MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING
CAMBRIDGE, MASSACHUSETTS 02139

2.29 NUMERICAL FLUID MECHANICS— SPRING 2007

Solution of Quiz 2

Takehome 48 hours, Totally 25 points
Due Thursday 4 p.m. 05/17/07, Focused on Lecture 12 to 25 (Last Lecture)

Note that you are not allowed to collaborate or share your thoughts about the problems. Please state your assumptions and write down clearly what you think about the problems even if you cannot solve them to the endpoint. Furthermore, note that we do not need you to attach your codes and we will not look through them to find what you have done, instead explain your method.

Problem 1 (5 points):

The boundary layer equation for the self-similar incompressible flow over a flat plate can be cast as the following equation and boundary conditions set:

$$f''' + \frac{1}{2} f'' f = 0, \; where \; f''' = \frac{d^3 f}{dx^3}, f'' = \frac{d^2 f}{dx^2}$$

$$\begin{cases} f(0) = 0 \\ f'(0) = 0 \\ f'(\infty) = 1 \end{cases}$$

1. Solve the equation with your method of choice and plot the f(x) curve.
2. Find at which "x" the f' value becomes equal to "0.99" . A minimum accuracy equal to 0.1% is expected.

Solution:

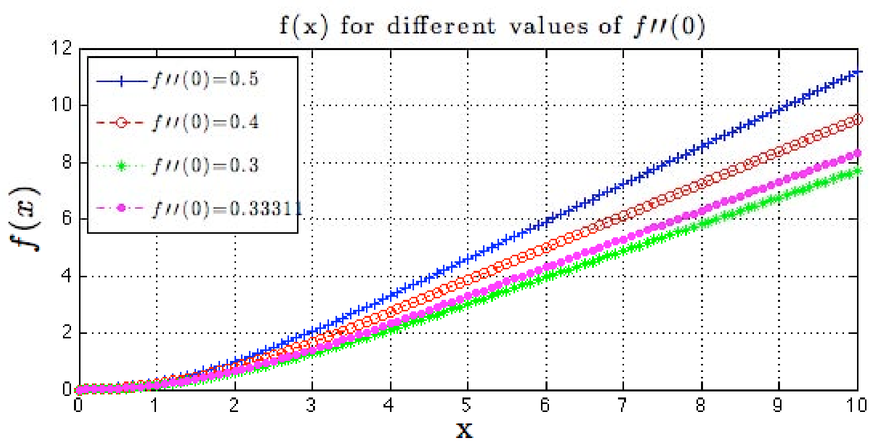
Note that this is a nonlinear differential equation and unfortunately most methods that we have studied (like finite difference and finite elements) cannot be applied routinely to this problem. However, while solution methods of BVP's (BVP: boundary value problem) are usually limited to linear equations, we can solve arbitrary complicated ODEs (ODE: ordinary differential equations).

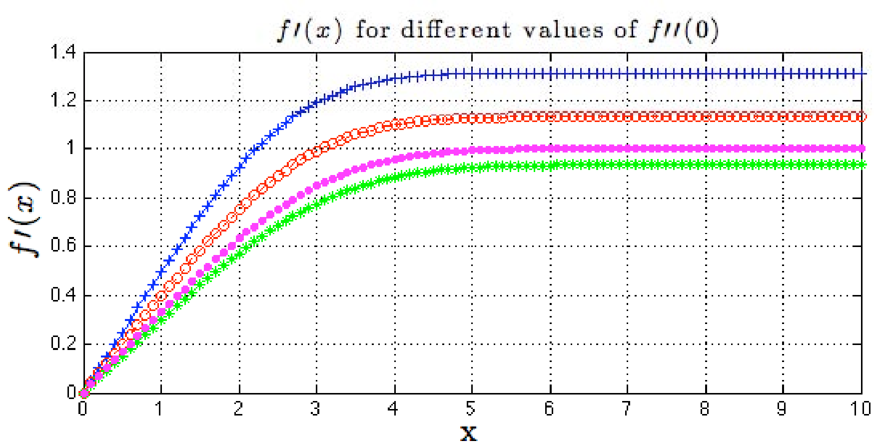
The key trick here is to transform the BVP into an ODE and it is accomplished by shooting method. Basically we choose an arbitrary value for $f''(0)$ and solve the below ODE set. Then we look at asymptotic behavior of our function at rather a large value, where we see a stationary value for $f'(\infty)$. Next we update the $f''(0)$ by trial and error (or more advanced techniques of root finding applied to $f'(\infty)-1$) until we achieve our required accuracy.

$$Y(x)=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}=\begin{bmatrix} f \\ f' \\ f'' \end{bmatrix}$$

$$\frac{dY}{dx}=\begin{bmatrix} f' \\ f'' \\ f''' \end{bmatrix}=\begin{bmatrix} y_2 \\ y_3 \\ -\frac{1}{2}y_1y_3 \end{bmatrix}$$

The whole process is done in attached "C2p29_Quiz2_1.m" file and here we have plots for some initial guesses on solution.

