Prandtl-Meyer Expansion Wave

Maciej Matyka
Computational Physics Section of Theoretical Physics at University of Wroclaw in Poland
Department of Physics and Astronomy

Exchange Student at University of Linköping in Sweden

maq@panoramix.ift.uni.wroc.pl
http://panoramix.ift.uni.wroc.pl/∼maq

May 6, 2003

Abstract

This project work report provides a full solution of the governing Euler equations in strong conservation form for the Prandtl-Meyer expansion wave problem. The well known analytical solution to the problem of expansion wave is compared with a numerical solution. In that paper, I will provide a full solution with simple C code. In solution an explicit MacCormack's predictor-corrector finite difference method was used. Some new ideas for visualization of vector and scalar fields will be shown. All figures in that report will be shown in a physical grid instead of computational grid, where all calculations were made.

1 Introduction

The main problem is shown in figure (1). We assume inviscid fluid flow (air). Initial conditions are described by several types of physics variables for vertical line at $x = 0$.

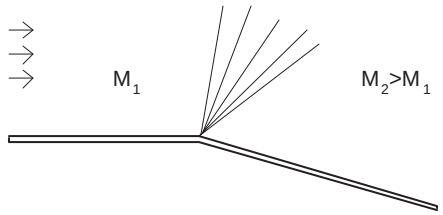


Figure 1: Geometric of The Prandtl-Meyer Problem

My goal is to solve strong conservation form of Euler equations in a computational grid. Numerical values for all physics variables on the whole grid should be calculated from initial data line. I will use MacCormack's predictor-corrector method which is an explicit one with flux variables of physics data representation.

2 Analytical Solution

First of all, I will show an analytical solution of that problem[1] which is well known.

Anderson in [1] provides a full analytical solution for PM problem. Let us define two angles: μ_1 and μ_2. The μ_1 angle is an angle made by the leading edge of the expansion with respect to the upstream flow direction.

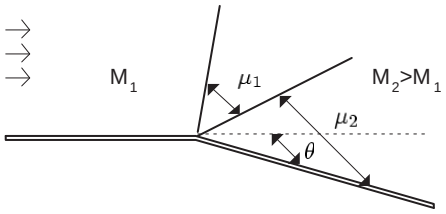


Figure 2: μ_1, μ_2 and θ angles in physics space.

The μ_2 angle is made by the trailing edge with respect to the downstream flow direction. Both angles are defined

[1] In that and next theory sections, I will show some results which are taken directly from [1] - my CFD course book.

as follows:

$$\mu_1 = arcsin\left(\frac{1}{M_1}\right), \mu_2 = arcsin\left(\frac{1}{M_2}\right) \tag{1}$$

where M_1 and M_2 are respectively upstream and downstream Mach numbers. The configuration of the angles is shown in the figure (2).

2.1 Prandtl-Meyer function

Analytical solution of the inviscid flow over expansion corner can be described with simple relation:

$$f_2 = f_1 + \theta \tag{2}$$

where f is the PM (Prandtl-Meyer) function. θ is the flow deflection angle. PM function f can be written as follows:

$$f = \sqrt{\frac{\gamma+1}{\gamma-1}}\arctan\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)} - \arctan\sqrt{M^2-1} \tag{3}$$

where $M_{1,2}$ are Mach numbers, and γ is taken directly from air properties (from state equation of the ideal gas).

2.2 Analytical procedure

For the analytical solution we have to make some steps to obtain the results of the PM expansion wave. For given initial condition - Mach number M_1 we calculate f_1 directly from the PM function (3). After that, for given θ we calculate f_2 from (2). Then implicity trial and error procedure is applied to the equation 3 to obtain the Mach number M_2 in the region behind the expansion corner. For the known M_2 number we can very easily obtain other physics variables like pressure, temperature or density (from the isentropic flow relations), for pressure:

$$p_2 = p_1\left(\frac{1+(\frac{\gamma-1}{2})M_1^2}{1+(\frac{\gamma-1}{2})M_2^2}\right)^{\frac{\gamma}{\gamma-1}} \tag{4}$$

and temperature:

$$T_2 = T_1\frac{1+(\frac{\gamma-1}{2})M_1^2}{1+(\frac{\gamma-1}{2})M_2^2} \tag{5}$$

The simple thermodynamics law (equation of state) gives us an expression for the density of the fluid:

$$\rho_2 = \frac{p_2}{RT_2} \tag{6}$$

3 Numerical Solution

In that section I will show governing ideas for numerical solution of the PM expansion wave problem.

3.1 Mathematical formulation

3.1.1 Governing equation

First of all, let us define a mathematical equation of the flow which should be numerically solved to obtain good results. I will use Euler equations which can be written in a generic form:

$$\frac{\partial\vec{F}}{\partial x} = J - \frac{\partial\vec{G}}{\partial y} \tag{7}$$

where $\vec{F}$ and $\vec{G}$ are column vectors with four elements, as it is shown above:

$$\vec{F} = \begin{cases} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u\left(e + \frac{V^2}{2}\right) + pu \end{cases} \tag{8}$$

$$\vec{G} = \begin{cases} \rho u \\ \rho uv \\ \rho v^2 + p \\ \rho v\left(e + \frac{V^2}{2}\right) + pv \end{cases} \tag{9}$$

3.1.2 Assumption #1: no body forces

First simplification of the equation (7) will be assumption of hence adiabatic flow with no body forces (source term in equation (7) will be removed). With that assumption equation (7) becomes:

$$\frac{\partial\vec{F}}{\partial x} = -\frac{\partial\vec{G}}{\partial y} \tag{10}$$

