```
%% PROGETTO 3

% PUNTO 1) Modello matematico

constants;

is = 0.1;
R = 20;

isat = 1e-9; % Corrente di saturazione inversa

A = (2/3) * R * isat;
B = A + (R * is) / 3;

% PUNTO 2) Modello numerico

x_a = 0.4;
x_b = 0.5;
xx = [x_a : (x_b - x_a)/10000 : x_b];

g1 = @(x) x;
g2 = @(x) B - A * exp(x ./ Vth);
figure(1);
plot(xx, g1(xx), "-b", xx, g2(xx), "-r");
xlabel("x");
legend("g_1(x)", "g_2(x)");

g = @(x) x + A * exp(x ./ Vth) - B; % g1(x) - g2(x)
xex = fzero(g, 0);

% 3. punto fisso

Tg = @(x) Vth * log((B - x) ./ A);
dTg = @(x) -(Vth ./ (B - x));

abs(dTg(xex));

% 4.
x0 = 0;
itmax = 1000;
toll = 1e-12;
[x, niter, err] = fixed_point(x0, Tg, toll, itmax);
x_fix = x(end)

EST_ERR = err(end)
TRUE_ERR = xex - x_fix

% PUNTO 3)
C = 1e-6;
t0 = 0;
tf = 0.1;
IT = [t0, tf];
is_bar = 0.1;
```

```
f = 100;
td = 1e-2;
omega = 2*pi*f;
i_s = @(t) is_bar * sin(omega * t) * exp((-t) ./ td);

f = @(t, x) (-3*x) ./ (2*R*C) + (isat / C) * (1 - exp(x ./ Vth)) + i_s(t) ./ (2*C);
dfdx = @(t, x) (-3)/(2*R*C) - (isat/C)*exp(x ./ Vth) * (1/Vth);

NT = 1000;
tol = 1e-12;
maxit = 100;

[xm, tm] = ode15s(f, IT, 0);

[tth, xth, iter_newt] = crank_nicolson(f, dfdx, t0, tf, 0, (tf - t0)/NT);

delta_xA = max(tm) - min(tm);
delta_xB = max(xth) - min(xth);

figure(2)
plot(xm, tm, "-b", tth, xth, "-r");
legend("sol. ode15s", "sol. crank-nicholson");
```