1 - y^2)$$

Also, there are different boundary conditions imposed:

$\phi = 1 + \tanh[(2x + 1)\alpha]$	For the inlet $\quad (x \in [-1, 0],\ y = 0)$
$\dfrac{\partial \phi}{\partial y} = 0$	For the outlet $\quad (x \in [0, 1],\ y = 0)$
$\phi = 1 - \tanh[\alpha]$	For the rest of the walls

Figure 2: Boundary Conditions

The considerations that are taken into account to solve this problem are:

- $\alpha = 10$
- The convective scheme used is CDS

The results that are asked are the following:

- Solve for the following ρ/Γ ratios: $\rho/\Gamma = 10$, $\rho/\Gamma = 1000$, $\rho/\Gamma = 1\ 000\ 000$
- Plot the value of ϕ at the outlet as a function of x.
- Plot a color map of the three different cases.

2 Algorithm

2.1 Physical Properties

The first thing to do is to enter the physical data of the problem. The boundary conditions are not defined here, but will be used later.

```
L=2;
H=1;
gamma=1;
rho=10;
alpha=10;
```

2.2 Numerical Properties

Now it is entered the numerical data necessary to carry out the problem. Therefore, a proper dt must be chosen and should be small enough for the solution to converge. Therefore, a dt of 10^{-3} has been applied. Even so, in the case of the ratio of $\rho/\Gamma = 10^6$ the differential is needed to be minimized even more.

On the other hand, the mesh of the problem has been created with the origin of coordinates at the lower edge and centered in the middle.

```
%Time discretization
dt=0.001;

w=1;                      % width of the VC
Nx=100;    Ny=50;         % Node discretization
dx=L/Nx;   dy=H/Ny;       % Differentials
ds=dx*w;   dv=dx*dy*w;

% Construction of x and y coordinate node matrices
x=zeros(Ny,Nx);
y=zeros(Ny,Nx);

for i=1:Nx
    x(:,i)=-1+0.5*dx+(i-1)*dx;
end

for i=1:Ny
    y(i,:)=0.5*dy+(i-1)*dy;
```

```
end
y=flipud(y);
```

2.3 Velocity Field

For this problem it is necessary to calculate the velocity camp at each node:

```
u=2*y.*(1-x.^2);
v=-2*x.*(1-y.^2);
```

2.4 Calculation of ϕ

For this part of the problem it has been decided to make a function that can be called from the main script and ables to modify the ρ/Γ ratios. In this function there is an iterative loop that obtains phi explicitly in the following way:

$$\phi_p^{n+1} = \phi_P^n + \frac{\Delta t}{\rho \Delta V}(-conv^n + diff^n)$$

```
function [phi_new]=getPhi(rho,gamma,dt,u,v,dx,ds,dv,Nx,Ny,alpha,x,epsilon)
    % Phi inicialization

    phi_new= zeros(Ny,Nx);              % Inicialization of the matrix to obtain phi^n+1
    phi_p= zeros (Ny,Nx);                % Initial values of phi at t=0

    % Iterative loop ensuring convergencefor t^n
    err=1;
    while err>epsilon
        % Calculation to obtain phi^n+1
        for j=Ny:-1:1
            for i=1:Nx
                [conv,diff]=getCoeff(i,j,Nx,Ny,gamma,rho,alpha,x,dx,ds,phi_p,u,v);
                phi_new(j,i)=phi_p(j,i)+dt/(rho*dv)*(-conv+diff);
            end
        end
        err=max(max(abs(phi_new-phi_p))); % Error calculation
        phi_p=phi_new;                   % Saves the previous results for next iteration
    end

```

```
end
```

The convective and diffusive terms vary depending on the node and boundary conditions. For this reason, it has been decided to create another function to calculate them for each node. Generally, the formulas used are:

$$\begin{aligned}
\text{conv} &= \rho\left(u_e\phi_e\Delta S_e - u_w\phi_w\Delta S_w + v_n\phi_n\Delta S_n - v_s\phi_s\Delta S_s\right)\\
\text{diff} &= \Gamma_e\frac{\phi_E-\phi_P}{d_{PE}}\Delta S_e + \Gamma_w\frac{\phi_W-\phi_P}{d_{PW}}\Delta S_w + \Gamma_n\frac{\phi_N-\phi_P}{d_{PN}}\Delta S_n + \Gamma_s\frac{\phi_S-\phi_P}{d_{PS}}\Delta S_s
\end{aligned}$$