3.1.3 Assumption #2: perfect gas

Calculation of e term from $\vec{F}$ and $\vec{G}$ flux variables could be complicated. It is convenient to eliminate that variable by making assumption (look into [1], [2]) that we consider perfect gas which provides us a nice expression for e:

$$e = c_v T = \frac{1}{1-\gamma}\frac{p}{\rho} \tag{11}$$

With that assumption we will get the best expressions for fourth terms in $\vec{F}$ and $\vec{G}$ vectors, for the perfect gas we obtain that:

$$\vec{F} = \begin{cases} \rho u \\ \rho u^2 + p \\ \rho uv \\ \frac{\gamma}{\gamma-1} pu + \rho u \frac{u^2+v^2}{2} \end{cases} \tag{12}$$

$$\vec{G} = \begin{cases} \rho u \\ \rho uv \\ \rho v^2 + p \\ \frac{\gamma}{\gamma-1} pv + \rho v \frac{u^2+v^2}{2} \end{cases} \tag{13}$$

Equation (10) with $\vec{F}$ and $\vec{G}$ in forms (12) and (13) will be solved.

3.2 Computational Grid

3.2.1 Boundary-fitted coordinate system

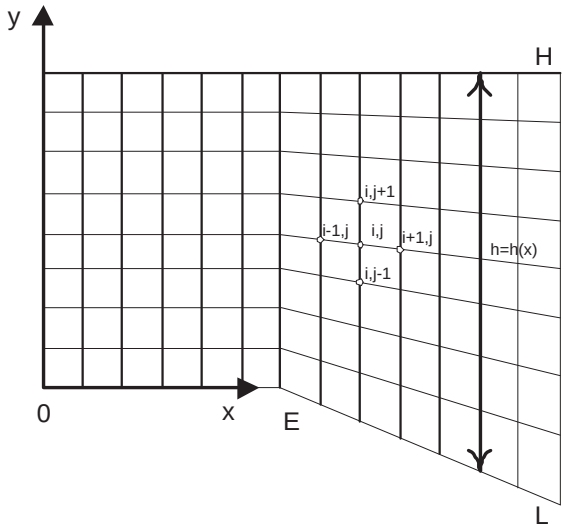


Figure 3: Physical grid.

Geometry of the problem, which is shown in the figure (1), gives us possibility to use boundary-fitted coordinate system. It means that once at the beginning of computations we will transform the physical plane to a rectangular computational grid of points.

In the figure (3) the physical plane was sketched. Grid points for five several locations on the grid were marked with outlined circles. We will not numerically solve Euler equations on that grid but, instead of that, simple transformation of the grid will be made. In the figure (4) computational (rectangular) grid was sketched. That grid can be easy implemented in a computer program with a simple two dimensional table.

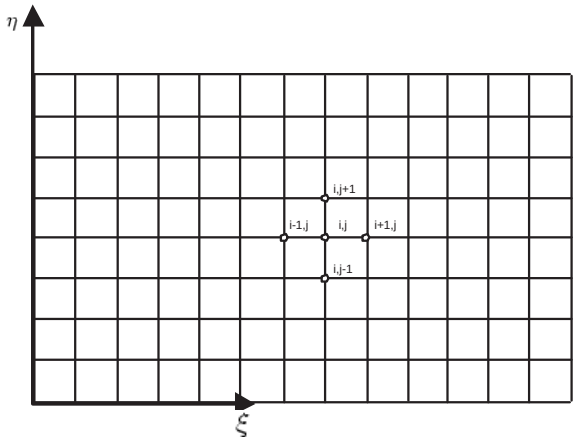


Figure 4: Computational grid.

3.2.2 Grid transformation

Now let us provide some simple math for grid transformations. Let $h = h(x)$ be the local height in the physics plane, and $y_s = y_s(x)$ be local y position of the surface (in the physics plane of course). Then the transformation can be defined as follows:

$$\begin{cases} \xi = x \\ \eta = \frac{y - y_s(x)}{h(x)} \end{cases} \tag{14}$$

After grid transformation, in computational plane ξ is placed between $(0, L)$ and η is placed between $(0, 1)$.

I will skip some parts of obvious transformation mathematics expressions which can be found in [1]. But let us provide complete derivative transformation between computational and physics planes:

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi} + \left(\frac{\partial \mu}{\partial x}\right) \frac{\partial}{\partial \mu} \tag{15}$$

$$\frac{\partial}{\partial y} = \frac{1}{h} \frac{\partial}{\partial \mu} \tag{16}$$

3.3 Finite-difference formulation - MacCormack's scheme

I will solve the equation (10) with explicite predictor-corrector MacCormack's technique. I will use a downstream marching scheme. In that section I will try to make a general overview of that method - you can find a more detailed description in here [1]. The first predictor step will be described then I will show how to apply the corrector step. I will try to concentrate on these parts of the algorithm which were hardest for me to implement in

my solution (or which are not included directly in provided example solution, which were not obvious for me).

3.3.1 Predictor Step (PS) # 1 - Derivatives

Predictor step in the Anderson's algorithm is rather easy to implement. First of all, we have to calculate derivatives[2] of flux F_k variable[3]:

$$\left(\frac{\partial F_k}{\partial \xi}\right)_{i,j} = \left(\frac{\partial \mu}{\partial x}\right)_{i,j} \frac{F_{k,i,j} - F_{k,i,j+1}}{\Delta\mu} + \frac{1}{h}\frac{G_{k,i,j} - G_{k,i,j+1}}{\Delta\mu} \tag{17}$$

Where (i, j) are positions in computational grid, and k numbers $\vec{F}$ component.

