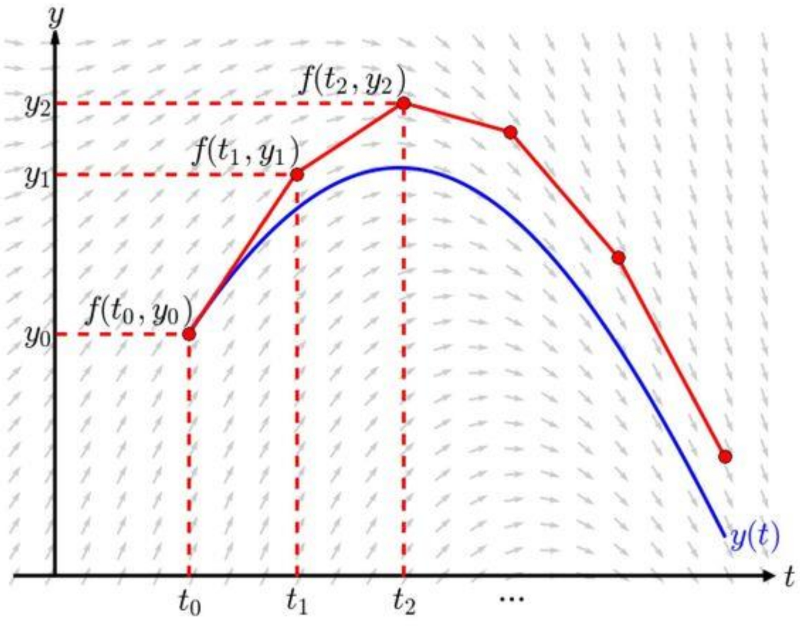
```
function [t, y] = explizitRK(F, t0, y0, T, n, methode)
% Initialisierung
    h = (T-t0)./n; %Schrittweite
    m = length(y0); %Dimension
    t = zeros(n+1,1); %Zeitintervall
    t(1) = t0;
    y = zeros(m,n); % m Dimensionen, n Werte
    y(:,1) = y0;

    switch methode %Fallunterscheidung der Verfahren
        case 'Euler'
            for i = 1:n
                t(i+1) = t(i)+h;
                y(:,i+1) = y(:,i) + h*F(t(i),y(:,i));
            end

        case 'Heun2'
            for i = 1:n
                t(i+1) = t(i)+h;
                k1 = F(t(i),y(:,i));
                k2 = F(t(i)+h,y(:,i)+h*k1);
                y(:,i+1) = y(:,i) + h/2*(k1+k2);
            end

        case 'Kutta3'
            for i = 1:n
                t(i+1) = t(i)+h;
                k1 = F(t(i),y(:,i));
                k2 = F(t(i)+h/2,y(:,i)+h/2*k1);
                k3 = F(t(i)+h,y(:,i)-h*k1+2*h*k2);
                y(:,i+1) = y(:,i) + h/6*(k1+4*k2+k3);
            end
        case 'RK4'
            for i = 1:n
                k1 = F(t(i),y(:,i));
                k2 = F(t(i)+h/2,y(:,i)+h/2*k1);
                k3 = F(t(i)+h/2,y(:,i)+h/2*k2);
                k4 = F(t(i)+h,y(:,i)+h*k3);
                t(i+1) = t(i)+h;
                y(:,i+1) = y(:,i) + h/6*(k1+2*k2+2*k3+k4);
            end
    end


end
```

