Assignment 3
Problem 1

```
function f = phi(xhat,yhat)
    a = -1;
    b = 1;
    tol = 1e-5;
    func = @(x,y) 1./sqrt((xhat-x).^2+(yhat-y).^2);

    Y = @(y) adsimpson(@(x) func(x,y),a,b,tol,0,100);
    f = adsimpson(Y,a,b,tol,0,100);
end
```

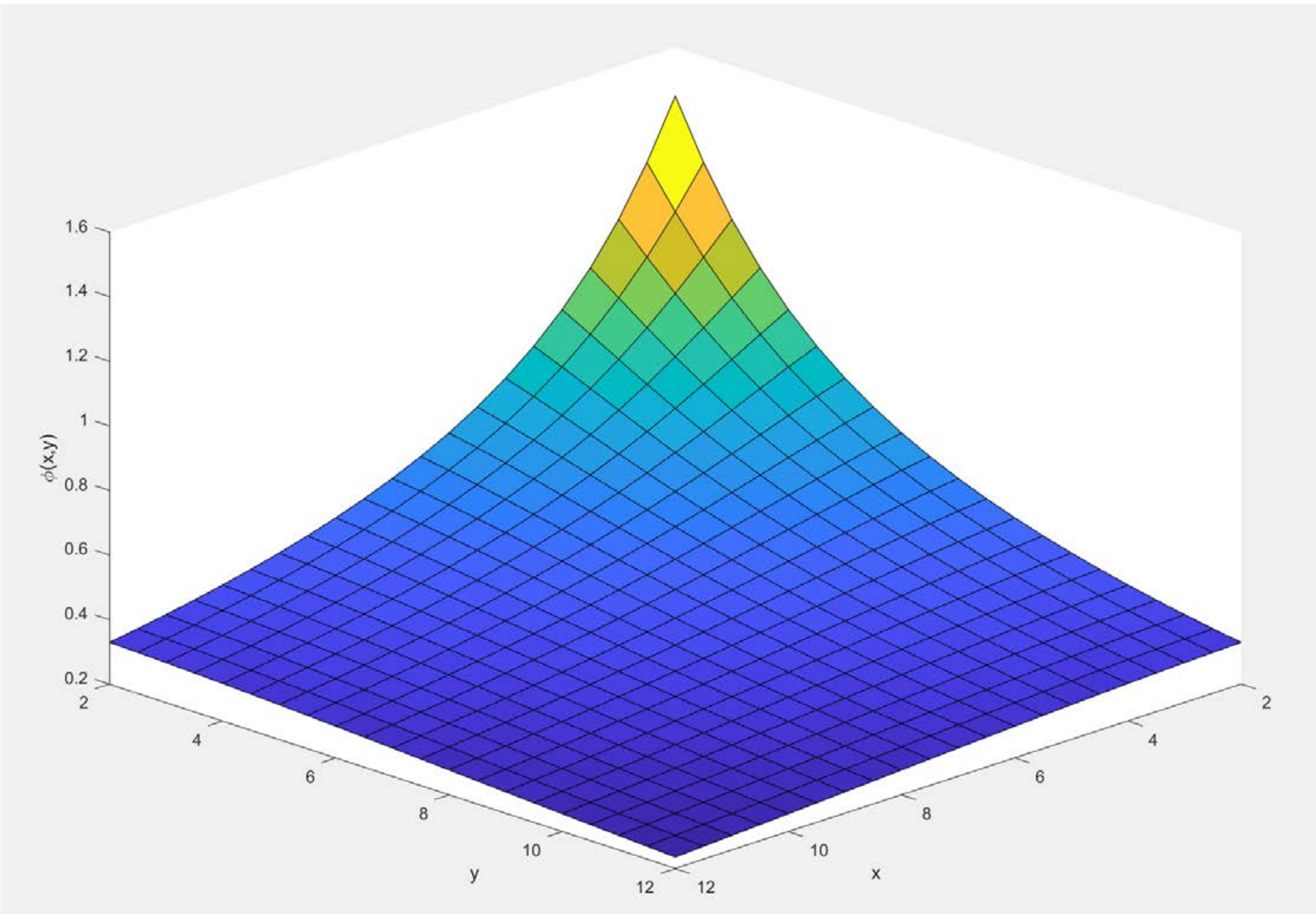

Figure 1: plot of `phi`

Problem 2

a) `midpoint.m`

b)

```
function [Q] = midpoint(f,a,b,n)
    format long
    h = (b-a)/n;
    if n == 0;  Q = 0;
    else;       Q = f(a+h/2)*h + midpoint(f,a+h,b,n-1); %using recursion
    end
end
```

