

Euler method functions to solve

Developed by Ibrahim Issah University of Eastern Finland Msc. Photonics

INITIALIZE

```
clear
close all
clc
```

ODE FUNCTION

```
dy = @(t,y) t*cos(sqrt(y)); %% Using functions
```

DASHBOARD

```
ta = 0; %% starting point
tn = 10; %% ending point
h = .05; %% stepsize
y = 1; %% initial condition

storery = []; %% for storing y values

fprintf('t_values \t\t Euler method(values) \t\n');
```

```
t_values Euler method(values)
```

```
for t = ta:h:tn-h
    y = y+dy(t,y)*h;
    fprintf('%f \t\t %f \n', t, y);
    t = t+h; %% increment with stepsize
    storery = [(storery), y];
end
```

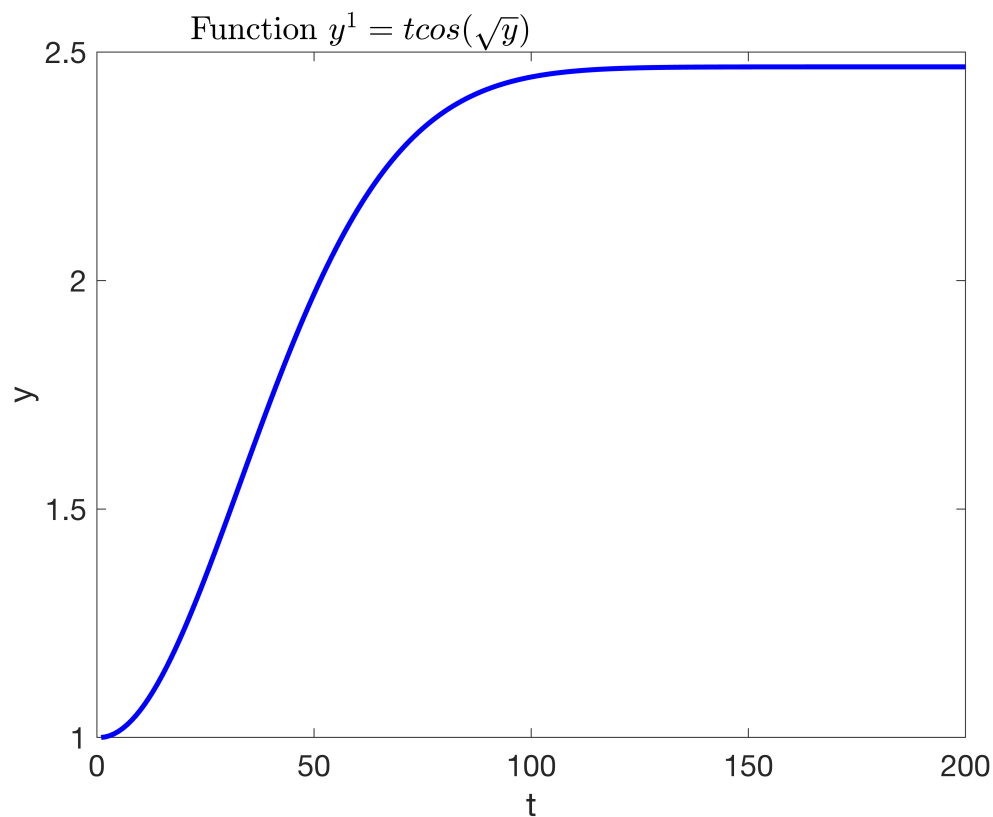
0.000000	1.000000
0.050000	1.001351
0.100000	1.004049
0.150000	1.008089
0.200000	1.013458
0.250000	1.020141
0.300000	1.028119
0.350000	1.037367
0.400000	1.047860
0.450000	1.059566
0.500000	1.072450
0.550000	1.086476
0.600000	1.101602
0.650000	1.117785