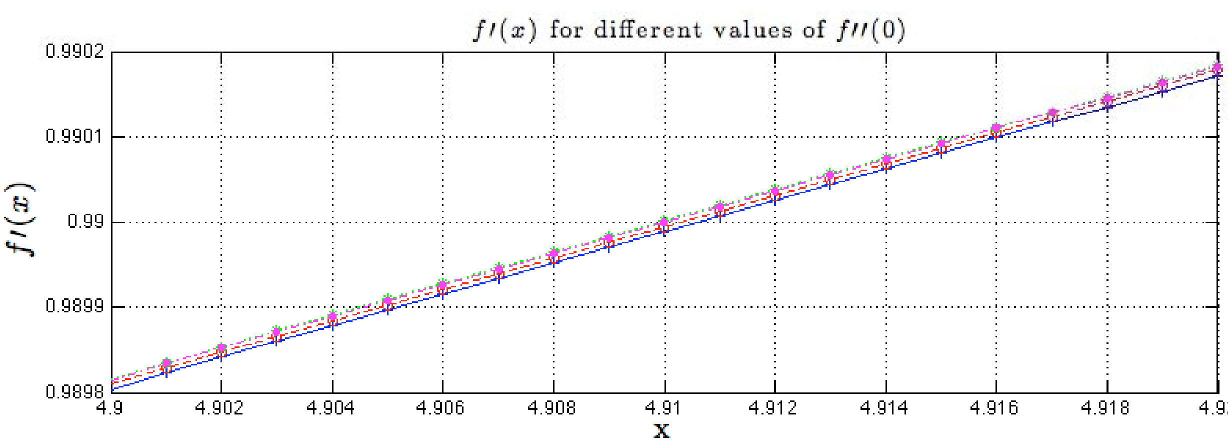




By playing with program we can find the right value of $f''(0) \cong .33206$. This value can be deducted by following different runs summarized below (in each set the last value is based on linear approximation of 2nd and 3rd value). Note that “ode” default settings are based on a relative and absolute tolerance about 10^{-3} and 10^{-6}. On the other hand we have set the relative and absolute tolerance of error to conservative value of 10^{-10} and 10^{-12} to ensure the reliability of our printed value to 9 significant digits.

```
f"(0)=0.50000 -->> f'(x=10.00000 )=1.31372546
f"(0)=0.40000 -->> f'(x=10.00000 )=1.13213428
f"(0)=0.30000 -->> f'(x=10.00000 )=0.93455625
f"(0)=0.33312 -->> f'(x=10.00000 )=1.00213835

f"(0)=0.50000 -->> f'(x=20.00000 )=1.31372546
f"(0)=0.40000 -->> f'(x=20.00000 )=1.13213428
f"(0)=0.30000 -->> f'(x=20.00000 )=0.93455626
f"(0)=0.33312 -->> f'(x=20.00000 )=1.00213835

f"(0)=0.32967 -->> f'(x=20.00000 )=0.99520122
f"(0)=0.33300 -->> f'(x=20.00000 )=1.00189168
f"(0)=0.33633 -->> f'(x=20.00000 )=1.00855987
f"(0)=0.33206 -->> f'(x=20.00000 )=0.99999596

f"(0)=0.3317679 -->> f'(x=20.00000 )=0.99941882
f"(0)=0.3321000 -->> f'(x=20.00000 )=1.00008565
f"(0)=0.3324321 -->> f'(x=20.00000 )=1.00075227
f"(0)=0.3320573 -->> f'(x=20.00000 )=0.99999998

f"(0)=0.3320400 -->> f'(x=20.00000 )=0.99996519
f"(0)=0.3320500 -->> f'(x=20.00000 )=0.99998527
f"(0)=0.3320600 -->> f'(x=20.00000 )=1.00000535
f"(0)=0.3320573 -->> f'(x=20.00000 )=1.00000000

f"(0)=0.3320520 -->> f'(x=100.00000 )=0.99998929
f"(0)=0.3320550 -->> f'(x=100.00000 )=0.99999531
f"(0)=0.3320580 -->> f'(x=100.00000 )=1.00000133
f"(0)=0.3320573 -->> f'(x=100.00000 )=1.00000000

f"(0)=0.3320520 -->> f'(x=1000.00000 )=0.99998929
f"(0)=0.3320550 -->> f'(x=1000.00000 )=0.99999531
f"(0)=0.3320580 -->> f'(x=1000.00000 )=1.00000133
f"(0)=0.3320573 -->> f'(x=1000.00000 )=1.00000000
```

Finally we can look at the graph and find that $f'(4.910) \approx 0.990$ reliably within required accuracy. Note that this graph corresponds to the last set of iterations on $f''(0)$, within which $f''(0)$ varies in 6th significant digit.



Problem 2 (5 points):

The steady state, fully developed laminar viscid flow in a rectangular channel with square cross section can be represented by this equation:

$$\nabla^2 u(x,y) = u_{xx} + u_{yy} = -1,\ |x| < 1, |y| < 1$$
$$\begin{cases} u(\pm 1, y) = 0 \\ u(x, \pm 1) = 0 \end{cases}$$

1. Assume an approximate solution of the form $U(x,y) = a_0 \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)$ and find a_0 value by Galerkin's method.
2. Evaluate the normalized flow rate value ($Q^* = \int_A U(x,y)dA = \int_{-1}^{1}\int_{-1}^{1} U(x,y)dxdy$) and compare it to the analytical value which is equal to $Q^*_{analytical} = 0.5623$.
3. (EXTRA CREDIT 2 Points) Evaluate the normalized maximum shear stress ($\tau^* = \max(|\nabla U(x,y)|)$) and compare it to the analytical value which is equal to $\tau^*_{analytical} = 0.675$.