Euler Verfahren: Explizit

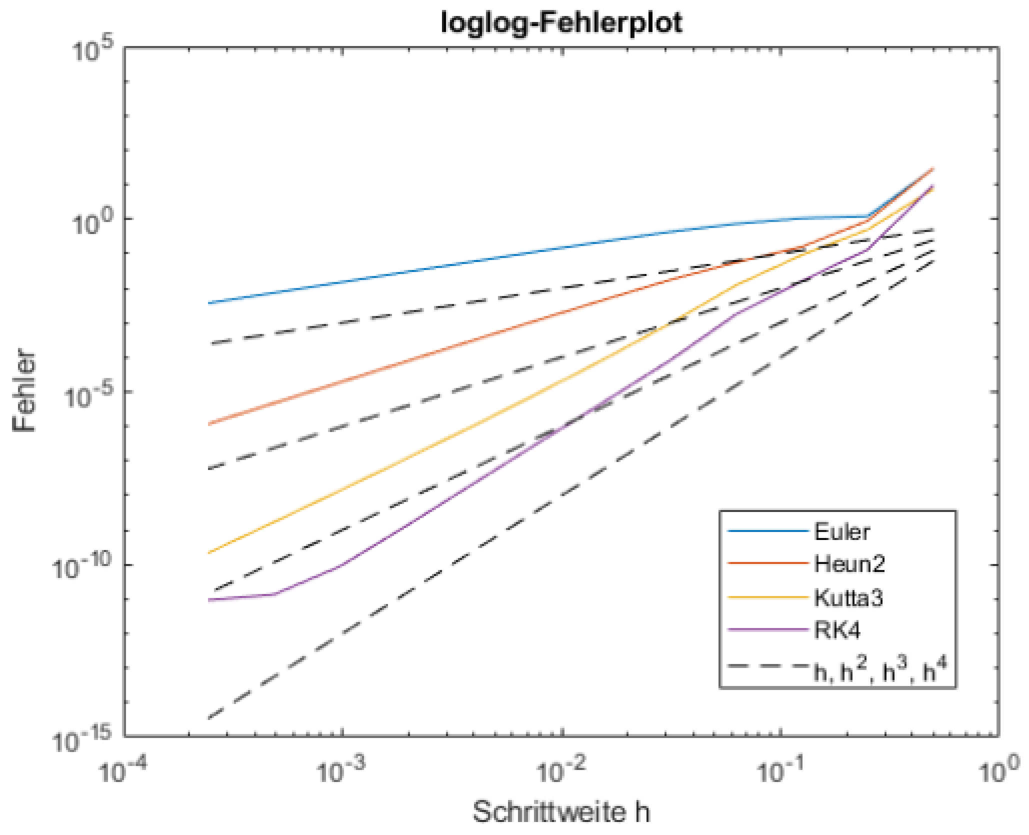
$$\left.\begin{aligned} \frac{\mathrm{d}y}{\mathrm{d}t} &= f(t, y(t)) \\ y(t_0) &= y_0 \end{aligned}\right\}$$



$$\frac{y_{j+1} - y_j}{\Delta t} = f(t_j, y_j)$$

$$y_{j+1} = y_j + \Delta t f(t_j, y_j)$$

```matlab
% Initialisierung: Intervall [t0,T], Anfangswert y0
t0 = 1;
y0 = [4 2 0].';
T = 3;

% Initisalisierung: Funktion F, analytische Lsg y_exact
F = @(t,y) [atan(y(1)^3+2*t)+cos(y(3)), y(2)^2-t*y(1), 1/t*exp(atan(y(3)^2))+
    exp(-y(2)^2)+13*cos(4*pi*t)].';
y_exact = [6.986845270192330; -4.461339962786163; 3.28773579037];
err = zeros(12,4); %
h = zeros(1,12); %

for i = 2:13
    n = 2^i; %2^2, 2^3, ..., 2^13
    h(i-1) = (T-t0)/n; % Schrittweite

    % Loesungsverfahren [Zeitpunkt t, Approximation y]
    [t,yEuler] = explizitRK(F, t0, y0, T, n, 'Euler');
    [t,yHeun2] = explizitRK(F, t0, y0, T, n, 'Heun2');
    [t,yKutta3] = explizitRK(F, t0, y0, T, n, 'Kutta3');
    [t,yRK4] = explizitRK(F, t0, y0, T, n, 'RK4');

    ind = find(t == 3); % Approximation fuer t =3 (ind = Spaltenindex)
    % Alternativ in diesem Fall: ind = length(t);

    %2-Norm des Fehlers zwischen Approximation & analytische Lsg
    err(i-1,1) = norm(y_exact-yEuler(:,ind));
    err(i-1,2) = norm(y_exact-yHeun2(:,ind));
    err(i-1,3) = norm(y_exact-yKutta3(:,ind));
    err(i-1,4) = norm(y_exact-yRK4(:,ind));
end

% loglog-Plots (beide Axen logarithmisch skaliert)
loglog(h,err(:,1),h,err(:,2),h,err(:,3),h,err(:,4),h,h,'k--',h,h.^2,'k--',h,h
    .^3,'k--',h,h.^4,'k--')
title('loglog-Fehlerplot')
legend('Euler','Heun2','Kutta3','RK4','h,␣h^2,␣h^3,␣h^4')
xlabel('Schrittweite␣h')
ylabel('Fehler')
```