0.700000	1.134979
0.750000	1.153136
0.800000	1.172206
0.850000	1.192137
0.900000	1.212875
0.950000	1.234366
1.000000	1.256553
1.050000	1.279381
1.100000	1.302793
1.150000	1.326731
1.200000	1.351140
1.250000	1.375962
1.300000	1.401141
1.350000	1.426622
1.400000	1.452352
1.450000	1.478275
1.500000	1.504341
1.550000	1.530499
1.600000	1.556699
1.650000	1.582895
1.700000	1.609041
1.750000	1.635092
1.800000	1.661008
1.850000	1.686748
1.900000	1.712275
1.950000	1.737553
2.000000	1.762549
2.050000	1.787230
2.100000	1.811569
2.150000	1.835537
2.200000	1.859110
2.250000	1.882265
2.300000	1.904982
2.350000	1.927240
2.400000	1.949024
2.450000	1.970319
2.500000	1.991111
2.550000	2.011390
2.600000	2.031146
2.650000	2.050372
2.700000	2.069061
2.750000	2.087209
2.800000	2.104814
2.850000	2.121872
2.900000	2.138386
2.950000	2.154354
3.000000	2.169781
3.050000	2.184668
3.100000	2.199021
3.150000	2.212846
3.200000	2.226147
3.250000	2.238934
3.300000	2.251213
3.350000	2.262994
3.400000	2.274286
3.450000	2.285099
3.500000	2.295443
3.550000	2.305329
3.600000	2.314768
3.650000	2.323773
3.700000	2.332354
3.750000	2.340525
3.800000	2.348297
3.850000	2.355684
3.900000	2.362697

3.950000	2.369349
4.000000	2.375654
4.050000	2.381623
4.100000	2.387269
4.150000	2.392605
4.200000	2.397643
4.250000	2.402394
4.300000	2.406873
4.350000	2.411089
4.400000	2.415055
4.450000	2.418782
4.500000	2.422281
4.550000	2.425564
4.600000	2.428640
4.650000	2.431520
4.700000	2.434213
4.750000	2.436731
4.800000	2.439081
4.850000	2.441273
4.900000	2.443316
4.950000	2.445218
5.000000	2.446988
5.050000	2.448632
5.100000	2.450158
5.150000	2.451574
5.200000	2.452886
5.250000	2.454100
5.300000	2.455224
5.350000	2.456262
5.400000	2.457220
5.450000	2.458104
5.500000	2.458919
5.550000	2.459669
5.600000	2.460359
5.650000	2.460992
5.700000	2.461574
5.750000	2.462108
5.800000	2.462597
5.850000	2.463044
5.900000	2.463453
5.950000	2.463827
6.000000	2.464169
6.050000	2.464480
6.100000	2.464764
6.150000	2.465022
6.200000	2.465257
6.250000	2.465470
6.300000	2.465664
6.350000	2.465839
6.400000	2.465999
6.450000	2.466143
6.500000	2.466273
6.550000	2.466390
6.600000	2.466497
6.650000	2.466592
6.700000	2.466679
6.750000	2.466756
6.800000	2.466826
6.850000	2.466889
6.900000	2.466945
6.950000	2.466995
7.000000	2.467041
7.050000	2.467081
7.100000	2.467117
7.150000	2.467150

7.200000	2.467178
7.250000	2.467204
7.300000	2.467227
7.350000	2.467247
7.400000	2.467265
7.450000	2.467282
7.500000	2.467296
7.550000	2.467308
7.600000	2.467320
7.650000	2.467330
7.700000	2.467338
7.750000	2.467346
7.800000	2.467353
7.850000	2.467359
7.900000	2.467364
7.950000	2.467369
8.000000	2.467373
8.050000	2.467377
8.100000	2.467380
8.150000	2.467383
8.200000	2.467385
8.250000	2.467387
8.300000	2.467389
8.350000	2.467391
8.400000	2.467392
8.450000	2.467393
8.500000	2.467394
8.550000	2.467395
8.600000	2.467396
8.650000	2.467397
8.700000	2.467397
8.750000	2.467398
8.800000	2.467398
8.850000	2.467399
8.900000	2.467399
8.950000	2.467399
9.000000	2.467400
9.050000	2.467400
9.100000	2.467400
9.150000	2.467400
9.200000	2.467400
9.250000	2.467400
9.300000	2.467401
9.350000	2.467401
9.400000	2.467401
9.450000	2.467401
9.500000	2.467401
9.550000	2.467401
9.600000	2.467401
9.650000	2.467401
9.700000	2.467401
9.750000	2.467401
9.800000	2.467401
9.850000	2.467401
9.900000	2.467401
9.950000	2.467401

```
figure (1); plot(storery,'b-', 'LineWidth', 2.5);
xlabel('t');
ylabel('y');
title('Function  $y^{[1]} = t \cos(\sqrt{y})$ ', 'Interpreter', 'Latex', 'Rotation', 0, ...
      'HorizontalAlignment', 'right');
```