Solution:

1. We consider operator $L(u) = u_{xx} + u_{yy} + 1,\ |x| < 1, |y| < 1$ and the shape function $\phi(x,y) = \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)$. Then we set the residual of operator "L" applied to the $U(x,y) = a_0\phi(x,y)$ equal to zero with respect $\phi(x,y)$:

$$L(U) = a_0\left\{-(\frac{\pi}{2})^2 \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y) + \cos(\frac{\pi}{2}x) \times -(\frac{\pi}{2})^2 \cos(\frac{\pi}{2}y)\right\} + 1$$
$$L(U) = -\frac{a_0\pi^2}{2}\cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y) + 1$$

$$< L(U), \phi > = \int_{|x|<1, |y|<1} L(U)\phi dA = 0$$
$$< L(U), \phi > = \int_{|x|<1, |y|<1} L(U)\phi dA = 0$$

$$< L(U), \phi > = \int_{-1}^{1}\int_{-1}^{1}\left\{-\frac{a_0\pi^2}{2}\cos^2(\frac{\pi}{2}x)\cos^2(\frac{\pi}{2}y) + \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)\right\}dxdy = 0$$

$$< L(U),\phi >= \left\{ -\frac{a_0\pi^2}{2}(1)^2 + (\frac{4}{\pi})^2 \right\} = 0$$

$$a_0 = \frac{32}{\pi^4} \cong 0.3285$$

2.

$$Q^* = \int_A U(x,y)dA = \int_{-1}^{1}\int_{-1}^{1} a_0 \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)dxdy$$

$$Q^* = a_0(\frac{4}{\pi})^2 = \frac{512}{\pi^6} \cong 0.5326$$

Compared to analytical value of $Q^*_{analytical} = 0.5623$ our estimate has an error about 5.3% .

3.

$$\nabla U(x,y) = -\frac{a_0\pi}{2}\begin{bmatrix} \sin(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y) \\ \cos(\frac{\pi}{2}x)\sin(\frac{\pi}{2}y) \end{bmatrix}$$

$$\left|\nabla U(x,y)\right|^2 = \left(\frac{a_0\pi}{2}\right)^2 \left(\sin^2(\frac{\pi}{2}x)\cos^2(\frac{\pi}{2}y) + \cos^2(\frac{\pi}{2}x)\sin^2(\frac{\pi}{2}y) \right)$$

A few lines of algebra (for example by finding the stationary points via derivative) proves that the maximum of $\left(\sin^2(\frac{\pi}{2}x)\cos^2(\frac{\pi}{2}y) + \cos^2(\frac{\pi}{2}x)\sin^2(\frac{\pi}{2}y) \right)$ is equal to “1” and happens at the crosses of axis with square boundary (as we know from our intuition or analytical results.

$$\tau^* = \max(\left|\nabla U(x,y)\right|) = \frac{a_0\pi}{2} = \frac{16}{\pi^3} \cong 0.516$$

Since $\tau^*_{analytical} = 0.675$ we have about 23.6% error in estimation of maximum shear stress. This is a rather higher error compared to flow rate and note that we have a low order approximation of solution. Consequently, the flow rate which is an integral of solution will be more accurate than the shear stress which is based on gradient of solution (because higher order terms will be magnified in derivatives).

Problem 3 (5 points):

The velocity of air passing over the surface of an airfoil has been measured at different distances from the surface and has been reported in the below table. Assume that air viscosity is given by $\mu = 1.65\times10^{-5}\,\frac{\mathrm{N}\times\mathrm{sec}}{\mathrm{m}^2}$ and compute the shear stress over the surface.

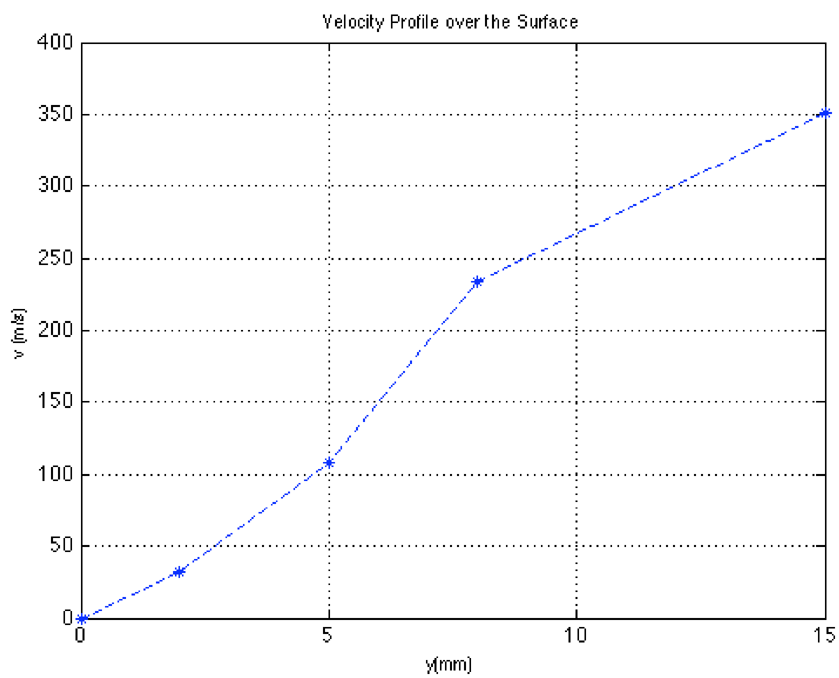
y(mm)	Velocity(m/s)
0	0
2	31.6
5	107.9
8	233.5
15	350.6

Solution:

Note that this is adopted from a homework problem (EXTRA CREDIT 23.26).

$$\tau\Big|_{y=0} = \mu\frac{dv}{dy}\Big|_{y=0}$$

We want to compute our derivative with at least $o(h^2)$ accuracy. However, we cannot use the central finite difference scheme and we have to rely on forward derivative approximations. Since the points are not equally spaced we will fit a polynomial to it and then we will compute the derivative of polynomial at y=0 (which is the coefficient of "y" at our fitted polynomial).



