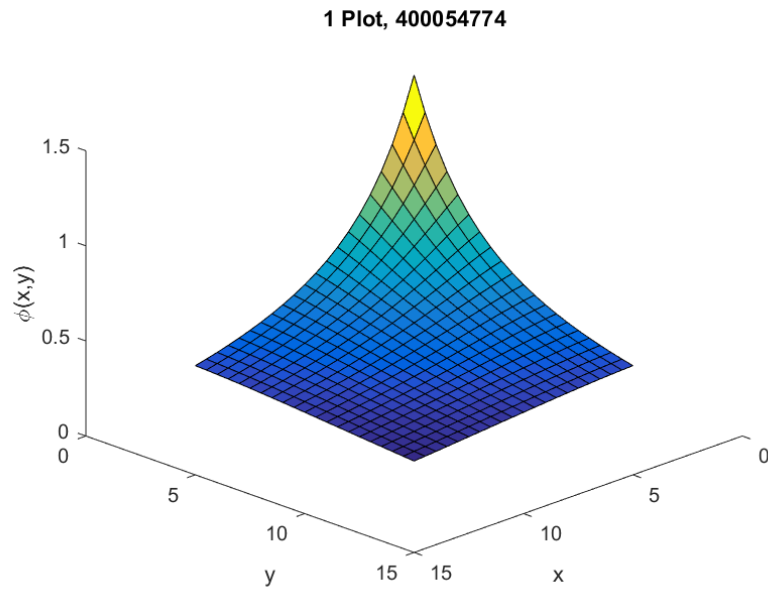


SFWR 4X03 Assignment 3

1.



Phi.m Code:

```
function f = phi(xhat,yhat)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% parameters %%%%%%%%%%%%%%

a = -1;

b = 1;

tol = 0.00001;

level = 0;

max = 100;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% function %%%%%%%%%%%%%%

fun = @(x,y) 1./sqrt((xhat-x).^2 + (yhat-y).^2);

inner_integral = @(y) adsimpson(@(x) fun(x,y),a,b,tol,level,max); %% do innerloop with respect to x,
make function of y

f = adsimpson(inner_integral,a,b,tol,level,max); %% do with respect to y

end
```

2 a)

midpoint.m code:

```
function [Q] = midpoint(f,a,b,n)
```

```
h = (b-a)/n;
```

```
i = 1:n; %% summation bounds
```

```
fi = f(a+(i-0.5).*h); %%vectorized compute
```

```
Q = h*sum(fi); %%add
```

End

Trapezoid.m code:

```
function [Q] = trapezoid(f,a,b,n)
```

```
h = (b-a)/n;
```

```
i = 1:n-1;
```

```
ti = a+ i.*h; %%make ti
```

```
Q = (h/2)*(f(a) + f(b)) + h*sum(f(ti)); %%sum and eq
```

end

Simpson.m code:

```
function [Q] = simpson(f,a,b,n)
```

```
h = (b-a)/n;
```

```
i = 1:n/2;
```

```
i1 = 1:n/2 -1;
```

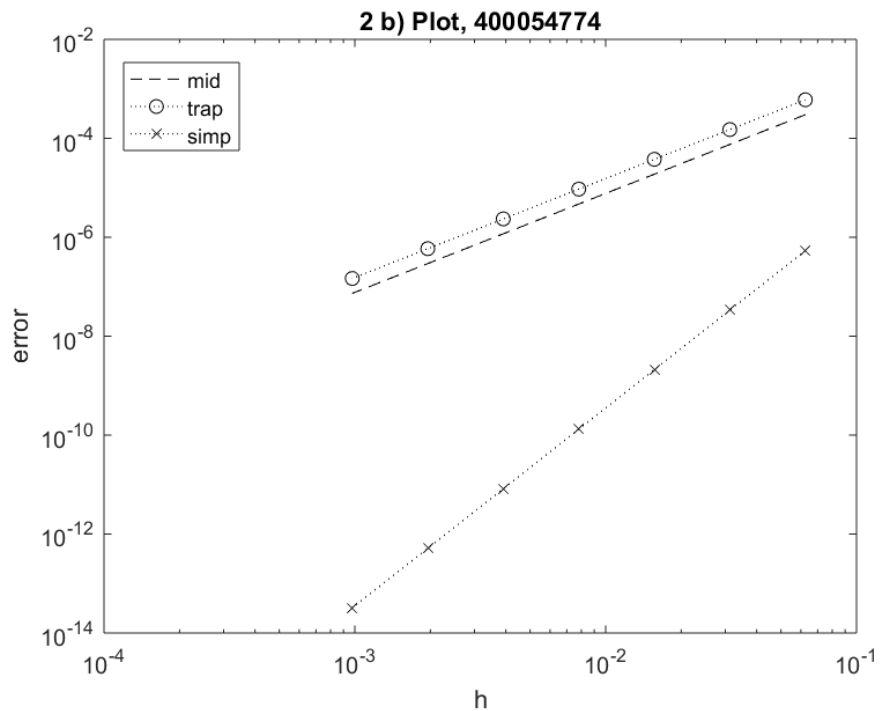
```
t2i = a+ (2.*i1).*h;
```

```
t2iMinus1 = a+(2.*i -1).*h;
```

$$Q = (h/3) * (f(a) + 2 * \sum(f(t_{2i})) + 4 * \sum(f(t_{2i-1})) + f(b));$$

end

2b)



Trapezoid is the worst method, and midpoint is not much better. Simpson is significantly better than the both. All methods are better as we use more intervals n .

2 c) we know: $err = ch^p$

So: $\log(err) = \log(c) + p * \log(h)$

In the form of $y = mx + b$ where: $y = \log(err)$, and $x = p * \log(h)$

This means p is the slope = $\frac{\log(err_2) - \log(err_1)}{\log(h_2) - \log(h_1)}$

Find p then sub p and a point to find c .

Findconstants.m Code:

```
function [c,p] = findconstants(rule,f,a,b)

actual = integral(f,a,b);

n1 = 2^4; %%use two points
n2 = 2^10;

h1 = 1/n1;
h2 = 1/n2;

err1 = abs(rule(f,a,b,n1) - actual);%% get error
err2 = abs(rule(f,a,b,n2) - actual);

p = log(err2/err1)/log(h2/h1); %% slope is p

c = err2/(h2^p);    %%sub p in to find c

end
```

Output on given equation:

RULE	C	P
midpoint	0.076919	1.999906
trapezoid	0.153881	1.999946
Simpson	0.034631	3.995839

3.

Used findconstant.m and solved for error using $\text{err} = \text{ch}^p$

$N = 18599$

$\text{Err} = 9.9995\text{e-}11$

4 a)

Planets	a	b	c	d	e
Jupiter	-1.1854e+00	2.2029e-02	-4.9504e-01	-1.4505e-01	2.6982e+01
Saturn	-1.1667e+00	3.5963e-02	1.1673e-01	-1.0899e+00	9.0382e+01
Uranus	-1.1941e+00	1.1627e-02	1.8271e+00	-2.5926e-01	3.6727e+02
Neptune	-1.1671e+00	2.0704e-02	-3.9269e-01	-4.2316e-01	9.0381e+02
Pluto	-1.0033e+00	2.3883e-01	1.1847e+01	1.2717e+01	1.2907e+03

4b)