$$\mathrm{d}_t\, \mathbf{y}(t) = \mathbf{A}\mathbf{y}(t) - \mathbf{B}\mathbf{D}^{-1}\Big(\mathbf{C}\mathbf{y}(t) + \mathbf{w}(t)\Big) + \mathbf{v}(t) \quad \text{für } t \in [t_0, T]$$

```matlab
function [t, y, z] = explizitRKlseDAG(A, B, C, D, v, w, t0, y0, T, n, methode)
    factor = B*D^-1; % Matrixprodukt
    dy = @(t,y) A*y-factor*(C*y+w(t))+v(t); % Differentialgleichung als Function-Handle

    [t,y] = explizitRK(dy, t0, y0, T, n, methode); % Approximation der Lsg
    m = length(y);

    z = zeros(length(D),m); % Spaltenweise Berechnung von (*) der Vektoren z_i
    for i =1:m
        z(:,i) = -inv(D)*(C*y(:,i) + w(t(i))); %(*) z = -D^-1*(C*y+w(t));
    end
end
```

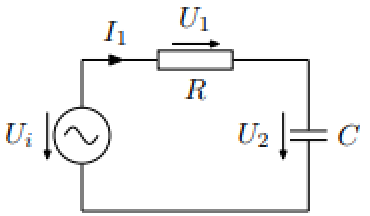


Abbildung 1: RC-Tiefpass-Filter

$$\mathrm{d}_t\, U_2(t) = 0U_2(t) + \begin{bmatrix} 0 & C^{-1} \end{bmatrix} \begin{bmatrix} U_1(t) \\ I_1(t) \end{bmatrix} + 0,$$

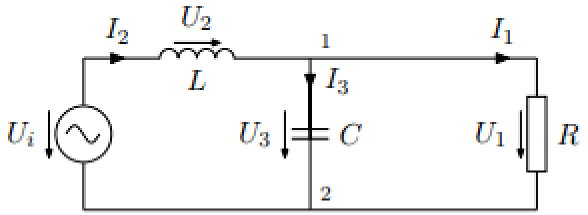
$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} U_2(t) + \begin{bmatrix} 1 & -R \\ 1 & 0 \end{bmatrix} \begin{bmatrix} U_1(t) \\ I_1(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -U_i(t) \end{bmatrix}.$$

$$\mathbf{y}(t) := U_2(t) \quad \text{und} \quad \mathbf{z}(t) := \begin{bmatrix} U_1(t) \\ I_1(t) \end{bmatrix}$$

```matlab
function [t, y, z] = tiefpassRC(R, Cap, Ui, t0, y0, T, n, methode)
    A = 0;
    B = [0, Cap^-1];
    v = @(t) 0;
    C = [0; 1];
    D = [1 -R; 1 0];
    w = @(t) [0; -Ui(t)];

    [t, y, z] = explizitRKlseDAG(A, B, C, D, v, w, t0, y0, T, n, methode);
end
```

$$\mathrm{d}_t \begin{bmatrix} U_3(t) \\ I_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_3(t) \\ I_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & C^{-1} \\ 0 & L^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} U_1(t) \\ U_2(t) \\ I_1(t) \\ I_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} U_3(t) \\ I_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 & -R & 0 \\ 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_1(t) \\ U_2(t) \\ I_1(t) \\ I_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -U_i(t) \end{bmatrix}$$