First we plot the data points to see their curve (previous page). Clearly from plot we can see that either data point 4 or 5 does not match the trend line that we observe[1]. Furthermore note that data point 4 corresponds to a higher shear stress at the middle of air layers, which does not make sense. Consequently we neglect the 4th data point and fit the rest of data points to a 3rd order polynomial which results in $\tau\big|_{y=0} \cong 0.17 \ \frac{N}{m^2}$. Other possible fits result in higher shear stresses and are derived in below calculations ($\tau\big|_{y=0} \cong 0.26, 0.21, 0.20 \ \frac{N}{m^2}$ corresponding to 4th, 3rd and 2nd order fits)[2].

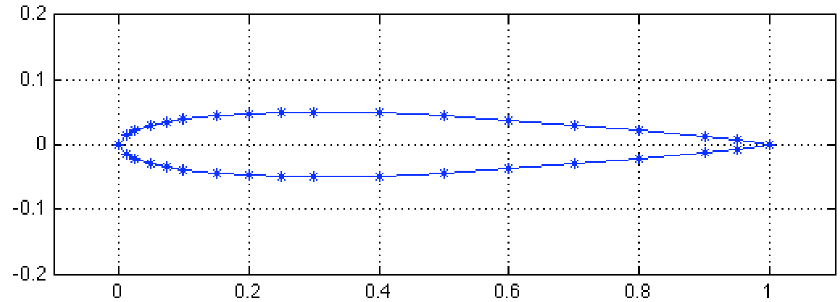
```
>> Mu=1.65e-5; %Viscosity in N.sec/m^2
>> y=[0 2 5 8 15]; %Height from Surface in mm
>> v=[0 31.6 107.9 233.5 350.6]; %Velocity in m/s
>>
>> plot(y,v,'*--')
>> xlabel('y(mm)'), ylabel('v (m/s)'), title('Velocity Profile over the Surface'), box on, grid on
>> set(gcf,'Position',[50 50 [25 25]*25],'color','w'),set(gca,'fontsize',12)
>> p3=polyfit(y([1:3 5]),v([1:3 5]),3) %Use the best 4 points fitted on a 3rd order Polynomial (o(h^3) accurate)

p3 =

  -0.134410256410256   2.867538461538450   10.602564102564170  -0.000000000000086

>> dv_dy=p3(end-1)*1000 % Derivative at y=0

dv_dy =

     1.060256410256417e+04

>> tau=Mu*dv_dy  % Evaluated at units of N/m^2

tau =

   0.174942307692309

>> p4=polyfit(y,v,4) %Use all data Points fitted on a 4th order Polynomial

p4 =

  -0.033705433455433   0.607109279609276  -1.008586385836353  15.658379120879014   0.000000000000063

>> dv_dy=p4(end-1)*1000; % Derivative at y=0
>> tau=Mu*dv_dy  % Evaluated at units of N/m^2

tau =

   0.258363255494504

>> p3=polyfit(y(1:4),v(1:4),3) %Use the 1st 4 points fitted on a 3rd order Polynomial (o(h^3) accurate)

p3 =

   0.101527777777778   1.215972222222218  12.961944444444470  -0.000000000000028

>> dv_dy=p3(end-1)*1000; % Derivative at y=0
>> tau=Mu*dv_dy  % Evaluated at units of N/m^2

tau =

   0.213872083333334

>> p2=polyfit(y(1:3),v(1:3),2) %Use the 1st 3 points fitted on a 2nd order Polynomial (o(h^2) accurate)

p2 =

   1.926666666666666  11.946666666666673  -0.000000000000009

>> dv_dy=p2(end-1)*1000; % Derivative at y=0
>> tau=Mu*dv_dy  % Evaluated at units of N/m^2

tau =

   0.197120000000000
```

[1] Unless we expect some adverse pressure effects (like regions before separation).

[2] Finally note that this is indeed a transonic flow and we should be careful about a constant viscosity assumption (due to varying pressure).

Problem 4 (10 points):

You are provided with "NACA0010.mat" file, which includes a discretized model of a standard symmetric NACA airfoil shown here:



1. Use simple 2D sources and solve the potential flow passing the above airfoil (assume that the angle of attack is zero). You are welcomed to use scripts given on the class but please write a few lines describing your work and the background math.
2. Plot velocity contours around the airfoil
3. Plot pressure contours around the airfoil.
4. Compute the total drag force on the airfoil
5. (EXTRA CREDIT 2 Points) Is your method applicable when the angle of attack is not zero? Explain why and provide alternatives if needed.