`trapezoid.m`

```
function [Q] = trapezoid(f,a,b,n)
    format long
    h = (b-a)/n;
    if n == 0;  Q = 0;
    else;       Q = (f(a)+f(a+h)).*h./2 + trapezoid(f,a+h,b,n-1); %using recursion
    end
end
```

simpson.m

```
function [Q] = simpson(f,a,b,n)
    format long
    h = (b-a)/(2*n);
    if n == 0;  Q = 0;
    else;       Q = (f(a)+4*f(a+h)+f(a+2*h))*h/3 + simpson(f,a+2*h,b,n-1); %using recursion
    end
end
```

b)

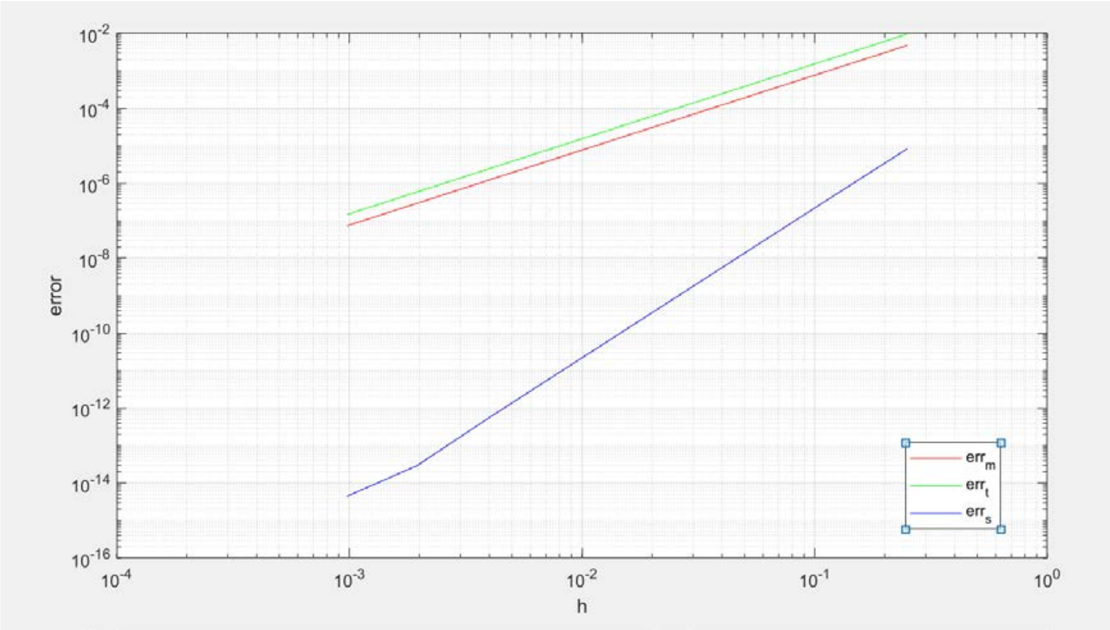

Figure 2: error plot for three integration methods

All integration methods increase in precision with the decrease in h but simpson's method seems to be the most accurate out of all methods. Both error from midpoint and trapezoid method decrease in error at the same rate but simpson's method decreases in error the fastest.

c)

The process behind the method is by comparing the values estimated by either `midpoint`, `trapezoid` or `simpson` with an accurate value, in this case the output of `integral`, `error` can be calculated. After `error` is calculated and because `h = (b-a)/n`, `c` and `p` can be calculated by calculating 2 arbitrary sets of `h` and `error` and solving for `c` and `p`

```
function [c,p] = findconstants(rule,f,a,b)
    %assume this is exact
    Q_true = integral(f,a,b);

    n = 5;
    h_5 = (b-a)/n;
    err_5 = Q_true - rule(f,a,b,n); % error for n = 5

    n = 10;
    h_10 = (b-a)/n;
    err_10 = Q_true - rule(f,a,b,n); % error for n = 10

    % let c of each error eqn equal to each other, isolate for p
    p = log(err_10/err_5)/log(h_10/h_5);
    c = err_5/(h_5^p);
end
```

Output:

```
midpoint:     c=0.076143          p=1.995746
trapezoid:    c=-0.152991         p=1.997566
simpson:      c=-0.002082         p=3.976895
```

Problem 3

```
n = 18580
err = 9.997591643440273e-11
```

Problem 4

a)

