Worksheet Solutions: Potential Flow Part A

Question 1:

LHS boundary:

Flux = 10 A/m

Q = 5 A

$$F_x = 10 = \frac{d\psi}{dy}$$

$$\psi_L = 10y + A$$

Top boundary:

Q = 2.5 A

Linear change in flux:

$$F_y = 1.25(2 - x) = -\frac{d\psi}{dx}$$

$$\psi_T = -1.25(2x - 0.5x^2) + B$$

RHS boundary:

Q = 0 A

$$\psi_R = C$$

Bottom boundary

Q = 2.5 A

Linear change in flux:

$$F_y = -1.25(2 - x) = -\frac{d\psi}{dx}$$

$$\psi_B = 1.25(2x - 0.5x^2) + D$$

We can arbitrarily choose one value of ψ. I will set $C = 0$. The values of ψ must meet at the corners where boundaries meet:

$$\psi_L = 10y - 2.5$$

$$\psi_T = 2.5 - 1.25(2x - 0.5x^2)$$

$$\psi_R = 0$$

$$\psi_B = 1.25(2x - 0.5x^2) - 2.5$$

Question 2:

```
%pylab inline
import math
imax=201
jmax=51
Psi=zeros((imax,jmax))
dx=2.0/(imax-1)

for i in range(imax):
   x=dx*i
   Psi[i,0]=1.25*(2*x-0.5*x*x)-2.5
   Psi[i,jmax-1]=2.5-1.25*(2*x-0.5*x*x)
for j in range(jmax):
   y=dx*j
   Psi[0,j]=10*y-2.5
   Psi[imax-1,j]=0

for i in range(1,imax-1,1):
   for j in range(1,jmax-1,1):
      Psi[i,j]=0.0

residave = 1000.0
cnt = 0

a = 1.0; b = 1.0; c = 1.0; d = 1.0; e = -4.0; f = 0.0

Psi_old = copy(Psi)

while residave>1.0e-6:
   Psi[1:-1, 1:-1] = (1.0 / e) * (f - (a * Psi_old[2:, 1:-1] + b * Psi_old[:-2, 1:-1] + c * Psi_old[1:-1, 2:] + d *
Psi_old[1:-1, :-2]))
   residave = sl.norm(Psi-Psi_old)/np.maximum(1.0e-10,sl.norm(Psi))

   temp = Psi
   Psi = Psi_old
   Psi_old = temp

   cnt+=1
   if cnt%100==0:
      print("%g: resid=%f" % (cnt,residave))
fig1=figure()
imshow(Psi.transpose(),origin='lower')
```

Question 3:

```
fx=zeros((imax,jmax))
fy=zeros((imax,jmax))
```

```
for i in range(0,imax,1):
   for j in range(0,jmax,1):
      if j>0 and j<jmax-1:
         fx[i,j]=(Psi[i,j+1]-Psi[i,j-1])/(2.*dx)
      elif j>0:
         fx[i,j]=(Psi[i,j]-Psi[i,j-1])/dx
      else:
         fx[i,j]=(Psi[i,j+1]-Psi[i,j])/dx

      if i>0 and i<imax-1:
         fy[i,j]=-(Psi[i+1,j]-Psi[i-1,j])/(2.*dx)
      elif i>0:
         fy[i,j]=-(Psi[i,j]-Psi[i-1,j])/dx
      else:
         fy[i,j]=-(Psi[i+1,j]-Psi[i,j])/dx

stride = 5
quiver(fx.transpose()[::stride,::stride],fy.transpose()[::stride,::stride])
```