

```

%Para a Carga no Capacitor
%Teorico
clear;
close all;
%Digite o valor de tau calculado na próxima linha e remova o % do inicio da mesma
%tau= xx;
polo=1/tau;
%Digite o valor de VCmax calculado na próxima linha e remova o % do inicio da mesma
%Vc_max= xx ;
t=linspace(0,25,1000);
t_ex=[0 1 2 3 4 6 8 10 12 14 16];
vc_t=Vc_max*(1-exp(-t/tau));
Vc_tau=Vc_max*(1-exp(-1));
p1=plot(t,vc_t);hold on;
%Experimental
%Digite o valor de tau experimental na próxima linha e remova o % do inicio da mesma
%tau_ex= xx ;
%Digite o valor de VCmax calculado na próxima linha e remova o % do inicio da mesma
%Vc_max_ex= xx ;
vc_ex_t=Vc_max_ex*(1-exp(-t/tau_ex));
vc_ex_k=Vc_max_ex*(1-exp(-t_ex/tau_ex));
plot(t,vc_ex_t)
p2=plot(t_ex,vc_ex_k, '*');
%Perfumaria no Gráfico
ax = gca;
ax.Box = 'off';
ax.YAxisLocation = 'origin';
ax.XAxisLocation = 'origin';
ax.FontName='times';
ax.FontSize=12;
ax.TickLength=[0 0];
xlim([0 25]);
ylim([0 16]);
xticks([0 tau 25]);
xticklabels({'0', ['\it \tau = ', num2str(tau), ' s'], '\it t(s)'});
yticks([0 Vc_tau Vc_max 16]);
yticklabels({'0', ['\it {v_C(\tau)} = ', num2str(Vc_tau)], '\it {V_{C_{max}}} = 14.2', '\it {v_C(t)'});
plot([0 tau],[Vc_tau,Vc_tau], ':');
plot([tau,tau],[0,Vc_tau], ':');
plot([0 25],[Vc_max Vc_max], ':')
%xlabel('Carga no Capacitor');
arrowsPlot(ax,1,1);
legend([p1 p2], 'Teórico', 'Experimental', 'Location', 'best')

```

```

%Para a Descarga no Capacitor

```

```

clear all;
close all;
%Teorico
%Digite o valor de tau calculado na próxima linha e remova o % do inicio da mesma
%tau= xx ;
polo=1/tau;
%Digite o valor de VCmax calculado na próxima linha e remova o % do inicio da mesma
%Vc_max= xx ;
t=linspace(0,25,1000);
vc_t=Vc_max*(exp(-t/tau));
Vc_tau=Vc_max*(exp(-1));
p1=plot(t,vc_t);hold on;
%Experimental
%Digite o valor de tau experimental na próxima linha e remova o % do inicio da mesma
%tau_ex= xx ;
%Digite o valor de VCmax calculado na próxima linha e remova o % do inicio da mesma
%Vc_max_ex= xx ;
t_ex=[0 1 2 3 4 6 8 10 12 14 16];
vc_ex_k=Vc_max_ex*exp(-t_ex/tau_ex);
vc_ex_t=Vc_max_ex*exp(-t/tau_ex);
p2=plot(t_ex,vc_ex_k, '*');
plot(t,vc_ex_t);
%Pefumaria do gráfico
ax = gca;
ax.Box = 'off';
ax.YAxisLocation = 'origin';
ax.XAxisLocation = 'origin';
ax.FontName='times';
ax.FontSize=12;
ax.TickLength=[0 0];
xlim([0 25]);
ylim([0 16]);
xticks([0 tau 25]);
xticklabels({'0', ['\it \tau = ', num2str(tau), ' s'], '\it t(s)'});
yticks([0 Vc_tau Vc_max 16]);
yticklabels({'0', ['\it v_C(\tau) = ', num2str(Vc_tau)], '\it V_{C_{max}} = 14.2', '\it{v_C(t)}'});
plot([0 tau],[Vc_tau,Vc_tau, ':');
plot([tau,tau],[0,Vc_tau, ':');
%plot([0 25],[Vc_max Vc_max, ':')
xlabel('Descarga no Capacitor');
arrowsPlot(ax,1,1);
legend([p1 p2], 'Teórico', 'Experimental', 'Location', 'best')