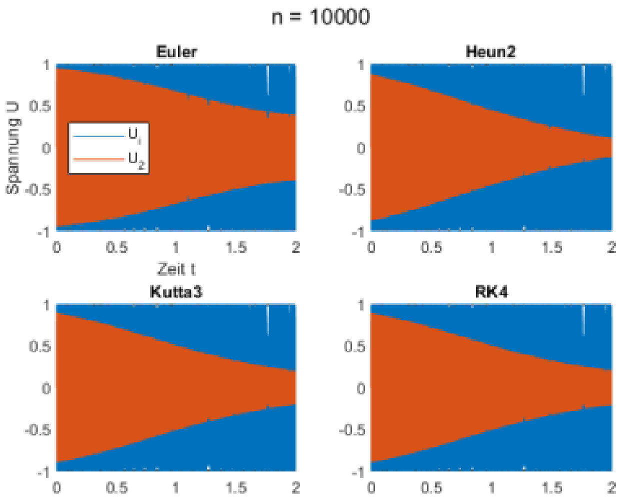
$$\mathbf{y}(t) := \begin{bmatrix} U_3(t) \\ I_2(t) \end{bmatrix} \quad \text{und} \quad \mathbf{z}(t) := \begin{bmatrix} U_1(t) \\ U_2(t) \\ I_1(t) \\ I_3(t) \end{bmatrix}$$

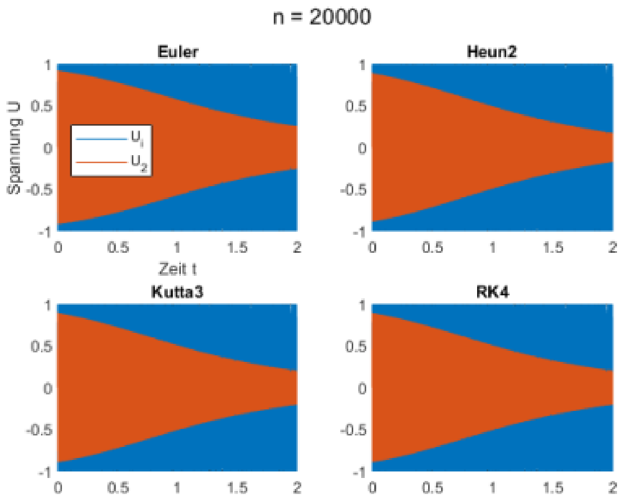
Abbildung 2: RLC-Tiefpass-Filter

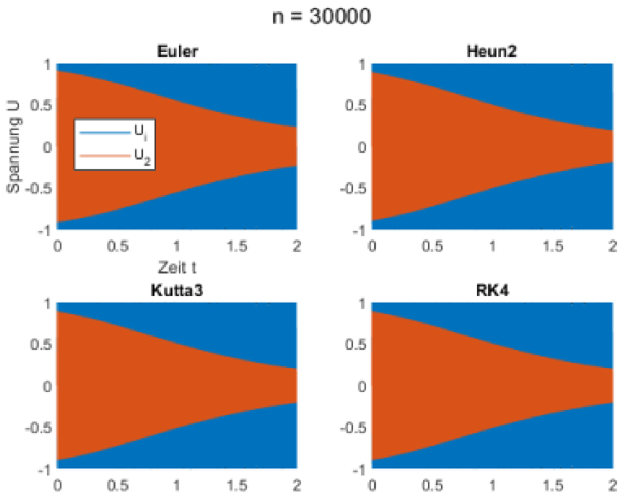

```matlab
function [t, y, z] = explizitRKlseDAG(A, B, C, D, v, w, t0, y0, T, n, methode)
    factor = B*D^-1; % Matrixprodukt
    dy = @(t,y) A*y-factor*(C*y+w(t))+v(t); % Differentialgleichung als Function-Handle

    [t,y] = explizitRK(dy, t0, y0, T, n, methode); % Approximation der Lsg
    m = length(y);

    z = zeros(length(D),m); % Spaltenweise Berechnung von (*) der Vektoren z_i
    for i =1:m
        z(:,i) = -inv(D)*(C*y(:,i) + w(t(i))); %(*) z = -D^-1*(C*y+w(t));
    end
end
```