One question that occured when I tried to implement these calculations was how to compute the term:

$$\left(\frac{\partial \mu}{\partial x}\right)_{i,j} \tag{18}$$

It is not obvious where to calculate it. On the page 401 in [1] Anderson suggests to use values calculated in current (i, j) position in the computational grid, I did. For that calculation I used the relation:

$$\left(\frac{\partial \mu}{\partial x}\right)_{i,j} = \begin{cases} 0 & , x < E \\ (1-\mu)\cdot h^{-1}(x)\tan\theta & , x \geq E \end{cases}$$

where E is obviously position of the expansion corner. For the $h(x)$ simple relation was used:

$$h(x) = \begin{cases} H & , x < E \\ H + (x-E)\cdot\tan\theta & , x \geq E \end{cases}$$

where H is the height before the expansion corner (for x¡H) in the physics plane.

3.3.2 PS # 2 - Artificial Viscosity

Anderson provides now equations[4] for predicted values of $\overline{F}_k$, but after that he adds artificial viscosity (AF) to them. Let us provide equations for AF first:

$$\begin{aligned}(SF)_{k,i,j} = & \frac{C_y|p_{i,j+1} - 2p_{i,j} + p_{i,j-1}|}{p_{i,j+1} + 2p_{i,j} + p_{i,j-1}} \\ & \cdot(F_{k,i,j+1} - 2F_{k,i,j} + F_{k,i,j-1})\end{aligned} \tag{19}$$

Where C_y is a constant ($C_y = 0.6$ in [1]). At this stage of algorithm it is important to notice that we do not calculate artificial viscosity terms for boundary values. It will be discussed more in the section about Abbet's boundary conditions.

[2] Equations 8.36a-c in [1].

[3] Variable k varies from 1 to 4

[4] Eq. 8.37a-d

3.3.3 PS # 3 - Predicted $\vec{F}$ values

Now, when we calculated artificial viscosity terms for all four components of $\vec{F}$ vector, we will be able to find predicted values of all four component of predicted $\vec{F}$:

$$\overline{F}_{k,i+1,j} = F_{k,i+1,j} + \left(\frac{\partial F_k}{\partial \xi}\right)_{i,j}\Delta\xi + (SF)_{k,i,j} \tag{20}$$

There are some questions about $\Delta\xi$ step size in that equation. Anderson provides an equation for that:

$$\Delta\xi = C \cdot \frac{\Delta y}{|\tan\theta \pm \mu|_{max}} \tag{21}$$

That equation is not so obvious, I suppose that a better idea to form it is to write the equation with a small change:

$$\Delta\xi = C \cdot \frac{\Delta y}{\max_{(i,1\leq j\leq ny)}|\tan\theta \pm \mu_{i,j}|} \tag{22}$$

Now it is rather obvious which value in the equation (21) should be taken for maximum computation[5]

In the equation (22) C is a Courant number which defines the relation between Δx and $\Delta\xi$. That number should vary from 0 to 1, and can be defined as follows:

$$\Delta\xi = C\Delta x$$

3.3.4 PS # 4 - Predicted $\vec{G}$ values

Now, a predicted ρ value can be easy obtained from equations provided by Anderson (eq. 8.38). Value of $\overline{\rho}$ will be needed for predicted values of the $\overline{G}$ calculation.

Other calculations ($\overline{G}_k$) of the predictor step were rather easy to implement and I will not repeat Anderson solution any more.

3.3.5 Corrector Step

The corrector step is very similar to the predictor one but instead of using $\vec{F}$ flux variables, we use the $\overline{F}$ predicted values of it. When we have computed respectively derivatives, we take average of derivatives computed for the predictor and corrector step. Simple equation defines results of our work - the final values of $\vec{F}_{i+1,j}$ the flux variable:

$$F_{k,i+1,j} = F_{k,i,j} + \left(\frac{\partial F_k}{\partial \xi}\right)_{av}\Delta\xi + \overline{SF}_{k,i+1,j} \tag{23}$$

[5] Of course in equation (21) only η varies in y direction, but simple $|\tan\theta \pm \mu|_{max}$ was not so obvious when I tried to implement that function.

where average derivative is combined out from the predictor and corrector calculations[6]:

$$\left(\frac{\partial F_1}{\partial \xi}\right)_{av} = 0.5 \cdot \left[\left(\frac{\partial F_k}{\partial \xi}\right)_{i,j} + \left(\frac{\partial \overline{F}_k}{\partial \xi}\right)_{i+1,j}\right] \quad (24)$$

3.3.6 Final calculations

After the predictor and corrector steps we have only one thing to do. We have to decode physical variables from flux $\vec{F}$ vectors. It's rather easy to do, and Anderson provides simple relations for that in [1].

3.4 Boundary Conditions

When we have done all the calculations for $j = 2 \ldots NY-1$ we have to apply boundary conditions to points $j = 1$ and $j = NY$. Anderson suggests to use Abbett's boundary conditions. At the beginning of the boundary conditions implementation we have to calculate trial values[7] of u_t and v_t velocities using one side differences (instead of the differences used in the predictor and corrector step which were shown before). We also have to remember that no artificial viscosity should be calculated at the boundary points.

First of all, we calculate predicted and corrected values with the known algorithm, then we obtain trial values of u_t and v_t velocities which I will mark as $\vec{\mathbf{V}} = (u_t, v_t)$. $\vec{\mathbf{V}}_\mathbf{t}$ vector is a velocity near the boundary, so now we have to do something to deal with the assumption that velocity near the boundary is tangent to wall. It means that our trial values of u_t and v_t velocity components are not correct.

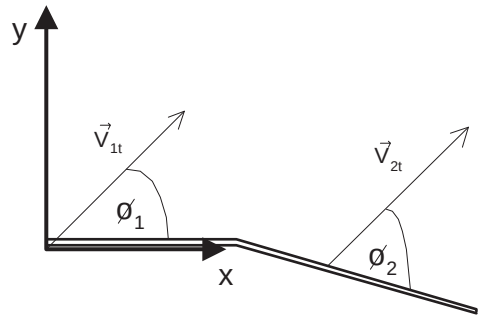


Figure 5: Non correct velocities near the boundaries.

