```
function [du] = rocketSimODE_Real(t,y)
% -> means "integrates or relates to"
%y(1) = x position
%y(2) = x velocity
%y(3) = y position
%y(4) = y velocity

%du(1) = x velocity     -> x position y(1)
%du(2) = x acceleration -> x velocity y(2)
%du(3) = y velocity     -> y position y(3)
%du(4) = y acceleration -> y velocity y(4)

%% Initial Values
global g0 r0 frontArea;
alt = y(3);
du=zeros(4,1);
du(1) = y(2); %x velocity -> x position y(1)
du(3) = y(4); %y velocity -> y position y(3)

%% Mass and Thrust Throughout Burn
m = valueAt(t, 'mass'); %kg total weight
thrust = valueAt(t, 'thrust');

%% Angle and Gravity Change
beta = asind(y(2)/(sqrt(y(2)^2 + y(4)^2)));
g = g0*((r0/(r0+y(3)))^2);

%% Drag Calulcation Per Iteration
area_ref = frontArea; %m^2
rho = findrho(alt); %rho = f(atl)
temp = findTemp(alt);
velVec = sqrt(y(2)^2 + y(4)^2);
Mach = valueOfMach(velVec, temp);
Cd = findCd(Mach, t);
drag = 0.5*rho*Cd*(velVec*abs(velVec))*area_ref;
%drag = 0;

%% Print Statements
%fprintf('t: %2.1f  velVec: %2.1f  temp: %2.1f  mach: %2.1f  drag: %2.1f  beta: %2.1f ↙
m: %2.1f\n',t,velVec, temp, Mach, drag,  beta, m);

%% Acceleration Changes - Velocity Propogation
du(2) = (thrust*sind(beta))/m - (drag*sind(beta))/m;      %x-accel
du(4) = (thrust*cosd(beta))/m - (drag*cosd(beta))/m - g;  %y-accel
end
```