Table 1: coefficients for each planet orbit

Planet	a	b	c	d	e
Jupiter	−1.185397	0.022029	−0.495039	−0.145054	26.982216
Saturn	−1.166745	0.035963	0.116729	−1.089852	90.381602
Uranus	−1.194134	0.011627	1.827051	−0.259256	367.268144
Neptune	−1.167128	0.020704	−0.392687	−0.423158	903.808671
Pluto	−1.003337	0.238833	11.847098	12.717063	1290.679928

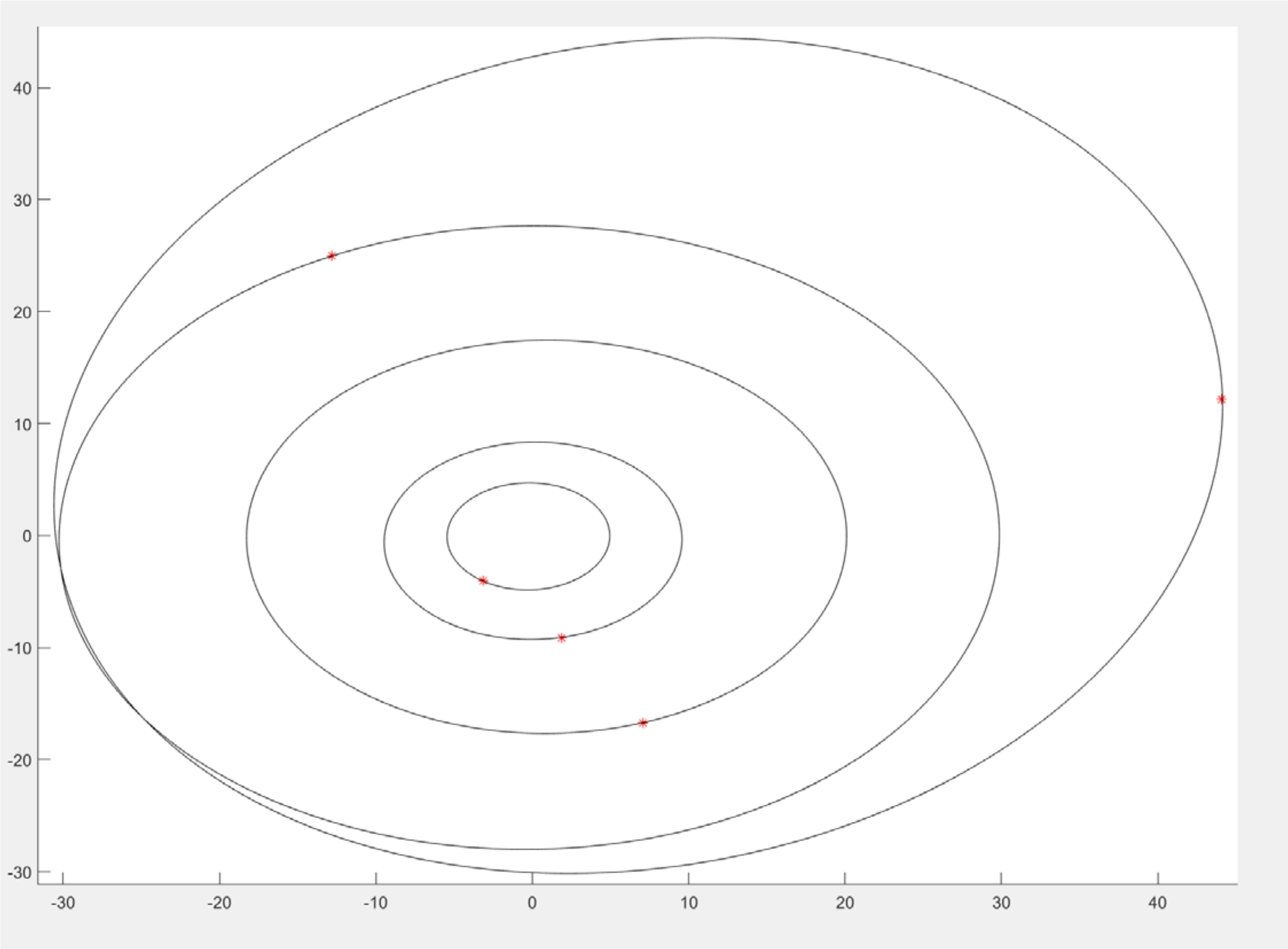

Figure 3: orbit path for each planet

To calculate the location of each planet at a certain time. the respective orbit functions must be parameterized so that it describes the x and y position of the planet at a given time. using this new function, the position of each planet can be calculated.