$$\mathbf{z}(t) = -\mathbf{D}^{-1}\Big(\mathbf{C}\mathbf{y}(t) + \mathbf{w}(t)\Big)$$

```matlab
for k = 1:3 % n = 10000, 20000, 30000

%Initialisierung
n = k*10000; % Anzahl Datenpunkte: n+1
R = 0.5; % Widerstand
Cap = 20.23e-4; % Kapazitaet des Kondensators
t0 = 0; % linker Randpunkt
T = 2; % rechter Randpunkt
Ui = @(t) sin(2*pi*t.*(220*2.^t)); % zugefuehrte Spannung
y0 = Ui(0); % Anfangswert

[t, yEuler, z] = tiefpassRC(R, Cap, Ui, t0, y0, T, n, 'Euler');
[t, yHeun2, z] = tiefpassRC(R, Cap, Ui, t0, y0, T, n, 'Heun2');
[t, yKutta3, z] = tiefpassRC(R, Cap, Ui, t0, y0, T, n, 'Kutta3');
[t, yRK4, z] = tiefpassRC(R, Cap, Ui, t0, y0, T, n, 'RK4');



figure

subplot(2,2,1)
plot(t,Ui(t),t,yEuler)
title('Euler')
xlabel('Zeit␣t')
ylabel('Spannung␣U')
legend('U_i','U_2')


subplot(2,2,2)
plot(t,Ui(t),t,yHeun2)
title('Heun2')

subplot(2,2,3)
plot(t,Ui(t),t,yKutta3)
title('Kutta3')

subplot(2,2,4)
plot(t,Ui(t),t,yRK4)
title('RK4')

sgtitle(['n␣=␣', num2str(k*10000)])
end
```







```matlab
for k = 1:3 % n = 10000, 20000, 30000

n = k*10000;
R = 0.5;
L = 3.61e-4;
Cap = 3.61e-4;
t0 = 0;
T = 2;
Ui = @(t) sin(2*pi*t.*(220*2.^t));
y0 = [0 0];

[t, yEuler, z] = tiefpassRLC(R, L, Cap, Ui, t0, y0, T, n, 'Euler');
[t, yHeun2, z] = tiefpassRLC(R, L, Cap, Ui, t0, y0, T, n, 'Heun2');
[t, yKutta3, z] = tiefpassRLC(R, L, Cap, Ui, t0, y0, T, n, 'Kutta3');
[t, yRK4, z] = tiefpassRLC(R, L, Cap, Ui, t0, y0, T, n, 'RK4');

figure

subplot(2,2,1)
plot(t,Ui(t),t,yEuler(1,:)) %nur erste Zeile weil yEuler = [U3,I2]
title('Euler')
xlabel('Zeit␣t')
ylabel('Spannung␣U')
legend('U_i','U_3')


subplot(2,2,2)
plot(t,Ui(t),t,yHeun2(1,:))
title('Heun2')

subplot(2,2,3)
plot(t,Ui(t),t,yKutta3(1,:))
title('Kutta3')

subplot(2,2,4)
plot(t,Ui(t),t,yRK4(1,:))
title('RK4')

sgtitle(['n␣=␣', num2str(k*10000)])

end
```