However, some exceptions need to be considered for the nodes at the edges as it is shown in the code:

```
function [conv,diff]=getCoeff(i,j,Nx,Ny,gamma,rho,alpha,x,dx,ds,phi_p,u,v)

    % Velocities
    if i>1,  uw=0.5*(u(j,i)+u(j,i-1)); else, uw=u(j,i); end
    if i<Nx, ue=0.5*(u(j,i)+u(j,i+1)); else, ue=u(j,i); end
    if j>1,  vn=0.5*(v(j,i)+v(j-1,i)); else, vn=v(j,i); end
    if j<Ny, vs=0.5*(v(j,i)+v(j+1,i)); else, vs=v(j,i); end


    % Calculation of phis by CDS
    if i>1,  phiW=0.5*(phi_p(j,i)+phi_p(j,i-1)); else, phiW=1-tanh(alpha); end
    if i<Nx, phiE=0.5*(phi_p(j,i)+phi_p(j,i+1)); else, phiE=1-tanh(alpha); end
    if j>1,  phiN=0.5*(phi_p(j,i)+phi_p(j-1,i)); else, phiN=1-tanh(alpha); end
    if j<Ny, phiS=0.5*(phi_p(j,i)+phi_p(j+1,i)); end


    % Bottom wall
    if j==Ny
        if x(j,i) <=0  %inlet
            phiS= 1+tanh((2*x(j,i)+1)*alpha);
            conv=rho*ds*(ue*phiE-uw*phiW+vn*phiN-vs*phiS);
            diff=gamma*ds/dx*(phiE+phiW+phiN+phiS-4*phi_p(j,i));

        else %outlet
            phiS=phi_p(j,i);
            conv=rho*ds*(ue*phiE-uw*phiW+vn*phiN-vs*phiS);
            diff=gamma*ds/dx*(phiE+phiW+phiN-3*phi_p(j,i));
```

```
        end
        return  %finalizes the execution of the function, useful for this specific case
    end



    % General case
    conv=rho*ds*(ue*phiE-uw*phiW+vn*phiN-vs*phiS);
    diff=gamma*ds/dx*(phiE+phiW+phiN+phiS-4*phi_p(j,i));


end
```

2.5 Complete code

Finally, the complete code is shown along with the scripts for displaying the desired results.

```
format long
clc;
clear;

%% PHYSICAL PROPERTIES
L=2;
H=1;
gamma=1;
rho=10;
alpha=10;

%% NUMERICAL PROPERTIES

%Time discretization
dt=0.001;

w=1;                        % width of the VC
Nx=100;    Ny=50;           % Node discretization
dx=L/Nx;   dy=H/Ny;         % Differentials
ds=dx*w;   dv=dx*dy*w;

% Construction of x and y coordinate node matrices
```

```
x=zeros(Ny,Nx);
y=zeros(Ny,Nx);

for i=1:Nx
    x(:,i)=-1+0.5*dx+(i-1)*dx;
end

for i=1:Ny
    y(i,:)=0.5*dy+(i-1)*dy;
end
y=flipud(y);


%% Velocity field
u=2*y.*(1-x.^2);
v=-2*x.*(1-y.^2);

% Plot of the velocity
%quiver(x,y,u,v,2);
%axis equal;


%% Phi inicialization

phi_new= zeros(Ny,Nx);              % Inicialization of the matrix to obtain phi^n+1
phi_p= zeros (Ny,Nx);               % Initial values of phi at t=0

%% Phi calculation

epsilon=10^-6;
err=1;
iter=0;

[phi_1]=getPhi(10,gamma,dt,u,v,dx,ds,dv,Nx,Ny,alpha,x,epsilon);
[phi_2]=getPhi(1000,gamma,dt,u,v,dx,ds,dv,Nx,Ny,alpha,x,epsilon);
[phi_3]=getPhi(10^5,gamma,10^-5,u,v,dx,ds,dv,Nx,Ny,alpha,x,epsilon);
```

```matlab

%% Data visualization
T = table(x(end,:).', phi_1(end,:).',phi_2(end,:).',phi_3(end,:).', 'VariableNames', {
    'x', 'rho10','rho1000','rho10e6'});
T=T(T.x>0,:);

disp(T);

%% plot the color map

figure
subplot(1,3,1);
surf(x,y,phi_1);
axis equal tight
shading interp;       % suaviza el gr fico
view(2);
colorbar;

subplot(1,3,2);
surf(x,y,phi_2);
axis equal tight
shading interp;
view(2);
colorbar;

subplot(1,3,3);
surf(x,y,phi_3);
axis equal tight
shading interp;
view(2);
colorbar;

%% Plot the outlet profile
figure
subplot(1,3,1);
plot(x(end,:), phi_1(end,:),'LineWidth', 1.5);
xlabel('x');
ylabel('\phi');
```

```
ylim ([0 2.1]);
title('\rho / \Gamma=10');

subplot(1,3,2);
plot(x(end,:), phi_2(end,:),'LineWidth', 1.5);
xlabel('x');
ylabel('\phi');
ylim ([0 2.1]);
title('\rho / \Gamma=1000');

subplot(1,3,3);
plot(x(end,:), phi_3(end,:),'LineWidth', 1.5);
xlabel('x');
ylabel('\phi');
ylim ([0 2.1]);
title('\rho / \Gamma=1000000');
```

3 Results

3.1 Comparison of ρ/Γ ratios

Below are some of the results of the outlet for different ratios ofρ/Γ. If it is wanted to display the results for all x's it can be done by running the main script of the code.