Solution:

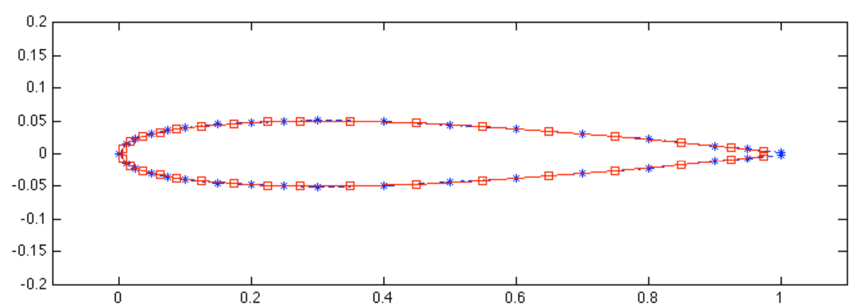
1.

THEORY:

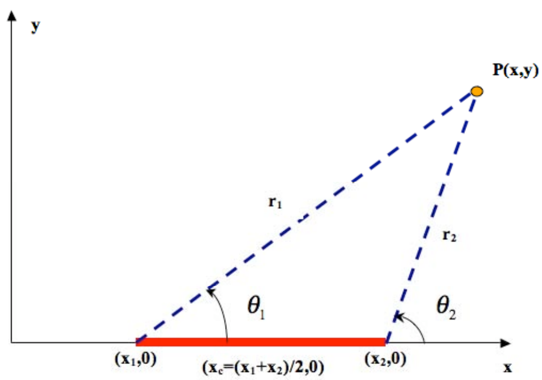
The total potential is defined as:

$$\phi(x, y) = Ux + \varphi(x, y)$$

Source potential or $\varphi(x, y)$ is based on a few sources positioned at $\vec{R}_c = (x_c, y_c)$, where "c" stands for source center (red squares in the next figures). The centers are basically middle points of consecutive vertices (blue stars) shown on the above figure.



Each source is a line segment which connects two consecutive vertices and has a source strength σ (in units of 2D source strength per unit length). The equations for each source are based on lecture 22 and are described in a local coordinate accordingly:



$$\varphi(x,y) = \frac{\sigma}{4\pi}[(x-x_1)\ln r_1^2 - (x-x_2)\ln r_2^2 + 2y(\theta_2-\theta_1)]$$ [3]

$$V_x = \frac{\partial\varphi}{\partial x} = \frac{\sigma}{2\pi}\ln\frac{r_1}{r_2}$$

$$V_y = \frac{\partial\varphi}{\partial y} = \frac{\sigma}{2\pi}(\theta_2-\theta_1)$$

Further detail can be found in the lecture notes. Note that above values normalized by source strength (divided by σ) are called as influence coefficients [4].

When we consider "n" sources, then we will need "n" equation to derive their strength. Usually we adopt Galerkin's method to impose the boundary condition (zero normal velocity at the boundary) for each source segment as a shape function. So for Galerkin's method we need an integration over the source segment. If we approximate the integration with central value (multiplied by source length,…), the Galerkin's equation will be equivalent to saying that normal velocity at the center of each source will be zero and here we use this as the simplest possible scheme. To that end we need

---

[3] In the current code $\frac{\sigma}{2\pi}$ is defined as source strength. Also the sign of U in current code is different.

[4] A critical aspect of panel method is that every source has an effect at every place. As a result the matrix of solution will be a dense matrix. On other hand if we solve the Laplace equation with finite difference or finite element over a forcibly equivalent large domain (because we have to approximate the infinite domain of the problem with a large domain) then the matrix will be a sparse one. Consequently, we can realize that by utilizing Green's theorem we transform the problem in the domain, to a problem on the boundary of domain, at the cost of changing a sparse matrix to a dense matrix.

to compute the influence coefficient of all sources at the center of each source. However, we should be careful about influence coefficient of a source on itself (due to local singularity). In particular:

$$V_x(x_c,0)=0$$
$$V_y(x_c,0)=\pm\frac{\sigma}{2} \text{ (sign dependet on the direction of approch or sign of y)}$$

Other than the central point of source, other points lying on the source segment have nonzero tangential velocity (in this coordinate V_x). Unfortunately, the vertices have infinite tangential velocity and the velocity field is non-continuous at those points. If we refine the mesh then velocity of vertices will be also less discontinuous[5].

Furthermore note that in general we need to describe velocities in global coordinate (instead of local coordinate of sources). To that end we utilize:

$$V_{\vec{n}}=(V_x\hat{i}+V_y\hat{j}).\vec{n}=V_xn_x+V_yn_y \quad \vec{n}\text{: unit vector in direction of interest}$$

IMPLEMENTATION:

We have used the lecture scripts. All the computations are normalize by unit velocity. We have to make only a slight change in "setpanels.m" to read our new geometry when running the program. Later main file "cpm_main.m" is used to calculate source strengths and produce incoming graphs.