[6]Please denote different i positions of both derivatives

[7]T subscript means that value is needed only temporarily for calculations at the boundaries and the actual corrected value should be recalculated.

In the figure (5) two velocity vectors near boundary are sketched. These velocities should be corrected, so an obvious way for that is to rotate them around z axis. Let us consider first velocity vector from figure (5). First, we calculate an angle ϕ_1:

$$\phi_1 = \arctan \frac{u_{1t}}{v_{1t}} \quad (25)$$

where u_1t and v_1t are components of V_1t vector. Now let us calculate the Mach number for that velocity vector:

$$(M_1)_t = \sqrt{\frac{u_{1t}^2 + v_{1t}^2}{|\gamma R T_t|}} \quad (26)$$

where γ and R are obtained from air properties. T_t is a trial value of temperature obtained with standard calculations. Now we are using the Prandtl-Meyer function (3) to obtain the f_t value. Then we have to add a correction angle ϕ_1 which was calculated before, that will give us an exactly correct value f of the Prandtl-Meyer function:

$$f = f_t + \phi_1 \quad (27)$$

For second velocity we have to make yet another correction and, obviously, add θ angle to the PM function, it means that for points behind expansion corner we will use the equation:

$$f = f_t + \theta - \phi_2 \quad (28)$$

where:

$$\phi_2 = \arctan \frac{u_{2t}}{v_{2t}}$$

Now, with correct new values of the PM function we need to go back with the calculations to obtain correct the Mach number and other physics variables. We can use the PM function (3) again to do that, but it will not be so easy. We have to find the Mach number M for which the Prandtl-Meyer function will be equal to f value obtained with (27) or (28). I used there a simple bisection method to find the zero value of the nonlinear function. I do not know, maybe Anderson thought about that but a simple "trial and error" procedure, which he suggested, was very slow. Some speed could be gained there by an implementation of the Newton method. It is very important to note there that we should calculate the Mach number with very high accuracy. Low accuracy can result in big errors to the calculation near boundaries.

After that procedure, simple equations are used for obtaining pressure, temperature and other physics variables at the boundary (see [1], page 394).

4 Results

4.1 Velocity u For Different x position

First result which was obtained with my PM solver is presented on figures (6), (7) and (8).

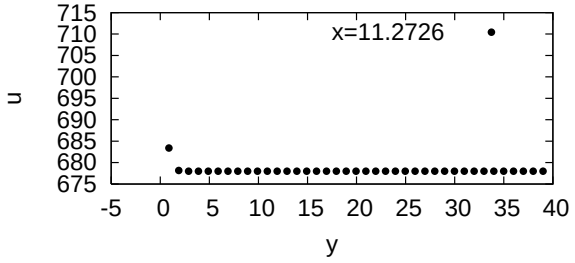


Figure 6: u velocity component for $x = 11.2726$

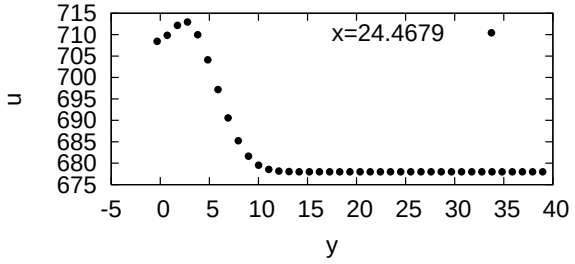


Figure 7: u velocity component for $x = 24.4679$

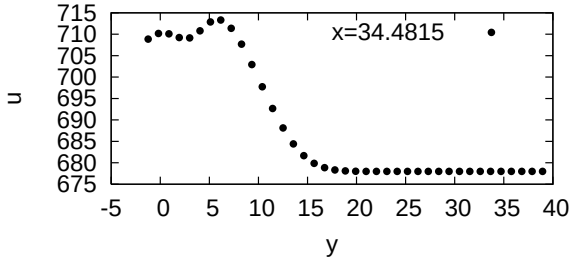


Figure 8: u velocity component for $x = 34.4815$

In the figures (6), (7) and (8) u vertical velocities for different x locations are shown. Spatial resolution of the grid for these computations is $NX = 65$ and $NY = 41$. Initial conditions (at $x = 0$) for these sketches are as follows:

$$\begin{cases} u_0 = 678.1 & [\frac{m}{s}] \\ v_0 = 0 & [\frac{m}{s}] \\ t_0 = 286 & [K] \\ \rho_0 = 1.23 & [\frac{kg}{m^3}] \\ p_0 = 101 \cdot 10^3 & [\frac{N}{m^3}] \\ M_0 = 2 \end{cases} \tag{29}$$

And the θ angle $= 5.352^o$, Courant number $C = 0.5$. These results can be compared with the results presented in the Anderson's book [1]. The same behavior of u velocity field is observed. As we see on the figures, u velocity behind the expansion corner increases to a constant value of $u_{max} \approx 7.11[\frac{m}{s}]$. The same deviation (like in [1]) of u velocity is observed for the first boundary point (lower value of u). In the tables (**??**), (**??**) and (**??**) there were shown numerical values of physical variables which correspond to sketches in figures (6), (7) and (8).