Position x	$\rho/\Gamma = 10$	$\rho/\Gamma = 10^3$	$\rho/\Gamma = 10^6$
0.01	1.8103	1.9996	2.0123
0.11	1.4930	2.0000	1.9988
0.21	1.2665	1.9999	2.0003
0.31	1.0623	1.9964	1.9988
0.41	0.8692	1.8643	1.9467
0.51	0.6861	0.8430	0.8037
0.61	0.5157	0.0630	0.0244
0.71	0.3611	0.0014	3.0898e-04
0.81	0.2238	2.514 e-05	-2.587e-04
0.91	0.1018	4.032e-06	5.416e-04
0.99	0.0112	2.801e-07	2.603e-04

Table 1: Numerical results of the outlet

These results maintain a similarity with those given by the teacher and can therefore be considered valid and correct.

3.2 Plot of the ϕ value at the outlet

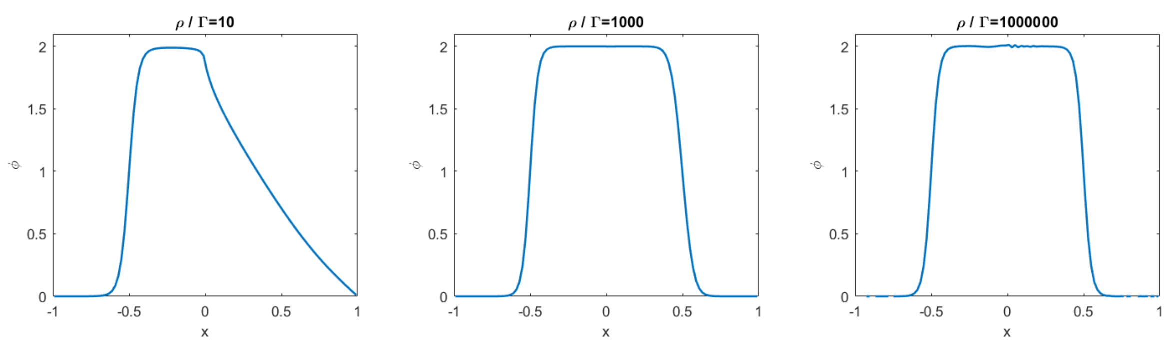

Figure 3: Plots of ϕ at the outlet for different ϕ/Γ ratios

Here, it can be observed graphically the behavior of ϕ at the outlet as a function of x. These graphs have the expected behavior in comparison with the results given by the professor. Even so, it is important to notice that for the last case, there have been difficulties to reach convergence. In fact, the instability can be observed with the small oscillations it has when it reaches its maximum values. This may be due to the CDS scheme used, which is less stable than the UDS. Also, for this last case, the dt has been reduced significantly to reach the solution, as well as the computational time and cost have been higher.

3.3 Color map plot

The color maps obtained for the steady-state Smith Hutton case are shown below.

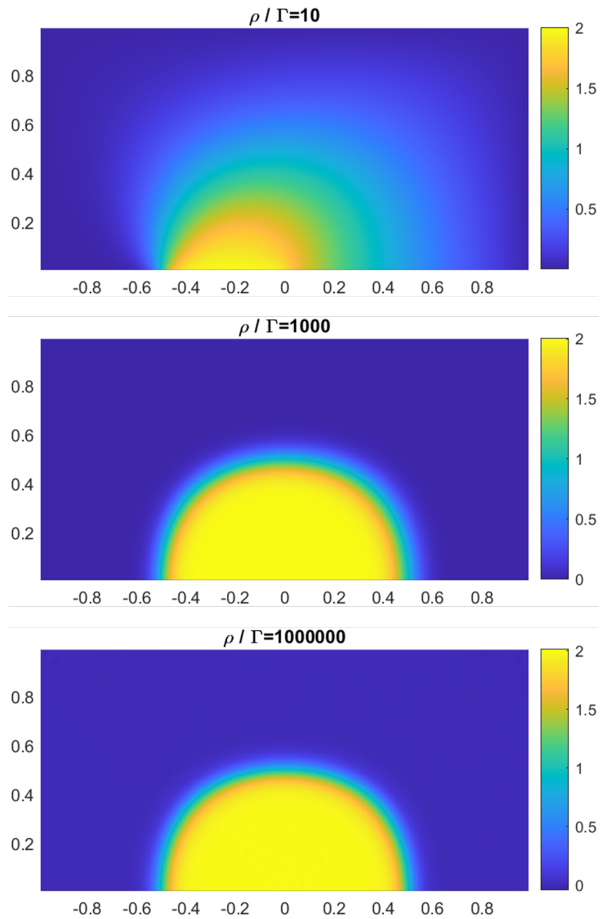

Figure 4: Color maps of ϕ at steady state of the Smith-Hutton case for different ϕ/Γ ratios

Visually, the color maps are coincident to those shown in the document, so the implementation of the code is validated. In addition, it is also possible to see the profile changes depending on the value ofρ/Γ . For lower values, there is a gradual transition from the inlet to outlet. For higher values some more abrupt changes appear, there are more marked regions.