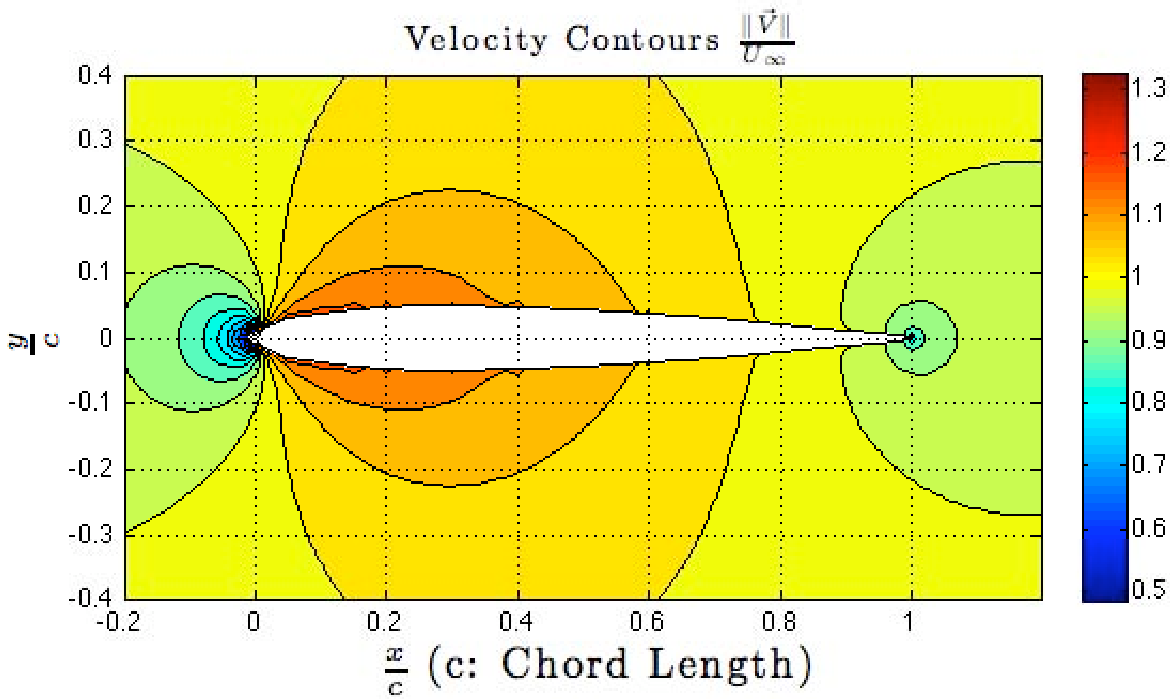
2.

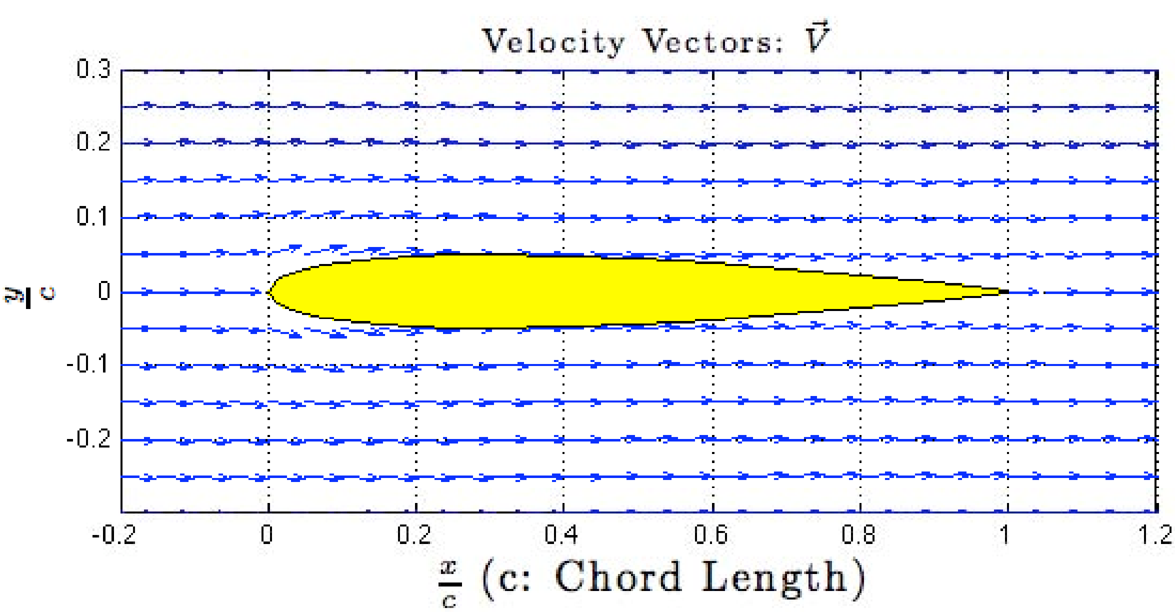
$$\phi(x,y)=Ux+\varphi(x,y)=Ux+\sum_{i=1}^{n}\sigma_i\varphi_i(x,y)$$
$$V_x=\frac{\partial\phi}{\partial x}=U+\sum_{i=1}^{n}\sigma_i\frac{\partial\varphi_i}{\partial x}(\ )$$
$$V_y=\frac{\partial\phi}{\partial y}=\sum_{i=1}^{n}\sigma_i\frac{\partial\varphi_i}{\partial y}$$

