Problem 1.d) The following is the code used for calculating the Δf vs Δx showing the convergence study

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 30 16:51:16 2018

@author: alfred_mac
"""

from __future__ import unicode_literals
import numpy as np
import matplotlib
matplotlib.rcParams['text.usetex'] = True
matplotlib.rcParams['text.latex.unicode'] = True
import matplotlib.pyplot as plt

x = 5       # Value of x around which the delta is taken
xk = pow(2.0,np.arange(-30,2)) # Taking different values of del x
yk = np.exp(x)
yk1 = np.exp(x+xk)
ykm1 = np.exp(x-xk)

ykb = (np.exp(xk/2) - np.exp(-xk/2))*yk/xk
yk1b = (np.exp(xk/2) - np.exp(-xk/2))*yk1/xk
ykm1b = (np.exp(xk/2) - np.exp(-xk/2))*ykm1/xk

diff = ((13/12)*ykb) - ((1/24)*(yk1b+ykm1b)) - yk

plt.plot(xk,diff)

plt.xscale('log')
plt.yscale('log')

plt.xlabel(r'$\Delta{x}$')
plt.ylabel(r'$\Delta{f}$')
plt.show()
```

Problem 3.a) The following is the code used for calculating the Δf vs Δx showing the convergence study for the first order Panel Integrator

```
#FUNCTION f.py
#!/usr/bin/env python3
```

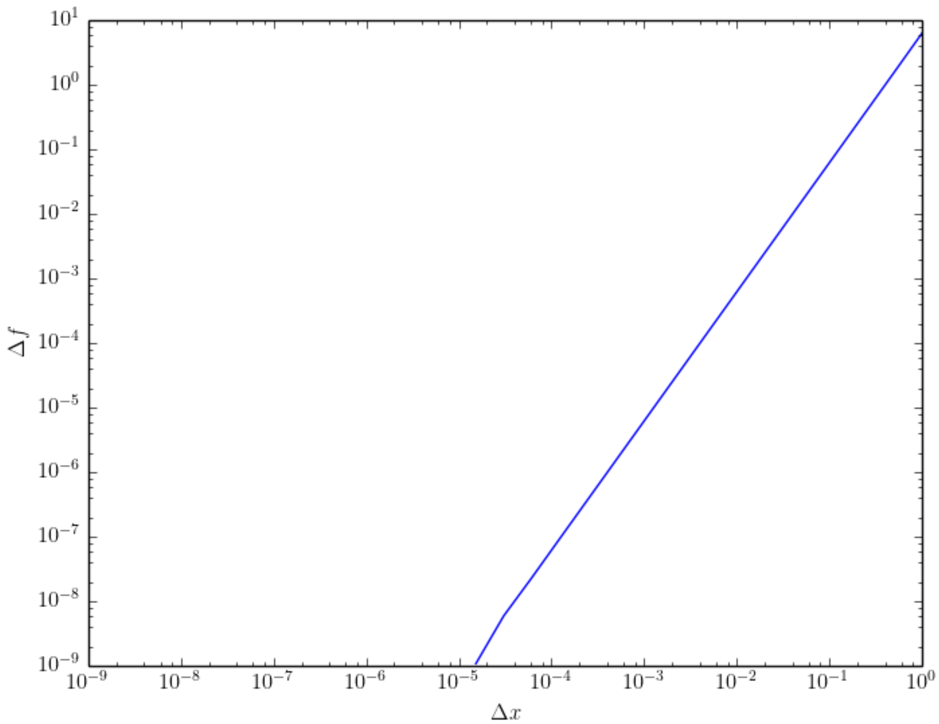

Figure 1: Δf vs Δx for the convergence study of $f(x) = e^x$ in problem 1(d)

```
# -*- coding: utf-8 -*-
"""
Created on Mon Oct  1 00:56:32 2018

@author: alfred_mac
"""

from __future__ import unicode_literals
import numpy as np
import matplotlib
matplotlib.rcParams['text.usetex'] = True
matplotlib.rcParams['text.latex.unicode'] = True
import matplotlib.pyplot as plt

def f(x):
    return np.sin(x)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 30 23:09:06 2018

@author: alfred_mac
"""

from __future__ import unicode_literals
import numpy as np
import matplotlib
matplotlib.rcParams['text.usetex'] = True
matplotlib.rcParams['text.latex.unicode'] = True
import matplotlib.pyplot as plt

def PanelIntegrator( n, a, b):  # integrate from a to b with n panels
    dx = (b-a)/n
    import f                    # a module that defines f(x)
    x = np.linspace(a,b,n+1)
    I = sum(dx*f.f(x))          # evaluate f at the point x
    return I                    # estimate of the integral

# Convergence study for Panel Integrator
N = 10
INTG = np.linspace(1,2,N)*0
DIFF = INTG*0

for i in range(N):
    INTG[i] = PanelIntegrator(1e1*pow(2,i),0,2)
    DIFF[i] = INTG[i] - 1 + np.cos(2)

X = 2/(1e1*pow(2,np.arange(N)))
plt.plot(X,DIFF)

plt.xlabel(r'$\Delta{x}$')
plt.ylabel(r'$\Delta{f}$')

plt.show()
```

Problem 3.b) The following is the code used for calculating the Δf vs Δx showing the convergence study for the sixth order Panel Integrator

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
```

