Euler' s Method

```
g[y_, t_] = 3 + t - y
```

```
3 + t - y
```

```
tzero = 0
yzero = 1
```

```
0
```

```
1
```

Here is the Euler method applied with stepsize 0.1

```
h = 0.1
```

```
0.1
```

```
For[j = 1; soln = {{tzero, yzero}}, j < 5, j ++, soln =
  Union[{{tzero + j h, soln[[j]][[2]] + g[soln[[j]][[2]], soln[[j]][[1]]] h}}, soln]]
```

```
approx1 = soln
```

```
{{0, 1}, {0.1, 1.2}, {0.2, 1.39}, {0.3, 1.571}, {0.4, 1.7439}}
```

Here is a way to figure out how arrays work :

```
testarray = {{1, 2}, {3, 4}, {5, 6}}
```

```
{{1, 2}, {3, 4}, {5, 6}}
```

```
testarray[[1]]
testarray[[2]][[2]]
```

```
{1, 2}
```

```
4
```

Here is the Euler method applied with stepsize 0.05

```
h = 0.05
```

```
0.05
```

```
For[j = 1; soln = {{tzero, yzero}}, j < 9, j ++, soln =
  Union[{{tzero + j h, soln[[j]][[2]] + g[soln[[j]][[2]], soln[[j]][[1]]] h}}, soln]]
```

```
approx2 = soln
```

```
{{0, 1}, {0.05, 1.1}, {0.1, 1.1975}, {0.15, 1.29263}, {0.2, 1.38549},
 {0.25, 1.47622}, {0.3, 1.56491}, {0.35, 1.65166}, {0.4, 1.73658}}
```

Here is the Euler method applied with stepsize 0.025

```
h = 0.025
```

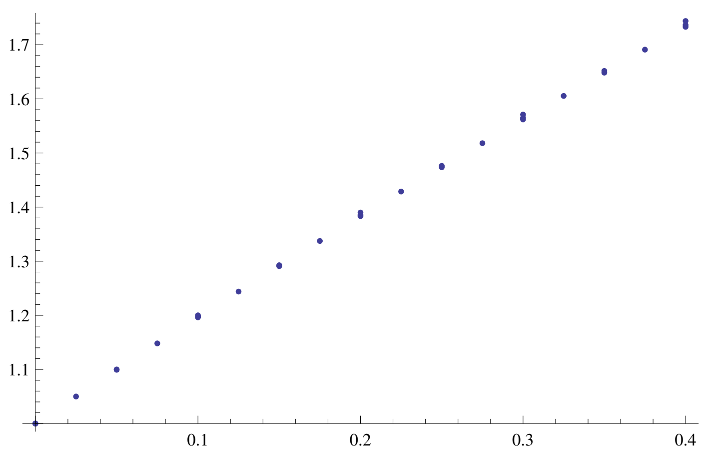
```
0.025
```

```
For[j = 1; soln = {{tzero, yzero}}, j < 17, j ++, soln =
  Union[{{tzero + j h, soln[[j]][[2]] + g[soln[[j]][[2]], soln[[j]][[1]]] h}}, soln]]
```

```
approx3 = soln
```

```
{{0, 1}, {0.025, 1.05}, {0.05, 1.09938}, {0.075, 1.14814}, {0.1, 1.19631},
 {0.125, 1.2439}, {0.15, 1.29093}, {0.175, 1.33741}, {0.2, 1.38335},
 {0.225, 1.42876}, {0.25, 1.47367}, {0.275, 1.51808}, {0.3, 1.562},
 {0.325, 1.60545}, {0.35, 1.64844}, {0.375, 1.69098}, {0.4, 1.73308}}
```

```
Show[ListPlot[approx1], ListPlot[approx2], ListPlot[approx3]]
```



The "actual" solution

```
nsoln = NDSolve[{ny'[t] == 3 + t - ny[t], ny[0] == 1}, ny, {t, 0, 0.4}]
```

```
{{ny → InterpolatingFunction[{{0., 0.4}}, <>]}}
```

```
Show[ListPlot[approx1], Plot[ny[t] /. nsoln, {t, 0, 0.4}]]
```