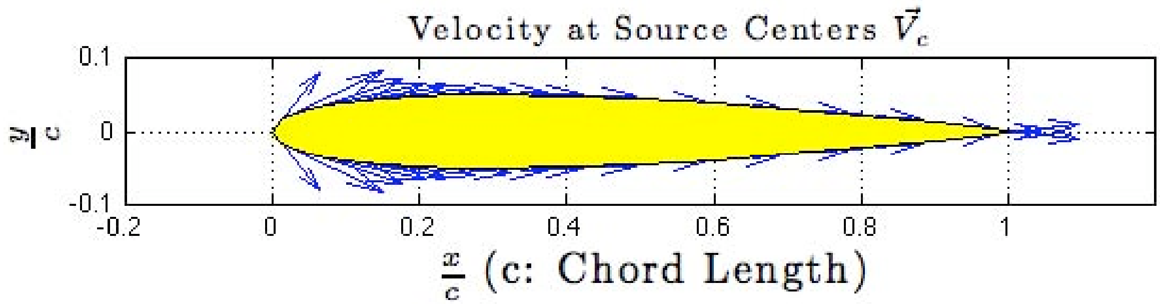
For the calculation of strength we only need the influence coefficient at the center of each source. However, to compute the velocities at arbitrary points we have to compute the influence coefficients at arbitrary points. Consequently we have to make a slight change at "inflcoef.m" file.

[5] Because in that case, two neighborhood sources will be almost parallel so they will almost cancel other's infinite tangential velocity at vertices.

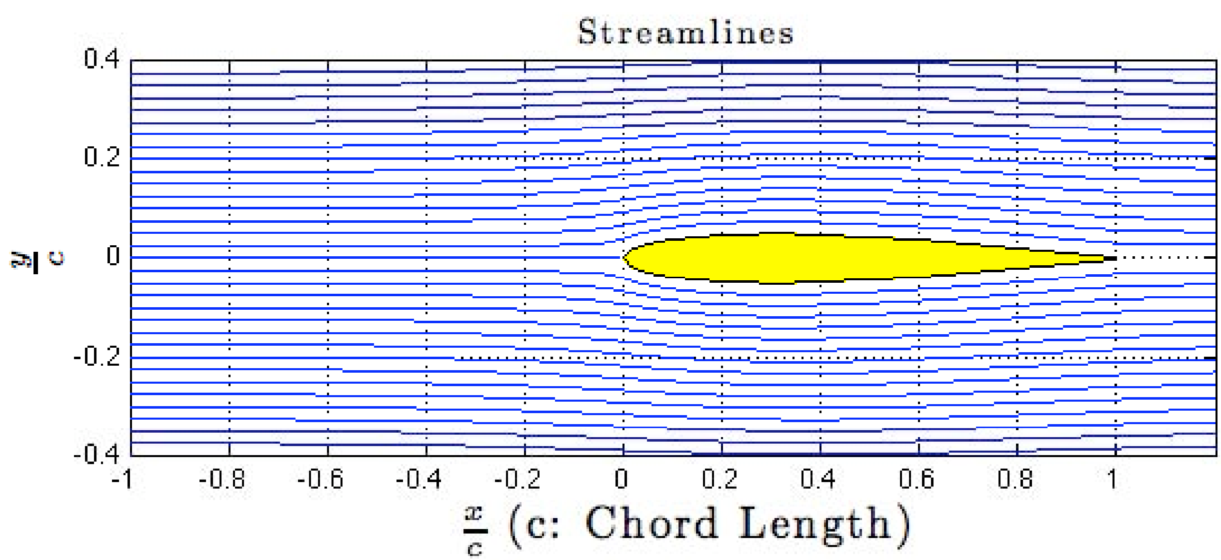
To plot the data the “contourf” command has been used. Note that if we refine our evaluation point too much, we might get very close to vertices and get infinite velocities which will spoil the contour plot. In that case we can either manually set the contour level or use “find” command to eliminate those points. Also note that “fill” command can be very useful to plot the airfoil as a solid object over the contours (which includes suspiciously calculated internal points).







Finally we generate the streamlines. The command "streamline" is very well suited for this purpose and the below graph has been produced by that. However, to get good results we should set the starting point of streamline calculation at a rather distant position (here x=-1). where we have uniform flow (because values of streamline function corresponds to flow rate).



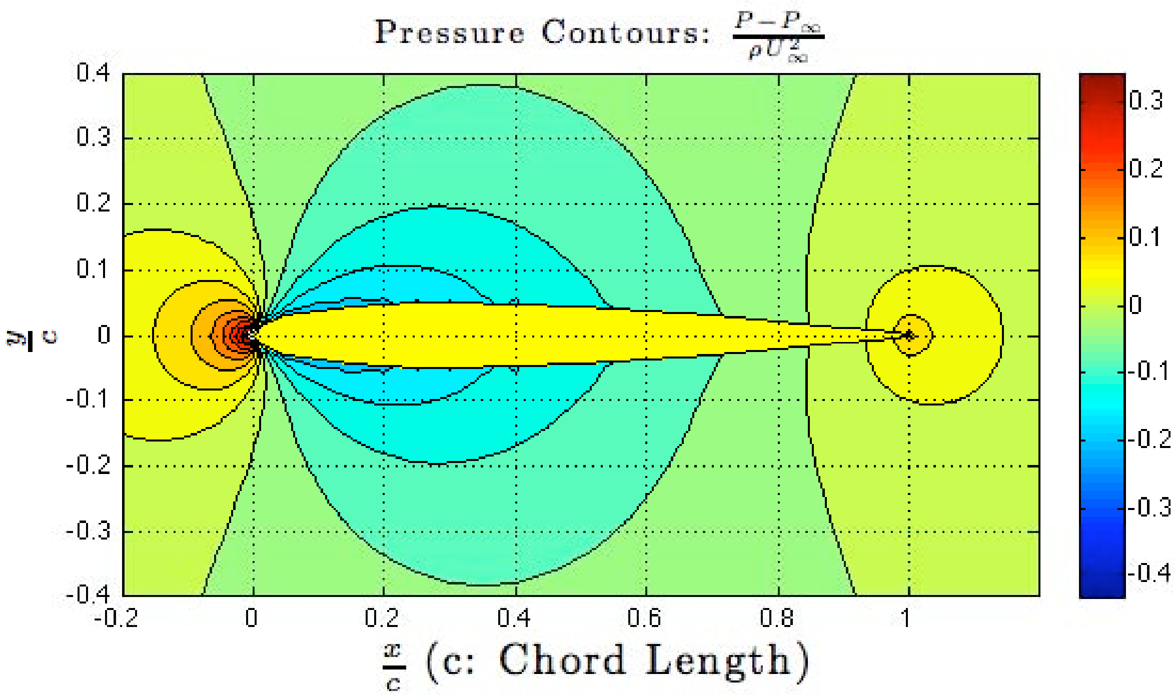
3.