5 Tables

$y[m]$	$u[\frac{m}{s}]$	$v[\frac{m}{s}]$	$p[\frac{N}{m^3}]$	$\rho[\frac{kg}{m^3}]$	$T[K]$	M
-0.225228	709.479	-66.4659	73785.2	1.00353	257.083	2.20172
0.780403	693.261	-33.051	86521.7	1.09385	276.567	2.08566
1.78603	682.251	-8.01057	97234.7	1.19637	284.177	2.0227
2.79166	678.526	-0.92451	100555	1.2261	286.754	2.00247
3.79729	678.016	-0.0274931	100987	1.22989	287.101	1.99975
4.80293	678	-4.67728e-006	101000	1.23	287.111	1.99967
5.80856	678	-6.43423e-015	101000	1.23	287.111	1.99967
6.81419	678	3.12784e-013	101000	1.23	287.111	1.99967
7.81982	678	-1.02743e-013	101000	1.23	287.111	1.99967
8.82545	678	1.21919e-013	101000	1.23	287.111	1.99967
9.83108	678	4.44073e-014	101000	1.23	287.111	1.99967
10.8367	678	6.71649e-017	101000	1.23	287.111	1.99967
11.8423	678	3.22534e-019	101000	1.23	287.111	1.99967
12.848	678	0	101000	1.23	287.111	1.99967
13.8536	678	0	101000	1.23	287.111	1.99967
14.8592	678	0	101000	1.23	287.111	1.99967
15.8649	678	0	101000	1.23	287.111	1.99967
16.8705	678	0	101000	1.23	287.111	1.99967
17.8761	678	0	101000	1.23	287.111	1.99967
18.8818	678	0	101000	1.23	287.111	1.99967
19.8874	678	0	101000	1.23	287.111	1.99967
20.893	678	0	101000	1.23	287.111	1.99967
21.8986	678	0	101000	1.23	287.111	1.99967
22.9043	678	0	101000	1.23	287.111	1.99967
23.9099	678	0	101000	1.23	287.111	1.99967
24.9155	678	0	101000	1.23	287.111	1.99967
25.9212	678	0	101000	1.23	287.111	1.99967
26.9268	678	0	101000	1.23	287.111	1.99967
27.9324	678	0	101000	1.23	287.111	1.99967
28.9381	678	0	101000	1.23	287.111	1.99967
29.9437	678	0	101000	1.23	287.111	1.99967
30.9493	678	0	101000	1.23	287.111	1.99967
31.955	678	0	101000	1.23	287.111	1.99967
32.9606	678	0	101000	1.23	287.111	1.99967
33.9662	678	0	101000	1.23	287.111	1.99967
34.9718	678	0	101000	1.23	287.111	1.99967
35.9775	678	0	101000	1.23	287.111	1.99967
36.9831	678	0	101000	1.23	287.111	1.99967
37.9887	678	0	101000	1.23	287.111	1.99967
38.9944	678	0	101000	1.23	287.111	1.99967
40	678	0	101000	1.23	287.111	1.99967

Table 1: Values of computed physics variables at $x = 12.4042$

$y[m]$	$u[\frac{m}{s}]$	$v[\frac{m}{s}]$	$p[\frac{N}{m^3}]$	$\rho[\frac{kg}{m^3}]$	$T[K]$	M
-0.975183	715.183	-67.0003	73936.9	1.00834	256.383	2.24264
0.049197	709.521	-67.8093	73395.2	0.973807	263.529	2.19421
1.07358	712.352	-72.9148	71743.6	0.962705	260.57	2.21691
2.09796	710.797	-68.8981	73131.4	0.976015	261.988	2.2049
3.12234	704.496	-53.7007	78445.9	1.02585	267.374	2.15938
4.14672	696.272	-35.456	85450.3	1.09066	273.941	2.10507
5.17109	688.777	-20.0317	91860.3	1.14894	279.553	2.0596
6.19547	683.332	-9.59586	96500.7	1.19043	283.439	2.02861
7.21985	680.16	-3.8069	99186.5	1.21414	285.639	2.01123
8.24423	678.701	-1.22154	100414	1.22489	286.636	2.0034
9.26861	678.18	-0.311522	100850	1.2287	286.99	2.00062
10.293	678.036	-0.0624565	100970	1.22974	287.087	1.99986
11.3174	678.006	-0.00975375	100995	1.22996	287.107	1.9997
12.3418	678.001	-0.0011705	100999	1.23	287.111	1.99967
13.3661	678	-0.00010548	101000	1.23	287.111	1.99967
14.3905	678	-6.86782e-006	101000	1.23	287.111	1.99967
15.4149	678	-3.0247e-007	101000	1.23	287.111	1.99967
16.4393	678	-7.95546e-009	101000	1.23	287.111	1.99967
17.4636	678	-9.23146e-011	101000	1.23	287.111	1.99967
18.488	678	-2.27649e-015	101000	1.23	287.111	1.99967
19.5124	678	8.50131e-029	101000	1.23	287.111	1.99967
20.5368	678	4.36294e-031	101000	1.23	287.111	1.99967
21.5612	678	1.53049e-033	101000	1.23	287.111	1.99967
22.5855	678	3.54494e-036	101000	1.23	287.111	1.99967
23.6099	678	5.21088e-039	101000	1.23	287.111	1.99967
24.6343	678	4.75405e-042	101000	1.23	287.111	1.99967
25.6587	678	2.49146e-045	101000	1.23	287.111	1.99967
26.6831	678	0	101000	1.23	287.111	1.99967
27.7074	678	0	101000	1.23	287.111	1.99967
28.7318	678	0	101000	1.23	287.111	1.99967
29.7562	678	0	101000	1.23	287.111	1.99967
30.7806	678	0	101000	1.23	287.111	1.99967
31.805	678	0	101000	1.23	287.111	1.99967
32.8293	678	0	101000	1.23	287.111	1.99967
33.8537	678	0	101000	1.23	287.111	1.99967
34.8781	678	0	101000	1.23	287.111	1.99967
35.9025	678	0	101000	1.23	287.111	1.99967
36.9269	678	0	101000	1.23	287.111	1.99967
37.9512	678	0	101000	1.23	287.111	1.99967
38.9756	678	0	101000	1.23	287.111	1.99967
40	678	0	101000	1.23	287.111	1.99967