From taylor series

$$y(x+h) = y(x) + y'(x)h;$$

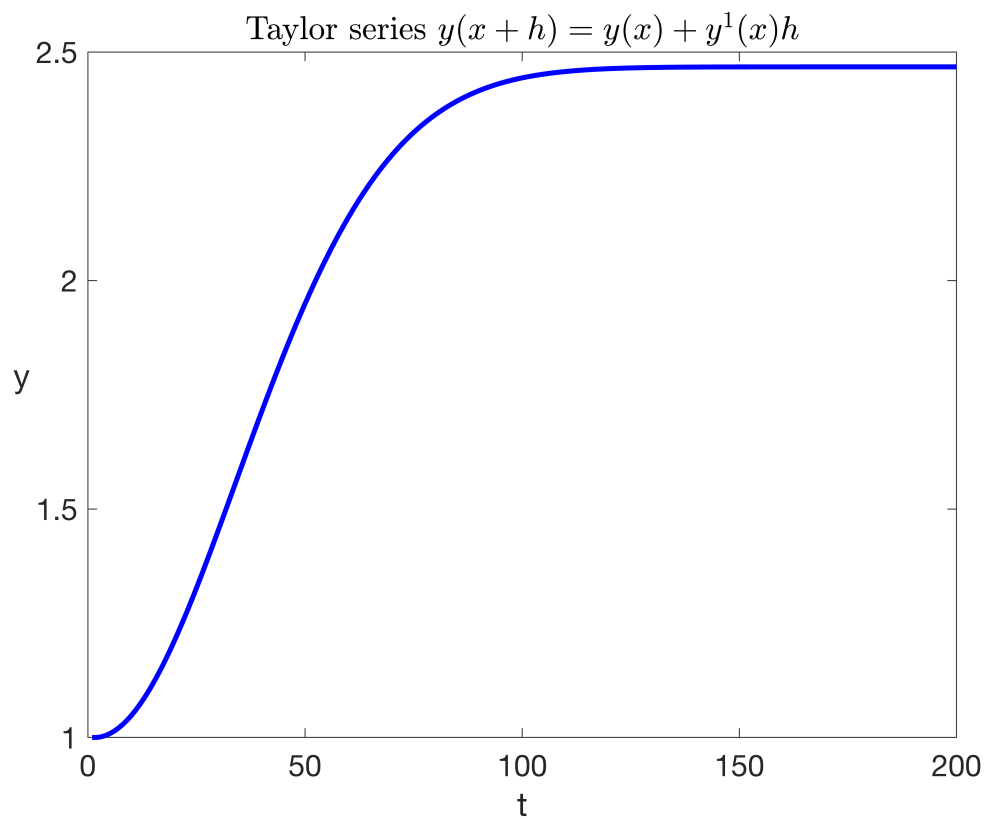
DASHBOARD 2

```
h = 0.05;           %% step size
y = zeros(1, 200);  %% pre-processing
y(1) = 1;           %% initial condition
t(1) = 0;           %% starting point of t
```

TAYLOR FORMULATION

```
for iter = 1:199
    t(iter+1) = t(iter) + h;
    y(iter+1) = y(iter) + h.*(t(iter).*cos(sqrt(y(iter))));
end

figure(2); plot(y, 'b-', 'LineWidth', 2.5);
xlabel('t');
ylabel('y', 'Rotation', 0, ...
        'HorizontalAlignment', 'right');
title('Taylor series  $y(x+h) = y(x) + y^{\{1\}}(x)h$ ', 'Interpreter', 'Latex');
```



Symbolic integration

```
syms y(t) t;
expr = diff(y,t) == t*cos(sqrt(y));
cond = y(0) == 1;
S = dsolve(expr, cond);
```

Warning: Unable to find explicit solution. Returning implicit solution instead.

```
disp(S);
```

$$\text{solve}\left(-4 \sqrt{y} \operatorname{atan}\left(e^{\sqrt{y}i}\right) i + 2 \operatorname{polylog}\left(2, -e^{\sqrt{y}i}\right) i - 2 \operatorname{polylog}\left(2, e^{\sqrt{y}i}\right) i = \frac{t^2}{2} - 4 \operatorname{atan}(e^i) i + 2 \operatorname{polylog}($$