```

```

%Energização do Indutor

```

```

close;
clear all;
%Teoria
%Digite o valor de tau calculado na próxima linha e remova o % do inicio da mesma
%tau= xx ;

```

```

polo=1/tau;
%Digite o valor de V1_max calculado na próxima linha e remova o % do inicio da mesma
%V1_max= xx ;
t=linspace(0,12e-6,1000);
v1_t=V1_max*(exp(-t/tau));
V1_tau=V1_max*(exp(-1));
p1=plot(t,v1_t);hold on;
%Experimental
t_ex=[0 0.5 1 1.5 2 3 4 5 6 8 10]*1e-6;
%Digite os valores das tensões observadas a linha abaixo depois remova % do inicio da linha
%v1_ex=[xx xx xx xx xx xx xx xx xx xx xx];
p2= plot(t_ex,v1_ex, '*');
plot(t_ex,v1_ex);
%Perfumaria do Gráfico
ax = gca;
ax.Box = 'off';
ax.YAxisLocation = 'origin';
ax.XAxisLocation = 'origin';
ax.FontName='times';
ax.FontSize=12;
ax.TickLength=[0 0];
xlim([0 12e-6]);
ylim([0 8]);
xticks([0 tau 12e-6]);
xticklabels({'0',['\it \tau = 1.9068\mus'],'\it t(s)'});
yticks([0 V1_tau V1_max 8]);
yticklabels({'0',['\it v_L(\tau) = ',num2str(V1_tau)],'\it V_{L_{max}} = 7.25','\it{v_L(t)'}'});
plot([0 tau],[V1_tau,V1_tau],':');
plot([tau,tau],[0,V1_tau],':');
%plot([0 15e-6],[Vc_max Vc_max],':')
%xlabel('Energização do Indutor');
arrowsPlot(ax,1,1);
legend([p1 p2], 'Teórico', 'Experimental');

```

## Funções auxiliares

```

function arrowsPlot(ax,dirX,dirY)
xl=ax.XLim;
yl=ax.YLim;
y0=yl(1);
if string(ax.XAxisLocation)=='top'
    y0=yl(2);
elseif string(ax.XAxisLocation)=='bottom'
    y0=yl(1);
else
    if yl(2)*yl(1)<=0
        y0=0;
    elseif yl(1)<=0
        y0=yl(2);
    end
end
x0=xl(1);

```

```

if string(ax.YAxisLocation)=='right'
    x0=xl(2);
elseif string(ax.YAxisLocation)=='left'
    x0=xl(1);
else
    if xl(2)*xl(1)<=0
        x0=0;
    elseif xl(1)<=0
        x0=xl(2);
    end
end
end
% Fix the axes sizes
axis([xl yl]);
if dirX<0; xf=xl(1); else; xf=xl(2); end
if dirY<0; yf=yl(1); else; yf=yl(2); end
Vx=dirX*[0 0 0.05]*(xl(2)-xl(1))+xf;
Vy=dirY*[-0.01 0.01 0]*(yl(2)-yl(1))+yf;
patch(Vx,Vy, 'k', 'clipping', 'off');
Vy=dirY*[0 0 0.05]*(yl(2)-yl(1))+yf;
Vx=dirX*[-0.01 0.01 0]*(xl(2)-xl(1))+xf;
patch(Vx,Vy, 'k', 'clipping', 'off');
if (string(ax.Title.String)~='')
    if string(ax.XAxisLocation)=='top'
        pp=ax.Title.Position;
        pp(2)=yl(1);
        ax.Title.VerticalAlignment='top';
        ax.Title.Position=pp;
    elseif (yl(2)==0)&& (string(ax.XAxisLocation)=='origin')
        pp=ax.Title.Position;
        pp(2)=yl(1);
        ax.Title.VerticalAlignment='top';
        ax.Title.Position=pp;
    end
end
end
end

```