Table 2: Values of computed physics variables at $x = 20.4094$

$y[m]$	$u[\frac{m}{s}]$	$v[\frac{m}{s}]$	$p[\frac{N}{m^3}]$	$\rho[\frac{kg}{m^3}]$	$T[K]$	M
-2.07087	715.978	-67.0748	73920.7	1.00835	256.324	2.24486
-1.01909	709.001	-66.7697	73786.4	0.977268	263.996	2.19038
0.0326779	710.033	-66.8724	73784.4	0.982333	262.627	2.19927
1.08445	709.405	-64.9821	74451.4	0.988911	263.239	2.19425
2.13622	709.101	-64.0915	74755	0.991917	263.511	2.19194
3.18799	710.526	-67.6551	73525.3	0.980233	262.266	2.20253
4.23976	712.707	-73.2129	71656.8	0.962446	260.325	2.21915
5.29154	713.317	-74.7648	71140.3	0.957529	259.775	2.22387
6.34331	711.349	-69.761	72811.8	0.973425	261.537	2.20875
7.39508	707.347	-59.9339	76216.4	1.00559	265.01	2.17925
8.44685	702.212	-47.983	80579.4	1.04634	269.269	2.14358
9.49862	696.702	-35.934	85247.4	1.08933	273.625	2.10765
10.5504	691.423	-25.062	89707.9	1.12985	277.616	2.0752
11.6022	686.848	-16.1209	93566.8	1.16447	280.95	2.04842
12.6539	683.283	-9.44336	96568.3	1.19112	283.474	2.02833
13.7057	680.82	-4.97205	98638.4	1.20935	285.185	2.01481
14.7575	679.331	-2.32663	99886.9	1.22029	286.206	2.00677
15.8093	678.552	-0.960216	100539	1.22598	286.737	2.0026
16.861	678.201	-0.348273	100832	1.22854	286.975	2.00073
17.9128	678.064	-0.11099	100947	1.22954	287.068	2.00001
18.9646	678.018	-0.0311268	100985	1.22987	287.099	1.99976
20.0163	678.004	-0.00769674	100996	1.22997	287.108	1.99969
21.0681	678.001	-0.00168036	100999	1.22999	287.111	1.99967
22.1199	678	-0.000324062	101000	1.23	287.111	1.99967
23.1717	678	-5.51761e-005	101000	1.23	287.111	1.99967
24.2234	678	-8.28105e-006	101000	1.23	287.111	1.99967
25.2752	678	-1.09263e-006	101000	1.23	287.111	1.99967
26.327	678	-1.26254e-007	101000	1.23	287.111	1.99967
27.3787	678	-1.271e-008	101000	1.23	287.111	1.99967
28.4305	678	-1.10713e-009	101000	1.23	287.111	1.99967
29.4823	678	-8.26712e-011	101000	1.23	287.111	1.99967
30.5341	678	-5.09967e-012	101000	1.23	287.111	1.99967
31.5858	678	-3.62091e-013	101000	1.23	287.111	1.99967
32.6376	678	-4.04607e-014	101000	1.23	287.111	1.99967
33.6894	678	-8.04491e-021	101000	1.23	287.111	1.99967
34.7411	678	-8.02739e-026	101000	1.23	287.111	1.99967
35.7929	678	-2.09317e-032	101000	1.23	287.111	1.99967
36.8447	678	-4.37427e-039	101000	1.23	287.111	1.99967
37.8965	678	-1.92917e-046	101000	1.23	287.111	1.99967
38.9482	678	-1.84923e-060	101000	1.23	287.111	1.99967
40	678	0	101000	1.23	287.111	1.99967

Table 3: Values of computed physics variables at $x = 32.1051$

5.1 Results for Different Courant Numbers

Now let us consider how different Courant numbers change the results. In the figure (9),(10) and (11) solutions for three different Courant numbers are sketched.

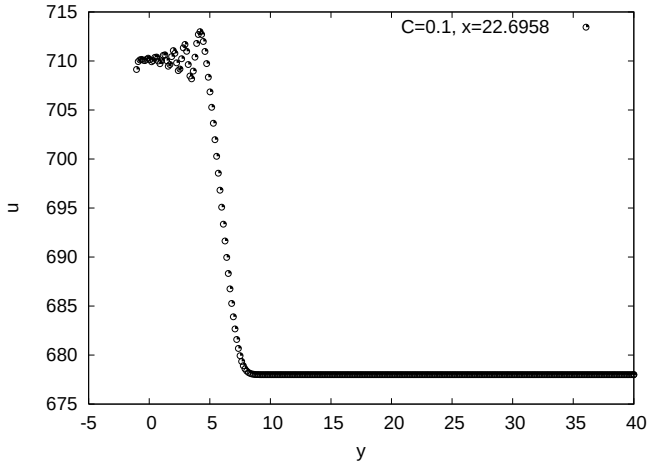


Figure 9: Velocity values for $C = 0.1$

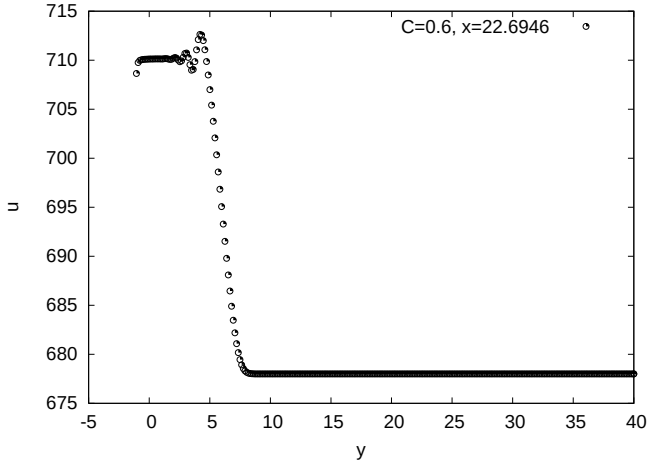


Figure 10: Velocity values for $C = 0.6$

As we see in the figures we obtain different results for different Courant numbers. For $C_y = 0.1$ some big oscillations near expansion corner are observed, that is quite interesting result. For $C = 0.6/0.9$ better results with small oscillations were obtained[8]