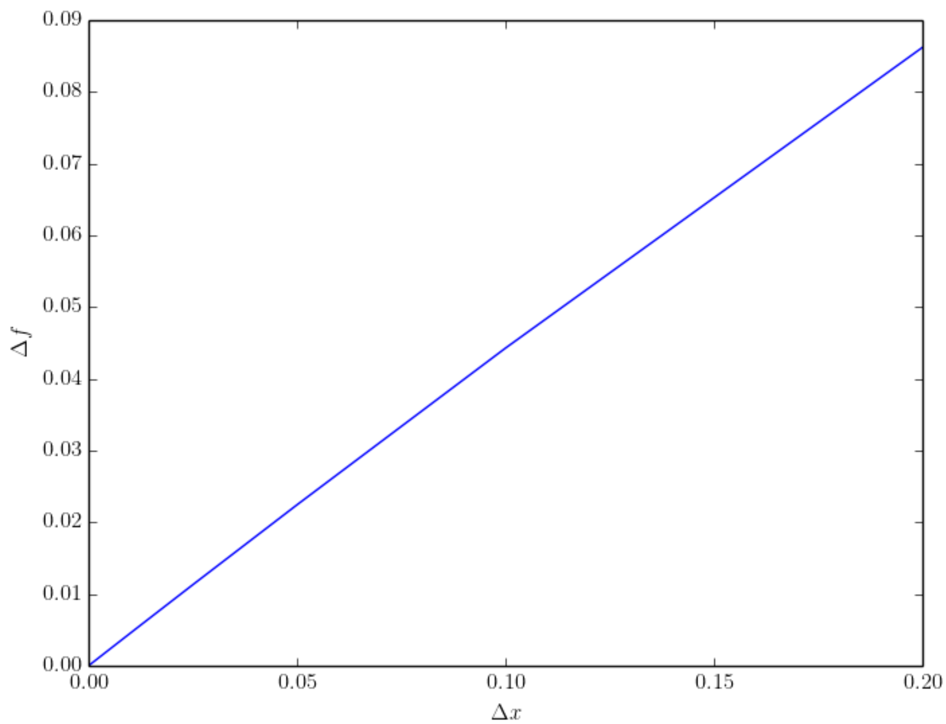

Figure 2: Δf vs Δx for the convergence study of the integral in problem 3 using linear order panel integrator

```
Created on Mon Oct  1 02:31:29 2018

@author: alfred_mac
"""


from __future__ import unicode_literals
import numpy as np
import matplotlib
matplotlib.rcParams['text.usetex'] = True
matplotlib.rcParams['text.latex.unicode'] = True
import matplotlib.pyplot as plt

def PanelIntegrator1( n, a, b):  # integrate from a to b with n panels
    dx = (b-a)/n
    eta = 0.5*np.sqrt(0.6)
```

```
    import f                        # a module that defines f(x)
    x = np.linspace(a,b,n+1)
    Ik = (dx/18)*(5*f.f(x+(dx/2)-(dx*eta))+8*f.f(x+(dx/2))+
    5*f.f(x+(dx/2)+(dx*eta))) # evaluate f at the point x+dx/2 or x{k+0.5} as dx
    I = sum(Ik)
    return I                        # estimate of the integral

# Convergence study for Panel Integrator
N = 10
INTG = np.linspace(1,2,N)*0
DIFF = INTG*0

for i in range(N):
    INTG[i] = PanelIntegrator1(1e1*pow(2,i),0,2)
    DIFF[i] = INTG[i] - 1 + np.cos(2)


X = 2/(1e1*pow(2,np.arange(N)))
plt.plot(X,DIFF)

plt.xlabel(r'$\Delta{x}$')
plt.ylabel(r'$\Delta{f}$')

plt.show()
```

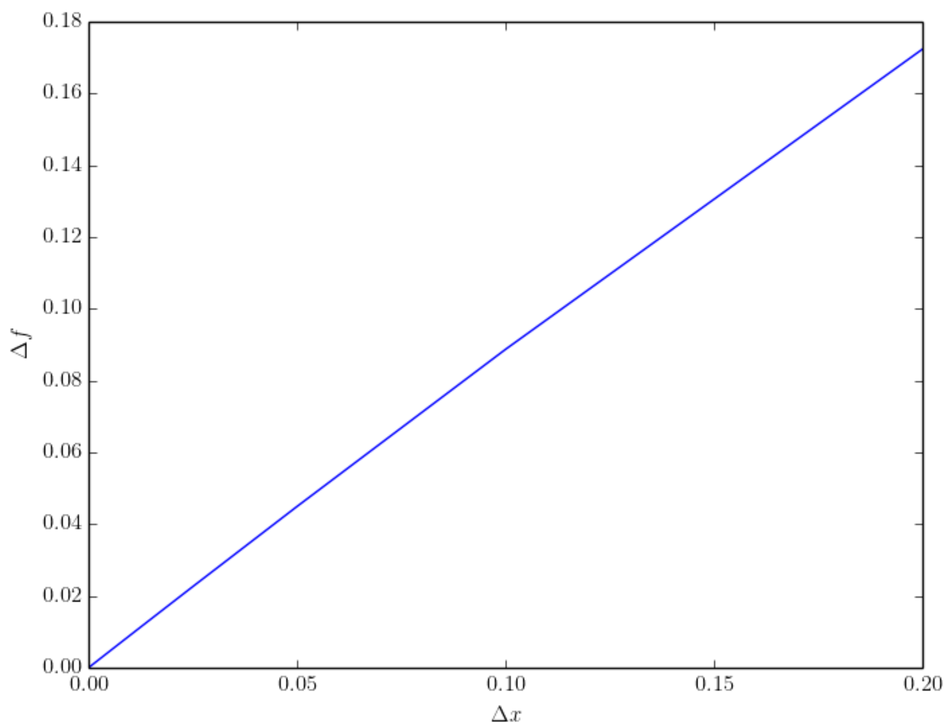

Figure 3: Δf vs Δx for the convergence study of the integral in problem 3 using sixth order panel integrator