$$const = \frac{1}{2}\left|\vec{V}\right|^2 + \frac{P}{\rho}$$

$$\frac{1}{2}U_\infty^2 + \frac{P_\infty}{\rho} = \frac{1}{2}\left|\vec{V}\right|^2 + \frac{P}{\rho}$$

$$P = \frac{1}{2}\rho(U_\infty^2 - \left|\vec{V}\right|^2) + P_\infty$$

As a result we can see that pressure contours are basically similar to contours of velocity magnitude.



4. Recall D'Alembert's Paradox and note the theorem which states that no forces act on a body moving at constant velocity in a straight line through a large mass of incompressible, inviscid fluid which was initially at rest (or in uniform motion). There might be some lift (perpendicular to uniform velocity), but only if we have some circulation introduced in the flow somehow else. Here we can integrate the pressure force and get a very small value which is due to numerical errors. The actual drag mostly due to separation and viscous effect, both not presented here.

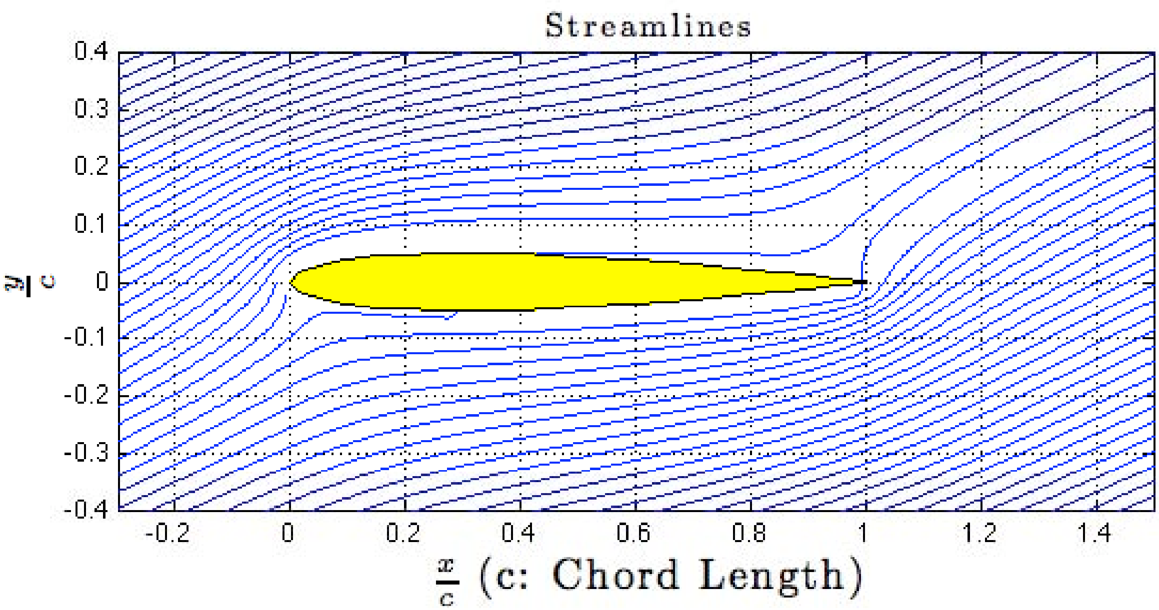
5. We can do this either by rotating vertices initially or changing the equation accordingly (which affect the boundary conditions as well):

$$\phi(x,y) = U(x\cos\theta + y\sin\theta) + \varphi(x,y) = U(x\cos\theta + y\sin\theta) + \sum_{i=1}^{n} \sigma_i \varphi_i(x,y)$$

An example run is shown on the next page using both methods. But there are two critical points which indeed makes them INCORRECT:

- This solution is only valid for small angles of attack (otherwise we have separation and it will change the whole solution)
- In those cases we have lift and much more complex flow behavior near trailing edge. Furthermore to compute lift we need to include circulations. Circulation requires incorporation of more complex sources (like dipoles), which is not included in our model. Indeed we need to incorporate the circulation to satisfy Kuta's condition (to have a finite and reasonable velocity profile on the trailing edge).

Solution by changing the equation for $\alpha = \frac{\pi}{8} = 22.5^o$ (angle of attack):



Solution by rotating vertices at the geometry reading section for $\alpha = \frac{\pi}{8} = 22.5^o$ (angle of attack):