[8]Small note about Andersson's book [1] : there are some strange results presented - on a figure 8.8 nice and smooth numerical results are shown, but in a table 8.4 we see... oscilations for u velocity component. It is not strange that two different results are obtained with the same method??? I spend about two weeks on searching bug in my code and I see now that it was not my mistake.

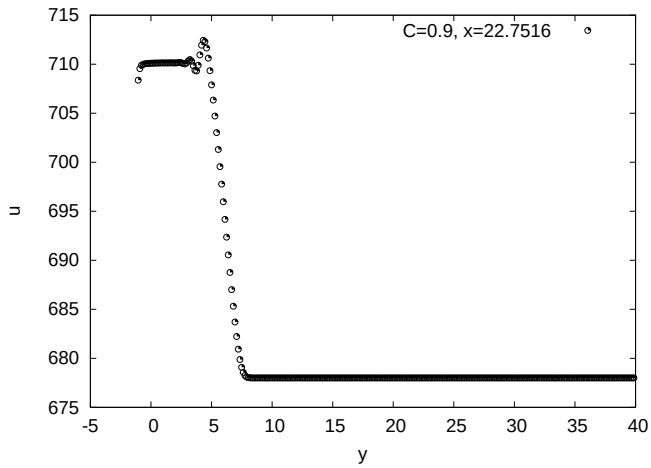
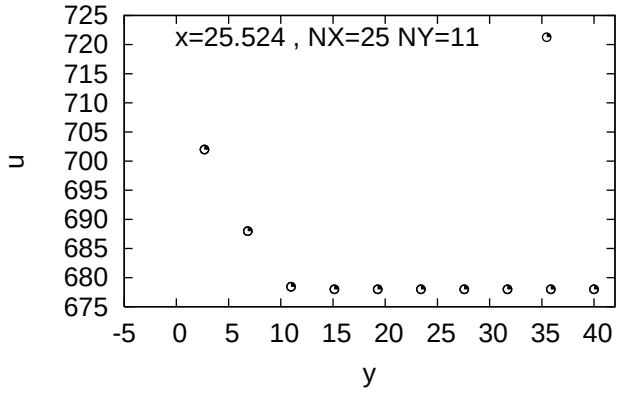


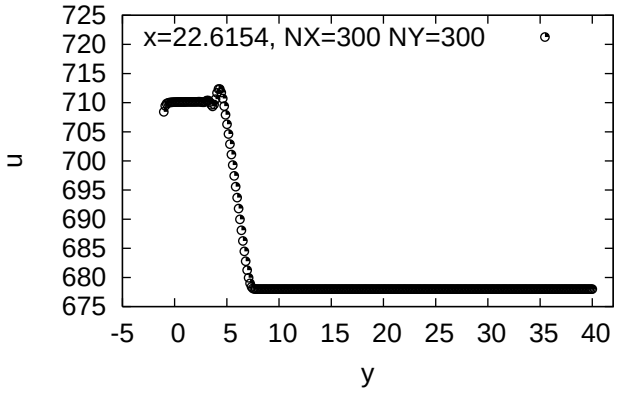
Figure 11: Velocity values for $C = 0.6$

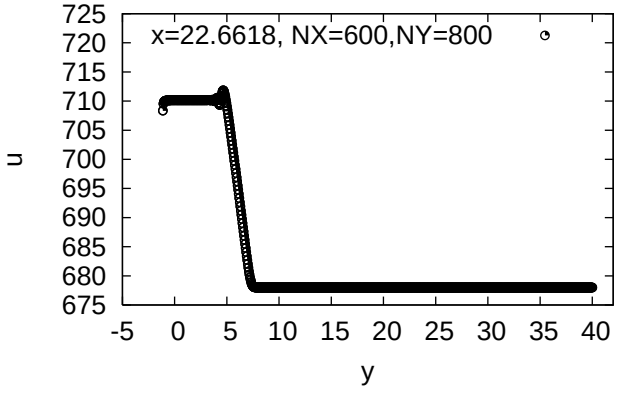
5.2 Results for Different Grid Resolution

Now let us consider how changes of spatial resolution of computational grid change the results. Previous sketches were provided for grid which contained NX points in x direction and NY points in y direction. Three results obtained for different spatial resolution of computational grid are shown in figures (5.2), (5.2) and (5.2). Plots were made with *spline* interpolation, computed points are plotted too, with black filled circles.



As it is shown on the figures (5.2), (5.2) and (5.2), accuracy of MacCormack's numerical procedure depends on spatial resolution of the computational grid. For $NX = 25$ and $NY = 11$ we obtained not quite correct result - u velocity does not increase too much, and behavior of u field is far from being correct (see fig. 5.2). For $NX = 300$





and $NY = 300$ we obtained much better results and it is enough to say that results are correct. However, still some oscillations at the corner of the sketch occurs. But take a look on figure (5.2) where solution for resolution $600x800$ is presented. It is impressive - very high accuracy of that results assures as that higher resolution of the grid will increase accuracy of the numerical solution.

