Brief summary of this function.

Detailed explanation of this function.

```
function [] = B747100Values()
```

Declare globals, constants, and conversions

```
global E1 E3 Constants Conv theta0 StabilityFrame A

Constants.g = 9.81; % [m/s^2]
% Constants.XDe = -3.818*10^-6; % non-dimensional elevator derivative for X
% Constants.ZDe = -0.3648; % Non-dimensional elevator derivatice for Z
% Constants.MDe = -1.444; % Non-dimensional elevator derivative for M

Conv.FtToM          = 0.3048; % ft to m
Conv.LbToN          = 4.44822; % lb to N
Conv.SlugFt2ToKgM2  = 1.35581795; % slug*ft^2 to kg*m^2
Conv.DegToRad       = pi / 180; % deg to rad
```

Values from table E.1, for Case 2.

```
E1.Altitude = 20000         * Conv.FtToM;         % Altitude of 747 in meters
E1.M        = 0.5;                                % Mach number
E1.Velocity = 518           * Conv.FtToM;         % Velocity in [m/s]
E1.Weight   = 6.366 * 10^5  * Conv.LbToN;         % Weight of the 747 in Newtons
E1.Ix       = 1.82  * 10^7  * Conv.SlugFt2ToKgM2; % Moment of Inertia about x axis
E1.Iy       = 3.31  * 10^7  * Conv.SlugFt2ToKgM2; % Moment of Inertia about y axis
E1.Iz       = 4.97  * 10^7  * Conv.SlugFt2ToKgM2; % Moment of Inertia about z axis
E1.Izx      = 9.70  * 10^5  * Conv.SlugFt2ToKgM2; % Product of Inertia about z and x axes
E1.Xi       = -6.8          * Conv.DegToRad;      % Angle between stability and body frames
E1.CD       = 0.040;                              % Coefficient of Drag
E1.m        = E1.Weight     / Constants.g;        % Mass of the 747
```

Values from table E.3, longitudinal, all converted to SI units.

```
E3.X.u      = -4.883 * 10^1 * Conv.LbToN / Conv.FtToM;  % [N * s / m]
E3.X.w      = 1.546  * 10^3 * Conv.LbToN / Conv.FtToM;  % [N * s / m]
E3.X.q      = 0             * Conv.LbToN;               % [N * s / rad]
E3.X.wDot   = 0             * Conv.LbToN / Conv.FtToM;  % [N * s^2 / m]
E3.X.deltae = 3.994  * 10^4 * Conv.LbToN;               % [N / rad]

E3.Z.u      = -1.342 * 10^3 * Conv.LbToN / Conv.FtToM;  % [N * s / m]
E3.Z.w      = -8.561 * 10^3 * Conv.LbToN / Conv.FtToM;  % [N * s / m]
E3.Z.q      = -1.263 * 10^5 * Conv.LbToN;               % [N * s / rad]
E3.Z.wDot   = 3.104  * 10^2 * Conv.LbToN / Conv.FtToM;  % [N * s^2 / m]
E3.Z.deltae = -3.341 * 10^5 * Conv.LbToN;               % [N / rad]

E3.M.u      = 8.176  * 10^3 * Conv.LbToN;               % [N * s]
E3.M.w      = -5.627 * 10^4 * Conv.LbToN;               % [N * s]
E3.M.q      = -1.394 * 10^7 * Conv.LbToN * Conv.FtToM;  % [N * m * s / rad]
E3.M.wDot   = -4.138 * 10^3 * Conv.LbToN;               % [N * s^2]
E3.M.deltae = -3.608 * 10^7 * Conv.LbToN * Conv.FtToM;  % [N * m / rad]
```

Conversions to stability frame using coordinate rotations in eqn (B.12, 6).

```
StabilityFrame.X.u = E3.X.u * cos(E1.Xi)^2 - (E3.X.w + E3.Z.u) ...
    * sin(E1.Xi) * cos(E1.Xi) + E3.Z.w * sin(E1.Xi)^2;
StabilityFrame.X.w = E3.X.w * cos(E1.Xi)^2 + (E3.X.u - E3.Z.w) ...
    * sin(E1.Xi) * cos(E1.Xi) - E3.Z.u * sin(E1.Xi)^2;
StabilityFrame.X.q = E3.X.q * cos(E1.Xi) - E3.Z.q * sin(E1.Xi);
StabilityFrame.X.uDot = E3.Z.wDot * sin(E1.Xi)^2;
StabilityFrame.X.wDot = -E3.Z.wDot * sin(E1.Xi) * cos(E1.Xi);

StabilityFrame.Z.u = E3.Z.u * cos(E1.Xi)^2 - (E3.Z.w - E3.X.u) ...
    * sin(E1.Xi) * cos(E1.Xi) - E3.X.w * sin(E1.Xi)^2;
StabilityFrame.Z.w = E3.Z.w * cos(E1.Xi)^2 + (E3.Z.u + E3.X.w) ...
    * sin(E1.Xi) * cos(E1.Xi) + E3.X.u * sin(E1.Xi)^2;
StabilityFrame.Z.q = E3.Z.q * cos(E1.Xi) + E3.X.q * sin(E1.Xi);
StabilityFrame.Z.uDot = -E3.Z.wDot * sin(E1.Xi) * cos(E1.Xi);
StabilityFrame.Z.wDot = E3.Z.wDot * cos(E1.Xi)^2;

StabilityFrame.M.u = E3.M.u * cos(E1.Xi) - E3.M.w * sin(E1.Xi);
StabilityFrame.M.w = E3.M.w * cos(E1.Xi) + E3.M.u * sin(E1.Xi);
StabilityFrame.M.q = E3.M.q;
StabilityFrame.M.uDot = -E3.M.wDot * sin(E1.Xi);
StabilityFrame.M.wDot = E3.M.wDot * cos(E1.Xi);
```

Question 3

Using the Stability frames constructed above, the 4x4 A matrix from eqn (4.9,18) for the linearized longitudinal dynamics is created.

```
A = zeros(4,4);
theta0 = 0;

A(1,1) = StabilityFrame.X.u / E1.m;
A(1,2) = StabilityFrame.X.w / E1.m;
A(1,3) = 0;
A(1,4) = -Constants.g * cos(theta0);

A(2,1) = StabilityFrame.Z.u / (E1.m - StabilityFrame.Z.wDot);
A(2,2) = StabilityFrame.Z.w / (E1.m - StabilityFrame.Z.wDot);
A(2,3) = (StabilityFrame.Z.q + E1.m * E1.Velocity) / (E1.m - StabilityFrame.Z.wDot);
A(2,4) = (-E1.Weight * sin(theta0)) / (E1.m - StabilityFrame.Z.wDot);

A(3,1) = E1.Iy ^ (-1) * (StabilityFrame.M.u + ((StabilityFrame.M.wDot * StabilityFrame.Z.u) / (
A(3,2) = E1.Iy ^ (-1) * (StabilityFrame.M.w + ((StabilityFrame.M.wDot * StabilityFrame.Z.w) / (
A(3,3) = E1.Iy ^ (-1) * (StabilityFrame.M.q + (StabilityFrame.M.wDot * (StabilityFrame.Z.q + E1
    / (E1.m - StabilityFrame.Z.wDot)));
A(3,4) = -(StabilityFrame.M.wDot * E1.Weight * sin(theta0)) / (E1.Iy * (E1.m - StabilityFrame.Z

A(4,1) = 0;
A(4,2) = 0;
A(4,3) = 1;
A(4,4) = 0;
```

```
end
```