6 True Color Visualization of the Results

In that section I will provide my original idea for visualization of the results which are obtained with my Prandtl-Meyer solver. Classic sketches which were shown in the previous sections are nice but I think that some recent ideas could help to better understand the wave nature of the Prandtl-Meyer Expansion. The first question which appears when we are analyzing figures with u velocity (like fig. 5.2) is: how all velocity field looks like? I can show there sketches for several i positions on the computational plane, but... how much? I tried to create full view of the u velocity field. I wrote a function which exports datas to the .ppm $24bit$ portable graphics file format. First picture in figure (12) represents the full u velocity field for standard initial conditions (described at the beginning of the "Results" section).

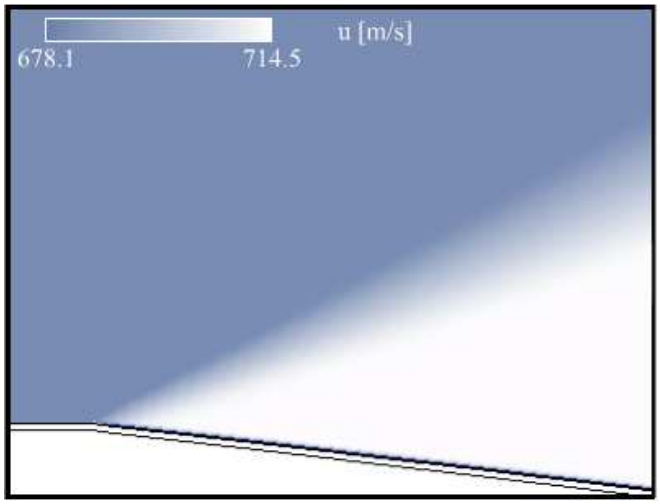


Figure 12: $24bit$ visualization of u velocity field in computational plane.

It is obvious that figure (12) contains more physics information that simple sketches presented before. Let us now try to look for some other physics variables i.e. temperature scalar field is provided in figure (5).

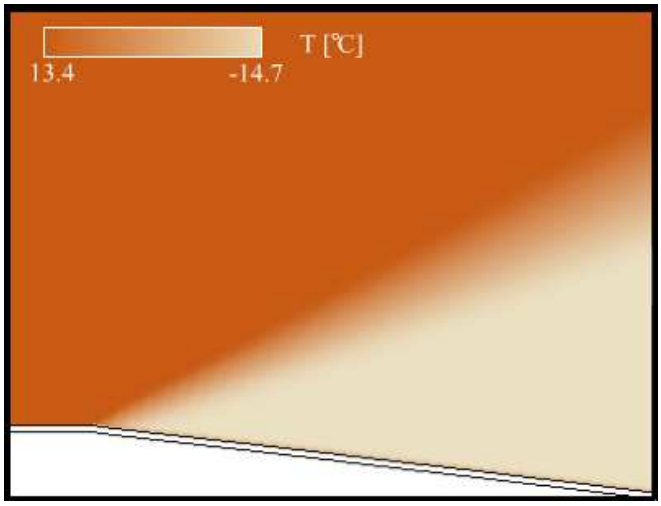


Figure 13: $24bit$ visualization of T scalar field in computational plane.

It is easy to realise where the highest temperature is. Anyway, on these figures some information about maxi-

mum and minimum values of considered variables is provided.

7 Visualization by animation

While stable pictures can be interesting, nothing is better that animation. For the Prandtl-Meyer solution there are some possibilities to improve some movements... Let us deal with the last assignment direction - we have to calculate solution for different θ expansion corner angles. I used export function which I wrote before and I used it for output data for several increasing θ values. I started with $\theta = 0.2$, after calculation output to a .ppm file was done. Then I added value 0.2^o to θ and I made the next computation step (now computation step means full PM solution). That was repeated from $\theta = 0.2^o$ to $\theta = 23^o$.

Presented results[9] show several frames from an animation. Every frame contains informations about u velocity field. I hope nobody will blame me for capturing the animation with Real One player skin.

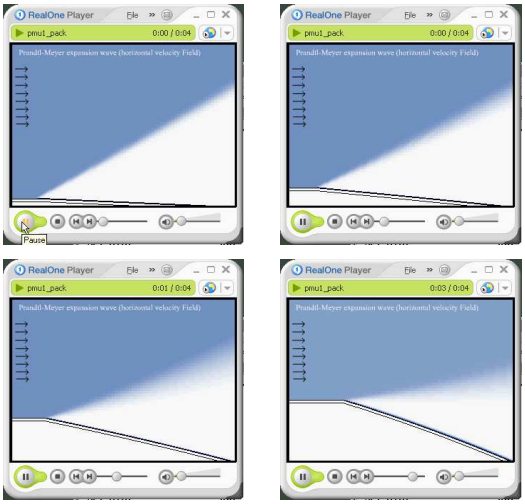


Figure 14: Animation: changes of θ value.

8 Conclusion

I think that some more work could be done on data visualization of the Prandtl-Meyer solution but because of little time left and some problems with numerical procedure I cannot continue my work on that topic. One thing I have done but is not included in that report is the visualization of vector velocity field (both v and u velocity components) in one picture (vector lines technique). If somebody is interesting in animations and pictures presented in that report, please contact me.

I would like to thank Johan Svensson for his help with some problems which occured during my work on that assignment. I would like also to thank Grzegorz Juraszek for checking my english mistakes in that report.

[9] http://panoramix.ift.uni.wroc.pl/∼maq

References

[1] John D. Anderson, Jr. 'Computational Fluid Dynamics: The Basics with Applications', McGraw-Hill Inc, 1995.

[2] Ryszard Grybos, 'Podstawy mechaniki plynow' (Tom 1 i 2), PWN 1998.

[3] David Potter 'Metody obliczeniowe fizyki', PWN 1982.

Contents